Generación Automática de Resúmenes de Un Solo Documento Basada en Algoritmos Meméticos

Clara Inés Noguera Solano
Jeimmy Susana Bonilla Méndez

Director: Ph.D. (c) Martha Eliana Mendoza Becerra
Asesor: Ph.D. (c) Carlos Alberto Cobos

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de I+D en Tecnologías de la Información (GTI)
Línea Investigación: Gestión de la Información, Recuperación de la Información
Popayán, Mayo de 2013

A nuestro Padre celestial, motor de nuestro existir, a nuestras familias, regalo maravilloso de Dios, y a todos aquellos que de una u otra forma hicieron suyos nuestros sueños.

Agradecimientos

Los autores expresan sus agradecimientos a:

Nuestro Padre Dios, a quien debemos todas nuestras realizaciones, por darnos la inspiración y fortaleza que nos permitieron superar toda dificultad y llevar a buen término nuestro trabajo, quien nos ha guiado para lograr todas las metas, que alguna vez se creyeron imposibles, pero que hoy, gracias a sus bendiciones y amor inigualable, son una hermosa realidad.

Nuestras familias, especialmente a nuestros padres María del Carmen, Carlos Arturo y María Oliva, quienes con su esfuerzo, comprensión y apoyo incondicional nos formaron e hicieron posible esta meta, y a quienes no nos alcanzaría la vida para retribuirles su generosidad y su amor.

Martha Mendoza y Carlos Cobos, por darnos la oportunidad de aprender y crecer en su grupo de trabajo, y a todos los profesores que nos orientaron a lo largo de nuestra carrera para formarnos como profesionales.

Nuestros amigos y compañeros de carrera, testigos también de nuestros esfuerzos, por su amistad, su apoyo y por los momentos gratos que compartimos, y a todos aquellos que nos acompañaron en este trayecto.

Contenido

Lista de tablas

Lista de figuras

Presentación

Los avances tecnológicos y el constante aumento de la información presente en la Web son factores que hacen cada vez más exigente la extracción e interpretación de la información por parte de los usuarios. En ese sentido, la generación automática de resúmenes ha sido explorada como una alternativa para enfrentar este problema, y su objetivo es identificar las ideas más importantes de un texto, presentando al lector una versión compacta del mismo. De esta manera, numerosos enfoques automáticos de generación de resúmenes han sido evaluados, sin embargo, lograr resultados que se asemejen completamente a extractos creados manualmente es una labor difícil, pues la producción de un resumen humano es un proceso de extracción y comprensión altamente complejo y subjetivo que trasladado a un nivel computacional requiere de cuidadosos análisis estadísticos y heurísticos. Por tal razón, la búsqueda de mayor calidad de los resúmenes generados es el aspecto que motiva la realización de nuevas investigaciones en la generación automática de resúmenes.

En este documento, se presenta un algoritmo memético que permite la generación de resúmenes automáticos de un solo documento. A lo largo del texto, se describen las bases teóricas y el proceso de desarrollo realizado.

En el capítulo 1 se presenta la problemática que motivó el planteamiento de este proyecto, la justificación de desarrollo del mismo, los objetivos propuestos y los principales resultados obtenidos.

En el capítulo 2 se exponen las bases teóricas necesarias para la realización de este proyecto. En ese sentido, se presentan los conceptos básicos y principales investigaciones en el área de generación de resúmenes, los métodos más conocidos de evaluación de resúmenes, aspectos clave en la representación de documentos y ponderación de sus términos, algunas medidas de similitud y la descripción general de los algoritmos meméticos y operadores reproductivos y de búsqueda local.

En el capítulo 3 se presenta el proceso llevado a cabo para el desarrollo del algoritmo memético propuesto, realizando una descripción de cada uno de los ciclos de desarrollo.

El capítulo 4 describe el Algoritmo Memético propuesto en este proyecto de grado para la generación automática de resúmenes de un solo documento.

El capítulo 5 expone las conclusiones obtenidas durante y después del proceso de desarrollo y plantea algunas ideas que pueden considerarse para trabajos futuros.

Capítulo 1

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

El gran volumen de información textual disponible en internet, a través de diversos almacenes de documentos electrónicos, y su crecimiento continuo y acelerado, ha hecho cada vez más difícil la tarea de abordar plenamente el contenido de los documentos e identificar la información más relevante, según la necesidad, involucrando mayor gasto de tiempo y esfuerzo para el usuario. La generación automática de resúmenes, enmarcada en el área de recuperación de información, ha sido considerada como una solución de este problema, al facilitar al usuario la rápida comprensión del eje temático del(os) documento(s) considerado(s), al ofrecerle un texto comprimido que contemple el tema principal de éstos, sin necesidad de leer completamente su contenido.

De esta forma, la generación automática de resúmenes de uno o múltiples documentos, ha captado la atención de varias investigaciones, intentando encontrar un método que permita realizar la tarea planteada, con resultados cada vez de mejor calidad. De esta forma, son muchos los acercamientos que hasta el momento han procurado resolver la problemática que se encierra en la generación automática de resúmenes, utilizando diferentes criterios o estrategias para obtener las oraciones más relevantes de un documento, como por ejemplo la frecuencia de términos [1], la posición de la oraciones en el documento [2], la presencia de palabras clave o palabras del título del documento en las oraciones [3], aprendizaje automático [4-10], conectividad de texto [11-13], grafos iterativos [14-16], y reducción algebraica [17-19].

Por otro lado, los métodos evolutivos han sido considerados también como una alternativa de solución a la problemática presentada, obteniendo muy buenos resultados en comparación con otros métodos del estado del arte. Entre este tipo de métodos, se han evaluado métodos basados en enjambres [20-22], algoritmos genéticos [6, 23, 24], búsqueda armónica [25], programación genética [26] y evolución diferencial [27]. Sin embargo, existe otro tipo de enfoque evolutivo que, hasta el momento de iniciar esta investigación, aún no ha sido examinado en el área de generación automática de resúmenes y son los algoritmos meméticos. Esta meta-heurística integra la búsqueda basada en poblaciones, propia de los algoritmos evolutivos primitivos, con la mejora local, que intenta la optimización individual de las soluciones [28]. Así mismo, una característica fundamental y determinante en el rendimiento de estos algoritmos, es la incorporación de conocimiento específico del problema tratado [29].

De este modo, considerando el buen desempeño registrado de los algoritmos meméticos en la resolución de problemas complejos de optimización, su aplicación en la generación automática de resúmenes resulta prometedora. Por esta razón, en el presente trabajo investigativo, se propone un algoritmo para generación automática de resúmenes extractivos de un solo documento, basado en un algoritmo memético, que permita la selección de las oraciones más relevantes de un documento para formar el resumen. Se

espera, entonces, que este nuevo enfoque de la generación automática de resúmenes, conduzca a mejores soluciones que las obtenidas hasta el momento con otros métodos evolutivos.

Conforme a la taxonomía de resúmenes presentada en [30], con el presente trabajo de investigación se busca la generación de resúmenes *extractivos de un solo documento*, que contengan secuencias de palabras tomadas directamente del texto original, además se busca que sean *indicativos*, al presentar información abreviada de los principales temas del documento, *de nivel superficial*, que empleen características superficiales para representar la información y que, finalmente, sean *mono-lenguaje genéricos*, en inglés, sin orientación hacia una audiencia o consulta en particular.

El mejoramiento en la calidad de los resúmenes obtenidos es uno de los aspectos que aún demanda la atención de las investigaciones de generación automática de resúmenes, de esta forma, mediante la producción de conocimiento exploratorio y descriptivo, se pretende dar respuesta a la pregunta: ¿Es posible generar resúmenes automáticos de un solo documento desde la perspectiva de un algoritmo memético, que permita obtener resúmenes de mayor calidad a los establecidos en el estado del arte con modelos evolutivos?

1.2 JUSTIFICACIÓN

El rápido aumento de la información textual disponible en la Web y el surgimiento de nuevos avances tecnológicos que incrementan las capacidades de almacenamiento y posibilitan el acceso a la información de un mayor número de usuarios a través de nuevos dispositivos portables, origina la necesidad de ajustar tal información para que sea procesada más fácilmente por el usuario final, sin perder los aspectos más importantes presentados en ella. De esta manera, la generación automática de resúmenes de texto, busca automatizar dicha labor, produciendo textos cortos que presenten al lector las ideas más relevantes de un documento, ahorrándole tiempo y esfuerzo de extracción y comprensión, y proporcionando una versión más adaptable visualmente a diversos dispositivos.

Existen varios acercamientos en la generación de resúmenes automáticos que han buscado mejorar la calidad de la salida generada, con el propósito de que sea lo más similar posible a un resumen producido manualmente. En tal sentido, la investigación en esta área aún despierta mucho interés. Diferentes estrategias han sido aplicadas y, entre ellas, los enfoques evolutivos han proporcionado resultados sobresalientes. Sin embargo, a pesar de los buenos resultados en la resolución de problemas de optimización, uno de los acercamientos evolutivos no explorados en esta área del Procesamiento de Lenguaje Natural son los Algoritmos Meméticos.

En la presente investigación, se propone un sistema de generación de resúmenes de un solo documento basado en un Algoritmo Memético. Para la creación de este sistema se consideraron varios aspectos de diseño del algoritmo como son el estudio de las características de evaluación que conformarían la función objetivo, estudio de operadores reproductivos y de optimización local, definición de un esquema inicial del algoritmo, definición experimental del esquema final del algoritmo, definición experimental de la función objetivo, afinación de parámetros y afinación de pesos de la función objetivo. Este

proceso fue realizado a través de herramientas tecnológicas cuya disponibilidad y documentación son provistas por la Universidad del Cauca, gracias al acuerdo MSDN Academic Alliance[1], y cuyas funcionalidades han sido amplia y exitosamente utilizadas dentro del Grupo de Tecnologías de la Información (GTI) de la institución en proyectos de Recuperación de Información.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Proponer un algoritmo de generación automática de resúmenes extractivos de un solo documento basado en un algoritmo memético.

1.3.2 OBJETIVOS ESPECÍFICOS

- Establecer la representación de los agentes y las características, estadísticas y de similitud, que harán parte de la función objetivo, buscando que las oraciones del resumen tengan en cuenta los tópicos principales contenidos en el documento.

- Especificar los esquemas de selección, cruce, mutación y reemplazo para el algoritmo memético, que permitan realizar una adecuada exploración del espacio de búsqueda de la solución y mantener la diversidad de los agentes.

- Determinar el algoritmo de búsqueda local apropiado para el problema de generación automática de resúmenes de un solo documento, que permita realizar una adecuada explotación de la vecindad de los agentes en el algoritmo memético.

- Evaluar la calidad de los resúmenes generados con el algoritmo propuesto, mediante ROUGE, y comparar los resultados con algoritmos basados en modelos evolutivos, utilizando documentos de noticias de la Conferencia de Entendimiento del Documento.

1.4 RESULTADOS OBTENIDOS

- *Monografía del trabajo de grado*. En ella se presentan los conceptos teóricos necesarios para el desarrollo del proyecto, el proceso de diseño, implementación y ajuste de parámetros del algoritmo de generación de resúmenes extractivos de un solo documento basado en algoritmos meméticos, los resultados de la evaluación de la calidad de los resúmenes generados por el algoritmo propuesto y la comparación con otros algoritmos basado en modelos evolutivos, conclusiones y recomendaciones para trabajos futuros.

- Código fuente del algoritmo memético de generación automática de resúmenes para un solo documento.

- Artículo para publicar en revista o evento nacional o internacional, que exponga los resultados obtenidos en este proyecto de investigación.

[1] Acuerdo que vincula a Microsoft con entidades educativas universitarias, en la cual se permite tener acceso a software de desarrollo con propósitos académicos.

Capítulo 2

2 CONTEXTO TEÓRICO

A lo largo de este capítulo se exponen las bases teóricas necesarias para la realización de este trabajo. En ese sentido, se presentan los conceptos básicos y principales investigaciones en el área de generación de resúmenes, los métodos más conocidos de evaluación de resúmenes, aspectos clave en el pre-procesamiento y representación de documentos y ponderación de sus términos, algunas medidas de similitud y la descripción general de los algoritmos meméticos y operadores reproductivos y de búsqueda local.

2.1 GENERACIÓN AUTOMÁTICA DE RESÚMENES DE TEXTO

2.1.1 Definición

En las investigaciones realizadas dentro de la generación automáticamente de resúmenes de texto, un *resumen* es entendido como un "texto producido a partir de uno o más documentos, que contiene información importante de los mismos, y que no es más extenso que la mitad del documento(s) original(es), siendo usualmente aún más corto" [31], también puede ser tomado como una "transformación del texto fuente a través de la reducción de su contenido al seleccionar lo que es importante en él" [32]. De esta forma, en este trabajo, se toma la definición de un *resumen* como un texto breve que contiene la información más importante de uno o varios documentos.

En ese sentido, la *generación automática de resúmenes*, como un área de la recuperación de información, cuyo estudio surge a finales de la década de los 50, tiene como propósito la creación automática de un resumen, por medio de un programa de computadora, a partir de la identificación de la información importante de uno o varios documentos [9].

2.1.2 Taxonomía

Los resúmenes pueden ser clasificados de acuerdo a diferentes aspectos [30]. Uno de ellos es según el tipo de información que contienen, en donde se distinguen el resumen descriptivo, en el que se detalla la forma y el contenido del texto de origen; el evaluativo, que ofrece algún tipo de respuesta crítica; el indicativo, que informa sobre los principales temas del documento original, ayudando al usuario a decidir si lo lee o no; y el informativo, que proporciona un resumen del documento completo conservando los detalles importantes.

Otra clasificación es según la cantidad de documentos que se procesan, en donde se diferencian el de un documento, el cual condensa un solo documento en una representación más corta, y el de múltiples documentos, que se utiliza para resumir la información contenida en un conjunto de documentos.

Otro aspecto de clasificación es de acuerdo a la audiencia a la que va dirigido, dentro de la que se reconocen el orientado a consulta, que presenta la información más relevante con respecto a la consulta del usuario; el genérico, que da una idea general del contenido del documento; y el enfocado en el usuario, que busca dar respuesta a las necesidades específicas del usuario.

Una forma adicional de categorización es con base en el nivel de procesamiento, distinguiéndose el resumen superficial, que representa el documento utilizando características poco profundas como posicionamiento de frases, frases claves, etc.; el profundo, en donde, para su construcción, se requiere algunas técnicas como por ejemplo el procesamiento de lenguaje natural.

Finalmente, un resumen puede clasificarse, de acuerdo a su forma, en extractivo, constituido por la reutilización de porciones del texto original; y abstractivo, que es más complejo puesto que requiere herramientas de análisis lingüístico para construir nuevas frases a partir de las extraídas, ya que se basa en resúmenes abstractos, cuya principal característica es que incluyen oraciones nuevas que no están necesariamente contenidas en el texto original [20, 30].

2.1.3 Métodos de generación automática de resúmenes de un solo documento

A continuación se presentan los principales métodos a través de los cuales ha sido abordada la generación automática de resúmenes, iniciando desde las primeras técnicas básicas de selección de oraciones hasta estrategias más elaboradas basadas en aprendizaje de máquina, grafos, conectividad de texto, reducción algebraica y algoritmos evolutivos.

2.1.3.1 Primeros métodos

Una de las investigaciones pioneras en el área de generación automática de resúmenes sugiere la selección de las oraciones más importantes en un documento a partir del análisis de sus términos [1]. Es así como el autor muestra la frecuencia de aparición de un término en un texto fuente, y su posición relativa dentro de una oración, como factores relevantes para su puntuación. El procedimiento inicia filtrando pronombres, preposiciones y artículos y continúa con una normalización de términos ordenados alfabéticamente, con el fin de llevarlos a su lexema[2] y, entonces, calcular la ocurrencia de términos similares, desechando aquellos con los valores más bajos de frecuencia. Estos resultados establecen el punto de entrada para la puntuación de las oraciones y la selección de aquellas que constituirán el resumen.

Un trabajo posterior planteó tres factores adicionales de significancia de las oraciones [3]: las expresiones clave, la aparición de palabras en el título y la posición de la frase en el documento. En el primero, la puntuación de una oración está sujeta a la aparición de palabras como “importante”, “difícilmente”, “en conclusión”, entre otras, que se encuentran en un listado de este tipo de palabras clave llamado “cue dictionary”. En el segundo, la aparición de las palabras de la frase en el título o subtítulos aumenta su peso. En el tercero, el valor de una frase se ve influenciado por su posición en el documento o párrafo

[2] Un lexema puede definirse como la mínima unidad de significación de una palabra

y su cercanía a los subtítulos. De estas tres técnicas, la basada en la posición, presentó los resultados más sobresalientes. Igualmente, cada uno de estos métodos fue comparado con el propuesto por Luhn [1], llegando a la conclusión de que ofrecían mejores resúmenes. Por su parte, la conjugación de los tres nuevos acercamientos fue mejor que cualquier otro enfoque.

2.1.3.2 Métodos basados en aprendizaje de máquina

Uno de los primeros esfuerzos por incluir el aprendizaje de máquina en la generación de resúmenes automáticos partió de un conjunto de documentos de entrenamiento junto a su resumen manual correspondiente y desarrolló una función de clasificación, con base en el teorema bayesiano, que estima la probabilidad de que una oración sea incluida en un resumen [7, 33]. De esta manera, las oraciones eran clasificadas según su probabilidad, formando parte del resumen sólo aquellas con el puntaje más alto.

Así mismo, existen otras investigaciones que, en forma similar, han hecho uso del clasificador Naive Bayes para la creación automática de resúmenes, incorporando nuevas características para las oraciones. Los resultados de algunos de estos trabajos son presentados en [4, 9, 10]. Por otro lado, se propone un clasificador basado en redes neuronales para la generación automática de resúmenes [6, 8], entrenando un modelo de clasificaciones y características para las oraciones de un documento, que pueda posteriormente inferir la categoría apropiada de oraciones para un documento de prueba. A diferencia de las anteriores técnicas, se plantea un enfoque basado en el modelo Hidden Markov (HMM) [5, 34], cuya característica principal es el reconocimiento de dependencias locales entre oraciones a través de un modelo secuencial. En este acercamiento, las características de las oraciones son pocas, y a partir de ellas se calcula una probabilidad a posteriori de que cada oración esté incluida en un resumen. Las desventajas comunes en estos métodos son la necesidad de datos de entrenamiento, así como su dependencia del lenguaje de entrenamiento y de las características empleadas en la extracción de oraciones.

2.1.3.3 Métodos basados en grafos

Los algoritmos basados en grafos son métodos no supervisados cuya tarea principal es resolver la importancia de un vértice dentro de un grafo, a partir de la información global extraída recursivamente de todo el grafo [14].

En la generación de resúmenes automáticos, las secuencias de una o más unidades léxicas extraídas de un texto, y las relaciones entre ellas, constituyen los vértices y arcos del grafo que representa el documento. En [15], los arcos muestran las relaciones de similitud entre oraciones, calculadas con respecto al contenido común en ellas. De esta forma, los pesos asociados a los arcos indican la fortaleza de las conexiones entre oraciones y sirven como punto de partida para el ordenamiento en el grafo. Aquellas frases o nodos con mayor puntuación son seleccionados para su inclusión en el resumen extractivo. Desde una perspectiva similar, existe un enfoque que intenta la unificación de metodologías sintáctica, semántica y estadística, reflejando además la importancia del encabezado de texto a través de la presencia de palabras clave en las oraciones. De esta forma, el puntaje de cada oración es obtenido tras la combinación de TextRank [14], WordNet y un factor de posición. El algoritmo TextRank, usa como medida de peso la

cantidad de nombres y adjetivos comunes entre las oraciones. Por su parte para la utilización de WordNet se da prioridad más alta a las oraciones que contienen las conjunciones y palabras más usuales del documento, mientras que el cálculo de la posición de una oración utiliza una fórmula basada en el modelo de regresión lineal: $P(S_i) = -19\ ln(d_i) + 51.926$, donde d_i representa la distancia de la oración S_i al título o encabezado del documento [35].

Por otra parte, en [16], se presenta un acercamiento que pretende mejorar la calidad de los resúmenes a través del uso de algoritmos de desambigüación léxica basados en grafos semánticos.

Existe además un enfoque particular basado en grafos, en el cual la generación automática del resumen de un solo documento se realiza al mismo tiempo que la generación del resumen de múltiples documentos [36]. De esta forma, se hace uso de una importancia local, que indica la relevancia de una oración dentro de un documento en la generación del resumen de un solo documento, y de una importancia global, que indica la relevancia de la misma oración pero a nivel de todo el conjunto de documentos en la generación del resumen para múltiples documentos. Estas medidas de importancia pueden influenciarse mutuamente, de esta forma, se dice que si una oración es sobresaliente en el conjunto de documentos, puede serlo también en un documento particular de ese conjunto y si una oración es sobresaliente en un documento en particular puede serlo para el conjunto de documentos completo. De esta manera, se construyen cuatro grafos de afinidad que reflejan los diferentes tipos de relaciones de similitud coseno entre oraciones y que servirán posteriormente para calcular, en forma iterativa y recursiva, los puntajes de importancia local y global de cada oración. Los algoritmos convergen, obteniendo los puntajes de importancia local y global de cada oración, de tal forma que las oraciones con importancia local más alta son escogidas para formar el resumen de un solo documento, mientras que las oraciones con importancia global más alta formarán el resumen de múltiples documentos. Finalmente, se utiliza un algoritmo voraz para eliminar redundancia y seleccionar las oraciones nuevas e informativas del resumen.

La ventaja principal de los métodos basados en grafos radica en su independencia del lenguaje y su fácil adaptación a uno en particular, mientras que el aumento de la complejidad computacional a medida que crece el número de nodos y arcos, constituye su mayor desventaja.

2.1.3.4 Métodos basados en conectividad del texto

La identificación de relaciones entre conceptos en un texto, es uno de los mayores desafíos en los métodos extractivos de generación de resúmenes automáticos. Como respuesta a este problema, varios enfoques han sido estudiados con el fin de establecer las conexiones que puedan existir entre diversas partes de un texto, y obtener, así, resúmenes más coherentes y fáciles de comprender. La teoría de estructura retórica (RST por sus siglas en inglés Rethorical Structure Theory) ha sido empleada en algunos métodos como herramienta para la organización del texto. En ella, se propone la representación de las relaciones retóricas entre partes de un texto a partir de una estructura de árbol. De este modo, las unidades de texto constituyen los nodos, que pueden ser clasificados como núcleo o satélite, según el grado de relevancia para el discurso. Los nodos núcleo representan unidades de texto que expresan los aspectos

más importantes del documento, mientras que los nodos satélite suponen partes menos centrales. En [12], los nodos que participan en cada relación retórica son penalizados de acuerdo a su importancia relativa, con el fin de determinar los segmentos de texto más relevantes. Por su parte, en [13] se hace uso de una estrategia de promoción, en la que las unidades más representativas del texto original son ascendidas más cerca de la raíz del árbol. Otro enfoque enmarcado dentro de las técnicas basadas en conectividad de texto es el denominado *cadenas léxicas*. En [11] se presenta un método que, valiéndose del tesauro de WordNet[3], determina las relaciones léxicas entre términos para formar cadenas entre expresiones conexas. Las relaciones más comúnmente codificadas son las de homonimia, sinonimia, antonimia, repetición, holonimia e hipernimia. En este método, el procedimiento para la generación de resúmenes inicia con la segmentación del texto original y continúa con la construcción de las cadenas léxicas. La identificación de las cadenas más fuertes y la extracción de las oraciones más significativas finalizan el proceso de producción del resumen.

Unas de las desventajas más notorias en estos métodos son su necesidad de técnicas complejas y su dependencia del lenguaje para el procesamiento del texto.

2.1.3.5 Métodos basados en reducción algebraica

El enfoque de reducción algebraica más utilizado dentro de la generación automática de resúmenes de texto es el basado en Análisis Semántico Latente (LSA por sus siglas en inglés Latent Semantic Analysis) [37], el cual es un método para extraer, representar y comparar significados de palabras mediante el análisis algebraico-estadístico de un texto, cuya hipótesis básica es que el significado de una palabra está determinado por su aparición frecuente junto a otras palabras. Gong y Liu [38] propusieron usar LSA para la generación automática de resúmenes genéricos, aplicando la descomposición de valores singulares (SVD). El proceso de análisis semántico está compuesto por dos pasos. El primero es la creación de una matriz de términos por oración $A = [A_1, A_2, \dots, A_n]$, donde cada columna A_i representa el vector de pesos, basado en frecuencia de términos, de la oración i en el documento. El siguiente paso consiste en aplicar la descomposición en valores singulares (SVD) a la matriz A, de tal forma que quede descompuesta en tres matrices U, D y V, de modo que las columnas de la primera son llamadas *vectores singulares de izquierda*, los elementos diagonales de la segunda son *valores singulares no negativos* en orden descendente y las columnas de la última se denominan *vectores singulares derechos*. Desde la perspectiva de NLP, lo que SVD hace es derivar la estructura semántica latente del documento representado por la matriz A, es decir, un desglose del documento original en r vectores base linealmente independientes que expresan los principales tópicos del documento. SVD puede captar las interrelaciones entre los términos, de modo que los términos y las oraciones puedan ser agrupados sobre una base semántica y no sólo sobre la base de las palabras. El método de generación automática de resúmenes propuesto en [38] usa la representación de un documento antes mencionada, para escoger las oraciones que van en el resumen, basándose en la importancia relativa de los tópicos que abordan, descritos por la matriz V. Para generar un resumen se selecciona la oración más importante por tópico, es decir por vector singular

[3] Gran base de datos léxica para el idioma Inglés que agrupa las palabras en conjuntos de sinónimos llamados "synsets", proporcionando definiciones cortas y generales, y almacenando las relaciones semánticas entre estos conjuntos de sinónimos (http://wordnet.princeton.edu/).

derecho, basado en el valor del índice más alto. El planteamiento de este enfoque permite la selección de oraciones de cualquier tópico, sin discriminar cuáles de ellos tiene mayor importancia. En ese sentido, en [39] se propone un criterio de selección para incluir en el resumen las oraciones cuya representación vectorial en la matriz D^2V, tengan la longitud más grande, en lugar de las oraciones que contiene el mayor valor del índice para cada tópico. Más formalmente, después de computar la SVD de la matriz de término por oraciones, se calcula la longitud del vector de cada oración en D^2V, el cual representa su puntaje para la generación del resumen. Otro método de generación automática de resúmenes que usa LSA fue propuesto por Yeh et *al* [37], en el cual después de realizar SVD sobre la matriz de términos por oración y reducir la dimensionalidad del espacio latente, reconstruyen la correspondiente matriz $A' = U'D'V'$, donde cada columna de A' denota la representación semántica de la oración. Estas representaciones de oración son usadas luego, en lugar del vector de frecuencia basado en palabras clave, para la creación de un grafo de relaciones del texto para representar la estructura de un documento, luego un algoritmo de ordenamiento es aplicado al grafo resultante. Por otro lado, en [19] se combina el sistema propuesto en [39] con un algoritmo de compresión de oraciones que elimina las partes poco importantes de una oración.

Otro enfoque basado en reducción algebraica se basa en las relaciones entre oraciones para denotar la riqueza de información de las oraciones [40].

2.1.3.6 Métodos basados en modelos evolutivos

- ***Programación genética***

La programación genética, siendo una técnica útil para la optimización de tareas, ha sido también utilizada en la generación automática de resúmenes como medio de mejoramiento en la selección de contenido de un documento. En [26], un algoritmo evolutivo basado en programación genética, es empleado como mecanismo de aprendizaje en un sistema adaptable de generación de resúmenes para instruirse en las funciones de clasificación de oraciones. El modelo de programación genética permite la generación de una función de clasificación de oraciones, a partir de una etapa de entrenamiento, que será aplicada posteriormente a los datos en la etapa de prueba.

- ***Métodos basados en Algoritmos Genéticos***

Los algoritmos genéticos han sido empleados en la generación automática de resúmenes como medio de extracción de las oraciones que compondrán un resumen y, además, como herramienta para calcular los pesos que dan significancia a cada oración en un documento fuente.

En [24], un algoritmo genético es utilizado para extraer las oraciones que formarán un resumen. Aquí, un documento es representado por medio de un grafo dirigido acíclico, en el que las oraciones y sus similitudes son consideradas como vértices y arcos, respectivamente. Por su parte, la población del algoritmo genético es vista como un conjunto de resúmenes de longitud fija, en el que cada elemento es representado de forma tal que pueda determinarse cuáles oraciones del documento están incluidas en él. De esta manera, el algoritmo se apoya en el uso de una función de aptitud que intenta acoplar tres factores con el fin de obtener un buen resultado, siendo éstos la legibilidad, la

similitud con el tema del documento y la cohesión entre las oraciones que conforman el resumen. Es así como el cálculo de aptitud para cada resumen candidato, se convierte en la base que determina la elección de mejores soluciones. Por otro lado, en [23], se propone un enfoque independiente del lenguaje para la generación de resúmenes extractivos de un documento[4], el cual, en la fase de entrenamiento, utiliza un algoritmo genético para encontrar los pesos óptimos de la combinación lineal compuesta por 31 métodos estadísticos, representando la solución como un vector. La función de aptitud propuesta mide la calidad del resumen generado por medio de la medida de Recuerdo de ROUGE-1 [41]. Para la creación de nuevas generaciones, el algoritmo genético propuesto selecciona las 100 mejores soluciones para que sean cruzadas genéticamente y mantiene la mejor solución para la siguiente generación. El proceso generacional se lleva a cabo hasta que no se encuentre un mejor conjunto de pesos. El mejor conjunto de pesos obtenido es usado para la puntuación de las oraciones en la fase de prueba. Similarmente, en [6], con el propósito de mejorar la selección de contenido, se aborda la generación automática de resúmenes desde diferentes perspectivas, basadas en algoritmos genéticos, regresión matemática, redes neuronales feed-forward, redes neuronales probabilísticas y modelo de mezclas gaussianas. En todas ellas se hace uso especial de determinadas características de las oraciones que permitan la generación de resúmenes, como la posición en el documento, la centralidad, la semejanza al título, entre otras. De esta forma, en el enfoque basado en algoritmos genéticos, un cromosoma es expresado como una combinación de todos los pesos de esas características. Tras la aplicación del algoritmo, se busca encontrar una combinación apropiada de pesos que sirva como base para la puntuación de cada oración. Es así como todas las oraciones del documento fuente son clasificadas y ordenadas en forma descendente según su puntaje, incluyendo en el resumen aquellas con el valor más alto según la tasa de compresión.

- ***Optimización mediante enjambre de partículas***

La Optimización mediante enjambre de partículas (PSO por sus siglas en inglés Particle Swarm Optimization) es un método estocástico de optimización global, desarrollado en 1995 por James Kennedy y Russell C. Eberhart, que se basa en imitar a nivel computacional el comportamiento social de un conjunto de animales, a partir de la interacción entre sus miembros y el entorno en el que éstos se desenvuelven [42]. La metaheurística PSO consiste en un algoritmo iterativo basado en una población de individuos denominada enjambre, en la que cada individuo o partícula sobrevuela el espacio de decisión en busca de soluciones óptimas, conociendo su posición actual, la velocidad con la que llegó a esa posición y la mejor posición. Una ventaja significativa del PSO frente a otros métodos de optimización es que solamente utiliza tres parámetros que son independientes del problema, el peso inercial, las tasas de aprendizaje C1 y C2, donde C1 es el parámetro cognitivo y C2 es el parámetro social [43].

Binwahlan et *al* [21], proponen un nuevo modelo para el problema de generación automática de resúmenes basado en la inteligencia de enjambre, en el que se aplican los pesos de las características producidos por la PSO. Este modelo se define como la combinación de las calificaciones de las características del texto, de acuerdo a las cuales se realiza el proceso de selección de las frases más importantes que formarán el resumen final, y que son ajustadas según los pesos obtenidos a partir del entrenamiento del PSO.

[4]Extractor de oraciones multilingüe (MUSE por sus siglas en inglés Multilingual Sentence Extractor)

En otra aplicación de este método PSO [22], se hace combinación con lógica difusa, que pretende mejorar el proceso de selección de las frases más importantes que serán incluidas en el resumen final, en el cual los riesgos, la incertidumbre, la ambigüedad y la imprecisión de los valores para el proceso de selección de los pesos de las características podrían ser flexiblemente tolerados. El PSO obtiene los pesos que ajustan las calificaciones de las características, usadas como entradas del sistema de inferencia difuso que determina la puntuación definitiva de la frase para decidir su inclusión en el resumen.

Por otro lado, en [20], se propone un modelo no supervisado de generación de resúmenes para múltiples documentos, que produce un texto comprimido a partir de la extracción de las frases destacadas de los documentos dados. Este modelo es planteado como un problema de programación lineal entera (ILP), estableciendo el algoritmo PSO binario para actuar en espacios con complicaciones binarias. Para la creación del resumen, este modelo optimiza conjuntamente tres propiedades, la relevancia, la redundancia y la longitud. La primera, garantiza que el resumen contenga unidades de texto importantes para el usuario. La segunda, evita que el resumen contenga varias unidades de texto que transmitan la misma información. La última, tiene un valor limitado.

- ***Búsqueda Armónica***

Shareghi y Hassanabadi [25], utilizan el algoritmo de búsqueda armónica (HS por sus siglas en inglés Harmony Search) como un método para extraer los párrafos u oraciones que conformarán el resultado final. Considerando un buen resumen como aquel donde sus oraciones son de fácil lectura, están altamente relacionadas entre sí y discuten sobre el tema central del documento original, los autores plantean un enfoque que, a diferencia de otros métodos extractivos, selecciona las oraciones basándose en tres factores, el factor de legibilidad (RF por sus siglas en inglés Readability Factor), el cual mide el grado de relación entre las frases consecutivas para garantizar un resumen comprensible, el factor de cohesión (CF por sus siglas en inglés Cohesion Factor), que, valiéndose de una matriz de similitud, determina si las frases del resumen discuten sobre la misma información, y el factor de relación con el tema (TRF por sus siglas en inglés Topic Relation Factor), que mide la similitud de las frases con el título del documento. Para evaluar la calidad del resumen obtenido, se propone una función de aptitud que utiliza los tres factores anteriores ponderados con los coeficientes α, β y γ, cuyos valores reales son definidos por el usuario, con el fin de ajustar la función de aptitud a sus necesidades, y varían entre 0 y 1. Comparado con otros algoritmos meta-heurísticos, se concluye que este método puede adaptarse fácilmente a diferentes problemas de optimización, ya que no emplea técnicas matemáticas muy complejas. Así mismo, el algoritmo basado en búsqueda armónica para la generación de resúmenes automáticos presentó mejores resultados de precisión y recuerdo que con un algoritmo basado en genéticos [24]

- ***Evolución diferencial***

La ejecución de la evolución diferencial es similar a la de otros algoritmos evolutivos, sin embargo, difiere principalmente en la representación de las soluciones y operadores. Por ejemplo, se emplean valores reales en lugar de cadenas binarias como es común en los algoritmos genéticos.

Una aplicación de este tipo de algoritmos en la generación automática de resúmenes se hace dentro de un enfoque basado en agrupamiento de oraciones [27]. El agrupamiento de datos es el proceso de identificar la unión natural de datos multidimensionales, basándose en una medida de similitud. En este enfoque la representación de los documentos se basa en el modelo espacio vectorial, y las oraciones son tratadas como secuencias de palabras. Por su parte, la similitud entre dos oraciones es medida a través de la distancia google normalizada. En este método, la evolución diferencial es utilizada como procedimiento de optimización para la asignación de oraciones a grupos, y es así como un individuo es representado por permutaciones que indican los grupos donde quedara ubicada cada oración correspondiente a un gen. Para la evolución hacia nuevas generaciones, se realizan cálculos escalares entre individuos seleccionados aleatoriamente y, finalmente, bajo un criterio que mide la importancia de la oración de acuerdo a su grado de pertenencia al grupo al que está asignada, se seleccionan las oraciones del resumen, mediante un esquema recursivo. La función objetivo a optimizar intenta un balance entre la distancia dentro de los elementos de un grupo y la distancia entre grupos. Una parte de esta función se concentra en minimizar la similitud promedio total entre pares de oraciones dentro de un grupo, mientras que otra parte busca maximizar la distancia entre aquellas oraciones asignadas a diferentes grupos.

- ***Optimización evolutiva difusa***

En [44] se propone un modelo de optimización evolutiva difusa (FEOM) mediante el cual se optimiza la asignación de oraciones a grupos en la tarea de agrupamiento de oraciones. De esta manera, cada individuo de la población se codifica mediante una cadena de números reales. En este esquema se aplican los operadores de selección, cruce y mutación, y además se utilizan tres parámetros de control para regular las probabilidades de cruce y mutación y la distancia relativa de los grupos de oraciones.

2.1.4 Métodos de evaluación de la calidad de los resúmenes

La evaluación es una de las etapas más importantes en el desarrollo de cualquier sistema o método y, en la generación automática de resúmenes, se convierte, además, en una tarea altamente compleja, debido a la dificultad para definir claramente lo que se va a medir y a la ausencia de un resumen ideal que sirva como base de comparación de una forma objetiva. Dicha complejidad se incrementa cuando la labor se hace manualmente, como ocurría en los primeros métodos de generación de resúmenes[4, 7, 10], por lo que automatizarla se convirtió en un objetivo elemental. Para determinar la calidad de un resumen, bien sea manual o automáticamente, se desarrollaron métricas tanto de forma como de contenido [31]. Entre las primeras se incluyeron la coherencia, consistencia y fácil lectura, las cuales se relacionan en gran medida con el sentido lógico de las ideas presentadas en el resumen y permiten llegar con facilidad a la idea principal y a la comprensión del documento original. Dentro de las segundas, que aparecieron posteriormente, se desarrollaron medidas estándar de recuperación de información como la precisión, recuerdo y medida F, que son detalladas en las siguientes secciones.

2.1.4.1 Medidas básicas

Como se mencionó en la Sección 2.1, un resumen debe ser más corto que el texto original y contener la información más importante del mismo. Teniendo en cuenta estos

aspectos, existen dos propiedades que deberían ser medidas a la hora de evaluar un resumen o un sistema de generación de resúmenes. Dichas propiedades son la base de muchas de las métricas de medición existentes en la actualidad. Se trata de la razón de compresión y la razón de retención [45]. La razón de compresión, como se muestra en la Ecuación (2.1), se refiere a qué tan corto es el resumen con respecto al texto original. La razón de retención, por su parte, hace referencia a qué tanta información del texto original es mantenida en el resumen (Ver Ecuación (2.2)).

$$CR = \frac{longitud\ del\ resumen}{longitud\ del\ texto\ completo} \qquad (2.1)$$

$$RR = \frac{información\ en\ el\ resumen}{información\ en\ el\ texto\ completo} \qquad (2.2)$$

La mayoría de investigaciones sobre métodos de generación de resúmenes automáticos ha orientado su interés hacia la tasa de retención más que hacia la de compresión, prefiriendo resultados que mantengan la información importante del texto, mientras que se utiliza una longitud fija para el resumen, independiente del tamaño del texto original.

2.1.4.2 Tipos de evaluación

Existen dos tipos de evaluación para los sistemas de generación automática de resúmenes, y en general para otros sistemas, ellos son la evaluación intrínseca y la evaluación extrínseca.

- ***Evaluación intrínseca***

La evaluación intrínseca se enfoca en la fase de producción del resumen y mide el sistema de generación de resúmenes a partir de la calidad de la salida, sin tener en cuenta el público objetivo. La mayoría de los esquemas de evaluación son intrínsecos y, a menudo, llevan a cabo una comparación con un conjunto de resúmenes ideales[5], que son generados por evaluadores humanos o sistemas de referencia por cada documento de prueba. Sin embargo, debido a la subjetividad que acompaña la creación de un resumen, por cada documento de prueba suele utilizarse un puntaje promedio entre más de un resumen ideal generado por humanos [45].

Al estar orientada hacia la calidad del resumen, la evaluación intrínseca se interesa por lo informativo y coherente del resumen saliente. De esta forma, surgen algunas medidas, utilizadas comúnmente en este tipo de evaluación, como la *precisión* y el *recuerdo* [46]. La primera se define como la razón entre el número de oraciones comunes del resumen generado y de referencia sobre el número total de las oraciones del resumen generado. La segunda hace referencia a la relación entre el número de oraciones comunes del resumen generado y de referencia sobre la cantidad total de oraciones del resumen de referencia. Como combinación lineal de estas dos medidas surge también la medida-F. Existen también otros enfoques de evaluación como los basados en el rango de

[5] Este tipo de conjuntos suele conocerse como gold-standard corpus, y usualmente son vistos como modelos de excelencia que representan el límite más alto al que razonablemente se puede llegar por medios automáticos. Dentro de éste trabajo, este tipo de resúmenes serán mencionados como resúmenes ideales, resúmenes modelo o resúmenes de referencia

oraciones, basados en la utilidad, y basados en la similitud del contenido [45]. En la evaluación basada en el rango de oraciones, se construye el resumen de referencia a partir de la clasificación de las oraciones extraídas del texto original; dicha clasificación se basa en un grado de inclusión, asignado por expertos, que indica la rentabilidad de la oración si fuera incluida en el resumen. Por su parte, en el método de utilidad los resúmenes ideales están formados por unidades de extracción variables (oraciones, párrafos, etc.) con valores de confianza que determinan su inclusión en un resumen [47]. La evaluación por medidas de similitud del contenido es aplicada para evaluar el contenido semántico y suele realizarse una comparación entre los vectores de frecuencia calculados sobre resúmenes lematizados y resúmenes de referencia de algún tipo [48].

Por otra parte, en estudios recientes, se ha utilizado el paquete de medidas de ROUGE[49] como una forma automatizada de evaluación de resúmenes, que se basa en la cantidad de unidades comunes entre un resumen generado y un resumen ideal.

- ***Evaluación extrínseca***

Contrario a la evaluación intrínseca, la evaluación extrínseca valora la salida generada de acuerdo a la ayuda que proporciona al usuario final en el desarrollo de una tarea determinada. De esta manera, la eficiencia del resultado en el apoyo a cierta tarea puede medirse a través de diferentes factores como por ejemplo el tiempo y esfuerzo para interpretar, comprender o editar la salida o el impacto del sistema de generación de resúmenes sobre otro sistema [45].

Para la aplicación de este tipo de evaluación han surgido algunos métodos basados en metodologías como el juego de Shannon, el juego de preguntas, el juego de categorización y la asociación de palabras clave [45]. A partir de la salida generada, estos enfoques buscan calificarla usando ciertas estrategias como la medición del esfuerzo para recrear el texto original, la evaluación del entendimiento de los lectores y su facilidad para transmitir los aspectos importantes del texto original, la evaluación de la asignación de un texto a una categoría y la evaluación de la asociación de un conjunto de palabras clave a un texto base.

2.1.4.3 Evaluación ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE es una técnica reciente que incluye medidas creadas especialmente para evaluar, de forma intrínseca y automática, la calidad de resúmenes de texto creados por computadora [49]. Su enfoque está basado en la aplicación de estadísticas de co-ocurrencias de unidades de texto[6] propuesta en BLEU [50] para la evaluación de traducciones automáticas. Así pues, la evaluación se lleva a cabo a través del conteo de unidades coincidentes entre el resumen generado y resúmenes ideales.
El paquete de evaluación de ROUGE incluye varias medidas como ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S y ROUGE-SU. Entre las más usadas están ROUGE-1, ROUGE-2 y ROUGE-SU4.

[6] Las unidades de texto pueden ser n-gramas, subsecuencias de palabras o pares de palabras, (Un n-grama es una subsecuencia de n elementos de una secuencia dada).

- ***ROUGE-N***

ROUGE-N es una medida basada en el recuerdo de n-gramas entre un resumen generado y un conjunto de resúmenes de referencia. La Ecuación (2.3) muestra el cálculo de esta medida.

$$\text{ROUGE} - \text{N} = \frac{\sum_{S \in \{ResúmenesDeReferencia\}} \sum_{grama_n \in S} Conteo_{Coincidencia}(grama_n)}{\sum_{S \in \{ResúmenesDeReferencia\}} \sum_{grama_n \in S} Conteo(grama_n)} \quad (2.3)$$

Donde n representa la longitud del n-grama $grama_n$ y $Conteo_{Coincidencia}(grama_n)$ es el máximo número de n-gramas coincidentes entre un resumen candidato y un conjunto de resúmenes de referencia. El denominador de esta fórmula corresponde a la suma de la cantidad de n-gramas en el resumen de referencia, de ahí que su valor crecerá conforme al número de resúmenes ideales. De esta manera, un resumen generado que comparta palabras con más de un resumen de referencia obtendrá un mejor valor para la medida ROUGE-N.

- ***ROUGE-L***

Dadas dos secuencias de palabras, la sub-secuencia común más larga LCS entre ellas es la sub-secuencia común con mayor longitud. Esta medida se basa en la premisa de que entre más larga es la LCS entre dos oraciones de los resúmenes comparados, más similares son. Para aplicar LCS en la evaluación de generación automática de resúmenes se propone un enfoque basado en la medida-F para estimar la similitud entre dos resúmenes de diferente longitud. Las Ecuaciones (2.4), (2.5) y (2.6) muestran la estimación de ROUGE-L como el cálculo de la medida F basado en LCS, suponiendo Y como una oración del resumen generado, con longitud n y X como una oración del resumen de referencia, con longitud m.

$$R_{lcs} = \frac{LCS(X,Y)}{m} \quad (2.4)$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \quad (2.5)$$

$$F_{lcs} = \frac{(1 + \beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \quad (2.6)$$

Donde $LCS(X,Y)$ es la longitud de una sub-secuencia común entre X e Y. $\beta = P_{lcs}/R_{lcs}$ cuando $F/R_{lcs} = F/P_{lcs}$; sin embargo, en algunos casos, como en DUC, sólo se considera el recuerdo, ya que β es configurado como un número muy grande. Teniendo en cuenta este cálculo de ROUGE-L, puede inferirse que su valor será 1 cuando $X = Y$ y 0 cuando $LCS(X,Y) = 0$, es decir cuando no haya nada en común entre las oraciones X e Y. La desventaja principal de este método radica en que solamente realiza el conteo de las palabras principales en secuencia y no diferencia relaciones espaciales entre ellas, por lo que LCS's alternativas o secuencias más cortas podrían ser excluidas en el puntaje final. Por ejemplo, dada una secuencia de referencia X y dos secuencias candidatas Y y Z como sigue:

X= [a b c d e f g]
Y= [a b c d h i k]
Z= [a h b k c i d]

A simple vista se puede notar que Y sería una mejor elección que Z, sin embargo, el puntaje ROUGE-L para ambas secuencias es el mismo.

- ***ROUGE-W***

La medida ROUGE-W surge en un esfuerzo por afrontar las debilidades de ROUGE-L, y para ello parte del método básico de esta medida y usa una tabla dinámica de dos dimensiones para manejar la longitud de las coincidencias consecutivas encontradas hasta el momento.

- ***ROUGE-S***

ROUGE-S es una medida basada en estadísticas de co-ocurrencias de bigramas-skip. Un bigrama-skip se refiere a un par de palabras, en el orden en que están en la oración, permitiendo saltos arbitrariamente. Este método mide la superposición de bigramas-skip entre un resumen candidato y un conjunto de resúmenes de referencia.
Dadas una oración de referencia X, de longitud m, y una oración candidata Y, de longitud n, el cálculo de la medida-F basada en bigramas-skip corresponde al cálculo de ROUGE-S como se aprecia en las Ecuaciones (2.7), (2.8) y (2.9).

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \qquad (2.7)$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \qquad (2.8)$$

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}} \qquad (2.9)$$

Donde $SKIP2(X,Y)$ es la cantidad de bigramas-skip que coinciden entre X e Y, β se encarga de controlar la importancia relativa de P_{skip2} y R_{skip2}, y C es la función de combinación que calcula la cantidad de bigramas-skip presentes en una oración[7]. Considerando el siguiente ejemplo:

S_1: police killed the gunman
S_2: the gunman police killed

[7] La fórmula general para calcular las combinaciones que se pueden obtener con n elementos, tomados de r en r, es C(n,r) = (n!/r!*(n-r)!)

Se infiere que cada oración tiene 6 bigramas-skip[8]. Para S_1 los bigramas-skip corresponden a {"police killed", "police the", "police gunman", "killed the", "killed gunman", "the gunman"}. S_2 tiene dos bigramas-skip que coninciden con S_1 y son {"police killed", "the gunman"}. De esta forma, P_{skip2} y R_{skip2} entre S_1 y S_2 son igual a 0,3333, así ROUGE-S se calcula como 0,3333.

- ***ROUGE-SU***

Una desventaja de ROUGE-S es que no da ningún valor a una oración candidata si ésta no tiene ningún par de palabras coincidentes con otro par en las oraciones de referencia. Por ejemplo, dadas las siguientes oraciones:

S_1: police killed the gunman
S_2: gunman the killed police

S_2 corresponde a S_1 en sentido inverso y no tiene ningún bigrama-skip en común con S_1. No obstante, sería pertinente poder diferenciar a S_2 de aquellas oraciones candidatas que no tienen ninguna palabra en común con S_1.

ROUGE-SU surge para contrarrestar este problema, y toma a ROUGE-S como punto de partida e incluye, además, el manejo de unigramas como conteo de unidades. De esta manera, ROUGE-SU adiciona un marcador al inicio de las oraciones candidata y de referencia.

2.1.4.4 Colección de documentos de evaluación

La Conferencia de Entendimiento del Documento (DUC por sus siglas en inglés Document Understanding Conference) es un evento anual en el que organizaciones interesadas en la generación de resúmenes automáticos participan en una serie de pruebas ordenadas de recolección y evaluación de diferentes sistemas, empleando los mismos datos de prueba y cuyo principal propósito es estandarizar la forma de evaluación de los sistemas de generación de resúmenes automáticos [51]. Los resultados de las tareas realizadas en este foro han sido publicados por los organizadores de DUC[9], así como varias colecciones de documentos que han sido tomados como datos de entrenamiento y evaluación en la mayoría de investigaciones orientadas a la generación automática de resúmenes.

2.2 NORMALIZACIÓN E INDEXACIÓN DE DOCUMENTOS

El primer paso antes de proceder a la generación automática de un resumen es realizar la *normalización* del texto del documento. Dicha normalización tiene como propósito evitar la pérdida de información relevante en el emparejamiento de resúmenes, por medio de la aplicación de una serie de operaciones sobre el texto, las cuales permitirán reducir los

[8] C(4,2) = (4!/2!*2!) = 6
[9] http://duc.nist.gov

términos a una forma canónica[10] que permitan la agrupación de términos conceptualmente relacionados [52]. Este proceso, puede incluir técnicas lingüísticas como segmentación de frases o palabras, eliminación de palabras vacías, eliminación de mayúsculas y signos ortográficos y lematización. Además de la normalización del documento, es necesaria también su *indexación*, identificando los términos clave que serán utilizados por el sistema de generación de resúmenes, de tal manera que se facilite la búsqueda y el ordenamiento [53].

En las siguientes secciones, se describen cada uno de los procesos de normalización e indexación ejecutados dentro del sistema propuesto en la presente investigación.

2.2.1 Segmentación

El proceso de segmentación consiste en dividir el texto en unidades significativas como palabras u oraciones. Esta tarea parece simple, sin embargo, su complejidad depende, en gran medida, de la estructura escrita del lenguaje, la cual determina la facilidad para diferenciar los marcadores que delimitan dichas unidades [54]. De esta manera, en la segmentación de oraciones, la identificación de los límites de las oraciones constituye uno de los grandes retos, debido a la ambigüedad que pueden presentar los marcadores de puntuación de un cierto lenguaje, por ejemplo un punto puede no denotar el final de una oración sino una abreviación, un punto decimal o una dirección de correo, según el lenguaje.

2.2.2 Eliminación de mayúsculas y signos ortográficos

La conversión de mayúsculas a minúsculas y la eliminación de signos ortográficos, como tildes o diéresis, normaliza el texto de tal forma que se facilite el emparejamiento de palabras u oraciones.

2.2.3 Eliminación de palabras vacías

Las *palabras vacías* o *Stopwords* son aquellas palabras que, por su bajo contenido semántico, no contribuyen a la discriminación de las oraciones más importantes de un texto, como por ejemplo preposiciones, artículos, pronombres, etc. [55]. Dichas palabras son muy frecuentes dentro de un texto y son consideradas como términos ruidosos o diccionario negativo, por lo que su eliminación puede ser realmente útil antes de la ejecución de una tarea de Procesamiento de Lenguaje Natural [56]. Tal eliminación suele realizarse mediante un filtrado de palabras con la ayuda de una lista de palabras vacías.

2.2.4 Lematización

En el lenguaje humano, un concepto puede ser planteado en diferentes formas que pueden ser denominadas *variantes*; por ejemplo, las palabras “walk”, “walking” y “walker” son variantes que hacen referencia a un concepto similar. Sin embargo, en el procesamiento de lenguaje natural, la discriminación entre dichas variantes puede resultar

[10] La forma canónica de una palabra hace referencia a su forma estándar que, por convención, representa a todas sus flexiones. Esta forma canónica varía según el idioma, por ejemplo, los verbos en inglés se representan mediante la raíz no flexionada, mientras en francés o español se representan con el infinitivo del verbo.

desfavorable, ya que términos referidos a ideas equivalentes pueden ser tratados, en el proceso de emparejamiento, como conceptos distantes. Con el fin de afrontar este hecho, en el área de Recuperación de Información se han utilizado técnicas de lematización o stemming.

La lematizacion es un procedimiento computacional que reduce las palabras con la misma raíz, o lema, a una forma común, eliminando los sufijos variables. Además de menguar el impacto de la discriminación entre conceptos semejantes, la aplicación de lematización, disminuye la cantidad de recursos de almacenamiento, tiempo de ejecución y el tamaño de la estructura de indexación [57]. Evidentemente, el proceso de lematización depende del lenguaje considerado. Entre los algoritmos de lematización más destacados se encuentra el de *Porter* [58] y el de *Lovins* [57]. Tanto el uno como el otro, realizan una eliminación de sufijos y posteriormente recodifican la cadena de texto tratada.

2.2.5 Indexación

En el proceso de indexación de texto las oraciones recolectadas y normalizadas son almacenadas en una estructura de datos para facilitar el acceso rápido y exacto por parte del proceso de recuperación de información, mediante el cual se realiza la búsqueda y emparejamiento de términos u oraciones [59].

2.3 REPRESENTACIÓN DE LOS DOCUMENTOS

Como en cualquier tarea de procesamiento de lenguaje natural, en la generación automática de resúmenes es necesario, antes de generar el resumen, llevar los documentos a una representación que pueda ser fácilmente manipulada por el algoritmo utilizado. Para este fin, dentro de este trabajo se utilizó la representación Modelo de espacio vectorial, la cual ha sido ampliamente utilizado en la generación automática de resúmenes como esquema de representación de los documentos, gracias a que facilita la precisión en el emparejamiento de consultas [53].

2.3.1 Modelo de espacio vectorial

El modelo de espacio vectorial fue propuesto dentro de los sistemas de recuperación de información a inicios de los años 70 [60]. En este modelo, el texto[11] se representa por medio de un vector de pesos de términos. Un término, que puede ser una palabra o una frase, se convierte en una dimensión independiente dentro de un espacio de vectores altamente dimensional. Así mismo, cualquier texto puede representarse por medio de un vector dentro de ese espacio (Ver Figura 2.1). De esta forma, si un término pertenece a un texto, obtiene un valor diferente de cero en la dimensión correspondiente dentro del vector específico [53]. Dicho valor establece la importancia del término dentro del texto según la técnica de ponderación de términos utilizada.

[11]Dependiendo de la aplicación, un texto puede ser una frase, un conjunto de frases o un documento.

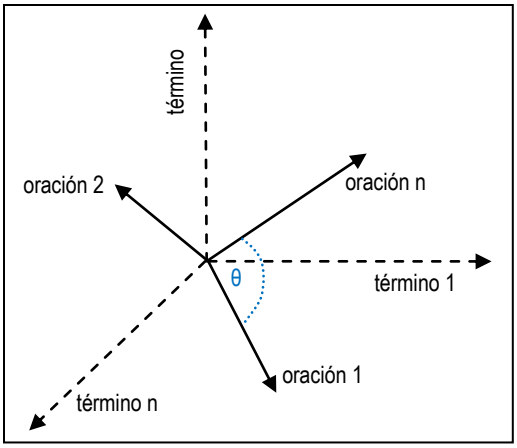

Figura 2.1. Representación Modelo Espacio Vectorial

Claramente, un texto se ubica dentro del espacio vectorial de acuerdo a las coordenadas determinadas por sus términos, por lo que es de esperar que se formen grupos de vectores que representen textos similares de acuerdo a su proximidad [61]. Según este modelo, dado un texto T_i de n términos, su representación vectorial correspondería a $T_i = (w_{i1}, w_{i2}, w_{i3}, \dots, w_{in})$, donde w_{ij} es el peso, ponderación o importancia del término j en el texto i.

2.3.2 Técnicas de ponderación de términos

2.3.2.1 Ponderación booleana

El peso $w_{ij} \in \{0,1\}$ indica la ausencia o presencia del término j dentro del texto i. Se define como muestra la Ecuación (2.10) [62].

$$w_{ij} = \begin{cases} 1, & si\ el\ término\ j\ está\ en\ el\ texto\ i \\ 0, & en\ caso\ contrario \end{cases} \quad (2.10)$$

2.3.2.2 Ponderación basada en la frecuencia del término o TF

TF es una de las primeras técnicas utilizadas para la ponderación de términos de un texto [63]. Como se ve en la Ecuación (2.11), en este esquema la importancia de un término radica en la cantidad de veces que aparezca en el texto.

$$w_{ij} = f_{ij} \quad (2.11)$$

Donde f_{ij} es la frecuencia del término j en el texto i.

La desventaja de la aplicación pura de este método reside en que existen términos muy comunes que pueden aparecer en cualquier parte del texto sin, que por ello, contenga información relevante para caracterizarlo o diferenciarlo. Este tipo de términos tendrían una alta importancia aun cuando sean mucho menos representativos que otros. Así mismo, los términos pertenecientes a textos con mayor longitud tendrían mayor frecuencia

que aquellos presentes en textos más cortos [64]. Para contrarrestar estos aspectos, suele hacerse uso de técnicas de normalización para regular el efecto de las altas frecuencias y de la longitud del texto [65]. De esta forma, el cálculo de frecuencia más comúnmente utilizado, es el que se ve en la Ecuación (2.12).

$$w_{ij} = \frac{f_{ij}}{MáxFreq_i} \quad (2.12)$$

Donde $MáxFreq_i$ indica la cantidad de ocurrencias del término más frecuente dentro del texto i.

2.3.2.3 Ponderación basada en la frecuencia inversa de un término o ISF

La ISF hace referencia a la frecuencia de un término dentro de una colección de textos [66]. De esta forma, la importancia de un término es inversamente proporcional a la cantidad de textos en los que aparezca. La Ecuación (2.13) refleja esta definición.

$$w_{ij} = \log\frac{N}{n_j} \quad (2.13)$$

Donde N es la cantidad de textos de la colección y n_j es la cantidad de textos donde aparece el término j.

2.3.2.4 Ponderación basada en la frecuencia relativa de un término o TF-ISF

Mediante esta técnica de ponderación se obtiene un peso compuesto que combina los métodos descritos en las secciones 2.3.2.2 y 2.3.2.3. De esta manera, al incorporar el cálculo del factor ISF, se busca dar mayor peso a aquellos términos poco frecuentes a nivel de la colección de textos, bajo la concepción de que, por su exclusividad, permitirán caracterizar y diferenciar unos textos de otros [65], mientras considera la relevancia del término a nivel del texto al que pertenece, a través del cálculo del factor TF. El acoplamiento entre estos factores pretende determinar la importancia de un término para un texto dentro de una colección [66]. La Ecuación (2.14) muestra el cálculo de este peso como el producto cruz entre el factor TF y el factor ISF.

$$w_{ij} = TF_{ij} \times ISF_j \quad (2.14)$$

Donde TF_{ij} es la frecuencia del término j en el texto i e ISF_j es la frecuencia inversa del término j, como se presentó en las Ecuaciones (2.12) y (2.13), respectivamente.

Reemplazando se tiene la Ecuación (2.15).

$$w_{ij} = \frac{f_{ij}}{MáxFreq_i} \times \log\frac{N}{n_j} \quad (2.15)$$

2.3.3 Medidas de similitud

En el modelo vectorial, para asignar un puntaje numérico a un documento u oración con respecto a otro(a), se mide la similitud entre los vectores que representan a cada uno(a). Dicha similitud no es inherente al modelo, por lo que puede ser establecida de acuerdo al problema. Sin embargo, en la generación de resúmenes, y en la mayoría de estudios del estado del arte, la medida de similitud más utilizada es la medida basada en cosenos [9, 25, 36, 67], aunque en estudios recientes se ha hecho uso también de una medida basada en la distancia de Google normalizada [20, 27, 68]. Estas dos medidas de similitud son detalladas en esta sección.

2.3.3.1 Medida de cosenos

Normalmente, la similitud en el espacio vectorial se considera como la cercanía que existe entre dos vectores y es medida a partir del ángulo que forman entre ellos[12], indicando que tan cercano está el uno del otro. El coseno de este ángulo es utilizado como similitud numérica, considerando la propiedad del coseno de ser igual a 1 cuando los vectores son idénticos y a 0 cuando son diferentes [53]. Así, entre más pequeño es el ángulo, mayor será la similitud, donde el valor de 1 indica que las oraciones son exactamente iguales. De esta manera, dadas dos oraciones representadas por los vectores $\overrightarrow{s_i}$ y $\overrightarrow{s_j}$, la medida de similitud de cosenos será calculada como se ve en la Ecuación (2.16).

$$sim_{cos}(\overrightarrow{s_i}, \overrightarrow{s_j}) = \frac{\sum_{k=1}^{m} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{m} w_{ik}^2 * \sum_{k=1}^{m} w_{jk}^2}} \quad (2.16)$$

Donde m es el número total de términos del documento, w_{ik} se refiere al peso del término k en la oración s_i y w_{jk} es el peso del término k en la oración s_j.

2.3.3.2 Medida basada en la Distancia de Google Normalizada

La distancia de Google normalizada fue desarrollada por R. Cilibrasi y P. Vitanyi [69], tras un estudio prolongado sobre las medidas de distancia en la recuperación de información [70]. Esta medida fue desarrollada con el fin de obtener un método que indicara la distancia relativa existente entre dos palabras basado en su significado, siendo su valor inversamente proporcional a la relación entre ellas. La principal ventaja que ofrece esta medida es que sin tener conocimiento a nivel semántico de las palabras, logra aprovechar el conocimiento léxico de las mismas. De este modo, se obtiene un valor que indica que a mayor distancia de Google normalizada menor relación existe entre las palabras y viceversa, es decir, un valor de 0 equivale a que las oraciones son idénticas. La Ecuación (2.17) refleja la distancia de Google normalizada entre dos términos t_k y t_l.

$$\mathrm{NGD}(t_k, t_l) = \frac{\max\{\log(f_k), \log(f_l) - \log(f_{kl})\}}{\log n - \min\{\log(f_k), \log(f_l)\}} \quad (2.17)$$

[12] Este ángulo se encuentra en el rango $0<\beta<90$

Donde f_k es el número de oraciones que contiene el término t_k, $f_{k,l}$ es el número de oraciones que contienen los dos términos t_k y t_l simultáneamente, y n es el número de oraciones del documento. Algunas investigaciones han utilizado esta fórmula para calcular la similitud entre dos oraciones [20, 27, 68]. En las Ecuaciones (2.18) y (2.19) se puede ver una de estas adaptaciones.

$$sim_{NGDt}(t_k, t_l) = \exp(-NGD(t_k, t_l)) \qquad (2.18)$$

$$sim_{NGDs}(s_i, s_j) = \frac{\sum_{t_k \epsilon\, s_i} \sum_{t_i \epsilon\, s_j} sim_{NGD}(t_k, t_l)}{|S_i|.|S_j|} \qquad (2.19)$$

Donde $sim_{NGDt}(t_k, t_l)$ es la similitud entre los términos t_k y t_l, $sim_{NGDs}(s_i, s_j)$ es la similitud entre las oraciones s_i y s_j. $NGD(t_k, t_l)$ se calcula como en la Ecuación (2.17), $|s_i|$ es el número de términos distintos en s_i y $|s_j|$ es el número de términos distintos en s_j.

2.4 ALGORITMOS MEMÉTICOS

2.4.1 El concepto de meme

El término meme, introducido en 1976 por Richard Dawkins [71], denota en la difusión cultural, una idea similar al concepto de gen en la evolución biológica, y hace referencia a una unidad de información cultural que puede transmitirse de un individuo a otro o, idealmente, de una generación a otra. En ese sentido, un meme puede ser una idea, concepto, costumbre, habilidad, técnica, etc., que se propaga a través de la enseñanza, aprendizaje, imitación o asimilación, de acuerdo a la forma en que se procesa en la mente de los individuos transmisores [72].

La diferencia básica entre la evolución genética y la evolución cultural se fundamenta en la fidelidad de la reproducción [73]. De esta forma, mientras un gen se replica independiente de las acciones del individuo, un meme, antes de su transmisión, puede ser adaptado por el individuo, conforme a la asimilación que haya logrado de él. Este concepto es aplicado dentro del campo informático a través de los algoritmos meméticos, en los cuales, como se describe en las siguientes secciones, un meme puede ser mejorado o adaptado a través de estrategias de optimización local que permitan el mejoramiento de un individuo [74]. Tales estrategias pueden ser algoritmos de aproximación, técnicas de búsqueda local, etc. [29].

2.4.2 Definición

El término algoritmo memético (MA por sus siglas en inglés Memetic Algorithm) surgió a finales de los 80 para describir un esquema que intenta imitar la evolución cultural de las poblaciones [73]. Este tipo de algoritmos combinan una búsqueda global basada en población con una búsqueda local heurística hecha por cada individuo, es decir, combinan la evolución genética con el aprendizaje que los individuos logren durante su tiempo de existencia [73]. El principal objetivo de los algoritmos meméticos, al incorporar optimizaciones individuales y procesos de cooperación y competencia poblacional, es

direccionar la exploración hacia las regiones más prometedoras del espacio de búsqueda [72]. Un proceso de competencia, involucra técnicas de selección de individuos, mientras que un proceso de cooperación se refiere a la generación de nuevos individuos a través del intercambio de información [73]. Además de las mejoras individuales, otro aspecto importante en los algoritmos meméticos es la inclusión del conocimiento del dominio del problema, bajo la concepción de que un algoritmo de búsqueda se desempeña según la calidad y cantidad de información que incorpore acerca del problema abordado [29].

2.4.3 Estructura

Un algoritmo memético se ejecuta a lo largo de poblaciones de individuos que, dentro de este contexto, son conocidos como *agentes*[13]. Un agente es una representación de una solución, o en algunos casos de varias [75], y se caracteriza por su *comportamiento activo* en la resolución del problema que aborda [76]. Los agentes de una población compiten y cooperan mutuamente durante la evolución, siendo esto, una característica sobresaliente dentro de los MA. La Figura 2.2 muestra la estructura general de un MA. Como cualquier esquema evolutivo, este algoritmo parte de una población inicial de agentes, que pueden ser generados de forma aleatoria o por medio de heurísticas existentes [77] [78]. Una de las técnicas más usuales para este fin, es la aplicación de optimizadores locales tras la generación aleatoria del nuevo agente.

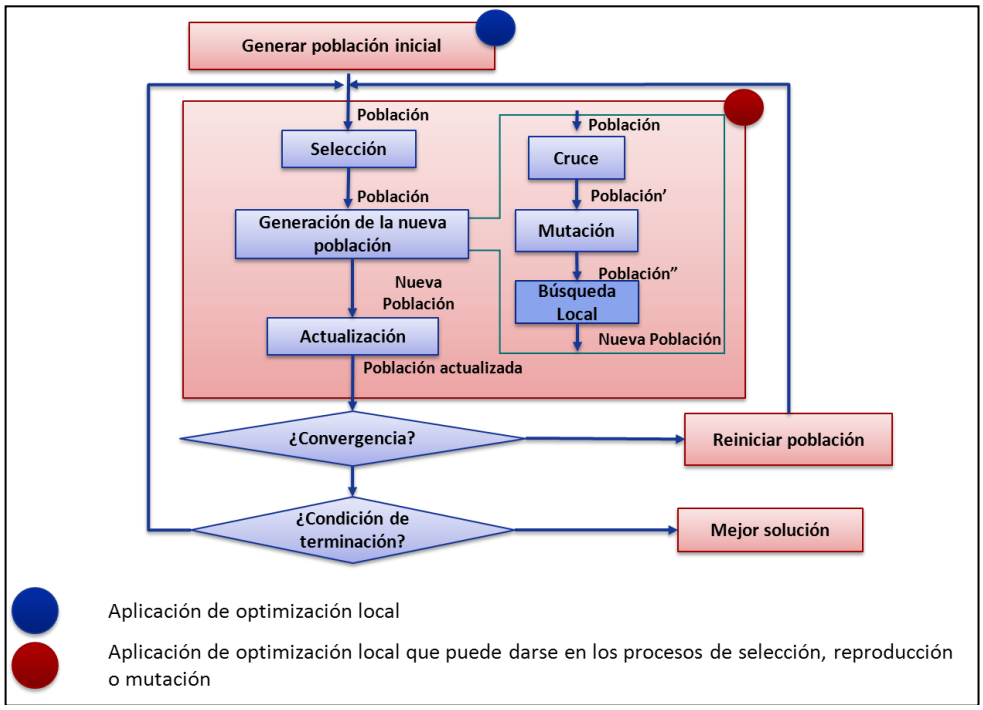

Figura 2.2. Estructura general de un MA

[13] En el presente trabajo los términos agente e individuo son utilizados indistintamente, sin embargo, es importante tener en cuenta que la primera denominación es la más adecuada dentro del contexto de los MA, debido a su cualidad de dinamismo dentro del proceso evolutivo.

Por otro lado, las interacciones, ya mencionadas, de competencia y cooperación, son en gran medida lo que define el paso generacional de una población a otra, y se reflejan en los procesos de selección, reemplazo y reproducción entre agentes. Antes de la reproducción, se seleccionan los agentes que actuarán como padres de la nueva generación[14], con base en los valores de aptitud de cada agente obtenidos a partir de una *función objetivo* diseñada para evaluar cuantitativamente la calidad de cada agente según su capacidad para resolver el problema abordado. Dentro de la competencia entre agentes, se envuelve también la actualización de la población a través del *reemplazo*, sustituyendo algunos agentes por otros nuevos basándose, usualmente, en el valor de aptitud. Por su parte, la naturaleza cooperativa del algoritmo está representada por la fase de reproducción, en la cual, a través de un operador de cruce, se intercambia la información de los padres para dar origen a nuevos agentes. Dentro de la reproducción se lleva a cabo también la inclusión de información ajena a los agentes implicados mediante un operador de *mutación*, el cual toma un agente existente y, después de modificarlo parcialmente, genera uno nuevo. En el MA la competencia y la cooperación son repetidas hasta satisfacer un *criterio de parada*, el cual es predeterminado a un número de iteraciones, un número de iteraciones sin mejora, alcanzar una mejora deseada, u otro criterio según el propósito.

Otra propiedad importante dentro de la estructura de un MA es la reiniciación de la población. Este componente es incluido con el fin de evitar el mal uso de recursos computacionales al explorar una población que ya se ha degenerado en un estado conocido como *convergencia*. De este modo, cuando la evolución ha caído por debajo de un valor límite y no hay mejora alguna, se dice que la población ha alcanzado la convergencia y es necesaria su reiniciación. En consecuencia, se conserva un porcentaje de la población actual y se crean nuevos agentes hasta completar el tamaño determinado.

2.4.4 Optimización local

Como rasgo más representativo de los MA, se encuentra la aplicación de estrategias de optimización local sobre los agentes de la población. La integración de estas estrategias dentro del algoritmo, tiene como propósito acelerar y garantizar el hallazgo de aquellas soluciones que difícilmente serían obtenidas sólo con la aplicación de evolución genética [79]. Existen diversos métodos para la optimización local, sin embargo, lo más común es el empleo de técnicas de búsqueda local. Un optimizador local es considerado un *meta-operador* que aplica iterativamente un operador de mutación arbitrario sobre un agente, mientras conserva los cambios que lo llevan a una mejora en su aptitud [72]. Dicha mejora, es evaluada de acuerdo a la función objetivo del problema abordado. Al ser considerado un operador, un optimizador local puede ser aplicado en diferentes lugares del MA, sobre una parte o sobre toda la población de agentes (Ver Figura 2.2). Lo más usual es la aplicación de optimización local sobre los agentes de la población en cada generación, basada en algún parámetro de probabilidad que determine si se debe ejecutar o no la optimización.

[14] Usualmente dos padres son seleccionados, sin embargo, según el problema, pueden tomarse grupos de mayor tamaño.

2.5 OPERADORES REPRODUCTIVOS Y DE OPTIMIZACIÓN LOCAL

Como se comentó en secciones previas, el algoritmo memético hace uso de operadores de selección, cruce, mutación, reemplazo y, en forma particular, de optimización local. A continuación, se presenta brevemente cada uno de ellos (Para ver una descripción más detallada de los operadores de un algoritmo memético remitirse al *Anexo A,* en las secciones 1 a la 6).

2.5.1 Operador de selección

El operador de selección es el encargado de elegir los agentes que participarán en el proceso de reproducción para crear nuevos agentes que conformen una nueva población [80]. A continuación se describe en forma breve las técnicas de selección más destacadas en los algoritmos evolutivos.

2.5.1.1 Selección por rueda de ruleta

Este método consiste en simular el comportamiento de una rueda de ruleta, en la cual cada agente ocupa un espacio proporcional a su valor de aptitud. De esta forma, para seleccionar un agente, la ruleta debe girar una cantidad aleatoria de veces. Tras la detención de la ruleta se conocerá cuál es el agente elegido por el selector de ruleta [81].

2.5.1.2 Selección por torneo

La selección por torneo consiste en la elección aleatoria de *n* candidatos de la población, los cuales deben competir entre sí para ser uno de los padres de la nueva generación [82]. El agente ganador será el que tenga mayor valor de aptitud. Este proceso es repetido hasta seleccionar la cantidad de padres correspondiente. La cantidad de competidores es conocida como *tamaño del torneo*. Este algoritmo suele denominarse también *Torneo Determinístico*. Adicionalmente, existe una variación de este método, conocida como *Torneo Probabilístico*, en la cual en lugar de escoger al agente más apto, se genera un número aleatorio entre [0,1], si dicho número es mayor que un parámetro de probabilidad de selección[15], entonces se decide por el agente más apto, o en caso contrario, por el menos apto.

2.5.1.3 Selección basada en el rango

Esta técnica consiste en dar a cada agente una probabilidad de selección basada en un rango relativo a la población entera. Así pues, este algoritmo ordena los agentes de la población de acuerdo a su valor de aptitud, de este modo, al mejor agente le corresponderá el rango *N*, siendo *N* el tamaño de la población, mientras que al peor le corresponderá el rango 1. El rango de cada agente es traducido a su probabilidad de selección. De esa forma, para el proceso de elección, en lugar de usar el valor de aptitud de un agente, se utiliza su rango escalado [83].

[15] La probabilidad de selección es establecida desde el inicio del proceso evolutivo.

2.5.1.4 Selección elitista

El mecanismo elitista consiste en seleccionar los *n* agentes más aptos para que sean copiados directamente a la siguiente generación [84].

2.5.1.5 Selección por emparejamiento restringido

Este mecanismo es una adaptación de la selección por Torneo determinístico [85, 86]. La diferencia está en que tras seleccionar el conjunto aleatorio de agentes, de ellos se escoge aquel cuyo valor de aptitud sea capaz de competir con un agente de referencia, es decir se escoge el más similar.

2.5.2 Operador de cruce

La operación de cruce se lleva a cabo al intercambiar parte de las cadenas de dos, o más, agentes para formar nuevos agentes [87]. A continuación se describen los métodos de cruce más sobresalientes dentro de los algoritmos evolutivos.

2.5.2.1 Cruce de un punto

Este método divide los agentes seleccionados en un punto específico escogido aleatoriamente, y genera dos segmentos diferenciados en cada uno de ellos. Seguidamente, se intercambian las partes localizadas después del corte entre los dos agentes, para de esta forma generar los nuevos descendientes [85, 87, 88].

2.5.2.2 Cruce de n puntos

Este método consiste en cortar los dos agentes en *n* puntos seleccionados al azar. Luego, el material genético situado entre los *n* puntos es intercambiado en forma intercalada [89].

2.5.2.3 Cruce uniforme

En este método se genera una máscara de cruce de valores binarios. De esta forma, si el valor de una de las posiciones de la máscara es 1, el gen situado en esa posición se hereda del primer padre. Por el contrario, si el valor es 0, el gen de esa posición es heredado del segundo padre. Para producir el segundo descendiente se intercambian los papeles de los padres o la interpretación de los unos y ceros en la máscara de cruce [90].

2.5.2.4 Cruce plano

Esta técnica consiste en determinar los bits comunes entre los padres seleccionados, los cuales son heredados a los descendientes. Los bits faltantes son llenados con valores aleatorios, repitiendo el proceso hasta que se obtenga la cantidad de hijos deseada [91].

2.5.2.5 Cruce de anillo

Este cruce propone la combinación en forma de anillo de agentes padres. Seguidamente se escoge un punto de corte aleatorio en el anillo formado. Con respecto a dicho punto de corte, el primer hijo es creado en dirección de las manecillas del reloj hasta la longitud de

los padres, mientras que el segundo hijo es creado en el sentido contrario a las manecillas del reloj [92].

2.5.3 Operador de mutación

Este operador altera, de forma aleatoria, uno o más bits de la estructura de un agente [93]. A continuación, se describen los métodos de mutación más reconocidos dentro de la computación evolutiva.

2.5.3.1 Mutación de intercambio

Este método selecciona aleatoriamente dos posiciones del agente e intercambia sus valores. Puede ser ajustado para que se lleve a cabo más de un intercambio. [94].

2.5.3.2 Mutación de inserción

Este método propone la selección aleatoria de uno de los bits de un agente para que sea eliminado e insertado en otra posición seleccionada arbitrariamente [95, 96].

2.5.3.3 Mutación de bit

Esta técnica consiste en la elección aleatoria de uno de los bits de un agente para que su valor sea modificado, de tal manera, que dentro de la codificación binaria, si el valor del bit es 1, su valor es cambiado por 0 y viceversa [84, 97].

2.5.3.4 Mutación Multi-bit

En esta técnica se analiza cada bit del agente para decidir cuáles bits deben ser modificados y cuáles no, de acuerdo a una segunda probabilidad de mutación. En caso de que un bit deba ser mutado se modifica su valor [84].

2.5.4 Operador de reemplazo

El operador de reemplazo se encarga de decidir la eliminación de algunos agentes para la inclusión de otros generados en la etapa de reproducción [80]. Las estrategias de reemplazo que se presentan a continuación son quizás las más aplicadas dentro de los algoritmos evolutivos.

2.5.4.1 Reemplazo Aleatorio

En esta técnica los agentes que serán reemplazados por los hijos generados son escogidos aleatoriamente, sin importar su aptitud [98].

2.5.4.2 Reemplazo del Peor

En este método se reemplazan los agentes menos aptos de la población por los nuevos agentes generados [99].

2.5.4.3 Reemplazo del Peor Padre

Bajo esta estrategia, se elimina uno de los padres para dar espacio al nuevo hijo generado. La decisión de cuál de los padres eliminar se apoya en un criterio predefinido que puede estar basado en los valores de aptitud, diferencia genética, etc. [89].

2.5.4.4 Reemplazo de Similares

En este esquema el agente con aptitud más cercano al hijo generado es reemplazado [93].

2.5.4.5 Reemplazo por Competencia Restringida

Dentro de este método, se selecciona un conjunto de *n* agentes aleatoriamente y de ellos se escoge el más similar en aptitud al hijo generado para que sea reemplazado por el mismo [97].

2.5.5 Operador de Búsqueda Local

Las técnicas de búsqueda local están destinadas a trabajar en espacios de búsqueda muy grandes y su propósito es encontrar una solución óptima con la mayor exactitud y menor costo computacional posibles [100]. La idea general de un mecanismo de búsqueda local es partir de una solución y, por medio de un proceso iterativo y una estrategia definida, reemplazarla por otra mejor en su vecindario, la cual se convertirá, entonces, en la solución actual. A continuación, se describen los métodos de búsqueda local más sobresalientes dentro de la computación evolutiva.

2.5.5.1 Búsqueda Local Básica

La búsqueda local básica consiste en ejecutar un movimiento sólo si la solución resultante es mejor que la actual, repitiendo este proceso hasta encontrar un óptimo local [101].

2.5.5.2 Búsqueda Local por Entornos Variables

La búsqueda local por entornos variables (VNS por sus siglas en inglés Variable Neighborhood Search) cambia sistemáticamente la estructura de entornos por la que se realiza la búsqueda. En ese sentido, bajo esta técnica, se exploran vecindarios distantes de la solución actual y se cambia de esta solución a otra sólo si hay una mejora [102]. Existen algunas extensiones de este método que tratan de mantener la simplicidad del esquema básico, entre ellas están la *VNS Descendente*, *VNS Reducida* y *VNS Básica*.

2.5.5.3 Búsqueda Local Guiada

La búsqueda local guiada (GLS por sus siglas en inglés Guided Local Search) es un mecanismo cuyo propósito es guiar la búsqueda hacia zonas prometedoras capturando y explotando la información relacionada con el problema y la búsqueda. De esta manera, la GLS incorpora la utilización de penalizaciones que regularizan las soluciones generadas por la búsqueda local para que estén acordes con la información reunida antes o durante la búsqueda [103].

2.5.5.4 Búsqueda Tabú

El objetivo de la Búsqueda Tabú (TS por sus siglas en inglés Tabu Search) es guiar la búsqueda local hacia la optimización global, a través de la incorporación de estrategias de aprendizaje [104]. En ese sentido, esta técnica introduce la utilización de una memoria adaptativa que permite mantener la historia del proceso de búsqueda, almacenando todos los movimientos visitados o aquellos que cumplan con alguna condición de acuerdo al problema abordado, los cuales son llamados movimientos tabú. El estado tabú de un movimiento puede ser cambiado con respecto al tiempo o a las condiciones del momento [105].

2.5.5.5 Búsqueda Local Iterativa

La búsqueda local iterativa (ILS por sus siglas en inglés Iterated Local Search) es una técnica de búsqueda local cuyo propósito es explorar el espacio de soluciones mediante un recorrido que lleve de una solución a otra sin la restricción de usar sólo los vecinos más cercanos. En ese sentido la ILS incorpora perturbaciones que permitan alcanzar nuevas soluciones [106].

Capítulo 3

3 PROCESO DE CONSTRUCCIÓN DEL ALGORITMO MEMÉTICO PROPUESTO

El desarrollo de esta investigación se llevó a cabo con base en la *Metodología Iterativa* diseñada especialmente para la ejecución de proyectos en ciencias de la computación y que involucran una solución computacional [107]. De esta forma, cada ciclo del proceso se conforma por cuatro etapas que son observación, identificación del problema, desarrollo de la solución y prueba de la solución. En la Figura 3.1 se presentan los ciclos definidos para el desarrollo de la presente investigación y en la Tabla 1 las actividades abarcadas por cada etapa de la metodología aplicada.

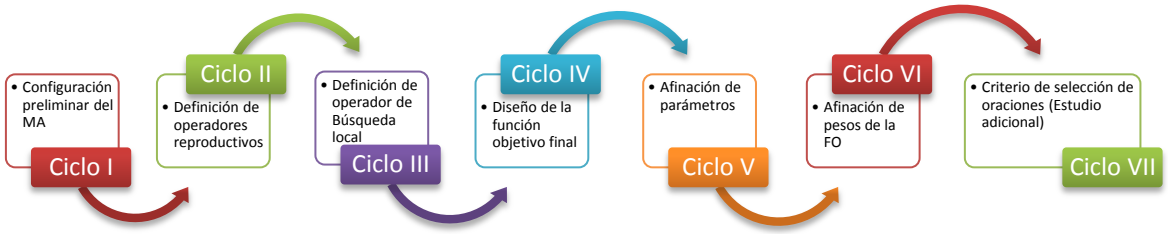

Figura 3.1. Ciclos de la Metodología Iterativa utilizada

Ciclo	ETAPA I: Observación	ETAPA II: Identificación del problema	ETAPA III: Desarrollo	ETAPA IV: Prueba
Ciclo I	▪ Estudio de la representación de las soluciones. ▪ Estudio de las características del texto que podrían hacer parte de la función objetivo. ▪ Estudio de esquemas reproductivos y de búsqueda local.	Establecimiento de las configuraciones preliminares del MA y de la función objetivo.	Establecimiento de la configuración inicial de los pesos de la FO y de los parámetros del MA.	▪ Preparación del conjunto de documentos para realizar la evaluación. ▪ Selección de las medidas de ROUGE a usar para la evaluación. ▪ Selección de los algoritmos con los cuales se comparará el algoritmo memético.
Ciclo II	Profundización en los esquemas reproductivos.	Establecimiento de diferentes configuraciones de operadores reproductivos.	Implementación de las diferentes configuraciones de operadores reproductivos.	Evaluación de las diferentes configuraciones de operadores reproductivos.
Ciclo III	§ Profundización en los esquemas de búsqueda local.	Establecimiento de diferentes configuraciones de operadores de búsqueda local.	Implementación de las diferentes configuraciones de operadores de búsqueda local.	Evaluación de las diferentes configuraciones de operadores de búsqueda local.
Ciclo IV	Profundización en las características del texto (estadísticas y de similitud).	Establecimiento de diferentes configuraciones de la FO con diferentes características.	Implementación de las diferentes configuraciones de la FO.	Evaluación de las diferentes configuraciones de la FO.

Ciclo V	-	-	Afinación de los parámetros del MA	Evaluación del MA con diferentes configuraciones de parámetros
Ciclo VI	-	-	Afinación de los pesos de la FO del MA	Evaluación del MA con diferentes configuraciones de pesos de la FO
Ciclo VII	Estudio de diferentes criterios de selección de oraciones.	Establecimiento de diferentes configuraciones de criterios de selección de oraciones.	Implementación de diferentes criterios de selección de oraciones	Evaluación de los diferentes criterios de selección de oraciones.

Tabla 1. Etapas de la Metodología Iterativa utilizada

A continuación se describe el proceso de cada uno de los ciclos desarrollados para llegar al diseño final del algoritmo propuesto.

3.1 CICLO I: DISEÑO PRELIMINAR DEL ALGORITMO MEMÉTICO

3.1.1 Representación de las soluciones

Teniendo en cuenta que los algoritmos meméticos no trabajan directamente sobre las soluciones del problema considerado, sino que lo hacen sobre una abstracción de los objetos solución, es importante definir la estructura por medio de la cual los agentes serán representados dentro del espacio de búsqueda. Una buena abstracción debe ser capaz de identificar las características que constituyen el espacio de soluciones, de tal manera que, representando diferentes perspectivas, se generen diferentes agentes o soluciones. Es así como se hace necesario definir una función de codificación sobre el espacio de búsqueda, que permita mapear cada punto de dicho espacio en un genotipo, y una función de decodificación que obtenga el fenotipo asociado a un agente (Ver Figura 3.2).

$$\text{Codificación} : \text{Espacio de soluciones} \rightarrow \text{Agente}$$

$$\text{Decodificación} : \text{Agente} \rightarrow \text{Espacio de soluciones}$$

Figura 3.2. Especificación de las funciones de codificación y decodificación

El genotipo, en terminología biológica, denota toda la información almacenada en los agentes, permitiendo describirlo a nivel de genes [108]. Por su parte, el fenotipo, como resultado de la decodificación del genotipo, describe la apariencia externa de un agente [89]. Para representar el gran número de posibles fenotipos, la información genotípica se almacena en secuencias de alelos en cromosomas (Ver Figura 3.3). Aunque la naturaleza a menudo utiliza más de un cromosoma, la mayoría de aplicaciones de algoritmos evolutivos utiliza uno solo para representar la información genotípica de un individuo, de tal forma, que los términos individuo, agente y cromosoma suelen utilizarse indistintamente. De ese modo, cada solución es codificada como una serie de genes que representan las diferentes restricciones o variables del problema.

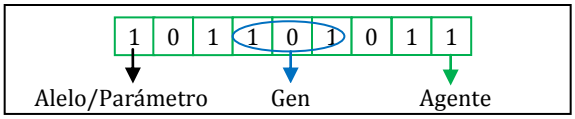

Figura 3.3. Estructura de un agente o solución

Teniendo en cuenta que la apariencia fenotípica de un individuo determina su éxito en la vida, las capacidades de los diferentes individuos deben ser juzgadas a nivel de fenotipo, sin embargo, en la reproducción los mismos individuos deben ser vistos a nivel de genotipo, ya que lo que se hereda a los descendientes tras el apareamiento es la información genotípica. Por lo tanto, bajo el contexto de los algoritmos evolutivos, los operadores reproductivos funcionan a nivel de genotipo, mientras la evaluación de los individuos es desarrollada a nivel de fenotipo [80]. Así pues, un agente es decodificado durante el proceso de evaluación de su función objetivo, con el propósito de obtener una serie de parámetros a partir de los cuáles se determina la calidad de una posible solución.

Considerando los aspectos anteriores, en el esquema memético propuesto, la codificación de una solución es realizada mediante un vector binario. La codificación binaria se caracteriza por su simplicidad y flexibilidad para codificar y trabajar con problemas de múltiples variables, la fácil manipulación por parte de los operadores genéticos y la facilidad para ser traducida en la solución real del problema, es decir, en el resumen de texto. De esta manera, si un documento está formado por n oraciones $\{s_0, s_1, s_2, \dots, s_{n-1}\}$, la representación de una solución sería como se ve en la Figura 3.4. Por medio del índice de cada elemento del vector solución, se puede identificar la posición de la oración en el documento original. El valor de cero para un gen dentro del vector indica que la oración correspondiente no pertenece al resumen representado, mientras que un valor de uno indica, por el contrario, que dicha oración sí está dentro de ese resumen. En el ejemplo que se muestra en la Figura 3.4 el resumen representado estaría conformado por las oraciones $s_1, s_3, \dots, s_{n-2}$.

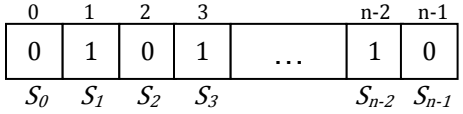

Figura 3.4. Representación binaria de un agente o solución

Por otro lado, la decodificación de una solución es necesaria para poder medir su calidad a través del cálculo de su valor de aptitud, y en la generación de resúmenes automáticos, consiste en buscar ordenadamente las oraciones cuya posición en el documento coincide con la posición de los genes del vector solución cuyo valor es uno, obteniendo así el resumen representado. De este modo, el cálculo de aptitud de un agente se realiza sobre las oraciones obtenidas tras este proceso de decodificación.

3.1.2 Configuración preliminar del algoritmo memético

Antes de realizar la primera configuración del Algoritmo Memético se llevó a cabo un estudio preliminar de operadores y, teniendo en cuenta los rasgos que comparte el proceso memético con la evolución genética, se revisó un conjunto de operadores

reproductivos utilizados en algoritmos genéticos [84, 87, 88, 93, 109]. Así mismo, se estudiaron varias técnicas de búsqueda local. Tras esta revisión y conforme a las características, ventajas y desventajas de cada método, presentadas en las Tablas 1 a la 5 del *Anexo A*, se identificaron los métodos que, teóricamente, podrían adaptarse al problema abordado y ayudar a conseguir un balance entre los procesos de exploración y explotación, conduciendo a buenos resultados. Tales métodos son presentados en la Tabla 2, junto a otros, destacados por la simplicidad en su implementación (Para más detalle de este estudio, referirse al *Anexo A*). Gran parte de los ciclos experimentales posteriores está orientada a la discriminación del comportamiento, dentro del problema abordado, entre las técnicas más simples y aquellas con estrategias más elaboradas.

Operador	Método más prometedor	Método más simple
Selección	Basada en el Rango	Aleatoria
Cruce	Cruce Uniforme	Cruce de un punto
Mutación	Mutación Multi-Bit	Mutación de bit
Reemplazo	Competencia Restringida	Competencia Restringida
Optimización Local	Búsqueda Local Guiada	Búsqueda Local Básica

Tabla 2. Métodos destacados según el estudio realizado

A pesar de haber identificado *teóricamente* los métodos con mejores cualidades, para llegar a la configuración del algoritmo memético que condujera a los mejores resultados para el problema abordado, no bastaba con seleccionar los métodos cuyas características sugirieran un buen desempeño, también era necesario probar, en la *práctica*, el comportamiento de cada método al acoplarse con los demás operadores, pues, como en cualquier sistema, el mejor desempeño del algoritmo no siempre se obtendría con las mejores técnicas de los operadores, sino con aquellas que logren el mejor comportamiento sinérgico. De este modo, las pruebas se iniciaron con el esquema *preliminar* del algoritmo memético presentado en la Tabla 3, cuyos esquemas fueron seleccionados con base en la simplicidad de adaptación. El establecimiento de los métodos de selección para padre y madre es uno de los primeros pasos en el ciclo de definición de operadores reproductivos, por tal motivo son los únicos que no están definidos dentro de la configuración preliminar del algoritmo memético.

Operador	Método
Selección padre	No establecido
Selección madre	No establecido
Cruce	Cruce de un Punto
Mutación Externa	Mutación de Bit
Mutación Interna	Mutación de Inserción
Reemplazo	Competencia Restringida
Optimización Local	Búsqueda Local Básica del Mejor

Tabla 3. Configuración preliminar del Algoritmo Memético propuesto

3.1.3 Diseño de la función objetivo preliminar

La definición de la función objetivo es uno de los pasos más importantes dentro del diseño de los algoritmos meméticos, ya que es ella quien guía el mecanismo de exploración. La

función objetivo se encarga de evaluar y asignar un puntaje (comúnmente denominado *valor de aptitud*) a los agentes o miembros de la población, con base en su capacidad para resolver el problema abordado. Los valores de aptitud determinan, en gran medida, cuáles agentes de la población tendrán mayor y menor posibilidad de reproducirse y dar lugar a las siguientes generaciones. Idealmente, la función objetivo debe reflejar lo mejor posible la aptitud de cada agente para resolver el problema, de tal manera que aquellos agentes cercanos en el espacio de búsqueda tengan también valores de aptitud cercanos o similares. En la siguiente sección se explican las principales características que pueden ser consideradas para conformar la función objetivo dentro del problema abordado.

3.1.3.1 Características para conformar la función objetivo

Dentro de la generación de resúmenes, existen ciertas características que pueden considerarse para decidir la inclusión de las oraciones que conformarán la salida o como medio para determinar su calidad. De esta forma, con base en el estudio del estado del arte, se reunió un conjunto de características, independientes del dominio y del lenguaje, que han sido utilizados para tales fines. De este modo, dada la gran influencia de la función objetivo en el desempeño y eficiencia del sistema presentado en este trabajo, las características recolectadas, y descritas a continuación, fueron la base para definir su configuración.

- ***Posición de la oración en el documento***

Si todas las oraciones de un documento tuvieran la misma importancia, al reducir el tamaño del documento para generar un resumen se perdería información significativa. Sin embargo, según los estudios realizados al respecto, la información relevante en un documento, sin importar su dominio, tiende a encontrarse en ciertas secciones como títulos, encabezados, oraciones iniciales de los párrafos, párrafos iniciales, etc.[9, 110]. Aun así, cabe destacar que la posición de las oraciones depende en gran medida del tipo de documento a considerar, por ejemplo, las noticias guardan una estructura de tipo piramidal invertida, en la cual la información sobresaliente se encuentra, por lo general, en las primeras oraciones, mientras que el resto del documento desarrolla sólo información complementaria[16] [111, 112]. De esta manera, para evaluar una oración con base en su posición, se define un criterio de selección que utiliza la distancia existente entre la oración y el inicio del documento, asignando un mayor valor a las oraciones iniciales.

En la recuperación de información se han aplicado varias técnicas basadas en la posición de las oraciones y, en muchos casos combinadas con otros criterios de selección, han probado su efectividad para determinar la relevancia de una oración [1, 3, 6, 36, 113-117]. Dentro de estos esquemas, existen dos que aplican un cálculo normalizado del factor basado en la posición y son el propuesto por Bossard [118], definido como se ve en la Ecuación (3.1), y el presentado por Wan [36], que se calcula como se ve en la Ecuación (3.2). Estos dos cálculos son adoptados en la presente investigación para evaluar la característica fundamentada en la posición.

[16] Este tipo de estructura en las noticias tiene como objetivo que el lector se informe de la noticia con rapidez y que le permita cortar la noticia sin complicaciones.

$$P(s_j) = \sqrt[2]{\frac{1}{n_j}} \qquad (3.1)$$

$$P(s_j) = 0.5 + \frac{1}{n_j + 1} \qquad (3.2)$$

Donde n_j indica la posición de la oración s_j en el documento.

- ***Longitud de la oración***

A través de la prueba, algunos estudios han concluido que las oraciones más cortas de un documento deberían tener menos probabilidad de aparecer en el resumen del mismo [7]. Las formalizaciones más comunes del criterio de selección de oraciones basado en la longitud, consisten en aceptar aquellas oraciones cuyo tamaño supera cierto umbral [7, 119], o emplear un cálculo relativo basado en el promedio de longitudes o en la longitud de la oración más larga del documento [6, 113]. Una propuesta reciente para el cálculo de este factor, realiza una normalización apoyada en la función sigmoidea [120, 121]. Lo interesante de esta estimación es que, además, toma en cuenta la distribución estándar de los datos con el fin de llegar a una evaluación más balanceada, pues si bien aún se seguirá privilegiando las oraciones más extensas, no descartará completamente aquellas de longitud media, presumiendo que puedan también tener información relevante para el resumen. De esta forma, teniendo en cuenta que la distribución estándar representa la tendencia de los datos a variar por encima o por debajo del valor medio, se espera que una oración con una longitud no muy corta obtenga una buena calificación en este factor, de acuerdo a las fórmulas expresadas en las Ecuaciones (3.3) y (3.4).

$$L = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}} \qquad (3.3)$$

$$\alpha = \frac{l(s) - \mu\big(l(s)\big)}{std\big(l(s)\big)} \qquad (3.4)$$

Donde $l(s)$ es la longitud de la oración, $\mu\big(l(s)\big)$ es la longitud media de las oraciones y $std\big(l(s)\big)$ es la desviación estándar de las longitudes de las oraciones.

- ***Frecuencia de las palabras en el contenido***

Esta característica permite determinar qué tan relacionada esta una oración con el tema del documento, a partir del análisis de la frecuencia de aparición de las palabras que la conforman. Este esquema se basa en la premisa de que “Si una oración contiene palabras que sobrepasan algún umbral de repetición, entonces esta oración contiene muy probablemente información relevante” [1]. Esta heurística de selección, pese a su simplicidad, ha presentado buenos resultados en varios enfoques de generación de resúmenes automáticos, en algunos casos, acoplada con otros criterios de selección [3, 5, 9, 71-74]. Una de las técnicas más recientes para el cálculo del factor basado en la frecuencia de sus palabras, realiza una normalización basada en la función sigmoidea, como se ve en las Ecuaciones (3.5), (3.6) y (3.7).

$$F(s) = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}} \quad (3.5)$$

$$\alpha = \frac{CW(s) - \mu\big(CW(s)\big)}{std\big(CW(s)\big)} \quad (3.6)$$

$$CW(s) = -\sum_{i=1}^{k} \log[Freq(w_i)], \qquad w_i \in s \quad (3.7)$$

Donde $\mathrm{Freq}(\mathrm{w_i})$ es la frecuencia del término $\mathrm{w_i}$ en el documento, $\mu(\mathrm{CW}(s))$ es el pesomedio de todas las oraciones, $\mathrm{std}\big(\mathrm{CW}(s)\big)$ es la desviación estándar de los pesos de las oraciones.

- ***Puntaje de la oración basado en sus términos***

El puntaje de la oración basado en sus términos (WSS por sus siglas en inglés Word Sentence Score) es una característica que permite determinar la importancia de una oración a partir de la ponderación de cada una de las palabras que la componen. El puntaje total de una oración es calculado como la sumatoria de las ponderaciones[17] de todos sus términos sobre los máximos pesos entre ellos (Ver Ecuaciones (3.8) y (3.9)) [21]. De esta forma, esta técnica podría ser considerada como una *cobertura a nivel de términos*.

$$WSS(s_i) = 0.1 + \frac{\sum_{t_j \in s_i} w_{ij}}{HTFS} \quad (3.8)$$

$$HTFS = \sum \max w_{ij} \quad (3.9)$$

Donde w_{ij} es el peso del término t_j en la oración s_i. $HTFS$ corresponde a la sumatoria de los pesos más altos de los términos de la oración y *0.1* es el puntaje mínimo obtenido de la oración en caso de que sus términos no sean importantes.

- ***Similitud de una oración con otra***

La similitud de una oración con otra es, de hecho, uno de los criterios más empleados recientemente en la generación automática de resúmenes para la selección de las oraciones, y es comúnmente utilizada como elemento para definir los factores que determinan la aptitud del resultado final. Como se mencionó en secciones anteriores, dado que el modelo vectorial es el esquema de representación de documentos mayormente utilizado, el enfoque de medida de similitud de oraciones más común es el basado en cosenos, gracias a su acoplamiento a ese modelo y a su buen desempeño en diversas investigaciones (Ver Sección 2.3.3.1). Para la representación de las relaciones de similitud entre las oraciones de un documento, suele definirse una matriz adyacente que almacena los valores de dichas medidas. La matriz debe reflejar, además, el orden cronológico de las oraciones en el documento, por lo que la representación más frecuente

[17] Con cualquiera de las técnicas revisadas en la Sección 2.3.2

es la de una matriz triangular. Es así como el cálculo de similitudes se basa en ciertas premisas, como se ve en la Ecuación (3.10).

$$sim\left(s_i, s_j\right) = \begin{cases} 1, & i = j \\ 0, s_i \; aparece \; después \; de \; s_j \\ > 0, s_i \; aparece \; antes \; de \; s_j \end{cases} \quad (3.10)$$

De esta forma, como se muestra en la Figura 3.5, en el presente estudio la estructura propuesta es una matriz triangular inferior.

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{array} \begin{array}{c} \begin{array}{ccccc} s_1 & s_2 & s_3 & \cdots & s_n \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ sim(s_1,s_2) & 1 & 0 & \cdots & 0 \\ sim(s_1,s_3) & sim(s_2,s_3) & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ sim(s_1,s_n) & sim(s_2,s_n) & sim(s_3,s_n) & \cdots & 1 \end{bmatrix} \end{array}$$

Figura 3.5. Matriz de similitudes

- ***Factor de relación con el título***

Este factor (TRF por sus siglas en inglés Title Relation Factor) se basa en el supuesto de que un buen resumen contiene oraciones similares al título del documento [122]. Para calcular esta similitud, se parte de su representación a través del modelo de espacio vectorial, de tal manera que la medida más usual, y que se utiliza en la presente investigación, es la basada en cosenos [24, 25] (Ver Ecuaciones (3.11), (3.12), (3.13) y (3.14)). Sin embargo, existen algunos enfoques de generación de resúmenes automáticos, que realizan un cálculo más simple, basado en la cantidad de palabras comunes entre la oración y el título, no obstante, éste método es poco utilizado, debido a la limitación en su cálculo [6].

$$sim(s_j, t) = \frac{\vec{s_j} \cdot \vec{t}}{|\vec{s_j}| \times |\vec{t}|} \quad (3.11)$$

$$sim(s_j, t) = \frac{\sum_{i=1}^{k}(w_{ij} * w_{it})}{\sqrt{\sum_{i=1}^{k} w_{ij}^2} * \sqrt{\sum_{i=1}^{k} w_{it}^2}} \quad (3.12)$$

$$TR_s = \frac{\sum_{s_j \in resumen} sim(s_j, t)}{L} \quad (3.13)$$

$$TRF_s = \frac{TR}{máximo_{\forall \, resumen} TR} \quad (3.14)$$

Donde w_{ij} es el peso del término i en la oración j, w_{it} es el peso del término i en el título t, L es el número de oraciones del resumen, TR_s es el promedio de la similitud de las oraciones con el título en el resumen s, TRF_s es el factor de similitud de las oraciones del resumen s con el título y $máximo_{\forall\, resumen} TR$ es el promedio de los máximos valores obtenidos de las similitudes de las oraciones con el título.

En particular, el cálculo de este factor es muy favorable en la evaluación de utilidad de resúmenes de noticias, debido a la importancia que tienen los títulos y encabezados en este tipo de documentos, ya que reflejan gran parte de lo esencial del texto que presentan [111].

- ***Factor de cohesión***

El factor de cohesión (CF por sus siglas en inglés Cohesion Factor) es una métrica que determina el grado de relación de las oraciones que conforman un resumen [24, 25]. Idealmente, la conexión entre las ideas expresadas en las oraciones debe ser tal, que permita dar una unidad conceptual al resumen. Para su cálculo se utiliza la medida de similitud de una oración con otra. Revisando la Ecuación (3.15) se advierte que el factor de cohesión tiende a cero cuando las oraciones no tienen ninguna relación entre sí, por lo tanto, se dice que entre mayor sea la relación existente entre las oraciones mayor es el valor de cohesión.

$$CF = \frac{log(C_s * 9 + 1)}{log(M * 9 + 1)} \qquad (3.15)$$

$$C_s = \frac{\sum_{\forall si,sj \in resumen} sim(s_i, s_j)}{N_s} \qquad (3.16)$$

$$N_s = \frac{(L) * (L-1)}{2} \qquad (3.17)$$

$$M = máxima\ Sim(i,j), \qquad i,j \leq N \qquad (3.18)$$

Donde CF corresponde al factor de cohesión de un resumen, C_s, como se ve en la Ecuación (3.16), es el promedio de similitud de todas las oraciones en el resumen S, $sim(s_i, s_j)$ es la similitud entre las oraciones s_i y s_j, L es el número de oraciones del resumen, N_s, como se presenta en la Ecuación (3.17), es la cantidad de relaciones de similitud diferentes de 0 en el resumen, M corresponde a la máxima similitud de las oraciones y N es la cantidad de oraciones en el resumen S (Ver Ecuación (3.18)).

Existe otra forma aún más simple de calcular este factor y se basa, igualmente, en la medición de la similitud entre las oraciones del resumen (Ver Ecuación (3.19)). Este tipo de cálculo es conocido en cierta literatura como factor de redundancia, y es considerado un aspecto negativo para un resumen de múltiples documentos de un mismo tópico, ya que esto implica que el resumen contiene oraciones que manejan la misma información [20]. Pero para el caso de resúmenes de un solo documento este factor significa que existe cohesión entre las oraciones del resumen.

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sim(\overrightarrow{s_i}, \overrightarrow{s_j}) \quad (3.19)$$

Donde $\overrightarrow{s_i}$ es el vector de pesos de los términos de la oración s_i, $\overrightarrow{s_j}$ es el vector de pesos de los términos de la oración s_j y $sim(\overrightarrow{s_i}, \overrightarrow{s_j})$ es la similitud entre las oraciones s_i y s_j.

- ***Factor de legibilidad***

El factor de legibilidad (RF por sus siglas en inglés Readability Factor) es una métrica que establece el grado en que las oraciones del resumen están relacionadas con las oraciones que las preceden [24, 25]. El cálculo de este factor se basa en el criterio de similitud de una oración con otra, medido entre oraciones consecutivas (ver Sección □). De esta manera, la evaluación de este factor se realiza como se muestra en la Ecuación (3.20).

$$RF_s = \frac{R_s}{máximo_{\forall i} R_i} \quad (3.20)$$

$$R_s = \sum_{0 \le i < L} sim(s_i, s_{i+1}) \quad (3.21)$$

Donde L es la longitud del resumen generado, R_s, como se ve en la Ecuación (3.21), es la legibilidad de un resumen S con longitud L y $máximo_{\forall i} R_i$ hace referencia al resumen más legible de longitud L.

Para encontrar el resumen más legible, basta con encontrar la ruta de longitud L con los máximos valores en la matriz de similitudes. Así, dado un resumen compuesto por las oraciones $s_1, s_2, \dots, s_L$, se trata de maximizar la suma $sim_{s_1,s_2} + sim_{s_2,s_3} + \cdots + sim_{s_{L-1},s_L}$. Para ello, se define una matriz auxiliar M en la que la i-ésima fila corresponde a la oración i, mientras que la j-ésima columna muestra la ruta de longitud j. De esta manera, $M_{i,j}$ sería igual al peso de la ruta más larga hacia la oración i con longitud j (Ver Ecuación (3.22)).

$$M_{l,j} = \max_{i<l}(M_{i,j-1} + sim_{i,l}) \quad (3.22)$$

Como se ve en la Figura 3.6, el llenado de la matriz se realiza a través de un algoritmo que toma como entrada la longitud del resumen deseado y la matriz de similitudes. Para obtener la ruta con el peso máximo de longitud x, se debe encontrar la celda con máximo valor en la columna x de M.

```
FUNC LlenarMatrizRutasMasCostosas
       (Sim:MatrizSimilitudes[][], l: ℕ, longDocumento: ℕ)
Variables
       m : ℕ;
       n : ℕ;
Inicio
       PARA i ← 0 HASTA m HACER
              PARA j ← 0 HASTA n HACER
                     M[i,j] ← 0;
              FINPARA
       FINPARA
       PARA j ← 1 HASTA m HACER
              PARA i ← 1 HASTA n HACER
                     PARA k ← 1 HASTA i - 1 HACER
                            SI (M[i,j] < M[k,j-1] + Sim[k,i])ENTONCES
                                   M[i,j] ← M[i,j-1] + Sim[k,i];
                     FINPARA
              FINPARA
       FINPARA
Fin
```

Figura 3.6. Algoritmo para llenar matriz auxiliar de pesos

- ***Factor de cobertura***

El factor de cobertura[18] intenta medir el grado en que un resumen proporciona al lector la información más importante del documento original, a través del contenido de las oraciones que lo constituyen [20]. De esta manera, este factor ha sido técnicamente definido como la similitud entre el vector de cada oración que compone el resumen y el vector de todas las oraciones del documento. Así pues, el documento, al igual que cada una de sus oraciones, puede ser representado a través del modelo vectorial y ser pesado con cualquiera de las técnicas de ponderación de términos. Siendo así, el factor de cobertura se define como en la Ecuación (3.23).

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} [sim(\vec{D}, \vec{s_i}) + sim(\vec{D}, \vec{s_j})] \quad (3.23)$$

Donde n es el número de oraciones del documento, $\vec{D}$ es el vector de pesos de los términos del documento, $\vec{s_i}$ corresponde al vector de pesos de los términos de la oración s_i y $\vec{s_j}$ es el vector de pesos de los términos de la oración s_j.

3.1.3.2 Configuración preliminar de la función objetivo

Como se citó anteriormente, la definición de la función objetivo es un aspecto de suma importancia en el diseño del algoritmo memético. Sin embargo, teniendo en cuenta que al inicio de la presente investigación aún no se tenía establecido el esquema evolutivo más apropiado conforme a las condiciones del problema abordado, pero aun así era imprescindible contar con un diseño de la función objetivo para su funcionamiento, se optó

[18] También llamado, en algunos casos, factor de relevancia.

por iniciar con una configuración provisional que abarcara gran parte de los factores estudiados en la Sección 3.1.3.1, considerando los buenos resultados de su aplicación, presentados en proyectos similares.

Así pues, la función objetivo inicial, con la que se realizaron las primeras pruebas, fue definida como se ve en la Ecuación (3.24), cuya fórmula matemática se muestra en la Ecuación (3.25). (Algunos aspectos adicionales sobre la configuración de la función objetivo preliminar pueden verse en el *Anexo D*).

$$f(x) = Posición + Longitud + CoberturaWSS + Cohesión + Legibilidad \quad (3.24)$$

$$f(x) = \sum_{i=1}^{m}\left[\sqrt[2]{\frac{1}{n_i}} + \frac{1 - e^{-\frac{l(s_i)-\mu(l)}{std(l)}}}{1 + e^{-\frac{l(s_i)-\mu(l(s_i))}{std(l)}}} + \left(0.1 + \frac{\sum_{t_j \in s_i}^{k} w_{ij}}{\sum_{i=1}^{k} \max w_{ij}}\right)\right] + \frac{log(C_{s_i} * 9 + 1)}{log(M * 9 + 1)} + \frac{\sum_{j=i}^{n-1} sim(s_i, s_{i+1})}{máximo_{\forall i} R_i} \quad (3.25)$$

3.1.4 Configuración preliminar de parámetros

Para iniciar los ciclos experimentales de esta investigación, fue preciso establecer un conjunto de valores preliminares de los parámetros del algoritmo memético, los cuales fueron definidos considerando las sugerencias presentadas en la literatura sobre el diseño de algoritmos evolutivos [84, 88, 96, 123-125], sin embargo, todas ellas concluyen que dichos valores dependen del problema, por lo que los valores se establecieron tratando de que estuvieran dentro de un rango apropiado para las características del problema abordado en esta investigación, de la siguiente forma:

- Tamaño de la población, se recomienda definir un valor que permita la exploración del espacio de búsqueda, sin que sea tan grande que lleve a la degeneración en la eficiencia del algoritmo de tal forma que ninguna solución pueda ser alcanzada en un tiempo razonable [88]. Dicha premisa, según Reeves [126], puede ser satisfecha si hay al menos una instancia del alfabeto considerado (0 o 1, para el alfabeto binario) en cada gen en la población completa de individuos. Reeves expone el cálculo de la probabilidad $\mathrm{P_q^*}$ de que al menos un alelo esté presente en cada gen de la población inicial, la cual para un alfabeto binario, se define como en la Ecuación (3.26).

  $$\mathrm{P_2^*} = (1 - (1/2)^{\mathrm{M-1}})^{\mathrm{L}} \quad (3.26)$$

  Donde, M es el tamaño de la población inicial y L es el tamaño de un cromosoma. En ese sentido, teniendo en cuenta que el tamaño medio de los cromosomas de los agentes en esta investigación es de 27 (cantidad media de oraciones en los documentos evaluados), puede establecerse que para el tamaño de la población, un valor de 20 es adecuado, pues asegura que la probabilidad $\mathrm{P_2^*}$ exceda el 99.99%.

- Elitismo, hace referencia a los agentes que pasan a la siguiente generación; se seleccionaron solamente 2 agentes, teniendo en cuenta el tamaño de la población, esto para evitar la predominancia de los agentes más aptos.

- Probabilidad de mutación, teniendo en cuenta las recomendaciones de la literatura [89, 95, 96], se definió un valor bajo de 0.4.

- Número de optimizaciones, indica la profundidad o intensidad de la búsqueda local, este parámetro se definió en 5, de acuerdo al tamaño de población y con el fin de evitar el aumento del costo computacional de valores de profundidad muy altos [125].

- Máxima longitud a evaluar del documento, este parámetro permite que durante el proceso de evolución del algoritmo se exceda la cantidad de frases que pueden ser incluidas en un agente, con el objetivo de realizar una mejor exploración del espacio de búsqueda. Sin embargo, cuando se termina el proceso de evolución (condición de parada del algoritmo) se valida que la cantidad de palabras que conforman el resumen final sea de 100 palabras, para cumplir con la restricción del problema. Este parámetro se definió en 150, un valor superior en 50 palabras a la longitud del resumen.

En la Tabla 4 se lista la configuración preliminar de los parámetros descritos.

Parámetro	Valor
Tamaño de la población	20
Cantidad de agentes elegidos por elitismo en generaciones	2
Cantidad de agentes elegidos por elitismo en reiniciación	2
Máxima longitud a evaluar de un documento	150
Probabilidad de mutación	0.4
Número de optimizaciones	5

Tabla 4. Configuración preliminar de los parámetros del MA

3.2 CICLO II: DEFINICIÓN DE LOS OPERADORES REPRODUCTIVOS DEL ALGORITMO MEMÉTICO

Este ciclo de definición de operadores se realizó a partir de la configuración inicial del algoritmo memético propuesto, presentada en la Tabla 3. Por otra parte, este ciclo se desarrolló utilizando sólo 10 conjuntos de documentos de DUC 2002. Esta decisión se tomó con el propósito de reducir el tiempo de ejecución y el costo computacional que conlleva el desenvolverse sobre un gran número de documentos como el de DUC 2002 dentro de un esquema de generación de un resumen por documento y bajo la evaluación de gran cantidad de características de complejidad media, enmarcadas en la función objetivo. Este proceso se dividió en cuatro etapas, donde cada una contempla el estudio sobre un operador reproductivo diferente. En seguida, se describe en forma breve cada una de estas etapas (Para más detalle referirse al *Anexo B*).

3.2.1 Primera etapa: Operador de Selección

Esta etapa se llevó a cabo con el fin de establecer la pareja de métodos con los que se seleccionarían los agentes padre y madre que participarían en la fase de reproducción del algoritmo. De esta forma, se estudió el comportamiento de los diferentes métodos de selección sobre cada progenitor, de tal modo que esta etapa se dividió en seis grupos. Cada grupo, consideró un método de selección para el padre, mientras lo combinaba con

siete métodos diferentes de selección para la madre. Los demás operadores reproductivos del algoritmo permanecieron invariantes durante esta etapa, con el fin de enfocar el análisis sólo en la discriminación del desempeño de cada pareja de técnicas de selección, obteniendo, así, las dos combinaciones con mejor comportamiento de cada grupo, es decir, doce combinaciones en total, de las cuales, al final de esta etapa, se escogieron las cinco con mejor desempeño, con las que se continuó el proceso de definición del operador de cruce (Ver Tabla 5).

Pareja	Selección Padre	Selección Madre
1	Rueda De Ruleta	Competencia Restringida
2	Torneo Probabilístico	Rango Aleatorio
3	Torneo Determinístico	Aleatorio
4	Basada En Rango	Rueda De Ruleta
5	Rango Aleatorio	Aleatorio

Tabla 5. Mejores parejas de operadores de selección obtenidas

3.2.2 Segunda etapa: Operador de Cruce

Las cinco parejas de los métodos de selección obtenidas en la etapa anterior (Ver Tabla 5), fueron evaluadas con cada uno de los métodos de cruce, para analizar en qué grado se ve afectado el comportamiento del algoritmo memético con las diferentes configuraciones de estos dos operadores. De este modo, se formaron cinco grupos correspondientes al *Cruce de un punto, Cruce de dos puntos, Cruce uniforme, Cruce de anillo* y *Cruce plano*. Tras el análisis de los resultados obtenidos en cada grupo, se llegó a dos aspectos destacables, el primero de ellos fue que el método de cruce que mejor desempeño presentó fue el *Cruce de un Punto*, acoplado con la pareja 5 de métodos de selección. De acuerdo a los resultados presentados en la Tabla 25 del *Anexo* B, no se puede suponer que la configuración más adecuada del algoritmo esté formada siempre por los mejores métodos de cada operador. Por tal motivo, con el fin de no obviar configuraciones que podrían mejorar los resultados, en la siguiente etapa, correspondiente a la evaluación de los métodos de mutación, se continuó realizando las pruebas con las cinco mejores parejas de los métodos de selección junto al mejor método de cruce: *el Cruce de un Punto.*

3.2.3 Tercera etapa: Operador de Mutación

Esta etapa inicia a partir de las cinco configuraciones formadas por las mejores parejas de los métodos de selección y el *Cruce de un Punto*, y se dividió en dos grupos: *Mutación de Bit* y *Mutación Multi-Bit*. Estas dos técnicas intervienen en la forma de llevar a cabo la mutación, decidiendo la cantidad de bits que serán afectados con dicho proceso, sin embargo, la alteración, como tal, de la información de una solución, es realizada por un método de mutación adicional, el cual es integrado dentro de cualquiera de estas dos técnicas, y es referido dentro de esta investigación como método de *Mutación Interna*, mientras que las dos técnicas mencionadas serán citadas como métodos de *Mutación Externa*. Por tal motivo, los dos grupos formados para esta etapa se dividen, a su vez, en

subgrupos, los cuales cada uno considera un método de Mutación Interna diferente, entre ellos, *Mutación de Inserción, Mutación de Intercambio* y *Mutación Compuesta*.

De esta forma, analizando los resultados obtenidos, se observó que el mejor desempeño, tanto en la Mutación de Bit como en la Multi-Bit, fue presentado por la *Mutación de Inserción*. En ese sentido, la combinación de *Mutación Multi-Bit* y *Mutación de Inserción* como técnicas externa e interna, respectivamente, fue la mejor dentro de esta etapa. Adicionalmente, se observó que la mayoría de pruebas con buenos resultados evaluaban configuraciones con la pareja 4 de métodos de selección, conformada por la selección *Basada En El Rango* para el padre y la selección de *Rueda de Ruleta* para la madre, por lo cual se presumió que esta pareja tendía a mejorar su desempeño al acoplarse con los demás operadores. A pesar de ello, las pruebas de la siguiente etapa se continuaron con cuatro parejas de selección, sin tener en cuenta la pareja 1 que presentó los resultados más bajos en la mayoría de las pruebas, con el fin de encontrar, entre ellas, otra pareja, que al acoplarse con el operador de reemplazo, proporcionara mejores resultados que los obtenidos hasta el momento.

3.2.4 Cuarta etapa: Operador de Reemplazo

Para realizar la evaluación de los mecanismos de reemplazo estudiados, se tomaron los mejores métodos de cruce y mutación obtenidos hasta el momento, junto a las cuatro parejas de métodos de selección que aún se encontraban en evaluación. En esta etapa, se formaron seis grupos correspondientes a los diferentes métodos de reemplazo, como son: *Reemplazo por Competencia Restringida, Reemplazo del Peor, Reemplazo del Peor Padre, Reemplazo del Cercano* y *Reemplazo de Similar*. De esta forma, analizando los resultados de esta etapa, se llegó a que el mejor desempeño era presentado por el método de *Competencia Restringida*. Así mismo, finalizada la evaluación de todos los operadores reproductivos, se apreció que la pareja que tuvo el mejor acople con todos los operadores fue la 4, como se había observado en etapas anteriores. Así pues, los mejores métodos de selección fueron la *Selección Basada en el Rango* para el padre y *Selección Rueda de Ruleta* para la madre. De este modo, como producto final de esta etapa se obtiene el esquema del MA que se muestra en la Tabla 6, con el cual se continuó el ciclo de definición del operador de búsqueda local.

Operador	Método
Selección padre	Basada en el Rango
Selección madre	Rueda de Ruleta
Cruce	Cruce de un Punto
Mutación Externa	Mutación Multi-Bit
Mutación Interna	Mutación de Inserción
Reemplazo	Competencia Restringida
Optimización Local	Búsqueda Local Básica del Mejor

Tabla 6. Configuración del MA al final del ciclo de definición de operadores reproductivos

3.3 CICLO III: DEFINICIÓN DEL OPERADOR DE BÚSQUEDA LOCAL DEL ALGORITMO MEMÉTICO

El proceso de definición del operador de búsqueda local se llevó a cabo tras la definición de los operadores reproductivos del algoritmo memético y, al igual que en dicho ciclo, se utilizaron sólo 10 conjuntos de documentos de la colección de DUC 2002. Por otro lado, este proceso se dividió en 5 etapas. Cada etapa se enfoca en el análisis de un mecanismo de búsqueda local. En forma general, el proceso queda dividido, entonces, en: *Búsqueda Local Básica, Búsqueda Local de Entorno Variable, Búsqueda Local Guiada, Búsqueda Tabú* y, finalmente*, Búsqueda Local Iterativa.* La primera etapa se enfocó en la evaluación de una técnica de búsqueda local simple, mientras que las demás etapas examinaron propuestas más elaboradas para ayudar al mecanismo de búsqueda local a enfrentar el estancamiento en óptimos locales y conseguir soluciones más prometedoras. De esta manera, cada uno de los métodos descritos a partir de la segunda etapa, partieron de la técnica especificada en la primera para llegar, a través de la adaptación, a una estrategia de búsqueda más sofisticada. En la mayoría de las etapas desarrolladas se evaluaron diferentes implementaciones de la búsqueda respectiva y se observó que el mejor desempeño lo presentaban aquellas adaptaciones que consideraban con más detalle las características del algoritmo memético propuesto. Además, en las pruebas ejecutadas, se afinaron algunos parámetros para aquellas búsquedas que lo requerían.

Al final de este proceso, se recopilaron las mejores configuraciones obtenidas por cada etapa, llegando a que el mejor desempeño fue presentado por la *Búsqueda Local Guiada*. De esta forma, al concluir este ciclo, se obtiene el esquema final del algoritmo memético como se ve en la Tabla 7. La descripción detallada de la definición del operador de búsqueda local es presentada en al *Anexo C*.

Operador	Método
Selección padre	Basada en el Rango
Selección madre	Rueda de Ruleta
Cruce	Cruce de un Punto
Mutación Externa	Mutación Multi-Bit
Mutación Interna	Mutación de Inserción
Reemplazo	Competencia Restringida
Optimización Local	Búsqueda Local Guiada

Tabla 7. Configuración del MA al final del ciclo de definición del operador de búsqueda local

3.4 CICLO IV: DISEÑO FINAL DE LA FUNCION OBJETIVO

El proceso para definir una función objetivo que se adaptara a las necesidades del problema, fue dividido en dos etapas. En la primera de ellas, se llevó a cabo un número de pruebas sobre sólo 10 conjuntos de documentos de DUC 2002, similar a como se hizo en el ciclo de definición de operadores reproductivos. En la segunda, se trabajó sobre todo el conjunto de datos del mismo corpus (Referirse al *Anexo D* para más detalle).

3.4.1 Primera etapa de diseño

En la primera etapa de definición de la función objetivo, las pruebas se distribuyeron en tres bloques. El primer bloque está conformado por nueve pruebas, donde siete se consideran pruebas básicos, en los que la configuración de la función objetivo está definida por una sola característica, y los otros dos presentan una estructura más compleja, ya que están definidos por una combinación de características. Las pruebas realizadas en este bloque tienen como finalidad evaluar el comportamiento individual de cada una de las características seleccionadas, así como su desempeño en conjunto. De acuerdo a los resultados obtenidos, se evaluó el nivel de desempeño de las características con el propósito de establecer criterios de acoplamiento entre ellas que permitieran definir las pruebas del segundo bloque.

Con el segundo bloque de pruebas, se busca definir las mejores combinaciones de las características probadas en la parte inicial del primer bloque. De esta manera, se procede a la evaluación mediante pequeñas agrupaciones de pruebas, donde los resultados arrojados por un grupo son determinantes para la definición del siguiente. Las pruebas que conforman cada grupo se caracterizan por la inclusión o exclusión de una o más características dentro de la configuración correspondiente. Dicho proceso tiene como finalidad determinar cuáles características conviene considerar en la función objetivo, de acuerdo a los resultados entregados. Los resultados obtenidos tras la ejecución de este bloque, presentados en la Tabla 73 del Anexo D, fueron concluyentes en varios aspectos. En primer lugar, se determinó que la *Cobertura a nivel de oraciones* presentaba un mejor desempeño que la *Cobertura a nivel de términos WSS*, por lo que se resolvió seguir experimentando sólo con la primera. Por otro lado, se concluyó que el comportamiento del factor de *Relación con el título* depende en gran medida del grupo de características con las que deba acoplarse y que se comporta mejor que la *Legibilidad*, al igual que la *Cobertura* ofrece un mejor desempeño que la *Cohesión*. Finalmente, se pudo detallar que la *Cohesión* afectaba la uniformidad de los resultados entre las medias de recuerdo, precisión y medida-F.

Considerando las configuraciones con mejor resultado del bloque anterior, el tercer bloque de pruebas busca mejorar el desempeño de la función objetivo, variando la manera de calcular la *Posición* y adicionando un nuevo cálculo de *Cohesión*. De acuerdo al análisis realizado, se observa que no es adecuado utilizar una función objetivo que considere la *Legibilidad* en lugar de la *Posición* y la *Longitud*. También se observó que al considerar otro cálculo de *Cohesión* y el cálculo de *Posición* basado en Wan [36] (Ver Ecuación (3.2)), el desempeño de la función objetivo es mejor que si se incluyen otras características, por lo que se decide tenerlas en cuenta para su análisis en futuras pruebas.

Como producto de esta primera etapa, se seleccionaron las 5 mejores configuraciones obtenidas en los tres bloques de pruebas realizados, de tal manera que constituyan el punto de partida para la siguiente etapa de este proceso. Entre ellas, el mejor desempeño se obtiene con la configuración de la función objetivo que se indica en la Ecuación (3.27) .

$$f(x) = Posición + Longitud + Relación\ con\ el\ título + Cobertura + Cohesion1 \quad (3.27)$$

3.4.2 Segunda etapa de diseño

En la segunda etapa de definición de la función objetivo, se continuó realizando dos bloques de pruebas más. El primer bloque se ejecutó con el fin de analizar el desempeño, con todo el conjunto de documentos, de las cinco mejores configuraciones de la función objetivo obtenidas en la primera etapa, incluyendo, además, como punto de referencia, la configuración preliminar de la función objetivo. De acuerdo a los resultados obtenidos, se confirmó que a partir de sólo los 10 conjuntos de documentos fue posible obtener configuraciones con un mejor desempeño a la función objetivo preliminar, logrando, a la vez, ahorro en costo computacional. Además, se observó que el desempeño de una función objetivo varía un poco de una etapa a otra, es decir, que depende de los documentos considerados, pues en este caso se evaluó el mismo bloque de pruebas con diferentes conjuntos de datos del mismo corpus y los resultados obtenidos fueron muy distintos. En ese sentido, se optó por evaluar el segundo bloque de pruebas, con el fin de investigar si es posible encontrar una configuración aún mejor. En este bloque se tomó, en forma aleatoria, una de las configuraciones realizadas en la primera etapa. Adicionalmente, se crean dos pruebas más que corresponden a pequeñas modificaciones, como el cambio en el cálculo de *Posición* y la exclusión de la *Legibilidad*. Finalmente, se diseñó una función objetivo que incluyó las dos características con mejor desempeño individual en la primera etapa, es decir, la *Posición* y la *Relación con el título*. Los resultados obtenidos fueron comparados con la prueba que obtuvo el mejor desempeño en el bloque anterior, sin encontrar ninguno que lo superara. Por lo tanto, se concluyó que la configuración de la función objetivo con mejor desempeño corresponde a la presentada en la Ecuación (3.28), con su correspondiente fórmula matemática en (3.29).

$$f(x) = Posición + Longitud + Relación\ con\ el\ título + Cohesión + Cobertura \tag{3.28}$$

$$f(x) = \sum_{i=1}^{m}\left[\sqrt[2]{\frac{1}{n_i}} + \frac{1 - e^{-\frac{l(s_i)-\mu(l)}{std(l)}}}{1 + e^{-\frac{l(s_i)-\mu(l)}{std(l)}}} + \frac{sim(s_i, t)}{L * \ máximo_{\forall\ resumen} TR}\right] + \frac{log(C * 9 + 1)}{log(M * 9 + 1)} + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left[sim\left(\vec{D}, \vec{s_i}\right) + sim\left(\vec{D}, \vec{s_j}\right)\right] \tag{3.29}$$

Tras examinar todo el proceso que llevó a definir la función objetivo final, se observa que gran parte de las configuraciones con mejor resultado incluyen características como la *Posición*, *Relación con el título* y *Cohesión*. Debido al tipo de datos con los que se está trabajando, estos resultados confirman investigaciones previas del estado del arte [24, 25, 36, 115], en las cuales, el uso de estas tres características brinda resultados muy favorables en la creación de resúmenes y, en forma particular, la *Posición* y la *Relación Con El Título* son determinantes en la evaluación de resúmenes automáticos de noticias.

3.5 CICLO V: AFINACIÓN DE PARAMETROS DEL ALGORITMO MEMÉTICO

Para realizar la afinación de los parámetros utilizados por el algoritmo memético, presentados en la Tabla 4 de la Sección 3.1.4, se utilizó la configuración obtenida en el ciclo de definición de los operadores reproductivos y de optimización local (Ver Tabla 7 de la Sección 3.3) al igual que la función objetivo obtenida, también, en un ciclo anterior (Ver Sección 3.4).

En este proceso, para efectuar la afinación de un parámetro particular, se varió su valor conservando constantes los demás parámetros. Por otro lado, teniendo en cuenta que la configuración final del algoritmo memético incluye el método de Competencia restringida como estrategia de reemplazo, el cual selecciona aleatoriamente una cantidad predefinida de agentes competidores establecida por un parámetro de tamaño del grupo de competencia, se decidió realizar también la afinación de ese parámetro, buscando mejorar el desempeño del algoritmo.

De esta forma, la afinación inicia con el *Tamaño de la población*, para el cual se evaluaron cinco valores diferentes como son 10, 15, 20, 25 y 30 agentes. La evaluación con más de 30 agentes no se realizó, teniendo en cuenta la premisa presentada anteriormente, ya que el tamaño de 30 es suficiente para asegurar que la probabilidad P_2^* exceda el 99.99%. Además, debido a que la cantidad de oraciones extraíbles de un solo documento es baja, el número de agentes diferentes que pueden crearse, y que cumplan con las restricciones de longitud del resumen, difícilmente logra superar dicho valor. Al final de esta evaluación el mejor comportamiento fue presentado por un tamaño de población de 30 agentes.

Partiendo del tamaño de población obtenido, la siguiente afinación fue realizada sobre el *número de optimizaciones*, con valores de 3, 4 y 5, llegando a que el mejor comportamiento fue conseguido al utilizar 3 optimizaciones.

Utilizando los valores de estos dos parámetros, se realizó la afinación de la cantidad de agentes elegidos por *elitismo en reiniciación*, evaluando valores de 1, 2, 3 y 4 agentes, donde el mejor desempeño fue presentado con el valor de 1. De esta forma, se procedió a afinar la cantidad de agentes elegidos por *elitismo en generaciones*, considerando los valores 1, 2, 3, 4 y 5 agentes, obteniendo los mejores resultados con un valor de 1.

Teniendo en cuenta la configuración de parámetros con mejores resultados obtenida hasta el momento, se afinó el tamaño del *grupo de competencia*, evaluando los valores 2, 3, 4, 5 y 6, obteniendo los mejores resultados con un tamaño de 4.

Con los valores obtenidos, la siguiente afinación se llevó a cabo sobre la *máxima longitud a evaluar de un documento*. Esta afinación se realizó en tres grupos. En el primero se evaluaron valores constantes de 250, 350 y 450 palabras. El segundo grupo se definió con el objetivo de eliminar la parcialidad asociada al manejo de valores fijos de este parámetro y dar iguales condiciones a todos los documentos, teniendo en cuenta que la diferencia de longitudes entre documentos es considerable (el documento más corto tiene 135 palabras, mientras que el más largo está formado por 2966), en ese sentido, el valor de este parámetro, en este grupo, se calcula de acuerdo a un porcentaje definido, de tal forma que los porcentajes evaluados fueron 40, 50, 60 y 70 palabras. El tercer grupo se constituyó por seis pruebas, en las cuales este parámetro es calculado como un

porcentaje seleccionado aleatoriamente entre un rango. Los rangos evaluados fueron 30-60%, 30-70%, 40-60%, 40-70%, 50-60%, 50-70%. Los mejores resultados de cada grupo fueron presentados por el valor fijo de 450, el porcentaje de 70%, y el rango de 50-70%; el primero fue descartado debido a que era un valor muy alto para algunos documentos (tomando prácticamente la totalidad de oraciones del documento); el segundo también era un valor muy alto, considerando gran parte de las oraciones del documento; de ese modo, el valor que se seleccionó para este parámetro fue el rango de 50-70%, ya que permite considerar diferentes valores para un agente, ayudando a eliminar la parcialidad asociada a valores fijos.

La siguiente afinación se realizó a la *probabilidad de optimización*, cuyo objetivo fue evaluar el impacto de la aplicación de optimización local sobre el comportamiento del algoritmo memético propuesto. De esta manera, con el fin de discriminar el desempeño del algoritmo al variar la frecuencia con que se aplica la optimización, se definieron diez diferentes valores de probabilidad de optimización desde 0.1 hasta 1.0 (con variaciones de 0.1), llegando a que la optimización frecuente de 100% (1.0) conducía a los mejores resultados.

Por último se realizó la afinación de la *probabilidad de mutación*, con el propósito de evaluar el impacto de la frecuencia de aplicación del operador de mutación en el algoritmo memético propuesto. De esta manera, se definieron cuatro diferentes valores de probabilidad de mutación como son 0.1, 0.2, 0.3 y 0.4. Como se explicó en la sección 3.1.4 se recomienda manejar una probabilidad de mutación baja, por esto el máximo valor evaluado para este parámetro fue de 0.4, siendo este valor el que logro obtener los mejores resultados.

La configuración obtenida al final de este proceso se presenta en la Tabla 8. (Para más detalle referirse al Anexo E).

Parámetro	Valor
Tamaño de la Población	30
Número de Optimizaciones	3
Elitismo en Generación	1
Elitismo en Reiniciación	1
Tamaño del Grupo de Competencia	4
Máxima longitud a evaluar del documento	Entre 50%-70% de la longitud del documento
Probabilidad de Optimización	1
Probabilidad de Mutación	0.4

Tabla 8. Configuración final de parámetros del Algoritmo Memético

3.6 CICLO VI: AFINACIÓN DE PESOS DE LA FUNCIÓN OBJETIVO

El proceso de afinación de pesos de la función objetivo del MA se inicia partir de la configuración presentada en la Ecuación (3.30) y se divide en dos etapas. En la primera se diseñó un algoritmo genético (GA), con el fin de obtener varios rangos de pesos, con los cuales, posteriormente, se evalúa la función objetivo con el MA, para determinar cuál presenta el mejor desempeño. En la segunda etapa, se utiliza como referencia el mejor

conjunto de pesos obtenido en la primera etapa, para generar nuevos conjuntos de pesos, que son evaluados con el fin de obtener un mejor desempeño de la función objetivo.

$$f(x) = \alpha * \text{Posición} + \beta * \text{Longitud} + \gamma * \text{Cohesión} + \mu * \text{Cobertura} + \rho * \text{Relación con el título} \quad (3.30)$$

3.6.1 Primera etapa: Pesos iniciales obtenidos con el GA diseñado

Esta etapa se dividió en tres grupos, en los cuales mediante el GA, se obtuvieron tres conjuntos de pesos diferentes. El primer grupo de pesos fue obtenido con tres ejecuciones diferentes del GA, en las cuales se consideraron tres funciones objetivo, correspondientes a la medida-F de ROUGE-1, ROUGE-2 y ROUGE-SU4, obtenidas al evaluar el resumen generado por el MA, que, en los tres casos, se ejecutaba internamente con 10 conjuntos de documentos[19]. Los otros dos grupos fueron configurados en forma similar, sólo que en lugar de usar la medida-F como función objetivo, utilizaron los valores de recuerdo y precisión de las tres métricas de ROUGE. Los nueve conjuntos de pesos obtenidos a partir del GA fueron evaluados en la función objetivo del algoritmo memético, con todos los documentos de DUC2002, llegando a que los mejores resultados fueron presentados por los conjuntos de pesos de la Tabla 9.

Posición	Longitud	Cobertura	Cohesión	Relación con el título
α	β	γ	μ	ρ
0,30	0,20	0,10	0,10	0,30
0,36	0,38	0,005	0,005	0,25
0,395	0,395	0,005	0,005	0,20

Tabla 9. Conjuntos de pesos obtenidos en la primera etapa de afinación de pesos

3.6.2 Segunda etapa: Afinación de pesos

En la segunda etapa se definieron cinco grupos de pruebas. En el primero se tomó el primer conjunto de pesos obtenido en la etapa anterior para realizar pequeñas variaciones decimales sobre cada uno de los pesos, con el fin de obtener mejores resultados, así mismo, el segundo y tercer grupo tomaron el segundo y tercer conjunto de pesos, respectivamente, para realizar pequeñas afinaciones. El cuarto grupo fue sólo una recopilación de las mejores pruebas obtenidas en los tres grupos anteriores, con el fin de analizar su comportamiento y determinar cuál de ellos presentaba el mejor desempeño. Finalmente, el quinto grupo, con base en el análisis del grupo anterior, toma la prueba más destacada para realizar pequeñas variaciones decimales por encima y por debajo de los pesos evaluados en dicha prueba, sin embargo, ninguna de esas variaciones logró superar los resultados de los pesos originales. De esta manera, al final de esta etapa, la función objetivo queda definida como en la Ecuación (3.31) (Para más detalle referirse al *Anexo E*).

$$f(x) = 0{,}35 * \text{Posición} + 0{,}29 * \text{Longitud} + 0{,}005 * \text{Cohesión} + 0{,}005 * \text{Cobertura} + 0{,}35 * \text{Relación con el título} \quad (3.31)$$

[19] D061j, D62j, D063j, D066j, D067j, D070j, D071f, D074b, D097e, D113h

3.7 CICLO VII: EVALUACIÓN ADICIONAL DE CRITERIOS DE SELECCIÓN DE ORACIONES

Considerando que durante la afinación del parámetro correspondiente a la *máxima longitud a evaluar del documento* (Sección 3.5) se observó que el mejoramiento de los resultados de las medidas de ROUGE crecía directamente proporcional al valor del dicho parámetro, se presumió que además de que el diseño de la función objetivo propuesta tenía una gran influencia favorable sobre el comportamiento del algoritmo memético, las características que la componían tenían, individualmente, un efecto positivo sobre el tipo de datos que se estaba evaluando. De acuerdo a estas consideraciones, a pesar de no estar dentro del alcance planteado en esta investigación, se decidió definir algunas pruebas adicionales sin la intervención del algoritmo memético, con el propósito de analizar la influencia individual que ejercen las características que conforman la función objetivo propuesta, sobre la selección de las oraciones del documento.

En ese sentido, las pruebas definidas, evaluaron diferentes criterios de selección de oraciones, los cuales utilizaron una o más características de la función objetivo, que se calcula para cada oración del documento, de modo que se obtenga un puntaje individual para cada oración, el cual determinaría el orden en que serían seleccionadas estas oraciones para conformar el resumen. Este análisis se realizó con los conjuntos de documentos de DUC 2002 y DUC 2001. Las pruebas se dividieron en seis grupos de acuerdo a la combinación de características analizada, variando, en cada uno, el tipo de criterio utilizado, y analizando las posiciones de las frases que conformaban los resúmenes generados. El primer grupo se conformó por dos pruebas, correspondientes a la evaluación de la línea base, que consiste en formar el resumen con las 100 primeras palabras del documento, y al criterio de selección basado en la combinación de Posición, Longitud, Relación con el Título, Cohesión y Cobertura. En el segundo grupo se evaluó el comportamiento individual de cada una de las características anteriores. En los grupos restantes, se analizó el comportamiento de los criterios de selección basados en parejas de características (por ejemplo, cobertura y longitud, cohesión y relación el título, etc). De acuerdo al análisis de estos resultados, se observó que las tres características que ofrecían los mejores resultados, correspondían a la Cohesión, Cobertura y Posición, que a su vez incrementaban su buen desempeño trabajando conjuntamente.

Por otro lado, según el análisis realizado a las oraciones que conforman los resúmenes, se observó que el criterio de selección basado en la Cohesión, permitía que el resumen estuviera conformado principalmente por las oraciones iniciales del documento, presentando un comportamiento bastante similar al obtenido con la combinación de las cinco características de la función objetivo, pero con mejores resultados en las medidas de ROUGE. Este comportamiento de la Cohesión, se atribuyó a que para ese caso, sin la utilización del algoritmo memético, se analizaba el documento en su totalidad, sin limitar su longitud, por lo tanto su cálculo se realizaba para todas las oraciones del documento, y no solamente para las oraciones activas de una solución (agente), como se hacía en el algoritmo memético, lo que implicaba que se diera mayor relevancia a las oraciones iniciales del documento de acuerdo al valor obtenido como Cohesión. Adicionalmente, la Cobertura a diferencia de la Cohesión, permitió seleccionar oraciones comprendidas en diferentes posiciones del documento, pero sacrificaba en cierto porcentaje los resultados de las medidas de ROUGE. Seguidamente, se determinó que de todas las pruebas el mejor desempeño se obtuvo con el criterio de selección basado en la pareja de

características formada por la Cohesión y la Posición, llegando a que ese buen desempeño estaba asociado, en gran medida, a la posición de las oraciones que conformaban el resumen, pues, el cálculo individual de la Cohesión tendía a favorecer las oraciones iniciales del documento, por lo que el hecho de que al trabajar conjuntamente con la Posición incrementara los resultados obtenidos confirmó la importancia de las oraciones iniciales en el conjunto de datos evaluado.

Finalmente, de acuerdo a la comparación de los resultados obtenidos sin la utilización del algoritmo memético con los de la última configuración del algoritmo memético, se observó que el primero ofrecía mejores resultados de las medidas de ROUGE, por lo tanto, se decidió probar el algoritmo memético con el criterio de ordenamiento basado en Cohesión y Posición, con el fin de observar si su comportamiento mejoraba. En las Tablas 10 y 11, se muestran los resultados obtenidos con DUC 2002 y con DUC2001, respectivamente.

Exp.	Método	ROUGE-1			ROUGE-2			ROUGE-SU4		
		Avg-R	Avg-P	Avg-F	Avg-R	Avg-P	Avg-F	Avg-R	Avg-P	Avg-F
0	**CS_CP**	**0,48966**	**0,49359**	**0,49146**	**0,23192**	**0,23358**	**0,23267**	**0,24772**	**0,2496**	**0,24858**
1	**MA (CS_PLRCC)**	0,482839	0,487281	0,48487	0,228578	0,230624	0,229505	0,244643	0,246893	0,245664
2	**MA (CS_CP)**	0,477245	0,481728	0,479315	0,220753	0,222727	0,221660	0,238732	0,240941	0,239747

Tabla 10. Resultados de las pruebas sin y con MA con DUC 2002

Exp.	Método	ROUGE-1			ROUGE-2			ROUGE-SU4		
		Avg-R	Avg-P	Avg-F	Avg-R	Avg-P	Avg-F	Avg-R	Avg-P	Avg-F
0	**CS_CP**	**0,45729**	**0,45736**	**0,45696**	**0,20615**	**0,20626**	**0,20605**	**0,22808**	**0,22817**	**0,22795**
1	**MA (CS_PLRCC)**	0,451696	0,451807	0,451405	0,198325	0,198363	0,198205	0,222169	0,222218	0,222033
2	**MA (CS_ChP)**	0,448457	0,448701	0,448169	0,19514	0,195272	0,194905	0,216218	0,216755	0,215983

Tabla 11. Resultados de las pruebas sin y con MA con DUC 2001.

Según los resultados obtenidos, se ratifica el buen desempeño que la Posición junto al cálculo de la Cohesión ofrecen al trabajar sobre todas las oraciones del documento y no sobre una cierta cantidad. El comportamiento de estos resultados junto al tipo de estructura de los datos evaluados, sugirió que la información más relevante se encontraba en las oraciones iniciales de los documentos, por lo que este método, por su simplicidad, puede considerarse como una buena opción para documentos con dicho tipo de estructura, pero su generalización a otro tipo de documentos, diferente a noticias de DUC, no presentará siempre el mismo buen desempeño, pues la información más importante no siempre se encuentra en las primeras oraciones del documento. Por lo tanto, ya que el algoritmo memético contempla en su función objetivo otro tipo de características como: longitud, relación con el título, cobertura y cohesión, éste puede llegar a obtener un mejor desempeño al ser utilizado con otros tipos de documentos, lo cual hace que sea un algoritmo más prometedor, ya que no depende del tipo de estructura de los documentos. (Referirse al *Anexo E* para más detalle).

Capítulo 4

4 SISTEMA DE GENERACIÓN DE RESUMENES DE UN SOLO DOCUMENTO BASADA EN ALGORITMOS MEMÉTICOS

4.1 CONFIGURACIÓN FINAL DEL ALGORITMO MEMÉTICO PROPUESTO

La configuración final del algoritmo memético, obtenida al término de los ciclos de prueba, es presentada en la Figura 4.1. El funcionamiento de cada uno de los operadores comprendidos en esta configuración es descrito en detalle en el *Anexo A* y aclaraciones adicionales sobre su adaptación al problema abordado son presentadas en los *Anexos B* y *C*.

Selección Padre:	BASADA EN EL RANGO
Selección Madre:	RUEDA DE RULETA
Cruce:	CRUCE DE UN PUNTO
Mutación Externa:	MUTACIÓN MULTI-BIT
Mutación Interna:	MUTACIÓN DE INSERCIÓN
Reemplazo:	COMPETENCIA RESTRINGIDA
Optimización Local:	BÚSQUEDA LOCAL GUIADA

Figura 4.1. Configuración Final de Operadores del Algoritmo Memético

4.2 FUNCIÓN OBJETIVO

La función objetivo para el algoritmo memético propuesto fue definida como se describió en la Sección 3.4 y es fijada finalmente como la combinación lineal presentada en la Ecuación (4.1).

$$f(x) = \alpha * Posición + \beta * Longitud + \gamma * Relación\ con\ el\ título + \mu * Cohesión + \rho * Cobertura \quad (4.1)$$

Donde,

$$\alpha + \beta + \gamma + \mu + \rho = 1 \quad (4.2)$$

Como se observa, cada uno de los factores presentes en esta función objetivo va acompañado de un coeficiente que determina su influencia en la medición de la calidad de los agentes. La sumatoria de tales coeficientes, como se muestra en la Ecuación (4.2), es igual a 1.

4.3 ADAPTACIÓN DEL ALGORITMO MEMÉTICO A LA GENERACIÓN AUTOMÁTICA DE RESÚMENES DE UN SOLO DOCUMENTO

En la Figura 4.2 se presenta el esquema propuesto para el algoritmo memético, el cual se basa en el enfoque planteado por Hao [127], y en la Figura 4.3 se muestra el mismo esquema en forma gráfica. En las siguientes secciones se describe los aspectos

importantes en el proceso de acoplamiento del algoritmo memético con el problema abordado. Es importante citar que para la implementación del algoritmo memético fue necesario tener en cuenta una restricción propia de este tipo de problema, dada por la longitud del resumen final, en este caso, esa longitud debía ser igual a 100 palabras, debido a que las investigaciones del estado del arte con las que se comparó el algoritmo propuesto realizan la evaluación con datos de DUC, en los cuales los resúmenes modelo de un solo documento restringen este valor a 100 palabras.

```
INICIO MA
1. HACER
     1.1 Crear Agente Aleatorio
     1.2 Calcular aptitud del agente creado
     1.3 Optimizar el agente creado con la búsqueda local guiada
   HASTA Completar tamaño de la población
2. MIENTRAS no se supere el número permitido de evaluaciones de la función
   objetivo HACER

    2.1 Seleccionar una cantidad de agentes por elitismo, para que pasen a la
        siguiente generación
    2.2 PARA n = 0 HASTA Cantidad de agentes faltantes/2 HACER
            2.2.1   Seleccionar el padre con el método basado en el rango
            2.2.2   Seleccionar la madre con el método rueda de ruleta
            2.2.3   HACER
                      2.2.3.1    Generar hijo con el cruce de un punto
                      2.2.3.2    Mutar hijo con la mutación multi-bit y la
                        mutación de inserción
                      2.2.3.3    Optimizar hijo con la búsqueda local guiada
                    HASTA Cantidad de hijos > 2
            2.2.4   Actualizar la población por medio del reemplazo de
                    competencia restringida
        FIN PARA
    2.3 Evaluar la convergencia de la población
   FIN MIENTRAS
FIN MA
```

Figura 4.2. Pseudocódigo del algoritmo memético propuesto

Debido a que Hao [127] presenta en su trabajo una plantilla general de un algoritmo memético, en la cual no se plantea ningún método determinado para cada operador, en la adaptación propuesta, como se ve en la Figura 4.2, se especifican los métodos que fueron utilizados dentro de esta investigación. El algoritmo memético propuesto, a diferencia del algoritmo presentado por Hao [127], utiliza una optimización de tipo poblacional, con el objetivo de acelerar el hallazgo de las soluciones más prometedoras y mejorar el rendimiento del algoritmo en tiempo de ejecución. De esta manera, en la etapa de generación de la población inicial se realiza la optimización local de todos los agentes y, además, cuando un nuevo agente es generado es también optimizado. Adicionalmente, en el algoritmo memético propuesto, antes de iniciar el proceso generacional se realiza una selección elitista, con el fin de conservar parte de la calidad alcanzada hasta el momento. Por otro lado, la actualización de la población no se lleva a cabo directamente sobre la población actual, sino sobre una nueva población, con el fin de evitar que los nuevos agentes sean seleccionados como padres de la nueva generación, validando, además, que ésta no contenga agentes repetidos. Por otra parte, también se utilizó una condición de terminación diferente a la de alcanzar un máximo número de generaciones,

ya que en este caso se manejó una máxima cantidad de evaluaciones de la función objetivo. Además, como parte específica del problema abordado, por la diversidad de tamaños de los documentos, se utilizó un porcentaje de la longitud del documento original para ser evaluada, verificando que ésta no fuera inferior a 100 palabras.

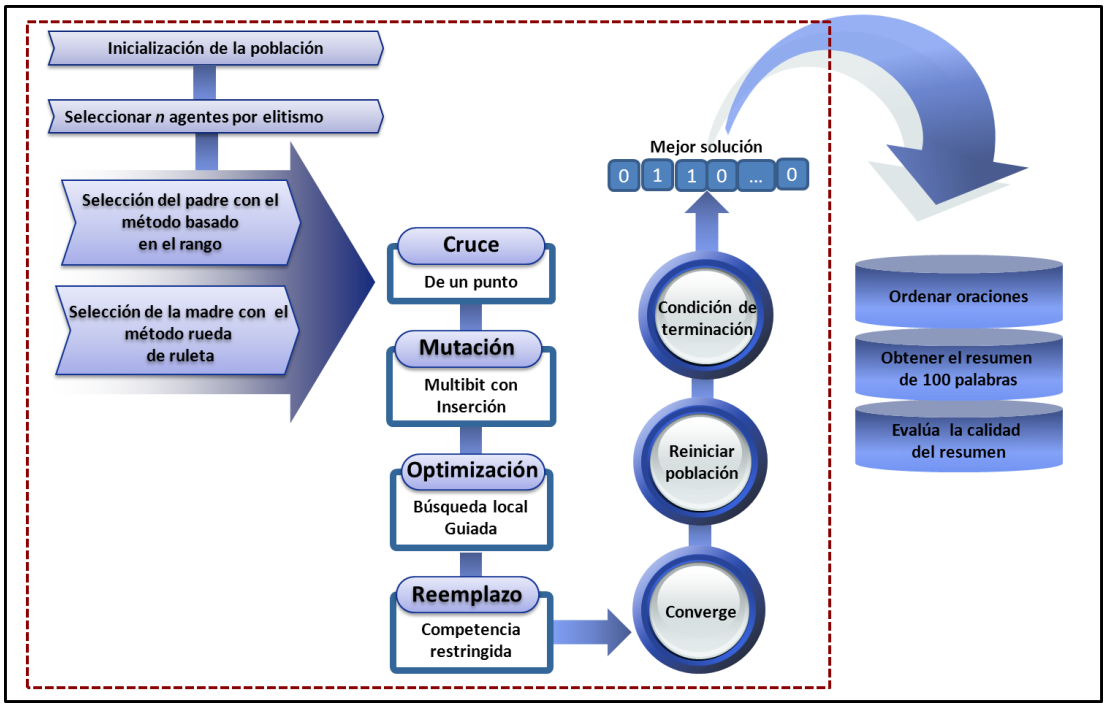

Figura 4.3. Esquema del algoritmo memético propuesto

4.3.1 Inicialización de la población

El proceso para inicializar la población, presentado en el paso 1 del esquema, está basado en una estrategia aleatoria. De este modo, un agente es creado al activar aleatoriamente algunas frases, hasta que se cumpla con las condiciones de longitud del resumen. Posteriormente, se evalúa que el nuevo agente no exista en la población actual para entonces calcular su valor de aptitud y finalmente optimizarlo por la técnica de *Búsqueda Local Guiada*. Este proceso es repetido hasta completar el tamaño de la población.

4.3.2 Condición de parada

La ejecución del algoritmo termina cuando se cumple con la condición de parada, la cual fue establecida como el número máximo de evaluaciones de la función objetivo (Paso 2 del esquema). Teniendo en cuenta que uno de los enfoques evolutivos con los que se compara esta investigación está basado en la propuesta de Shareghi y Hassanabadi [25], la cual realiza un total de 1600 evaluaciones de la función objetivo, en el presente trabajo se toma ese valor como referencia para la condición de parada.

4.3.3 Proceso generacional

El proceso de reproducción es representado por los pasos 2.1 y 2.2 de la Figura 4.2. El primer paso de este proceso consiste en seleccionar, por medio de una estrategia elitista, una cantidad determinada de agentes para que pasen sin modificación a la siguiente generación. Seguidamente, hasta completar el tamaño de la población, se escogen los padres de la nueva generación y se crean nuevos agentes. Para ello, el padre es seleccionado mediante la estrategia *Basada en el Rango*, mientras que la madre es seleccionada por medio de la técnica de *Rueda de Ruleta*. Así, el primer hijo es creado mediante el *Cruce de un Punto*. Para la creación del segundo hijo, el padre toma entonces el rol de madre y viceversa. Una vez un hijo es creado, éste es sometido al proceso de mutación de acuerdo a una probabilidad y luego es optimizado por la estrategia de *Búsqueda Local Guiada*. Finalmente, se calcula el valor de aptitud del nuevo agente y se actualiza la población con base en la estrategia de *Competencia Restringida*. En las Figuras 4.4 a 4.9 se detallan algunos ejemplos sobre los métodos anteriormente mencionados.

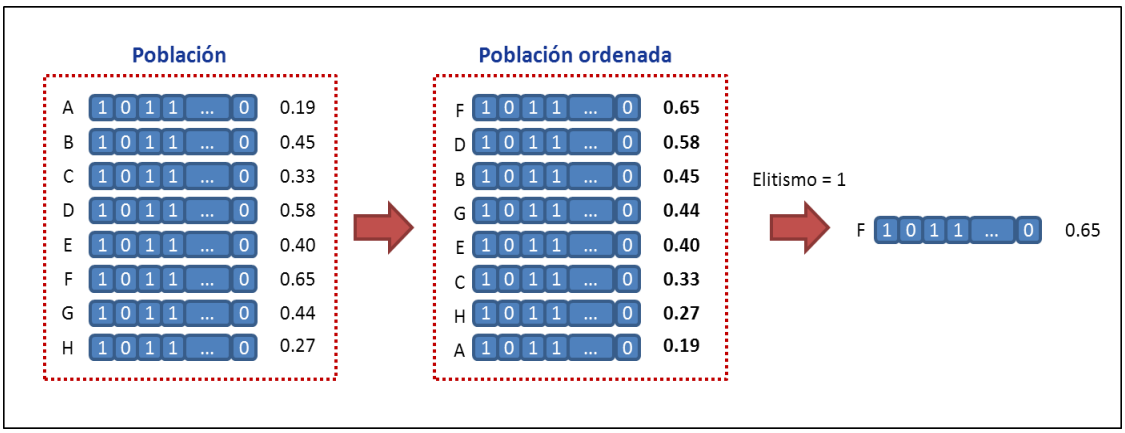

Figura 4.4. Ejemplo de la selección elitista

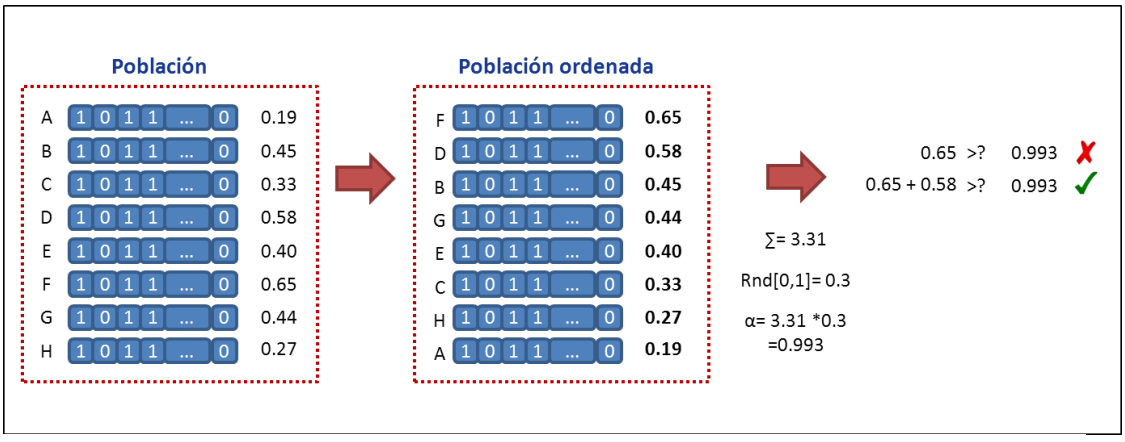

Figura 4.5. Ejemplo de la selección por rueda de ruleta

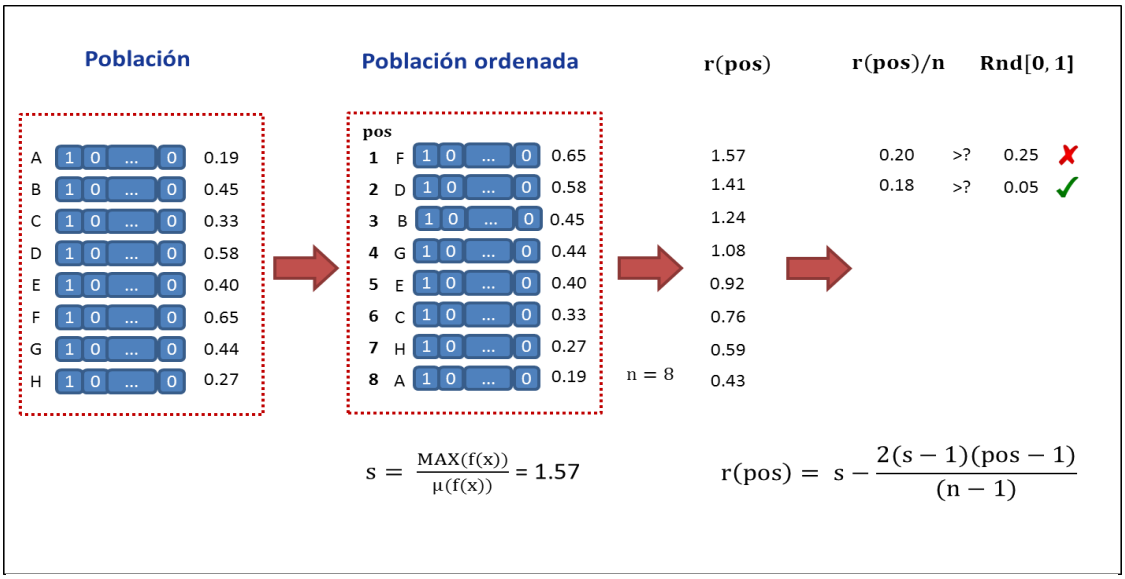

Figura 4.6. Ejemplo de la selección basada en rango

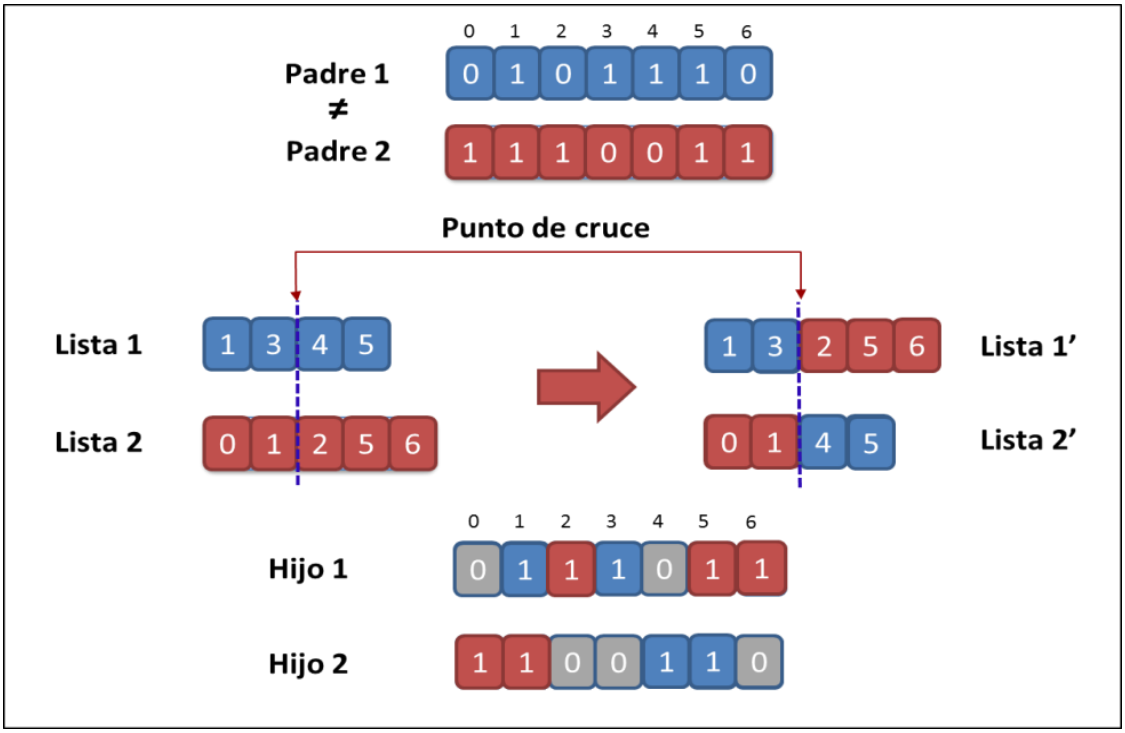

Figura 4.7. Ejemplo del cruce de un punto

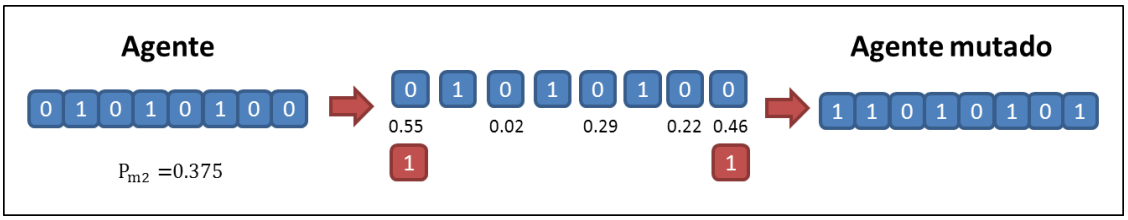

Figura 4.8. Ejemplo de la mutación multi-bit con inserción

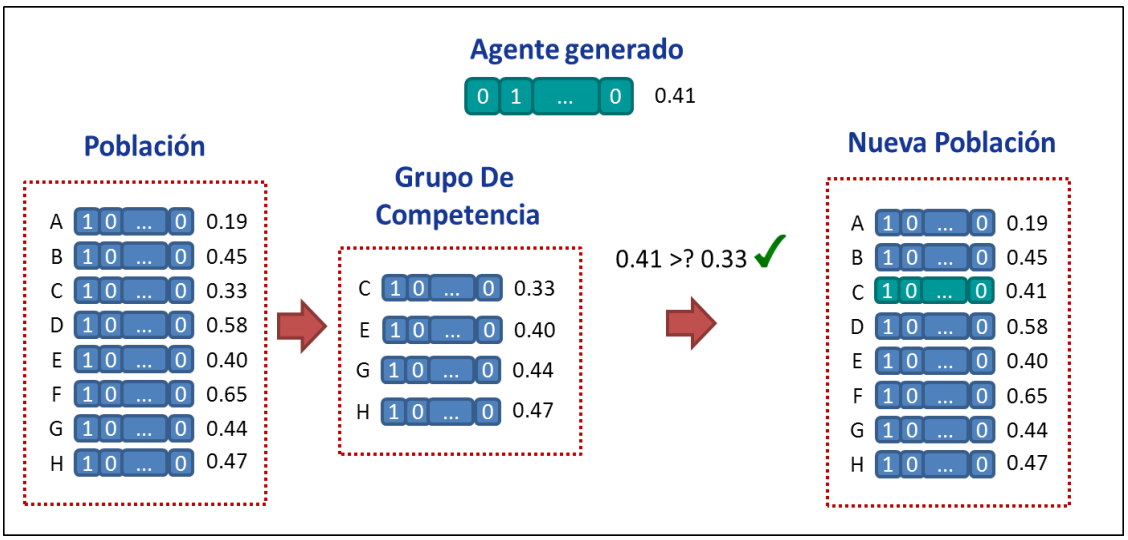

Figura 4.9. Ejemplo de reemplazo con competencia restringida

4.3.4 Evaluación de la convergencia de la población

La convergencia de la población es evaluada tras la generación de una nueva descendencia. Para determinar la tendencia de aptitudes entre los agentes de la población actual, tal esquema de evaluación determina si el 90%, o más, de ellos presentan valores de aptitud cercanos en más o menos un 5% al promedio de aptitud de toda la población. Si tras este proceso se determina que existe convergencia en la población, se procede al reinicio de la misma, el cual se lleva a cabo en forma similar al proceso de inicialización de la población, descrito en la Sección 4.3.1, conservando, además, un número predefinido de los mejores agentes de la población actual.

4.4 ESQUEMA DEL SISTEMA DE GENERACIÓN DE RESÚMENES PARA UN SOLO DOCUMENTO BASADO EN ALGORITMO MEMÉTICOS

En la Figura 4.10 se presenta el esquema general del sistema propuesto de generación automática de resúmenes basado en algoritmos meméticos. El proceso de generación de un resumen inicia con el pre-procesamiento del documento original, en el cual, se segmenta el texto para obtener las oraciones que lo conforman, se normalizan dichas oraciones eliminando mayúsculas y palabras vacías, se llevan las palabras con la misma raíz a una forma común, y finalmente las frases son indexadas en una estructura de datos. El siguiente paso del proceso consiste en tomar los términos de cada una de las

oraciones obtenidas y llevarlas a la representación del modelo espacio vectorial descrito en la Sección 2.3.1, de esta manera para cada término se calcula su peso en función de su frecuencia relativa para que sea almacenado en una matriz de pesos, así mismo se ponderan los términos correspondientes al título del documento. Seguidamente, con base en los pesos calculados, se determina la similitud de cosenos entre oraciones, la similitud de cosenos de cada oración y el documento, y la similitud de cosenos de cada oración con el título para que sean almacenados en una matriz de similitudes. Dichos valores de similitud servirán posteriormente para el cálculo de aquellos factores de la función objetivo que los requieran, como la cohesión, cobertura y relación con el título. El último paso corresponde a la ejecución del algoritmo memético como fue descrito en la sección anterior, obteniendo al final un vector solución, cuyas posiciones con valor igual a 1 son ordenadas descendentemente de acuerdo al valor obtenido a través de la Ecuación (4.3), donde x es una frase del documento. Dicha ecuación es una variación de la función objetivo que se está maximizando, que permite obtener un puntaje por oración individual. Posteriormente cada gen del agente es decodificado para obtener las oraciones originales del documento respectivas, que finalmente conforman el resumen generado, el cual es truncado a 100 palabras.

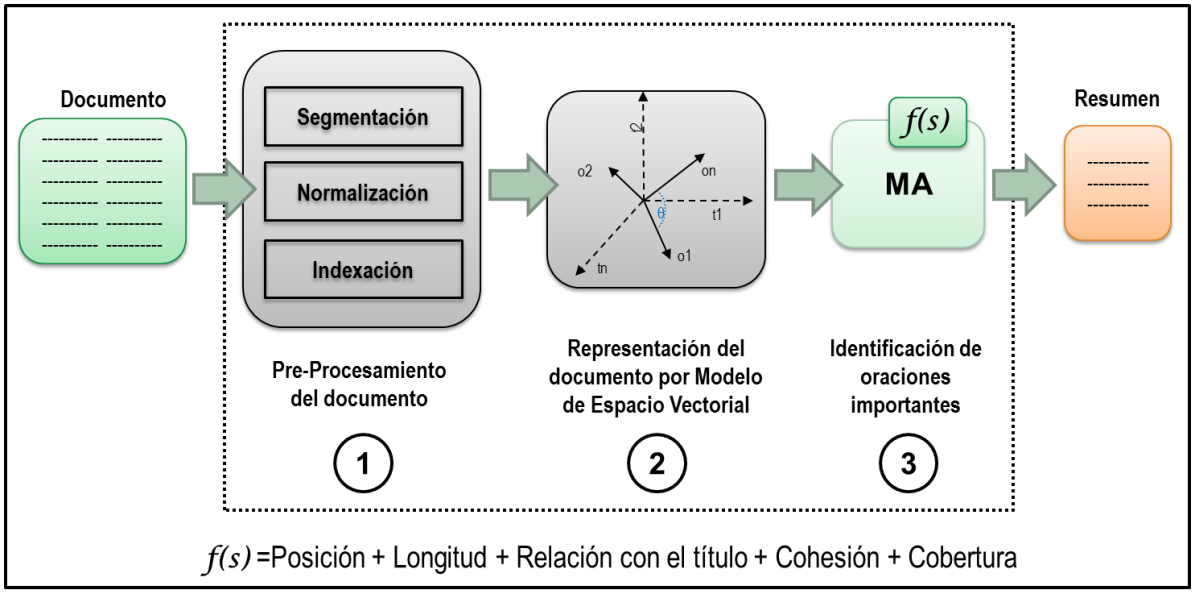

Figura 4.10. Esquema general del sistema de generación automática de resúmenes basado en Algoritmos Meméticos propuesto

$$f(x) = Posición(x) + Longitud(x) + Relación\ con\ el\ título(x) + Cohesión(x) + Cobertura(x) \quad (4.3)$$

4.5 AFINACIÓN DE PARÁMETROS

El proceso de afinación de parámetros del Algoritmo Memético fue descrito en la Sección 3.5 y detallado en el *Anexo E*. En la Tabla 12 se muestra la mejor combinación de valores obtenida para los parámetros después de realizar el proceso de afinación.

Parámetro	Valor
Tamaño de la Población	30
Número de Optimizaciones	3
Elitismo en Generación	1
Elitismo en Reiniciación	1
Tamaño del Grupo de Competencia	4
Máxima longitud a evaluar del documento	Entre 50%-70% de la longitud del documento
Probabilidad de Optimización	1
Probabilidad de Mutación	0.4

Tabla 12. Mejor combinación de valores para los parámetros del MA

Capítulo 5

5 EVALUACIÓN

La evaluación del desempeño del algoritmo propuesto se realizó teniendo en cuenta la calidad de los resúmenes obtenidos. De esta manera, el algoritmo fue ejecutado treinta veces por documento, evaluando los resúmenes generados en cada ejecución, obteniendo al final un resultado promedio de todo el conjunto de documentos (Ver Figura 5.1).

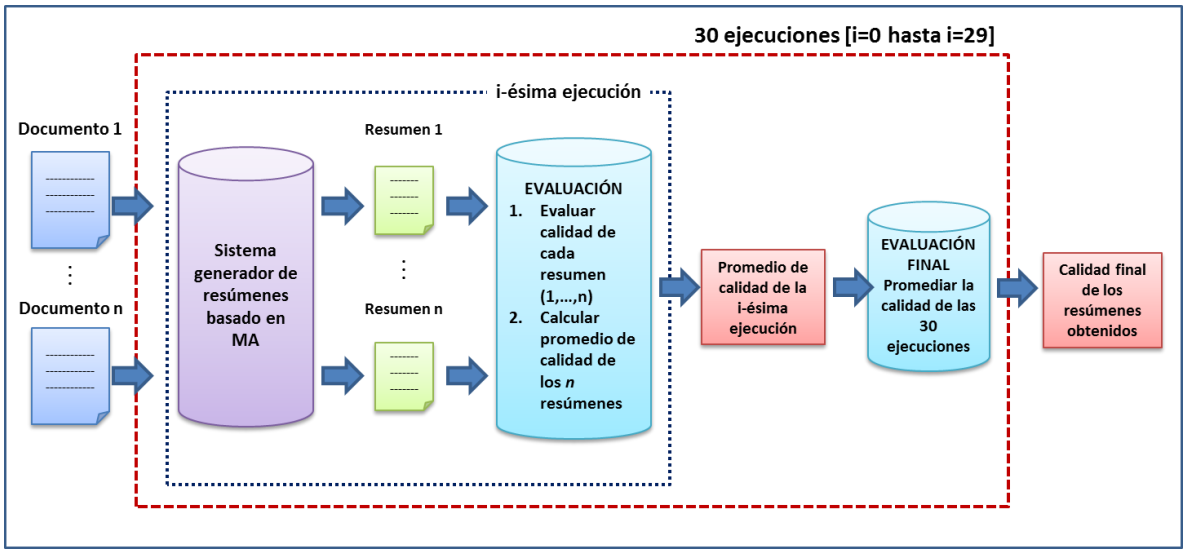

Figura 5.1. Esquema general de la evaluación de resúmenes

En las siguientes secciones de este capítulo se describen algunos aspectos de implementación del sistema propuesto, así como algunos elementos importantes que se tomaron en cuenta para realizar la evaluación del algoritmo propuesto. Finalmente, se presentan los resultados obtenidos tras el proceso de evaluación, confrontándolos, además, con otros modelos evolutivos y no evolutivos del estado del arte.

5.1 PRE-PROCESAMIENTO DE LOS DOCUMENTOS

Para realizar la normalización e indexación de los documentos, en la presente investigación se utilizó una herramienta de segmentación de fuente abierta denominada "*splitta*"[20], la cual utiliza un enfoque estadístico que intenta lidiar con los problemas de ambigüedad en la detección de los límites de las oraciones de un texto, y cuyo desempeño ha sido reportado en un estándar de *Wall Street Journal*, con buenos resultados [128]. Además, se utilizó una lista de palabras vacías construida para el

[20] Esta herramienta se encuentra disponible en *http://code.google.com/p/splitta*.

sistema de recuperación de información *SMART*[21] [129]. Para la lematización el algoritmo utilizado fue el de Porter[22].

5.2 LUCENE

Lucene es una librería de código abierto bajo la licencia Apache Software Licence, cuyo objetivo es facilitar la indexación y búsqueda en tareas de recuperación de información. Fue implementada originalmente en Java, pero en la actualidad ha sido adaptada a otros lenguajes de programación como C#, C++, Delphi, PHP, Phyton y Ruby [130]. Una de las características principales de esta herramienta, es la abstracción de los documentos como un conjunto de campos de texto, muy útil para el acoplamiento con sistemas basados en el Modelo de Espacio Vectorial para la representación de los documentos. En el presente trabajo, la librería de Lucene[23]es utilizada para la indexación de términos, a la vez que contribuye a las tareas de eliminación de mayúsculas y signos ortográficos, eliminación de palabras vacías y lematización.

5.3 COLECCIÓN DE DOCUMENTOS DE EVALUACIÓN

Para la evaluación del sistema propuesto en este trabajo, se utilizaron los conjuntos de datos de DUC2001 y DUC2002, debido a que las investigaciones de referencia de generación de resúmenes automáticos de un solo documento utilizaron estos conjuntos. Ambos conjuntos son producto de las investigaciones del Instituto Nacional de Estándares y Tecnología[24] en el área de generación de resúmenes automáticos y están constituidos por noticias periodísticas en inglés, tomadas de distintos periódicos y agencias de noticias como Financial Times, Associated Press o Wall Street Journal [131]. La colección de datos de DUC2002 consiste en 59 conjuntos de aproximadamente 10 documentos de noticias periodísticas en ingles cada uno, completando un total de 567 documentos, los cuales abarcan temáticas como acontecimientos de desastres naturales, información biográfica sobre un individuo, entre otros. Cada conjunto está acompañado por resúmenes de referencia para uno y múltiples documentos. Los resúmenes de referencia para un sólo documento, están conformados por alrededor de 100 palabras (Ver Tabla 13). Por su parte, la colección de DUC2001 consta de 309 documentos divididos en 30 conjuntos. Al igual que DUC2002, cada conjunto cuenta con resúmenes de referencia para uno y múltiples documentos, con una longitud cercana a 100 palabras (Ver Tabla 13).

	DUC 2002	DUC 2001
Número de conjuntos	59	30
Número de documentos	567	309
Fuente de datos	TREC[25]	TREC
Longitud del resumen	100 palabras	100 palabras

Tabla 13. Resumen de los conjuntos de datos utilizados

[21] Esta lista se encuentra disponible en *ftp://ftp.cs.cornell.edu/pub/smart/english.stop*.
[22] Disponible en http://tartarus.org/martin/PorterStemmer/
[23] Esta librería se encuentra disponible en http://lucenenet.apache.org/
[24] National Insitute Standars and Technology por sus siglas en inglés NIST. Página oficial: *http://www-nlpir.nist.gov/*
[25] http://trec.nist.gov/overview.html

5.4 MÉTRICAS DE EVALUACIÓN

La calidad de los resúmenes generados a través del sistema propuesto en este trabajo, fue estimada por medio de las métricas proporcionadas por la herramienta de evaluación ROUGE (Ver Sección 2.1.4.3) en su versión 1.5.5, la cual ha sido manejada ampliamente por DUC en la evaluación de resúmenes automáticos. Las métricas utilizadas en esta investigación fueron ROUGE-1, ROUGE-2 y ROUGE-SU4, por ser las más utilizadas en varias de las propuestas evolutivas del estado del arte.

5.5 RESULTADOS Y ANÁLISIS

Los resultados presentados en esta sección fueron obtenidos evaluando resúmenes generados de 100 palabras, y, como fue citado previamente, promediando 30 ejecuciones del algoritmo, el cual fue desarrollado en la plataforma .NET en lenguaje c# sobre Pentium 4 CPU 3.00GHz, 2.99GHz con 1GB de RAM en Windows XP.

En la Tabla 14 se presentan los resultados obtenidos con los conjuntos de documentos de DUC 2001 y DUC 2002.

MÉTODO	ROUGE-1	ROUGE-2
DUC 2001	0,44947	0,20250
DUC 2002	0,48284	0,22858

Tabla 14. Resultados finales del MA con DUC 2001 y DUC 2002

5.5.1 MA propuesto con respecto a otros sistemas

Los resultados obtenidos con el MA propuesto se compararon con otros métodos del estado del arte en generación de resúmenes automáticos de un solo documento. A continuación se presenta una descripción breve de cada uno de ellos.

- **DE**: Este modelo evolutivo utiliza la Evolución Diferencial para optimizar la asignación de oraciones a grupos. La selección de las oraciones del resumen se realiza bajo un esquema recursivo, que tiene en cuenta el grado de pertenencia de cada oración al grupo correspondiente [27].
- **FEOM**: Este trabajo aplica un modelo de optimización evolutiva difusa para optimizar el agrupamiento de oraciones. Las oraciones más importantes de cada grupo se seleccionan para formar el resumen final [44].
- **UnifiedRank**: Enfoque basado en grafos, en el cual la generación automática del resumen de un solo documento se realiza al mismo tiempo que la generación del resumen de múltiples documentos, mediante cálculos iterativos y recursivos [36].
- **NetSum**: Es un enfoque basado en redes neuronales que utiliza el algoritmo de aprendizaje RankNet, el cual entrena un clasificador de oraciones basado en parejas, para determinar las oraciones más importantes de un documento [8].
- **CRF**: Este trabajo aborda la generación de resúmenes automáticos como un problema de etiquetado de secuencias. En ese sentido, las oraciones se etiquetan como 1ó 0 utilizando el modelo de distribución de Campo Aleatorio Condicional (CRF) [33].

- **QCS**: Los resúmenes son generados utilizando el modelo oculto de Markov (HMM) para calcular la probabilidad de que una oración pertenezca a un buen resumen. Las oraciones con mayor probabilidad forman el resumen final [34].
- **SVM**: Este trabajo propone dos métodos, el enfoque basado en corpus modificado (MCBA) y un enfoque T.R.M. basado en LSA. El primero está basado en una función de puntuación combinada con el análisis de características sobresalientes. El segundo explota el Análisis Semántico Latente (LSA) y un mapa de relaciones de texto (TRM) para determinar las estructuras semánticamente importantes de un documento [37].
- **Manifold Ranking**: El proceso de Manifold-Ranking utiliza las relaciones entre oraciones y de la relación entre una oración y un tópico determinado, para denotar la riqueza de información de una oración. El resumen se forma al escoger las oraciones con información rica y nueva [40].

En la Tabla 15 se presenta los resultados obtenidos en las medidas de ROUGE por el MA propuesto y los otros métodos del estado arte, para el conjunto de datos de DUC2001. En la Tabla 16 se presenta la información para DUC2002.

DUC 2001		
MÉTODO	**ROUGE-1**	**ROUGE-2**
MA	0,44947 (6)	**0,20250** (1)
DE	**0,47856** (1)	0,18528 (3)
FEOM	0,47728 (2)	0,18549 (2)
UnifiedRank	0,45377 (5)	0,17646 (6)
NetSum	0,46427 (3)	0,17697 (5)
CRF	0,45512 (4)	0,17327 (7)
QSC	0,44852 (7)	0,18523 (4)
SVM	0,44628 (8)	0,17018 (8)
Manifold Ranking	0,43359 (9)	0,16635 (9)

Tabla 15. Resultado de los métodos con DUC 2001

DUC 2002		
MÉTODO	**ROUGE-1**	**ROUGE-2**
MA	0,48284 (2)	**0,22858** (1)
DE	0,46694 (3)	0,12368 (5)
FEOM	0,46575 (4)	0,12490 (4)
UnifiedRank	**0,48478** (1)	0,21462 (2)
NetSum	0,44963 (5)	0,11167 (6)
CRF	0,44006 (7)	0,10924 (7)
QSC	0,44865 (6)	0,18766 (3)
SVM	0,43235 (9)	0,10867 (8)
Manifold Ranking	0,42325 (8)	0,10677 (9)

Tabla 16. Resultado de los métodos con DUC 2002

De acuerdo a los datos presentados en las Tablas 15 y 16, se puede observar que el MA supera a todos los demás métodos en la medida de ROUGE-2, tanto para DUC2001 como DUC2002. En la medida ROUGE1 para DUC2002, el MA solo es superado por UnifiedRank, mientras que en el caso de DUC2001, es superado por cinco métodos.

En la Tabla 17 se muestra la mejora en la medida ROUGE-2 sobre los datos DUC2001 y DUC2002, del MA con respecto a los otros métodos, calculada por medio de la Ecuación (5.1). Como se observa con DUC2001, el método superado con el valor más bajo es FEOM con un 8.59% y el valor más alto es para Manifold Ranking con un 21.8%. Por otro lado, ED es superado por 8.71% y UnifiedRank por 14.14%. En el caso de DUC2002, el MA supera a UnifiedRank con el valor más bajo de 6,50% y el valor más alto es, nuevamente, para Manifold Ranking con un 114.09%. Por su parte los *modelos evolutivos* ED y FEOM, son superados por porcentajes bastante altos de 84.82% y 83.01%, respectivamente.

$$\frac{\text{Método Propuesto} - \text{Otro Método}}{\text{Otro método}} \times 100 \quad (5.1)$$

Método	Mejoramiento del MA (%)	
	DUC2001	DUC2002
DE	9,29	84,82
FEOM	9,17	83,01
UnifiedRank	14,76	6,50
NetSum	14,43	104,69
CRF	16,87	109,25
QSC	9,32	21,81
SVM	18,99	110,34
Manifold Ranking	21,73	114,09

Tabla 17. Mejoramiento del MA con respecto a otros métodos con ROUGE-2

En la Tabla 18 se muestra la mejora del MA en la medida de ROUGE-1 sobre los otros métodos, de acuerdo a la Ecuación (5.1). Como se observa, para los datos de DUC 2001, el MA es superado por cinco métodos, sin embargo, lo es por porcentajes bajos. De esta manera, el método que supera al MA con el porcentaje más alto es ED, con 6.08%. Por su parte, el método superado por el MA con el porcentaje más alto es Manifold Ranking con 3.66%. Para los datos de DUC 2002 el MA es superado sólo por UnifiedRank en un 0.40%, mientras que sobrepasa, nuevamente, a Manifold Ranking con el porcentaje más alto de 14.08%.

Método	Mejoramiento del MA (%)	
	DUC2001	DUC2002
ED	-6,08	3,41
FEOM	-5,83	3,67
UnifiedRank	-0,95	-0,40
NetSum	-3,19	7,39
CRF	-1,24	9,72
QSC	0,21	7,62
SVM	0,71	11,68
Manifold Ranking	3,66	14,08

Tabla 18. Mejoramiento del MA con respecto a otros métodos con ROUGE-1

Considerando que el MA no obtiene los mejores resultados en la medida de ROUGE-1, para DUC2001 y DUC2002, se plantea un ranking de todos los métodos, teniendo en cuenta el puesto que ocupa el método en cada medida. Para obtener los rangos resultantes de los métodos, se transforman las Tablas 15 y 16 en la
Tabla **19**. El rango resultante de esta tabla (presentado en la última columna) es calculado de acuerdo a la Ecuación (5.2).

$$Rango(método) = \sum_{r=1}^{9} \frac{(9 - r + 1)\, R_r}{9} \quad (5.2)$$

Donde R_r denota el número de veces que el método respectivo ocupa el r-ésimo lugar.

MÉTODO	1	2	3	4	5	6	7	8	9	Rango resultante
	$R_r =$									
MA	2	1	0	0	0	1	0	0	0	**3.33**
ED	1	0	2	0	1	0	0	0	0	**3.11**
FEOM	0	2	0	2	0	0	0	0	0	**3.11**
UnifiedRank	1	1	0	0	1	1	0	0	0	**2.89**
NetSum	0	0	1	0	2	1	0	0	0	**2.33**
QSC	0	0	1	1	0	1	1	0	0	**2.22**
CRF	0	0	0	1	0	0	3	0	0	**1.67**
SVM	0	0	0	0	0	0	0	3	1	**0.78**
Manifold Ranking	0	0	0	0	0	0	0	1	3	**0.56**

Tabla 19. Rango resultante de los métodos

Según los resultados de la
Tabla **19**, se observa que el MA ocupa el primer lugar en el ranking, superando métodos como ED y UnifiedRank, a pesar de que en la medida de ROUGE-1, estos métodos obtenían los mejores valores, aunque con una complejidad computacional mayor a la del MA, pues cabe resaltar que, para la generación del resumen extractivo, ED lleva a cabo una estrategia que mide la centralidad de cada oración con respecto al grupo que

pertenece, con base en distancia google normalizada y recursividad. El inconveniente más importante de esto último, es el costo computacional que supone realizar una serie de llamadas recursivas [132], sumado al costo de la previa ejecución del algoritmo de evolución diferencial. Por su parte, UnifiedRank involucra costos de recursividad a través del cálculo de la importancia tanto global como local de cada oración propuestos, además de la complejidad de los grafos que, medida en términos de su matriz de adyacencia, corresponde a $O(\frac{m*\log(v^2/m)}{\log(v)})$, donde v es el número de vértices del grafo y m es el número de aristas [133], mientras que el algoritmo memético, para optimizar la función objetivo tiene un costo de $O(\frac{\mu}{\rho}\log(\mu))$, donde μ es el tamaño de la población y ρ es la cantidad de descendientes producidos en una generación [125], que en este caso sería igual a $O(\frac{\mu}{(\mu-e)}\log(\mu))$, siendo e la cantidad de agentes seleccionados por elitismo para pasar a la siguiente generación, la cual es igual a 1 en la configuración final del MA, por lo que finalmente es igual a $O(\frac{\mu}{(\mu-1)}\log(\mu))$. En la Figura 5.2 se presenta gráficamente el comportamiento de las complejidades del esquema basado en grafos y basado en MA (para una mejor discriminación, se muestra el comportamiento para un grafo con 5 aristas, aunque cabe mencionar que la cantidad de aristas depende de la cantidad de relaciones de similitud entre oraciones manejada por el grafo).

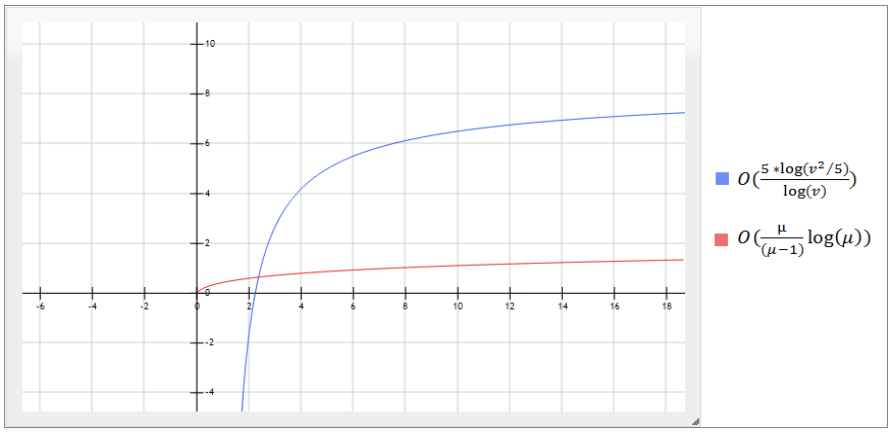

Figura 5.2. Complejidades de los esquemas basados en grafos y en MA

Por otro lado, los resultados de la
Tabla **19** muestran también que el rango de los métodos de ED y FEOM es el mismo, los cuales al igual que el MA abordan la generación automática de resúmenes como un problema de optimización, pero ED y FEOM utilizan el concepto de clustering en la representación de la solución.

Finalmente, es importante anotar que el MA presentó los mejores resultados en la medida de ROUGE-2, lo cual resalta nuevamente el buen desempeño del MA frente a los métodos del estado del arte, ya que es una métrica un poco más rigurosa que ROUGE-1, al evaluar bi-gramas coincidentes entre el resumen generado y el resumen de referencia, en lugar de uni-gramas.

5.5.2 Evaluaciones adicionales

5.5.2.1 Evaluación del comportamiento de la función objetivo final con un enfoque basado en Búsqueda Armónica

Uno de los trabajos del estado del arte que aplica un enfoque evolutivo basado en HS para la generación de resúmenes de un solo documento es el propuesto por Shareghi y Hassanabadi [25]. Este enfoque fue considerado, en un inicio, como un esquema de referencia para la presente investigación, sin embargo, los resultados expuestos en dicho trabajo no corresponden a medidas de ROUGE sino a valores promedio de precisión y recuerdo, por lo que la comparación con dicho trabajo no era posible en forma directa. Por esta razón, y aprovechando que se contaba con el código fuente de un sistema de generación automática de resúmenes de múltiples documentos basado en HS presentado en [134], se decidió adaptar dicho esquema a la generación de resúmenes de un solo documento y tomar la configuración de parámetros presentada por Shareghi y Hassanabadi. Teniendo en cuenta que el sistema HS mencionado fue una adaptación realizada dentro de la presente investigación, sus resultados no se incluyeron como comparaciones principales con algoritmos evolutivos en la sección anterior, por cuestiones de imparcialidad en los resultados comparados. No obstante, en el transcurso de esta investigación dicho enfoque fue tomado como base de referencia para comparar el desempeño que se obtenía con cada nuevo ajuste del algoritmo memético propuesto y además se utilizó para hacer algunas pruebas adicionales con el propósito de evaluar el desempeño de la función objetivo obtenida en esta investigación. En ese sentido, tanto el algoritmo memético como el HS fueron evaluados con tres funciones objetivo diferentes, donde la primera corresponde a la presentada en la investigación de Shareghi y Hassanabadi, mientras que las otras dos corresponden a la función objetivo propuesta en esta investigación, evaluada con y sin optimización de pesos (Ver Tabla 20).

Función Objetivo	Descripción	Configuración
S&H	Función objetivo propuesta por Sharegui y Hassanabadi [25], sin optimización de pesos.	$f(s) = 0.33 * Relación\ con\ el\ título + 0.33 * Cohesión + 0.33 * Legibilidad$
PLRCC_P	Función objetivo propuesta con optimización de pesos.	$f(s) = 0.35 * Posición + 0.29 * Longitud + 0.005 * Relación\ con\ el\ título + 0.005 * Cohesión + 0.35 * Cobertura$
PLRCC	Función objetivo propuesta sin optimización de pesos.	$f(s) = 0.2 * Posición + 0.2 * Longitud + 0.2 * Relación\ con\ el\ título + 0.2 * Cohesión + 0.2 * Cobertura$

Tabla 20. Funciones objetivo evaluadas en pruebas adicionales

- ***Evaluación con DUC 2002***

En la Tabla 21 se muestran los resultados de la evaluación de la función objetivo con MA y HS con los datos de DUC 2002. En la Figura 5.3, se comparan gráficamente estos resultados.

DUC 2001				
FO	**Método**	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
S&H	MA	**0,47365**	**0,21540**	**0,23396**
	HS	0,46860	0,20937	0,22889
PLRCC	MA	**0,48007**	**0,22514**	**0,24191**
	HS	0,47610	0,22056	0,23818
PLRCC_P	MA	**0,48284**	**0,22858**	**0,24464**
	HS	0,47617	0,22136	0,23905

Tabla 21. Resultados primera evaluación adicional de la función objetivo con DUC 2002

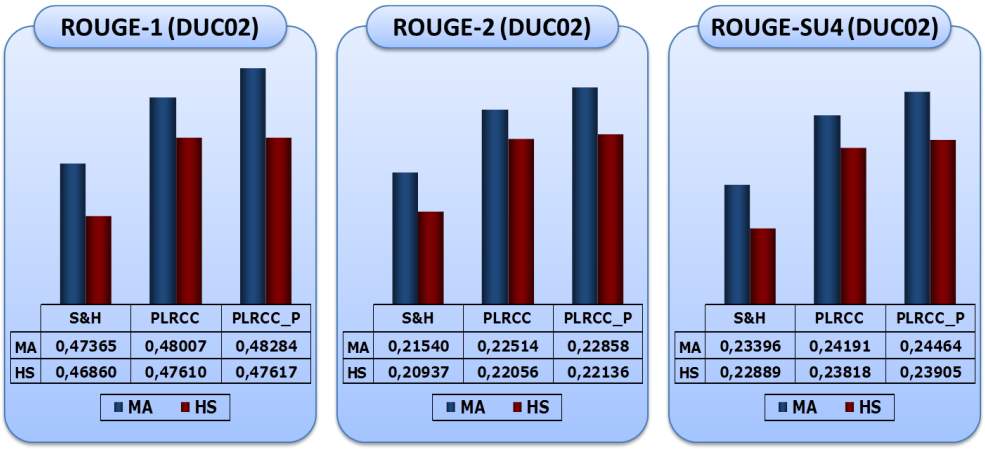

Figura 5.3. Resultados evaluación adicional de la función objetivo con DUC 2002

De acuerdo a los resultados presentados, se observa que al evaluar HS con las funciones objetivo *PLRCC_P* y *PLRCC* se obtiene un mejor desempeño que con *S&H*. En la Tabla 22 se muestra el mejoramiento relativo en cada medida de ROUGE basado en la Ecuación (5.1), donde el signo (+) indica un mejoramiento en el rendimiento del HS. Este comportamiento indica que las configuraciones propuestas de las funciones objetivo *PLRCC_P* y *PLRCC* permiten una mejor discriminación de las oraciones importantes de un documento.

FO	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
PLRCC	1,60 (+)	5,34 (+)	4,06 (+)
PLRCC_P	1,62 (+)	5,73 (+)	4,44 (+)

Tabla 22. Mejoramiento de las funciones objetivo *PLRCC* y *PLRCC_P* con respecto a *S&H* en HS con DUC 2002

Por otro lado, se observa, además, que, aunque el HS no logra superar al MA, el acoplamiento con las funciones objetivo *PLRCC y PLRCC_P* lo llevan a mejorar su rendimiento. En la Tabla 23, se muestra el porcentaje de mejoramiento, con base en la Ecuación (5.1), del MA sobre el HS, con cada una de las funciones objetivo evaluadas. El

objetivo principal de esta comparación es evidenciar un poco más la mejora que conlleva la aplicación de las funciones objetivo *PLRCC y PLRCC_P* sobre un esquema.

FO	ROUGE-1	ROUGE-2	ROUGE-SU4
S&H	1,08 (+)	2,88 (+)	2,22 (+)
PLRCC	0,83 (+)	2,08 (+)	1,57 (+)
PLRCC_P	1,40 (+)	3,26 (+)	2,34 (+)

Tabla 23. Rendimiento de cada función objetivo entre el MA y HS con DUC 2002

Como se observa, el porcentaje de diferencia entre MA y HS con la función objetivo *PLRCC* es menor que con la *S&H*, esto confirma que dicha función objetivo mejora el rendimiento del HS, haciendo sus resultados más cercanos a los del MA. Por otro lado, de acuerdo al porcentaje de *PLRCC_P*, y teniendo en cuenta los resultados presentados en la Tabla 22, donde se muestra que la función objetivo *PLRCC_P* mejora el desempeño del HS, es evidente que, a pesar de ello, el MA tiene un mejor comportamiento que HS.

- ***Evaluación con DUC 2001***

En la Tabla 24 se muestran los resultados de la evaluación de la función objetivo con MA y HS con los datos de DUC 2001. En las Figura 5.4 se comparan gráficamente estos resultados.

DUC 2001				
FO	**Método**	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
S&H	MA	**0,44493**	**0,19320**	**0,21644**
	HS	0,43929	0,18650	0,21058
PLRCC	MA	**0,44581**	**0,19851**	**0,22092**
	HS	0,44364	0,19487	0,21761
PLRCC_P	MA	**0,44947**	**0,20250**	**0,22430**
	HS	0,44705	0,19906	0,22168

Tabla 24. Resultados primera evaluación adicional de la función objetivo con DUC 2001

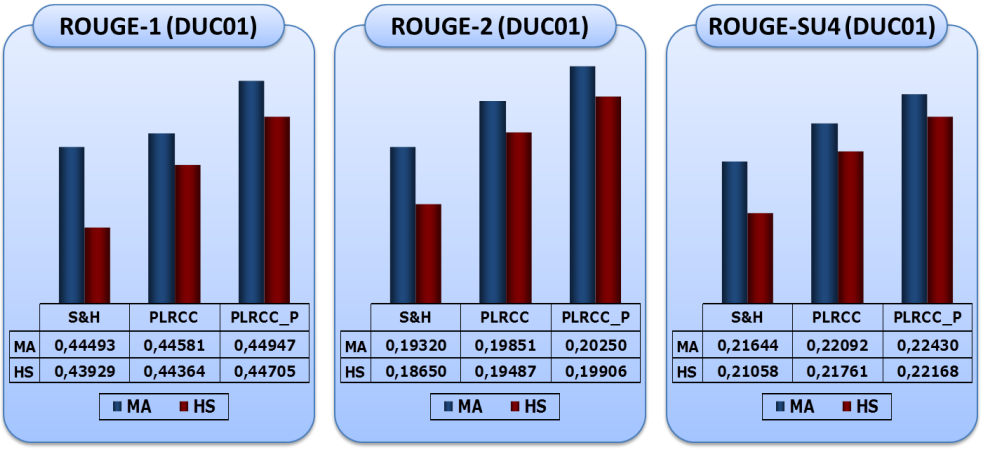

Figura 5.4. Resultados evaluación adicional de la función objetivo con DUC 2001

De acuerdo a los resultados presentados, se observa un comportamiento similar que con los datos de DUC 2002. De esta manera, las funciones objetivo *PLRCC_P* y *PLRCC* presentan un mejor desempeño que *S&H*, como se muestra en la Tabla 25, en la cual el signo (+) implica un mejoramiento calculado como en la Ecuación (5.1).

FO	ROUGE-1	ROUGE-2	ROUGE-SU4
PLRCC	0,99 (+)	4,49 (+)	3,34 (+)
PLRCC_P	1,77 (+)	6,73 (+)	5,27 (+)

Tabla 25. Mejoramiento de las funciones objetivo PLRCC y PLRCC_P con respecto a S&H en HS con DUC 2001

De otra parte, de acuerdo a la Tabla 26, nuevamente se observa que la función objetivo *PLRCC* mejora el rendimiento del HS, haciendo sus resultados más comparables a los del MA y, para este conjunto de datos, se advierte el mismo comportamiento con la función objetivo *PLRCC_P*, ratificando la influencia de la función objetivo propuesta sobre el rendimiento del HS. A pesar de estos aspectos, se observa que el enfoque basado en MA sigue superando el desempeño del HS.

FO	ROUGE-1	ROUGE-2	ROUGE-SU4
S&H	1,28 (+)	3,59 (+)	2,78 (+)
PLRCC	0,49 (+)	1,87 (+)	1,52 (+)
PLRCC_P	0,54 (+)	1,73 (+)	1,18 (+)

Tabla 26. Rendimiento de cada función objetivo entre el MA y el HS con DUC 2001

5.5.2.2 Evaluación adicional de criterios de selección de oraciones

Durante el transcurso de la investigación se observó un comportamiento particular del algoritmo memético al variar la máxima cantidad de oraciones a evaluar de los documentos, lo cual llevó a definir algunas pruebas adicionales. A pesar de no estar dentro del alcance planteado en esta investigación, tales pruebas se llevaron a cabo con el propósito de analizar el comportamiento de las características de la función objetivo diseñada sin la intervención del algoritmo memético. De las pruebas realizadas la más sobresaliente, como se citó en la Sección 3.7, es aquella en la que se evaluó la selección de las oraciones del resumen con base en un criterio de posición y cohesión, y al que se hará referencia en esta sección como *CriterioCP*.

En las Tablas 34 y 35 se muestran dichos resultados, con DUC 2002 y DUC 2001 respectivamente, comparados con los obtenidos al seleccionar las oraciones del resumen de acuerdo a un criterio basado en las cinco características de la función objetivo diseñada inicialmente para el MA: posición, longitud, relación con el título, cobertura y cohesión (*CriterioPLRCC*). Adicionalmente, se incluyen los resultados de la línea base, la cual consiste en formar el resumen con las 100 primeras palabras del documento respectivo [9].

DUC 2002			
Criterio de selección	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
Línea Base	0,47395	0,22243	0,23922
CriterioPLRCC	0,48246	0,22920	0,24512
CriterioCP	**0,48966**	**0,23192**	**0,24772**

Tabla 27. Resultados sin Algoritmo Memético con DUC 2002

DUC 2001			
Criterio de selección	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
Línea Base	0,44066	0,19679	0,21944
CriterioPLRCC	0,44767	0,20106	0,22333
CriterioCP	**0,45388**	**0,20601**	**0,22673**

Tabla 28. Resultados sin Algoritmo Memético con DUC 2001

En la Tabla 29 se presenta el mejoramiento relativo con DUC 2002 y DUC 2001.

Conjunto de datos	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
DUC 2002	1,49 (+)	1,19 (+)	1,06 (+)
DUC 2001	1,39 (+)	2,46 (+)	1,52 (+)

Tabla 29. Mejoramiento del criterio de selección por posición y cohesión

Este comportamiento, sumado a los resultados relativamente buenos presentados por la *línea base*, sugieren que para los conjuntos de documentos evaluados, las primeras oraciones de cada documento abarcan gran parte de la información relevante contenida en los mismos, lo que lleva a presumir que los buenos resultados presentados por la posición y la cohesión pueden estar muy alineados con el conjunto de datos. De hecho, en un estudio realizado por Nenkova, partiendo de la ejecución del método de *Línea base*, se reportó que para los conjuntos de datos de DUC2001 a DUC2004 la información de la posición era muy efectiva en la generación de resúmenes genéricos [135, 136]. Sin embargo, como el algoritmo memético contempla en su función objetivo otro tipo de características como: longitud, relación con el título, cobertura y cohesión, éste puede llegar a obtener un mejor desempeño para documentos con una estructura diferente a la de noticias de DUC, lo cual permite que sea un algoritmo que no depende de este tipo de estructura. Por otro lado, como se citó en la Sección 3.7, el criterio de selección de oraciones basado en Posición y Cohesión (*CriterioCP*), presentó resultados más altos que el MA para los conjuntos de datos evaluados. Así mismo, dicho método logró superar al mejor método no evolutivo cuya comparación con el MA se presentó en la Sección **¡Error! No se encuentra el origen de la referencia.** (*UnifiedRank*). En la Tabla 30 se recopilan los resultados para los tres métodos.

Método	**DUC 2001**		**DUC 2002**	
	ROUGE-1	**ROUGE-2**	**ROUGE-1**	**ROUGE-2**
CriterioCP	**0,45388**	**0,20601**	**0,48966**	**0,23192**

UnifiedRank	0,45377	0,17649	0,48478	0,21462
MA	0,44947	0,20250	0,48284	0,22858

Tabla 30. Comparación entre CriterioCP, UnifiedRank y MA

De acuerdo a estos datos, el CriterioCP, con el conjunto de DUC2001, supera a MA en un 0,98% en ROUGE-1 y en un 1,73% en ROUGE-2, mientras que a UnifiedRank lo supera en un 0,02% en ROUGE-1 y en un 16,73% en ROUGE-2. Con el conjunto de DUC2002, el CriterioCP sobrepasa a MA en un 1,41% para ROUGE-1 y en un 1,46% en ROUGE-2, mientras que a UnifiedRank los sobrepasa en un 1,01% en ROUGE-1 y en un 8,06 % en ROUGE-2.

5.5.2.3 Evaluación del mejoramiento obtenido con la aplicación de optimización local

Con el propósito de analizar el mejoramiento de resultados obtenido con la incorporación de optimización local sobre el proceso evolutivo, en la Tablas 38 y 39 se presentan los resultados finales obtenidos con el MA comparados con el mismo algoritmo, pero sin la aplicación de estrategias de optimización local, con los conjuntos DUC2002 y DUC2001, respectivamente.

DUC 2002			
Método	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
MA con optimización local	**0,48284**	**0,22858**	**0,24464**
MA sin optimización local	0,47772	0,22249	0,23967

Tabla 31. Resultados de la evaluación del mejoramiento de la aplicación de optimización local con DUC2002

DUC 2001			
Método	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
MA con optimización local	**0,44947**	**0,20250**	**0,22430**
MA sin optimización local	0,44435	0,19641	0,21933

Tabla 32. Resultados de la evaluación del mejoramiento de la aplicación de optimización local con DUC2001

En la Tabla 33 se muestra el mejoramiento relativo, con DUC 2002 y DUC2001, respectivamente, calculado como en la Ecuación (5.1), de la aplicación de optimización local sobre el proceso evolutivo, indicado con un (+) un mejor desempeño con el método que aplica optimización local sobre el que no lo hace.

Conjunto de datos	ROUGE-1	ROUGE-2	ROUGE-SU4
DUC 2002	1,07 (+)	2,74 (+)	2,07 (+)
DUC 2001	1,15 (+)	3,10 (+)	2,27 (+)

Tabla 33. Mejoramiento relativo al aplicar optimización local

Conforme a estos resultados, se aprecia que la aplicación de optimización local sobre el proceso evolutivo conduce a mejores resultados.

5.5.3 Discusión final

Los resultados presentados en esta sección destacaron al MA como un método muy competitivo frente a los métodos del estado del arte, pues presentó los mejores resultados de ROUGE-2 tanto para DUC 2001 como para DUC2002, siendo esta métrica más rigurosa que ROUGE-1, ya que compara coincidencias de bi-gramas en lugar de uni-gramas. De esta manera, el MA superó al mejor método, FEOM, en un 14,76% con DUC 2001 y a UnifiedRank en un 6,50%, con DUC2002. Por otro lado, en la medida de ROUGE-1, para el conjunto de DUC 2001, el MA fue superado por el método ED en un 6,08% y para DUC 2002 lo supera UnifiedRank en un 0,40%. Por otro lado, de acuerdo a la técnica de Ranking, el método ocupó el primer lugar superando métodos como ED y UnifiedRank que lo superaban en la medida de ROUGE-1. Considerando que el ranking no tiene en cuenta el porcentaje de mejora, es importante citar que, para el caso de ROUGE-2, el MA con DUC2002, supera con porcentajes bastante altos a ED y a FEOM con 84,82% y 83,01%, respectivamente; y con DUC2001 a UnifiedRank, ED y FEOM en un 14,76%, 9,29% y 9,17%, respectivamente, mientras que el MA, es superado por porcentajes más bajos en la medida ROUGE-1, de 6,08% y 0,40%, para DUC2001 y DUC2002, respectivamente, los cuales son mucho menores en comparación a los altos porcentajes con los que el MA supera a los mejores métodos del estado del arte. Estos resultados indican que la optimización que combina la búsqueda global basada en población, con una búsqueda local heurística para cada agente, acoplando de esta forma la evolución genética con el aprendizaje de los individuos, como ocurre con el MA, es realmente una línea de investigación prometedora.

Por otro lado, según los resultados de la evaluación del comportamiento de la función objetivo, se mostró que la función objetivo *PLRCC* planteada en esta investigación (compuesta por las características de *Posición*, *Longitud*, *Cohesión*, *Cobertura* y *Relación con el título*) permitió una mejor discriminación de las oraciones importantes de un documento frente a la función objetivo propuesta por Sharegui y Hassanabadi [25], pues, además de mejorar el desempeño del MA, también aumentó el desempeño de un esquema basado en Búsqueda Armónica (HS). Adicionalmete, se observó que la función objetivo propuesta con los pesos optimizados *PLRCC_P* fue la que mayor mejoramiento permitió para ambos esquemas de generación de resúmenes.

De otra parte, la evaluación de un criterio de selección de oraciones basado en la Posición y en un cálculo de Cohesión evaluado sobre cada frase del documento, sin la intervención del MA, sumado a los buenos resultados presentados por el método de *Línea base*, sugirieron que los conjuntos de datos utilizados contenían gran parte de la información importante en las oraciones iniciales, por su estructura de noticias, ratificando estudios como el de Nenkova [135], que, mediante la ejecución de la *Línea base*, concluye que las primeras oraciones son importantes para la generación automática de resúmenes genéricos de noticias.

Finalmente, tras la evaluación del MA con y sin aplicación de búsqueda local, se confirmó que la aplicación de optimización local sobre el proceso evolutivo conduce a mejores resultados que si sólo se aplicara una técnica puramente generacional.

Capítulo 6

6 CONCLUSIONES Y TRABAJO FUTURO

6.1 CONCLUSIONES

En este trabajo se propuso un algoritmo memético para generación automática de resúmenes extractivos de un solo documento, conformado por los operadores reproductivos de selección *Basada en el rango* para escoger el padre de un nuevo descendiente, mediante el cual se pretende evitar la dominancia de los agentes más aptos, favoreciendo la diversidad en la población; selección por *Rueda de ruleta* para escoger la madre, a través de la cual se favorece mayormente la presión selectiva; *Cruce de un punto* para generar la descendencia, el cual favorece la presión selectiva mediante la conservación de gran parte del material genético de los padres; mutación *Multi-bit de inserción*, que favorece la diversidad de la población; y reemplazo por *Competencia restringida,* cuya adaptación involucra un balance entre diversidad, con la escogencia aleatoria del grupo, y presión selectiva, al eliminar al peor. La definición de los operadores del algoritmo memético propuesto se desarrolló buscando un balance entre los procesos de exploración y explotación del espacio de búsqueda, y después de todo un proceso de prueba, llegar a esta configuración que permitió un aumento en la calidad de los resúmenes generados.

El método de optimización local que conforma el algoritmo planteado es la *Búsqueda Local Guiada*, la cual mantiene una estrategia de exploración dirigida por la información del problema y la búsqueda, mejorando el desempeño del algoritmo en la calidad de los resúmenes obtenidos, mucho más que otras técnicas de optimización local evaluadas, debido a que incorpora estrategias para explotar las mejores características de evaluación de las frases. En ese sentido, en su configuración se definen las características de la búsqueda como las oraciones de un documento, una combinación de los factores de *Posición* y *Relación con el Título* para calcular el costo de las características y un valor constante para el parámetro λ de 0.05.

Se definió una función objetivo para el algoritmo memético, formada por características como *Posición, Longitud, Cohesión, Cobertura* y *Relación con el título*, que mostró ser muy efectiva en la selección de las oraciones relevantes de un documento, pues además de que permitió llegar a mejores resultados con el algoritmo memético comparado con otros métodos del estado del arte, mejoró el desempeño de un algoritmo basado en HS. Después del proceso de afinación de los pesos de la función objetivo, se encontró que las características más influyentes eran la *Posición, Longitud* y *Relación con el título.*

La elección de un rango de porcentajes de longitud del documento como base para definir la máxima cantidad de oraciones a evaluar, permitió considerar una muestra más representativa del documento para que fuera analizada durante el proceso evolutivo del algoritmo memético. Teniendo en cuenta la diferencia de longitudes entre los documentos evaluados, este aspecto permitió, además, que la cantidad máxima de oraciones a

evaluar para un documento fuera más equitativa con respecto a las longitudes de los demás documentos, ayudando, adicionalmente, a mejorar los resultados obtenidos.

El algoritmo memético propuesto se evaluó por medio de las medidas ROUGE-1 y ROUGE-2, sobre los conjuntos de datos de DUC2001 y DUC2002. Al compararse frente a otros métodos del estado del arte, con la medida ROUGE-2 el MA presenta los mejores resultados, superando al mejor método FEOM en un 14,76% con DUC2001 y a UnifiedRank en un 6,50% con DUC2002. En el caso de la medida ROUGE-1 para el conjunto de DUC2001 es superado por el método ED en un 6,08%; y para DUC2002 por UnifiedRank en 0.40%. Además en el ranking realizado de todos los métodos, el MA ocupa el primer lugar, superando métodos como DE y UnifiedRank que lo superaban en la medida de ROUGE-1 y con una complejidad computacional menor, pues los mecanismos recursivos utilizados por el método de DE incrementan la carga computacional, y la complejidad algorítmica $O((m * \log(v^2/m)/\log(v))$ del método UnifiedRank es muy alta comparada con la del MA que es $O((\mu/(\mu-1))\log(\mu))$. Por otro lado, considerando que el ranking no tiene en cuenta el porcentaje de mejora, es importante resaltar que, para el caso de ROUGE-2, el MA con DUC2002, supera con porcentajes bastante altos a ED y a FEOM con 84,82% y 83,01%, respectivamente; y con DUC2001 a UnifiedRank, ED y FEOM en un 14,76%, 9,29% y 9,17%, respectivamente. Por su parte, el MA, es superado por porcentajes más bajos en la medida ROUGE-1, de 6,08% y 0,40%, para DUC2001 y DUC2002 respectivamente. Finalmente, ROUGE-2 es una métrica un poco más rigurosa que ROUGE-1, ya que evalúa bi-gramas coincidentes entre el resumen generado y el resumen de referencia, en lugar de uni-gramas, lo cual resalta nuevamente el buen desempeño y alta competitividad del MA frente a los métodos del estado del arte, ya que presentó los resultados más altos en esta medida tanto en DUC2001 como en DUC2002.

Un criterio de selección de oraciones relevantes de un documento basado en la combinación de *Posición* con un cálculo de *Cohesión,* evaluado individualmente sobre cada frase, sin la intervención del algoritmo memético, sugirió que los conjuntos de datos utilizados contenían gran parte de la información importante en las oraciones iniciales, debido a su estructura de noticias, lo cual es afirmado en el estudio realizado por Nenkova, que indica que la información de la posición es muy efectiva en la generación de resúmenes genéricos de noticias [135]. En ese sentido, ya que el algoritmo memético contempla en su función objetivo otro tipo de características como: longitud, relación con el título, cobertura y cohesión, éste puede llegar a obtener un mejor desempeño para documentos con una estructura diferente a la de noticias de DUC, lo cual hace que sea un algoritmo que no depende del tipo de estructura de los documentos.

En este trabajo se propuso un algoritmo memético para la generación automática de resúmenes extractivos de un solo documento, el cual, de acuerdo a los buenos resultados presentados, mostró ser un esquema con muy buen desempeño y muy competitivo frente a los modelos evolutivos del estado del arte, e incluso frente a otros modelos no evolutivos estudiados, indicando, además, que la optimización que combina la búsqueda global basada en población, con una búsqueda local heurística para cada agente, acoplando así la evolución genética con el aprendizaje de los individuos, como ocurre con el MA, es realmente una línea de investigación prometedora dentro de la generación automática de resúmenes.

6.2 RECOMENDACIONES

Buscando disminuir el costo computacional del algoritmo propuesto, se modificó la forma de calcular ROUGE, de tal forma que se basara en la cantidad de ejecuciones y no en la cantidad de documentos, evitando, además, pérdidas decimales. Así mismo, se optimizó el código para evitar realizar cálculos repetitivos. Además, teniendo en cuenta que la diferencia de longitudes entre los documentos era considerable, se utilizó un porcentaje variable entre 50 y 70 por ciento de la longitud del documento original, con el fin de evaluar los documentos en iguales condiciones. También se verificó que la longitud de los resúmenes candidatos no fuera inferior a 100 palabras, para que tanto la evaluación con ROUGE como la comparación de los resultados obtenidos con los otros métodos fueran imparciales.

6.3 TRABAJO FUTURO

Con el propósito de mejorar los resultados obtenidos con la configuración del MA propuesto, se espera estudiar otros métodos de selección, cruce, mutación, reemplazo y búsqueda local, que puedan ser también adaptados al problema, y de esta forma encontrar una nueva configuración para el algoritmo memético.

Evaluar el comportamiento del algoritmo memético propuesto con otros conjuntos de datos, para intentar generalizar las conclusiones obtenidas del desempeño del MA en esta investigación.

Realizar un estudio del comportamiento de la función objetivo propuesta con diferentes tipos de documentos (que no sean noticias), con el fin de analizar si su diseño es aplicable directamente sobre documentos de diversos géneros o si es necesaria su adaptación. Además, analizar el comportamiento comparado con el método sin el MA encontrado al final de esta investigación.

BIBLIOGRAFÍA

[1] H. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development,* pp. 159-165, 1958.

[2] P. Baxendale, "Machine-made index for technical literature - an experiment. IBM " *Journal of Research Development,* vol. 2, pp. 354-361, 1958.

[3] H. P. Edmundson, "New Methods in Automatic Extracting," *J. ACM,* vol. 16, pp. 264-285, 1969.

[4] C. Aone, M. E. Okurowski, J. Gorlinsky, and B. s. Larsen, "Trainable, scalable summarization using robust NLP and Machine Learning," *Advances in Automatic Text Summarization,* vol. Mani, I. and Maybury, M. T., pp. 71-80, 1999.

[5] J. Conroy and D. O'leary, "Text summarization via hidden Markov models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* New Orleans, Louisiana, United States: ACM, 2001.

[6] M. A. Fattah and F. Ren, "GA, MR, FFNN, PNN and GMM based models for automatic text summarization," *Computer Speech & Language,* vol. 23, pp. 126-144, 2009.

[7] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval* Seattle, Washington, United States: ACM, 1995.

[8] K. Svore, Vanderwende, L., and Burges, C., "Enhancing single-document summarization by combining RankNet and third-party sources.," *In Proceedings of the EMNLP-CoNLL,* pp. 448-457, 2007.

[9] E. Villatoro Tello, "Generación automática de resúmenes de múltiples documentos," Méjico: Instituto Nacional de Astrofísica, Óptica y Electrónica, 2007, p. 114.

[10] T. C. Wesley and Y. Jihoon, "Extracting sentence segments for text summarization: a machine learning approach," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* Athens, Greece: ACM, 2000.

[11] R. Barzilay, Elhadad, M, "Using Lexical Chains for Text Summarization. ," *In Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain.,* pp. 10–17, 1997.

[12] K. Ono, K. Sumita, and S. Miike, "Abstract generation based on rhetorical structure extraction," in *Proceedings of the 15th conference on Computational linguistics - Volume 1* Kyoto, Japan: Association for Computational Linguistics, 1994.

[13] D. Marcu, " Improving summarization through rhetorical parsing tuning.," *Proceedings of The Sixth Workshop on Very Large Corpora. Montreal, Canada,* pp. 206-215, 1998.

[14] R. Mihalcea, Tarau, P. , "Text-rank - bringing order into texts.," *In Proceeding of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain.,* 2004.

[15] R. Mihalcea, Tarau, P., " An Algorithm for Language Independent Single and Multiple Document Summarization.," *In Proceedings of the International Joint Konference on Natural Language Processing, Korea.,* 2005.

[16] L. Plaza, "Uso de grafos semánticos en la generación automática de resúmenes y estudio de su aplicación en distintos dominios: Biomedicina, periodismo y turismo," in *Departamento de Ingeniería del Software e Inteligencia Artificial*. vol. PhD Madrid: Universidad Complutense de Madrid, 2011, p. 176.

[17] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science,* vol. 41, pp. 391-407, 1990.

[18] T. K. Landauer and S. T. Dumais, "Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge," *Psychological Review,* 1997.

[19] J. Steinberger and K. Ježek, "Sentence Compression for the LSA-based Summarizer," pp. 141–148, 2006.

[20] R. Alguliev, R. Aliguliyev, M. Hajirahimova, and C. Mehdiyev, "MCMR: Maximum coverage and minimum redundant text summarization model," *Expert Systems with Applications,* vol. In Press, Corrected Proof, 2011.

[21] M. S. Binwahlan, N. Salim, and L. Suanmali, "Swarm Based Text Summarization," in *Computer Science and Information Technology - Spring Conference, 2009. IACSITSC '09. International Association of*, 2009, pp. 145-150.

[22] M. S. Binwahlan, N. Salim, and L. Suanmali, "Fuzzy swarm based text summarization," *Journal Computer Sciences,* vol. 5, pp. 338–346, 2009.

[23] M. Litvak, M. Last, and M. Friedman, "A new approach to improving multilingual summarization using a genetic algorithm," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* Uppsala, Sweden: Association for Computational Linguistics, 2010.

[24] V. Qazvinian, L. Sharif, and R. Halavati, "Summarising text with a genetic algorithm-based sentence extraction," *International Journal of Knowledge Management Studies (IJKMS),* vol. 2, pp. 426-444, 2008.

[25] E. Shareghi and L. S. Hassanabadi, "Text summarization with harmony search algorithm-based sentence extraction," in *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology* Cergy-Pontoise, France, 2008.

[26] P.-K. Dehkordi, F. Kumarci, and H. Khosravi, "Text Summarization Based on Genetic Programming," in *International Journal of Computing and ICT Research*, 2009, pp. 57-64.

[27] R. M. Aliguliyev, "A new sentence similarity measure and sentence based extractive technique for automatic text summarization," *Expert Syst. Appl.,* vol. 36, pp. 7764-7772, 2009.

[28] W. Hart, N. Krasnogor, J. Smith, and W. Hart, "Memetic Evolutionary Algorithms Recent Advances in Memetic Algorithms." vol. 166: Springer Berlin / Heidelberg, 2005, pp. 3-27.

[29] P. Moscato, "A gentle introduction to memetic algorithms," *Handbook of Metaheuristics,* pp. 105-144, 2003.

[30] K. Ježek and J. Steinberger, "Automatic Text Summarization (The state of the art 2007 and new challenges)," in *Znalosti 2008*, Bratislava, Slovakia, 2008, pp. 1-12.

[31] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Comput. Linguist.,* vol. 28, pp. 399-408, 2002.

[32] K. Sparck Jones and J. R. Galliers, *Evaluating Natural Language Processing Systems: An Analysis and Review*: Springer-Verlag New York, Inc., 1996.

[33] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in *Proceedings of the 20th international joint conference on Artifical intelligence* Hyderabad, India: Morgan Kaufmann Publishers Inc., 2007.

[34] D. M. Dunlavy, D. P. Oâ€™Leary, J. M. Conroy, and J. D. Schlesinger, "QCS: A system for querying, clustering and summarizing documents," *Information Processing & Management,* vol. 43, pp. 1588-1605, 2007.

[35] B. Araly and V. Rakesh, "Automated extractive single-document summarization: beating the baselines with a new approach," in *Proceedings of the 2011 ACM Symposium on Applied Computing* TaiChung, Taiwan: ACM, 2011.

[36] X. Wan, "Towards a Unified Approach to Simultaneous Single-Document and Multi-Document Summarizations," *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010),* pp. 1137-1145, August 2010.

[37] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I. H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," *Information Processing & Management,* vol. 41, pp. 75-95, 2005.

[38] Y. Gong, "Generic text summarization using relevance measure and latent semantic analysis," *in Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 2001.

[39] J. Steinberger and K. Jezek, "Using Latent Semantic Analysis in Text Summarization and Summary Evaluation," in *Proceedings of the 7th International Conference ISIM*, 2004.

[40] X. Wan, "A novel document similarity measure based on earth mover's distance," *Inf. Sci.,* vol. 177, pp. 3718-3730, 2007.

[41] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using N-gram co-occurrence statistics," in *Proceedings of the 2003 Conference of the*

North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 Edmonton, Canada: Association for Computational Linguistics, 2003.

[42] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on,* vol. 4, pp. 1942-1948 vol.4, November 1995.

[43] J. Q. Lima M. and B. Baran C., "Optimización de Enjambre de Particulas aplicada al Problema del Cajero Viajante Bi-objetivo.," *Inteligencia Artificial, Revista Iberoamericana de IA,* vol. 10, pp. 67-76, 2006.

[44] W. Song, L. Cheon Choi, S. Cheol Park, and X. Feng Ding, "Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization," *Expert Systems with Applications,* vol. 38, pp. 9112-9121, 2011.

[45] M. Hassel, "Resource Lean and Portable Automatic Text Summarization," in *Computer Science and Communication*. vol. Doctoral Stockholm, Sweden KTH School of Computer Science and Communication, 2007, p. 144.

[46] C. J. V. Rijsbergen, *Information Retrieval*. Newton, MA, USA: Butterworth-Heinemann, 1979.

[47] D. Radev, H. Jing, and M. Budzikowska, "Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation, and User Studies," *In ANLP/NAACL Workshop on Summarization,* pp. 21-29, 2000.

[48] R. L. Donaway, K. W. Drummey, and L. A. Mather, "A comparison of rankings produced by summarization evaluation measures," in *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization* Seattle, Washington: Association for Computational Linguistics, 2000.

[49] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop,* pp. 74-81, July 2004.

[50] P. Kishore, R. Salim, W. Todd, and Z. Wei-Jing, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002.

[51] A. Nenkova, "Automatic Text Summarization of Newswire: Lessons Learned from the {D}ocument {U}nderstanding {C}onference," in *Proceedings of the 20th AAAI*, 2005, pp. 1436-1441.

[52] C. Galvez, F. de Moya-Anegón, and V. H. Solana, "Term conflation methods in information retrieval: Non-linguistic and linguistic approaches," *Journal of Documentation,* vol. 61, pp. 520-547, 2005.

[53] A. Singhal, "Modern Information Retrieval: A Brief Overview," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,* vol. 24, pp. 35-42, 2001.

[54] J. C. Reynar, "Topic Segmentation: Algorithms and Applications," 1998.

[55] H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," *IBM Journal of Research and Development,* vol. 1, pp. 309-317, 1957.

[56] R. Lo, B. He, and I. Ounis, "Automatically Building a Stopword List for an Information Retrieval System," in *Proceedings of the Fifth Dutch-Belgian Information Retrieval Workshop*, 2005.

[57] J. Lovins, "Development of a Stemming Algorithm," *Mechanical Translation and Computational Linguistics,* vol. 11, 1968.

[58] M. F. Porter, "An algorithm for suffix stripping," *Program: Electronic Library & Information Systems,* vol. 40, pp. 211-218, 2006.

[59] H. Huang and B. Zhang, "Text Indexing and Retrieval," in *Encyclopedia of Database Systems*, 2009, pp. 3055-3058.

[60] G. Salton, A. Wong, and C. Yang, "A vector space model for information retrieval," *Communications of The ACM,* 1975.

[61] A. G. López Herrera, "Modelos de Sistemas de Recuperación de Información Documental Basados en Información Lingüística Difusa," in *Departamento de la Ciencias de la computación y la Inteligencia artificial*. vol. Phd Granada: Universidad de Granada, 2006, p. 255.

[62] R. Montiel Soto, R. A. García-Hernández, Y. Ledeneva, and R. Cruz Reyes, "Comparison of three text models for automatic generation of summaries," *Procesamiento del lenguaje natural,* vol. 43, 2009.

[63] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[64] E. Greengrass, "Information Retrieval: A Survey," 2000.

[65] A. Pons Porrata, "Desarrollo de algoritmos para la Estructuración dinámica de información y su aplicación a la detección de sucesos," in *Departamento de Lenguajes y Sistemas Informáticos*. vol. Phd Castellón: Universidad Jaume I, 2004, p. 161.

[66] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*: Cambridge University Press, 2008.

[67] M. S. Binwahlan, N. Salim, and L. Suanmali, "Fuzzy swarm diversity hybrid model for text summarization," *Information Processing and Management,* vol. 46, pp. 571-588, 2010.

[68] R. Alguliev and R. Alguliyev, "Evolutionary Algorithm for Extractive Text Summarization," *Journal of Intelligent Information Management,* vol. 1, pp. 128-138, 2009.

[69] R. Cilibrasi and P. Vitanyi, "Automatic Meaning Discovery Using Google," *Manuscript, CWI, 2004; http://arxiv.org/abs/cs.CL/0412098,* 2004.

[70] C. H. Bennett, C. H. Bennett, C. H. Bennett, P. Gacs, M. Li, P. M. B. Vitanyi, and W. H. Zurek, "Information Distance," 1997.

[71] R. Dawkins, *The selfish gene*. Oxford; New York: Oxford University Press, 1989.

[72] P. Moscato and C. Cotta, "Introducción a los Algoritmos Meméticos," *Revista Iberoamericana de Inteligencia Artificial,* vol. 7, pp. 131-148, 2003.

[73] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts - Towards Memetic Algorithms," 1989.
[74] N. J. Radcliffe and P. D. Surry, "Formal Memetic Algorithms," 1994.
[75] R. Berretta, C. Cotta, and P. Moscato, " Enhancing the performance of memetic algorithms by using a matching-based recombination algorithm: Results on the number partitioning problem - Results on . . . ," *METAHEURISTICS: COMPUTER-DECISION MAKING,* pp. 65-90, 2003.
[76] C. Cotta, "Una Visión General de los Algoritmos Meméticos," 2007, pp. 139-166.
[77] S. J. Louis, Y. Xiangying, and Y. Zhen Ya, "Multiple vehicle routing with time windows using genetic algorithms," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, 1999, p. 1808 Vol. 3.
[78] P. D. Surry and N. J. Radcliffe, "Inoculation to Initialise Evolutionary Search," *Selected Papers from AISB Workshop on Evolutionary Computing,* pp. 269-285, 1996.
[79] D. E. Goldberg and S. Voessner, "Optimizing global-local search hybrids," *In GECCO,* pp. 220-228, 1999.
[80] D. Beasley, D. R. Bull, and R. R. Martin, "An Overview of Genetic Algorithms: Part 1, Fundamentals," 1993.
[81] T. Blickle and L. Thiele, "A Comparison of Selection Schemes Used in Genetic Algorithms," 1995.
[82] D. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of Genetic Algorithms*, 1991, pp. 69-93.
[83] J. E. Baker, "An analysis of the effects of selection in genetic algorithms," Vanderbilt University, 1989.
[84] V. Yanibelli, "Algoritmos Genéticos y Meméticos," Instituto de Sistemas Tandil, Buenos Aires RR001-2007, 2007.
[85] K. De Jong, "Analysis of the behavior of a class of genetic adaptive systems," University of Michigan, 1975.
[86] G. Harik, "Finding Multimodal Solutions Using Restricted Tournament Selection," in *Proceedings of the 6th International Conference on Genetic Algorithms*, 1995, pp. 24-31.
[87] A. P. Alves da Silva, "Tutorial on Genetic Algorithms," *Journal of the Brazilian Neural Network Society Learning and Nonlinear Models,* vol. 1, pp. 43-58, 2002.
[88] C. Reeves, F. Glover, and G. Kochenberger, "Genetic Algorithms," in *Handbook of Metaheuristics*. vol. 57: Springer New York, 2003, pp. 55-82.
[89] S. Sánchez Caballero, "Optimización estructural y topológica de estructuras morfológicamente no definidas mediante algoritmos genéticos," in *Departamento de Ingeniería mecánica y de materiales*. vol. Ph.D Valencia: Universidad de Valencia, 2012, p. 398.
[90] G. Syswerda, "Uniform Crossover in Genetic Algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*: Morgan Kaufmann Publishers Inc., 1989.

[91] N. J. Radcliffe, "Forma Analysis and Random Respectful Recombination," *ICGA'91,* pp. 222-229, 1991.
[92] Y. Kaya, M. Uyar, and R. Tekin, "A Novel Crossover Operator for Genetic Algorithms: Ring Crossover," *CoRR,* vol. abs/1105.0355, 2011.
[93] M. Gestal Pose, "Introducción a los algoritmos genéticos," p. 16, 2005.
[94] W. Banzahf, "The "molecular" traveling salesman," *Biological cybermetics,* 1990.
[95] D. B. Fogel, "An evolutionary approach to the traveling salesman problem," *Biological cybermetics,* vol. 9, pp. 139-144, 1988.
[96] Z. Michalewicz *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, 1992.
[97] C. García Martínez, "Algoritmos genéticos locales," in *Departamento de Ciencias de la Computación* Granada: Universidad de Granada, 2008.
[98] L. Davis, *Handbook of Genetic Algorithms*, 1991.
[99] K. De Jong and J. Sarma, "Generation gaps revisited," *Foundations of Genetic Algorithms,* vol. 2, pp. 19-28, 1993.
[100] D. Poole and A. Mackworth, "Local Search," in *Artificial Intelligence: Foundations of computational agents*, C. U. Press, Ed. Cambridge, 2010.
[101] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.,* vol. 35, pp. 268-308, 2003.
[102] P. Hansen and N. Mladenovic, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research,* vol. 130, pp. 449-467, 2001.
[103] C. Voudouris and E. Tsang, "Guided Local Search," University of Essex, Colchester CSM-247, 1995.
[104] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers and Operations Research,* vol. 13, pp. 533-549, 1986.
[105] F. Glover and B. Melián-Batista, "Búsqueda Tabú," *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial,* vol. 7, pp. 29-48, 2003.
[106] H. Lourenco, O. Martin, and T. Stutzle, "A Beginner's Introduction to Iterated Local Search," in *Proceedings of MIC 2001*, 2001.
[107] K. s. Pratt, "Design Patterns for Research Methods: Iterative Field Research," *Association for the Advancement of Artificial Intelligence,* 2009.
[108] F. Rothlauf, *Representations for genetic and evolutionary algorithms*. pub-SV:adr: Spring\-er-Ver\-lag, 2006.
[109] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing,* vol. 4, pp. 65-85, 1994.
[110] R. Brandow, K. Mitze, and L. F. Rau, "Automatic condensation of electronic publications by sentence selection," *Information Processing & Management,* vol. 31, pp. 675-685, 1995.
[111] J. M. González, "Los textos periodísticos," in *Las variedades temáticas del texto*. vol. 1, A. d. letras, Ed. Sevilla, 2010.

[112] M. P. Diezhandino Nieto, "El dilema de la Pirámide invertida: Estructura de la noticia," in *El quehacer informativo*, E. H. Unibertsitatea, Ed. Bilbao, 1994.
[113] J. L. Neto, A. A. Freitas, and C. A. A. Kaestner, "Automatic Text Summarization Using a Machine Learning Approach," *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence,* pp. 205-215, 2002.
[114] D. R. Radev, H. Jing, M. g. StyÅ›, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing & Management,* vol. 40, pp. 919-938, 2004.
[115] C.-Y. Lin and E. Hovy, "Identifying topics by position," *Proceedings of the fifth conference on Applied natural language processing,* pp. 283-290, 1997.
[116] D. Gillick, B. Favre, D. Gillick, B. Favre, D. Hakkani-tur, B. Bohnet, Y. Liu, and S. Xie, "The ICSI/UTD Summarization System at TAC 2009," 2009.
[117] F. Schilder and R. Kondadadi, "FastSum: fast and accurate query-based multi-document summarization," *In Proceedings of ACL-08: HLT,* 2008.
[118] A. Bossard, M. Genereux, and T. Poibeau, "Description of the LIPN Systems at TAC 2008: Summarizing Information and Opinions," *In Notebook Papers and Results, Text Analysis Conference (TAC-2008,* 2008.
[119] S. Teufel and M. Moens, " Argumentative Classification of Extracted Sentences as a First Step towards Flexible Abstracting," *Advances in Automatic Text Summarization,* pp. 155-171, 1999.
[120] V. Gupta, P. Chauhan, and S. Garg, "An Statistical Tool for Multi-Document Summarization," *International Journal of Scientific and Research Publications,* vol. 2, 2012.
[121] Z. Xie, X. Li, B. Di Eugenio, P. C. Nelson, W. Xiao, and T. M. Tirpak, "Using gene expression programming to construct sentence ranking functions for text summarization," *Proceedings of the 20th international conference on Computational Linguistics,* 2004.
[122] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management,* pp. 513-523, 1988.
[123] L. Eshelman and D. Schaffer, "Preventing Premature Convergence in Genetic Algorithms by Preventing Incest," in *{P}roceedings of the Fourth International Conference on Genetic Algorithms*, 1991, pp. 115-122.
[124] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
[125] D. Sudholt, "Computational complexity of evolutionary algorithms, hybridizations, and swarm intelligence.," Dortmund University of Technology, 2008.
[126] C. R. Reeves, "Using Genetic Algorithms With Small Populations," *Proceedings of the Fifth International Conference on Genetic Algorithms,* pp. 92-99, 1993.
[127] F. Neri, C. Cotta, P. Moscato, and J.-K. Hao, "Memetic Algorithms in Discrete Optimization," in *Handbook of Memetic Algorithms*. vol. 379, Springer, Ed.: Springer Berlin Heidelberg, 2012, pp. 73-94.

[128] D. Gillick, "Sentence Boundary Detection and the Problem with the U.S," in *Proc. Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL: HLT), Companion Volume: Short Papers*, Boulder, Colorado, 2009, pp. 241-244.

[129] G. Salton, *The SMART Retrieval System---Experiments in Automatic Document Processing*: Prentice-Hall, Inc., 1971.

[130] A. S. Foundation, "Apache Lucene Core." vol. 2013, 2011.

[131] P. Over and W. Liggett, "Introduction to DUC-2002: an Intrinsic Evaluation of Generic News Text Summarization Systems," in *Document Understanding Conferences*, Philadelphia, 2002.

[132] J. Bisbal Riera, "Transformación de algoritmos recursivos en iterativos," in *Recursividad, Complejidad y Diseño de Algoritmos*, UOC, Ed. Barcelona: Laburo, 2009.

[133] D. L. Neel and M. E. Orrison, "The Linear Complexity of a Graph," *Electronic Journal of Combinatorics,* vol. 13, 2006.

[134] W. Tamayo and M. Vela, "Generación Automática De Resúmenes De Múltiples Documentos Basada En El Algoritmo GHS+LEM," in *Departamento de Sistemas*. vol. Ing. Popayán: Universidad del Cauca, 2012, p. 67.

[135] A. Nenkova, "Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference," in *Proceedings of the 20th AAAI*, 2005, pp. 1436-1441.

[136] Y. Ouyang, W. Li, Q. Lu, and R. Zhang, "A study on position information in document summarization," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* Beijing, China: Association for Computational Linguistics, 2010.