Generación Automática de Resúmenes de Múltiples Documentos mediante la Hibridación de la Metaheurística de la Mejor Búsqueda Armónica Global y el Algoritmo basado en Grafos LexRank

César Marino Cuéllar Chacón

Trabajo de Grado para optar al título de Master en Computación

Director: Dr. Martha Eliana Mendoza Becerra
Co-Director: Dr. 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 - 2018

Generación Automática de Resúmenes de Múltiples Documentos mediante la Hibridación de la Metaheurística de la Mejor Búsqueda Armónica Global y el Algoritmo basado en Grafos LexRank

César Marino Cuéllar Chacón

Director: Dr. Martha Eliana Mendoza Becerra
Co-Director: Dr. 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 - 2018

Dedicado a:

A mi esposa por todo su apoyo y ánimo que me brinda día a día para alcanzar nuevos logros tanto profesionales como personales.
A mí hijo Andrés Felipe por hacerme el papá más orgulloso

Agradecimientos

A la Ingeniera Dra. Marta Eliana Mendoza y al Ingeniero Dr. Carlos Alberto Cobos por darme la oportunidad de realizar este trabajo bajo su dirección, por su dedicación, tiempo y por todo el apoyo recibido para poder culminar este proyecto de investigación.

A los demás profesores de la Maestría en Computación de la Universidad del Cauca, por sus enseñanzas, apoyo y colaboración durante mi proceso de formación en la maestría.

A la Vicerrectoría de Investigaciones por el apoyo brindado mediante la aprobación del proyecto de Investigación que me permitió poder profundizar mis tareas de investigación, así como la oportunidad de realizar una pasantía internacional y el apoyo para poder participar en congreso internacional para la presentación del artículo.

Al Servicio Nacional de Aprendizaje SENA por apoyarme financieramente durante mi formación, por el apoyo en la realización de la pasantía internacional y por el apoyo en las demás actividades en el desarrollo de la maestría.

TABLA DE CONTENIDO

Presentación
Capítulo 1 1
1 INTRODUCCIÓN 1
1.1 PLANTEAMIENTO DEL PROBLEMA 1
1.2 APORTES 3
1.3 OBJETIVOS 3
1.3.1 OBJETIVO GENERAL 3
1.3.2 OBJETIVOS ESPECÍFICOS 3
1.4 RESULTADOS OBTENIDOS 3
Capítulo 2 6
2 CONTEXTO TEÓRICO Y ESTADO DEL ARTE 6
2.1 GENERACIÓN AUTOMÁTICA DE RESÚMENES DE TEXTO 6
2.1.1 Métodos para la generación automática de resúmenes de múltiples documentos 7
2.1.2 Métodos de evaluación de la calidad de los resúmenes 11
2.2 REPRESENTACIÓN DE LOS DOCUMENTOS 13
2.2.1 Modelo de espacio vectorial 14
2.2.2 Técnicas de ponderación de términos 15
2.2.3 Medidas de Similitud 16
2.2.4 Representación de documentos por medio de matrices 16
2.2.5 Vector Centroide 17
2.3 ALGORITMOS BASE 18
2.3.1 Mejor Búsqueda Armónica Global (GBHS) con Búsqueda Local 18
2.3.2 Proceso de Optimización Búsqueda Local Codiciosa 19
2.3.3 LexRank 20
Capítulo 3 27
3 PROCESO DE CONSTRUCCIÓN: ALGORITMOS HÍBRIDOS 27
3.1 CICLO I: Definición hibridación y función objetivo 27
3.1.1 Identificación formas de hibridación 27
3.1.2 Selección formas de Hibridación algoritmos base 28
3.1.3 Diseño de la función objetivo 31
3.1.4 Configuraciones de la Función Objetivo 34
3.1.5 Configuración parámetros algoritmos híbridos 36
3.1.6 Afinación de las funciones objetivo 38
3.2 CICLO II: Diseño algoritmo híbrido Versión 1 38
3.2.1 Afinación primera función objetivo 40
3.2.2 Afinación segunda función objetivo 41

3.2.3 Afinación tercera función objetivo 42
3.2.4 Configuración definitiva de la función objetivo 42
3.2.5 Afinación de Parámetros 43
3.2.6 Esquema del Algoritmo LexRank-GBHS 44
3.3 CICLO III: Diseño algoritmo híbrido versión 2 45
3.3.1 Afinación primera función objetivo 45
3.3.2 Afinación segunda función objetivo 46
3.3.3 Afinación tercera función objetivo 46
3.3.4 Configuración definitiva de la función objetivo 47
3.3.5 Afinación de Parámetros 47
3.3.6 Esquema del Algoritmo GBHS-LexRank 48
3.4 CICLO IV: Diseño algoritmo híbrido versión 3 49
3.4.1 Afinación primera función objetivo 49
3.4.2 Afinación segunda función objetivo 50
3.4.3 Afinación tercera función objetivo 50
3.4.4 Configuración definitiva de la función objetivo 50
3.4.5 Afinación de Parámetros 51
3.4.6 Esquema del Algoritmo LexRank-GBHS-2Resumen 51
3.5 CICLO V: Diseño algoritmo híbrido versión 4 53
3.5.1 Afinación primera función objetivo 53
3.5.2 Afinación segunda función objetivo 54
3.5.3 Afinación tercera función objetivo 54
3.5.4 Configuración definitiva de la función objetivo 54
3.5.5 Afinación de Parámetros 55
3.5.6 Esquema del Algoritmo GBHS-LexRank-2Resumen 55
3.6 CICLO VI: Diseño algoritmo híbrido versión 5 57
3.6.1 Afinación primera función objetivo 57
3.6.2 Afinación segunda función objetivo 57
3.6.3 Afinación tercera función objetivo 58
3.6.4 Configuración definitiva de la función objetivo 58
3.6.5 Afinación de Parámetros 59
3.6.6 Esquema del Algoritmo Versión LexRank-GBHS-Paralelo 59
Capítulo 4 61
4 ALGORITMO PROPUESTO: LEXRANK-GBHS 61
4.1 REPRESENTACIÓN DE LAS SOLUCIONES 61
4.2 FUNCIÓN OBJETIVO 61
4.3 ADAPTACIONES DEL ALGORITMO 62
4.3.1 Algoritmo base LexRank 63
4.3.2 Por Hibridación 64
4.3.3 Algoritmo GBHS 64
4.4 ESQUEMA DE GENERACIÓN DE RESÚMENES 65
4.5 EVALUACIÓN DE CALIDAD 66
4.5.1 Normalización e Indexación de Documentos 66

4.5.2 Métricas de Evaluación 68
4.5.3 Afinación de Parámetros 68
4.5.4 Comparación con Métodos del Estado del Arte 69
Capítulo 5 72
5 APLICACIÓN WEB 72
5.1 DISEÑO DE LA APLICACIÓN WEB 72
5.1.1 Arquitectura de la Aplicación 72
5.1.2 Diagrama de Paquetes 73
5.1.3 Interfaces de Usuario 73
5.2 EVALUACIÓN DE SATISFACCIÓN DEL USUARIO 77
5.2.1 Diseño de la Encuesta 78
5.2.2 Aplicación de la Encuesta 79
5.2.3 Análisis de Resultados 79
Capítulo 6 82
6 CONCLUSIONES Y TRABAJO FUTURO 82
6.1 CONCLUSIONES 82
6.2 RECOMENDACIONES Y TRABAJO FUTURO 83
BIBLIOGRAFÍA 85

Lista de tablas

Tabla 1. Subconjunto de documentos del tópico d1003t de DUC 2004..... 21
Tabla 2. Matriz similitud de Cosenos para tópico d1003t de DUC 2004..... 22
Tabla 3. Grado de Centralidad..... 23
Tabla 4 Parámetros Algoritmo LexRank 36
Tabla 5 Parámetros Algoritmo Mejor Búsqueda Armónica..... 37
Tabla 6 Parámetros Asociados al Problema 38
Tabla 7 Parámetros Asociados a la Función Objetivo..... 38
Tabla 8 Medidas ROUGE afinación primera Función Objetivo..... 41
Tabla 9 Medidas ROUGE, afinación Segunda Función Objetivo..... 42
Tabla 10 Medidas ROUGE, afinación Tercera Función Objetivo..... 42
Tabla 11 Medidas Rouge mejores para cada función objetivo 43
Tabla 12 Mejor configuración de parámetros Algoritmo LexRank-GBHS..... 43
Tabla 13 Medidas ROUGE, afinación Primera Función Objetivo 45
Tabla 14 Medidas ROUGE, afinación Segunda Función Objetivo..... 46
Tabla 15 Medidas ROUGE, afinación Tercera Función Objetivo..... 46
Tabla 16 Resultados de afinación de las funciones objetivo 47
Tabla 17 Mejor configuración de parámetros Algoritmo GBHS-LexRank..... 47
Tabla 18 Medidas ROUGE, afinación Primera Función Objetivo 49
Tabla 19 Medidas ROUGE, afinación Segunda Función Objetivo..... 50
Tabla 20 Medidas ROUGE, afinación Tercera Función Objetivo..... 50
Tabla 21 Resultados de afinación de las funciones objetivo 51
Tabla 22 Mejor configuración parámetros LexRank-GBHS-2Resumen..... 51
Tabla 23 Medidas ROUGE, afinación Primera Función Objetivo 53
Tabla 24 Medidas ROUGE, afinación Segunda Función Objetivo..... 54
Tabla 25 Medidas ROUGE, afinación Tercera Función Objetivo..... 54
Tabla 26 Resultados de afinación de las funciones objetivo 55
Tabla 27 Mejor configuración parámetros GBHS-LexRank-2Resumen..... 55
Tabla 28 Medidas ROUGE afinación primera Función Objetivo..... 57
Tabla 29 Medidas ROUGE afinación Segunda Función Objetivo..... 58
Tabla 30 Medidas ROUGE afinación Tercera Función Objetivo..... 58
Tabla 31 Resultados de afinación de las funciones objetivo 59
Tabla 32 Mejor configuración de parámetros LexRank-GBHS-Paralelo..... 59
Tabla 33 Mejor Resultado de Cada Ciclo..... 60
Tabla 34 Descripción conjuntos de datos utilizados..... 68
Tabla 35 Parámetros obtenidos mejor configuración LexRank-GBHS 69
Tabla 36 Valores Rouge de los métodos en DUC2005 y DUC2006..... 70
Tabla 37 Posición resultante de los métodos..... 71
Tabla 38 Métricas para evaluar la calidad norma ISO/IEC 25023 78
Tabla 39 Clasificación preguntas de acuerdo a características de calidad..... 78
Tabla 40 Escala de Likert 79
Tabla 41 Resultados evaluación según escala de Likert..... 80

Lista de figuras

Figura 2.1. Representación oraciones espacio vectorial tridimensional 14
Figura 2.2 Improvisación de una nueva armonía 19
Figura 2.3 Búsqueda Codiciosa 20
Figura 2.4. Grafos de Similitud con umbrales 0, 0.1, 0.2 y 0.3. 22
Figura 2.5 Algoritmo Método de Potencia 25
Figura 2.6 Algoritmo Cálculo de puntuación de LexRank, ... 26
Figura 3.1 Formas Hibridación Tipo Alto Nivel-Relevo 27
Figura 3.2 Formas Hibridación Tipo Bajo Nivel 28
Figura 3.3 Hibridación Alto Nivel- Trabajo en Equipo 28
Figura 3.4 Hibridación 1 LexRank-GBHS 29
Figura 3.5 Hibridación 1 GBHS-LexRank 29
Figura 3.6 Hibridación 1 LexRank-GBHS 30
Figura 3.7 Hibridación 1 GBHS-LexRank-2Resumen 30
Figura 3.8 Hibridación 3 Paralela LexRank-GBHS-Paralelo 31
Figura 3.9 Algoritmo Procedimiento Técnica GRASP 40
Figura 3.10 Algoritmo Versión 1 LexRank-GBHS 45
Figura 3.11 Algoritmo Versión 2 GBHS-LexRank 49
Figura 3.12 Algoritmo Versión 3 LexRank-GBHS-2Resumen 53
Figura 3.13 Algoritmo Versión 4 GBHS-LexRank-2Resumen 57
Figura 3.14 Algoritmo Versión 5 LexRank-GBHS-Paralelo 60
Figura 4.1 Algoritmo propuesto LexRank-GBHS 63
Figura 4.2 Algoritmo proceso de reparación de una armonía 65
Figura 4.3 Algoritmo habilitar oración en la armonía 65
Figura 4.4 Esquema de Generación de Resúmenes 66
Figura 5.1 Arquitectura de la Aplicación Web 72
Figura 5.2 Diagrama de Paquetes 73
Figura 5.3 Interfaz inicial de la Aplicación 74
Figura 5.4 Interfaz Agregar Documentos Fuente 74
Figura 5.5 Interfaz Lista y Revisión Documentos Fuente 75
Figura 5.6 Interfaz Configuración de Parámetros Algoritmo 76
Figura 5.7 Interfaz Resumen Generado 76
Figura 5.8 División de la norma ISO 25000. Fuente [74] 77
Figura 5.9 Diagrama de barras resultados evaluación satisfacción 81

Presentación

Actualmente las personas y las organizaciones producen gran cantidad de documentos que se publican en internet y su acceso desde cualquier lugar o dispositivo crece de forma exponencial. Cuando un usuario necesita encontrar información relacionada con un tema específico, se enfrenta a una sobrecarga de información, lo que hace necesario contar con un resumen con las ideas principales contenidas en los documentos, que le permita al usuario decidir cuales documentos realmente resuelven sus necesidades de información, reduciendo de esta forma el tiempo de búsqueda y aumentando la satisfacción de dichos usuarios.

En este documento, se plantea un algoritmo híbrido que permite la generación automática de resúmenes extractivos de múltiples documentos, basados en la metaheurística de la Mejor Búsqueda Armónica Global (GBHS) [1] y el algoritmo basado en grafos LexRank [2]. A lo largo del documento se describen las bases teóricas y el proceso de desarrollo realizado.

En el capítulo 1 se presenta una descripción del problema motivo de este proyecto, la generación automática de resúmenes extractivos de múltiples documentos, así como los aportes de la investigación mediante la generación de nuevo conocimiento; se relacionan los objetivos propuestos para la generación de una alternativa de solución al problema planteado, así como los resultados obtenidos al concluir el desarrollo de esta investigación.

El capítulo 2 presenta los conceptos más importantes del área de investigación de generación de resúmenes, incluyendo los métodos del estado del arte en esta área de investigación y las medidas de calidad de resúmenes automáticos aceptadas por la comunidad científica. También se explica la representación de documentos en el modelo espacio vectorial, las medidas asociadas para ponderación de términos, junto a la medida de similitud de cosenos. Además se describen los algoritmos base LexRank y GBHS que hacen parte del algoritmo híbrido propuesto.

En el capítulo 3 se presenta el proceso llevado a cabo para el desarrollo del algoritmo híbrido propuesto LexRank-GBHS, realizando la descripción de los diferentes ciclos de acuerdo a la metodología de desarrollo utilizada. En el primer ciclo se identifican las posibles formas de hibridar los algoritmos base (GBHS y LexRank) y se hace la selección de cada una de las versiones del algoritmo híbrido; también se define el proceso de diseño de la función objetivo teniendo en cuenta diferentes características de similitud y sus diferentes configuraciones; se explica el proceso de configuración de parámetros asociados al problema y a cada uno de los algoritmos base; y por último el proceso de afinación de los pesos asociados a cada una de las características de la función objetivo. En los siguientes ciclos se diseñaron y afinaron las versiones del algoritmo híbrido así: segundo ciclo (LexRank-GBHS), tercer ciclo (GBHS-LexRank), cuarto ciclo (LexRank-GBHS-2Resumen), quinto ciclo (GBHS-LexRank-2Resumen) y en último ciclo (LexRank-GBHS-Paralelo).

En el capítulo 4 se presenta el algoritmo híbrido propuesto LexRank-GBHS, realizando la descripción de la representación de las soluciones para el problema de la generación automática de resúmenes de múltiples documentos, la definición de la función objetivo, las diferentes adaptaciones de los algoritmos base y el esquema general de la generación de resúmenes de texto. Además, se describen las tareas de pre-procesamiento realizadas a los documentos originales (conjuntos de datos DUC2005 y DUC2006) y finalmente se presentan los resultados de la evaluación de calidad de los resúmenes generados por el Algoritmo LexRank-GBHS de acuerdo a las medidas ROUGE-1, ROUGE-2 y ROUGE-SU4 sobre los conjuntos de datos DUC2005 y DUC2006, comparados con otros métodos del estado del arte.

En el capítulo 5 se presenta la aplicación web desarrollada en VS .NET, para la generación automática de resúmenes extractivos de múltiples documentos, que implementa el algoritmo híbrido propuesto LexRank-GBHS y la evaluación de satisfacción de usuario de la aplicación.

En el capítulo 6 se presentan las conclusiones del proceso de adaptación del algoritmo híbrido propuesto al problema de generación automática de resúmenes de múltiples documentos y del resultado de la evaluación de calidad. Además, se exponen recomendaciones para trabajos de investigación similares y probables líneas de trabajo futuro que nacen de la presente investigación

Al final se presentan las referencias bibliográficas utilizadas durante el desarrollo del proyecto.

Capítulo 1

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

Debido al crecimiento exponencial de documentos de texto digitales en internet y en las organizaciones, cada día se incrementa la necesidad de que los sistemas de búsqueda sean más efectivos y ofrezcan a sus usuarios resúmenes de documentos de una misma temática, permitiéndoles poder decidir más rápidamente cuales documentos realmente resuelven sus necesidades de información, buscando de esta manera reducir el tiempo de búsqueda y la satisfacción de dichos usuarios. En este contexto, la generación automática de resúmenes de texto es una herramienta importante que permite obtener de manera rápida y sencilla la información más relevante de los documentos.

El área de investigación de generación automática de resúmenes extractivos de múltiples documentos es una de las tareas prioritarias del área de procesamiento de lenguaje natural, que tiene por objetivo resumir el contenido de múltiples documentos conservando la información más importante en un texto corto. Esta tarea, involucra retos relativos con la calidad de los resúmenes (que tan parecido es el resumen generado automáticamente con resúmenes generados por humanos), la coherencia o la redundancia de los mismos, entre otros. Algunas áreas de aplicación de la generación automática de resúmenes de múltiples documentos incluyen: noticias y reportes relacionados con los desastres [3]; E-learning [4], para seleccionar la información más importante de un texto; Motores de búsqueda [5] para obtener un breve resumen del documento o página web, también para obtener un resumen basándose en información contextual del usuario [6]; E-mail, que resume las discusiones de correo electrónico o muestra un correo electrónico en dispositivos móviles con un tamaño reducido de pantalla [7]; hilos de e-mail (mensaje inicial del e-mail y las respuestas subsecuentes a éste [8]; etiquetado de grupos en agrupamiento de documentos web [9].

Además, teniendo en cuenta la forma cómo se obtiene el resumen [10], estos pueden ser abstractivos o extractivos. En los abstractivos las oraciones del resumen no necesariamente están en el documento original y se enfocan en la coherencia del resumen, para lo cual usan herramientas de análisis lingüístico que requieren de memoria y capacidad de procesamiento debido a la complejidad de la tarea. De otra parte, los extractivos para formar el resumen seleccionan el conjunto de oraciones más relevantes del texto original, en general incluyendo las oraciones en el resumen en el mismo el orden en el que estaban en el documento original. En la actualidad las técnicas extractivas son más usadas debido a su simplicidad y tiempo de cómputo.

En la generación automática de resúmenes extractivos de múltiples documentos, se encuentra gran variedad de métodos [10] como: *conectividad de textos* [11], combina roles retóricos y análisis semántico basado en corpus. El enfoque es capaz de obtener las relaciones semánticas y retóricas entre oraciones para combinarlas y producir resúmenes coherentes; *reducción algebraica* [12, 13], a través de Análisis Semántico Latente, valor

singular de descomposición y factorización matricial no negativa, para encontrar las oraciones más significativas que representan los documentos, realizando descomposición de matrices no negativas o descomposición de valores singulares; *técnicas de aprendizaje de máquina* [14], que combina tres modelos basados en máxima entropía, un clasificador ingenuo de Bayes y uno de máquinas de soporte vectorial para seleccionar las oraciones que harán parte del resumen, en [15], utilizan vectores de soporte de regresión para estimar la importancia de una oración de un conjunto de documentos; *grafos* [2, 16-18], representando el conjunto de oraciones como un grafo no dirigido, en el cual los nodos son las oraciones y los arcos la relación de similitud entre pares de oraciones; *metaheurísticas* (evolución diferencial, algoritmos genéticos y algoritmos Meméticos) [19-21], que buscan optimizar una función objetivo aproximada que mide la calidad del resumen para encontrar las oraciones que harán parte del resumen. Uno de estos métodos, MA-MultiSumm [20], es un algoritmo memético que combina operadores genéticos (CHC) con una búsqueda local codiciosa, el cual está ubicado entre los primeros métodos en el estado del arte. Sin embargo, se requiere de un esfuerzo adicional para la configuración de los operadores de selección, cruce, mutación, reemplazo y búsqueda local; *híbridos* [22], basado en un enfoque de aprendizaje híbrido para extraer oraciones de múltiples documentos y generar el resumen. El proceso realiza la hibridación de dos pasos: un modelo generativo para el descubrimiento de temas jerárquicos y un modelo de regresión para la inferencia, también se utiliza un modelo jerárquico de temas para obtener las características latentes de las oraciones y calcular el puntaje de estas en el grupo de documentos, igualmente entrenan un modelo de regresión para puntuar las oraciones de nuevos documentos para generar el resumen; [23], es un algoritmo genético paralelo híbrido cuyo enfoque es obtener conceptos de palabras basados en HowNet[1][24] y utilizar el concepto como característica, en lugar de las palabras, se crea un modelo conceptual de espacio vectorial y mediante el algoritmo k-means mejorado se genera el resumen.

Existen metaheurísticas menos complejas que no requieren de la configuración de operadores y que han contribuido en la solución de problemas continuos y discretos mostrando muy buenos resultados, como, la Mejor Búsqueda Armónica Global (Global Best Harmony Search, GBHS) [25], pero hasta el momento no han sido usadas para resolver el problema de la generación automática de resúmenes para múltiples documentos.

De otro lado, los algoritmos basados en grafos definen la importancia de una oración haciendo un filtro de las oraciones más relevantes de la colección de documentos. LexRank [2], es un algoritmo basado en grafos, sencillo de implementar, que define la importancia de una oración basado en su centralidad, la cual se calcula usando pageRank [26], este último, un algoritmo muy popular por sus excelentes resultados en áreas como la búsqueda web (parte de google search) y el análisis de redes sociales.

Hasta el momento no se han propuesto algoritmos que hibriden metaheurísticas con grafos, por lo tanto, en este proyecto se propone un algoritmo híbrido de generación automática de resúmenes extractivos para múltiples documentos basado en la metaheurística GBHS con búsqueda local y el algoritmo de grafos LexRank, aprovechando de esta forma, la fortaleza de explotación y exploración de GBHS y la definición del grado de centralidad de las oraciones con respecto a la colección de documentos (grafo) por medio de LexRank.

[1] HowNet es una base de conocimientos de sentido común en línea que revela las relaciones interconceptuales y las relaciones entre los atributos de los conceptos como connotación en los léxicos de los chinos y sus equivalentes en inglés.

Teniendo en cuenta lo anterior, para este proyecto se plantea la siguiente pregunta de investigación: ¿Qué método híbrido entre los algoritmos de GBHS y el algoritmo de LexRank permite obtener resúmenes extractivos genéricos de múltiples documentos con resultados similares o de mayor calidad a los establecidos por métodos híbridos del estado del arte?

1.2 APORTES

Con la definición del algoritmo propuesto, se aporta nuevo conocimiento en el área de generación automática de resúmenes de múltiples documentos, especialmente en algoritmos híbridos que combinan estrategias metaheurísticas con grafos. Este conocimiento es de gran importancia para la comunidad de Procesamiento de Lenguaje Natural y de Recuperación de Información.

Con el desarrollo de este proyecto se busca contribuir a la línea de investigación de Gestión de la Información y Sistemas Inteligentes del Grupo de I+D en Tecnologías de la Información (GTI) de la Universidad del Cauca, específicamente en el desarrollo de una nueva solución informática híbrida que permita soportar la generación automática de resúmenes de múltiples documentos.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Proponer un algoritmo híbrido para la generación automática de resúmenes extractivos de múltiples documentos basado en la metaheurística de la Mejor Búsqueda Armónica Global y el algoritmo basado en grafos LexRank, buscando obtener resultados similares o superiores a los reportados en el estado del arte.

1.3.2 OBJETIVOS ESPECÍFICOS

- Definir un algoritmo híbrido de generación automática de resúmenes extractivos y genéricos para múltiples documentos basado en la metaheurística de la mejor búsqueda armónica global y el algoritmo LexRank.

- Evaluar la calidad promedio de los resúmenes generados por el algoritmo híbrido propuesto sobre colecciones de noticias como DUC2005 y DUC2006, comparándolo con los algoritmos del estado del arte por medio de métricas ROUGE.

- Desarrollar una aplicación web que permita la generación automática de resúmenes de múltiples documentos ingresados por un usuario en formato textual y evaluar la satisfacción del usuario con los resultados obtenidos.

1.4 RESULTADOS OBTENIDOS

- Monografía del trabajo de grado, en la que se presenta el contexto teórico y el estado del arte necesarios en el desarrollo del proyecto, los algoritmos base utilizados en el algoritmo híbrido, el proceso de construcción del algoritmo propuesto, que incluye el

diseño de la función objetivo, configuraciones de la función objetivo, configuración de parámetros, y afinación de parámetros y diseño y afinamiento de la función objetivo; la descripción del algoritmo híbrido propuesto *LexRank-GBHS* y la evaluación de calidad de los resúmenes generados por este y su comparación con otros algoritmos del estado del arte. Además, la aplicación web que implementa el algoritmo híbrido propuesto. También se describe la aplicación web desarrollada que implementa el algoritmo híbrido propuesto y la evaluación de satisfacción de usuario de la aplicación. Por último las conclusiones, recomendaciones y el trabajo futuro que el Grupo de I+D en tecnologías de La Información (GTI) desarrollará con base en esta investigación.

- Código del algoritmo propuesto, junto con la especificación de su lógica y componentes.

- Aplicación web, que implementa el algoritmo híbrido propuesto junto a su arquitectura, diagrama de clases e interfaces.

- Pasantía internacional en el SECABA LAB (Quality Evaluation & Information Retrieval), laboratorio del Grupo de Investigación SCI2S – Soft Computing and Intelligent Information Systems, en la Escuela Técnica Superior de Ingeniería Informática y de Telecomunicaciones de la Universidad de Granada (España), donde se realizaron las siguientes actividades:

  - Definición de nuevas versiones del algoritmo híbrido entre la Metaheurística de la Mejor Búsqueda Armónica Global y el algoritmo LexRank.
  - Conocimiento de los proyectos de I+D desarrollados por el SECABA Lab y los trabajos futuros.
  - Avances en un artículo con la propuesta de dos versiones del algoritmo híbrido entre la Mejor Búsqueda Armónica Global y LexRank, para fue presentado en el “16th Mexican International Conference on Artificial Intelligence. MICAI”.
  - Presentación del proyecto de investigación ante miembros del SECABA Lab, con el objetivo de recibir retroalimentación de la experiencia de nuestro grupo y realizar mejoras.

- Aceptación del artículo “Automatic generation of multi-document summaries based on the Global-Best Harmony Search meta-heuristic and the LexRank graph-based algorithm”, que fue presentado en la 16th Mexican International Conference on Artificial Intelligence. En espera de ser publicado en revista LNCS (SJR), revista tipo A2, Publindex de COLCIENCIAS. El artículo propone dos algoritmos híbridos entre la metaheurística de la Mejor Búsqueda Armónica Global y el algoritmo basado en grafos LexRank para la generación automática de resúmenes de múltiples documentos, que incluye la representación de la solución, la configuración de sus parámetros, la función objetivo, y la adaptación de cada uno de los algoritmos base en los dos algoritmos híbridos propuestos. Además, se presentan los resultados obtenidos al evaluar los resúmenes generados y compararlos con otros algoritmos del estado del arte.

- Participación en el evento Neiva Knowledge Time 2018-Investigación para la Formación, organizado por el Sistema de Investigación y Desarrollo Tecnológico del SENA, Regional Huila, con el artículo corto titulado “Aplicación web para la generación automática de resúmenes extractivos de múltiples documentos”, que será publicado por

el SENA en un libro con la memoria de los artículos del evento. En la presentación del artículo se realizó una demostración de la aplicación web para la generación de resúmenes de automáticos de múltiples documentos, y se presentaron los resultados de la evaluación de la aplicación web, que permitió medir la satisfacción del usuario teniendo en cuenta características de calidad como: adecuación funcional, eficiencia de desempeño y usabilidad; obteniendo muy buenos resultados ya que en todas las preguntas los encuestados estuvieron de acuerdo.

Capítulo 2

2 CONTEXTO TEÓRICO Y ESTADO DEL ARTE

En este capítulo se presentan los conceptos más importantes del área de investigación de generación de resúmenes, se realiza una revisión de los trabajos más relevantes en esta área y las medidas de calidad de resúmenes automáticos aceptadas por la comunidad científica. También, la representación de documentos en el modelo espacio vectorial y medidas asociadas para ponderación de términos, y la medida de similitud de cosenos. Además la descripción general de los algoritmos base LexRank [2] y la Metaheurística de la Mejor Búsqueda Armónica Global [1], y el procedimiento de búsqueda local codiciosa.

2.1 GENERACIÓN AUTOMÁTICA DE RESÚMENES DE TEXTO

La generación automática de resúmenes de textos es una tarea del procesamiento de lenguaje natural, que tiene por objetivo identificar el contenido más significativo de uno o múltiples documentos para ser extraído en un resumen de tamaño corto, que represente el contenido del documento conservando su información importante. En la literatura existen diversas formas de clasificar los resúmenes [27], según:

- Número de documentos: El resumen puede ser obtenido de uno o múltiples documentos.

- Propósito: Indicativo, describe brevemente la idea principal del documento; Informativo, busca sustituir el documento brindando una versión abreviada del contenido; Crítico, recoge el punto de vista del autor del resumen.

- Audiencia a la que va dirigido: Genérico, da igual importancia a los temas principales del documento y están destinados a un amplio grupo de usuarios; Basados en consultas, el resumen recoge la información más relevante según la consulta realizada por un usuario; Enfocados en el usuario o en tópicos, adaptados al usuario, son elaborados de acuerdo al perfil del lector.

- Profundidad de procesamiento: Estrategias poco profundas o superficiales, no van más allá del nivel sintáctico y solo utilizan características superficiales como frecuencia de términos, ubicación de oraciones, entre otras; Estrategias profundas, utilizan técnicas avanzadas de procesamiento de lenguaje para modelar las entidades que aparecen en el texto y sus relaciones.

- La forma como el resumen es extraído: Técnicas abstractivas, usan conocimientos lingüísticos para generar el resumen mediante el análisis de la gramática y la semántica; Técnicas extractivas, extraen las oraciones más relevantes del documento origen para ser incluidas en el resumen.

Con respecto a las demás taxonomías, se puede generar el resumen de acuerdo al género de los documentos, como artículos científicos, blogs, noticias entre otros; con respecto a la cantidad de documentos, pueden ser para un documento o para múltiples documentos y con respecto al lenguaje pueden ser monolenguaje o multilenguaje.

El algoritmo propuesto en el presente trabajo de investigación, el resumen se genera a partir de técnicas extractivas, para múltiples documentos y monolenguaje.

2.1.1.1 Métodos para la generación automática de resúmenes de múltiples documentos

Estos métodos se pueden clasificar de acuerdo a la técnica usada (estado del arte [28]): conectividad de textos, reducción algebraica, aprendizaje de máquina, grafos, metaheurísticas, e híbridos. A continuación se presenta una actualización de estas referencias.

2.1.1.2 Conectividad de Textos

C. Yan-Min et al. [28] proponen un modelo basado en el concepto de cadena léxica o secuencia de palabras relacionadas en el texto, que abarca distancias cortas (palabras u oraciones) o largas (todo el texto). El modelo realiza la segmentación del texto, identifica las cadenas léxicas y sus relaciones en términos de distancia WordNet[2][29]. Las cadenas son calificadas por su longitud y homogeneidad, seleccionando las cadenas léxicas más fuertes, luego de cada una de estas cadenas se seleccionan oraciones para crear el resumen de cada documento. Para evitar la redundancia de oraciones se utiliza un umbral para las similitudes entre las oraciones del resumen.

Atkinson et al. [11], plantean un modelo que combina roles retóricos y análisis semántico basado en corpus, para el resumen de múltiples documentos en la web. El enfoque utiliza una fuente que contiene varios documentos de noticias de la web sobre un tema de búsqueda específico y aplica la extracción de información, el análisis semántico basado en corpus y los métodos estocásticos de NLP para generar un resumen. El modelo se compone de cinco componentes principales: limpieza y preprocesamiento del texto; identificación de roles: tarea que se hace con un clasificador de campos aleatorios condicionales; extracción de oraciones: se extraen de acuerdo a su relevancia donde se utiliza un modelo de distribución para medir la ocurrencia de las palabras, después ordenan las oraciones teniendo en cuenta el puntaje obtenido y las agrupan de acuerdo a los roles retóricos, ordenadas en cada grupo de acuerdo a la similitud; clasificación y generación de resumen: las oraciones que serán parte del resumen se seleccionan teniendo en cuenta la preferencia de los roles retóricos definidos.

2.1.1.3 Reducción Algebraica

Estos métodos se basan en el análisis semántico latente (Latent Semantic Analysis, LSA), que es una técnica matemática que permite extraer e inferir relaciones contextuales entre las palabras de una colección de textos, para esto se basan en descomposición de matrices no negativas o en la descomposición de valores singulares.

[2] WordNet es una base de datos léxica del Idioma inglés que agrupa palabras en conjuntos de sinónimos llamados synsets, proporcionando definiciones cortas y generales y almacenando las relaciones semánticas entre los conjuntos de sinónimos

Wang et al. [30] en 2008, proponen un trabajo basado en el análisis semántico a nivel de oraciones y la factorización de matriz simétrica no negativa (NMF). Primero se calcula las similitudes oración-oración utilizando el análisis semántico y la matriz de similitudes. Luego la factorización de la matriz simétrica es usada para agrupar oraciones dentro de grupos. Por último, las oraciones más informativas desde cada grupo son seleccionadas para formar el resumen.

Xiong y Luo [13] en 2014, plantean un método basado en LSA que evalúa un conjunto de oraciones de resumen en función de su similitud de proyección con la de las oraciones completas establecidas; mediante la descomposición de valores singulares (SVD) se crea una matriz de valores singulares que contiene el conjunto total de oraciones y a partir de ahí se genera otra matriz de valores singulares con la representación de la similitud de proyección de cada una de las oraciones; para la selección de oraciones que harán parte del resumen, se realiza un proceso iterativo de acuerdo al número de oraciones establecido, seleccionándolas las de menor costo en la función.

2.1.1.4 Aprendizaje de Máquina

Ravindra et al. [28], proponen una técnica de agrupación de oraciones usando medidas de entropía para la generación automática de resúmenes, mediante la asignación de puntajes, basados en la entropía de un conjunto reducido de oraciones, obtenidas utilizando una representación gráfica de la similitud de oraciones.

Luego, Ouyang et al. [15], proponen un método que utiliza vectores soporte de regresión para estimar la importancia de una oración de un conjunto de documentos. Los modelos de regresión utilizan funciones continuas y estiman directamente la importancia de las oraciones. Este método clasifica y extrae oraciones de acuerdo con un conjunto de características predefinidas y una función de puntuación compuesta.

Fattah [14] en 2014, propone un algoritmo de aprendizaje de máquina híbrido para mejorar la selección de contenidos en la generación de resúmenes automáticos. Las ocho características se combinan para construir resúmenes de texto, mediante modelos basados en máxima entropía, un clasificador ingenuo de Bayes y uno de máquinas de soporte vectorial. Para producir el resumen final, los tres modelos se combinan en un modelo híbrido, buscando la probabilidad máxima de los tres métodos para clasificar una oración como que pertenece o no al resumen.

2.1.1.5 Grafos

Mihalcea y Tarau [28], plantean un modelo basado en grafos mediante el algoritmo TextRank, para la extracción de oraciones. Un vértice es una unidad léxica extraída del texto y los bordes definen las relaciones entre las unidades léxicas. TextRank hace uso del algoritmo PageRank [31] para seleccionar las oraciones que van a quedar en el grafo, midiendo la importancia de cualquier página web en internet en función de los enlaces que dicha página recibe.

Erkan y Radev [32], proponen un algoritmo con umbral (LexRank) basado en el concepto de prestigio[3] en las redes sociales, estas se representan comúnmente en forma de grafos,

[3]Prestigio y Centralidad en esta propuesta, representan el mismo concepto con la diferencia de que el primero se define usualmente para grafos dirigidos, mientras el segundo se define para grafos no dirigidos.

donde los nodos representan las entidades y los enlaces representan las relaciones entre los nodos. Un conjunto de documentos puede verse como una red de oraciones relacionadas, algunas son más similares entre sí, mientras otras pueden compartir poca información con el resto de las oraciones. Si una oración es muy similar a muchas de las otras oraciones, esta se considera como la más sobresaliente o representativa de un tema.

Luego, Zhang et al. [16], proponen el método GSPSummary que obtiene los subtemas más representativos de la colección de documentos, iniciando el proceso mediante la creación de un grafo para representar toda la colección de documentos, y mediante GSPRank se seleccionan las oraciones más centrales para obtener los subtemas más importantes en el grafo global de forma iterativa; finalmente el resumen se crea con las oraciones más significativas de los diferentes subtemas.

Ferreira et al. [17], plantean un algoritmo de agrupación, el cual convierten el texto de los documentos en un grafo mediante la similitud de cosenos entre las oraciones. Luego hacen uso del algoritmo Textrank para calcular los puntajes a cada una de las oraciones, obteniendo el vértice con mayor puntaje llamado líder principal. Este puntaje es comparado con todos los vértices del grafo que cumplan con un umbral para encontrar los demás líderes. Finalmente se construye un grafo para cada líder mediante el uso del algoritmo Dijkstra que utiliza la ruta más corta entre cada uno de los vértices y el líder, donde se obtienen *n* grafos, uno por cada vértice líder y por último se seleccionan las oraciones con mayor puntaje de cada grafo para formar parte del resumen.

John and M. Wilscy. [33], proponen un método que utiliza el algoritmo de cubrimiento de vértices, cuyo objetivo es encontrar la menor cantidad de vértices que incluyan todas las aristas del grafo. Inicialmente se construye un grafo no dirigido para todo el conjunto de documentos, donde los vértices representan cada una de las oraciones y las aristas las similitudes entre ellas. Para el cálculo de la similitud se utiliza la combinación de la medida de similitud de cosenos y la distancia normalizada de Google. A este grafo de similitudes se le aplica el enfoque de cobertura de vértices mediante un proceso iterativo que permite obtener un grafo reducido. Este proceso inicia seleccionando la oración de mayor peso, luego para obtener las demás oraciones que formarán parte del resumen se vuelven a calcular los pesos de todas las oraciones del grafo sin tener en cuenta las ya seleccionadas. Además valida que se cumplan las restricciones de longitud del resumen, que la oración no haya sido incluida previamente y que su peso sea superior a un umbral. Este proceso se repite hasta completar la longitud del resumen.

Más recientemente, Tohalino et al. [34], plantea un método basado en mediciones de red complejas para el resumen de varios documentos mediante el modelado de un conjunto de documentos como una red multicapa, donde los nodos representan las oraciones y los ejes representan la similitud en función de la similitud de coseno entre dos oraciones. Se realiza una distinción entre la similitud entre oraciones de diferentes documentos frente a la similitud de oraciones que pertenecen al mismo documento. Las oraciones que van a ser parte del resumen, se seleccionan de acuerdo con algunas mediciones y estrategias de selección.

2.1.1.6 Metaheurísticas

Liu et al. [19], plantean un modelo para generar un resumen maximizando la cobertura de temas y minimizando la redundancia, por medio de un algoritmo genético, cuya función

objetivo es la combinación lineal del peso de cada oración (por medio de varios enfoques) menos la redundancia (similitud entre las oraciones). Luego un modelo también basado en la máxima cobertura y mínima redundancia, pero usando un algoritmo de *evolución diferencial binario*, es propuesto por Alguliev et al. [35], donde la función objetivo es una combinación ponderada de la cobertura de contenido y disminución de la redundancia. Mendoza et al. [20], también plantean una función objetivo como la combinación lineal de los factores de cobertura y redundancia, pero basado en el algoritmo evolutivo CHC y la búsqueda local codiciosa.

Más recientemente, Ansamma et al. [36], proponen un método de optimización multicriterio, que utiliza tres funciones objetivo. La primera función objetivo basada en centroides, combina los factores de cobertura y diversidad; la segunda función objetivo se basa en análisis semántico latente (LSA) para encontrar un resumen con la máxima cobertura semántica; y la tercera función objetivo se basa en características semánticas mediante la Factorización de Matriz no Negativa (FNM). Usando el enfoque basado en la población, mediante un algoritmo genético, se generan tres resúmenes candidatos óptimos al maximizar las tres funciones objetivo de forma independiente. Al final se genera un resumen del tamaño requerido, seleccionando las oraciones que son comunes en más de un resumen candidato y, si no se cumple el tamaño requerido, se seleccionan las oraciones mejor calificadas de las oraciones restantes de los tres resúmenes candidato.

2.1.1.7 Hibridación

La generación de resúmenes se ha abordado por Celikyilmaz y Hakkani-Tur [28] como un problema de predicción basado en un modelo híbrido de dos pasos: un modelo generativo para el descubrimiento de patrones y un modelo de regresión para la inferencia. El modelo calcula puntuaciones para las oraciones en grupos de documentos basados en sus características latentes utilizando un modelo jerárquico del tema. Usando las puntuaciones, entrena un modelo de regresión basado en las características léxicas y estructurales de las oraciones, y usan el modelo para dar una puntuación a las oraciones de los nuevos documentos para generar el resumen.

Binwahlan et al. [37], proponen un modelo híbrido que combina tres métodos basados en: diversidad, enjambres (PSO) y lógica difusa. El método basado en diversidad se enfoca en reducir los problemas de redundancia, creando grupos de oraciones y ordenándolas en un árbol binario de acuerdo a los puntajes, las oraciones que harán parte del resumen se seleccionan aplicando el método de importancia marginal máxima. El método basado en enjambres (PSO) binario, es utilizado para optimizar el peso de cada característica de la función objetivo; el peso de cada característica se obtiene de acuerdo a la posición de la partícula que es representada como una cadena de bits, después se seleccionan las características cuyo valor sea uno y se calcula el puntaje de cada oración, donde las oraciones con el mayor puntaje se incluyen en el resumen; y por último, el método basado en enjambres y lógica difusa calcula el puntaje de las oraciones a través de un sistema de inferencia, que utiliza los pesos encontrados con PSO para obtener los puntajes finales de las oraciones, después se ordenan las oraciones de acuerdo al puntaje obtenido y de ahí se seleccionan las oraciones que harán parte del resumen. Al final, con los resúmenes generados por los tres métodos anteriores, el modelo híbrido mediante otro procedimiento selecciona las oraciones que harán parte del resumen final.

Meng y Xinlai [23], proponen un algoritmo genético híbrido para hacer agrupamiento de las oraciones. En este enfoque se obtienen los conceptos de las oraciones usando el método HowNet[24], en lugar de las palabras para la representación en el modelo espacio vectorial. El algoritmo genético inicializa dos poblaciones, donde cada gen del cromosoma es una oración centroide, el algoritmo k-means se utiliza para agrupar las oraciones para todos los individuos de la población, luego el algoritmo evoluciona y el resumen se obtiene de la mejor solución de cada población que está compuesta por oraciones de las diferentes agrupaciones.

2.1.2 Métodos de evaluación de la calidad de los resúmenes

La evaluación en los sistemas de generación automática de resúmenes, es una tarea compleja debido a las irregularidades de los lenguajes (humanos), y a que la concepción de lo que es un buen resumen varía entre las personas. En la actualidad no existe un esquema de evaluación definitivo, sin embargo, se han construido diversas herramientas que permiten su automatización haciendo uso de las medidas estándar de recuperación de información como la precisión, recuerdo y medida F [38]. Los métodos para evaluar los sistemas de generación automática de resúmenes de texto, se dividen en dos grandes categorías intrínsecas y extrínsecas [39].

2.1.2.1 Evaluación intrínseca

Este tipo de evaluación mide la calidad del resumen sin tener en cuenta la audiencia a quien va dirigido el resumen, y con frecuencia se lleva a cabo mediante la comparación con un conjunto de resúmenes ideales[4], que pueden ser generados por evaluadores humanos o sistemas de referencia. La mayoría de los esquemas de evaluación de resúmenes son intrínsecos.

La evaluación intrínseca se centra principalmente en la coherencia y la capacidad informativa de los resúmenes, midiendo de ese modo solamente la calidad del producto (resumen) [39]. De esta forma, surgen algunas medidas, utilizadas comúnmente en este tipo de evaluación, como la *precisión* y el *recuerdo* [40]. El recuerdo es la razón del número de oraciones comunes entre el resumen generado y el de referencia, sobre el número total de las oraciones del resumen generado. Análogamente, la precisión se define como el número de oraciones en el resumen generado que están presentes en el resumen de referencia. Precisión y la recuperación son medidas estándar para la recuperación de información y, a menudo se combinan en la denominada medida F.

En la actualidad el paquete de medidas de ROUGE [41] se ha utilizado 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.

2.1.2.2 Evaluación extrínseca

Este tipo de evaluación se enfoca hacia el usuario al que va dirigido el resumen, dándole más importancia a la utilidad que este puede tener sobre ese usuario que la calidad como resumen. Por lo tanto, mide la eficacia y aceptabilidad de los resúmenes generados en

[4] 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

alguna tarea, por ejemplo, la evaluación de la pertinencia o la comprensión de la lectura. Además, si el resumen contiene algún tipo de instrucciones, es posible medir hasta qué punto es posible seguir las instrucciones y el resultado del mismo. Otras posibles tareas medibles son la recopilación de información en una gran colección de documentos, el esfuerzo y el tiempo necesario para enviar a editar el resumen generado por una máquina con un propósito específico, o el impacto del sistema generación de resúmenes en un sistema del que forma parte, por ejemplo comentarios relevantes (ampliación de consultas) en un motor de búsqueda o un sistema de pregunta-respuesta [39].

2.1.2.3 Evaluación ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) es un paquete de software que utiliza un conjunto de medidas para determinar automáticamente la calidad de un resumen comparándolo con otros resúmenes ideales creados por humanos. Las medidas cuentan el número de unidades superpuestas, tales como n-gramas de palabras, y pares de palabras, entre el resumen generado por computador y los resúmenes ideales [41]. Así pues, la evaluación se lleva a cabo a través del conteo de unidades coincidentes entre los resúmenes.

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.

- **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. En la Ecuación (2.1) se presenta el cálculo de esta medida.

$$\text{ROUGE} - \text{N} = \frac{\sum_{S \in \{ResúmenesDeReferencia\}a} \sum_{grama_n \in S} Conteo_{Coincidencia}(grama_n)}{\sum_{S \in \{ResúmenesDeReferencia\}} \sum_{grama_n \in S} Conteo(grama_n)} \qquad (2.1)$$

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-S**

ROUGE-S mide la superposición de saltos de bigramas (bigramas-skip) entre un resumen candidato y un conjunto de resúmenes de referencia. Un bigrama-skip se refiere a un par de palabras, en el orden en que están en la oración, permitiendo saltos arbitrariamente. 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 (2.2),(2.3) y (2.4).

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \quad (2.2)$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \quad (2.3)$$

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}} \quad (2.4)$$

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[5]. Considerando el siguiente ejemplo:

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

Se infiere que cada oración tiene 6 bigramas-skip[6]. 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**

Un problema que presenta 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. ROUGE-SU evita este problema tomando como punto de partida ROUGE-S, pero incluyendo 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.

2.1.2.4 Colección de documentos de evaluación

La Conferencia de Entendimiento del Documento (DUC por sus siglas en inglés, Document Understanding Conference), ofrece a la comunidad académica un conjunto de documentos relacionados (noticias) a resumir y los resúmenes "ideales" o modelos creados por varios expertos humanos, con el objetivo de permitir evaluar y comparar los resultados obtenidos por sistemas de generación automática de resúmenes. DUC tiene una gran aceptación por parte de la comunidad académico-científica, ya que su conjunto de documentos es frecuentemente utilizado como conjunto de referencia por diversos estudios.

2.2 REPRESENTACIÓN DE LOS DOCUMENTOS

A continuación, se presenta el modelo de representación de oraciones de los documentos en el espacio vectorial, las técnicas de ponderación de términos en una oración, las medidas de similitud, y la representación de documentos mediante matrices.

[5] 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)!)
[6] C(4,2) = (4!/2!*2!) = 6

2.2.1 Modelo de espacio vectorial

La representación de un conjunto de documentos como vectores en un espacio vectorial, se conoce como el modelo de espacio vectorial [42]. Un vector documento $\vec{d}_j$ identifica en qué grado se satisface cada una de las m características del conjunto. Las características (componentes) del vector son un valor numérico. El concepto de característica suele definir la ocurrencia de determinadas palabras o términos en el documento, aunque se pueden considerar otros aspectos (oraciones, párrafos) [43].

Un conjunto de documentos se puede representar como, $\mathrm{D} = \{d_1, d_2, \dots, d_k\}$, donde k es el número de documentos. Cada documento d_j contiene un conjunto de oraciones, $d_j = \{s_1, s_2, \dots, s_p\}$, donde p es el número de oraciones en d_j. De esta forma la colección de documentos se representa como el conjunto de todas las oraciones de la colección, es decir, $D = \{s_1, \dots, s_j, \dots, s_n\}$, donde s_j denota la oración j-ésima en D, y n es el número de oraciones en la colección, $s_j \in D$ si y solo si $s_i \in d_j \in D$. Sea $s_i = \{t_{\mathrm{i}1}, t_{\mathrm{i}2}, \dots, t_{ik}, .. t_{im}\}$, donde t_{ik} es el *k*-ésimo término de la oración s_i, y m es el número total de términos en la oración.

Por lo anterior, considere el espacio vectorial de una oración $\vec{s}_j$, donde cada una es representada por uno o más términos t_i; los términos pueden ser ponderados (w_{ij}) o no ponderados de acuerdo a su importancia. Un espacio típico tridimensional se muestra en la Figura 2.1 (Adaptado de [44]) , donde cada oración es identificada por tres términos distintos, éste espacio puede ser extendido a m dimensiones cuando m términos diferentes están presentes, en este caso, cada oración $\vec{s}_j$ es representada por un *m*-vector de la siguiente forma [44], y de acuerdo a la Ecuación (2.5).

$$\vec{s}_{jm} = (w_{j1}, w_{j2}, \dots, w_{jm}) \tag{2.5}$$

Donde $\boldsymbol{w_{ij}}$ representa el peso del término *i-ésimo* de la oración $\boldsymbol{j}$.

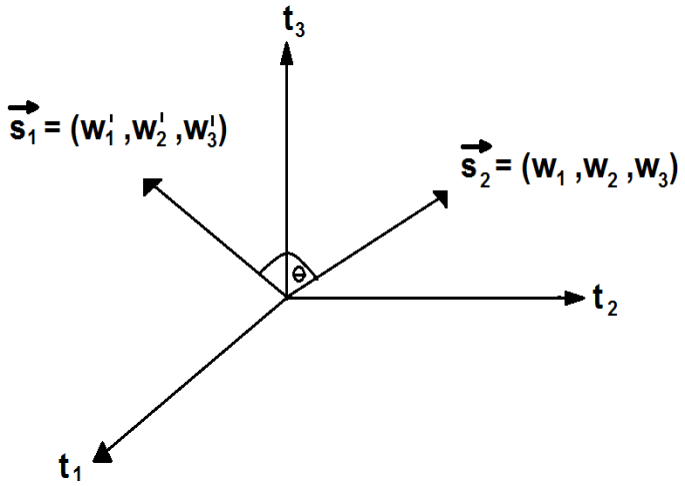

Figura 2.1. Representación oraciones espacio vectorial tridimensional

Cada término $\boldsymbol{t_i}$ se convierte en una dimensión independiente en el espacio dimensional del vector, la mayoría de los vectores operan en los cuadrantes positivos, dado que a ningún término se le asigna un valor negativo [45].

2.2.2 Técnicas de ponderación de términos

En esta sección, se presentan algunas de las técnicas más utilizadas en la generación automática de resúmenes para la ponderación de términos.

2.2.2.1 Booleana

Este esquema binario concede la misma relevancia a cada término que aparece en el documento. Puede ser usado cuando no se considera importante el número de apariciones del término. 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 (2.6) [46].

$$w_{ij} = \begin{cases} 1, & si\ el\ término\ j\ está\ en\ el\ texto\ i \\ 0, & en\ caso\ contrario \end{cases} \tag{2.6}$$

2.2.2.2 Frecuencia del término

Este esquema cuenta cuántas veces se utiliza el término en un documento. Si la frecuencia del término (TF, Term Frequent) en el documento es mayor, aumenta la probabilidad de que sea relevante para el documento. Como se observa en la Ecuación (2.7), en este esquema la importancia de un término radica en la cantidad de veces que aparezca en el texto.

$$w_{ij} = f_{ij} \tag{2.7}$$

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

La desventaja de la aplicación 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 [47]. De esta forma, el cálculo de frecuencia más comúnmente utilizado, es el que se observa en la Ecuación (2.8).

$$w_{ij} = \frac{f_{ij}}{MáxFreq_i} \tag{2.8}$$

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

2.2.2.3 Frecuencia inversa de un término

Es un mecanismo para atenuar el efecto de los términos demasiado frecuentes en el texto, la idea es reducir el peso TF de un término con un factor que crece con su frecuencia de aparición. La frecuencia inversa de documento (IDF, Inverse document frequency), hace referencia a la frecuencia de un término dentro de una colección de textos [38]. La Ecuación (2.9) expresa esta definición.

$$w_{ij} = \log \frac{N}{n_j} \tag{2.9}$$

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

2.2.2.4 Frecuencia relativa de un término

Este esquema combina la definición de los métodos descritos en las secciones 2.2.2.2 y 2.2.2.3, para producir un peso compuesto [38]. El ponderado TF-IDF asigna al término *i* un peso en el texto *j*, como se muestra en la Ecuación (2.10).

$$w_{ij} = TF_{ij} \times IDF_j \tag{2.10}$$

Donde TF_{ij} es la frecuencia del término j en el texto i e IDF_j es la frecuencia inversa del término j, como se presentó en las Ecuaciones (2.8) y (2.9), respectivamente [38]. Reemplazando se tiene la Ecuación (2.11).

$$w_{ij} = \frac{f_{ij}}{MáxFreq_i} \times \log \frac{N}{n_j} \tag{2.11}$$

De esta manera, TF-IDF asigna al término *j* un peso en el texto *i* que es:

- Más grande cuando *j* está presente muchas veces dentro de un pequeño número de textos.
- Pequeño cuando el término aparece pocas veces en un texto, o aparece en muchos textos (se toma como una señal de baja relevancia).
- Más pequeño cuando el término aparece en casi todos los documentos.

2.2.3 Medidas de Similitud

Para calcular la similitud entre las oraciones, cada una de ellas debe presentarse como un vector en el modelo de espacio vectorial [48]. Teniendo las oraciones como vectores es posible calcular su semejanza, por medidas de la similitud de coseno u otra medidas como el producto punto [44].

2.2.3.1 Similitud de Coseno

Sean $\vec{s_i}, \vec{s_j}$ dos *m*-vectores diferentes del vector cero, donde el peso de los términos se calcula con TF-ISF, Ver ecuación (2.11). Entonces el ángulo φ o la similitud entre $\vec{s_i}$ y $\vec{s_j}$ se define en el intervalo [0, 1] de acuerdo a la ecuación (2.12) [48, 49]:

$$simcos(\vec{s_i}, \vec{s_j}) = \frac{\sum_{k=1}^{n} w_{ik} \; w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2 \quad \sum_{k=1}^{n} w_{jk}^2}} \tag{2.12}$$

2.2.4 Representación de documentos por medio de matrices

La representación en el espacio multidimensional del conjunto de vectores de oraciones y la similitud de las mismas, se puede hacer con la Matriz de Términos por Oraciones y la Matriz de similitud de Cosenos.

2.2.4.1 Matriz de Términos por Oraciones

La matriz TF-ISF es una matriz dispersa de mxn de pesos dispuestos en m términos (filas) y n oraciones (columnas), el elemento ij de la Matriz es denotado por w_{ij}, que corresponde al peso del término i en la oración j, este peso es calculado de acuerdo a la ecuación (2.11); la Matriz se representa por la siguiente ecuación (2.13) [49, 50]:

$$Matriz\ TF-ISF = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1j} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2j} & \dots & w_{2n} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ w_{i1} & w_{i2} & \dots & w_{ij} & \dots & w_{in} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ w_{m1} & w_{m2} & \dots & w_{mj} & \dots & w_{mn} \end{pmatrix} \quad (2.13)$$

2.2.4.2 Matriz de Similitud de Cosenos

La matriz de similitud de cosenos es una matriz cuadrada de nxn, donde n son las oraciones, el elemento ij de la matriz denotado por $\boldsymbol{a_{ij}}$, representa la similitud entre las oraciones, calculada con la ecuación (2.12), lo anterior se reprensenta de acuerdo a la Ecuación (2.14) [49]:

$$MatrizDeSimilitud = \begin{pmatrix} 1_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & 1_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ a_{i1} & a_{i2} & \dots & 1_{ij} & \dots & a_{in} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nj} & \dots & 1_{nn} \end{pmatrix} \quad (2.14)$$

El elemento a_{ij} es la similitud entre la oración $\vec{s}_i$ y $\vec{s}_j$ que aparece en la fila i y la columna j, los elementos de diagonal principal son iguales a 1, ya que es el cálculo de cada oración con ella misma, $\vec{s}_i$ con $\vec{s}_i$ cuando $i = j$.

2.2.5 Vector Centroide

El Vector Centroide representa el conjunto de oraciones de la colección de documentos como una sola oración, donde cada componente es el peso promedio del término i. Para su cálculo se utiliza la Matriz TF-ISF Ecuación (2.13) [42, 51]. El término i (vector fila i) de la Matriz TF-ISF se obtiene de acuerdo a la Ecuación (2.15).

$$\overrightarrow{término}_{in} = (w_{i1}, w_{i2}, \dots, w_{ij}, \dots, w_{in}) \quad (2.15)$$

Es decir, cada término es el promedio del término en todas las oraciones, para su cálculo se suma la fila del término i y se divide por el número total de términos(m) para obtener el peso promedio $\overline{w_i}$, de acuerdo a la Ecuación (2.17).

$$\overline{w_i} = \frac{1}{m} \sum_{k=1}^{n} w_{ik} \quad (2.16)$$

Por lo tanto, el vector centroide se define en la Ecuación (2.17):

$$\overrightarrow{vectorCentroide} = (\overline{w}_1, \overline{w}_2, \dots, \overline{w}_i, \dots, \overline{w}_m) \quad (2.17)$$

2.3 ALGORITMOS BASE

El algoritmo propuesto hibrida la metaheurística de la Mejor Búsqueda Armónica Global (GBHS) con Búsqueda Local Codiciosa y el algoritmo LexRank. En esta sección se presenta una descripción general de estos dos algoritmos.

2.3.1 Mejor Búsqueda Armónica Global (GBHS) con Búsqueda Local

La Mejor Búsqueda Armónica Global (GBHS), es un algoritmo metaheurístico basado en el proceso musical que busca un estado perfecto de la armonía. El proceso de improvisación en los músicos es comparable a los esquemas de búsqueda global y local de las técnicas de optimización. El algoritmo GBHS con búsqueda local [20], utilizado en esta investigación se puede resumir en los siguientes pasos:

Paso 1: Inicializar los parámetros del problema y los parámetros de HS: los parámetros del problemas a inicializar son: máximo número de evaluaciones de la función objetivo, definido en 15000, máxima longitud del resumen para Rouge, definido en 250 palabras, máxima longitud del resumen para evolucionar, definido en 285 palabras, además el parámetro umbral de oraciones que se utiliza en el pre-procesamiento, el cual asegura que cada oración del resumen tenga un mínimo de similitud a la colección de documentos; y los parámetros a inicializar de HS son: tamaño de la memoria armónica (HMS), tasa de consideración de la memoria armónica (HMCR), número de optimizaciones (Nop), probabilidad de optimización (Po), la tasa mínima de ajuste al tono definida en 0.01 y la tasa de máxima de ajuste al tono definida en 0.99.

Paso 2: Inicializar la memoria armónica: la memoria armónica (HM) se inicializa aleatoriamente teniendo en cuenta su tamaño (HMS). En una armonía (vector solución) se representan las oraciones de la colección de documentos, donde uno indica la presencia de la oración en el resumen candidato y cero en caso contrario. Al inicializar cada armonía, la mitad de las oraciones se colocan en uno (1) utilizando una probabilidad de 0.5. Cada vez que una oración se coloca en uno, se verifica que no se viole la restricción del máximo número de palabras en el resumen. Si es necesario se aplica un proceso de reparación, incluyendo conocimiento del problema para eliminar las oraciones que contribuyen menos al resumen (medido en la similitud de cosenos de la oración frente a la colección de documentos, dividida por la longitud de la oración), es decir, las oraciones con menos cobertura para la colección de documentos, y se siguen incluyendo oraciones que no sobrepasen el número máximo de palabras en el resumen, seleccionando primero las oraciones de mayor similitud a la colección de documentos (mayor cobertura).

Paso 3: Evaluar la población inicial: la aptitud para cada armonía en la memoria armónica se calcula con base a la función objetivo, que contempla las características de cobertura y redundancia.

Paso 4: Optimizar la población inicial: cada armonía en la población inicial se optimiza mediante el algoritmo de búsqueda local codiciosa (Ver Figura 2.3), de acuerdo con una probabilidad de optimización. La armonía es reemplazada por el vecino solo si el valor aptitud del vecino mejora la aptitud de la armonía.

Paso 5: Improvisar la nueva armonía: la nueva armonía se genera teniendo en cuenta las tres reglas definidas en GBHS (Ver Figura 2.2), consideración de la Memoria Armónica (HMCR), ajuste de tono (PAR) basado en los conceptos de optimización de enjambre de partículas (PSO) y selección aleatoria de la búsqueda espacial; proceso que se realiza para cada oración de la armonía. Si es necesario, se realiza el proceso de reparación explicado en el Paso 1. El valor de aptitud se calcula para la nueva armonía y se optimiza de acuerdo con la probabilidad de optimización.

Paso 6: Actualizar la memoria armónica: la peor armonía en la memoria armónica (HM) es reemplazada por la nueva armonía, solo si su valor de aptitud es peor que la aptitud de la nueva armonía.

```
01  Para cada i ∈ [1, N] hacer
02      si U(0,1) < HMCR entonces                    //Consideración de la memoria
03              x_i' = x_i^j, donde j ~ U(1, ..., HMS)
04              si U(0,1) ≤ PAR entonces             //Ajuste del tono
05                  x_i = x_k^mejor, donde mejor es el indice de la mejor armonía,
                            en la HM y K ~ U(1,..,Hms)
06              fin_si
07          fin
08      sino                                         //Selección aleatoria
09          x_i' = U(0,1)
10      fin_si
11  Fin Para
```

Figura 2.2 Improvisación de una nueva armonía

Paso 7: Verificar el criterio de parada: Terminar cuando el número máximo de evaluaciones de la función objetivo se alcanza, repitiendo los pasos 5 al 6.

Paso 8: Generación del Resumen: Para generar el resumen el algoritmo selecciona la armonía que tenga el mejor valor de aptitud; y para decidir cuales oraciones harán parte el resumen, se evalúa el valor de aptitud de cada oración, seleccionando las de mayor valor a menor valor, teniendo en cuenta la restricción máxima del número de palabras en el resumen.

2.3.2 Proceso de Optimización Búsqueda Local Codiciosa

Este algoritmo GBHS incorpora la Búsqueda Local con un enfoque voraz [20] (Ver Figura 2.3), donde cada armonía es optimizada de acuerdo a una probabilidad y el número de vecinos es definido por el máximo número de optimizaciones especificado por (*MaxNo*). Un vecino se genera primero agregando una oración de acuerdo al factor de cobertura, es decir, a la armonía se le adiciona la oración (colocando un uno) que tenga la mayor similitud con respecto a la colección de documentos, de acuerdo a una lista con las oraciones

ordenadas por este criterio (Loo); luego se elimina una oración con la menor similitud de la misma lista, controlando el número de oraciones en la armonía. Si el valor de la función objetivo de la nueva armonía mejora la de la armonía anterior, se realiza el reemplazo, de lo contrario, se retiene la armonía anterior.

```
Loo: Lista de oraciones ordenadas por similitud con la colección de documentos
MaxNo: Máximo número de optimizaciones
ArmoníaBase: Armonía a optimizar
Para i = 1 hasta MaxNo Haga
    ArmoniaActual = Copia(ArmoniaBase)
    Agregar_Oraciones(ArmoniaActual)          //La oración de Loo con mayor similitud
    Eliminar_Oraciones(ArmoniaActual).        //La oración de Loo con menor similitud
    Validar_Restriccion_Longitud(ArmoniaActual)
    CacularFuncionObjetivo(ArmoniaActual)
    Si(FuncionObjetivo(ArmoniaActual) > FuncionObjetivo(ArmoniaBase)) entonces
        ArmoniaBase = ArmoniaActual
    Fin si
Fin Para
```

Figura 2.3 Búsqueda Codiciosa

2.3.3 LexRank

Erkan y Radev en [32] plantean el Algoritmo LexRank con Umbral basado en el concepto de prestigio[7] en las redes sociales. Una red social es un mapa de las relaciones entre las entidades que interactúan, por ejemplo, personas y organizaciones. Las redes sociales se representan comúnmente en forma de grafos, donde los nodos representan las entidades y los enlaces representan las relaciones entre los nodos. Un grupo de documentos puede verse como una red de oraciones relacionadas; algunas son más similares entre sí, mientras otras pueden compartir poca información con el resto de las oraciones. Los autores plantean que si una oración es muy similar a muchas de las otras oraciones, esta se puede considerar como la más central (o sobresaliente) o representativa de un tema. Hay dos puntos clave para esta definición de centralidad, primero, cómo definir la similitud entre dos oraciones y segundo, cómo calcular la centralidad global de una oración dada su similitud con otras oraciones.

Para definir la similitud, primero se representan las oraciones del conjunto de documentos en el modelo de espacio vectorial descrito en la sección (2.2.1) y para el peso de las palabras o términos se usa el ponderado TF-ISF de la Ecuación (2.10). La similitud entre dos oraciones se define por la similitud de coseno que se muestra en la Ecuación (2.12). Luego las oraciones de los documentos se representan como un grafo a través de una matriz de adyacencia entre las oraciones donde cada valor corresponde a la similitud de coseno entre las oraciones de la colección de documentos (ver Ecuación (2.14).

La Tabla 1 muestra un subconjunto de noticias de la colección DUC2004 y la Tabla 2, la matriz de similitud de cosenos correspondiente (datos tomados de [32]**)**. Esta matriz de adyacencia se usa para representar el grafo ponderado que relaciona todas las oraciones.

[7]Prestigio y Centralidad en esta propuesta, representan el mismo concepto con la diferencia de que el primero se define usualmente para grafos dirigidos, mientras el segundo se define para grafos no dirigidos.

No	ID	Text
1	d1s1	Iraqi Vice President Taha Yassin Ramadan announced today, Sunday, that Iraq refuses to back down from its decision to stop cooperating with disarmament inspectors before its demands are met.
2	d2s1	Iraqi Vice president Taha Yassin Ramadan announced today, Thursday, that Iraq rejects cooperating with the United Nations except on the issue of lifting the blockade imposed upon it since the year 1990.
3	d2s2	Ramadan told reporters in Baghdad that “Iraq cannot deal positively with whoever represents the Security Council unless there was a clear stance on the issue of lifting the blockade off of it”
4	d2s3	Baghdad had decided late last October to completely cease cooperating with the inspectors of the United Nations Special Commission (UNSCOM), in charge of disarming Iraq’s weapons, and whose work became very limited since the fifth of August, and announced it will not resume its cooperation with the Commission even if it were subjected to a military operation.
5	d3s1	The Russian Foreign Minister, Igor Ivanov, warned today, Wednesday against using force against Iraq, which will destroy, according to him, seven years of difficult diplomatic work and will complicate the regional situation in the area.
6	d3s2	Ivanov contended that carrying out air strikes against Iraq, who refuses to cooperate with the United Nations inspectors, “will end the tremendous work achieved by the international group during the past seven years and will complicate the situation in the region.”
7	d3s3	Nevethless, Ivanov stressed that Baghdad must resume working with the Special Commission in charge of disarming the Iraqi weapons of mass destruction (UNSCOM).
8	d4s1	The Special Representative of the United Nations Secretary-General in Baghdad, Prakash Shah, announced today, Wednesday, after meeting with the Iraqi Deputy Prime Minister Tariq Aziz, that Iraq refuses to back down from its decision to cut off cooperation with the disarmament inspectors.
9	d5s1	British Prime Minister Tony Blair said today, Sunday, that the crisis between the international community and Iraq “did not end” and that Britain is still “ready, prepared, and able to strike Iraq.”
10	d5s2	In a gat hering with the press held at the Prime Minister’s office, Blair contended that the crisis with Iraq “will not end until Iraq has absolutely and unconditionally respected its commitments” towards the Unite Nations.
11	d5s3	A spokesman for Tony Blair had indicated that the British Prime Minister gave permission to British Air Force Tornado planes stationed in Kuwait to join the aerial bombardment against Iraq.

Tabla 1. Subconjunto de documentos del tópico d1003t de DUC 2004.

	d1s1	d2s1	d2s2	d2s3	d3s1	d3s2	d3s3	d4s1	d5s1	d5s2	d5s3
d1s1	1	0,45	0,02	0,17	0,03	0,22	0,03	0,28	0,06	0,06	0
d2s1	0,45	1	0,16	0,27	0,03	0,19	0,03	0,21	0,03	0,15	0
d2s2	0,02	0,16	1	0,03	0	0,01	0,03	0,04	0	0,01	0
d2s3	0,17	0,27	0,03	1	0,01	0,16	0,28	0,17	0	0,09	0,01
d3s1	0,03	0,03	0	0,01	1	0,29	0,05	0,15	0,2	0,04	0,18
d3s2	0,22	0,19	0,01	0,16	0,29	1	0,05	0,29	0,04	0,2	0,03
d3s3	0,03	0,03	0,03	0,28	0,05	0,05	1	0,06	0	0	0,01
d4s1	0,28	0,21	0,04	0,17	0,15	0,29	0,06	1	0,25	0,2	0,17
d5s1	0,06	0,03	0	0	0,2	0,04	0	0,25	1	0,26	0,38
d5s2	0,06	0,15	0,01	0,09	0,04	0,2	0	0,2	0,26	1	0,12
d5s3	0	0	0	0,01	0,18	0,03	0,01	0,17	0,38	0,12	1

Tabla 2. Matriz similitud de Cosenos para tópico d1003t de DUC 2004.

En este algoritmo se usa el concepto de umbral para eliminar aquellas relaciones débiles entre las oraciones (nodos del grafo), es decir, aquellos vértices con similitud de coseno que no supera un valor dado. La Figura 2.4 (Adaptado de [32]), muestra el efecto del umbral sobre el grafo de oraciones para diversos valores, a saber 0, 0.1, 0.2 y 0.3. Aunque en la Tabla 2 se evidencia que existe una fuerte relación de cada nodo consigo mismo (valor de similitud de 1 en toda la diagonal), en la Figura 2.4 no se muestran dichas relaciones para que el grafo sea visualmente más simple, no obstante para los cálculos respectivos se tendrán en cuenta.

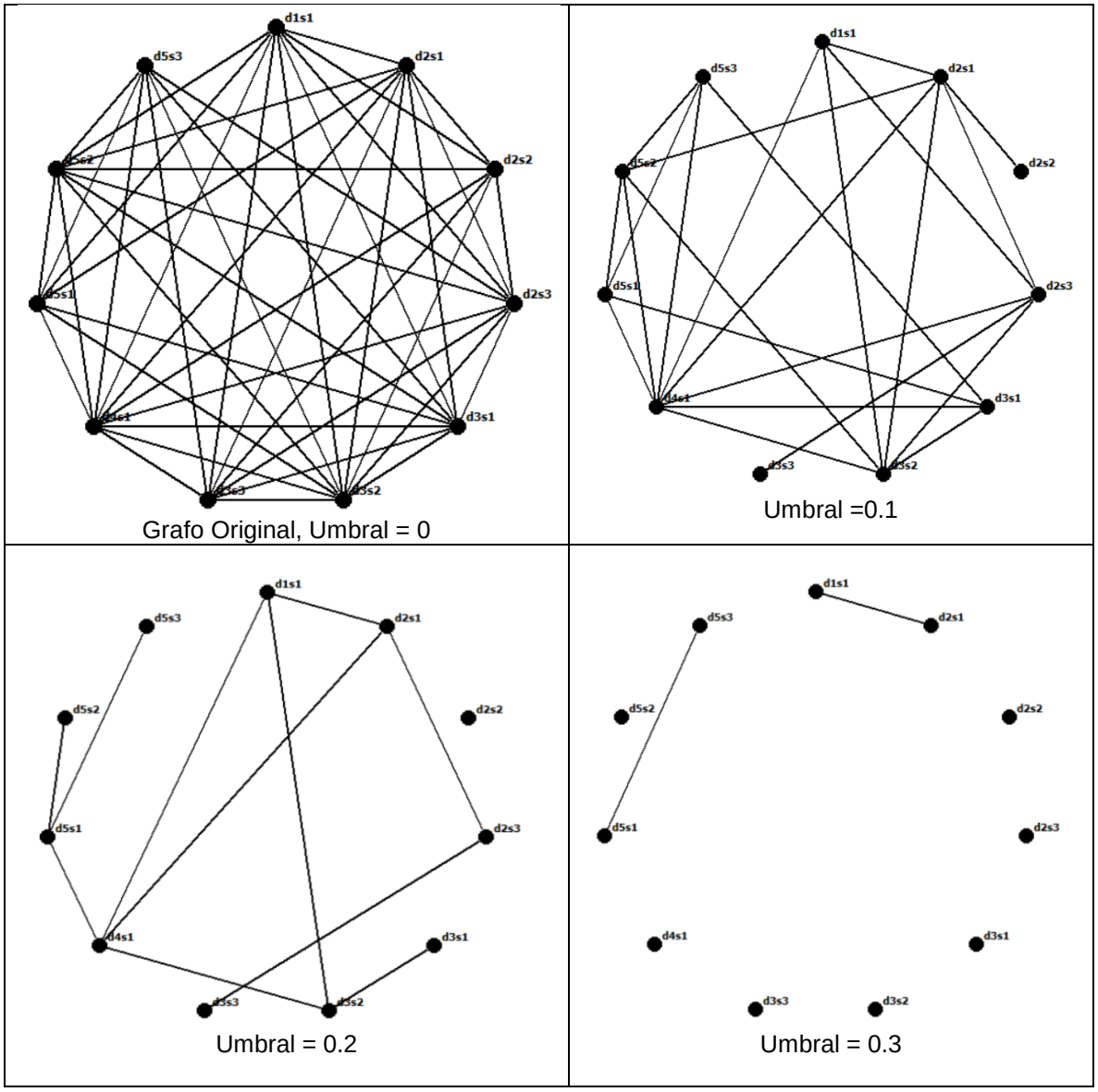

Figura 2.4. Grafos de Similitud con umbrales 0, 0.1, 0.2 y 0.3.

Una forma sencilla de evaluar la centralidad de una oración es contar el número de oraciones que tienen relación de similitud con otra oración después de haber aplicado el umbral (número de vértices que llegan a cada nodo), incluyendo la relación de similitud que tiene la oración consigo misma. En este sentido, se define el grado de centralidad de una oración como el grado del nodo que la representa en el grafo de similitud. En la Tabla 3, se muestra el grado de centralidad para cada una de las oraciones de acuerdo a los umbrales 0.1, 0.2 y 0.3, y resaltando que la oración *d4s1* es la más central para los umbrales 0.1 y 0.2.

La elección del umbral como se muestra en la Tabla 3 (Adaptado de [32]), influye en la interpretación de la centralidad, dado que umbrales bajos tienen en cuenta similitudes débiles, mientras que umbrales grandes pueden eliminar muchas de las relaciones de similitud en el grupo de oraciones.

ID	Grado (0.1)	Grado (0.2)	Grado (0.3)
d1s1	5	4	2
d2s1	7	4	2
d2s2	2	1	1
d2s3	6	3	1
d3s1	5	2	1
d3s2	7	4	1
d3s3	2	2	1
d4s1	**9**	**5**	**1**
d5s1	5	4	2
d5s2	6	2	1
d5s3	5	2	2

Tabla 3. Grado de Centralidad.

Hasta ahora, al calcular el grado de centralidad se ha tratado a cada vértice (relación) como un voto para determinar el valor total de centralidad de cada nodo; este es un método totalmente democrático donde cada voto tiene el mismo valor. Sin embargo se puede tener un efecto negativo en la calidad de los resúmenes, porque en algunos casos varias oraciones no deseadas votan la una a favor de la otra aumentando el grado de centralidad. En las redes sociales las personas están conectadas entre sí con relación de amistad, sin embargo, en muchos tipos de redes sociales no todas las relaciones son consideradas igualmente importantes; el prestigio de una persona no depende solo de la cantidad de amigos que tiene, sino con quienes comparte una mayor relación de amistad y que tan influyentes son esas amistades en la red social.

La misma idea se puede aplicar a la generación automática de resúmenes, haciendo que el grado de centralidad tenga en cuenta no solo los votos de cada nodo sino también de donde vienen esos votos (la importancia o centralidad de los nodos que se relacionan con el nodo actual). Una formulación de esta idea es considerar que cada nodo tiene un valor de centralidad distribuido entre el nodo mismo y sus vecinos; esto se puede expresar en la Ecuación (2.18):

$$p(u) = \sum_{v \,\epsilon\, adj[u]} \frac{p(v)}{\deg(v)} \tag{2.18}$$

Donde $p(u)$ es la centralidad del nodo u, $adj[u]$ es el conjunto de nodos que son adyacentes a u, y $\deg(v)$ es el grado del nodo v. La Ecuación (2.19) se puede escribir en notación matricial de la siguiente manera:

$$p = B^T p, \text{ que es igual a: } p^T B = p^T \tag{2.19}$$

Donde la matriz B se obtiene de la matriz de adyacencia del grafo de similitud dividiendo cada elemento por la suma de la fila correspondiente, de acuerdo a la Ecuación (2.20).

$$B(i,j) = \frac{A(i,j)}{\sum_k A(i,k)} \tag{2.20}$$

Es preciso tener en cuenta que la suma de una fila es igual al grado del nodo correspondiente. Puesto que cada oración es similar al menos a sí misma, todas las sumas de fila son distintas de cero. La Ecuación (2.19) establece que p^Tes el vector propio izquierdo de la matriz B con el valor propio correspondiente a 1. Para garantizar la existencia de un vector propio y que pueda ser identificado y calculado de forma única, se necesita tener en cuenta los siguientes fundamentos matemáticos.

Una matriz *estocástica* X, es la matriz de transición de una cadena de Markov. En esta, un elemento $X(i,j)$ especifica la probabilidad de transición de un estado i a un estado j en la cadena de Markov correspondiente. Por los axiomas de probabilidad, todas las filas de una matriz estocástica deben sumar 1. $x^n(i,j)$, da la probabilidad del estado i para alcanzar el estado j en n transiciones. Una cadena de Markov con la matriz estocástica X converge a una distribución estacionaria de acuerdo a la Ecuación (2.21):

$$\lim_{n\to\infty} X^n = 1^T r \tag{2.21}$$

Donde $1 = (1,1,\ldots,1)$, y el vector r se llama la distribución estacionaria de la cadena de Markov. Una interpretación intuitiva de la distribución estacionaria puede entenderse por el concepto de una caminata aleatoria. Cada elemento del vector r da la probabilidad asintótica de terminar en el estado correspondiente a largo plazo, independientemente del estado de partida.

Una cadena de Markov es irreducible si cualquier estado es accesible desde cualquier otro estado, es decir, para todos i,j existe un n de tal manera que $X^n\ (i,j) \neq 0$. Una cadena de Markov es aperiódica si para todo i, $\gcd\{n : X^n(i,i) > 0\} = 1$. Por el teorema de Perron-Frobenius, una cadena de Markov irreducible y aperiódica converge a una distribución estacionaria única. Si una cadena de Markov tiene componentes reducibles o periódicos, un caminante aleatorio puede atascarse en estos componentes y nunca visitar las otras partes del grafo.

Dado que la matriz de similitud B en la Ecuación (2.19) satisface las propiedades de una matriz estocástica, se puede tratar como una cadena de Markov. El vector de centralidad P corresponde a la distribución estacionaria de B. Sin embargo, se debe asegurar que la matriz de similitud sea siempre irreductible y aperiódica. Para resolver este problema, se reserva una baja probabilidad para saltar a cualquier nodo en el grafo. De esta manera el caminante puede "escapar" de componentes periódicos o desconectados, lo que hace que el grafo sea irreducible y aperiódico. Si se asigna una probabilidad uniforme para saltar a

cualquier nodo en el grafo, se obtiene la siguiente versión modificada de la Ecuación (2.18), que se conoce como el algoritmo PageRank, ver Ecuación (2.22).

$$p(u) = \frac{d}{N} + (1-d)\sum_{v\ \epsilon\ adj[u]} \frac{p(v)}{\deg(v)} \tag{2.22}$$

Donde $\boldsymbol{N}$ es el número total de nodos en el grafo y $\boldsymbol{d}$ es un "factor de amortiguamiento", que típicamente se elige en el intervalo [0.1, 0.2]. La Ecuación (2.22) se puede escribir en forma matricial como se muestra en la Ecuación (2.23).

$$p = [dU + (1-d)B]^T\, p \tag{2.23}$$

Donde U es una matriz cuadrada de NxN con todos los elementos iguales a $\frac{1}{N}$. El kernel de transición $[dU + (1-d)B]$ de la cadena de Markov resultante es una mezcla de dos núcleos (kernels) U y B. Una caminata (recorrido) aleatoria en esta cadena de Markov elige uno de los estados adyacentes del estado actual con probabilidad $1-d$, o salta a cualquier estado en el grafo, incluyendo el estado actual, con probabilidad d. La fórmula PageRank fue propuesto por primera vez para el cálculo del prestigio de una página y aún hoy es usada por Google.

La propiedad de convergencia de las cadenas de Markov también proporciona un algoritmo iterativo simple, llamado el método de potencia, para calcular la distribución estacionaria, de acuerdo a su representación en la Figura 2.5, adaptado de [32] . El algoritmo comienza con una distribución uniforme; en cada iteración, el vector propio se actualiza multiplicándolo con la transpuesta de la matriz estocástica. Dado que la cadena de Markov es irreducible y aperiódica, se garantiza que el algoritmo termina.

Entrada: Una matriz M estocástica, irreducible y aperiódica
Entrada: Tolerancia de error £
Salida: Vector propio p
01 $p_o = \frac{1}{N}1$
02 $t = 0$
03 repita
04 $t = t + 1$
05 $p_t = M^T P_{t-1}$
06 δ = $\|p_t - p_{t-1}\|$
07 hasta que δ < £;
08 retorne p_t

Figura 2.5 Algoritmo Método de Potencia

A diferencia del método original de PageRank, el grafo de similitud para oraciones no es dirigido, ya que la matriz de similitud de cosenos es simétrica. Sin embargo, esto no hace ninguna diferencia en el cálculo de la distribución estacionaria. La Figura 2.6 Adaptado de [32], resume cómo calcular las puntuaciones LexRank para un determinado conjunto de oraciones.

```
Entrada: Arreglo S de n oraciones, umbral t, valor de amortiguamiento dampingFactor
Salida:  Arreglo L con los scores definidos por LexRank para cada oración
Arreglo  CosineMatriz[n][n], L[n];
01 Para i=1 hasta n haga
02   suma=0
03   Para j=1 hasta n haga
04        CosineMatriz[i][j] = idf-modified-cosine(S[i],S[j]);
05        Si CosineMatriz[i][j] > t haga
06             CosineMatrix[i][j] = 1;
07             suma++;
08        Si No
09             CosineMatriz[i][j] = 0;
10        Fin Si
11   Fin Para
15 Fin Para
16 Para i=1 hasta n haga
17   Para j=1 hasta n haga
18        CosineMatriz[i][j] = CosineMatriz[i][j] / suma;
19   Fin Para
20 Fin Para
21 Para i=1 hasta n haga
22   Para j=1 hasta n haga
23        CosineMatriz[i][j] = (dampingFactor/n) + (1- dampingFactor)* CosineMatriz[i][j];
24   Fin Para
25 Fin Para
26 L = PowerMethod(CosineMatrix, n, £ ); //Algoritmo previamente presentado
27 retorne L;
```

Figura 2.6 Algoritmo Cálculo de puntuación de LexRank,

Capítulo 3

3 PROCESO DE CONSTRUCCIÓN: ALGORITMOS HÍBRIDOS

Para el desarrollo de este proyecto se utiliza el Patrón de Investigación Iterativa propuesto por Pratt [52] diseñado especialmente para proyectos de investigación que involucran una solución computacional. Está compuesto por cuatro etapas principales que son: observación, identificación del problema, desarrollo de la solución y prueba de la solución. 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: Definición hibridación y función objetivo

En este primer ciclo se realizó el proceso de identificación de las posibles formas de hibridar los algoritmos base GBHS y LexRank; también se explica cada una de las versiones del algoritmo híbrido de acuerdo a las diferentes formas seleccionadas para hibridar los algoritmos base, el proceso de diseño de la función objetivo teniendo en cuenta diferentes características de similitud y sus diferentes configuraciones, el proceso de configuración de parámetros asociados al problema y a cada uno de los algoritmos base, y por último el proceso de afinación de los pesos asociados a cada una de las características de la función objetivo.

3.1.1 Identificación formas de hibridación

Para identificar las diferentes formas de hibridar los algoritmos LexRank [2] y GBHS [20], se siguió lo planteado en [53, 54], representadas de acuerdo a:

- Hibridación 1 Tipo: Alto Nivel–Relevo. En este tipo de hibridación (ver Figura 3.1), los algoritmos trabajan de forma secuencial, donde el primer algoritmo obtiene unos resultados que son utilizados por el segundo algoritmo para continuar con el proceso de generar el resumen final.

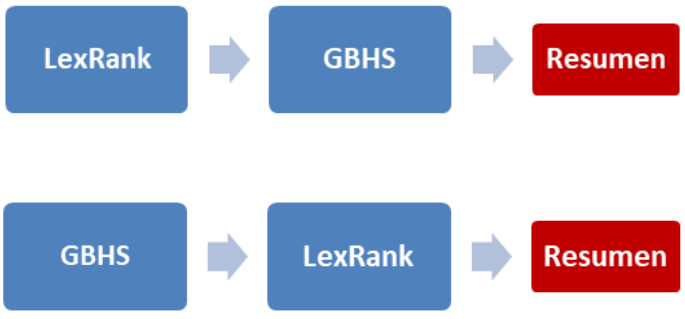

Figura 3.1 Formas Hibridación Tipo Alto Nivel-Relevo

- Hibridación 2 Tipo: Bajo Nivel. En este tipo, un algoritmo es el encargado de generar el resumen y el otro algoritmo se encarga de realizar una o varias tareas que debe hacer el primer algoritmo (Ver Figura 3.2).

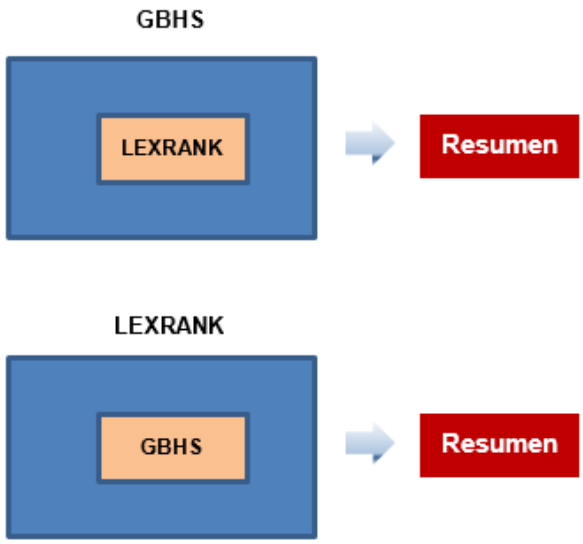

Figura 3.2 Formas Hibridación Tipo Bajo Nivel

- Hibridación 3 Tipo: Alto Nivel–Trabajo en Equipo. En esta forma (Ver Figura 3.3), los algoritmos trabajan en paralelo, cada uno de ellos genera un conjunto de oraciones candidatas a pertenecer el resumen y al finalizar se seleccionan las mejores para obtener el resumen final.

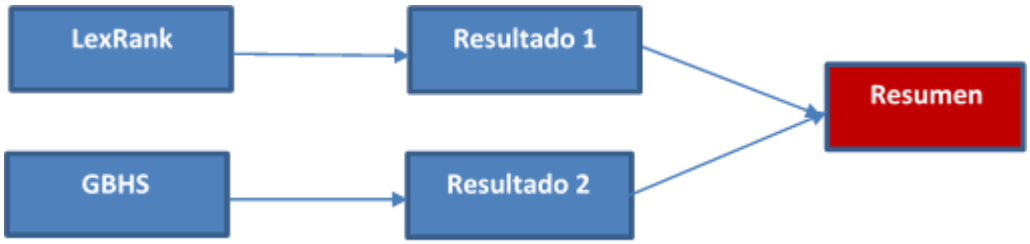

Figura 3.3 Hibridación Alto Nivel- Trabajo en Equipo

3.1.2 Selección formas de Hibridación algoritmos base

De acuerdo a las diferentes formas posibles de hibridar, en este proyecto se definieron las siguientes versiones de hibridación:

a. Híbrido versión 1 (LexRank-GBHS): este algoritmo híbrido (Ver Figura 3.4) primero ejecuta LexRank, donde inicialmente hace una selección de oraciones mediante la

eliminación de relaciones débiles entre oraciones y después realiza un proceso iterativo que le permite obtener las oraciones ordenadas de mayor a menor relevancia, (mayor cobertura, medido en la similitud de cosenos entre la oración a la colección de documentos), luego el algoritmo híbrido define un parámetro Poda que permite eliminar las oraciones menos relevantes que se obtienen de LexRank y actualiza la matriz de similitudes teniendo en cuenta solo las oraciones resultantes del proceso de Poda; después con esas oraciones continua la ejecución de GBHS, quien al finalizar devuelve un resumen con las oraciones de la mejor armonía de la memoria armónica, teniendo en cuenta las restricciones del máximo número de palabras que debe tener el resumen.

Figura 3.4 Hibridación 1 LexRank-GBHS

b. Híbrido versión 2 (GBHS-LexRank): este algortimo híbrido (Ver Figura 3.5), primero ejecuta GBHS, de donde se obtienen las oraciones no repetidas de la última población de su proceso evolutivo, luego el algoritmo híbrido actualiza la matriz de similitudes teniendo en cuenta solo las oraciones que se obtienen de GBHS, continuando con la ejecución de LexRank (sin modificaciones con respecto a la primera versión) y entrega también un vector que contiene la ponderación de las oraciones ordenadas de mayor a menor relevancia. Por último, se genera el resumen incluyendo las oraciones que tienen mayor relevancia hasta que se complete el tamaño máximo del resumen.

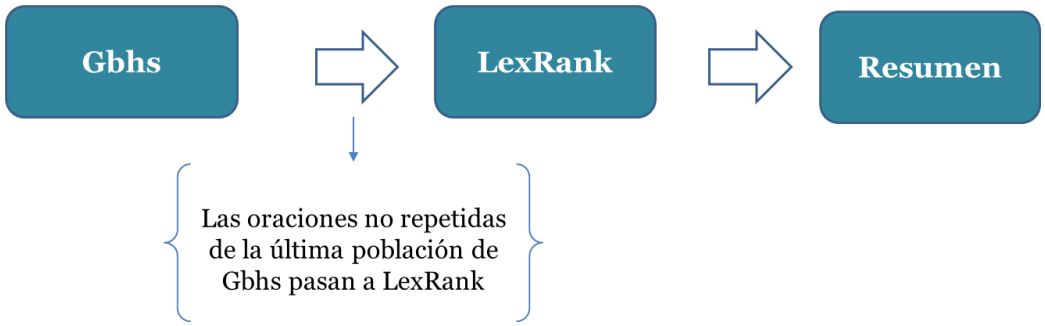

Figura 3.5 Hibridación 1 GBHS-LexRank

c. Híbrido versión 3 (LexRank-GBHS-2Resumen): este algoritmo híbrido (Ver Figura 3.6), realiza el mismo proceso del algortimo híbrido versión 1, con la modificación de que genera dos resúmenes, uno al finalizar la ejecución de LexRank, y el otro al finalizar la ejecución del algoritmo híbrido. El resumen final del algoritmo híbrido versión 3, se obtiene a partir de las oraciones de los dos resúmenes, seleccionando primero las oraciones que se repiten en los dos resúmenes, y las demás se seleccionan de mayor a menor diversidad.

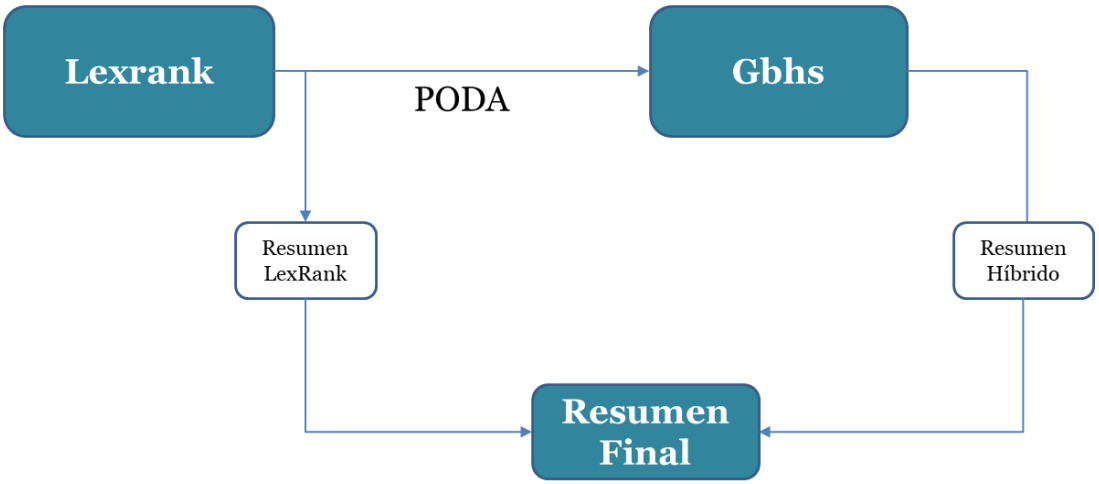

Figura 3.6 Hibridación 1 LexRank-GBHS

d. Híbrido versión 4 (GBHS-LexRank-2Resumen): este algoritmo híbrido (Ver Figura 3.7), realiza el mismo proceso del algoritmo híbrido versión 2, con la modificación de que se generan dos resúmenes, uno al terminar la ejecución de GBHS y otro al terminar la ejecución del algoritmo híbrido. El resumen final del algoritmo híbrido versión 4, se obtiene de igual forma que en el hibrido Versión 3.

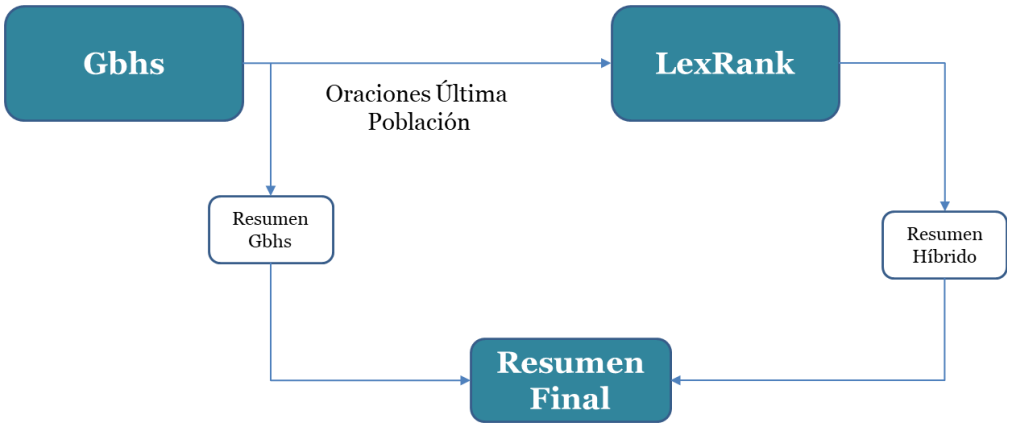

Figura 3.7 Hibridación 1 GBHS-LexRank-2Resumen

e. Híbrido versión 5 (LexRank-GBHS-Paralelo): este algoritmo híbrido (Ver Figura 3.8), ejecuta los dos algoritmos base LexRank y GBHS por separado, donde cada uno genera un resumen. Las oraciones del resumen de LexRank son seleccionadas de mayor a menor cobertura y las oraciones del resumen de Gbhs son seleccionadas de mayor a menor valor de aptitud de acuerdo a la función objetivo. El resumen final del algoritmo híbrido versión 5, se obtiene de igual forma que en el hibrido Versión 3.

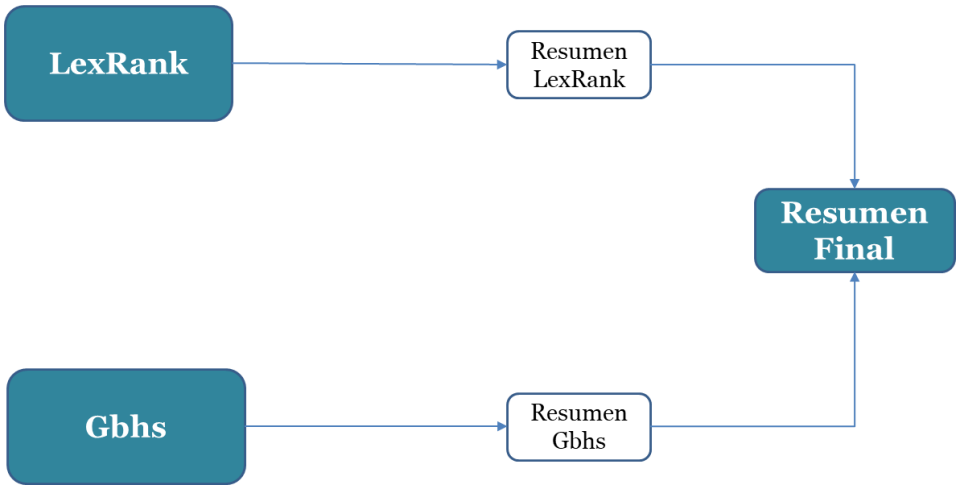

Figura 3.8 Hibridación 3 Paralela LexRank-GBHS-Paralelo

3.1.3 Diseño de la función objetivo

La generación automática de resúmenes para múltiples documentos busca seleccionar las oraciones más relevantes de un conjunto de documentos, como en este caso, los documentos están relacionados con el mismo tema, es importante tener en cuenta que las oraciones seleccionadas como parte del resumen no deben ser iguales o similares, evitando de esta forma la redundancia para obtener mejores resúmenes. La función objetivo se convierte en uno de los principales mecanismos que aproximan conocimiento del problema específico, ésta rige la exploración de soluciones evaluando su competencia para resolver el problema abordado y determinar su calidad.

Para realizar el diseño de la función objetivo se toma como base la función objetivo definida en [20] y además algunas características encontradas durante la revisión del estado del arte. Estas características son:

3.1.3.1 Cobertura

Característica que selecciona las oraciones más relevantes contenidas en la colección de documentos con la menor pérdida de información. La cobertura se calcula teniendo en cuenta la similitud de coseno entre cada oración del resumen candidato y las oraciones de toda la colección de documentos, de acuerdo a lo planteado en [20], como se muestra en la Ecuación (3.1).

$$\boldsymbol{Cobertura} = \frac{\sum_{i=1}^{o} \boldsymbol{sim}(\boldsymbol{D}, \boldsymbol{s_i})}{\boldsymbol{o}} \tag{3.1}$$

Donde *o* es el número de oraciones seleccionadas en el resumen candidato, D representa el centroide de la colección de documentos. $sim(D, s_i)$, es la similitud de coseno entre el vector centroide de la colección de documentos y el vector que representa la oración.

3.1.3.2 Diversidad

Cuando se está tratando el problema de la generación de resúmenes para múltiples documentos que tratan de un mismo tema, la diversidad busca que el resumen generado no contenga información repetida. La diversidad se calcula por medio de la similitud

promedio de las oraciones del resumen candidato, donde se le hace una modificación a la Ecuación de la característica de redundancia propuesta en [48], donde se realiza una normalización al factor de diversidad para que tome valores entre 0 y 1 de acuerdo a la Ecuación (3.2).

$$\boldsymbol{Diversidad} = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(1 - sim(s_i, s_j)\right)}{(n * (n-1)/2)} \tag{3.2}$$

Donde s_i y s_j, son oraciones en el resumen candidato, n es el número de oraciones en el resumen y $sim(s_i, s_j)$ es la similitud de coseno entre la oración s_i y s_j.

3.1.3.3 Posición

La posición de una oración en un documento es una de las heurísticas más efectivas para seleccionar oraciones relevantes en la generación de resúmenes automáticos de texto, particularmente para noticias [55]. Según estudios previos, la información relevante de un documento, independientemente de su dominio, tiende a encontrarse en algunas secciones como títulos, encabezados, oraciones principales de los párrafos, párrafos iniciales, etc. En la generación de resúmenes de texto automáticos, se han aplicado diferentes técnicas basadas en la característica de posición de la oración, para probar su efectividad en determinar la relevancia de una oración [55].

- Posición 1

En Bossard et al. [56], se utiliza un método basado en la posición de la oración (posición que hay entre el inicio del documento y la oración). El puntaje más alto es asignado a la primera oración cuya posición es la más cercana al inicio del documento y va disminuyendo de manera no uniforme a medida que la oración se aleja respecto al inicio del documento. El puntaje de la última oración tiende a cero. El cálculo de esta característica está definido como se observa en la Ecuación (3.3), de modo que el puntaje del resumen candidato es la sumatoria de los puntajes de cada una de las oraciones que lo componen.

$$P(s_i) = \sum_{\forall s_i \in Resumen}^{n} \sqrt{\frac{1}{q_i}} \tag{3.3}$$

Donde q_i indica la posición de la oración s_i en el documento.

- Posición 2

En [57], se define un esquema que calcula el factor de posición, como la posición promedio en el resumen (ver Ecuación (3.4)), basado en la clasificación de la posición de cada oración en el documento de acuerdo a la Ecuación (3.5). El puntaje de la clasificación de la posición en la oración en el documento va disminuyendo de manera uniforme desde el mayor valor dado a la primera oración, con un valor de $2/n$, hasta el puntaje mínimo dado a la última oración con un valor de cero; y el factor de posición, PFs, se obtiene a partir de la sumatoria de la clasificación de la posición de cada oración en el documento dividida por el número de oraciones en el resumen de acuerdo a la Ecuación (3.4).

$$PFs = \frac{\sum_{\forall s_i \in S} PosicionRanking(S_i)}{O} \tag{3.4}$$

Donde O es el número de oraciones en el resumen S y $PosicionRanking(S_i)$ es la clasificación de la posición de cada oración S_i calculada de acuerdo a la Ecuación (3.5).

La Ecuación (3.5) se basa en el método de selección lineal de rango usada en algoritmos genéticos. El mejor en la clasificación recibe un valor de $2/n$ y el más bajo en la clasificación un valor de cero

$$PosicionRanking(S_i) = \frac{2-2*\left(\frac{pos_i-1}{n-1}\right)}{n} \tag{3.5}$$

Donde pos_i es la posición de la oración i según el orden de aparición en el documento, y n es el número total de oraciones en el documento.

- Posición 3

En [58], presentan un esquema muy parecido al presentado en Posición 1, donde la primera oración en el documento se considera la oración más importante y altamente candidata para ser incluida en el resumen. El puntaje más alto (uno) es asignado a la primera oración y el más bajo (cero) a la última oración. La diferencia con la Posición 1, radica en que la oración más lejana al inicio del documento el valor tiende a cero y en la Posición 3 el valor es cero. El cálculo de esta característica está de finido como se observa en la Ecuación (3.6), de modo que el puntaje del resumen candidato es la sumatoria de los puntajes de cada una de las oraciones que lo componen.

$$P(s_i) = \sum_{\forall s_i \in Resumen} 1 - \frac{i}{n} \tag{3.6}$$

Donde i es la i-ésima oración en el documento, i comenzando en cero y n es el número total de oraciones en el documento.

3.1.3.4 Longitud

Algunos estudios han concluido que las oraciones más cortas de un documento deberían tener menos probabilidades de aparecer en el resumen del documento [59]. En [60], se presenta una normalización basada en la función sigmoide, para calcular esta característica. Esta estimación toma en cuenta la distribución estándar de los datos para llegar a una evaluación más equilibrada, que aún favorece las oraciones más largas, pero no descarta completamente las de longitud media, con la presunción de que también podrían tener información relevante para el resumen. 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 demasiado corta obtenga una buena calificación en esta característica. Con base en estas premisas, las ecuaciones (3.7), y (3.8), muestran la forma de calcular el factor de longitud para las oraciones del resumen L.

- Longitud 1

Proponen un método que evalúa favorablemente los resúmenes compuestos de oraciones de tamaño mediano y oraciones largas

$$L = \sum_{\forall s_i \in S} \frac{1 - e^{-\alpha_i}}{1 + e^{-\alpha_i}} \tag{3.7}$$

$$\alpha_i = \frac{l_i - \mu(l_i)}{std(l_i)} \tag{3.8}$$

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

- Longitud 2

La longitud 2 resulta de una modificación de la longitud 1, donde se hace énfasis en la búsqueda de soluciones que tengan menos oraciones cortas. El factor de longitud es calculado como el promedio de las longitudes, teniendo en cuenta el número total de oraciones en el documento n. La modificación de esta fórmula se muestra en la Ecuación (3.9) y la Ecuación (3.8) se mantiene igual.

$$L = \frac{\sum_{\forall s_i \in S} \frac{1 - e^{-\alpha_i}}{1 + e^{-\alpha_i}}}{n} \tag{3.9}$$

Cuando n es grande se disminuye el valor de esta característica en la función objetivo, comparado con un resumen cuyo n es más pequeño.

3.1.4 Configuraciones de la Función Objetivo

Se definieron tres configuraciones de la función objetivo que incluyen las características mencionadas en la sección 3.1.3, las cuales fueron evaluadas para determinar cuál se adaptaba mejor a las necesidades del problema de la generación automática de resúmenes extractivos de múltiples documentos.

3.1.4.1 Primera Función Objetivo

Teniendo en cuenta los buenos resultados obtenidos en [20], se decidió que la primera Función objetivo debía incluir las características de Cobertura y Diversidad como se observa en la Ecuación (3.10), con una restricción que no permite que el resumen supere una cantidad de palabras específicas de acuerdo a la Ecuación (3.11). Las características de la función objetivo presenta unos pesos, los cuales deben ser afinados para encontrar un valor apropiado, pero la suma de esos no deben superar uno para facilidad del proceso de afinamiento (Ver Ecuación (3.12)).

$$\textbf{Maximizar } \mathrm{F(x)} = \alpha * \mathrm{Cobertura} + \beta * \mathrm{Diversidad} \tag{3.10}$$

Sujeto a:

$$\sum_{i=1}^{n} l_i x_i \leq L \tag{3.11}$$

$$\alpha + \beta = 1 \tag{3.12}$$

En la Ecuación (3.11), x_i indica uno si la oración S_i se selecciona y cero de lo contrario; l_i es la longitud de la oración S_i (medida en palabras) y L es el número máximo de palabras permitidas en el resumen.

Todas las funciones objetivo definidas presentan la restricción de máximo número de palabras del resumen de la Ecuación (3.11).

3.1.4.2 Segunda Función Objetivo

Esta segunda versión de la función objetivo se diseñó teniendo en cuenta las mismas características de la primera versión, pero se agregó la característica de posición de la oración en el documento como se observa en la Ecuación (3.13).

$$\textbf{Maximizar } \mathrm{F(x)} = \alpha * \mathrm{Cobertura} + \beta * \mathrm{Diversidad} + \delta * \mathrm{posición\ 2} \tag{3.13}$$

Sujeto a:

$$\alpha + \beta + \delta = 1 \tag{3.14}$$

Para definir cuál característica de posición utilizar en la Ecuación (3.13), se realizaron pruebas con cada una de las versiones descritas en (3.1.3.3) y se seleccionó la posición 2 quien fue la que obtuvo mejores resultados.

3.1.4.3 Tercera Función Objetivo

Para esta versión se incluyeron las mismas características utilizadas en la segunda función objetivo, agregando la característica de longitud de la oración. La función objetivo se calcula de acuerdo a la Ecuación (3.15).

$$\textbf{Max } \mathrm{F(x)} = \alpha * \mathrm{Cobertura} + \beta * \mathrm{Diversidad} + \delta * \mathrm{posición\ 2} + \sigma * \mathrm{Longitud\ 2} \tag{3.15}$$

Sujeto a:

$$\alpha + \beta + \delta + \sigma = 1 \tag{3.16}$$

Para definir la característica de longitud a utilizar en la Ecuación (3.15), se realizó el mismo mecanismo utilizado en la segunda función objetivo para seleccionar la característica de posición; la longitud 2 fue quien obtuvo mejores resultados, según las versiones descritas en la 3.1.3.4.

3.1.5 Configuración parámetros algoritmos híbridos

El algoritmo híbrido necesita de 14 parámetros, de los cuáles 3 parámetros corresponden al algoritmo basado en grafos LexRank, 7 parámetros corresponden a GBHS y 4 parámetros están asociados al problema.

3.1.5.1 Algoritmo LexRank

El algoritmo LexRank presenta tres parámetros que son:

- **Threshold**: Parámetro utilizado para eliminar relaciones débiles entre las oraciones del grafo. Valores demasiado bajos pueden incluir de forma errónea similitudes débiles (oraciones con poco prestigio), mientras que valores demasiado grandes pueden perder similitudes fuertes (oraciones con alto prestigio). El rango de posibles valores se seleccionó de acuerdo a experimentos hechos con la ejecución del algoritmo LexRank en forma individual.
- **Damping Factor**: Probabilidad que permite que dos oraciones tengan una similitud diferente a 0. El rango de posibles valores se seleccionó de acuerdo a experimentos hechos con la ejecución del algoritmo LexRank en forma individual.
- **Error de tolerancia**: Es el valor definido para terminar el método de potencia (*PowerMethod*) cuando se está calculando la matriz estacionaria en LexRank. La diferencia entre los vectores $p_n - p_{n-1}$, debe ser menor que el error de tolerancia para terminar el proceso. El rango de posibles valores se seleccionó de la misma forma como se seleccionó el parámetro threshold.

El rango de posibles valores para cada uno de los parámetros del algoritmo LexRank, se presenta en la Tabla 4.

Parámetro	Valor
Threshold	0.0,0.01,…,0.15
Damping Factor	0,06,…,0.19
Error de tolerancia	0.01,…,0.10

Tabla 4 Parámetros Algoritmo LexRank

3.1.5.2 Algoritmo Mejor Búsqueda Armónica Global

El algoritmo GBHS cuenta con 6 parámetros que son:

- **Tamaño de la Población (Hms)**: Indica la cantidad de armonías (población inicial) que serán inicializadas en el algoritmo.
- **Hmcr**: La tasa de consideración de la memoria armónica, permite controlar la intensificación del espacio de búsqueda. El rango de valores utilizados se seleccionan de acuerdo a la experiencia y resultados en otros proyectos de investigación.
- **Parmin**: Es la tasa mínima de ajuste al tono, la cual define el porcentaje mínimo que se permite modificar de la nueva armonía. El valor definido de acuerdo a la Tabla 5 , permite junto a Parmax calcular la tasa de ajuste del tono PAR.
- **Parmax**: Tasa máxima de ajuste al tono. El valor definido es el valor máximo posible que puede tener la tasa de Ajuste al tono PAR.

- **PAR:** Tasa de ajuste al tono. Valor calculado de forma dinámica, varía a medida que cambia el número de evaluaciones de la función objetivo.
- **Probabilidad de Optimización (Po)**: Tasa utilizada para definir la cantidad de armonías a optimizar.
- **Máximo número de optimizaciones (Nop):** Cantidad de optimizaciones que se le realizan a una armonía (vecinos).

El rango de posibles valores para cada uno de los parámetros del algoritmo GBHS, se presenta en la Tabla 5.

Parámetro	Valor
Hms	5,10, 15,…,90,95,100
Hmcr	0.5,0.6,0.7,0.8,0.9
Parmin	0,01
Parmax	0,99
PAR	Parmin…Parmax
Po	0.1,0.2,…,0.8,0.9
Nop	10,15,20,25,30,35,40,45,50

Tabla 5 Parámetros Algoritmo Mejor Búsqueda Armónica

3.1.5.3 Asociados al problema

A continuación se presentan los parámetros asociados al problema:

- **Máximo número de evaluaciones de la función objetivo**: Cantidad máxima de veces que se puede evaluar la función objetivo. El valor definido en la, permite hacer una comparación bajo las mismas condiciones con los otros algoritmos metaheurísticos del estado del arte.
- **Máxima Longitud del resumen para Rouge:** Cantidad de palabras que puede tener el resumen, valor que permite la comparación con los resúmenes de referencia.
- **Máxima Longitud del resumen para evolucionar:** Parámetro que indica el límite de palabras que puede tener el resumen durante la evolución del algoritmo. Al final de la ejecución del algoritmo se aplica el segundo parámetro para que el resumen cumpla con la restricción.
- **UmbralPoda:** Parámetro que permite eliminar oraciones menos relevantes de acuerdo a su cobertura (medido en la similitud de cosenos entre la oración a la colección de documentos).
- **Criterio para deshabilitar una oración:** establece que característica(s) de la función objetivo serán utilizadas para calcular el puntaje de cada oración de la colección de documentos, y seleccionar la oración con el puntaje más bajo para ser deshabilitada. Se consideró que la cobertura era la característica más prometedora para medir este criterio, porque es una característica que intenta seleccionar las oraciones que contienen los aspectos principales de los documentos con la menor pérdida de información, intentado abarcar la mayor cantidad de información.
- **Criterio de selección de oraciones del resumen final:** define que característica(s) de la función objetivo serán utilizadas para especificar el orden en que las oraciones aparecen en el resumen generado. Para las versiones 1 y 2 el criterio de selección es

por la característica de cobertura y en las otras tres versiones donde se generan dos resúmenes y a partir de ahí se genera el resumen final, el criterio se realiza seleccionando primero las oraciones que coinciden en los dos resúmenes y después si se requiere se seleccionan por diversidad.

El rango de posibles valores para cada uno de los parámetros asociados al problema, se presenta en la Tabla 6.

Parámetro	Valor
Máximo número de evaluaciones de la función objetivo	15000
Máxima Longitud del resumen para Rouge	250
Máxima Longitud del resumen para evolucionar	250, 255,…,280,285
Umbral de Poda	0.1,0.2,…,0.8,0.9
Criterio para deshabilitar una oración	Cobertura
Criterio de selección de oraciones del resumen final	Cobertura, Diversidad

Tabla 6 Parámetros Asociados al Problema

3.1.6 Afinación de las funciones objetivo

Para el afinamiento de los pesos asociados a cada una de las características, inicialmente se tomaron como referencia los valores dados por los Arreglos de cobertura [61]. Luego se realizaron experimentos con dichos valores y se fueron modificando cada uno de los pesos de las características con valores cercanos a los valores encontrados inicialmente. La función objetivo a optimizar, incorpora diferentes características, las cuales tienen asociados los siguientes pesos:

- **Alfa**: Parámetro asociado a la característica de cobertura, valor entre 0 y 1.
- **Beta**: Parámetro asociado a la característica de diversidad, valor entre 0 y 1.
- **Delta**: Parámetro asociado a la característica de posición, valor entre 0 y 1.
- **Sigma**: Parámetro asociado a la característica de longitud, valor entre 0 y 1.

Como condición se debe cumplir alfa + beta + delta + sigma = 1, de acuerdo a la Ecuación (3.16).

El rango de posibles valores para cada uno de los parámetros asociados a la función objetivo, se presenta en la Tabla 7.

Parámetro	Valor
Alfa	0.0,0.1,...,0.9,1
Beta	0.0,0.1,…,0.9,1
Delta	0.0,0.1,...,0.9,1
Sigma	0.0,0.1,...,0.9,1

Tabla 7 Parámetros Asociados a la Función Objetivo

3.2 CICLO II: Diseño algoritmo híbrido Versión 1

Inicialmente para el Diseño de las diferentes versiones del algoritmo híbrido, se realizaron algunas modificaciones a GBHS, las cuales se relacionan a continuación:

1. Inicialización de la población inicial:

   - GRASP: se utiliza esta técnica para cada nueva armonía (Ver Figura 3.9) realizando un número fijo de iteraciones (5) con los siguientes pasos:
     - Construcción de una Lista restringida de oraciones candidatas (LRC) viables, para la primera oración del vector solución se define de acuerdo a la lista de cobertura y posteriormente teniendo en cuenta la lista de cobertura y diversidad.
     - Selección aleatoria de una oración de la Lista restringida de oraciones candidatas, teniendo en cuenta la restricción de la longitud del resumen. Estos dos primeros pasos se repiten hasta obtener una nueva armonía.
     - Cada nueva armonía se optimiza con la búsqueda local codiciosa.
     - Evaluación del valor de aptitud de la nueva armonía, si es mejor que el de la mejor solución actual en el proceso iterativo se remplaza.

     Al finalizar el número de iteraciones definido, el algoritmo GRASP devuelve la mejor solución obtenida.

   - Aleatoria y GRASP: un porcentaje de la población se obtiene de forma aleatoria y otro porcentaje utilizando la técnica GRASP.

   - Aleatoria y dispersa: Esta forma de combinación genera inicialmente una población de dos veces el tamaño de la memoria armónica (2 * HMS) de forma aleatoria. A partir de este proceso inicial, se seleccionan dos grupos que harán parte de la población inicial definitiva así: el primer grupo corresponde al 25% de armonías generadas inicialmente, seleccionadas de mayor a menor valor de aptitud, y el segundo grupo se seleccionan del 75% de armonías restantes, seleccionando las armonías más diversas comparadas con las armonías del primer grupo. Cada armonía se optimiza con búsqueda local codiciosa.

2. Búsqueda Local codiciosa:

   - Intercambio de una y dos oraciones: En el proceso de la generación de un vecino se adicione una o dos oraciones que tengan la mayor similitud con respecto a la colección de documentos, y de la misma forma se elimine una o dos oraciones respectivamente con la menor similitud, controlando el número de oraciones en la armonía.

En este ciclo se presenta el diseño de la primera versión del algoritmo híbrido, denominado *LexRank-GBHS*, de acuerdo a la Figura 3.9, donde el algoritmo inicia ejecutando primero LexRank, después ejecuta un proceso de poda a las oraciones que entrega LexRank para continuar con la ejecución de GBHS quien genera el resumen final.

```
01  Iteraciones = número de iteraciones de Grasp
02  listaOraciones = lista completa de oraciones candidatas
03  listaLRC = lista restringida de candidatos
04  tamaLrc = Tamaño de la listaLRC
05  listaCoberturas =lista con el valor de cobertura de las oraciones
06  listaDiversidadOraciones = valor diversidad + valor de cobertura de cada oración.
07  MaxLen: Máxima Longitud para evolucionar en el resumen.

08  Procedimieto GRASP(Iteraciones=5, listaOraciones)
```

```
09  Inicio
10  mejorSolucion=nuevaSolucion()
11  tamaLrc = Longitud(listaOraciones) * 20%
12  listaCoberturas = calcularCoberturas(listaOraciones)

13  para i<-1 hasta iteraciones haga
14      solucion=nuevaSolucion()
15      longitudResumen = 0

16      //Construcción de una nueva solución
17      mientras (solucion incompleta) haga
18          //Construcción Lista Restringida de Candidatos
19          si (cantidad(frases en la solucion)=0)
20              listaLRC(j)<-listaCoberturas(j)
21          sino
22              listaDiversidadOraciones = coberturaOracion + diversidadOracion
23              para j<-1 hasta tamaLrc haga
24                  listaLRC(j)<-listaDiversidadOraciones(j)
25              fin para
26          fin si
27
28          //selección aleatoria de una oración de la LRC
29          oracion = seleccionAleatoria(listaLRC)
30          si (LongitudResumen<MaxLen)
31              solucion = solucion  + oracion
32              eliminarOracion(listaOraciones)
33          fin si
34      fin mientras

35      //Optimización búsqueda local codiciosa
36      nuevaSolucion = BusquedaLocal(solucion)
37
38      //Actualización de la solución si hay mejora
39      si (valorAptitud(nuevaSolucion) > valorAptitud(mejorSolucion))
40          mejorSolucion = nuevaSolucion
41      fin si
42  fin para
43  retornar mejorSolucion
44  Fin Procedimieto GRASP
```

Figura 3.9 Algoritmo Procedimiento Técnica GRASP

3.2.1 Afinación primera función objetivo

La configuración de la primera función objetivo incluye las características de cobertura y diversidad de acuerdo a la Ecuación (3.10). Después de ejecutar el proceso de afinamiento, el valor de las medidas Rouge para la función objetivo 1, se muestra en la Tabla 8.

Procedimiento

A. Inicialización aleatoria (IA).
B. Inicialización GRASP, intercambiando una oración en la optimización de GRASP (IG-1O).
C. Inicialización GRASP, intercambiando dos oraciones en la optimización de GRASP (IG-2O).

D. Inicialización aleatoria del 30% y el 70% restante de la población inicial mediante la técnica GRASP, intercambiando una oración en la optimización de GRASP (30IA-70IG-1O).
E. Inicialización aleatoria del 30% y el 70% restante de la población inicial mediante la técnica GRASP, intercambiando dos oraciones en la optimización de GRASP (30IA-70IG-2O).
F. Inicialización aleatoria del 50% y el otro 50% se seleccionan con las armonías más diversas comparadas con las ya existentes en la población, intercambiando una oración en la búsqueda local (50IA-50ID-1O).

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38982	0,07930	0,13709	0,40718	0,08971	0,14502
IG-1O	0,39028	**0,07962**	**0,13742**	0,40665	0,08989	0,14515
IG-2O	0,39007	0,07952	0,13742	0,40664	0,09001	0,14519
30IA-70IG-1O	**0,39061**	0,07951	0,13739	**0,40760**	0,08984	**0,14521**
30IA-70IG-2O	0,39033	0,07936	0,13707	0,40714	**0,09006**	0,14513
50IA-50ID-1O	0,39060	0,07937	0,13717	0,40748	0,08993	0,14510

Tabla 8 Medidas ROUGE afinación primera Función Objetivo

Para escoger cuál es la combinación que obtiene mejores resultados en ambos conjuntos de datos para el afinamiento de todas las funciones objetivo, se utiliza la Ecuación (3.17) que permite obtener un orden de clasificación unificado teniendo en cuenta la clasificación para cada uno de los procedimientos en cada medida ROUGE. De acuerdo a la Tabla 8 , la combinación *30IA-70IG-1O* obtiene los mejores resultados con valores de 0.32 para el parámetro alfa y 0.68 para el parámetro beta, según la función objetivo representada en la Ecuación (3.10).

$$Orden(Procedimiento) = \sum_{r=1}^{\mathrm{N}} \frac{\left((\mathrm{N} - r + 1)R_r\right)}{\mathrm{N}} \quad (3.17)$$

Donde N, indica el número de procedimientos a evaluar, $\boldsymbol{R_r}$ indica el número de veces que el procedimiento aparece en el *r*-ésimo procedimiento. El denominador corresponde al número de procedimientos con los que se hace la comparación.

3.2.2 Afinación segunda función objetivo

La configuración de la segunda función objetivo incluye las características de cobertura, diversidad y posición como se muestra en la Ecuación (3.13). El proceso de afinamiento incluye los mismos procedimientos utilizados en el afinamiento de la primera función objetivo. Después de ejecutar el proceso de afinamiento, el valor de las medidas Rouge para la función objetivo 2, se muestra en la Tabla 9.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38990	0,07923	0,13713	0,40674	**0,09003**	**0,14524**
IG-1O	0,38963	0,07945	0,13716	0,40650	0,08978	0,14498
IG-2O	0,38911	0,07923	0,13700	0,40647	0,08995	0,14509
30IA-70IG-1O	0,38996	0,07957	0,13727	**0,40675**	0,08994	0,14520
30IA-70IG-2O	0,38909	0,07907	0,13679	0,40640	0,08983	0,14496
50IA-50ID-1O	**0,39024**	**0,07979**	**0,13748**	0,40562	0,08959	0,14477

Tabla 9 Medidas ROUGE, afinación Segunda Función Objetivo

De acuerdo a la Tabla 9 y a la Ecuación (3.17), la combinación *30IA-70IG-1O* obtiene los mejores resultados con valores de 0.34 para el parámetro alfa, 0.61 para el parámetro beta, y 0.05 para el parámetro delta, según la función objetivo representada en la Ecuación (3.13).

3.2.3 Afinación tercera función objetivo

La configuración de la tercera función objetivo incluye las características de cobertura, diversidad, posición y longitud como se muestra en la Ecuación (3.15). El proceso de afinamiento incluye los mismos procedimientos utilizados en el afinamiento de la primera función objetivo. Después de ejecutar el proceso de afinamiento, el valor de las medidas Rouge para la función objetivo 3, se muestra en la Tabla 10.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38947	0,07961	0,13742	0,40512	0,08980	0,14492
IG-1O	0,38937	0,07958	0,13729	0,40537	0,08970	0,14488
IG-2O	**0,38979**	**0,07998**	**0,13766**	0,40438	0,08916	0,14441
30IA-70IG-1O	0,38922	0,07977	0,13739	**0,40604**	**0,08999**	0,14503
30IA-70IG-2O	0,38949	0,07987	0,13758	0,40512	0,08973	0,14484
50IA-50ID-1O	0,38926	0,07959	0,13734	0,40554	0,08955	**0,14504**

Tabla 10 Medidas ROUGE, afinación Tercera Función Objetivo

De acuerdo a la Tabla 10 y a la Ecuación (3.17), la combinación *30IA-70IG-1O* obtiene los mejores resultados con valores de 0.35 para el parámetro alfa, 0.53 para el parámetro beta, 0.07 para el parámetro delta y 0.05 para el parámetro sigma, según la función objetivo representada en la Ecuación (3.15).

3.2.4 Configuración definitiva de la función objetivo

Los resultados de la evaluación de las tres funciones objetivo afinadas se presentan en la Tabla 11, donde se observa que la primera función objetivo obtiene los mejores resultados en las medidas ROUGE-1 y ROUGE-SU4 sobre DUC2005 y DUC2006, y la tercera función

objetivo obtiene los mejores resultados para ROUGE-2 en DUC2005 Y DUC2006. De acuerdo a la Ecuación (3.17), que permite obtener una clasificación unificada para todas las medidas, se establece la *primera función* como la función objetivo definitiva, con sus correspondientes pesos como se presenta en la Ecuación (**3.18)** .

$$\textbf{Maximizar } F(x) = 0.32 * \text{Cobertura} + 0.68 * \text{Diversidad} \quad (3.18)$$

Función Objetivo	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
Primera Función	**0,39061**	0,07951	**0,13739**	**0,40760**	0,08984	**0,14521**
Segunda Función	0,38996	0,07957	0,13727	0,40675	0,08994	0,14520
Tercera Función	0,38922	**0,07977**	0,13739	0,40604	**0,08999**	0,14503

Tabla 11 Medidas Rouge mejores para cada función objetivo

3.2.5 Afinación de Parámetros

Para el proceso de afinamiento, se creó un conjunto de datos compuesto por 15 temas seleccionados aleatoriamente del conjunto de datos DUC2005 y 15 temas seleccionados aleatoriamente del DUC2006.

Debido a que el algoritmo híbrido contiene varios parámetros para su afinación, se utilizaron los Arreglos de cobertura [61] para seleccionar un grupo más reducido de combinación de parámetros. Luego se realizó la experimentación a través de 10 ejecuciones por cada combinación de parámetros; después se seleccionaron las 10 mejores configuraciones para continuar con la experimentación a través de 30 ejecuciones por configuración en todos los conjuntos de datos DUC2005 y DUC2006 para obtener la mejor configuración de parámetros como se muestra en la Tabla 12.

Parámetro	LexRank-GBHS
Alfa	0.32
Beta	0.68
Delta	0.00
Sigma	0.00
Threshold	0.04
dampingFactor	0.12
Error de Tolerancia	0.01
Hms	20
Hmcr	0.90
Po	0.90
Nop	10
Poda	0.30
Mlse	275

Tabla 12 Mejor configuración de parámetros Algoritmo LexRank-GBHS

3.2.6 Esquema del Algoritmo LexRank-GBHS

El algoritmo híbrido llamado LexRank-GBHS, como se muestra en la Figura 3.10, resaltando en negrilla donde se hicieron modificaciones, se describe de acuerdo a los siguientes pasos:

Paso 1: Inicialización de parámetros: En este paso se inicializan los parámetros asociados al problema, así como los parámetros de cada uno de los algoritmos base.

Paso 2: Ejecución de LexRank: En este paso se ejecuta el algoritmo base LexRank, a partir del cual se obtiene un vector con la ponderación de las oraciones ordenadas de lo más relevante a lo menos relevante. Se realizó una modificación al algoritmo iterativo Método de Potencia utilizado por LexRank [32], donde el vector inicial P_0 con todas las oraciones (nodos del grafo) ya no comienzan con el mismo peso (relevancia), modificándose para que el peso inicial de las oraciones se base en la cobertura (similitud entre la oración y la colección de los documentos) dividida por la suma de todas las similitudes de las oraciones.

Paso 3: Proceso de Poda: LexRank-GBHS define un parámetro de poda que permite eliminar las oraciones menos relevantes del vector entregado por LexRank.

Paso 4: Actualización de la matriz de similitudes: Después de realizado el proceso de poda, es necesario actualizar la matriz de similitud de coseno debido a la reducción de oraciones.

Paso 5: Ejecución de GBHS: El algoritmo GBHS inicia generando su población inicial seleccionando las armonías con un 70% mediante la técnica GRASP (Ver Figura 3.9) y el 30% de forma aleatoria, de acuerdo al parámetro tamaño de la memoria armónica HMS, donde cada nueva armonía se optimiza con búsqueda local codiciosa; después realiza su proceso evolutivo, terminando de acuerdo al parámetro del máximo número de evaluaciones de la función objetivo.

Paso 6: Generación del Resumen: al final el algoritmo híbrido LexRank-GBHS genera el resumen con las oraciones de la mejor armonía de la memoria armónica de GBHS, teniendo en cuenta la restricción del número máximo de palabras en el resumen. Las oraciones que harán parte del resumen se seleccionan de mayor a menor cobertura.

```
01  Hms=Tamaño memoria armónica, HM=Memoria armónica, Po=Probabilidad de Optimización,
02  Threshold= umbral Lexrank, Poda=Tasa para remover oraciones,
03  X(n)=Vector con las oraciones, Csm=Matriz de similitud de cosenos
04  Neof=Número de evaluaciones de la función objetivo
05  Mneof=Máximo número de evaluaciones de la función objetivo
06  Inicio                                  //Inicio LexRank-GBHS
07  InicioLexRank                           //Inicio LexRank
08      CalcularCsm(x(n))                   //Calcular matriz de similitud de cosenos
09      EliminarRD(Csm,Threshold)           //Eliminar relaciones débiles
10      NormalizarSM(Csm)                   //Normalizar Csm a Matriz Estocástica
11      IrreducibleAperiodica(DampingFactor) //Convertir matriz Csm a irreducible y aperiódica
12      MetodoPotencia(Csm,toleranceError)  //Calcular Vector propio con ponderación  de oraciones
13      OrdenarVectorPropio()               //Ordenar de mayor a menor valor el vector resultante
14  FinLexRank
15  //Aplicar la Poda
16  AplicarPoda(Poda)                       //Aplicar poda a vector resultante de LexRank
17  ActualizarMS()                          //Actualizar la matriz de similitudes
```

```
18  InicioGbhs                          //Inicio Gbhs
19      InicializarHM(Hms)              //Inicializar la memoria armónica, 30% aleatoria, el otro
                                        //70% mediante GRASP
20      EvaluarHM()                     //Evaluar la memoria armónica
21      OptimzarHM(Po)                  //Optimizar la memoria armónica búsqueda Local
22      Neof=1
23      Mientras (Neof<Mneof)
24          Improvisar()                //Improvisar la nueva armonía
25          EvaluarNH()                 //Evaluar la nueva armonía
26          OptimizarNH(Po)             //Optimizar la nueva armonía. Búsqueda Local Codiciosa
27          ActualizarHM()              //Actualizar la memoria armónica
28      Fin Mientras
29      Ordenar(HM))                    //Ordenar la memoria armónica
30  Fin Gbhs
31  GenerarResumen()
32  Fin
```

Figura 3.10 Algoritmo Versión 1 LexRank-GBHS

3.3 CICLO III: Diseño algoritmo híbrido versión 2

En este ciclo se presenta el diseño de la segunda versión del algoritmo híbrido, denominado *GBHS-LexRank*, de acuerdo a la Figura 3.11, donde el algoritmo inicia con la ejecución de GBHS, después se seleccionan las oraciones no repetidas de la última población para que LexRank las utilice como el conjunto total de oraciones para su ejecución y generación del resumen final.

Para el afinamiento de las funciones objetivo se tuvo en cuenta los mismos procedimientos de la sección 3.2.1.

3.3.1 Afinación primera función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra en la Tabla 13.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,37925	0,07475	0,13311	0,39316	0,08467	0,13978
IG-1O	0,37884	0,07477	0,13264	0,39316	0,08467	0,13978
IG-2O	0,37698	0,07468	0,13234	0,39316	**0,08606**	**0,14065**
30IA-70IG-1O	**0,37973**	0,07540	**0,13350**	**0,39484**	0,08586	0,14064
30IA-70IG-2O	0,37815	**0,07557**	0,13327	0,39272	0,08559	0,14040
50IA-50ID-1O	0,37884	0,07477	0,13264	0,39316	0,08467	0,13978

Tabla 13 Medidas ROUGE, afinación Primera Función Objetivo

De acuerdo a la Tabla 13 y a la Ecuación (3.17), la combinación *30IA-70IG-1O* obtuvo los mejores resultados con valores de 0.30 para el parámetro alfa y 0.70 para el parámetro beta de acuerdo a la función objetivo según la Ecuación (3.10).

3.3.2 Afinación segunda función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra la Tabla 14.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,37790	0,07527	0,13315	0,38751	0,08455	0,13820
IG-1O	**0,37931**	**0,07565**	**0,13363**	**0,39270**	**0,08528**	**0,14009**
IG-2O	0,37487	0,07461	0,13214	0,38776	0,08419	0,13853
30IA-70IG-1O	0,37842	0,07541	0,13352	0,39044	0,08492	0,13910
30IA-70IG-2O	0,37635	0,07508	0,13271	0,38841	0,08390	0,13845
50IA-50ID-1O	0,37468	0,07524	0,13216	0,38751	0,08455	0,13820

Tabla 14 Medidas ROUGE, afinación Segunda Función Objetivo

De acuerdo a la Tabla 14 y a la Ecuación (3.17), la combinación *IG-1O* obtuvo los mejores resultados con valores de 0.34 para el parámetro alfa, 0.61 para el parámetro beta y 0.05 para el parámetro delta de acuerdo a la función objetivo según en la Ecuación (3.13).

3.3.3 Afinación tercera función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra la Tabla 15.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	**0,37836**	0,07550	**0,13353**	0,38774	0,08456	0,13846
IG-1O	0,37734	0,07501	0,13297	**0,39069**	0,08503	0,13960
IG-2O	0,37392	0,07467	0,13189	0,38912	0,08417	0,13886
30IA-70IG-1O	0,37697	0,07529	0,13301	0,39063	**0,08592**	**0,13996**
30IA-70IG-2O	0,37376	0,07502	0,13195	0,38729	0,08493	0,13878
50IA-50ID-1O	0,37534	**0,07551**	0,13260	0,38565	0,08454	0,13782

Tabla 15 Medidas ROUGE, afinación Tercera Función Objetivo

De acuerdo a la Tabla 15 y a la Ecuación **(**3.17), la combinación *30IA-70IG-1O* obtuvo los mejores resultados con valores de 0.35 para el parámetro alfa, 0.53 para el parámetro beta, 0.07 para el parámetro delta y 0.05 para el parámetro sigma de acuerdo a la función objetivo según la Ecuación (3.15).

3.3.4 Configuración definitiva de la función objetivo

Los resultados de la evaluación de las tres funciones objetivo afinadas se presentan en la Tabla 16, donde se observa que la primera función obtiene los mejores resultados en todas las medidas ROUGE-1 y ROUGE-SU4 sobre DUC2006 y ROUGE-1 de DUC2005, la segunda función objetivo obtuvo los mejores resultados en ROUGE-2 y ROUGE-SU4 sobre DUC2005 y la tercera función objetivo en ROUGE-2 sobre DUC2006. De acuerdo a la Ecuación (3.17), que permite obtener una clasificación unificada para todas las medidas se establece la *primera función* como la función objetivo definitiva, con sus correspondientes pesos como se presenta en la Ecuación (3.18) .

$$\textbf{Maximizar } \mathrm{F(x)} = 0.30 * \mathrm{Cobertura} + 0.70 * \mathrm{Diversidad} \qquad (3.19)$$

Función Objetivo	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
Primera Función	**0,37973**	0,07540	0,13350	**0,39484**	0,08586	**0,14064**
Segunda Función	0,37931	**0,07565**	**0,13363**	0,39270	0,08528	0,14009
Tercera Función	0,37697	0,07529	0,13301	0,39063	**0,08592**	0,13996

Tabla 16 Resultados de afinación de las funciones objetivo

3.3.5 Afinación de Parámetros

La afinación de parámetros se realizó igual que en la versión 1. La mejor configuración de parámetros obtenida se muestra en la Tabla 17.

Parámetro	GBHS-LexRank
Alfa	0.3
Beta	0.7
Delta	0
Sigma	0
Threshold	0
dampingFactor	0.12
Error de Tolerancia	0.01
Hms	5
Hmcr	0.9
Po	0.5
Nop	10
Poda	0.00
Mlse	285

Tabla 17 Mejor configuración de parámetros Algoritmo GBHS-LexRank

3.3.6 Esquema del Algoritmo GBHS-LexRank

El algoritmo híbrido llamado GBHS-LexRank, como se muestra en la Figura 3.11, resaltando en negrilla donde se hicieron modificaciones, se describe de acuerdo a los siguientes pasos:

Paso 1: Inicialización de parámetros: En este paso se inicializan los parámetros asociados al problema, así como los parámetros de cada uno de los algoritmos base.

Paso 2: Ejecución de GBHS: El algoritmo GBHS inicia generando su población inicial seleccionando las armonías con un 70% mediante la técnica GRASP (Ver Figura 3.9) y el 30% de forma aleatoria, de acuerdo al parámetro tamaño de la memoria armónica HMS, donde cada nueva armonía se optimiza con búsqueda local codiciosa; después realiza su proceso evolutivo, terminando de acuerdo al parámetro del máximo número de evaluaciones de la función objetivo.

Paso 3: Seleccionar las oraciones no repetidas de la última población de la ejecución del algoritmo de GBHS para que continúen en la ejecución del algoritmo híbrido.

Paso 4: Actualización de la matriz de similitudes: Debido a que el número de oraciones se reduce, es necesario actualizar la matriz de similitud de coseno.

Paso 5: Ejecución de LexRank: En este paso se ejecuta el algoritmo base LexRank, tomando como insumos las oraciones obtenidas en el paso 3 y la matriz de similitudes del paso 4. Al terminar su ejecución se obtiene un vector con la ponderación de las oraciones ordenadas de lo más relevante a lo menos relevante. Se realizó la misma modificación al método de potencia como se mencionó en el esquema del algoritmo versión 1.

Paso 6: Generación del Resumen: El algoritmo híbrido GBHS-LexRank genera el resumen con las oraciones del vector resultante de LexRank. Las oraciones que harán parte del resumen se seleccionan de mayor a menor valor, teniendo en cuenta la restricción del número máximo de palabras en el resumen.

```
01  Hms=Tamaño memoria armónica, HM=Memoria armónica, Po=Probabilidad de Optimización,
02  Umbral= Utilizado por Lexrank, Poda=Tasa para remover oraciones,
03  X(n)=Vector con las oraciones, Csm=Matriz de similitud de cosenos
04  Neof=Número de evaluaciones de la función objetivo
05  Mneof=Máximo número de evaluaciones de la función objetivo
06  Inicio                              //Inicio GBHS-LexRank
07  InicioGbhs
08      InicializarHM(Hms)              //Inicializar la memoria armónica, 30% de forma
                                        //aletoria y 70% mediante GRASP
09      EvaluarHM()                     //Evaluar la memoria armónica
10      OptimizarHM(Po)                 //Optimizar la memoria armónica
11      Neof=1
12      Mientras (Neof<Mneof)
13          Improvisar()                //Improvisar la nueva armonía
14          EvaluarNH()                 //Evaluar la nueva armonía
15          OptimizarNH(Po)             //Optimizar la nueva armonía
16          ActualizarHM()              //Actualizar la memoria armónica
17      Fin Mientras
18      Ordenar(HM))                    //Ordenar la memoria armónica
```

```
19  Fin Gbhs
20  //seleccionar oraciones que pasan de Gbhs a LexRank
21  ObtenerOracionesDiferentes(Gbhs.HM())     //Obtener oraciones diferentes de la memoria armónica
22  ActulizarMatrizSimilitudes()              //Actualizar Matriz de Similitudes
23  InicioLexRank
24      CalcularCsm(x(n))                     //Calcular matriz de similitud de cosenos
25      EliminarRD(Csm,Umbral)                //Eliminar relaciones débiles
26      NormalizarSM(Csm)                     //Normalizar Csm a Matriz Estocástica
27      IrreducibleAperiodica(DampingFactor)  /Convertir matriz Csm a irreducible y aperiódica
28      MetodoPotencia(Csm,ErrorTolerancia)   //Calcular Vector propio con ponderación  de oraciones
29      OrdenarVectorPropio()                 //Ordenar de mayor a menor valor el vector resultante
30  Fin LexRank
31  GenerarResumen()
32  Fin GBHS-LexRank
```

Figura 3.11 Algoritmo Versión 2 GBHS-LexRank

3.4 CICLO IV: Diseño algoritmo híbrido versión 3

En este ciclo se presenta el diseño de la versión 3 del algoritmo híbrido, denominado *LexRank-GBHS-2Resumen*, de acuerdo a la Figura 3.12, el algoritmo inicia con la ejecución de LexRank, quien genera un resumen; después se realiza una poda a todas las oraciones que entrega LexRank para que GBHS se ejecute y genere otro resumen. Al finalizar el algoritmo híbrido selecciona las oraciones no repetidas de ambos resúmenes para generar el resumen final.

Para el afinamiento de las funciones objetivo se tuvo en cuenta los procedimientos de la sección 3.2.1 que realizan un intercambio de una oración en la búsqueda local, y se excluyen los que intercambian dos oraciones debido a que los mejores resultados de las dos versiones anteriores todas fueron con intercambio de una oración.

3.4.1 Afinación primera función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra en la Tabla 18.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38299	0,07757	0,13506	**0,39105**	**0,08560**	**0,14013**
IG	0,38295	0,07752	0,13493	0,39056	0,08549	0,14000
30IA-70IG-1O	**0,38328**	**0,07764**	**0,13518**	0,39079	0,08541	0,13991
50IA-50ID-1O	0,38292	0,07758	0,13509	0,39057	0,08554	0,13988

Tabla 18 Medidas ROUGE, afinación Primera Función Objetivo

De acuerdo a la Tabla 18 y a la Ecuación (3.17), la combinación *IA* obtuvo los mejores resultados con valores de 0.32 para el parámetro alfa y 0.68 para el parámetro beta de acuerdo a la función objetivo según la Ecuación (3.10).

3.4.2 Afinación segunda función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra la Tabla 19.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	**0,38302**	0,07762	0,13509	0,39033	0,08537	0,13971
IG	0,38300	**0,07772**	0,13513	0,39016	**0,08540**	0,13963
30IA-70IG-1O	0,38298	0,07767	**0,13515**	0,39047	0,08537	0,13974
50IA-50ID-1O	0,38273	0,07734	0,13492	**0,39049**	0,08538	**0,13976**

Tabla 19 Medidas ROUGE, afinación Segunda Función Objetivo

De acuerdo a la Tabla 19 y a la Ecuación (3.17), la combinación *30IA-70IG-1O* obtuvo los mejores resultados con valores de 0.34 para el parámetro alfa, 0.61 para el parámetro beta y 0.05 para el parámetro delta de acuerdo a la función objetivo según la Ecuación (3.13).

3.4.3 Afinación tercera función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra la Tabla 20.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38278	0,07743	0,13493	0,39038	0,08536	0,13978
IG	**0,38305**	**0,07753**	**0,13502**	0,39038	0,08536	0,13978
30IA-70IG-1O	0,38287	0,07748	0,13495	0,39042	0,08544	**0,13981**
50IA-50ID-1O	0,38299	0,07750	0,13496	**0,39053**	**0,08546**	0,13980

Tabla 20 Medidas ROUGE, afinación Tercera Función Objetivo

De acuerdo a la Tabla 20 y a la Ecuación (3.17), la combinación *50IA-50ID-1O* obtuvo los mejores resultados con valores de 0.35 para el parámetro alfa, 0.60 para el parámetro beta, 0.03 para el parámetro delta y 0.02 para el parámetro sigma de acuerdo a la función objetivo según la Ecuación (3.15).

3.4.4 Configuración definitiva de la función objetivo

Los resultados de la evaluación de las tres funciones objetivo afinadas se presentan en la Tabla 21, donde se observa que la primera función obtiene los mejores resultados en las medidas ROUGE-1 para DUC2005 y en todas las medidas ROUGE de DUC2006, la segunda función obtiene los mejores resultados en ROUGE-2 y ROUGE-SU4 para DUC2005. De acuerdo a la Ecuación (3.17), que permite obtener una clasificación unificada

para todas las medidas se establece la *primera función* como la función objetivo definitiva, con sus correspondientes pesos como se presenta en la Ecuación (3.18).

$$\textbf{Maximizar}\ \mathrm{F(x)} = 0.32 * \mathrm{Cobertura} + 0.68 * \mathrm{Diversidad} \quad (3.20)$$

Función Objetivo	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
Primera Función	**0,38299**	0,07757	0,13506	**0,39105**	**0,08560**	**0,14013**
Segunda Función	0,38298	**0,07767**	**0,13515**	0,39047	0,08537	0,13974
Tercera Función	0,38299	0,07750	0,13496	0,39053	0,08546	0,13980

Tabla 21 Resultados de afinación de las funciones objetivo

3.4.5 Afinación de Parámetros

Para el afinamiento de parámetros se realiza el mismo proceso explicado en la sección 3.2.5. En la Tabla 22, se presenta la mejor configuración de parámetros obtenida para la versión 3 del algoritmo.

Parámetro	LexRank-GBHS-2Resumen
Alfa	0.32
Beta	0.68
Delta	0.00
Sigma	0.00
Threshold	0.04
dampingFactor	0.12
Error de Tolerancia	0.01
Hms	5
Hmcr	0.60
Po	0.40
Nop	30
Poda	0,40
Mlse	265

Tabla 22 Mejor configuración parámetros LexRank-GBHS-2Resumen

3.4.6 Esquema del Algoritmo LexRank-GBHS-2Resumen

El algoritmo híbrido llamado LexRank-GBHS-2Resumen, como se muestra en la Figura 3.12, resaltando en negrilla donde se hicieron modificaciones, se describe de acuerdo a los siguientes pasos:

Paso 1: Inicialización de parámetros: En este paso se inicializan los parámetros asociados al problema, así como los parámetros de cada uno de los algoritmos base.

Paso 2: Ejecución de LexRank: En este paso se ejecuta el algoritmo base LexRank, a partir del cual se obtiene un vector con la ponderación de las oraciones ordenadas de lo más relevante a lo menos relevante. Se realizó una modificación al algoritmo iterativo Método

de Potencia utilizado por LexRank [32], donde el vector inicial P_0 con todas las oraciones (nodos del grafo) ya no comienzan con el mismo peso (relevancia), modificándose para que el peso inicial de las oraciones se base en la cobertura (similitud entre la oración y la colección de los documentos) dividida por la suma de todas las similitudes de las oraciones.

Paso 3: Generación ResumenLexRank: LexRank genera un resumen con las oraciones del vector resultante de la ejecución del algoritmo método de potencia. Las oraciones en el resumen se seleccionan de mayor a menor valor, teniendo en cuenta la restricción del número máximo de palabras en el resumen.

Paso 4: Proceso de Poda: LexRank-GBHS-2Resumen define un parámetro de poda que permite eliminar las oraciones menos relevantes del vector entregado por LexRank.

Paso 5: Actualización de la matriz de similitudes: Después de realizado el proceso de poda, es necesario actualizar la matriz de similitud de coseno debido a la reducción de oraciones.

Paso 6: Ejecución de GBHS: El algoritmo GBHS inicia generando su población inicial de manera aleatoria de acuerdo al parámetro tamaño de la memoria armónica HMS, donde cada nueva armonía se optimiza con búsqueda local codiciosa, después realiza su proceso evolutivo, terminando de acuerdo al parámetro del máximo número de evaluaciones de la función objetivo.

Paso 7: Generación Resumen GBHS: El algoritmo híbrido LexRank-GBHS-2Resumen al finalizar la ejecución de GBHS, genera un resumen con las oraciones de la mejor armonía de la memoria armónica de GBHS, teniendo en cuenta la restricción del número máximo de palabras en el resumen.

Paso 8: Generación Resumen Final: LexRank-GBHS-2Resumen genera el resumen final a partir de las oraciones de los dos resúmenes generados en el paso 3 y paso 7. Las oraciones que harán parte del resumen final se seleccionan primero teniendo en cuenta las oraciones que coinciden en los dos resúmenes y las otras de mayor a menor diversidad.

```
01  Hms=Tamaño memoria armónica, HM=Memoria armónica, Po=Probabilidad de Optimización,
02  Umbral= Utilizado por Lexrank, Poda=Tasa para remover oraciones,
03  X(n)=Vector con las oraciones, Csm=Matriz de similitud de cosenos
04  Neof=Número de evaluaciones de la función objetivo
05  Mneof=Máximo número de evaluaciones de la función objetivo
06  Inicio                                          //Inicio LexRank.GBHS-2Resumen
07  InicioLexRank                                   //Inicio LexRank
08      CalcularCsm(x(n))                           //Calcular matriz de similitu de cosenos
09      EliminarRD(Csm,Umbral)                      //Eliminar relaciones débiles
10      NormalizarSM(Csm)                           //Normalizar Csm a Matriz Estocástica
11      IrreducibleAperiodica(DampingFactor)        /Convertir matriz Csm a irreducible y aperiódica
12      MetodoPotencia(Csm,toleranceError)          //Calcular Vector propio con ponderación de
                                                    //oraciones
13      OrdenarVectorPropio()                       //Ordenar de mayor a menor vector resultante
14      GenerarResumenLexRank()                     //LexRank genera un resumen
15  FinLexRank
16  //Aplicar la Poda
17  AplicarPoda(Poda)                               //Aplicar poda a vector resultante de LexRank
18  ActualizarMS()                                  //Actualizar la matriz de similitudes
19  InicioGbhs                                      //Inicio Gbhs
```

```
20      InicializarHM(Hms)                          //Inicializar la memoria armónica, 100% de
                                                    //forma aleatoria.
21      EvaluarHM()                                 //Evaluar la memoria armónica
22      OptimizarHM(Po)                             //Optimizar la memoria armónica
23      Neof=1
24      Mientras (Neof<Mneof)
25          Improvisar()                            //Improvisar la nueva armonía
26          EvaluarNH()                             //Evaluar la nueva armonía
27          OptimizarNH(Po)                         //Optimizar la nueva armonía
28          ActualizarHM()                          //Actualizar la memoria armónica
29      Fin Mientras
30      Ordenar(HM))                                //Ordenar la memoria armónica
31  Fin Gbhs
32  GenerarResumenHibrido()                         //Generar otro resumen
33  GenerarResumen(ResLexRank, ResHibrido)          //Generar resumen a partir de los dos resúmenes
34  Fin
```

Figura 3.12 Algoritmo Versión 3 LexRank-GBHS-2Resumen

3.5 CICLO V: Diseño algoritmo híbrido versión 4

En este ciclo se presenta el diseño de la versión 4 del algoritmo híbrido, denominado *GBHS-LexRank-2Resumen*, de acuerdo a la Figura 3.13, el algoritmo inicia con la ejecución de GBHS quien genera un resumen, después el algortimo híbrido selecciona las oraciones no repetidas de la última población de GBHS para que LexRank genere otro resumen. Al finalizar el algoritmo híbrido selecciona las oraciones no repetidas de ambos resúmenes para generar el resumen final. Para la selección de oraciones del resumen final, primero se selecionan las oraciones que coinciden en los dos resúmenes, y las otras oraciones se seleccionan de acuerdo a un criterio de diversidad.

Para el afinamiento de las funciones objetivo se realizó igual que para el ciclo IV, solo se tienen en cuenta algunos procedimientos.

3.5.1 Afinación primera función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra en la Tabla 23.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38461	**0,07705**	0,13527	0,39909	0,08762	0,14302
IG	0,38400	0,07662	0,13501	0,39928	0,08733	0,14299
30IA-70IG-1O	**0,38498**	0,07690	**0,13541**	**0,39928**	**0,08788**	**0,14312**
50IA-50ID-1O	0,38454	0,07683	0,13527	0,39886	0,08744	0,14284

Tabla 23 Medidas ROUGE, afinación Primera Función Objetivo

De acuerdo a la Tabla 23 y a la Ecuación (3.17), la combinación *30IA-70IG-1O* obtuvo los mejores resultados con valores de 0.35 para el parámetro alfa y 0.65 para el parámetro beta de acuerdo a la función objetivo según la Ecuación (3.10).

3.5.2 Afinación segunda función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra la Tabla 24.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38364	0,07634	0,13469	0,39936	0,08722	0,14285
IG	0,38429	0,07645	0,13511	**0,39937**	0,08753	0,14302
30IA-70IG-1O	**0,38449**	**0,07661**	**0,13523**	0,39930	**0,08761**	**0,14307**
50IA-50ID-1O	0,38430	0,07653	0,13506	0,39888	0,08758	0,14293

Tabla 24 Medidas ROUGE, afinación Segunda Función Objetivo

De acuerdo a la Tabla 24 y a la Ecuación (3.17), la combinación *30IA-70IG-1O* obtuvo los mejores resultados con valores de 0.34 para el parámetro alfa, 0.61 para el parámetro beta y 0.05 para el parámetro delta de acuerdo a la función objetivo según la Ecuación (3.13).

3.5.3 Afinación tercera función objetivo

Después de ejecutar el proceso de afinamiento, la combinación de pesos que obtuvo mejores resultados se muestra la Tabla 25.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38249	0,07641	0,13459	0,39561	0,08734	0,14192
IG	**0,38351**	**0,07659**	**0,13506**	**0,39837**	0,08774	**0,14298**
30IA-70IG-1O	0,38336	0,07640	0,13490	0,39818	**0,08784**	0,14288
50IA-50ID-1O	0,38313	0,07657	0,13489	0,39621	0,08746	0,14214

Tabla 25 Medidas ROUGE, afinación Tercera Función Objetivo

De acuerdo a la Tabla 25Tabla 8 y a la Ecuación (3.17), la combinación *IG* obtuvo los mejores resultados, con valores de 0.35 para el parámetro alfa, 0.53 para el parámetro beta, 0.07 para el parámetro delta y 0.05 para el parámetro sigma, de acuerdo a la función objetivo según la Ecuación (3.15).

3.5.4 Configuración definitiva de la función objetivo

Los resultados de la evaluación de las tres funciones objetivo afinadas se presentan en la Tabla 26, donde se observa que la primera función obtiene los mejores resultados en todas las medidas ROUGE excepto en ROUGE 1 de DUC2006, donde la segunda función obtuvo el mejor resultado. De acuerdo a la Ecuación (3.17), que permite obtener una clasificación unificada para todas las medidas se establece la *primera función* como la función objetivo definitiva, con sus correspondientes pesos como se presenta en la Ecuación (3.21).

$$\textbf{Maximizar } F(x) = 0.35 * \text{Cobertura} + 0.65 * \text{Diversidad} \quad (3.21)$$

Función Objetivo	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
Primera Función	**0,38498**	**0,07690**	**0,13541**	0,39928	**0,08788**	**0,14312**
Segunda Función	0,38449	0,07661	0,13523	**0,39930**	0,08761	0,14307
Tercera Función	0,38351	0,07659	0,13506	0,39837	0,08774	0,14298

Tabla 26 Resultados de afinación de las funciones objetivo

3.5.5 Afinación de Parámetros

Para el afinamiento de parámetros se realiza el mismo proceso explicado en la sección 3.2.5. En la Tabla 27, se presenta la mejor configuración de parámetros obtenida para la versión 4 del algoritmo.

.

Parámetro	GBHS-LexRank-2Resumen
Alfa	0.35
Beta	0.65
Delta	0.00
Sigma	0.00
Threshold	0.06
dampingFactor	0.06
Error de Tolerancia	0.01
Hms	5
Hmcr	0.60
Po	0.30
Nop	20
Poda	0,00
Mlse	265

Tabla 27 Mejor configuración parámetros GBHS-LexRank-2Resumen

3.5.6 Esquema del Algoritmo GBHS-LexRank-2Resumen

El algoritmo híbrido llamado GBHS-LexRank-2Resumen, como se muestra en la Figura 3.13, resaltando en negrilla donde se hicieron modificaciones, se describe de acuerdo a los siguientes pasos:

Paso 1: Inicialización de parámetros: En este paso se inicializan los parámetros asociados al problema, así como los parámetros de cada uno de los algoritmos base.

Paso 2: Ejecución de GBHS: El algoritmo GBHS inicia generando su población inicial seleccionando las armonías con un 70% mediante la técnica GRASP (Ver Figura 3.9) y el 30% de forma aleatoria, de acuerdo al parámetro tamaño de la memoria armónica HMS, donde cada nueva armonía se optimiza con búsqueda local codiciosa; después realiza su

proceso evolutivo, terminando de acuerdo al parámetro del máximo número de evaluaciones de la función objetivo.

Paso 3: Generación Resumen GBHS: El algoritmo híbrido GBHS-LexRank-2Resumen al finalizar la ejecución de GBHS, genera un resumen con las oraciones de la mejor armonía de la memoria armónica de GBHS, teniendo en cuenta la restricción del número máximo de palabras en el resumen.

Paso 4: Seleccionar las oraciones no repetidas de la última población de la ejecución del algoritmo de GBHS para que continúen en la ejecución del algoritmo híbrido.

Paso 5: Actualización de la matriz de similitudes: Debido a que el número de oraciones se reduce, es necesario actualizar la matriz de similitud de coseno.

Paso 6: Ejecución de LexRank: En este paso se ejecuta el algoritmo base LexRank, tomando como insumos las oraciones obtenidas en el paso 4 y la matriz de similitudes del paso 5. Al terminar su ejecución se obtiene un vector con la ponderación de las oraciones ordenadas de lo más relevante a lo menos relevante. Se realizó una modificación al algoritmo iterativo método de potencia mencionado anteriormente en el esquema del algoritmo versión 1.

Paso 7: Generación Resumen LexRank: El algoritmo híbrido al finalizar la ejecución de LexRank genera un resumen con las oraciones del vector resultante de la ejecución del algoritmo método de potencia de LexRank. Las oraciones en el resumen se seleccionan de mayor a menor valor, teniendo en cuenta la restricción del número máximo de palabras en el resumen.

Paso 8: Generación Resumen Final: El algoritmo híbrido GBHS-LexRank-2Resumen, genera el resumen final a partir de las oraciones de los dos resúmenes generados en el paso 3 y paso 7. Las oraciones que harán parte del resumen final se seleccionan primero teniendo en cuenta las oraciones que coinciden en los dos resúmenes y las otras se seleccionan de mayor a menor diversidad.

```
01  Hms=Tamaño memoria armónica, HM=Memoria armónica, Po=Probabilidad de Optimización,
02  Umbral= Utilizado por Lexrank, Poda=Tasa para remover oraciones,
03  X(n)=Vector con las oraciones, Csm=Matriz de similitud de cosenos
04  Neof=Número de evaluaciones de la función objetivo
05  Mneof=Máximo número de evaluaciones de la función objetivo
06  Inicio                                  //Inicio GBHS-LexRank-2Resumen
07  InicioGbhs
08      InicializarHM (Hms)                 //Inicializar la memoria armónica, 30% aleatoria,
                                            //70% mediante GRASP
09      EvaluarHM()                         //Evaluar la memoria armónica
10      OptimizarHM(Po)                     //Optimizar la memoria armónica
11      Neof=1
12      Mientras (Neof<Mneof)
13          Improvisar()                    //Improvisar la nueva armonía
14          EvaluarNH()                     //Evaluar la nueva armonía
15          OptimizarNH(Po)                 //Optimizar la nueva armonía
16          ActualizarHM()                  //Actualizar la memoria armónica
17      Fin Mientras
18      Ordenar(HM))                        //Ordenar la memoria armónica
```

```
19      GenerarResumenGbhs(HM(0))          //Generar resumen
19  Fin Gbhs
20  //seleccionar oraciones que pasan de Gbhs a LexRank
21  ObtenerOracionesDiferentes(Gbhs.HM())  //Obtener oraciones diferentes de la memoria armónica
22  ActulizarMatrizSimilitudes()           //Actualizar Matriz de Similitudes
23  InicioLexRank
24      CalcularCsm(x(n))                  //Calcular matriz de similitud de cosenos
25      EliminarRD(Csm,Umbral)             //Eliminar relaciones débiles
26      NormalizarSM(Csm)                  //Normalizar Csm a Matriz Estocástica
27      IrreducibleAperiodica(DampingFactor)  /Convertir matriz Csm a irreducible y aperiódica
28      MetodoPotencia(Csm,ErrorTolerancia)  //Calcular Vector propio con ponderación  de oraciones
29      OrdenarVectorPropio()              //Ordenar de mayor a menor valor el vector resultante
30  Fin LexRank
31  GenerarResumenHíbrido()                //El algoritmo híbrido genera un resumen
33  GenerarResumen(ResGBHS, ResHíbrido)    //Generar resumen final a partir de los dos resúmenes
32  Fin
```

Figura 3.13 Algoritmo Versión 4 GBHS-LexRank-2Resumen

3.6 CICLO VI: Diseño algoritmo híbrido versión 5

En este ciclo se presenta el diseño de la versión 5 del algoritmo híbrido denominado *LexRank-GBHS-Paralelo,* de acuerdo a la Figura 3.14, el algoritmo ejecuta los dos algoritmos base LexRank y GBHS por separado, donde cada uno genera un resumen. Al finalizar el algoritmo híbrido selecciona las oraciones de los dos resumenes verificando que no se repitan para generar el resumen final.

Para el afinamiento de las funciones objetivo se realizó igual que para el ciclo IV, solo se tienen en cuenta algunos procedimientos.

3.6.1 Afinación primera función objetivo

Después de ejecutar el proceso de afinamiento, el valor de las medidas Rouge para la función objetivo 1, se muestra en la Tabla 28.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38269	0,07646	0,13435	0,39229	0,08493	0,14024
IG	0,38209	0,07652	0,13421	0,39215	0,08522	0,14028
30IA-70IG-1O	**0,38217**	**0,07668**	**0,13421**	**0,39233**	**0,08522**	**0,14040**
50IA-50ID-1O	0,38261	0,07638	0,13420	0,39240	0,08504	0,14028

Tabla 28 Medidas ROUGE afinación primera Función Objetivo

De acuerdo a la Tabla 28 y a la Ecuación (3.17), la combinación *30IA-70IG-1O* obtuvo los mejores resultados, con valores de 0.32 para el parámetro alfa y 0.68 para el parámetro beta de acuerdo a la función objetivo según la Ecuación (3.10).

3.6.2 Afinación segunda función objetivo

Después de ejecutar el proceso de afinamiento, el valor de las medidas Rouge para la función objetivo 2, se muestra en la Tabla 29.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	**0,38200**	**0,07685**	**0,13432**	**0,39178**	**0,08512**	**0,14011**
IG	0,38187	0,07663	0,13423	0,39188	0,08515	0,14006
30IA-70IG-1O	0,38155	0,07664	0,13414	0,39159	0,08518	0,13999
50IA-50ID-1O	0,38189	0,07665	0,13411	0,39192	0,08524	0,14015

Tabla 29 Medidas ROUGE afinación Segunda Función Objetivo

De acuerdo a la Tabla 29 y a la Ecuación (3.17), la combinación *IA* obtuvo los mejores resultados, con valores de 0.34 para el parámetro alfa, 0.61 para el parámetro beta y 0.05 para el parámetro delta, de acuerdo a la función objetivo según la Ecuación (3.13).

3.6.3 Afinación tercera función objetivo

Después de ejecutar el proceso de afinamiento, el valor de las medidas Rouge para la función objetivo 3, se muestra en la Tabla 30.

Procedimiento	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
IA	0,38197	0,07670	0,13424	0,39233	0,08510	0,14015
IG	**0,38247**	**0,07656**	**0,13433**	**0,39231**	**0,08503**	**0,14017**
30IA-70IG-1O	0,38192	0,07688	0,13433	0,39116	0,08489	0,13982
50IA-50ID-1O	0,38188	0,07673	0,13418	0,39185	0,08515	0,14016

Tabla 30 Medidas ROUGE afinación Tercera Función Objetivo

De acuerdo a la Tabla 30 y a la Ecuación (3.17), la combinación *IG* obtuvo los mejores resultados, con valores de 0.35 para el parámetro alfa, 0.53 para el parámetro beta, 0.07 para el parámetro delta y 0.05 para el parámetro sigma, de acuerdo a la función objetivo según la Ecuación (3.15).

3.6.4 Configuración definitiva de la función objetivo

Los resultados de la evaluación de las tres funciones objetivo afinadas se presentan en la Tabla 31, donde se observa que la primera función objetivo obtiene los mejores resultados en las medidas ROUGE sobre DUC2006, para DUC2005 la tercera función objetivo obtiene mejores resultados en ROUGE1 y ROUGE-SU4, y la segunda función en ROUGE-2. De acuerdo a la Ecuación (3.17), que permite obtener una clasificación unificada para todas las medidas se establece la *primera función* como la función objetivo definitiva, con sus correspondientes pesos como se presenta en la Ecuación (3.22) .

$$\textbf{Maximizar}\ \mathrm{F(x)} = 0.32 * \mathrm{Cobertura} + 0.68 * \mathrm{Diversidad} \qquad (3.22)$$

Función Objetivo	DUC2005			DUC2006		
	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-SU4 Recall
Primera Función	0,38217	0,07668	0,13421	**0,39233**	**0,08522**	**0,14040**
Segunda Función	0,38200	**0,07685**	0,13432	0,39178	0,08512	0,14011
Tercera Función	**0,38247**	0,07656	**0,13433**	0,39231	0,08503	0,14017

Tabla 31 Resultados de afinación de las funciones objetivo

3.6.5 Afinación de Parámetros

Para el afinamiento de parámetros se realiza el mismo proceso explicado en la sección 3.2.5. En la Tabla 32, se presenta la mejor configuración de parámetros obtenida para la versión 5 del algoritmo.

Parámetro	LexRank-GBHS-Paralelo
Alfa	0.32
Beta	0.68
Delta	0.00
Sigma	0.00
Threshold	0.02
dampingFactor	0.08
Error de Tolerancia	0.01
Hms	70
Hmcr	0.30
Po	0.20
Nop	10
Poda	0.0
Mlse	275

Tabla 32 Mejor configuración de parámetros LexRank-GBHS-Paralelo

3.6.6 Esquema del Algoritmo Versión LexRank-GBHS-Paralelo

El algoritmo híbrido llamado *LexRank-GBHS-Paralelo*, como se muestra en la Figura 3.14, realiza la ejecución de LexRank y GBHS en paralelo, generando cada uno un resumen. La generación de la población inicial en GBHS se obtiene un 70% mediante la técnica GRASP y el otro 30% de forma aleatoria. Al finalizar genera el resumen final a partir de las oraciones de los dos resúmenes. Las oraciones que harán parte del resumen final se seleccionan primero teniendo en cuenta las oraciones que coinciden en los dos resúmenes y las otras de mayor a menor diversidad.

```
01  Hms=Tamaño memoria armónica, HM=Memoria armónica, Po=Probabilidad de Optimización,
02  Umbral= Utilizado por Lexrank, Poda=Tasa para remover oraciones,
03  X(n)=Vector con las oraciones, Csm=Matriz de similitud de cosenos
04  Neof=Número de evaluaciones de la función objetivo
05  Mneof=Máximo número de evaluaciones de la función objetivo
06  Inicio                                  //Inicio LexRank-GBHS-Paralelo
07  InicioLexRank                           //Inicio LexRank
08        CalcularCsm(x(n))                 //Calcular matriz de similitud de cosenos
```

```
09        EliminarRD(Csm,Umbral)                  //Eliminar relaciones débiles
10        NormalizarSM(Csm)                       //Normalizar Csm a Matriz Estocástica
11        IrreducibleAperiodica(DampingFactor)    //Convertir matriz Csm a irreducible y aperiódica
12        MetodoPotencia(Csm,toleranceError)      //Calcular Vector propio con ponderación  de oraciones
13        OrdenarVectorPropio()                   //Ordenar de mayor a menor valor el vector resultante
14        GenerarResumenLexRank()                 //LexRank genera un resumen
15    FinLexRank
16    InicioGbhs                                  //Inicio Gbhs
17        InicializarHM (Hms)                     //Inicializar la memoria armónica, 30% aleatoria, y el
                                                  //70% mediante GRASP
18        EvaluarHM()                             //Evaluar la memoria armónica
19        OptimizarHM(Po)                         //Optimizar la memoria armónica
20        Neof=1
21        Mientras (Neof<Mneof)
22            Improvisar()                        //Improvisar la nueva armonía
23            EvaluarNH()                         //Evaluar la nueva armonía
24            OptimizarNH(Po)                     //Optimizar la nueva armonía
25            ActualizarHM()                      //Actualizar la memoria armónica
26        Fin Mientras
27        Ordenar(HM))                            //Ordenar la memoria armónica
28        GenerarResumenGBHS()                    //Gbhs genera su resumen
29    Fin Gbhs
30    GenerarResumen(ResLexRank, ResGBHS)         //Generar resumen final
31    Fin
```

Figura 3.14 Algoritmo Versión 5 LexRank-GBHS-Paralelo

Finalizados los cinco ciclos de diseño se presenta en la Tabla 33, el mejor resultado obtenido por el algoritmo propuesto en cada ciclo, donde se observa que el algoritmo con mejor desempeño fue *LexRank-GBHS* debido a que mostró los mejores resultados en todas las medidas de DUC2005 y DUC2006, y por lo tanto se selecciona como propuesta definitiva.

Algoritmo	DUC2005			DUC2006		
	R1R	R2R	RSU4R	R1R	R2R	RSU4R
LexRank-GBHS	**0,39061**	**0,07951**	**0,13739**	**0,40760**	**0,08984**	**0,14521**
GBHS-LexRank	0.37973	0.07540	0.13350	0.39484	0.08586	0.14064
LexRank-GBHS-2Resumen	0,38299	0,07757	0,13506	0,39105	0,08560	0,14013
GBHS-LexRank-2Resumen	0,38498	0,07690	0,13541	0,39928	0,08788	0,14312
LexRank-GBHS-Paralelo	0,38217	0,07668	0,13421	0,39233	0,08522	0,14040

Tabla 33 Mejor Resultado de Cada Ciclo

Capítulo 4

4 ALGORITMO PROPUESTO: LEXRANK-GBHS

En este capítulo se describe el nuevo algoritmo híbrido propuesto llamado LexRank-GBHS, para la generación automática de resúmenes extractivos de múltiples documentos. En las siguientes secciones se presenta: la representación de las soluciones escogida, la función objetivo que se definió para la optimización, la descripción de los componentes del algoritmo y el seudocódigo, además evaluación de calidad de los resúmenes generados.

4.1 REPRESENTACIÓN DE LAS SOLUCIONES

En el algoritmo híbrido LexRank-GBHS, la representación de una solución es realizada mediante codificación binaria. La codificación binaria utiliza un vector binario, en este caso, el tamaño del vector es igual al número de oraciones (*n*) que componen la colección de documentos representado como, $\{s_1, s_2, \dots, s_n\}$, de acuerdo a la Ecuación (4.1), y cada elemento del vector toma el valor de uno o cero para representar la presencia o ausencia de una oración en el resumen. Por ejemplo, si el número total de oraciones de la colección de documentos es de doce oraciones $(n = 12)$, la representación de un vector solución puede ser la siguiente $[1,0,1,0,1,0,1,0,1,1,0,0]$, en este caso, indicando que las oraciones primera, tercera, quinta, séptima, novena y décima están habilitadas para ser parte de la solución candidata (resumen candidato) [57, 62].

$$X_i = \{s_1, s_2, \dots, s_n\} \tag{4.1}$$

Donde $X_i \in \{0,1\}$, n es el número total de oraciones de la colección de documentos, e $i = 1 \dots n$.

4.2 FUNCIÓN OBJETIVO

La definición de la función objetivo es una de las tareas más importantes en la construcción de un algoritmo, ya que es la que guía la exploración y explotación del espacio de búsqueda. La función objetivo a optimizar para el algoritmo LexRank-GBHS está compuesta por las características de: *Cobertura*, que permite seleccionar las oraciones más relevantes contenidas en la colección de documentos con la menor pérdida de información, y *Diversidad*, que busca que el resumen generado no contenga información repetida, tratándose de resúmenes de múltiples documentos. Estas características fueron calculadas con las Ecuaciones de cobertura (3.1) y diversidad (3.2), dando como resultado la función objetivo que se observa en la Ecuación (4.2). Además, la función objetivo presenta una restricción sobre el número máximo de palabras permitidas en el resumen de acuerdo a la Ecuación (3.11) y otra restricción para que la suma de los coeficientes de las características sea igual a uno, ver Ecuación (3.12). El coeficiente $\boldsymbol{\alpha}$ permite darle un peso a la característica de Cobertura (0.32) y el coeficiente $\boldsymbol{\beta}$ (0.68) a la característica de Diversidad.

$$\text{Maximizar F(x)} = 0.32 * \left(\frac{\sum_{i=1}^{o} sim(D, s_i)}{o}\right) + 0.68 * \left(\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(1 - sim(s_i, s_j)\right)}{(n * (n-1)/2)}\right) \quad (4.2)$$

Donde *o* es el número de oraciones seleccionadas en el resumen candidato, D representa el centroide de la colección de documentos, $sim(D, s_i)$ es la similitud de coseno entre el vector centroide de la colección de documentos y el vector que representa la oración (si); n es la cantidad de oraciones que hay en el resumen, y $sim(s_i, s_j)$ es la similitud de coseno entre las dos oraciones.

4.3 ADAPTACIONES DEL ALGORITMO

A continuación se presenta el esquema general del Algoritmo propuesto denominado *LexRank-GBHS* y las modificaciones que se realizaron a los algoritmos base LexRank y GBHS, que hacen parte del algoritmo híbrido propuesto para la generación automática de resumes de texto para múltiples documentos.

Paso 1: Inicialización de parámetros: En este paso se inicializan los parámetros asociados al problema, así como los parámetros de cada uno de los algoritmos base.

Paso 2: Ejecución de LexRank: En este paso se ejecuta el algoritmo base LexRank, a partir del cual se obtiene un vector con la ponderación de las oraciones ordenadas de lo más relevante a lo menos relevante. Se realizó una modificación al algoritmo iterativo Método de Potencia utilizado por LexRank [32] como se explica en la sección 4.3.1.

Paso 3: Proceso de Poda: LexRank-GBHS define un parámetro de poda que permite eliminar las oraciones menos relevantes del vector entregado por LexRank.

Paso 4: Actualización de la matriz de similitudes: Después de realizado el proceso de poda, es necesario actualizar la matriz de similitud de coseno debido a la reducción de oraciones.

Paso 5: Ejecución de GBHS: El algoritmo GBHS inicia generando su población inicial seleccionando las armonías con un 70% mediante la técnica GRASP (Ver Figura 3.9) y el 30% de forma aleatoria, de acuerdo al parámetro tamaño de la memoria armónica HMS, donde cada nueva armonía se optimiza con búsqueda local codiciosa; después realiza su proceso evolutivo, terminando de acuerdo al parámetro del máximo número de evaluaciones de la función objetivo.

Paso 6: Generación del Resumen: al final el algoritmo híbrido LexRank-GBHS genera el resumen con las oraciones de la mejor armonía de la memoria armónica de GBHS, teniendo en cuenta la restricción del número máximo de palabras en el resumen. Las oraciones que harán parte del resumen se seleccionan de mayor a menor diversidad.

En la Figura 4.1, se presenta el algoritmo híbrido propuesto LexRank-GBHS.

```
01  Hms=Tamaño memoria armónica, HM=Memoria armónica, Po=Probabilidad de Optimización,
02  Threshold= umbral Lexrank, Poda=Tasa para remover oraciones,
03  X(n)=Vector con las oraciones, Csm=Matriz de similitud de cosenos
04  Neof=Número de evaluaciones de la función objetivo
```

```
05  Mneof=Máximo número de evaluaciones de la función objetivo
06  Inicio                                  //Inicio LexRank-GBHS
07  InicioLexRank                           //Inicio LexRank
08      CalcularCsm(x(n))                   //Calcular matriz de similitud de cosenos
09      EliminarRD(Csm,Threshold)           //Eliminar relaciones débiles
10      NormalizarSM(Csm)                   //Normalizar Csm a Matriz Estocástica
11      IrreducibleAperiodica(DampingFactor) //Convertir matriz Csm a irreducible y aperiódica
12      MetodoPotencia(Csm,toleranceError)  //Calcular Vector propio con ponderación  de oraciones
13      OrdenarVectorPropio()               //Ordenar de mayor a menor valor el vector resultante
14  FinLexRank
15  //Aplicar la Poda
16  AplicarPoda(Poda)                       //Aplicar poda a vector resultante de LexRank
17  ActualizarMS()                          //Actualizar la matriz de similitudes
18  InicioGbhs                              //Inicio Gbhs
19      InicializarHM(Hms)                  //Inicializar la memoria armónica, 30% aleatoria, el otro
                                            //70% mediante GRASP
20      EvaluarHM()                         //Evaluar la memoria armónica
21      OptimizarHM(Po)                     //Optimizar la memoria armónica. Búsqueda Local
22      Neof=1
23      Mientras (Neof<Mneof)
24          Improvisar()                    //Improvisar la nueva armonía
25          EvaluarNH()                     //Evaluar la nueva armonía
26          OptimizarNH(Po)                 //Optimizar la nueva armonía. Búsqueda Local Codiciosa
27          ActualizarHM()                  //Actualizar la memoria armónica
28      Fin Mientras
29      Ordenar(HM))                        //Ordenar la memoria armónica
30  Fin Gbhs
31  GenerarResumen()
32  Fin
```

Figura 4.1 Algoritmo propuesto LexRank-GBHS

Las adaptaciones realizadas que se realizaron a los algoritmos base se presentan a continuación:

4.3.1 Algoritmo base LexRank

El algoritmo Método de Potencia es el encargado de generar el vector resultante con la puntuación de las oraciones. La modificación se hizo en la forma de iniciar el vector que representa la distribución estacionaria y que al final hace referencia a los pesos de cada una de las oraciones. En la versión original el vector inicia con una distribución uniforme de acuerdo a la Ecuación (4.3), y en la nueva versión se incluye conocimiento del problema al utilizar la similitud de coseno de las oraciones para iniciar el vector de acuerdo a la Ecuación (4.4). La modificación se realiza para que el peso inicial de todas las oraciones se obtenga a partir de la similitud de cobertura entre la oración y la colección de los documentos dividido por la suma de todas las similitudes de todas las oraciones, que permita que algoritmo converja de una manera más eficiente a la solución.

$$p_i = \sum_{i=0}^{N-1} \frac{1}{N} \tag{4.3}$$

$$p_i = \sum_{i=0}^{N-1} \frac{sim(s_i)}{\left(\sum_{j=0}^{N-1} sim(s_j)\right)} \tag{4.4}$$

En las ecuaciones N, es el número total de oraciones, $sim(s_i)$ y $sim(s_j)$ es la similitud de coseno de la oración i, y j respectivamente.

4.3.2 Por Hibridación

Entre los dos algoritmos base fue necesario incluir dos procedimientos que son:

- **Proceso de poda**: El algoritmo propuesto realiza un proceso de poda donde se incluye conocimiento del problema, mediante la eliminación de las oraciones menos relevantes (similitud entre la oración y la colección de los documentos) entregadas por LexRank a GBHS, permitiéndole a GBHS iniciar su proceso evolutivo con las oraciones que más pueden aportar a la solución del problema.

- **Actualización de matrices**: Debido a la disminución de oraciones después del proceso de poda, se hace necesario actualizar las matrices de similitud.

Además, al finalizar el algoritmo fue necesario adaptar el ***Criterio de selección de oraciones en el resumen***, que permite que las oraciones que harán parte del resumen se seleccionan de mayor a menor cobertura.

4.3.3 Algoritmo GBHS

Para el algoritmo GBHS fue necesario adaptar dos procedimientos que son:

- **Inicialización de la Población**: El algoritmo GBHS inicia generando su población inicial seleccionando las armonías con un 70% mediante la técnica GRASP (Ver Figura 3.9) y el 30% de forma aleatoria, de acuerdo al parámetro tamaño de la memoria armónica HMS, donde cada nueva armonía se optimiza con búsqueda local codiciosa; después realiza su proceso evolutivo, terminando de acuerdo al parámetro del máximo número de evaluaciones de la función objetivo. Utilizar la técnica GRASP le permite a GBHS iniciar con una población donde cada armonía es seleccionada incluyendo conocimiento del problema, debido a que la lista restringida de oraciones de candidatas usada por GRASP se basa en la cobertura y diversidad de cada oración.

- **Reparación**: El proceso de reparación de una solución (Ver Figura 4.2) se da cuando la longitud del resumen candidato sobrepasa el máximo número de palabras permitidas en el resumen en su proceso evolutivo. Para llevar a cabo este proceso se incluye nuevamente conocimiento del problema, deshabilitando (colocándola en 0) la oración que tenga menos aporte en el resumen (medido en la similitud de cosenos de la oración frente a la colección de documentos dividido por la longitud de la oración) y luego incluyendo oraciones (colocándolas en 1), ver Figura 4.3, verificando que cumpla con la longitud del máximo número de palabras permitidas en el resumen candidato, seleccionando primero las de mayor valor, es decir, las de mayor similitud a la colección de documentos (mayor cobertura).

```
01  armonia: armonia a reparar
02  LongitudResumen: longitud resumen de la armonía
03  MaxLRE: Máxima Longitud del Resumen Para Evolucionar
04  listaFrases: Lista oraciones ordenadas similitud de cosenos de la oración frente a la colección de
    documentos dividida por la longitud de la oración

05  Procedimieno Reparar(armonia)
```

```
06  Inicio
07      posFinal = Tamaño(armonia) - 1
08      Mientras(LongitudResumen > MaxLRE) Haga
09      Inicio
10          laFrase = listaFrases(posFinal)
11          si (laFrase existe en la armonia) entonces
12              armonia.deshabilitar(laFrase)          //Deshabilitar la oración
13              longitudResumen -= longitud(frase)
14          fin si
15          posFinal--
16          si(posFinal < 0) entonces
17              salir
18          fin si
19      fin Mientras
20      habilitarFrases(armonia)                   //Ir a procedimiento para habilitar oraciones
21  fin
```

Figura 4.2 Algoritmo proceso de reparación de una armonía

```
01  armonia: armonia a habilitar
02  LongitudResumen: longitud resumen de la armonía
03  MaxLRE: Máxima Longitud del Resumen Para Evolucionar
04  frasesOrdenadas: Lista con todas las oraciones ordenadas por cobertura

05  Procedimiento habilitarFrases(armonia)
06  Inicio
07      posInicio=0
08      Mientras(LongitudResumen < MaxLRE) Haga
09      Haga
10          laFrase = frasesordenadas(posInicio)
11          si(longitudResumen + longitud(laFrase) < MaxLRE) entonces
12              si(laFrase no existe en la armonia) entonces
13                  armonia.habilitar(laFrase)          //Habilitar oración
14                  longitudResumen += longitud(laFrase)
15              fin si
16          fin si
17          posInicio++
18          si(posInicio >= tamaño(armonia) entonces
19              salir
20          fin si
21      fin Mientras
22  fin
```

Figura 4.3 Algoritmo habilitar oración en la armonía

4.4 ESQUEMA DE GENERACIÓN DE RESÚMENES

A continuación se describe el proceso de generación de resúmenes de múltiples documentos de acuerdo a la Figura 4.4.

Pre-procesamiento: los documentos de texto de entrada se les realiza la tarea de segmentación del texto para obtener las oraciones que lo conforman; las oraciones se normalizan eliminando mayúsculas y palabras vacías; las palabras se llevan con la misma raíz a una forma común; y finalmente las oraciones se indexan en una estructura de datos (ver la sección 4.5.1, tareas de pre-procesamiento).

Representación oraciones espacio vectorial tridimensional: en este paso los términos de cada una de las oraciones se representan en el modelo de espacio vectorial descrito en

la Sección 2.2.1, donde a cada uno de los términos se le calcula su peso en función de su frecuencia relativa para que se almacene en una matriz de pesos; después con base en los pesos calculados, se obtiene la similitud de cosenos entre las oraciones y la similitud de cosenos de cada oración y los documentos. Dichos valores de similitud servirán posteriormente para el cálculo de los factores de la función objetivo que los requieran, como la cobertura y diversidad.

Ejecución Algoritmo híbrido LexRank-GBHS: la ejecución del algoritmo híbrido *LexRank-GBHS* como fue descrito en la sección 3.2.6, obtiene al final un vector solución, cuyas posiciones con valor igual a uno indican las oraciones candidatas del resumen; las oraciones que harán parte del resumen se seleccionan de acuerdo a su diversidad; posteriormente el vector solución es decodificado para obtener las oraciones originales de los documentos respectivos de entrada, que finalmente harán parte del resumen generado, el cual es truncado a doscientas cincuenta palabras para realizar la evaluación de calidad de los resúmenes con otros métodos del estado del arte.

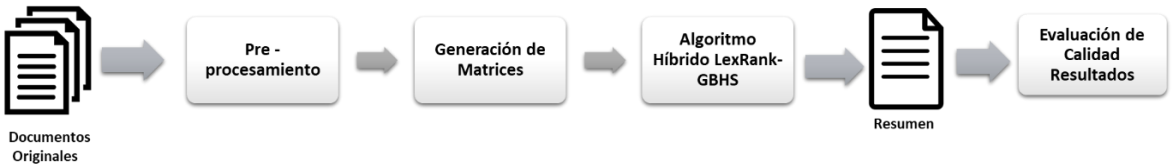

Figura 4.4 Esquema de Generación de Resúmenes

4.5 EVALUACIÓN DE CALIDAD

A continuación se describen las tareas de pre-procesamiento de los documentos, la descripción de la colección de los documentos y las métricas usadas en la experimentación. También se describe el proceso de afinamiento de parámetros del algoritmo propuesto y por último los resultados obtenidos con los resúmenes generados por los esquemas planteados en los cuatro ciclos de desarrollo del algoritmo propuesto *LexRank-GBHS*, y la comparación con los métodos del estado del arte.

4.5.1 Normalización e Indexación de Documentos

La normalización consiste en homogeneizar todo el texto de la colección de documentos sobre la que se trabajará, y que afecta por ejemplo a la consideración de los términos en mayúscula o minúscula; el control de determinados parámetros como cantidades numéricas o fechas; el control de abreviaturas y acrónimos, eliminación de palabras vacías mediante la aplicación de listas de palabras (preposiciones, artículos, etc.), la identificación de N-Gramas, entre otras [63]. En la generación automática de resúmenes de texto la normalización permite la selección de los términos que mejor representan el contenido de los documentos, reduciéndolos a formas canónicas que faciliten los procesos de búsqueda y ordenamiento [64].

4.5.1.1 Segmentación

El proceso de segmentación consiste en descomponer o dividir el texto, en unidades significativas más manejables (normalmente oraciones); es una técnica que recupera y

evalúa los elementos de un texto, tales como ideas, expresiones, etc., permitiendo determinar su valor e importancia. Para realizar las tareas de detección de los límites de las oraciones de un texto se utilizó la herramienta de código abierto "splitta" [8], basados en el buen desempeño reportado en [65].

4.5.1.2 Eliminación de mayúsculas y signos ortográficos

Para permitir un proceso de indexación limpio, previamente es necesario normalizar el texto mediante la conversión de mayúsculas a minúsculas y la eliminación de signos ortográficos, facilitando el emparejamiento de palabras u oraciones [38].

4.5.1.3 Eliminación de palabras vacías

Las palabras vacías, irrelevantes o "stop words" son aquellas que por sí solas carecen de significado y que por su alta frecuencia de aparición en los textos, generan un ruido innecesario para la recuperación de información. La eliminación de estos términos (preposiciones, artículos determinados, artículos indeterminados, pronombres, conjunciones, contracciones y ciertos verbos y adverbios) mejora la afinación en los modelos de recuperación. Los estudios correspondientes a este fenómeno fueron iniciados por Hans Peter Luhn[9] en 1958 con su investigación sobre el índice KWIC una técnica de indexación que organizaba las palabras según su consideración como claves para la recuperación o no de la información, teniendo en cuenta el contexto del documento. Para realizar esta tarea se utilizó la lista de palabras vacías creada para el sistema de recuperación de información denominada *SMART*[10] [66].

4.5.1.4 Lematización

La lematización es un proceso lingüístico que generalmente se refiere a hacer las cosas correctamente con el uso de un vocabulario y un análisis morfológico de las palabras, normalmente con el objetivo de eliminar solo los extremos de inflexión y devolver la forma de base o diccionario de una palabra, que se conoce como el lema [67]. Por razones gramaticales, los documentos van a utilizar diferentes formas de una palabra, debido a que existen familias de palabras derivadas con significados similares, como por ejemplo "democracia", "democrático", y "democratización" [64]. El algoritmo utilizado para realizar las tareas de lematización es el algoritmo de Porter[11] [68], algoritmo de tipo Eliminación de afijos, que consiste en la eliminación de sufijos y prefijos, donde se aplican reglas gramaticales inversas para obtener una forma común de una palabra.

4.5.1.5 Indexación

El proceso de indexación de texto consiste en almacenar las oraciones recolectadas y normalizadas en una estructura de datos, que permita realizar las tareas de recuperación de información de una manera más eficiente. Para llevar a cabo las tareas de indexación,

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

[9] *Hans Peter Luhn, fue un informático alemán. Trabajó para IBM, siendo el primero en emplear la estadística en los análisis textuales en Recuperación de información y fue el creador del indexado KWIC (Key Words In Context)*

[10] Esta lista se encuentra disponible en *ftp://ftp.cs.cornell.edu/pub/smart/english.stop*.

[11] Disponible en http://tartarus.org/martin/PorterStemmer/

conversión de mayúsculas a minúsculas, eliminación de signos ortográficos, se utilizó la librería Lucene[12], que es una librería de código abierto bajo la licencia de Apache Software Licence. Esta librería fue implementada en su primera versión en java, pero en la actualidad ha sido adaptada en otros lenguajes de programación como C#, C++, Delphi, PHP, Phyton y Ruby [69].

4.5.1.6 Colección de Documentos de Evaluación

Para la evaluación de todas las versiones del algoritmo híbrido, se utilizaron los conjuntos de datos de la Conferencia de Entendimiento del Documento (DUC, Document Understanding Conference) para los años 2005 y 2006. La colección DUC2005 está conformada por cincuenta tópicos, cada uno contiene entre 25 y 50 documentos; y DUC2006 comprende cincuenta tópicos, cada uno con 25 documentos. Cada tópico contiene varios resúmenes de referencia (generados por humanos expertos) para que la comunidad académica los pueda comparar con el resumen que se genere automáticamente. El resumen generado debe ser menor a 250 palabras, y se cuenta con varios resúmenes de referencia para cada tópico. Para cada tópico el algoritmo se ejecutó treinta veces (30) para obtener el promedio de cada medida para cada conjunto de datos. La Tabla 34, presenta una breve descripción del conjunto de datos.

Item	DUC2005	DUC2006
Número de tópicos	50	50
Número de documentos	1593	1250
Fuente de datos	TREC[13]	AQUAINT
Longitud del resumen	250	250

Tabla 34 Descripción conjuntos de datos utilizados

4.5.2 Métricas de Evaluación

Para evaluar la calidad de los resúmenes generados por las cinco versiones del algoritmo híbrido propuesto se utilizaron las métricas ROUGE-1, ROUGE-2 y ROUGE-SU4, proporcionadas por la herramienta de evaluación ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [41] en su versión 1.5.5, las cuales son aceptadas por la Conferencia de entendimiento de documentos (DUC, por sus siglas inglés) y la comunidad académica como métricas oficiales para la evaluación de resúmenes automáticos. Esta herramienta permite comparar el contenido de un resumen con uno o más resúmenes de referencia, y obtener el número de n-gramas de palabras que tienen en común.

4.5.3 Afinación de Parámetros

Inicialmente para el proceso de afinamiento de parámetros se creó un conjunto de datos que consta de 15 noticias seleccionados al azar del conjunto de datos DUC2005 y 15 seleccionados al azar del conjunto de datos DUC2006.

Por otra parte, debido a que el algoritmo híbrido propuesto contiene varios parámetros para su ajuste, se utilizó arreglos de cobertura [25] para obtener un grupo más reducido de

[12] Esta librería se encuentra disponible en http://lucenenet.apache.org/
[13] http://trec.nist.gov/overview.html

combinaciones de los parámetros. Con este grupo reducido se inició la experimentación a través de 10 ejecuciones por cada combinación de parámetros; después se seleccionaron las 10 mejores configuraciones para continuar con la experimentación a través de 30 ejecuciones por configuración en todos los conjuntos de datos DUC2005 y DUC2006 para obtener la mejor configuración de parámetros.

Los parámetros del algoritmo híbrido son: el tamaño de la memoria de armónica (Hms), la tasa de consideración de la memoria de armónica (Hmcr), la probabilidad de optimización (Po), el número de optimizaciones (Nop), la longitud máxima del resumen para evolucionar (Mlse), el umbral de LexRank (Umbral), el dampingFactor utilizado en LexRank y la Poda. Así mismo, se seleccionaron las mejores combinaciones para los factores para cada una de las características de la función objetivo (alfa y beta). El número de evaluaciones de la función objetiva se estableció en 15.000, para hacer una comparación con los otros algoritmos metaheurísticos en las mismas condiciones. El error de tolerancia (ToleranceError) del algoritmo Método de Potencia se estableció en 0.01. La longitud máxima del resumen medido en palabras es de 250. En la Tabla 35 se relacionan los parámetros con los valores obtenidos de la mejor configuración.

Parámetro	LexRank-GBHS
Alfa	0.32
Beta	0.68
Threshold	0.04
dampingFactor	0.12
Error de Tolerancia	0.01
Hms	20
Hmcr	0.90
Po	0.90
Nop	10
Poda	0.30
Mlse	275

Tabla 35 Parámetros obtenidos mejor configuración LexRank-GBHS

El algoritmo híbrido LexRank-GBHS fue ejecutado treinta veces por noticia, evaluando los resúmenes generados en cada ejecución a través de las métricas de ROUGE, obteniendo al final un resultado promedio de todo el conjunto de documentos por cada métrica. La implementación del algoritmo se realizó en el lenguaje de programación C# de la plataforma .NET, las ejecuciones del algoritmo se realizaron en un computador de escritorio con procesador Intel Core ii7 2.6 GHz, RAM de 8 GB, sistema operativo Windows 10.

4.5.4 Comparación con Métodos del Estado del Arte

En la Tabla 36, se presentan los resultados obtenidos en la evaluación de calidad de los resúmenes generados por las cinco (5) versiones del algoritmo híbrido propuestos en esta investigación, junto a otros métodos del estado del arte sobre los conjuntos de datos DUC2005 y DUC2006, usando las medidas ROUGE-1, ROUGE-2 y ROUGE-SU4 [41]. Los resultados de la mejor solución se resaltan en negrilla y la columna después de cada medición ROUGE indica la clasificación de cada método.

Las versiones de los algoritmos híbridos propuestos se compararon con los métodos del estado del arte: DESAMC+DocSum [21], MA-Multisumm [20], SVR [15], GHS [70], TMR [71], LexRank [2], LEX [72], Hiersum [73], SNMF +SLSS [30], Centroide [18].

Como se observa en la Tabla 36, de las cinco versiones propuestas del Algoritmo Híbrido, la versión denominada LexRank-GBHS es la que obtuvo los mejores resultados. En el conjunto de datos DUC2005 solo es superado por los métodos MA-Multisum y DESAMC+DocSum en todas las medidas ROUGE. En el conjunto de datos DUC2006, ocupa la tercera posición en la medida ROUGE-1, la sexta posición en ROUGE-2 y la quinta posición en ROUGE-SU4.

Método	**DUC2005**						**DUC2006**					
	ROUGE-1		**ROUGE-2**		**ROUGE-SU4**		**ROUGE-1**		**ROUGE-2**		**ROUGE-SU4**	
GBHS	0,3844	7	0,0774	5	0,1352	6	0,4023	6	0,0880	7	0,1438	7
LexRank	0,3868	4	0,0767	7	0,1354	**4**	0,4006	9	0,0848	14	0,1419	10
DESAMC+DocSum	0,3937	2	0,0822	2	0,1418	2	**0,4345**	**1**	**0,0989**	**1**	**0,1569**	**1**
SVR	0,3849	6	0,0757	9	0,1335	10	0,4018	7	0,0926	3	0,1485	4
TMR	0,3775	11	0,0715	13	0,1304	13	0,4063	4	0,0913	4	0,1504	3
LEX	0,376	12	0,0735	12	0,1316	12	0,403	5	0,0913	5	0,1449	6
HierSum	0,3753	13	0,0745	11	0,1324	11	0,401	8	0,0860	9	0,143	8
SNMF+SLSS	0,3501	15	0,0604	15	0,1172	15	0,3955	11	0,0855	11	0,1429	9
Centroid	0,3535	14	0,0638	14	0,1198	14	0,3807	15	0,0785	15	0,133	15
MA-MultiSumm	**0,4001**	**1**	**0,0868**	**1**	**0,1434**	**1**	0,4195	2	0,0986	2	0,1526	2
LexRank-GBHS	0,3906	3	0,0795	3	0,1374	3	0,4076	3	0,0898	6	0,1452	5
GBHS-LexRank	0,38	10	0,0755	10	0,1337	9	0,3945	12	0,0855	12	0,1337	14
LexRank-GBHS-2Resumen	0,383	8	0,0776	4	0,1351	7	0,391	14	0,0856	10	0,1351	12
GBHS-LexRank-2Resumen	0,385	5	0,0769	6	0,1354	5	0,3993	10	0,0879	8	0,1354	11
LexRank-GBHS-Paralelo	0,3822	9	0,0767	8	0,1342	8	0,3923	13	0,0852	13	0,1342	13

Tabla 36 Valores Rouge de los métodos en DUC2005 y DUC2006

Para identificar que método obtiene los mejores resultados en ambos conjuntos de datos, se aplica la misma clasificación unificada utilizada en el capítulo 3, teniendo en cuenta la posición de cada uno de los métodos para cada medida. En la Tabla 37, la columna *Rank* nos indica la posición del método en la clasificación, valor que es calculado de acuerdo a la Ecuación (3.17), mencionada anteriormente. Los mejores resultados de las cinco versiones propuestas las obtuvo la versión denominada *LexRank-GBHS*, quien superó los resultados de los algoritmos base de forma individual (LexRank y GBHS) y es superado solamente por los métodos MA-MultiSumm [20] y DESAMC+DocSum [21]; la versión denominada GBHS-LexRank2Resumen ocupa la 6 posición, LexRank-GBHS-2Resumen ocupa la 10 posición, LexRank-GBHS-Paralelo y GBHS-LexRank ocupan la posición 12ª y 13ª posición respectivamente.

Método	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Rank
DESAMC+DocSum	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	5,8
MA-MultiSumm	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	5,8
LexRank-GBHS	0	0	4	0	1	1	0	0	0	0	0	0	0	0	0	4,9
GBHS	0	0	0	0	1	2	3	0	0	0	0	0	0	0	0	3,9
SVR	0	0	1	1	0	1	1	0	1	1	0	0	0	0	0	3,8
GBHS-LexRank-2Resumen	0	0	0	0	2	1	0	1	0	1	1	0	0	0	0	3,4
LexRank	0	0	0	2	0	0	1	0	1	1	0	0	0	1	0	3,2
TMR	0	0	1	2	0	0	0	0	0	0	1	0	2	0	0	3,2
LEX	0	0	0	0	2	1	0	0	0	0	0	3	0	0	0	2,9
LexRank-GBHS-2Resumen	0	0	0	1	0	0	1	1	0	1	0	1	0	1	0	2,7
HierSum	0	0	0	0	0	0	0	2	1	0	2	0	1	0	0	2,4
LexRank-GBHS-Paralelo	0	0	0	0	0	0	0	2	1	0	0	0	3	0	0	2,1
GBHS-LexRank	0	0	0	0	0	0	0	0	1	2	0	2	0	1	0	1,9
SNMF+SLSS	0	0	0	0	0	0	0	0	1	0	2	0	0	0	3	1,3
Centroid	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0,6

Tabla 37 Posición resultante de los métodos

Teniendo en cuenta los resultados de la Tabla 37 , se puede observar lo siguiente:

- *LexRank-GBHS* solo es superado por los métodos DESAMC+DocSum y MA-Multisum (primeros puestos), métodos evolutivos donde el primero realiza 50.000 evaluaciones de la función objetivo y el segundo las mismas 15.000 del algoritmo LexRank-GBHS. Sin embargo, LexRank-GBHS no requiere del esfuerzo para la configuración de operadores como: selección, cruce, mutación, reemplazo, búsqueda local; como ocurre con MA-Multisum.
- *LexRank-GBHS* ocupa una mejor posición que métodos basados en: metaheurísticas, GBHS (cuarta posición); grafos, LexRank (octava Posición); reducción algebraica, SVR (quinta posición), SNMF+SLSS (catorceava posición) que utiliza factorización matricial no negativa simétrica para agrupar las oraciones; modelos probabilísticos para estimar la distribución de los tópicos, TMR (octava posición); agrupamiento de términos, LEX (novena posición); agrupamiento de centroides, Centroid (quinceava posición).
- Todas las versiones de los algoritmos híbridos ocuparon una mejor posición que Centroid y SNMF+SLSS.

Capítulo 5

5 APLICACIÓN WEB

En este capítulo se presenta el diseño de la aplicación web que usa el algoritmo híbrido propuesto LexRank-GBHS, se incluye la arquitectura, diagramas de clases y las interfaces; también se presenta la evaluación de satisfacción del usuario de la aplicación que contiene el diseño de la encuesta, su aplicación y el análisis de resultados.

5.1 DISEÑO DE LA APLICACIÓN WEB

La aplicación web construida utiliza el Algoritmo híbrido propuesto para la generación automática de resúmenes extractivos de múltiples documentos (LexRank-GBHS). En las siguientes secciones se presenta la arquitectura de la aplicación, diagrama de clases y la interfaz de usuario.

5.1.1 Arquitectura de la Aplicación

Esta aplicación usa la arquitectura de ASP .NET de Microsoft (Ver Figura 5.1), y fue desarrollada en lenguaje C# teniendo como soporte el Framework de Visual Studio versión 4.5. Se utilizó la librería Jquery como apoyo para la programación Front-End.

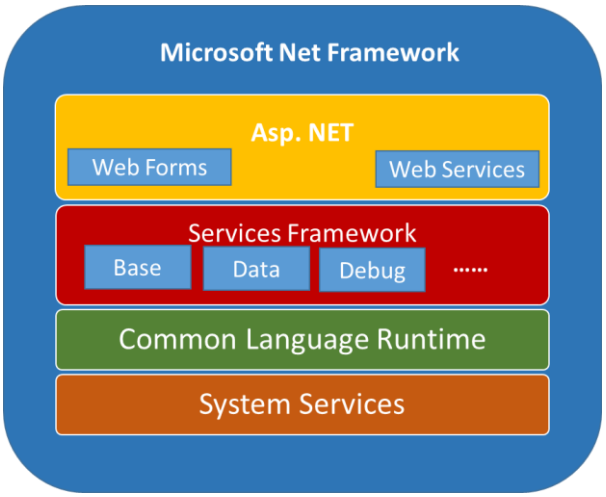

Figura 5.1 Arquitectura de la Aplicación Web

La arquitectura utilizada es de N-capas permitiendo una separación de responsabilidades para cada una de ellas. En este proyecto se definieron dos capas, estas son:

- **Presentación**: Encargada de interactuar con el usuario mediante formularios para obtener el texto de los diferentes documentos fuente para generar el resumen, de la misma forma permite al usuario poder configurar cada uno de los parámetros del algoritmo híbrido LexRank-GBHS.

- **Lógica del Negocio**: Es la encargada de utilizar los documentos fuente cargados por el usuario para realizar el proceso de la generación de resúmenes de texto para múltiples documentos mediante el algoritmo híbrido LexRank-GBHS.

5.1.2 Diagrama de Paquetes

En la Figura 5.2, se observa el diagrama de paquetes que permite visualizar de manera organizada la distribución de responsabilidades en la aplicación, lo que permite maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo. Se incluye un paquete relacionado con las reglas del negocio, que contiene las clases que permiten realizar las tareas de preprocesamiento y la generación de la matriz de similitudes de las oraciones (Modelo Vectorial); las clases que permiten generar el resumen mediante el algoritmo híbrido propuesto LexRank-GBHS, LexRank y GBHS (SummarizationExtractiva); y las clases para la evaluación de calidad (Utilidades). También se incluye un paquete con las clases de la capa de presentación que representan cada una de las interfaces de usuario (Presentación).

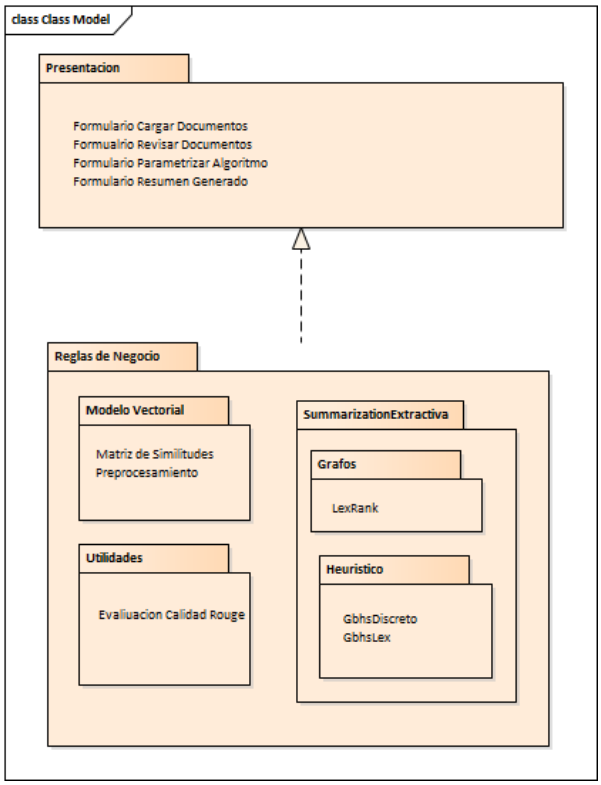

Figura 5.2 Diagrama de Paquetes

5.1.3 Interfaces de Usuario

La primera interface de la aplicación presenta inicialmente al usuario una introducción sobre el tema de Investigación: “Generación automática de resúmenes de texto de múltiples documentos” Ver Figura 5.3. Para cargar los documentos fuentes necesarios para generar el resumen, el usuario debe dar clic en la opción *Documentos* del menú, y tendrá acceso a la interface de la Figura 5.4.

Figura 5.3 Interfaz inicial de la Aplicación

Figura 5.4 Interfaz Agregar Documentos Fuente

En esta interface (Figura 5.4), el usuario por medio del botón *Cargar*, selecciona los documentos fuente necesarios para generar el resumen (uno o más documentos al mismo tiempo), teniendo en cuenta que los archivos deben estar en formato txt. Los documentos cargados aparecen relacionados en la parte inferior de la interface, además, puede *Eliminar* los archivos cargados en la misma interface. Si el usuario quiere revisar el contenido de cada uno de ellos lo puede hacer ingresando por la opción *Revisar* del menú Documentos (Ver Figura 5.5).

En la opción *Parametrizar* del menú *Algoritmo*, la aplicación presenta una interface de acuerdo a la Figura 5.6, con los parámetros de los algoritmos LexRank, GBHS y los parámetros asociados al problema. Inicialmente el valor de cada uno de los parámetros se encuentra establecido, de acuerdo a los valores obtenidos de la mejor versión del algoritmo híbrido LexRank-GBHS. También el usuario puede cambiar el valor de los parámetros si así lo requiere y finalmente por medio del botón *Generar Resumen*, obtener el resumen automático generado por el algoritmo híbrido. Este resumen se presenta como lo muestra la interface de acuerdo a la Figura 5.7.

En la misma interface donde se muestra el resumen generado, se presenta un enlace para que el usuario realice la evaluación de satisfacción de la aplicación (Para más detalle Ver sección 5.2).

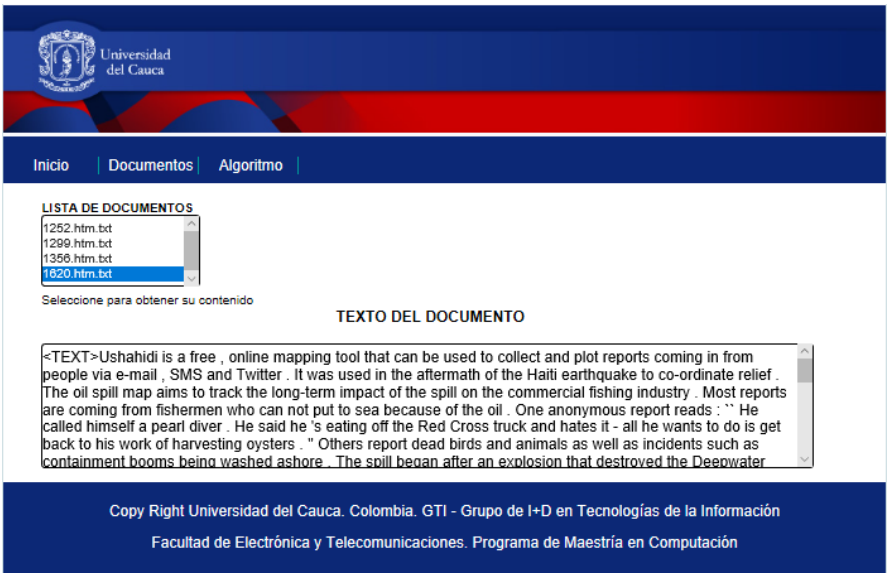

Figura 5.5 Interfaz Lista y Revisión Documentos Fuente

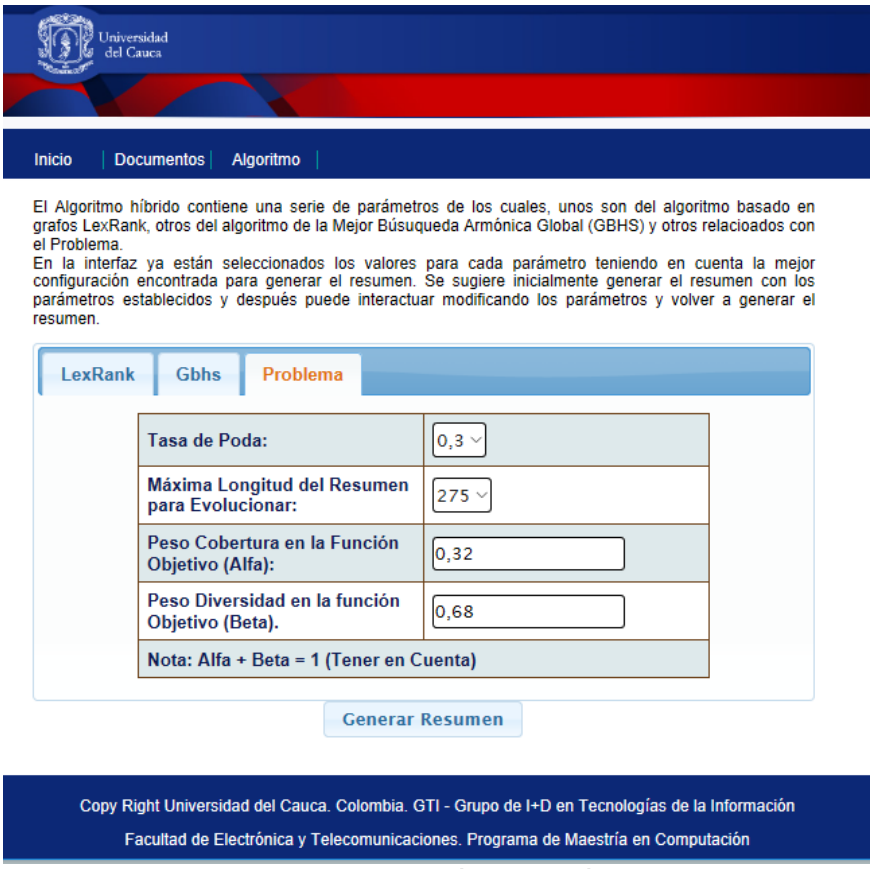

Figura 5.6 Interfaz Configuración de Parámetros Algoritmo

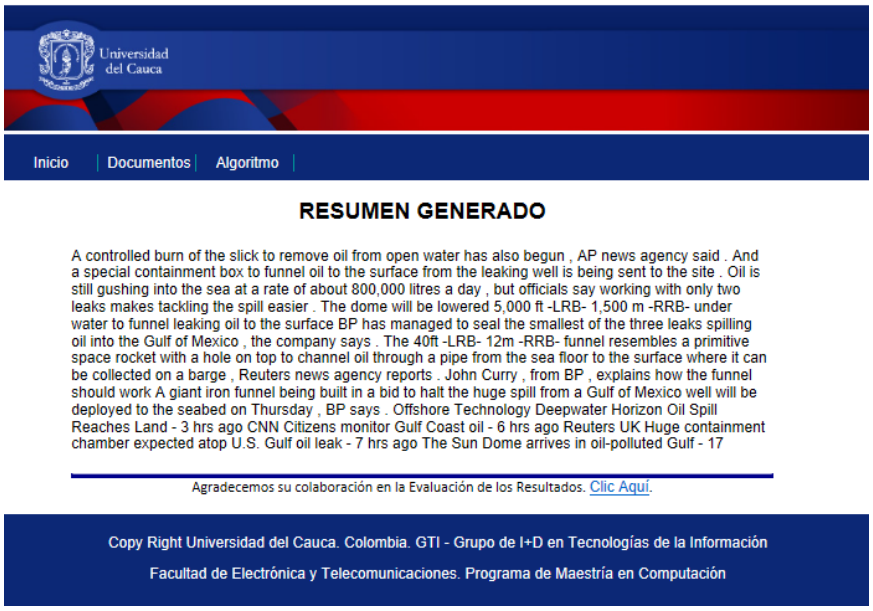

Figura 5.7 Interfaz Resumen Generado

5.2 EVALUACIÓN DE SATISFACCIÓN DEL USUARIO

Para realizar la evaluación de satisfacción del usuario con respecto a los resúmenes que genera el algoritmo propuesto por medio de la aplicación Web, se usaron modelos y métricas de acuerdo a la norma internacional ISO/IEC 25000 [74]. La ISO/IEC 25000 conocida como SQuaRE (Requisitos y evaluación de calidad de productos Software), tiene como objetivo proporcionar un marco de trabajo común para evaluar la calidad de un producto Software. De acuerdo a la Figura 5.8, esta se compone de cinco divisiones y cada una de ellas compuesta por diferentes estándares.

Figura 5.8 División de la norma ISO 25000. Fuente [74]

En este proyecto se abordaron las Divisiones ISO/IEC 2501n y 2502n, donde la primera contiene el estándar ISO/IEC 25010, Modelo de Calidad Genérico, compuesto por dos modelos, uno relacionado con la calidad del producto software y uno para la calidad en uso. Se seleccionó el modelo de calidad del producto Software, el cual define 8 características de calidad, de las cuáles se escogieron tres características de calidad como (Adecuación Funcional, Eficiencia de Desempeño y Usabilidad) para evaluar la satisfacción del usuario. La segunda división contiene el estándar ISO/IEC 25022 Medidas de Calidad en Uso, este provee un conjunto de métricas que permiten evaluar el cumplimiento de las características de calidad en uso de la ISO/IEC 25010 en un producto o sistema software, en este caso, se seleccionó la métrica nivel de satisfacción. En la Tabla 38, se relacionan las métricas de calidad del producto software.

METRICAS PARA LA CALIDAD DEL PRODUCTO SOFTWARE		
Característica	**Subcaracterística**	**Métrica**
Adecuación Funcional	Completitud funcional	Completitud de la implementación funcional
Eficiencia y Desempeño	Comportamiento temporal	Tiempo de Respuesta

Usabilidad	Capacidad de ser entendido	Efectividad de la documentación del usuario o ayuda del sistema
	Protección contra errores del usuario	Verificación de entradas válidas

Tabla 38 Métricas para evaluar la calidad norma ISO/IEC 25023

A continuación se describe las características utilizadas en las métricas de calidad, de acuerdo a la Tabla 38:

- Adecuación funcional: Representa la capacidad del producto software para proporcionar las funciones necesarias para satisfacer el usuario.

- Eficiencia y Desempeño: Capacidad de un producto software en proporcionar un rendimiento apropiado, respecto a la cantidad de recursos utilizados bajo determinadas condiciones.

- Usabilidad: Capacidad de un producto software para ser entendido, aprendido, recordado, usado y atractivo para el usuario, cuando se usa bajo determinadas condiciones.

5.2.1 Diseño de la Encuesta

En la elaboración de la encuesta, se definieron seis preguntas que se clasificaron de acuerdo a las características de calidad seleccionadas en la sección anterior, como lo muestra la Tabla 39. Para la valoración de las preguntas se utilizó la escala de Likert [75] (Ver Tabla 40), que le permite al usuario hacer una valoración desde lo más positivo con *Totalmente de Acuerdo* hasta lo más negativo con *Totalmente en Desacuerdo*. Adicionalmente; además se dejó un espacio para que el usuario hiciera comentarios.

Características de Calidad	No. Pregunta	Pregunta
Adecuación Funcional	1	¿Considera usted que el resumen generado cubre los aspectos principales del conjunto de documentos utilizados para generar el resumen?
	2	¿Considera usted que el resumen generado contiene oraciones que no repiten la misma información?
Eficiencia y Desempeño	6	¿El tiempo de respuesta a la hora de generar el resumen es corto?
Usabilidad	5	¿La Herramienta hace una debida validación de los datos de ingreso en los formularios?
	3	¿La Herramienta permite al usuario seleccionar los documentos fuentes para generar el resumen?
	4	¿La Herramienta permite al usuario poder configurar los parámetros del algoritmo para generar el resumen?

Tabla 39 Clasificación preguntas de acuerdo a características de calidad

Elemento
Totalmente de Acuerdo
De acuerdo
Indeciso
En Desacuerdo
Totalmente en Desacuerdo

Tabla 40 Escala de Likert

La Ecuación (5.1), permite valorar cada pregunta X_i del cuestionario relacionada en la Tabla 39, teniendo en cuenta cada elemento de la escala de Likert de la Tabla 40.

$$X_i = \frac{A_j}{B} \tag{5.1}$$

Donde A, hace referencia a la cantidad de respuestas del elemento j de la escala de Likert, y B, corresponde al número total de encuestados.

Ejemplo:

$$X_1 = \frac{10\ Totalmente\ de\ Acuerdo}{12} = 0.83$$

$$X_1 = \frac{2\ De\ Acuerdo}{12} = 0.17$$

$$X_1 = \frac{0\ Indeciso}{12}$$

$$X_1 = \frac{0\ En\ Desacuerdo}{12}$$

$$X_5 = \frac{0\ Totalmente\ en\ Desacuerdo}{12}$$

De acuerdo al ejemplo anterior podemos concluir que el 83% está Totalmente de acuerdo con la pregunta 1, el 17% está de acuerdo.

5.2.2 Aplicación de la Encuesta

La encuesta se aplicó a doce (12) personas profesionales de diferentes áreas como la ingeniería y administración, de los cuales siete (7) encuestados fueron de la ciudad de Neiva y los otros cinco (5) de la ciudad de Popayán.

5.2.3 Análisis de Resultados

En la Tabla 41, se presentan los resultados de la evaluación de la aplicación web para cada una de las seis (6) preguntas, de acuerdo a la escala de Likert y al número de encuestados de 12; también se relaciona de manera general los comentarios de los encuestados. Teniendo en cuenta estos resultados, se aprecia lo siguiente con respecto a las características medidas:

- *Adecuación funcional*: las preguntas 1 y 2, validan la cobertura y diversidad de las oraciones del resumen generado, el 75% de los encuestados contestaron Totalmente de Acuerdo y el 25% estuvo de Acuerdo; mostrando que el 100% está de acuerdo. En este resultado se aprecia que los encuestados expresan que el resumen generado está

cubriendo los temas más importantes de la colección de documentos y que las oraciones que componen el resumen no son similares o repetidas.

- *Eficiencia y desempeño*: la pregunta 6, valida el tiempo de respuesta en generar el resumen, el 75% de los encuestados contestaron Totalmente de Acuerdo y el 25% estuvo de Acuerdo, mostrando que el 100% está de acuerdo.

- *Usabilidad*: las preguntas 3 y 4, validan la forma de seleccionar los documentos fuente y la posibilidad de que el usuario pueda configurar algunos parámetros del algoritmo, el 100% de los encuestados contestaron Totalmente de Acuerdo.

  La pregunta 5, valida la entrada de datos necesarios para generar el resumen, en este caso los documentos fuentes, el 92% de los encuestados contestaron Totalmente de Acuerdo y el 8% de Acuerdo.

Elemento Escala Likert	**Preguntas**					
	1	**2**	**3**	**4**	**5**	**6**
Totalmente de Acuerdo	75%	75%	100%	100%	92%	75%
De acuerdo	25%	25%	0	0	8%	25%
Indeciso	0	0	0	0	0	0
En Desacuerdo	0	0	0	0	0	0
Totalmente en Desacuerdo	0	0	0	0	0	0

Tabla 41 Resultados evaluación según escala de Likert

En cuanto a los comentarios de los encuestados se encontraron aspectos buenos como:

- La mayoría coincide en que la herramienta es sencilla de usar.
- La herramienta realiza una buena validación de los datos, incluyendo la captura de los documentos fuente.
- El resumen generado cumple con el tamaño máximo de longitud de 250 palabras.
- El resumen es coherente de acuerdo al contenido de los documentos fuente.

Y aspectos por mejorar como:

- Definir con claridad como poder medir mejor el tiempo de respuesta a la hora de generar el resumen.

Por último, se muestra por medio de un diagrama de barras el número de respuestas obtenidas por los encuestados de acuerdo a la escala de Likert, como se observa en la Figura 5.9.

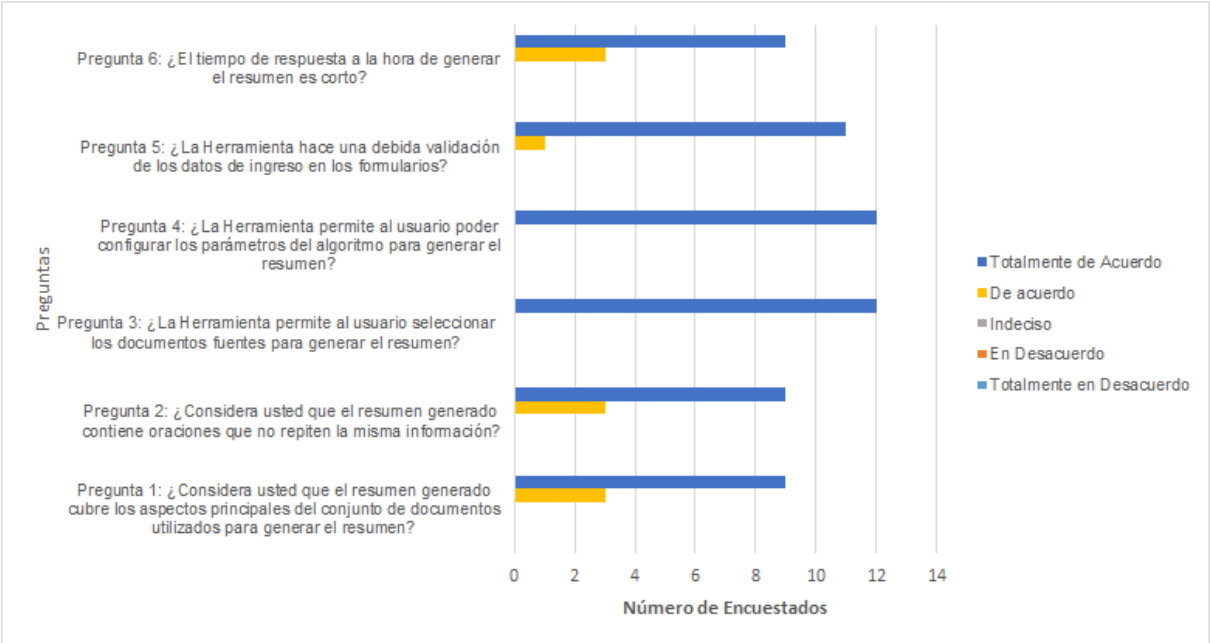

Figura 5.9 Diagrama de barras resultados evaluación satisfacción

Capítulo 6

6 CONCLUSIONES Y TRABAJO FUTURO

6.1 CONCLUSIONES

En este proyecto se presentó el diseño de varias versiones de un algoritmo híbrido para la generación de resúmenes extractivos de múltiples documentos entre la metaheurística de la Mejor Búsqueda Armónica Global (GBHS) y el algoritmo basado en grados LexRank. Se realizaron diferentes modificaciones a los algoritmos base, al Método de potencia de LexRank, para obtener las oraciones ordenadas de mayor a menor relevancia al generar un resumen. Un proceso de poda aplicado a las oraciones que entrega LexRank, para eliminar relaciones menos relevantes y actualizar la matriz de similitudes debido a la disminución de oraciones. En GBHS se modificó la generación de la población inicial así: (1) Uso de la técnica GRASP para generar el 100% de la población, intercambiando una y dos oraciones en la optimización de GRASP; (2) Combinación de la técnica GRASP y aleatoria, con un 30% generada aleatoriamente y el 70% mediante GRASP, intercambiando una y dos oraciones en la optimización de GRASP; (3) Combinación de la técnica aleatoria y dispersa, donde un 50% se obtiene de forma aleatoria y el otro 50% se selecciona de acuerdo a las más diversas comparadas con las seleccionadas aleatoriamente, intercambiando una oración en la optimización. Además en la búsqueda local codiciosa de GBHS se modificó, la generación de un nuevo vecino con el intercambio de una y dos oraciones; y el criterio de selección de las oraciones que conforman el resumen utilizando el criterio de mayor a menor cobertura.

El algoritmo híbrido propuesto que obtuvo los mejores resultados en la evaluación de calidad fue el algoritmo LexRank-GBHS, el cual, inicia con la ejecución de LexRank, al que se le realiza una modificación en el algoritmo iterativo Método de Potencia, para que el vector inicial con los pesos de las oraciones se base en la cobertura dividida por la suma de todas las similitudes de las oraciones; al finalizar la ejecución de LexRank, se entrega un vector con las oraciones ordenadas de mayor a menor relevancia. Después LexRank-GBHS, realiza un proceso de poda que permite eliminar las oraciones menos relevantes del vector entregado por LexRank y actualiza la matriz de similitudes entre oraciones. Luego, el algoritmo híbrido continua con la ejecución de GBHS, cuya generación de la población inicial es modificada, seleccionando un 70% de armonías con la Técnica GRASP y el 30% de forma aleatoria; al finalizar el proceso evolutivo, GBHS genera el resumen con las oraciones de la mejor armonía de la memoria armónica, teniendo en cuenta la restricción del número máximo de palabras en el resumen. Las oraciones que harán parte del resumen se seleccionan de mayor a menor cobertura.

La evaluación de calidad de los resúmenes generados por las diferentes versiones del algoritmo híbrido, se realizó mediante las medidas ROUGE-1, ROUGE-2 y ROUGE-SU4 para cada uno de los conjuntos de datos DUC2005 y DUC2006, comparándolos con otros métodos del estado del arte. Al realizar la comparación, se observó que el algoritmo LexRank-GBHS ocupa la tercera posición en las tres medidas de ROUGE para el conjunto de datos DUC2005 y en la medida ROUGE 1 para DUC2006; en las medidas ROUGE-2 ocupa la sexta posición y ROUGE-SU4 la quinta posición. *LexRank-GBHS* es superado por

los métodos DESAMC+DocSum y MA-Multisum, métodos evolutivos donde el primero realiza 50.000 evaluaciones de la función objetivo y el segundo las mismas 15.000 del algoritmo LexRank-GBHS. Sin embargo, LexRank-GBHS no requiere del esfuerzo para la configuración de operadores como: selección, cruce, mutación, reemplazo, como ocurre con MA-Multisum.

De acuerdo a la pregunta de Investigación, el algoritmo propuesto *LexRank-GBHS*, no obtuvo resultados de mayor calidad a los reportados en el estado del arte, pero sí muy similares, logrando ubicarse en un tercer puesto, lo que le permitió obtener resultados de mayor calidad a diferentes métodos del estado del arte.

Para la definición de la función objetivo en las diferentes versiones del algoritmo híbrido, se tomó como base las características de Cobertura y Redundancia propuestas en [20] y se revisaron otras del estado del arte como: posición (tres formas de cálculo) y longitud de la oración (dos formas de cálculo). Después se definió una función objetivo conformada por las características de Cobertura y Diversidad, que permite que LexRank-GBHS obtenga resultados comparables a los reportados por otros métodos del estado del arte. Al finalizar el proceso de afinación de la función objetivo, se encontró que la característica con mayor peso fue la de la diversidad, un factor importante cuando se trata el problema de generación de resúmenes de múltiples documentos que abordan un mismo tema, evitando de esta forma que el resumen generado contenga información repetida. También, se muestra la importancia de la Cobertura que permite seleccionar las oraciones que más se parecen al contenido central de toda la colección de documentos, sobre característica como la posición y longitud.

Además, se desarrolló una aplicación web para la generación de resúmenes extractivos de múltiples documentos mediante el algoritmo híbrido LexRank-GBHS. Esta aplicación fue evaluada para medir la satisfacción del usuario, teniendo en cuenta características de calidad como: adecuación funcional, eficiencia de desempeño y usabilidad; obteniendo muy buenos resultados, donde las dos preguntas relacionadas con la característica de adecuación funcional el 75% de los encuestados estuvo Totalmente de Acuerdo y el otro 25% de Acuerdo, la pregunta relacionada con la característica de eficiencia de desempeño el 75% contestaron Totalmente de Acuerdo y el 25% de Acuerdo, dos preguntas relacionadas con la característica de Usabilidad los encuestados contestaron 100% Totalmente de Acuerdo y en una pregunta el 92% Totalmente de Acuerdo y el 8% de Acuerdo.

6.2 RECOMENDACIONES Y TRABAJO FUTURO

Modificar el algoritmo híbrido entre LexRank y GBHS, cambiando LexRank por otro algoritmo basado en grafos, con el objetivo de verificar su comportamiento y desempeño en la generación de resúmenes de múltiples documentos.

Con respecto al algoritmo evolutivo GBHS, se espera explorar otros métodos de optimización local, como la búsqueda local guiada, o realizando variaciones en la búsqueda local codiciosa donde la oración que se adicione se haga de forma aleatoria, buscando que el algoritmo híbrido pueda obtener mejores resultados que los métodos del estado del arte.

Evaluar la calidad del algoritmo propuesto *LexRank-GHS* con otros tipos de documentos (diferentes a noticias), para revisar si reporta buenos resultados y si es necesario hacer ajustes a su configuración de parámetros y de la función objetivo.

BIBLIOGRAFÍA

[1] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation,* vol. 198, pp. 643-656, 2008.

[2] G. s. Erkan and D. R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *Artificial Intelligence Research,* vol. 22, pp. 457-479, 2004.

[3] K. Wu, L. Li, J. Li, and T. Li, "Ontology-enriched multi-document summarization in disaster management using submodular function," *Information Sciences,* vol. 224, pp. 118-129, 2013.

[4] N. Kumaresh and B. Ramakrishnan, "Graph Based Single Document Summarization," in *Data Engineering and Management*. vol. 6411, R. Kannan and F. Andres, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 32-35.

[5] A. Porselvi and S. Gunasundari, "Survey on web page visual summarization," *International Journal of Emerging Technology and Advanced Engineering,* vol. 3, pp. 26-32, 2013.

[6] A. Alhindi, U. Kruschwitz, C. Fox, and M.-D. Albakour, "Profile-Based Summarisation for Web Site Navigation," *ACM Trans. Inf. Syst.,* vol. 33, pp. 1-39, 2015.

[7] D. M. Zajic, B. J. Dorr, and J. Lin, "Single-document and multi-document summarization techniques for email threads using sentence compression," *Information Processing & Management,* vol. 44, pp. 1600-1610, 2008.

[8] D. M. Zajic, B. J. Dorr, and J. Lin, "Single-document and multi-document summarization techniques for email threads using sentence compression," *Information Processing and Management,* vol. 44, pp. 1600-1610, 2008.

[9] C. Cobos, H. Muñoz-Collazos, R. Urbano-Muñoz, M. Mendoza, E. León, and E. Herrera-Viedma, "Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion," *Information Sciences,* vol. 281, pp. 248-264, 2014.

[10] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review,* vol. 47, pp. 1-66, 2016.

[11] J. Atkinson and R. Munoz, "Rhetorics-based multi-document summarization," *Expert Systems with Applications,* vol. 40, pp. 4346-4352, 2013.

[12] S. Park, B. Cha, and D. An, "Automatic Multi-document Summarization Based on Clustering and Nonnegative Matrix Factorization," *IETE Technical Review,* vol. 27, p. 167, 2010.

[13] S. Xiong and Y. Luo, "A New Approach for Multi-document Summarization Based on Latent Semantic Analysis," *14 Seventh International Symposium on Computational Intelligence and Design,* pp. 177-180, 2014.

[14] M. A. Fattah, "A hybrid machine learning model for multi-document summarization," *Applied Intelligence,* vol. 40, pp. 592-600, 2013.

[15] Y. Ouyang, W. Li, S. Li, and Q. Lu, "Applying regression models to query-focused multi-document summarization," *Information Processing & Management,* vol. 47, pp. 227-237, 2011.

[16] J. Zhang, X. Cheng, and H. Xu, "GSPSummary: A Graph-Based Sub-topic Partition Algorithm for Summarization," *Springer-Verlag Berlin Heidelberg,* pp. 321-334, 2008.

[17] R. Ferreira, L. de Souza Cabral, F. Freitas, R. D. Lins, G. de França Silva, S. J. Simske*, et al.*, "A multi-document summarization system based on statistics and linguistic treatment," *Expert Systems with Applications,* vol. 41, pp. 5780-5787, 2014.

[18] D. R. Radev, H. Jing, and M. Budzikowska, "Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies," *Information Processing & Management,* pp. 919-938, 2004.

[19] D. Liu, Y. Wang, C. Liu, and ZhiqiWang, "Multiple Documents Summarization Based on Genetic Algorithm," *Springer-Verlag Berlin Heidelberg,* pp. 355-364, 2006.

[20] M. Mendoza, C. Cobos, E. León, M. Lozano, F. Rodríguez, and E. Herrera-Viedma, "A New Memetic Algorithm for Multi-document Summarization Based on CHC Algorithm and Greedy Search," in *Human-Inspired Computing and Its Applications*. vol. 8856, A. Gelbukh, F. Espinoza, and S. Galicia-Haro, Eds., ed: Springer International Publishing, 2014, pp. 125-138.

[21] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "DESAMC+DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization," *Knowledge-Based Systems,* vol. 36, pp. 21-38, 2012.

[22] A. Celikyilmaz and D. Hakkani-Tur, "A Hybrid Hierarchical Model for Multi-Document Summarization," *48th Annual Meeting of the Association for Computational Linguistics,* pp. 815-824, 2010.

[23] W. Meng and T. Xinlai, "Extract Summarization Using Concept-Obtained and Hybrid Parallel Genetic Algorithm," *8th International Conference on Natural Computation,* pp. 662-664, 2012.

[24] C. Wang, L. Long, and L. Li, "HowNet Based Evaluation for Chinese Text Summarization," *2008 International Conference on Natural Language Processing and Knowledge Engineering,* pp. 1-6, 2008.

[25] O. Abdel-Raouf and M. A.-B. Metwally, "A Survey of Harmony Search Algorithm," *A Survey of Harmony Search Algorithm,* vol. 70, pp. 17-26, 2013.

[26] R. Mihalcea, P. Tarau, and E. Figa, "PageRank on Semantic Networks, with Application toWord Sense Disambiguation," *Proceedings of the 20th international conference on Computational Linguistics,* p. 1126, 2004.

[27] E. Lloret and M. Palomar, "Text summarisation in progress: a literature review," *Artificial Intelligence Review,* vol. 37, pp. 1-41, 2012.

[28] M. Mendoza and L. Elizabeth, "Una Revisión de la Generación Automática de Resúmenes Extractivos," *Revista UIS Ingenierías.,* vol. 12, pp. 7-27, 2013.

[29] A. R. Pal and D. Saha, "An Approach to Automatic Text Summarization using WordNet," *2014 IEEE International Advance Computing Conference (IACC),* pp. 1169-1173, 2014.

[30] T. L. Dingding Wang, S. Zhu, and C. Ding, "Multi-Document Summarization via Sentence-Level Semantic Analysis and Symmetric Matrix Factorization," *31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 307-314, Julio 20-24 2008.

[31] S. Brin and L. Page, "The PageRank Citation Ranking Bringing Order to the Web," *Universidad de Stanford,* pp. 1- 17, January 29 1998.

[32] G. Erkan and D. R. Radev, "LexRank: graph-based lexical centrality as salience in text summarization," *J. Artif. Int. Res.,* vol. 22, pp. 457-479, 2004.

[33] A. John and M. Wilscy, "Vertex Cover Algorithm Based Multi-document Summarization Using Information Content of Sentences," *Procedia Computer Science,* vol. 46, pp. 285-291, 2015.

[34] J. V. Tohalino and D. R. Amancio, "Extractive multi-document summarization using multilayer networks," *Physica A: Statistical Mechanics and its Applications,* vol. 503, pp. 526-539, 2018.

[35] R. M. Alguliev, R. M. Aliguliyev, and M. S. Hajirahimova, "GenDocSum+MCLR: Generic document summarization based on maximum coverage and less redundancy," *Expert Systems with Applications,* vol. 39, pp. 12460-12473, 2012.

[36] A. John, P. S. Premjith, and M. Wilscy, "Extractive multi-document summarization using population-based multicriteria optimization," *Expert Systems with Applications,* vol. 86, pp. 385-397, 2017.

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

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

[39] M. Hassel, "Resource Lean and Portable Automatic Text Summarization," Doctoral, Computer Science and Communication, KTH School of Computer Science and Communication, Stockholm, Sweden 2007.

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

[41] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop,* pp. 74-81, 2004.

[42] C. D. Manning, P. Raghavan, and H. Schütze, *An introduction to information retrieval*. Cambridge: Cambridge University Press, 2009.

[43] Á. F. Z. Rodríguez, C. G. F. Paniagua, J. L. A. Berrocal, and R. G. Díaz, "Recuperación de información utilizando el Modelo Vectorial," Universidad de Salamanca Mayo, 2002.

[44] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM,* vol. 18, pp. 613-620, 1975.

[45] A. Singhal, "Modern Information Retrieval: A Brief Overview," *IEEE Data Eng. Bull.,* vol. 24, pp. 35-43, 2001.

[46] R. M. Soto, R. A. G. Hernández, Y. Ledeneva, and R. C. Reyes, "Comparación de tres modelos de Texto para la Generación Automática de Textos," *Procesamiento de Lenguaje Natural,* vol. 43, pp. 303-311, 2009.

[47] E. Greengrass. (2000). *Information Retrieval: A Survey*. Available: https://www.csee.umbc.edu/csee/research/cadip/readings/IR.report.120600.book.pdf

[48] R. M. Alguliev, R. M. Aliguliyev, M. S. Hajirahimova, and C. A. Mehdiyev, "MCMR: Maximum coverage and minimum redundant text summarization model," *Expert Systems with Applications,* vol. 38, pp. 14514-14522, 2011.

[49] S. I. Grossman, *Álgebra Lineal*: Editorial Offset, S.A. de C.V, Durazno No. 1 esq. Ejido, Col. Las Peritas, Tepepan Xochimilco, C.P. 16010 México D.F, 2001.

[50] Y. Gong and X. Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis," *24th annual international ACM SIGIR conference on Research and development in information retrieval,* vol. New Orleans, Louisiana, USA, pp. 19-25, 2001.

[51] H. B. Christensen, *Estadística Paso a Paso*: Editorial Trillas, S.A de C.V. Av. Río Churubusco 385, Col. Pedro María Anaya, Deleg. Benito Juárez, 03340, México, D.F., 1983.

[52] K. S. Pratt, "Design Patterns for Research Methods: Iterative Field Research," in *Association for the Advancement of Artificial Intelligence*, 2009.

[53] Francisco J. Rodriguez, Carlos García-Martínez, and M. Lozano, "Hybrid Metaheuristics Based on Evolutionary Algorithms and Simulated Annealing: Taxonomy, Comparison, and Synergy Test," *IEEE Transaction on Evolutionary Computation,* vol. 16, pp. 787-800, 2012.

[54] E.-G. TALBI, "A Taxonomy of Hybrid Metaheuristics," *Journal of Heuristics,* vol. 8, pp. 541-564, 2002.

[55] H. Oliveira, R. Ferreira, R. Lima, R. D. Lins, F. Freitas, M. Riss*, et al.*, "Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization," *Expert Systems with Applications,* vol. 65, pp. 68-86, 2016.

[56] A. Bossard, M. Genereux, and T. Poibeau, "Description of the LIPN Systems at TAC 2008: Summarizing Information and Opinions," *Text Analysis Conference (TAC-2008),* pp. 282-191, 2008.

[57] M. Mendoza, C. Cobos, and E. León, "Extractive Single-Document Summarization Based on Global-Best Harmony Search and a Greedy Local Optimizer," in *Advances in Artificial Intelligence and Its Applications: 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Cuernavaca, Morelos, Mexico, October 25-31, 2015, Proceedings, Part II*, O. Pichardo Lagunas, O. Herrera Alcántara, and G. Arroyo Figueroa, Eds., ed Cham: Springer International Publishing, 2015, pp. 52-66.

[58] A. Abuobieda, N. Salim, A. T. Albaham, A. H. Osman, and Y. J. Kumar, "Text summarization features selection method using pseudo Genetic-based

model," *2012 International Conference on Information Retrieval & Knowledge Management,* pp. 193-197, 2012.

[59] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *18th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, Washington, United States, pp. 68-73, 1995.

[60] Mr.Vikrant Gupta, Ms Priya Chauhan, D. S. Garg, Mrs. Anita Borude, and P. S. Krishnan, "An Statistical Tool for Multi-Document Summarization," *International Journal of Scientific and Research Publications,* vol. Volume 2, 2012.

[61] N. I. o. S. a. Technology. (2008). *NIST Covering Array Tables - About these pages*. Available: http://math.nist.gov/coveringarrays/coveringarray.html

[62] M. Mendoza, S. Bonilla, C. Noguera, C. Cobos, and E. León, "Extractive single-document summarization based on genetic operators and guided local search," *Expert Systems with Applications,* vol. 41, pp. 4158-4169, Julio, 2014 2014.

[63] Mari Vallez and R. Pedraza-Jimenez, "El Procesamiento del Lenguaje Natural en la Recuperación de Información Textual y áreas afines," *Hipertext.net,* vol. 5, 2007.

[64] 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.

[65] 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.

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

[67] Christopher D. Manning, Prabhakar Raghavan, and H. Schütze, *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England ed.: Cambridge University Press, Cambridge England, 2009.

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

[69] A. S. Foundation. (2011, Marzo). *Apache Lucene Core*. Available: http://lucene.apache.org/

[70] Carlos Cobos, J. Perez, and C. Estupiñan, "Una Revisión de la Búsuqeda Armónica," *Revista Avances en Sistemas e Informática,* vol. 8, p. 14, 2011.

[71] J. Tang, L. Yao, and D. Chen, "Multi-topic based Query-oriented Summarization," *SIAM international conference on data mining,* pp. 1148-1159, 2009.

[72] L. Huang, Y. He, F. Wei, and W. Li, "Modeling Document Summarization as Multi-objective Optimization," *Third International Symposium on Intelligent Information Technology and Security Informatics,* pp. 382-386, 2010.

[73] A. Haghighi and L. Vanderwende, "Exploring Content Models for Multi-Document Summarization," *Conference of the North American Chapter of the ACL,* pp. 362-370, 2009.

[74] Portal ISO. (21-5-2018). *Familia Normas ISO 25000*. Available: http://iso25000.com/index.php/normas-iso-25000

[75] S. E. Harpe, "How to analyze Likert and other rating scale data," *Currents in Pharmacy Teaching and Learning,* vol. 7, pp. 836-850, 2015.