GENERACION AUTOMATICA DE RESUMENES EXTRACTIVOS GENERICOS DE UN DOCUMENTO BASADO EN N-GRAMAS SINTACTICOS NO CONTINUOS

ANDRES MAURICIO SALAZAR PIEDRAHITA

Tesis de Maestría en Computación

Director: PhD. Carlos Alberto Cobos Lozada

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Grupo de I+D en Tecnologías de la Información (GTI)

Línea de Investigación: Sistemas Inteligentes y Recuperación de la Información

Popayán, diciembre de 2019

ANDRES MAURICIO SALAZAR PIEDRAHITA

GENERACION AUTOMATICA DE RESUMENES EXTRACTIVOS GENERICOS DE UN DOCUMENTO BASADO EN N-GRAMAS SINTACTICOS NO CONTINUOS

Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de

Magíster en
Computación

Director
Director: PhD. Carlos Alberto Cobos Lozada

Popayán
2019

Agradecimientos

A la PhD. Martha Eliana Mendoza por sus asesorías y apoyo en el inicio de la investigación, sus conocimientos fueron importantes.

Al PhD. Carlos Alberto Cobos Lozada, por su invaluable legado de conocimientos que permitieron el aprendizaje y desarrollo de la investigación. Espero continuar aprendiendo cada día más desde otras instancias de la investigación.

Al Instituto Politécnico Nacional de México, en especial al PhD. Alexander Gelbukh y al PhD. Grigori Sidorov por sus orientaciones en el desarrollo de esta investigación.

A mi madre Miryam Piedrahíta Galvis, mi padre Oscar Salazar Valencia a quienes dedico cada logro alcanzado.

A mi amiga y compañera fiel Diana Arias Agredo, mis hijas Danna Alejandra Salazar Arias y Diana Carolina Salazar Muñoz por su paciencia, apoyo y fortaleza cuando debí apartarme del hogar para investigar.

A mis familiares y amigos por su admiración y orgullo al reconocer el esfuerzo y triunfos alcanzados.

Resumen Estructurado

El Procesamiento de Lenguaje Natural (PLN) y la Recuperación de Información (RI) utilizan modelos de representación como: booleano, probabilístico y espacio vectorial (más usado en la actualidad) para agrupar, buscar, clasificar y determinar el grado de relevancia de las características (tokens, términos, frases, conceptos, entre otros) de un documento, luego con diversos enfoques, entre ellos: estadísticos, probabilísticos, basados en grafos, conectividad de textos, retórica del discurso, reducción algebraica, metaheurísticas de optimización, métodos de aprendizaje de máquina, entre otros enfoques, deciden qué características deben ser parte de un resumen generado en forma automática para dicho documento. Aunque los resultados de las investigaciones previas mejoran día a día, aún distan de los resúmenes que los seres humanos pueden elaborar.

Los *N-gramas* en PLN, son secuencias de *N* elementos textuales (fonemas, palabras, lemas, etiquetas gramaticales, entre otros) construidos según su orden de aparición en el texto fuente. Los *N-gramas sintácticos* son un nuevo concepto de N-gramas y se construyen siguiendo las rutas del árbol sintáctico, concepto muy utilizado en tareas del RI para detección de autoría. El presente trabajo integró el uso de los N-gramas sintácticos (continuos y no continuos) en el modelo espacio vectorial para identificar las relaciones que los términos tienen con su contexto (información lingüística que no está disponible con los N-gramas tradicionales) y con ello mejorar la calidad de los resúmenes que se generan automáticamente. La comparación realizada con la representación por bolsa de palabras y N-gramas tradicionales, mostró resultados prometedores para los algoritmos basados en grafos (LexRank y LexRank Continuo), los cuales obtienen mejores resultados cuando la matriz de frases por términos se realiza con N-gramas sintácticos no continuos de 2, 3 o 4 gramas. Respecto al algoritmo ESDS-GHS-GLO, los mejores resultados se alcanzaron con N-gramas sintácticos no continuos de 1 grama, usando como esquema de representación del documento el centroide de todos los N-gramas. Además, todos los tipos de sintagmas (nominal, adjetival, verbal,

preposicional y adverbial) aportan información para definir si una frase debe o no hacer parte del resumen del documento.

Palabras claves: resumen extractivo, modelo espacio vectorial, n-grama sintáctico, árbol sintáctico, grafo, metaheurística, recuperación de información.

Structured Abstract

Natural Language Processing (NLP) and Information Retrieval (IR) use representation models such as boolean, probabilistic and vector space to group, search, classify and determine the degree of relevance of the characteristics (tokens, terms, phrases, concepts, among others) of a document, then with various approaches, among them: statistical, probabilistic, graph-based, text connectivity, speech rhetoric, algebraic reduction, optimization metaheuristics, machine learning methods, among other approaches, decide which characteristics should be part of the automatic summary, but their results are still far from the summaries that human beings can produce, although significant progress has been made.

N-grams in PLN are sequences of N textual elements (phonemes, words, lemmas, grammatical labels, among others) constructed according to their order of appearance in the source text. Syntactic N-grams are a new concept of traditional N-grams and it construct following the paths of the syntactic tree, a concept widely used in IR tasks for authorship detection. The present work integrated the use of syntactic n-grams (continuous and non-continuous) in the vectorial space model to identify the relations that terms have with their context (linguistic information that is not available with traditional n-grams) and with it improve the quality of the summaries that are generated automatically. The comparison made with the representation by bag of words and traditional N-grams, showed promising results for the algorithms based on graphs (LexRank and Continuous LexRank), which obtain better results when the matrix of phrases by terms is made with non-continuous syntactic n-grams of 2, 3 or 4 grammes. With respect to the ESDS-GHS-GLO algorithm, the best results it achieved with non-continuous syntactic n-grams of 1 gram and using the centroid of all n-grams as a representation scheme of the document. In addition, all types of syntagmas (nominal, adjectival, verbal, prepositional and adverbial) provide information to define whether or not a phrase should be part of the document's summary.

Keywords: extractive summary, vector space model, syntactic n-gram, syntactic tree, graph, metaheuristics, information retrieval.

Tabla de Contenido

Lista de Figuras

Lista de Tablas

Lista de Ecuaciones

Lista de Algoritmos

1 Introducción

La transformación del texto impreso a digital ha generado un exceso de información por el crecimiento exponencial en la producción de documentos; generalmente compartidos en redes como Internet. La comunidad científica y académica, propuso una alternativa para contrarrestar este exceso de información mediante la generación automática de resúmenes y la definió como "*el proceso de destilar la información más importante de una fuente (o de varias fuentes) para producir una versión abreviada destinada a un usuario (usuarios) determinado y para una tarea (tareas) determinada"* (traducción libre) (Mani and Maybury 1999), proceso posible dado que un resumen es *"una transformación de un texto fuente a un texto más corto por reducción de su contenido mediante selección y/o generalización de lo que es importante en el texto fuente"* (traducción libre) (Jones 1999).

La generación automática de resúmenes de texto día a día se utiliza con varios propósitos, por ejemplo: i) análisis de textos, para determinar el contenido clave (resumen), i) análisis de sentimientos para conocer la naturaleza del comentario sobre un tema o determinar que motiva el comentario, iii) análisis investigativo para identificar cuáles son los casos particulares de un tema específico, iv) clasificación del contenido para establecer el tema, v) detección de autoría para identificar patrones de escritura, longitud de las palabras o frases utilizadas, la riqueza del vocabulario, la frecuencia de las palabras, entre otros propósitos. Asimismo, diferentes aplicaciones generan automáticamente resúmenes, por ejemplo: i) los motores de búsqueda como Google y Yahoo! despliegan un breve resumen de las fuentes (páginas Web, documentos, videos, entre otros), ii) sistemas de gestión de aprendizaje electrónico resumen los contenidos de los objetos de aprendizaje y demás recursos disponibles, iii) sistemas de visualización para dispositivos móviles que despliegan la información más relevante teniendo en cuenta el reducido tamaño de la pantalla, entre otros usos.

Aunque la generación automática de resúmenes ha tenido varias aplicaciones, aún es una tarea compleja y motiva investigaciones con diferentes enfoques, entre ellos:

estadísticos, probabilísticos, basados en grafos, conectividad de textos, retórica del discurso, reducción algebraica, metaheurísticas de optimización, métodos de aprendizaje de máquina (Mendoza and Leon Guzmán 2013), entre otros enfoques, los cuales lograron avances importantes pero sus resultados aún distan de los resúmenes que los seres humanos pueden elaborar.

La generación de resúmenes es un campo de investigación relacionado con la Recuperación de Información (RI) y el Procesamiento de Lenguaje Natural (PLN) donde se utilizan modelos de representación con esquemas de pesos como el booleano, el probabilístico y el espacio vectorial (Hiemstra 2009) para agrupar, buscar, clasificar y determinar el grado de relevancia de las características (tokens, términos, frases, conceptos, entre otros) de un texto. El modelo más comúnmente utilizado a la fecha, es el modelo espacio vectorial, por su sencillez, fácil implementación y buenos resultados obtenidos al representar documentos mediante el uso de vectores en un espacio lineal multidimensional (matriz de características por documentos). Los vectores representan los documentos y las dimensiones del vector representan las características en el documento (conocido también como bolsa de palabras). El cruce de las características contra los vectores representa el grado de relevancia según la fórmula de ponderación o ranking utilizada para determinar el peso.

Por otro lado, en PLN se describe a los *N-gramas* como secuencias de *N* elementos textuales (fonemas, palabras, lemas, etiquetas gramaticales, entre otros) construidos según su orden de aparición en el texto fuente y los ha utilizado como características en el modelo espacio vectorial para representar textos (Sidorov et al. 2014). Los *N-gramas sintácticos* son un novedoso concepto de N-gramas que se construyen siguiendo las rutas del árbol sintáctico. La ventaja de los N-gramas sintácticos es que introducen información puramente lingüística (sintáctica) en los métodos de aprendizaje de máquina y han demostrado que su uso en la atribución de autoría permiten obtener resultados superiores al uso de los N-gramas tradicionales pero con una desventaja, el tiempo requerido para la ejecución del análisis sintáctico previo (Sidorov 2013b).

En este trabajo se usan los N-gramas sintácticos para introducir información sintáctica (relaciones sintácticas entre palabras) y semántica en el modelo espacio vectorial al representar las características de un texto, previo al proceso de generación automática de resúmenes extractivos de un solo documento.

1.1 Definición del Problema

Trabajos como (Sidorov et al. 2014), (Sidorov 2013b), (Sidorov 2013a), (Sidorov 2013c), (Sidorov et al. 2013b), (Sidorov et al. 2013a) y (Posadas Durán et al. 2015) plantearon un nuevo concepto para definir las características del modelo espacio vectorial, los *N-gramas sintácticos, N-gramas con* información sintáctica (reglas de combinación) y semántica (significado, sentido e interpretación al combinar palabras) que han sido usados en diferentes tareas como por ejemplo la atribución de autoría, donde se utilizaron usando rutas continuas en los árboles sintácticos (*N-gramas sintácticos continuos*) dejando para futuros trabajos el análisis de las bifurcaciones en las rutas (*N-gramas sintácticos no continuos*). Cabe mencionar que los N-gramas sintácticos continuos son un caso particular de los N-gramas sintácticos no continuos (Sidorov 2013b).

El presente trabajo de investigación se enfocó en proponer un modelo de representación de documentos a partir de palabras relacionadas tanto sintáctica como semánticamente, aunque éstas no tengan una ruta continua, pero que sí cuenten con alguna ruta que las conecte (características para la representación). Se buscó dar respuesta a las siguientes preguntas: ¿Cómo representar un documento con N-gramas sintácticos no-continuos que permita generar automáticamente resúmenes extractivos genéricos de un documento para obtener resultados similares o superiores al estado del arte?¿Qué parámetros de construcción de los N-gramas sintácticos no-continuos (tamaño del N-grama y los elementos léxicos involucrados) permiten alcanzar resultados similares o superiores al estado del arte para la generación automática de resúmenes extractivos genéricos de un documento?.

1.2 Aportes

El modelo de representación basado en N-gramas sintácticos continuos y no continuos buscó aportar nuevo conocimiento a la comunidad científica y académica del PLN y RI, específicamente en métodos utilizados para la generación automática de resúmenes extractivos genéricos de un documento que utilizan el modelo espacio vectorial para representar las características de un texto. Los N-gramas sintácticos continuos y no continuos aportan información sintáctica al identificar las relaciones que los términos tienen con su contexto y que no está disponible con los N-gramas tradicionales o bolsa de palabras, información que se consideró útil para obtener resúmenes con calidad que permitieron alcanzar resultados similares o superiores a los reportados por los métodos seleccionados basados en grafos (LexRank y LexRank Continuo) y optimización (FSP y EDS-GHS-GLO).

Se logró un aporte en innovación al introducir el modelo de representación basado en N-gramas sintácticos continuos y no continuos en las matrices de términos que utilizan tres de los algoritmos implementados en el Framework del Grupo de I+D en Tecnologías de la Información (GTI) para generar resúmenes automáticos de un documento. Con esto se aportó a la línea de investigación de Gestión de la Información y Sistemas Inteligentes del GTI de la Universidad del Cauca, motivando a la academia a realizar trabajos futuros relacionados con el uso de N-gramas sintácticos continuos y no continuos en métodos para la generación automática de resúmenes extractivos genéricos de un documento.

1.3 Objetivos

A continuación, se presentan los objetivos desarrollados en el presente trabajo de grado de tesis de maestría de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad el Cauca.

1.3.1 Objetivo general

Proponer un modelo de representación de documentos basado en N-gramas sintácticos no-continuos para soportar la generación automática de resúmenes extractivos genéricos de un documento buscando obtener resultados similares o superiores a los reportados en el estado del arte.

1.3.2 Objetivos Específicos

- Definir un modelo de representación para un documento basado en N-gramas sintácticos no-continuos, que permita identificar la información sintáctica entre palabras, aunque éstas no tengan una ruta continua y permita obtener resultados similares o superiores a los del estado del arte en la generación automática de resúmenes extractivos de un documento.
- Adaptar los algoritmos de generación automática de resúmenes de un documento usando el Framework del Grupo GTI de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca para que utilice el modelo de representación propuesto.
- Adaptar los algoritmos de generación automática de resúmenes de un documento del estado del arte, Mejor Búsqueda Armónica Global (metaheurístico), LexRank (Grafos) y Análisis Semántico Latente (Reducción Algebraica); para que utilicen el modelo de representación propuesto.
- Evaluar la calidad promedio de los resúmenes generados por los algoritmos adaptados con el modelo de representación propuesto, utilizando documentos de noticias de la Conferencia de Entendimiento del Documento (DUC 2001 y DUC 2002) y las métricas ROUGE-N y ROUGE-SU.

1.4 Resultados Obtenidos

El primer resultado, es la ***Monografía de la Tesis de Grado*** (el presente documento) en la cual se detalla la información relevante relacionada con el marco teórico, el estado del arte, las actividades que se llevaron a cabo para alcanzar los objetivos planteados para la investigación, las conclusiones y los trabajos futuros.

El ***modelo de representación de documentos con N-gramas sintácticos continuos y no-continuos*** y su implementación en una ***herramienta software*** con una interfaz de usuario para cargar el análisis sintáctico de Freeling en formato XML. La herramienta genera las matrices de términos por documento con los N-gramas sintácticos que representan el modelo espacio vectorial de un dataset. Además, la herramienta permite configurar los parámetros para generar varios tipos de matrices, ellos son: 1) idioma del documento, 2) inclusión o no de “stop words”, 3) análisis por lema o palabra, 4) tamaño del n-grama (de 1 a 7 gramas), 5) tipo de representación (vector o centroide), 6) tipo de n-grama sintáctico (léxicos, categorías gramaticales y relaciones sintácticas), 7) tipos de sintagmas a incluir en la representación (nominal, adjetival, verbal, preposicional, adverbial o la combinación) y 8) ordenar los N-gramas sintácticos a incluir en cada matriz. El código fuente de esta herramienta se entrega en un CD anexo, además del instalador del analizador sintáctico Freeling, la guía para replicar el escenario de pruebas, los dataset utilizados para la experimentación y los ejemplos de las matrices de términos obtenidas.

Un ***artículo de investigación*** con los resultados obtenidos durante la experimentación, el análisis, su comparación con el estado del arte, conclusiones y el t|rabajo futuro. El artículo se publicó en una revista internacional reconocida por el PUBLINDEX de COLCIENCIAS en categoría B y tiene la siguiente referencia: Salazar-Piedrahíta Andrés-Mauricio, Cobos-Lozada Carlos-Alberto. Generación automática de resúmenes extractivos genéricos de un documento basado en n-gramas sintácticos no continuos. Rev Ibérica Sist e Tecnol Informação 2019; E22:323–36. Disponible en línea en

https://search.proquest.com/openview/69fcef4b61d6ec86ce73ded1c03a1ed0/1?pq-origsite=gscholar&cbl=1006393.

1.5 Organización del resto del documento

La presente Monografía está organizada de la siguiente manera:

Capítulo 1. Presenta una introducción al tema, el contexto del problema, los objetivos definidos para la investigación, los resultados obtenidos y la organización de la monografía. Corresponde al presente capitulo.

Capítulo 2. Describe el marco teórico, los conceptos básicos para comprender el tema del trabajo realizado.

Capítulo 3. Presenta el estado del arte en generación automática de resúmenes extractivos genéricos, los algoritmos utilizados por la comunidad científica y académica para generar resúmenes extractivos de un documento, las métricas utilizadas para evaluar los resúmenes candidatos y los N-gramas Sintácticos.

Capítulo 5. Describe el diseño experimental, los resultados de la experimentación, el análisis de los resultados obtenidos y la comparación de los resultados frente al estado del arte.

Capítulo 6. Presenta las conclusiones y el trabajo futuro que motiva nuevas investigaciones en el área de los sistemas inteligentes, el aprendizaje de máquina y la R.I (proceso de clasificación) para la generación de resúmenes extractivos de un solo documento.

2 Marco Teórico

2.1 Representación de texto

En Recuperación de la Información (RI) comúnmente se utilizan modelos de representación para seleccionar las características (tokens, términos, frases, conceptos, entre otros) que identifican el contenido o la temática del texto, además del grado de relevancia (peso, ponderación o ranking) de estas características en los documentos. No todas las características deben tener un mismo grado de relevancia o ser consideradas al resumir o identificar el tema de un texto, por ejemplo, en un sistema de búsqueda de documentos, algunas palabras comunes como los pronombres, conjunciones, artículos determinados o indeterminados, contracciones, preposiciones, ciertos verbos y adverbios no se tienen en cuenta porque no aportan capacidad de selección o discriminación entre la consulta del usuario y los documentos, estas palabras se conocen como palabras vacías o “stop words”. En otras ocasiones el grado de relevancia de las palabras se afecta por la longitud de los textos o la cantidad de los documentos. Trabajos como los de Salton et al. (Salton, Wong, and Yang 1975) presentan diferentes fórmulas de ponderación y resultados experimentales, otras propuestas como la de Robertson et al. con la función Okapi BM25 (Robertson and Walker 1994) (Robertson, Zaragoza, and Taylor 2004) tienen en cuenta la frecuencia de los términos en los documentos, la importancia relativa de cada término y la longitud del documento, mostrando ser más robusta y con mejores resultados en diferentes aplicaciones.

La literatura científica destaca tres modelos de representación comúnmente utilizados: el booleano, el probabilístico y el espacio vectorial (Hiemstra 2009).

Modelo booleano: Se fundamenta en la teoría de conjuntos y en el álgebra de Boole. La presencia de un término en un documento es binaria (1 indica que está presente y 0 no presente) y por esto no contempla la posibilidad de establecer diferentes grados de pertenencia. El modelo elimina las palabras vacías entre ellas

los números, las preposiciones, conjunciones y algunos verbos (Baeza Yates and Ribeiro Neto 1999). La **Figura 1** muestra la representación de este modelo.

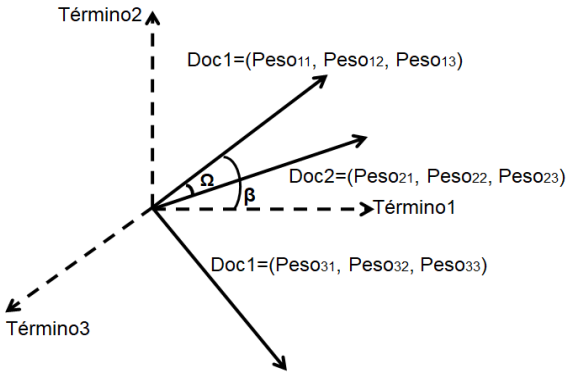
	Término1	Término2	Término3	Término4	…	TérminoM
Documento1	1	1	0	1		1
Documento2	0	1	0	0		1
…						
DocumentoN	0	0	1	1		1

***Figura 1**. Representación del Modelo Booleano*

Independencia binaria (Probabilístico): es el modelo de recuperación más sencillo de la familia de los modelos probabilísticos. Se fundamenta en la teoría de las probabilidades para estimar la relevancia de un documento. Un documento es relevante si su probabilidad de ser relevante es mayor que la probabilidad de no ser relevante. Según la consulta planteada por el usuario, los documentos de la colección se clasifican en dos grupos; 1) Conjunto de Documentos Relevantes y 2) Conjunto de Documentos Irrelevantes (clasificación binaria de los documentos) (Baeza Yates and Ribeiro Neto 1999). Teniendo en cuenta que la probabilidad de relevancia de un documento es estimada a partir de las probabilidades de los términos que lo componen, se puede calcular el grado de similitud de un documento con una consulta. El modelo probabilístico se basa en un proceso iterativo que inicia con un primer conjunto de documentos relevantes, el cual se refina iterativamente en función de la información que proporciona el usuario en relación con los documentos que considera relevantes y no relevantes, consiguiendo ordenar la colección en orden descendente de probabilidad de relevancia en relación con la consulta. Este modelo elimina la interacción con el usuario, asumiendo que los primeros n resultados son relevantes y de esta forma logra hacer el proceso de búsqueda en forma automática (Baeza Yates and Ribeiro Neto 1999).

Espacio Vectorial: Representa uno o varios documentos por medio de un vector de pesos de términos (ver **Figura 2**) donde cada término, puede ser una palabra, expresión o token (unidad mínima al descomponer un texto). De esta forma, si un término pertenece a un texto, obtiene un valor dependiendo de su importancia dentro del texto según la técnica de ponderación de términos utilizada (Salton et al.

1975). El modelo espacio vectorial es usado en los sistemas de recuperación de información desde los inicios de los años 70 (Salton et al. 1975) para soportar tareas como: ordenamiento de resultados basados en una consulta, clasificación de documentos u oraciones, agrupamiento de documentos u oraciones, entre otras.



	Término1	Término2	Término3
Doc1	$Peso_{11}$	$Peso_{12}$	$Peso_{13}$
Doc2	$Peso_{21}$	$Peso_{22}$	$Peso_{23}$
Doc3	$Peso_{31}$	$Peso_{32}$	$Peso_{33}$

***Figura 2**. Representación del Modelo Espacio Vectorial*

2.2 Okapi BM25

Una de las limitaciones del modelo de independencia binaria (Probabilístico) es que fue diseñado para ser usado sobre textos cortos y de longitud similar, por ejemplo, repositorios de resúmenes, por ello el modelo no presta atención a factores como la frecuencia de los términos dentro del documento porque sólo tiene en cuenta la presencia o no del término y la longitud del mismo. Por lo tanto, si se quiere trabajar con otro tipo de colecciones más generales, se hace necesario contar con un modelo probabilístico más potente que si tenga en cuenta estos factores. el BM25 es un modelo probabilístico de recuperación perteneciente a la familia de modelos Poisson-2. Estos modelos asumen que las apariciones de un término en un documento tienen una naturaleza aleatoria, de tal forma que un documento es visto como una secuencia aleatoria de términos. Dicha distribución puede aproximarse mediante una distribución Poisson, pero además asume que dicha distribución es diferente en aquellos documentos que tratan sobre el tema de ese término –llamados documentos élite–, y aquéllos que no tratan sobre el tema del término –llamados no-élite–, por lo que han de considerarse dos distribuciones de Poisson

diferentes, de ahí la denominación 2-Poisson (Robertson and Walker 1994) (Robertson et al. 2004). Okapi BM25 tiene en cuenta conceptos básicos como la importancia del término en la frase (TF) y la importancia del término en la colección (IDF), pero hace esto teniendo en cuenta la longitud. El peso del término en la frase que se usa en esta investigación está dado por la **Ecuación 1**, donde $k_1 = 2$, $b = 0.75$, $F_{i,j}$ es la frecuencia del término i en la frase j, N es el número de frases en el documento, n_i es el número de frases en las que aparece el termino i, $|Frase_j|$ es el número de términos de la frase j y $avgD$ es el número promedio de términos en las frases del documento.

$$w_{i,j} = \frac{(k_1 + 1) * F_{i,j}}{F_{i,j} + k_1 * (1 - b + b * \frac{|Frase_j|}{avgD})} * Ln(N/n_i)$$

***Ecuación 1**. Okapi BM25*

2.3 N-gramas

Los *N-gramas* en PLN, son secuencias de *N* elementos textuales (fonemas, palabras, lemas, etiquetas gramaticales, entre otros) construidos según su orden de aparición en el texto fuente (Salton et al. 1975). La *N* representa cuantos elementos se toman para construirlos (longitud de la secuencia), de esta forma se pueden generar bigramas (2-grama), trigramas (3-grama), cuatrigramas (4-grama), entre otros (Sidorov 2013a) (Sidorov 2013c) (Sidorov et al. 2013a). Por ejemplo, los 2-gramas basados en palabras del texto "*la casa en el árbol*" son: "*la casa", "casa en", "en el"* y "*el árbol"*. Los N-gramas tradicionales se basan únicamente en la posición de las palabras en el texto, pero también se pueden usar N-gramas basados en la información sintagmática (Sidorov 2013c), es decir, la relación de las palabras respecto del núcleo sintáctico o palabra fundamental del *sintagma*. Una palabra o grupo de palabras que constituyen una unidad sintáctica y cumplen una función determinada según su núcleo sintáctico, se le denomina *sintagma*. Por ejemplo: sintagma verbal (SV), sintagma adjetival (SADJ), sintagma nominal (SN), sintagma

adverbial (SADV), sintagma preposicional (SP), entre otros, como se muestra en la **Figura 3** para la frase “este muchacho va muy lejos con su moto”.

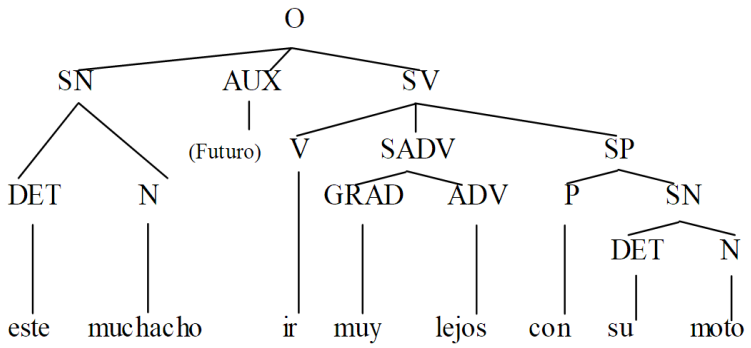


***Figura 3**. N-grama tradicional*

2.4 Taxonomía de los N-gramas

Los N-gramas se clasifican en dos grupos según el orden que se sigue para construirlos (Sidorov 2013a), como se observa en la **Figura 4**:

- **N-grama Lineal**. Se forman a nivel superficial del texto donde los elementos obligatoriamente siguen el orden lineal unos a otros (secuencialmente), su desventaja es que ignoran la información sintáctica (Sidorov 2013a).
- **N-grama No Lineal (o No Continuo)**. Se crean siguiendo las rutas en un árbol sintáctico, esto permite evitar el ruido introducido por la estructura superficial del lenguaje, el cual puede aparecer porque en el nivel superficial las palabras no relacionadas sintácticamente pueden aparecer juntas, fenómeno que es posible controlar si se sigue las relaciones sintácticas reales que unen las palabras, aunque éstas no sean vecinos inmediatos (Sidorov 2013a).

Los N-gramas No Lineales se clasifican de la siguiente forma (Sidorov 2013a) (ver **Figura 4**):

- **Skip-gramas**. Forma secuencias de elementos de manera aleatoria saltando algunas gramas. Su desventaja es que contienen más ruido que los N-gramas tradicionales. En (Sidorov 2013c) se utilizan los Skip-gramas con mayor frecuencia llamados secuencias frecuentes maximales, sin embargo, la

construcción de estos requiere de algoritmos sofisticados y su interpretación es difícil. La realidad lingüística que les corresponde no va más allá de la búsqueda de algunas combinaciones de palabras (Sidorov 2013a).

- **N-gramas filtrados.** Secuencias de elementos para construir N-gramas filtrados a partir de la matriz *tf-idf*, es decir, la frecuencia del término y frecuencia inversa del documento, lo cual es una medida numérica que expresa cuan relevante es una palabra para un documento o colección de documentos (Sidorov 2013a).
- **N-gramas generalizados.** En lugar de utilizar la palabra del texto se usa el primer término de una lista de sinónimos (synset), o cambiar las palabras por conceptos más generales extraídos de una ontología y después construir los N-gramas basados en esos conceptos (Sidorov 2013a).
- **N-gramas sintácticos.** Se basan en información sintáctica (relaciones entre palabras) y se obtienen siguiendo el orden en las rutas de los árboles sintácticos. Contienen mayor información de naturaleza lingüística (sintáctica) que los N-gramas tradicionales (Sidorov et al. 2013a). La **Figura 5** presenta un ejemplo de un fragmento (tomado de un libro de Julio Verne) al cual se le aplicó el parseador de Freeling para construir el árbol sintáctico en el cual se pueden observar las parejas de palabras y la relación sintáctica entre ellas, es decir, la información sintáctica. Sobre cada flecha aparece el nombre de la relación sintáctica correspondiente. La lista de palabras al lado derecho del árbol es: la etiqueta de la relación sintáctica, la palabra principal con su posición en el fragmento y la palabra dependiente con su posición en el fragmento.

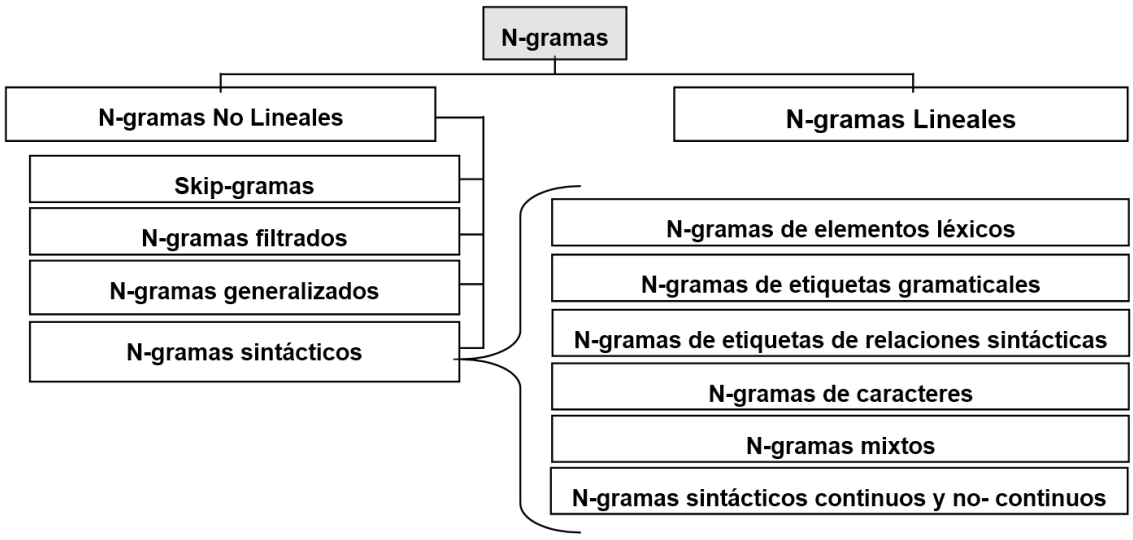


***Figura 4**. Taxonomía de los N-gramas*

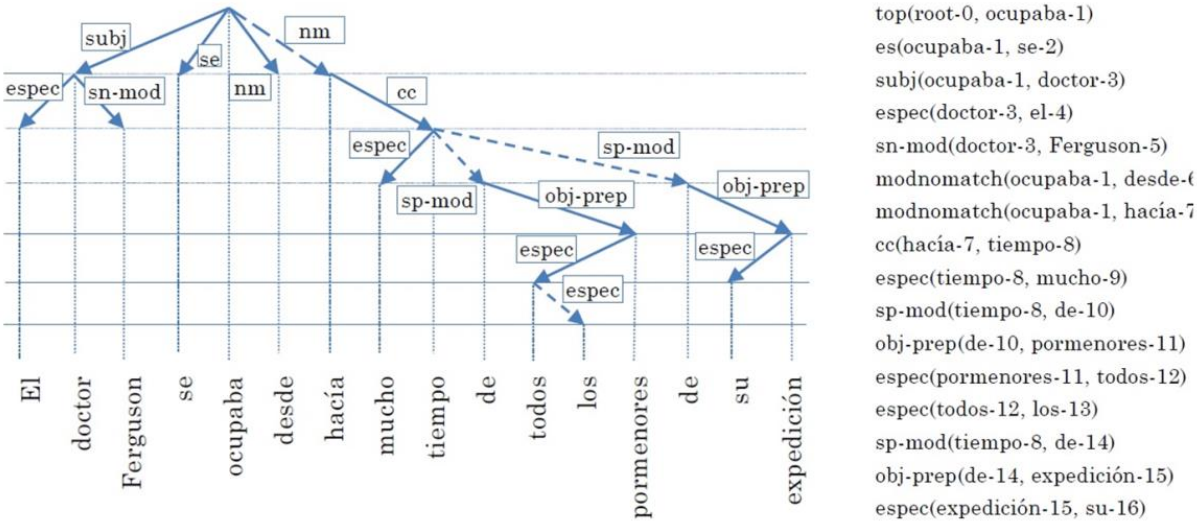


***Figura 5**. Árbol sintáctico en español con etiquetas y bigramas*

Los N-grama sintácticos se pueden diferenciar según los elementos que los componen en (Sidorov 2013a) (Sidorov et al. 2013b) (ver **Figura 4**):

- **N-gramas de elementos léxicos.** La opción más directa son las palabras, pero también se puede utilizar los lemas que son las formas normalizadas de las palabras y se obtienen después del procedimiento de normalización morfológica (lematización). Otra opción es utilizar las raíces unificadas de palabras (el proceso de obtención de las raíces se denomina *stemming*). En este sentido el

lema y la raíz unificada tienen la misma función: representan todo el conjunto de las formas gramaticales que corresponden a una palabra (Sidorov 2013a).

- **N-gramas de elementos gramaticales.** De manera similar, en lugar de palabras se puede utilizar la información gramatical (etiquetas gramaticales) correspondiente a cada palabra. Por ejemplo, para el caso del español se puede utilizar las etiquetas que producen los analizadores morfológicos o sintácticos, como FreeLing: NCFS000, VMII1S0, entre otras (Sidorov 2013a) (Sidorov 2013c).
- **N-gramas de etiquetas de relaciones sintácticas.** Se utilizan las etiquetas de relaciones sintácticas que están presentes en el árbol sintáctico o en inglés SR-tags (syntactic relations tags) como elementos de los N-gramas (Sidorov 2013a) (Sidorov 2013c).
- **N-gramas de caracteres.** Otra posibilidad consiste en utilizar los caracteres como elementos de los N-gramas, por ejemplo, en la frase "*Juan lee"*, se encuentran los siguientes bigramas: "ju", "ua", "an", "n ", " l", "le" y "ee". En este caso se utiliza el espacio entre palabras como un elemento de N-gramas, también se puede utilizar los signos de puntuación. Sin embargo, para algunas tareas es mejor no considerar los caracteres auxiliares (Sidorov 2013a).
- **N-gramas mixtos.** Algunos elementos de un N-grama pueden ser de un tipo, y otros elementos del mismo N-grama pueden ser de otro tipo. Los caracteres no pueden participar en la construcción de los N-gramas mixtos porque representan partes de la palabra y su naturaleza semántica y sintáctica es diferente a los otros tipos de elementos textuales (Sidorov 2013a).
- **N-gramas sintácticos continuos.** Secuencias de *N* elementos textuales relacionados construidas siguiendo la ruta de un árbol sintáctico. Durante su construcción no se permiten bifurcaciones en las rutas sintácticas, desde cualquier punto de la ruta puede moverse exactamente a un punto siguiente. Los N-gramas sintácticos continuos son un caso particular de los N-gramas sintácticos no-continuos (Sidorov 2013a). La **Figura 6** representa un 5-grama

sintáctico en una ruta de un árbol sintáctico de un fragmento de la novela de Julio Verne (*y di par de vueltas*). El 5-grama es construido a partir de relaciones sintácticas entre palabras una tras otra en la ruta y corresponde a las frases que están en las flechas más gruesas.

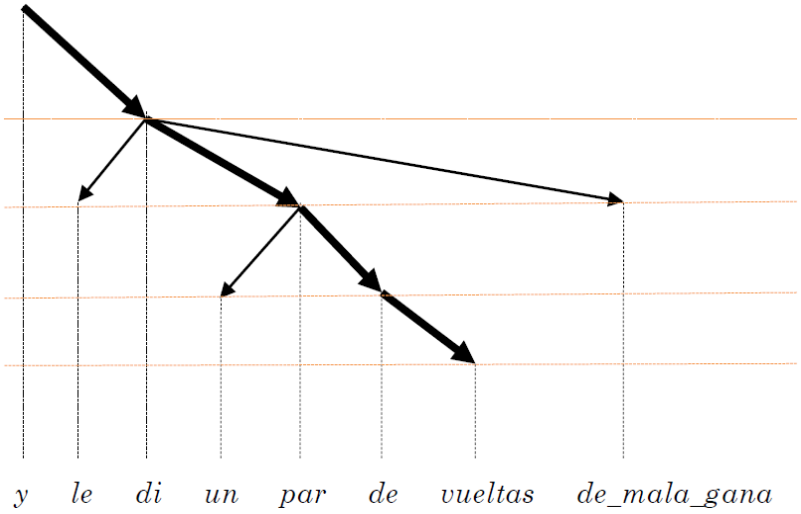


***Figura 6**. 5-grama sintáctico continuo en el fragmento del árbol sintáctico. (Extraído de* (Sidorov 2013a)*)*

- **N-gramas sintácticos no-continuos o árbol-gramas (t-gramas).** Secuencias de *N* elementos textuales relacionados, construidas siguiendo la ruta de un árbol sintáctico, incluidas las bifurcaciones en las rutas. Se consideran todos los subárboles de longitud *N* de un árbol sintáctico. Este tipo de N-gramas une las palabras relacionadas tanto sintáctica (reglas de combinación) como semánticamente (significado, sentido e interpretación al combinar palabras), aunque éstas no tengan una ruta continua, pero sí cuenten con alguna ruta que las conecte. La **Figura 7** es un ejemplo de 5-grama (*y di par [un, de] vueltas*) que se conecta por valencias verbales (o patrones de rección, es decir, relación entre una palabra y otras relacionadas sintácticamente con ella), para el ejemplo, el verbo *dar* es trivalente y tiene los actantes: *quién* (yo)*, cuántas* (un par)*, qué* (vueltas)*,* por lo tanto es importante tenerlos en cuenta al mismo tiempo en un N-grama (Sidorov 2013a).

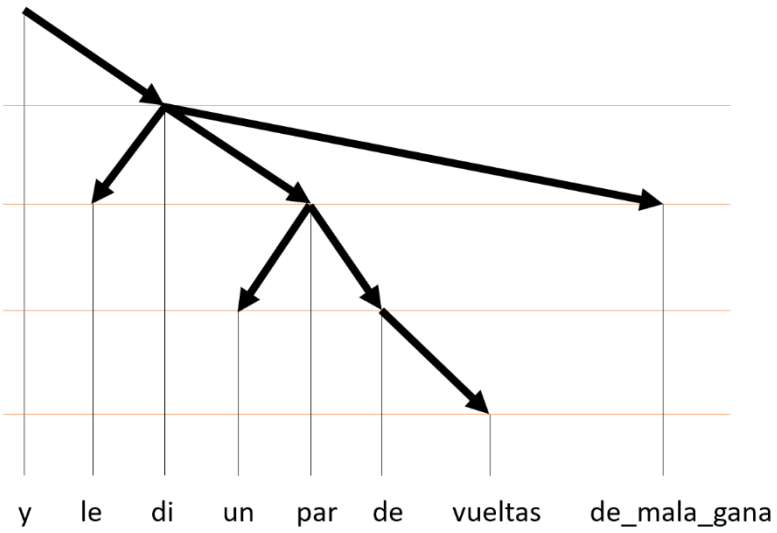


***Figura 7**. 5-gramas sintácticos no-continuos en el fragmento del árbol sintáctico. (Extraído de* (Sidorov 2013a)*)*

3 Estado del arte

3.1 Generación automática de resúmenes extractivos genéricos de un documento

Los primeros trabajos relacionados con generación automática de resúmenes se desarrollaron por investigadores de IBM alrededor de 1958 (Lunh 1958)(Baxendale 2010). En 1969 se propuso un sistema de extracción de oraciones que utilizó la frecuencia de las palabras y la posición de la oración, junto con otras dos características: presencia de palabras de referencia (como “importante” o “relevante”) y presencia de palabras del título del documento (Edmundson 1969)(Mendoza and Leon Guzmán 2013), desde entonces, se han propuesto y evaluado diversos métodos (Lloret and Palomar 2012)(Louis, Joshi, and Nenkova 2010) acorde a criterios como: el medio de información (textos, imágenes, videos o voz), el número de documentos del cual se va a generar el resumen (mono documento o multidocumento), el propósito (indicativo, describiendo brevemente la idea principal del documento; o informativo, buscando reemplazar el documento por una versión abreviada del contenido y las críticas, que reflejen el punto de vista del autor) de acuerdo a la audiencia objetivo (genérico, porque refleja el contenido principal de los documentos sin ninguna información adicional; basado en consulta, cuyo objetivo no sólo es extraer la información importante contenida en los documentos, sino también garantizar que la información extraída esté enfocada a la consulta) y finalmente el criterio del lenguaje soportado (mono lenguaje o multilenguaje) (Mani and Maybury 1999)(Jones 1999)(Jiménez, Gelbukh, and Sidorov 2014).

Considerando la forma como se obtienen los resúmenes, ellos pueden ser extractivos o abstractivos (Mendoza and Leon Guzmán 2013) (Montiel Soto et al. 2009).

El sistema extractivo, divide el texto fuente en fragmentos (palabras, oraciones, párrafos, entre otros) e identifica los más relevantes a partir de métodos de

selección y ponderación, luego se unen libremente, procurando perder la menor cantidad de información posible evitando la redundancia. El resultado final es un texto abreviado, uniendo los diferentes fragmentos relevantes o completando frases que puedan haber quedado incompletas. En este enfoque, que se ha estudiado ampliamente (Nenkova and McKeown 2011) (Lloret and Palomar 2012) se hace un análisis superficial de los textos, a nivel de palabras; por lo que, en general, los resúmenes no tienen coherencia y solo se da una idea de lo que es sobresaliente en el texto (Jiménez et al. 2014). Generalmente, las oraciones seleccionadas se presentan en el mismo orden en que aparecen en los documentos originales.

El sistema abstractivo, analiza el texto con mayor profundidad para comprenderlo y generar un nuevo resumen reescribiendo algunos o todos los fragmentos extraídos por medio de procesos de fusión, combinación o supresión de términos (Mani and Maybury 1999)(Jones 1999) con términos o frases que no necesariamente se encuentran en el texto origen (nueva redacción como lo harían los humanos). Se requiere de una representación que emule la comprensión humana del texto. Además, se ha demostrado que para propósitos indicativos (líneas o párrafos relevantes del texto sin necesidad de estar coherentemente relacionados) los resúmenes extractivos han sido adecuados, pero para otros propósitos como informativos (descripción de los aspectos relevantes y la relación lógica del tema tratado) o que el resumen se adapte a los intereses del usuario es necesario generar resúmenes abstractivos (Jiménez et al. 2014). Comprender el texto de forma más profunda implica redactar texto desde cero a partir de una base de conocimiento lingüístico del lenguaje. El sistema abstractivo es más complejo y consume mayor tiempo de cómputo, pero sus resultados aún no son significativamente mejores que los alcanzados por el sistema extractivo (López Condori and Salgueiro Pardo 2017).

3.2 Algoritmos para generación automática de resúmenes extractivos de un documento

Los sistemas extractivos han sido ampliamente estudiados y la literatura contiene una gran variedad de métodos para la generación automática de resúmenes de un documento, por ejemplo, basados en: estadísticas (Edmundson 1969)(Meena and Gopalani 2015), técnicas de reducción algebraica (Gong and Liu 2001)(Lee et al. 2009)(Steinberger and Ježek 2006)(Mashechkin et al. 2011)(Yeh et al. 2005), técnicas de aprendizaje de máquina (ML) (C. Aone, M. E. Okurowski, J. Gorlinsky 1999)(Conroy and O'leary 2001)(Hannah, Geetha, and Mukherjee 2011)(Kupiec, Pedersen, and Chen 1995)(Kyoomarsi et al. 2008)(Muratore et al. 2010)(Yousefi-Azar and Hamey 2017), conectividad de texto (Barzilay and Elhadad 1997)(Ibrahim and Elghazaly 2013)(Louis et al. 2010)(Marcu 1998)(Ono, Sumita, and Miike 1994), grafos (Amancio et al. 2012)(Chatterjee, Mittal, and Goyal 2012)(Erkan and Radev 2004)(Ledeneva, García-Hernández, and Gelbukh 2014)(Mihalcea and Tarau 2004)(Ferreira et al. 2013)(Wan 2010), agrupación (Aliguliyev 2007)(Ledeneva et al. 2011)(Yazhini and Vishnu 2014), metaheurísticas como algoritmos de búsqueda armónica (Shareghi and Hassanabadi 2008), optimización de enjambres de partículas (Rasim M. Alguliev, Aliguliyev, and Mehdiyev 2011a)(Alguliev, Aliguliyev, and Isazade 2013b)(Asgari, Masoumi, and Sheijani 2014)(Binwahlan, Salim, and Suanmali 2009), programación genética (Dehkordi and Kumarci 2009)(Uy et al. 2012), algoritmos genéticos (Chatterjee et al. 2012)(García-Hernández and Ledeneva 2013)(Litvak, Last, and Friedman 2010)(Qazvinian, Hassanabadi, and Halavati 2008), evolución diferencial (Abuobieda et al. 2013a), y algoritmos meméticos (Mendoza et al. 2014), optimización hibridada con k-means y un algoritmo de evolución diferencial adaptativa mono-objetivo (Alguliyev et al. 2019) o mapa autoorganizado (SOM) y optimización multiobjetivo utilizando tres metaheurísticas: exploración de la evolución diferencial autoorganizada, optimizador del lobo gris y algoritmo del ciclo del agua (Saini et al. 2019), y otras técnicas híbridas que combinan dos o más métodos.

El enfoque de reducción algebraica más comúnmente utilizado es el Análisis Semántico Latente (LSA), que extrae, representa y compara significados de palabras utilizando el análisis algebraico estadístico de un texto cuya hipótesis básica es que el significado de una palabra está determinado por su aparición frecuente junto a otras palabras o en contextos similares. LSA aplica el algoritmo de descomposición de valores singulares (SVD) u otro método de descomposición a la matriz de términos por oración, lo que permite captar las interrelaciones entre términos, de modo que los términos y las oraciones son agrupadas sobre una base "semántica" de palabras, en lugar de utilizar únicamente sus términos individuales (Gong and Liu 2001)(Ozsoy, Alpaslan, and Cicekli 2011)(Steinberger and Ježek 2006)(Yeh et al. 2005)(Mendoza et al. 2014). La Factorización de Matriz-No-Negativa (NMF) (Lee et al. 2009)(Tsarev, Petrovskiy, and Mashechkin 2011) es otra técnica de reducción algebraica no supervisada que utiliza componentes no negativos para realizar la descomposición; estos componentes están más cerca del proceso cognitivo humano, mostrando mejores resultados que el LSA.

Los métodos basados en conectividad de texto buscan identificar las relaciones entre los conceptos del documento. Utilizan estrategias tales como cadenas léxicas y estructuras retóricas. En los métodos basados en cadenas léxicas, la extracción de palabras candidatas del documento se determina utilizando un tesauro como WordNet (Barzilay and Elhadad 1997)(Pal and Saha 2014), luego comprueba palabra por palabra observando si se pueden incluir en una cadena léxica existente o crea una nueva cadena léxica, para finalmente identificar las cadenas más fuertes y extraer las oraciones más significativas con las cuales forman el resumen (Mendoza et al. 2014). Los métodos basados en estructuras retóricas (Marcu 1998)(Ono et al. 1994) extraen segmentos retóricos del documento original para formar una estructura en árbol, donde las unidades de texto constituyen nodos (clasificados como núcleo o satélite, según su grado de relevancia para el discurso). La puntuación de cada estructura retórica se establece de acuerdo con las métricas definidas por el algoritmo y las estructuras que obtienen las mejores puntuaciones que son incluidas en el resumen. Aunque este enfoque mantiene el resumen en contexto y con cohesión, la precisión disminuye a medida que aumenta la cantidad

de texto. Para superar esto, en (Ibrahim and Elghazaly 2013) se presenta un modelo híbrido que combina estructuras retóricas y un modelo espacial vectorial (VSM). Estos métodos tienen como desventajas el uso de técnicas complejas y la dependencia del lenguaje en el procesamiento del texto (Lloret and Palomar 2012).

Métodos de aprendizaje de máquina: Incluyendo técnicas como las redes bayesianas (C. Aone, M. E. Okurowski, J. Gorlinsky 1999)(Kupiec et al. 1995)(Wong, Wu, and Li 2008), redes neuronales artificiales (Muratore et al. 2010)(Svore, Vanderwende, and Burges 2007)(Yousefi-Azar and Hamey 2017), modelos ocultos de Markov (Conroy and O'leary 2001)(Mendoza et al. 2014), campos aleatorios condicionales (Shen et al. 2007), árboles de decisión y la lógica difusa (Kyoomarsi et al. 2008)(Hannah et al. 2011). Éstos requieren datos de entrenamiento para definir ponderaciones o probabilidades de características de texto y utilizar esos valores con los nuevos documentos para seleccionar e incluir en el resumen las frases con mayores probabilidades (Lloret and Palomar 2012). La desventaja de estas técnicas consiste en la necesidad de disponer de datos de entrenamiento difíciles de encontrar o crear, además de generar una dependencia del idioma en el que se redactan los documentos de entrenamiento.

En los métodos basados en grafos, las palabras clave u oraciones se representan como nodos en un grafo no dirigido, cuando dos oraciones son similares se conectan con un arco que tiene un peso asociado que indica la fortaleza de la conexión. Así se crea un grafo que representa las relaciones entre todas las oraciones del documento que permite ser iterado hasta que converja a un criterio, para luego ordenar y seleccionar las oraciones con mayor peso (Chatterjee et al. 2012)(Erkan and Radev 2004)(Mendoza et al. 2014). Adicionalmente, en (Ferreira et al. 2013) se combina el peso de cuatro tipos de arcos: Similitud, Similitud semántica, Resolución de la Correferencia y Relaciones con el Discurso. A diferencia de los estudios mencionados anteriormente, en (Ledeneva et al. 2014) las Secuencias Máximas Frecuentes (MFS) se representan como nodos. Las MFS son N-gramas frecuentes que no pertenecen a ningún otro n-grama frecuente, que al ser encontrados y seleccionados ofrecen la información más importante de un

documento. En (Wan 2010) se abordó de manera unificada la generación automática de resúmenes de un documento y múltiples documentos con un algoritmo basado en grafos, incluyendo en el resumen los nodos u oraciones que obtiene las mejores puntuaciones. Estos métodos no supervisados tienen la ventaja de ser independiente del lenguaje y de mejorar la cohesión en los resúmenes generados, de otro lado su mayor desventaja radica en el aumento de la complejidad computacional a medida que el número de nodos y arcos del grafo se incrementa. En (Erkan and Radev 2004) introducen tres nuevas medidas para centralidad: Grado de Centralidad, LexRank con Umbral y LexRank Continuo, inspiradas del concepto del prestigio en las redes sociales. LexRank es un algoritmo de ranking de propósito general basado en grafos para PLN y uno de los más aceptados para calcular la centralidad en un grafo. LexRank construye un grafo para el conjunto de documentos a resumir en el que existe un vértice por cada oración del mismo, para determinar los enlaces entre los vértices, las oraciones se representan por sus vectores de frecuencias (tf × idf), y se calcula la similitud léxica entre ellos utilizando la métrica del coseno, obteniendo así una matriz de similitudes. Aquellos pares de oraciones que presenten una similitud superior a un determinado umbral se enlazan entre sí en el grafo. Partiendo de la hipótesis de que las oraciones que son similares a muchas otras son las más importantes en relación al tema central del documento. La extracción de oraciones relevantes, consiste en identificar las oraciones que actúan como centroides en el grafo (Erkan and Radev 2004).

Los métodos basados en metaheurísticas, tratan la generación de resúmenes como un problema de optimización global, buscando seleccionar el mejor conjunto de oraciones para formar el resumen, obteniendo buenos resultados y confirmando que son un área de investigación importante. Las metaheurísticas se han utilizado de dos maneras principales: i) para generar el resumen automáticamente (Abuobieda et al. 2013b)(Aliguliyev 2009a)(Asgari et al. 2014)(Chatterjee et al. 2012)(García-Hernández and Ledeneva 2013)(Mendoza et al. 2014)(Qazvinian et al. 2008)(Shareghi and Hassanabadi 2008) y ii) para calcular los pesos de las características presentes en una ecuación que mide la relevancia de cada oración con respecto al documento original. En este último caso (Abbasi-ghalehtaki,

Khotanlou, and Esmaeilpour 2016)(Binwahlan et al. 2009)(Dehkordi and Kumarci 2009)(Litvak et al. 2010)(Uy et al. 2012), las soluciones candidatas representan la combinación de pesos de las características. El objetivo de estos trabajos consiste en encontrar una combinación adecuada que permita evaluar la relevancia de cada oración del documento y así incluir en el resumen a las que tengan mejor puntuación.

Metaheurísticas como algoritmos de búsqueda armónica se trataron en (Shareghi and Hassanabadi 2008) donde se plantea un algoritmo de búsqueda armónica binaria con una función objetivo que evalúa las oraciones basadas en aspectos como el grado de relación entre las oraciones consecutivas (legibilidad), la similitud entre las oraciones del resumen (cohesión) y la similitud de las oraciones con el título del documento (relación con el tema). Estos factores se ponderan y las frases con el mejor valor fitness forman el resumen (Mendoza et al. 2014).

Los algoritmos meméticos también han obtenido buenos resultados para generar resúmenes de un documento. Un algoritmo memético que combina un algoritmo genético y búsqueda local guiada (Mendoza et al. 2014) (Mendoza et al. 2014), en el cual cada agente se representa como el conjunto de frases que forman un resumen candidato, maximizando una función objetivo que evalúa cinco características para cada solución. El resumen se obtiene a partir del conjunto de frases que forman la solución con el valor fitness (aptitud) más alto (Mendoza, Cobos, and León 2015). Una desventaja de este enfoque, es que la configuración (diseño) de un algoritmo memético (o genético) es una tarea compleja, que requiere la selección de una variedad de esquemas para realizar operaciones de selección, cruce, mutación y reemplazo, además del afinamiento de los parámetros del algoritmo

En (Mendoza et al. 2015) se propone la optimización binaria memética basada en la metaheurística de la mejor búsqueda armónica global (GHS) y un algoritmo de búsqueda local codiciosa (Freddy) que añade frases al resumen candidato con la mayor probabilidad de mejorar el fitness de la solución y elimina aquellas frases con la mayor probabilidad de perjudicar su fitness (ESDS-GHS-GLO).

Trabajos como (Rasim M Alguliev et al. 2011) proponen un modelo que aborda la generación automática de resúmenes como un problema de programación lineal entera, basado en máxima cobertura y mínima redundancia. Dicho trabajo usa el algoritmo de Optimización por Enjambre de Partículas (Particle Swarm Optimization, PSO), para maximizar una función objetivo que es la combinación lineal de la similitud entre las oraciones del resumen y las oraciones de la colección de documentos y la redundancia entre las oraciones; con una restricción en el tamaño del resumen. En (Rasim M. Alguliev, Aliguliyev, and Mehdiyev 2011b) se propone un algoritmo de evolución diferencial con parámetros adaptativos para el cruce y la mutación, cuya función objetivo es la división entre la cobertura y la redundancia. En (Alguliev, Aliguliyev, and Isazade 2013a) abordan la generación automática de resúmenes como un problema de programación entera cuadrática, usando PSO y proponiendo dos funciones objetivo, una orientada a la diversidad y otra a la cobertura. En (Alguliyev et al. 2019) se propone un modelo de selección de frases en dos etapas basado en técnicas de clustering y optimización, denominado COSUM. En la primera etapa, se encuentran los temas de un texto, las frases se agrupan utilizando el método k-means. En la segunda etapa, para la selección de las frases más destacadas de los clústeres, se propone un modelo de optimización. Este modelo optimiza una función objetivo que se expresa como una media armónica de las funciones objetivo cubriendo las frases y la diversidad de las frases seleccionadas en el resumen.

En la presente investigación, el trabajo se enfocó en la generación automática de resúmenes extractivos genéricos de un documento con dos métodos del estado del arte que utilizan el modelo espacio vectorial para representar las características de un texto: LexRank y LexRank Continuo (grafos), y, FSP y ESDS-GHS-GLO (metaheurística con optimización binaria). El objetivo principal que se busca es evaluar si al introducir la información lingüística que tienen los N-gramas sintácticos en estos métodos se logran mejoras en la calidad de los resúmenes generados.

3.3 Métricas ROGUE para evaluación de resúmenes

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) es un conjunto de métricas para medir la cobertura léxica (similitud basada en N-gramas) y determinar automáticamente la calidad de un resumen candidato (generado por un sistema) comparado con otro u otros resúmenes de referencia (calificados como Gold Standard por ser generados por humanos expertos) (Lin 2004). ROUGE incluye varios métodos de evaluación automática que miden la similitud entre resúmenes (Lin and Rey 2004).

ROUGE-N (N-gram Co-Occurrence Statistics)

Contabiliza el número de N-gramas que coinciden entre un resumen candidato y uno o varios resúmenes de referencia (traducción libre). Se calcula mediante la **Ecuación 2**.

$$ROUGE - N = \frac{\sum_{S \in \{ ResúmenesReferencia\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ ResúmenesReferencia\}} \sum_{gram_n \in S} Count(gram_n)}$$

***Ecuación 2**. Cálculo de ROUGE-N*

Donde *n* representa la longitud del n-grama, $gram_n$, y $Count_{match}(gram_n)$ es el número máximo de N-gramas co-ocurrentes en un resumen candidato y un conjunto de resúmenes de referencia. ROUGE-N es una medida relacionada con la repetición porque el denominador de la ecuación es la suma total del número de N-gramas que aparecen en el resumen de referencia. Nótese que el número de N-gramas en el denominador de la fórmula ROUGE-N aumenta a medida que se añade más referencias. Esto es intuitivo y razonable porque podrían existir múltiples y buenos resúmenes. Cada vez que se añade una referencia al conjunto, se amplía el espacio de resúmenes alternativos. Al controlar qué tipos de referencias se añaden al conjunto de referencias, se puede diseñar evaluaciones que se enfocan en diferentes aspectos de la integración. Se debe tener en cuenta también que el numerador suma todos los resúmenes de referencia. Efectivamente, esto da más peso a los N-gramas coincidentes que aparecen en múltiples referencias. Por lo

tanto, un resumen candidato que contenga palabras compartidas por más referencias se ve favorecido por la medida ROUGE-N. Esto es muy apropiado ya que normalmente es preferible un resumen candidato que sea más similar al consenso entre los resúmenes de referencia (Lin 2004).

Hasta ahora, se ha mostrado cómo calcular ROUGE-N usando una sola referencia. Cuando se utilizan múltiples referencias, se calcula ROUGE-N por parejas nivel-resumen entre un resumen candidato s y cada referencia, r_i, en el conjunto de referencias. Luego selecciona el máximo de puntuaciones ROUGE-N de parejas nivel-resumen como la puntuación ROUGE-N final de referencia múltiple. Esto se puede escribir de la siguiente manera (traducción libre) **Ecuación 3** (Lin 2004).

$$ROUGE - N_{multi} = argmax_i\, ROUGE - N(r_i, s)$$

***Ecuación 3**. Múltiples Referencias*

ROUGE-S (Skip-Bigram Co-Occurrence Statistics)

Skip-bigram es cualquier par de palabras en su orden en la oración, lo que permite separaciones arbitrarias. Las estadísticas de coincidencia de Skip-bigram miden la superposición Skip-bigram entre una traducción candidata y un conjunto de traducciones de referencia (traducción libre) (Lin 2004). Por ejemplo, si se consideran las siguientes oraciones:

S1. police killed the gunman
S2. police kill the gunman
S3. the gunman kill police
S4. the gunman police killed

Cada oración tiene $C(4,2)^1$ = 6 combinaciones de Skip-bigrams, por ejemplo, S1 tiene los siguientes Skip-bigrams: *("police killed", "police the", "police gunman", "killed the", "killed gunman", "the gunman"*). S2 tiene tres coincidencias Skip-bigram con S1: (*"police the", "police gunman", "the gunman"*), S3 tiene un Skip-bigram coincidente con S1 (*"the gunman"*), y S4 tiene dos coincidencias Skip-bigram con S1 (*"police killed", "the gunman"*). Dadas las traducciones *X* de longitud *m* e *Y* de la

longitud *n*, suponiendo que *X* es una traducción referencia e *Y* es una traducción candidata, se calcula la medida *F* basada en Skip-bigram según la **Ecuación 4**.

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

***Ecuación 4**. Importancia relativa de los Skip-gram de la traducción referencia X*

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

***Ecuación 5**. Importancia relativa de los Skip-gram de la traducción candidata Y*

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

***Ecuación 6**. Medida F basado en Skip-bigram o ROUGE-S*

ROUGE-S es la medida F basada en Skip-bigram donde $SKIP2(X,Y)$ es el número de coincidencias Skip-bigram entre *X* e *Y*, β controla la importancia relativa de P_{skip2} y R_{skip2}, y *C* es la función combinación (**Ecuación 6**).

Usando la **Ecuación 6** con $\beta = 1$ y S1 como referencia, el puntaje ROUGE-S de S2 es 0.5, S3 es 0.167, y S4 es 0.333. Por lo tanto, S2 es mejor que S3 y S4, y S4 es mejor que S3. Aplicando el Skip-bigram sin ninguna restricción sobre la distancia entre las palabras, coincidencias falsas como "*the*" o "*of in*" podrían contarse como coincidencias válidas. Para reducir estas coincidencias falsas, podría limitarse la distancia máxima de salto, d_{skip}, entre dos palabras en orden permitidas para formar un Skip-bigram. Por ejemplo, si se asigna 0 a d_{skip} entonces ROUGE-S equivale a bigram y traslapa la medida F. Si se asigna 4 a d_{skip} entonces sólo pares de palabras de un máximo de 4 palabras aparte pueden formar los Skip-bigram.

ROUGE-SU (Extensión de ROUGE-S)

Un problema potencial para ROUGE-S es que no da crédito a una frase candidata si la frase no tiene cualquier par de palabras coincidiendo con las referencias. Por ejemplo, la siguiente oración tiene un puntaje ROUGE-S de cero:

S1. police killed the gunman

S5. gunman the killed police

S5 es el reverso exacto de S1 y no hay ninguna coincidencia de Skip-bigram entre ellos. Sin embargo, es posible diferenciar oraciones similares a S5 de las oraciones que no tienen coincidencia de una sola palabra con S1. Para lograr esto, se extiende ROUGE-S con la adición de unigrama como unidad de conteo. La versión extendida se llama ROUGE-SU. También se puede obtener ROUGE-SU de ROUGE-S añadiendo un marcador de inicio de frase al principio de las frases candidatas y referencia. Esta medida elimina ese problema (traducción libre) (Lin 2004).

Otras métricas ROUGE

ROUGE cuenta con otras métricas para medir la similitud entre resúmenes candidatos y de referencia, ellas son:

- **ROUGE-L (*Longest Common Subsequence*)**. Utiliza la longitud de las secuencias de caracteres más largas que coinciden en el resumen candidato y un conjunto de resúmenes de referencia.
- **ROUGE-W (Weighted Longest Common Subsecuence)**. Mide tanto la longitud de la secuencia de caracteres como la ausencia de diferenciación de los caracteres entre un resumen candidato y un conjunto de resúmenes de referencia.
- **ROUGE-S (Skip-Bigram Co-Occurrence Statistics)**. Tiene en cuenta las secuencias que pueden aparecer en el texto generado por el sistema y que presentan un máximo número de términos en ellos, estas secuencias son llamadas bigramas.

En el estado del arte, las medidas más utilizadas para valorar los resúmenes generados automáticamente para un documento representados por bolsa de palabras o N-gramas, son ROUGE-N (con N igual a 1 y 2) y ROUGE-SU.

3.4 N-gramas sintácticos

En la literatura de recuperación de información se encuentran varios modelos de dependencia de términos (N-gramas) y algunos de ellos han producido resultados interesantes. Van Rijsbergen (Rijsbergen 1979) presentó uno de los modelos de dependencia más citados, el cual es una extensión del enfoque de clasificación Bayesiano para la recuperación de información. Gao et al. (2004) presentaron un nuevo enfoque de modelado del lenguaje de dependencia para la recuperación de información y logró mejoras significativas y sustanciales para dependencias secuenciales (N-gramas) (Gao et al. 2004). Croft et al. (2014) demostraron que las frases y la proximidad de términos pueden potencialmente mejorar la efectividad de la tarea de recuperación de información modelándolos como dependencias en el modelo de red de inferencia (Huston and Croft 2014). Investigaciones recientes con colecciones de pruebas más grandes han demostrado que la información de proximidad de términos es una característica útil en la tarea de recuperación de información (Zhao, Huang, and Ye 2014).

A comienzos del 2013 se propuso un nuevo concepto de construcción de N-gramas, los *N-gramas sintácticos* o *SN-gramas*, denominados así porque se obtienen siguiendo relaciones en árboles sintácticos. Los N-gramas sintácticos se utilizaron en la tarea de atribución de autoría y se compararon con los N-gramas tradicionales de tamaño *n* igual a 2, 3, 4 y 5, y, N-gramas con enfoques puramente estadísticos. Los resultados obtenidos fueron superiores cuando se aplicaron N-gramas sintácticos para representar las características en el proceso de clasificación. Se concluyó que los N-gramas sintácticos mantienen la propiedad de ser N-gramas con el valor agregado de la información lingüística y sintáctica (relaciones sintácticas entre palabras) a diferencia de los N-gramas tradicionales que se obtienen según la secuencia de aparición de los elementos textuales en el documento, por lo tanto, se pueden usar en tareas del PLN donde se utilizan N-gramas tradicionales. La investigación determinó además que el análisis sintáctico previo es necesario, pero dos aspectos deben tenerse en cuenta: primero los analizadores requieren tiempo

y no todos los lenguajes tienen analizadores, segundo no es claro si es suficiente el análisis superficial o es necesario el análisis completo para obtener N-gramas sintácticos de calidad. Además, se identificó que las etiquetas de relaciones sintácticas o *SR tags* también pueden ser usadas para construir los N-gramas sintácticos por su composición y se obtienen durante el procesamiento lingüístico previo (Sidorov et al. 2013b) (Sidorov et al. 2013a).

Al finalizar el 2013 se investigó sobre N-gramas sintácticos no-continuos para presentarlos como posibles características en un modelo de espacio vectorial que conserve la estructura multidimensional. Comparan la construcción de N-gramas no-continuos con N-gramas continuos (comúnmente usados en investigaciones previas) para el lenguaje inglés y español. La conclusión es que se necesitan más estudios para determinar qué parámetros en la construcción de N-gramas sintácticos no-continuos son mejores y para qué tipo de tareas existentes dentro de la lingüística computacional pueden obtener mejores resultados (Sidorov 2013b).

3.5 Framework del Grupo GTI

El Grupo de I+D en Tecnologías de la Información (GTI) de la Universidad del Cauca cuenta con un Framework para la generación automática de resúmenes extractivos de un documento que implementa entre otros, los siguientes algoritmos LexRank, LexRank Continuo y la Mejor Búsqueda Armónica Global (Global-Best Harmony Search, ESDS-GHS-GLO), todos ellos utilizan desde un documento HTML o texto hasta un archivo que representa un documento con bolsa de palabras, lo transforman en una matriz espacio vectorial de frases (filas) por términos (columnas) y ponderan los términos con Okapi BM25.

A continuación, se introducen los algoritmos del Framework que fueron usados para la experimentación en la presente tesis:

- **LexRank**: Basado en grafos, similar a Page Rank (versión inicialmente usada por Google para ranquear las páginas web cuando un usuario hace una consulta). El algoritmo ranquea los nodos de un grafo (frases, sentencias u

oraciones) en términos de su centralidad, i.e. los nodos más conectados con los demás nodos del grafo (Erkan and Radev 2004).

- **LexRank Continuo**: Corresponde a una variación de LexRank que evita perder información en la construcción de los pesos en las conexiones del grafo que representa al documento. Es propuesto por los mismos autores de LexRank (Erkan and Radev 2004).
- **ESDS-GHS-GLO**: Inspirado en la mejor búsqueda armónica global, una metaheurística de optimización basada en la forma como los cantantes de Jazz improvisan su música. El algoritmo incluye un optimizador local voraz y optimiza una función objetivo que es la combinación lineal de la posición de la frase en el documento, la longitud de la frase y el cubrimiento de las frases candidatas seleccionadas para generar el resumen. La representación del documento completo se puede hacer como un vector de los N-gramas más importantes o el centroide de todos los N-gramas presentes en el documento (este último fue usado en la experimentación). Este algoritmo fue comparado con diversos algoritmos del estado del arte, entre ellos MA-SingleDocSum, DE, FEOM, UnifiedRank, Net-Sum, QCS, CRF, SVM y Manifold Ranking con resultados muy competitivos (Mendoza et al. 2015).

4 Modelo de representación con N-gramas Sintácticos

4.1 Proceso general

Se seleccionaron las colecciones de noticias DUC2001 y DUC2002, comúnmente utilizadas por la comunidad científica y académica como dataset de evaluación en la generación automática de resúmenes para un documento. La **Figura 8** resume el proceso, el cual inicia con el análisis sintáctico, realizado con el analizador Freeling que entrega un archivo XML con el análisis por dependencias y la estructura de rutas de los subárboles en forma de nodos anidados. Los archivos XML se procesan con una herramienta software creada por el autor de esta tesis, para crear los N-gramas sintácticos continuos y no continuos, luego, se determina el grado de relevancia de los N-gramas generados en relación con los documentos usando Okapi BM25 y posteriormente se almacenan en una matriz de términos por documento (archivo de texto). Las matrices se cargan con el Framework del Grupo GTI y se procesan con los 4 algoritmos seleccionados (LexRank, LexRank Continuo, FSP y ESDS-GHS-GLO) para generar automáticamente los resúmenes de cada documento de los datasets, después se evalúan los resultados con las métricas ROUGE-1 y ROUGE-2. El proceso final, consiste en realizar la comparación de los resultados obtenidos con los alcanzados por los algoritmos del estado del arte.

4.2 Preprocesamiento y Análisis Sintáctico

La herramienta software cuenta con una interfaz de usuario que carga el dataset de entrenamiento para dos tipos de formatos: i) archivos XML con análisis sintáctico de Freeling, o, ii) archivos texto con etiquetas HTML. El trabajo utilizó las colecciones de noticias DUC2001 y DUC2002 como datasets de evaluación y se realizaron cargas con los dos formatos mencionados. Si el archivo es un texto con etiquetas HTML, la herramienta software crea la carpeta *analisys* con la misma estructura de carpetas del dataset (subcarpetas), luego efectúa el proceso de alistamiento por

cada archivo (extrae el contenido del archivo, convierte etiquetas y limpia caracteres no necesarios), genera el Shell para procesar cada archivo con Freeling y como salida obtiene el análisis sintáctico de cada noticia y lo almacena en un archivo en formato XML con las dependencias y arboles sintácticos. Si el archivo es XML con su análisis sintáctico, continúa con la creación de los N-gramas sintácticos.

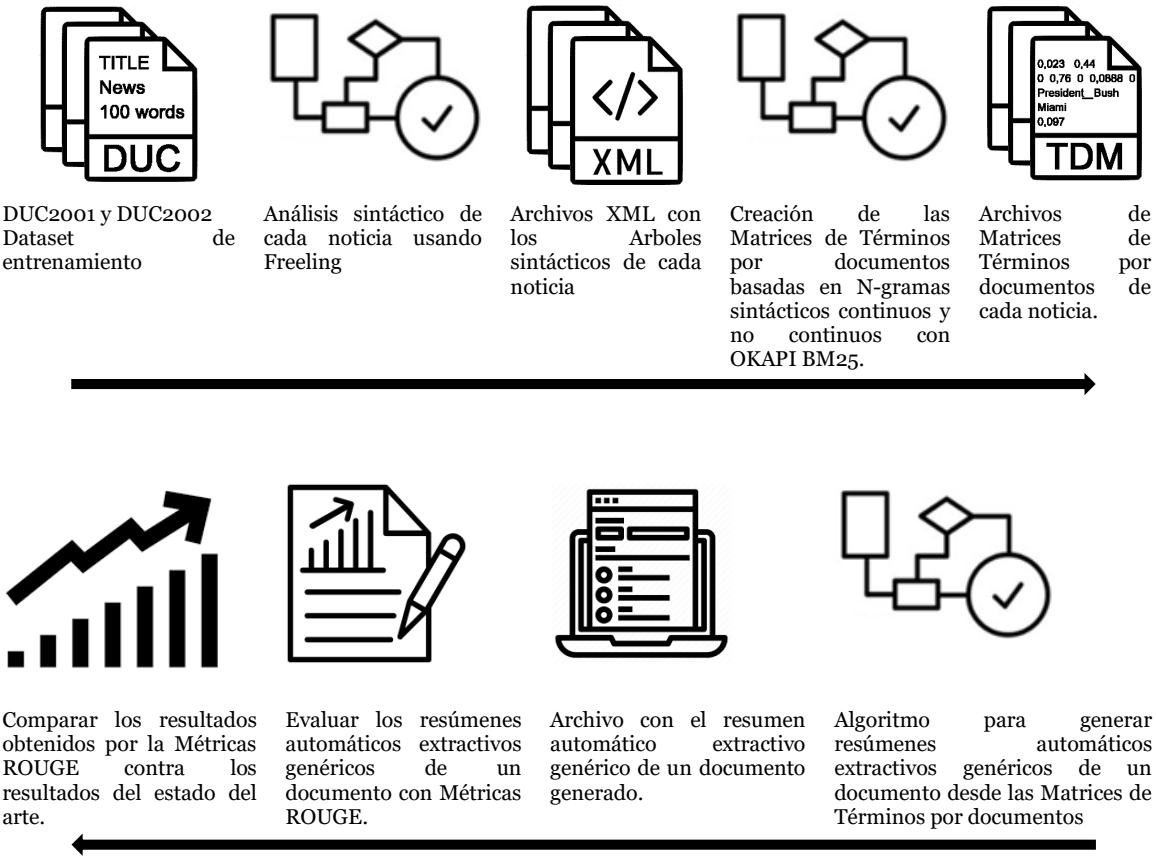


***Figura 8**. Diagrama del proceso para crear matrices TDM basada en N-gramas sintácticos*

El preprocesamiento inicia con la carga del dataset, la herramienta software valida si es uno de los dos formatos permitidos para carga: archivo texto con etiquetas HTML (noticia de la colección DUC) o archivo con análisis sintáctico de Freeling en formato XML. Si es un archivo texto con etiquetas HTML, se verifica que exista la etiqueta <TEXT>, se extrae el contenido, se remplazan los caracteres especiales o etiquetas HTML, se guarda el texto en un archivo para realizar el análisis sintáctico con Freeling y se obtiene el análisis sintáctico en formato XML por cada archivo. Si el archivo cargado es un análisis sintáctico en formato XML, se omite el proceso anterior y se valida que al menos uno de sus descendientes sea un párrafo con

análisis morfosintáctico y su respectivo árbol de dependencias. Una vez cargados los archivos como lo describe el **Algoritmo 1**, se puede parametrizar la herramienta para generar las matrices de términos basada en N-gramas sintácticos continuos y no continuos.

Inputs:	Colección de documentos *CDocs*
Condiciones previas:	Colección de archivos (dataset) tipo texto con etiquetas HTML o XML con análisis sintáctico
Salida:	Colección de archivos con análisis sintáctico *CDocsStc*

```
1. begin
2.     for each Doc in CDocs do // Selecciona cada archivo de la ruta origen
3.         Crear la estructura de carpetas para el folder analisys y matrices
4.         Archivo = Leer(Doc) // Cargar el archivo y lee el contenido
5.         if IsText(Archivo) y Archivo.atributo == 'TEXT' then
6.             Txt = Replace(caracteres especiales y etiquetas HTML)
7.         ArchFix = CrearArchivo(Txt) // Crear el archivo en la respectiva subcarpeta
8.     end for each
9.     for each ArchFix in analisys do // Selecciona cada archivo de análisis sintáctico
10.        CDocsStc = Add(Freeling(ArchFix))
11.    end for each
12.    return
13. end
```

***Algoritmo 1**. Cargar la colección de documentos*

4.3 Herramienta para crear matrices de N-gramas sintácticos continuos y no continuos

Antes de crear los N-gramas sintácticos y las matrices de términos por documento, se debe realizar la tarea previamente explicada del análisis sintáctico con Freeling. La herramienta software cuenta con una variedad de opciones y parámetros implementados para crear diferentes matrices de términos basadas en N-gramas sintácticos continuos y no continuos (ver **Tabla** 1)

Seleccionados los parámetros y el archivo o carpeta contenedora de archivos de análisis sintácticos en formato XML (generada en la fase previa), se procede a generar las matrices de términos por documento (ver **Figura 9** y **Algoritmo 2**).

***Tabla** 1. Parámetros para crear Matrices de Términos por Documento*

Item	Parámetro
1	Idioma del documento
2	Inclusión o no de “stop words”
3	Análisis por lema o palabra
4	Tamaño del N-grama (opciones del 1 a 7 gramas)
5	Tipo de representación (vector o centroide)
6	Tipo de N-grama sintáctico (léxicos, categorías gramaticales y relaciones sintácticas)
7	Tipos de sintagmas a incluir en la representación (nominal, adjetival, verbal, preposicional, adverbial o la combinación)
8	Ordenar los N-gramas sintácticos a incluir en la representación.

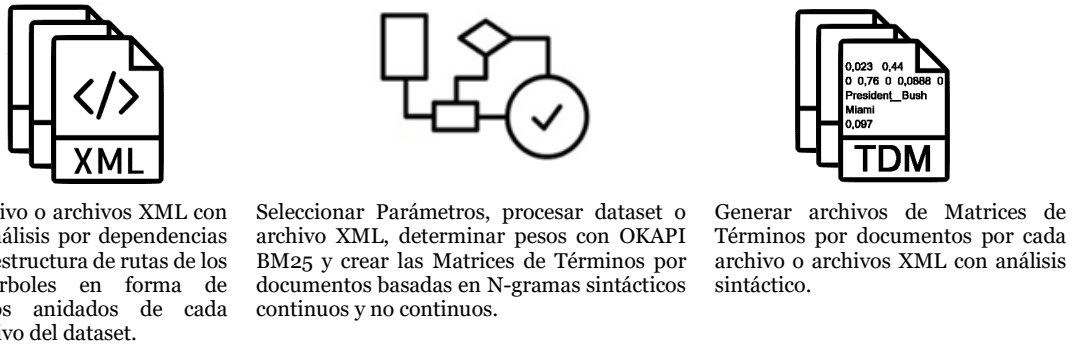


***Figura 9**. Diagrama del algoritmo para crear matrices TDM de N-gramas sintácticos*

En el **Algoritmo 2** se carga cada análisis sintáctico de la colección de documentos analizados y se crea el archivo de texto por noticia con: los N-gramas sintácticos, la lista de frases del documento, la matriz de términos del documento, el diccionario de términos del documento, el diccionario de frecuencias de los N-gramas del documento y el diccionario de índices con los N-gramas del documento. Si se parametrizó ordenar los términos del N-grama, se efectúa dicha tarea.

Cargada la colección con el análisis sintáctico, se procede a crear los N-gramas por cada documento (ver **Algoritmo 3**). Se tiene en cuenta los ocho parámetros configurados en la herramienta software (ver **Tabla** 1). El proceso inicia creando la lista de etiquetas válidas según los sintagmas seleccionados, recorre el documento a nivel de cada párrafo, oración y término. Determina si es una palabra vacía y si se incluye en el N-grama sintáctico (determinado por el parámetro de la herramienta). Valida si el término está en la lista de sintagmas y determina si selecciona el lema

o el término original. Calcula la frecuencia y agrega el término a la lista de N-gramas sintácticos (ver **Algoritmo 3**). Verifica si el tamaño de la lista es igual al tamaño seleccionado en la herramienta. Si tiene el tamaño, verifica si debe ordenar los términos del N-grama sintáctico. Finalmente, asigna los N-gramas sintácticos al diccionario de todos los términos, al diccionario de las frecuencias de los N-gramas sintácticos y al diccionario de los N-gramas sintácticos con su respectivo índice.

Inputs:	Colección de documentos con análisis sintáctico *CDocsStc*, Tamaño del N-grama *n*, Tipo de N-grama *tng*, Tipo de sintagma *tsg*, Idioma *i*, Lista de tipos de sintagmas a incluir *lsg*, Indicador de Stop words *stw*, Indicador para ordenar elementos textuales del N-grama *idx*, Indicador del elemento fuente (lema o palabra original) *src*, y Indicador del tipo de representación (centroide o vector) *rps*
Condiciones previas: Realizar el análisis sintáctico del dataset	
Salida:	Matriz de todos los N-gramas *MNg*, Lista de todas las frases *Lph*, Matriz de Términos por documento *TDMxd*, Matriz de los Diccionarios de todos los términos por documento *MDtxd*, Diccionario de todos los N-gramas con frecuencia por documento *DNgxd*, y Diccionario de todos los N-gramas con índice por documento *DIdxd*
1. **begin**	
2. **for each** row *DocsStc*$_i$ **in** *CDocsStc* **do** // Recorre colección de noticias analizadas	
3. CrearNgramas*(DocsStc, n, lsg, rps)* recibe *Lphxd, TDMxd, Dtxd, DNgxd, DIdxd*	
4. **if** *idx* **then** Ordenar los términos del *TDMxd*	
5. *MNg* = Add(*TDMxd*) // Anexa los N-gramas por documento a la matriz global	
6. *Dtxd* = Add(*TDMxd*) // Anexa los N-gramas por documento (frecuencias e índices)	
7. *Lph* = Add(*Lphxd*)	
8. CrearMatrizTDM(*Lph*, *DIdxd*) recibe MTFIDF, VFrcNgxd, VCtdrPhxd, VIDFxd, VTFIDFCDoc, VAvgPh	
9. CrearRepresentaciónTDM(*MNg, TDMxd, Lph, MDtxd*)	
10. **end for each**	
11. **return**	
12. **end**	

***Algoritmo 2**. Recorrer la colección para crear las Matrices de Términos por Documento (TDM por archivo con análisis sintáctico)*

4.4 Crear N-gramas sintácticos

El **Algoritmo 3** se encarga de crear cada N-grama sintáctico continuo y no continuo teniendo en cuenta los parámetros seleccionados en la herramienta software (ver **Tabla** 1). Se recorre cada párrafo de la lista de párrafos y por cada párrafo recorre las oraciones que lo componen. A cada oración, le evalúa término por término según los parámetros seleccionados. Si un término cumple las condiciones, se anexa a la lista de gramas hasta completar la cantidad *N*. Una vez creado el N-grama, se anexa a la matriz de N-gramas.

Inicialmente se crea la lista de etiquetas válidas según los tipos de sintagmas seleccionados en los parámetros. A continuación, se recorre cada párrafo de la lista de párrafos, por cada párrafo recorre las oraciones que lo componen y por cada oración evalúa término por término. Se evalúa si es un stop word y se define si se incluye o no según indicador, se evalúa si el término es un elemento válido en la lista de sintagmas, determina si tiene en cuenta el lemma o el término original según indicador, calcula la frecuencia del término en la oración y anexa la grama a la lista hasta completar la cantidad *N* de gramas. Ordena los gramas en el N-grama según el indicador. Una vez creado el N-grama, se anexa a la matriz de N-gramas y al diccionario de elementos únicos por documento. La salida del proceso por archivo analizado es la lista de frases analizadas, la Matriz de N-gramas, Diccionarios de: todos los términos, términos únicos, N-gramas con frecuencia e índice.

Inputs: Lista de Párrafos del análisis sintáctico de un documento *lphg,* Idioma *i*,
Tamaño del N-grama *n*, Tipo de N-grama *tng*, Tipo de sintagma *tsg*,
Lista de tipos de sintagmas a incluir *lsg*,
Indicador de Stop words *stw*,
Indicador para ordenar elementos textuales *idx*,
Indicador del elemento fuente (lema o palabra original) *src*, y
Indicador del tipo de representación (centroide o vector) *rps*

Condiciones previas: Realizar el análisis sintáctico del dataset
Parametrizar las ocho variables de la herramienta software

Salida: Lista de todas las frases por documento *Lphxd*,
Matriz de N-gramas por documento *MNgxd*,
Diccionario de todos los términos por documento *Dtxd*,
Diccionario de todos los términos únicos por documento *DUtxd*,
Diccionario de todos los N-gramas con frecuencia por documento *DNgxd*, y
Diccionario de todos los N-gramas con índice por documento *DIdxd*

```
1. begin
2.    for each row lsg in lsg do
3.        lesg = Add(etiqueta) // Crea la lista de etiquetas a tener en cuenta
4.    end for each
5.    for each row Phg, in lphg // Evalúa cada párrafo de lista de párrafos del documento
6.        for each row Stc, in Phg // Evalúa cada oración del párrafo
7.            ctdr_wrd = 0
8.            for each row Wrd, in Stc // Evalúa cada oración del párrafo
9.                if IsStopword(Wrd.atributo) == stw then // Incluye o no palabras vacías
10.               if Wrd.atributo in lsg then // Incluye término según lista de sintagmas
11.               if src == 'lemma' then // Si el término debe ser lema
12.                   Ng = Wrd.lemma
13.               else // Selecciona la palabra original
14.                   Ng = Wrd.word
15.               Ng = calcularFrecuencia
16.               NgSn = Add(Ng)
17.               ctdr_wrd++
18.               if ctdr_wrd >= n then
19.                   ctdr_wrd = 0
20.                   if idx == 'S' then Dtxd = Add(Ng)
21.               NgSn = Ordenar(NgSn)
22.               Dtxd = Add(Ng)
23.               DNgxd = Add(NgSn(frecuencia), NgSn(indice))
24.               DNgxd = Add(NgSn(indice), NgSn(término))
25.           end for each
26.           lphxd = Add(Stc)
27.       end for each
28.   end for each
29.   return Lphxd, TDMxd, Dtxd, DNgxd, Dldxd
30. end
```

Algoritmo 3*. Crear N-gramas sintácticos*

4.5 Crear la Matriz de Términos por Documento (TDM)

Antes de crear la Matriz de Términos por Documento, se debe realizar el **Algoritmo 3** porque las salidas son necesarias para este algoritmo. El **Algoritmo 4** Crea la matriz TF-IDF (TF=frecuencia de ocurrencia del término e IDF=Frecuencia inversa del término, en la colección de documentos), en la que se expresa cuán relevante (peso) es cada término para un documento en una colección. Para esto calcula la importancia del N-grama en la frase (TF), la importancia del N-grama en el dataset (IDF), la cantidad de N-gramas procesados por frase y el número de archivos analizados en los que aparece cada N-grama en el dataset. Se tiene en cuenta los parámetros seleccionados en la herramienta software (ver **Algoritmo 2** y **Tabla** 1).

Inicialmente calcula el tamaño promedio de una frase (cantidad promedio de N-gramas procesados en una frase). Se recorre cada frase de la lista de frases, por cada frase recorre el diccionario de N-gramas únicos procesados en la frase y verifica si el diccionario contiene el N-grama de la frase, en caso contrario se debe eliminar el N-grama de la frase. A continuación, se determina el grado de relevancia de los N-gramas en relación con los archivos analizados usando Okapi BM25. Finalmente, según el tipo de representación (centroide o vector), se realiza los cálculos de la frecuencia acumulada por cada N-grama y se almacena en un arreglo.

La salida del proceso por archivo analizado es la Matriz de Frecuencias, los vectores de: conteo de frases por documento, frecuencia inversa de cada N-grama, frecuencia de la importancia del N-grama en la frase, frecuencias del archivo analizado en el dataset y promedios de las frases.

La Matriz es almacenada en un archivo tipo texto el cual es utilizado por el Framework del Grupo GTI para generar los resúmenes automáticos.

Inputs: Lista de todas las frases *Lphxd*, y Diccionario de todos los N-gramas con índice por documento *DIdxd*
Condiciones previas: Crear N-gramas sintácticos, Lista de Frases por documento y Diccionarios de Indice y Frecuencias de los N-gramas
Salida: Matriz de Frecuencias *MTFIDF,* Vector de Frecuencias del N-grama sintáctico por documento *VFrcNgxd*, Vector del contador de frases por documento *VCtdrPhxd*, Vector de la frecuencia inversa del N-grama sintáctico por documento *VIDFxd*, Vector de Frecuencias del documento en la colección *VTFIDFCDoc*, Vector de Promedios de las frases *VAvgPh*

```
1. begin
2.    AvgSxPh = Tamaño Promedio Frases / Total de Frases del documento
3.    VCtdrPhrxNg[] // Inicializa el vector del conteo de frases con el N-grama
4.    LTermRmv[] // Lista de Términos a remover de la frase
5.    for each row Phr in Lph do // Selecciona cada frase de la lista de frases
6.        for each row Term in Phr do
7.            if Term in DIdxd then
8.                TermPos = DIdxd.Pos // Posición del término en la frase
9.                MTFIDF[Phr.Pos][TermPos] = Term.tfidf
10.               VFrcNgxd = Term.tfidf
11.               VCtdrPhrxNg[TermPos]++ // Conteo de la aparición del término
12.           else
13.               LTermRmv[] = Add(Term)
14.       end for
15.       for each row Term in LTermRmv do // Por cada término a remover
16.           Phr = Remover(Term)
17.       end for
18.   end for
19.   for each row Phr in Lph do // Selecciona cada frase de la lista de frases
20.       for each row Term in Phr do
21.           TtlPhrInNgr = VCtdrPhrxNg[Term.Pos] // Total de frases con el N-grama
22.           VIDFxd[Term.Pos] = Log(Lph.count() / TtlPhrInNgr) // Okapi BM25
23.       end for
24.   end for
25.   return
26. end
```

***Algoritmo 4**. Crear Matriz TDM por documento*

5 Experimentación y resultados

En este capítulo inicialmente se describe la línea base que se definió para obtener la información sintáctica y semántica necesaria para crear N-gramas sintácticos. Posteriormente se describe los resultados obtenidos al introducir la información lingüística de los N-gramas sintácticos (características), en el modelo de representación espacio vectorial utilizado por los 4 métodos reportados en el estado del arte: LexRank (grafos), LexRank Continuo (grafos), FSP (metaherística) y ESDS-GHS-GLO (metaheurística con optimización binaria).

5.1 Recursos para la experimentación

5.1.1 Datasets: DUC2001 y DUC2002

Se seleccionaron las colecciones de noticias DUC2001 y DUC2002, comúnmente utilizadas por la comunidad científica y académica como datasets de evaluación en la generación automática de resúmenes para un documento. Dichas colecciones son producto de investigaciones del National Institute of Standards and Technology (NIST) y están disponibles en línea en http://www-nlpir.nist.gov. Además, cuentan con los resúmenes de referencia hechos por humanos y aprobados por la Conferencia de Entendimiento de los Documentos (DUC), necesarios para determinar automáticamente la calidad de los resúmenes candidatos. La colección DUC2001 contiene 30 conjuntos de aproximadamente 10 noticias en inglés, que en total son 309 documentos con diversos temas, y, DUC2002, por su parte, consta de 567 documentos en 59 conjuntos. En estas colecciones los resúmenes generados deben tener máximo 100 palabras.

5.1.2 El Analizador sintáctico

El análisis sintáctico es una tarea indispensable para crear N-gramas sintácticos porque determina las relaciones de concordancia y jerarquía al agrupar las palabras según el lenguaje del texto y su orden en la frase. La comunidad científica y académica utiliza analizadores sintácticos denominados parser (término en inglés que hace referencia a un programa que analiza la estructura lógica de una frase).

El parser se encarga de la descomposición y transformación en componentes individuales a partir del análisis automático de las relaciones entre las palabras que conforman una frase, determinando el tipo de relación de dependencia, el tipo de sintagma (palabra núcleo y cuáles son las palabras dependientes), la categoría gramatical, el lema, el género, el etiquetado de las relaciones, la posición en la frase, entre otras características. Los analizadores sintácticos comúnmente utilizados por la comunidad científica y académica para crear N-gramas sintácticos son: FreeLing (Carreras et al. 2004) y Natural Language Toolkit (NLTK). FreeLing es una librería de código abierto para el procesamiento multilenguaje automático con una amplia variedad de utilidades para análisis lingüístico. NLTK es una biblioteca de Procesamiento de Lenguaje Natural que utiliza el lenguaje de programación Python.

Se evaluaron los analizadores Freeling y NLTK; instalándolos en los sistemas operativos Windows (versiones 7 Pro y 10 Pro) y, Linux (distribuciones Mint 17.3 y Ubuntu 16.04 LTS Xenial) para identificar las ventajas y desventajas de cada uno de ellos como se describe en la **Tabla 2**. Se seleccionó Freeling por sus prestaciones y niveles de análisis: morfológico, etiquetado, tokenizado, dividido, parseado, constituyentes y dependencias, este último el más adecuado en información y estructura para el trabajo, especialmente la salida en formato XML por su distribución al encadenar las rutas de los subárboles en forma de nodos anidados. Se encontraron las siguientes limitantes con Freeling: 1) problemas al tratar de integrar las librerías de análisis a la herramienta desarrollada en Visual Studio de Microsoft (el analizador no ofrece el mismo nivel de soporte que en Linux). Esto se solucionó mediante el uso de Freeling desde línea de comando (instrucción

analyzer), y, 2) perdida de información de análisis dependiendo de las librerías de C# que se utilizaron en Visual Studio de Microsoft para leer el archivo XML del análisis. Para solucionar esto, se leyó el archivo XML como un texto codificado según el lenguaje y luego se convirtió en una cadena texto que se transformó nuevamente en un árbol XML (usando el método *Parse* de la librería *XDocument*). La lectura del archivo se realizó con patrones de consulta de LINQ.

***Tabla 2**. Ventajas y Desventajas de los analizadores FreeLing y NLTK*

Analizador	Ventajas	Desventajas
FreeLing	• Reconoce 13 lenguajes • Creado con plantillas estándar STL y lenguaje de programación C++, por lo tanto, es portable y adaptable a los sistemas operativos: Linux, Unix, Windows y MacOSX • Cuenta con un demo en Internet • Cuenta con un graficador de árboles sintácticos (análisis basado en gramática de dependencias) con cierto grado de complejidad y varias opciones de análisis como: detección de entidad con nombre, detección de múltiples palabras, codificación fonética, entre otras • Realiza varios procesos de análisis en múltiples hilos en uno o varios procesadores. • El resultado de los análisis puede ser visualizado en diferentes formatos (gráfico, XML, texto, JSON, entre otros) • Trabaja en sistemas de 32 bits y 64 bits • El mayor soporte y respaldo se obtiene para distribuciones de Linux/Unix	• No reconoce algunos acentos en las variantes de un mismo lenguaje, por ejemplo, español España y español México • Requiere complementos y librerías propias para cada sistema operativo dependiendo de su versión, por ejemplo, para Windows requiere Cygwin, MinGW y MSVS • El Demo tiene limitantes en la cantidad de palabras a analizar • Requiere recursos hardware de alto rendimiento para un procesamiento rápido • La versión compilada para Windows no es totalmente funcional • El soporte para Windows es limitado y requiere como IDE a Visual Studio 2015 o superior para recompilar el código fuente
NLTK	• Reconoce el lenguaje inglés y español • Portable y adaptable a los sistemas operativos Linux, Unix, Windows y MacOSX. • Cuenta con un graficador de árboles sintácticos (análisis basado en gramática de dependencias)	• Ciertas funcionalidades del lenguaje español aún están en desarrollo • Requiere librerías y complementos propios de cada sistema operativo (dependiendo de la versión del sistema) • No cuenta con un Demo en internet • Recomienda evitar las plataformas de 64 bits

	• Requiere conocimientos básicos de programación en Python

5.1.3 Algoritmos Seleccionados

El Grupo de I+D en Tecnologías de la Información (GTI) de la Universidad del Cauca cuenta con un Framework para la generación automática de resúmenes extractivos de un documento que implementó varios algoritmos reportados en la literatura. Se realizó experimentos con 4 de los algoritmos del Framework del Grupo GTI que utilizan el modelo de representación espacio vectorial, ellos son: LexRank, LexRank Continuo, Procedimiento de Búsqueda del Pescador (FSP) y la Mejor Búsqueda Armónica Global (ESDS-GHS-GLO); usando como esquema de representación del documento, el vector y el centroide de todos los N-gramas. El experimento para los algoritmos LexRank y LexRank Continuo se realizó con una repetición (estos algoritmos son determinísticos, es decir, siempre entregan el mismo resultado ante la misma entrada), y para el algoritmo ESDS-GHS-GLO se realizaron 30 repeticiones (este algoritmo es probabilístico y por eso se debe calcular el promedio de 30 repeticiones). En este trabajo, el Framework tomó la representación matricial de términos por frases (N-gramas sintácticos) y generó los resúmenes candidatos que se evaluaron usando el recall de las métricas ROUGE-1 y ROUGE-2 para determinar la calidad.

5.2 Experimentos con DUC2001 y DUC2002 para todos los sintagmas

En este experimento se incluyeron todos los tipos de sintagmas (nominal, adjetival, verbal, preposicional y adverbial). Los resultados del experimento se presentan en la **Tabla 3**. Esta tabla permite observar tres aspectos importantes, el primero es que los dos algoritmos basados en grafos (LexRank y LexRank Continuo) obtuvieron mejores resultados cuando usan N-gramas sintácticos no continuos en lugar de una representación de bolsa de palabras con términos sencillos. En LexRank se lograron buenos resultados con 2-gramas, obteniendo mejoras entre el 4,5% y el 22,1% en

el dataset DUC2001 y mejoras entre el 5,1% y el 16,4% en el dataset DUC2002. En LexRank Continuo se alcanzaron buenos resultados para 2-gramas (1 caso), 3-gramas (2 casos) y 4-gramas (1 caso) con mejoras entre el 1,5% y el 10,7% en el dataset DUC2001 y mejoras entre el 3,2% y el 10,7% en el dataset DUC2002.

***Tabla 3**. Resultados del experimento con todos los sintagmas en DUC2001 y DUC2002 (mejores resultados en negrita)*

Algoritmo	Modelo	DUC2001		DUC2002	
		R1R	R2R	R1R	R2R
LexRank	bolsa de palabras	42,224	15,858	45,171	18,704
	1-grama	42,224	15,858	45,674	19,325
	2-gramas	**44,133**	**19,361**	**47,481**	**21,779**
	3-gramas	44,014	19,264	47,423	21,630
	4-gramas	44,025	19,040	47,178	21,352
LexRank Continuo	bolsa de palabras	43,420	17,389	45,919	19,589
	1-grama	43,420	17,389	44,697	18,215
	2-gramas	44,025	19,193	47,359	**21,687**
	3-gramas	44,004	**19,242**	**47,386**	21,596
	4-gramas	**44,040**	19,051	47,169	21,344
ESDS-GHS-GLO	bolsa de palabras	44,139	18,890	47,629	**22,107**
	1-grama	**44,874**	**19,370**	**48,194**	22,080
	2-gramas	42,917	17,613	46,661	20,749
	3-gramas	43,130	17,943	46,268	20,215
	4-gramas	43,386	18,203	46,169	20,178
FSP	bolsa de palabras	38,807	9,250	41,384	10,716
	1-grama	38,807	9,250	41,384	10,716
	2-gramas	**39,654**	**14,123**	**41,688**	**11,089**
	3-gramas	36,987	9,175	40,906	9,451
	4-gramas	37,026	9,213	40,737	9,025

El segundo aspecto que se puede observar es que el algoritmo ESDS-GHS-GLO obtuvo los mejores resultados de todos en la representación por centroide con 1-grama, logró mejoras entre 1,7% y el 2,5% en el dataset DUC2001 y mejoras del 1,2% en el dataset DUC2002, pero estos resultados no se mejoraron con el uso de N-gramas con n > 1, perdió calidad entre el 1,7% y el 8,7%.

El tercer aspecto está relacionado con el algoritmo FSP que obtuvo los mejores resultados de todos en la representación por centroide con 2-grama, logró mejoras entre 2,1% y el 34,5% en el dataset DUC2001 y mejoras entre el 0,7% y 3,4% en el dataset DUC2002, pero estos resultados no se mejoraron con el uso de N-gramas con n > 2, perdió calidad entre el 0,4% y el 18,7%.

Las observaciones anteriores, permiten identificar que el impacto o positivo o negativo de los N-gramas sintácticos no continuos en los resultados del resumen dependen de la forma como el algoritmo de generación de resúmenes use los datos de entrada, en este caso, la matriz espacio vectorial basada en N-gramas sintácticos que representa el dataset.

Además, se observó que los resultados entre 1-grama y la bolsa de palabras (algoritmo clásico usado en el Framework del Grupo GTI) para la representación por vector y centroide son iguales en DUC2001, pero son ligeramente diferentes en DUC2002. Esto se debe a que Freeling realiza una fase de detección de entidades y en la bolsa de palabras eso no se hace, por esta razón la matriz de 1-grama en general tiene menos columnas, por ejemplo, con 1-grama “President Bush” es un solo grama, mientras que en la tradicional bolsa de palabras “President” es un término y “Bush” es otro término.

5.3 Experimentos combinando sintagmas para DUC2001

En este experimento se combinaron algunos tipos de sintagmas (nominal, adjetival, verbal, preposicional y adverbial) para el dataset DUC2001. Las matrices espacio vectorial basadas en N-gramas sintácticos fueron las siguientes: 1-grama combinando los sintagmas: i) nominal, adjetival, verbal y preposicional (sin adverbial), ii) nominal, adjetival, verbal y adverbial (sin preposicional), iii) nominal, verbal, preposicional y adverbial (sin adjetival) y iv) nominal, adjetival, preposicional y adverbial (sin verbal). Los resultados se presentan en la **Tabla 4**. En este caso, solamente se logró mejores resultados con el algoritmo LexRank para la combinación ii) nominal, adjetival, verbal y adverbial (sin preposicional) usando

ROUGE-2 recall (R2R). Los demás resultados son inferiores a la línea base (bolsa de palabras), aunque algunos son cercanos. Se puede deducir que para 1-grama todos los tipos de sintagmas son importantes por todas las relaciones de combinación o relaciones sintagmáticas que una palabra mantiene con los demás vocablos en el contexto de un texto aún sin las preposiciones (podrían ser consideradas palabras vacías o stop words), esto determina si una frase debe o no hacer parte del resumen del documento. Otra posibilidad es la variación lingüística, es decir, la posibilidad de utilizar diferentes palabras o expresiones para comunicar una misma idea o la posibilidad de expresar lo mismo, pero cambiando el orden de la estructura sintáctica de la frase al suprimir términos.

***Tabla 4**. Resultados del experimento con combinaciones de sintagmas en DUC2001 con 1-grama*

Algoritmo	Sintagmas	DUC2001	
		R1R	R2R
LexRank	Todos	**42,224**	15,858
	Nom-Adj-Ver-Adv	41,916	16,016
	Nom-Adj-Ver-Prep	42,097	**16,324**
	Nom-Ver-Prep-Adv	41,677	15,654
	Ver-Adj-Prep-Adv	41,222	15,140
LexRank Continuo	Todos	**43,420**	**17,389**
	Nom-Adj-Ver-Adv	41,866	15,441
	Nom-Adj-Ver-Prep	42,002	15,542
	Nom-Ver-Prep-Adv	41,810	15,270
	Ver-Adj-Prep-Adv	41,111	15,182
ESDS-GHS-GLO	Todos	**44,874**	**19,370**
	Nom-Adj-Ver-Adv	43,998	18,539
	Nom-Adj-Ver-Prep	44,047	18,523
	Nom-Ver-Prep-Adv	44,123	18,533
	Ver-Adj-Prep-Adv	43,088	17,829
FSP	Todos	**39,654**	**14,123**
	Nom-Adj-Ver-Adv	36,841	9,564
	Nom-Adj-Ver-Prep	37,004	9,612
	Nom-Ver-Prep-Adv	38,877	9,856
	Ver-Adj-Prep-Adv	38,795	9,923

5.4 Comparativo con los resultados del estado del arte

En este apartado se tomaron en cuenta los resultados de la **Tabla 3** por ser los mejores obtenidos al incluir todos los tipos de sintagmas (nominal, adjetival, verbal, preposicional y adverbial) en la creación del modelo espacio vectorial utilizado por los tres algoritmos y los resultados obtenidos se compararon con los alcanzados por métodos del estado del arte basados en diferentes enfoques para la generación automática de resúmenes de un documento y evaluados con las métricas ROUGE-1 y ROUGE-2 usando el recall.

Los 4 algoritmos seleccionados se renombraron con la terminación –*SNg* (N-gramas sintácticos) para diferenciarlos y compararlos con los algoritmos base los cuales utilizaron bolsa de palabras.

La **Tabla 5** muestra los resultados e incluye un método adicional llamado FPGAC (Abbasi-ghalehtaki et al. 2016) que presenta solamente los resultados del dataset DUC2002, porque los resultados del dataset DUC2001 no se encuentran disponibles. Se observa que los algoritmos propuestos se ubican entre el 4º y 8º puesto en ROUGE-2 para los dos datasets, pero entre los puestos 6º y 15 para ROUGE-1 en los dos datasets. Luego, se aplicó la **Ecuación 7** para calcular el porcentaje de mejora de los resultados obtenidos por los algoritmos seleccionados basados en N-gramas sintácticos respecto a los otros algoritmos del estado del arte. La **Tabla 6**, **Tabla 7** y **Tabla 8** presentan los porcentajes de mejora obtenidos respecto a los otros métodos del estado del arte.

$$100 * (MétodoPropuesto - MétodoDelEstado)/(MétodoPropuesto)$$

***Ecuación 7**. Cálculo del porcentaje de mejora de los resultados obtenidos*

ESDS-GHS-GLO-SNg supera dieciséis algoritmos (incluido ESDS-GHS-GLO con bolsa de palabras) en la métrica ROUGE-1 con el dataset DUC2001, y es superado por siete algoritmos del estado del arte en los que los mejores resultados son obtenidos por DE, que supera a ESDS-GHS-GLO-SNg en un 6,65% (sólo una diferencia de 0,02982) y por FEOM, que supera a ESDS-GHS-GLO-SNg en un

6,36% (sólo una diferencia de 0,02854), ambos son métodos metaheurísticos. ESDS-GHS-GLO-SNg, por su parte, supera a otros algoritmos con mejoras entre un 0,03% y un 18,21% (ver **Tabla 5** y **Tabla 6**).

***Tabla 5**. Puntuaciones ROUGE de los métodos en DUC2001 y DUC2002 (los mejores resultados en negrita)*

#	Método	Año	Enfoque	DUC2001				DUC2002			
				ROUGE-1	Pos	ROUGE-2	Pos	ROUGE-1	Pos	ROUGE-2	Pos
1	LexRank Continuo-SNg	2004	Grafos	0,44004	15	0,19242	6	0,47386	9	0,21687	8
2	LexRank-SNg	2004	Grafos	0,44133	13	0,19361	5	0,47481	8	0,21779	7
3	LexRank	2004	Grafos	0,42224	18	0,15858	18	0,45171	14	0,18704	15
4	LexRank Continuo	2004	Grafos	0,4342	16	0,17389	13	0,45919	13	0,19589	11
5	SVM	2005	Algebráico	0,44628	11	0,17018	15	0,43235	18	0,10867	21
6	FEOM	2006	ML, Optimización	0,47728	2	0,18549	8	0,46575	12	0,1249	16
7	Manifold Ranking	2007	Optimización	0,43359	17	0,16635	16	0,42325	19	0,10677	23
8	CRF	2007	ML	0,45512	5	0,17327	14	0,44006	17	0,10924	20
9	NetSum	2007	ML	0,46427	4	0,17697	11	0,44963	15	0,11167	18
10	QCS	2007	Agrupamiento, Algebráica, ML	0,44852	10	0,18523	10	0,44865	16	0,18766	14
11	CollabSum	2007	Agrupamiento, Grafos	0,4404	14	0,1623	17	0,4719	10	0,201	10
12	DE	2009	Optimización	**0,4786**	**1**	0,18528	9	0,46694	11	0,12368	17
13	UnifiedRank	2010	Grafos	0,45377	6	0,17646	12	0,48478	4	0,21462	9
14	DPSO-EDASum	2011	Optimización	0,3993	19	0,0832	23	0,4172	22	0,1026	24
15	0-1 non-linear	2013	Optimización	0,3876	22	0,0778	24	0,4097	25	0,0937	25
16	FSP-SNg	2014	Optimización	0,39654	20	0,14123	21	0,41688	23	0,11089	19
17	FSP	2014	Optimización	0,38807	21	0,0925	22	0,41384	24	0,10716	22
18	MA-SingleDocSum	2014	Optimización	0,44862	9	0,20142	2	0,4828	5	0,2284	4
19	ESDS-GHS-GLO-SNg	2015	Optimización	0,44874	**8**	0,1937	**4**	0,48194	**6**	0,2208	**6**
20	ESDS-GHS-GLO	2015	Optimización	0,44139	12	0,1889	7	0,47629	7	0,22107	5
21	FPGAC	2016	Optimización, Difuso, Automata		25		25	0,48685	3	0,2291	3
22	ESDS_SMODE	2019	Agrupamiento, Optimización	0,45214	7	**0,2145**	**1**	**0,4912**	**1**	**0,2413**	**1**
23	COSUM	2019	Agrupamiento, Optimización	0,4727	3	0,2012	3	0,4908	2	0,2309	2
24	ESDS_MGWO	2019	Agrupamiento + Optimización	0,37108	23	0,15228	19	0,41849	20	0,18838	12
25	ESDS_MWCA	2019	Agrupamiento + Optimización	0,36702	24	0,14997	20	0,418	21	0,18812	13

Respecto a la métrica ROUGE-2 con el dataset DUC2001, ESDS-GHS-GLO-SNg supera veinte algoritmos (incluido ESDS-GHS-GLO con bolsa de palabras) y es superado por tres algoritmos del estado del arte en los que los mejores resultados son obtenidos por ESDS_SMODE, que supera a ESDS-GHS-GLO-SNg en un 10,74% (sólo una diferencia de 0,02080). ESDS-GHS-GLO-SNg, por su parte, supera a otros algoritmos con mejoras entre un 0,05% y un 59,83% (ver **Tabla 5** y **Tabla 6**).

ESDS-GHS-GLO-SNg supera diecinueve algoritmos (incluido ESDS-GHS-GLO con bolsa de palabras) y es superado por cinco algoritmos del estado del arte en la métrica ROUGE-1 con el dataset DUC2002 donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a ESDS-GHS-GLO-SNg en un 1,92% (diferencia de 0,00923) y por COSUM, que supera a ESDS-GHS-GLO-SNg en un 1.84% (diferencia de 0,00886). ESDS-GHS-GLO-SNg, por su parte, supera a otros algoritmos con mejoras entre un 1,17% y un 14,99% (ver **Tabla 5** y **Tabla 6**).

***Tabla 6**. Porcentaje de mejora obtenida por ESDS-GHS-GLO-SNg (%)*

#	Método	DUC2001		DUC2002	
		ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
1	LexRank-SNg	1,65%	0,05%	1,48%	1,36%
2	LexRank Continuo-SNg	1,94%	0,66%	1,68%	1,78%
3	LexRank-Continuo	3,24%	10,23%	4,72%	11,28%
4	LexRank	5,91%	18,13%	6,27%	15,29%
5	SVM	0,55%	12,14%	10,29%	50,78%
6	FEOM	**-6,36 %**	4,24%	3,36%	43,43%
7	CollabSum	1,86%	16,21%	2,08%	8,97%
8	QCS	0,05%	4,37%	6,91%	15,01%
9	NetSum	-3,46 %	8,64%	6,70%	49,42%
10	CRF	-1,42 %	10,55%	8,69%	50,53%
11	Manifold Ranking	3,38%	14,12%	12,18%	51,64%
12	DE	**-6,65 %**	4,35%	3,11%	43,99%
13	UnifiedRank	-1,12 %	8,90%	-0,59 %	2,80%
14	DPSO-EDASum	11,02%	57,05%	13,43%	53,53%
15	0-1 non-linear	13,62%	59,83%	14,99%	57,56%
16	FSP-SNg	11,63%	27,09%	13,50%	49,78%
17	FSP	13,52%	52,25%	14,13%	51,47%
18	MA-SingleDocSum	**0,03%**	-3,99 %	-0,18 %	-3,44 %
19	ESDS-GHS-GLO	1,64%	2,48%	1,17%	-0,12 %
20	ESDS-GHS-GLO-SNg	0,00%	0,00%	0,00%	0,00%
21	FPGAC			-1,02 %	-3,76 %
22	ESDS_SMODE	-0,76 %	**-10,74 %**	**-1,92 %**	**-9,29 %**
23	COSUM	-5,34 %	-3,87 %	**-1,84 %**	-4,57 %
24	ESDS_MGWO	17,31%	21,38%	13,17%	14,68%
25	ESDS_MWCA	**18,21%**	22,58%	13,27%	14,80%

Respecto a la métrica ROUGE-2 con el dataset DUC2002, ESDS-GHS-GLO-SNg supera diecinueve algoritmos (incluido ESDS-GHS-GLO con bolsa de palabras) y es superado por cinco algoritmos del estado del arte donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a ESDS-GHS-GLO-SNg en un 9,29% (diferencia de 0,02052). ESDS-GHS-GLO-SNg, por su parte, supera a otros algoritmos con mejoras entre un 1,36% y un 57,56% (ver **Tabla 5** y **Tabla 6**).

Al analizar el algoritmo LexRank, seleccionado para el trabajo de tesis, se alcanzó mejores resultados cuando se utilizó con 2-gramas sintácticos (ver **Tabla 3**) en lugar de bolsa de palabras con términos sencillos (algoritmos base comparados en la **Tabla 5**).

La **Tabla 7** presenta los resultados obtenidos con LexRank-SNg el cual supera once algoritmos (incluido LexRank con bolsa de palabras) en la métrica ROUGE-1 con el dataset DUC2001, y es superado por doce algoritmos del estado del arte en los que los mejores resultados son obtenidos por los métodos metaheurísticos DE, que supera a LexRank-SNg en un 8,44% (una diferencia de 0,03723) y por FEOM, que supera a LexRank-SNg en un 8,15% (una diferencia de 0,03595). LexRank-SNg, por su parte, supera a otros algoritmos con mejoras entre un 0,21% y un 16,84% (ver **Tabla 5** y **Tabla 7**).

LexRank-SNg en la métrica ROUGE-2 con el dataset DUC2001, supera diecinueve algoritmos (incluido LexRank con bolsa de palabras) y es superado por cuatro algoritmos del estado del arte, donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a LexRank-SNg en un 10,79% (una diferencia de 0,02089). LexRank-SNg, por su parte, supera a otros algoritmos con mejoras entre un 0,61% y un 59,82% (ver **Tabla 5** y **Tabla 7**).

LexRank-SNg supera diecisiete algoritmos (incluido LexRank con bolsa de palabras) y es superado por siete algoritmos del estado del arte en la métrica ROUGE-1 con el dataset DUC2002, donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a LexRank-SNg en un 3,45% (diferencia de 0,01636) y por COSUM, que supera a LexRank-SNg en un 3.37% (diferencia de

0,01599). LexRank-SNg, por su parte, supera a otros algoritmos con mejoras entre un 0,2% y un 13,71% (ver **Tabla 5** y **Tabla 7**).

***Tabla 7**. Porcentaje de mejora obtenida por LexRank-SNg (%)*

#	Método	DUC2001		DUC2002	
		ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
1	LexRank-SNg	0,00%	0,00%	0,00%	0,00%
2	LexRank Continuo-SNg	0,29%	**0,61%**	**0,20%**	**0,42%**
3	LexRank Continuo	1,62%	10,19%	3,29%	10,06%
4	LexRank	4,33%	18,09%	4,87%	14,12%
5	SVM	-1,12 %	12,10%	8,94%	50,10%
6	FEOM	**-8,15 %**	4,19%	1,91%	42,65%
7	CollabSum	0,21%	16,17%	0,61%	7,71%
8	QCS	-1,63 %	4,33%	5,51%	13,83%
9	NetSum	-5,20 %	8,59%	5,30%	48,73%
10	CRF	-3,12 %	10,51%	7,32%	49,84%
11	Manifold Ranking	1,75%	14,08%	10,86%	50,98%
12	DE	**-8,44 %**	4,30%	1,66%	43,21%
13	UnifiedRank	-2,82 %	8,86%	-2,10 %	1,46%
14	DPSO-EDASum	9,52%	57,03%	12,13%	52,89%
15	0-1 non-linear	12,17%	**59,82%**	**13,71%**	56,98%
16	FSP-SNg	10,15%	27,05%	12,20%	49,08%
17	FSP	12,07%	52,22%	12,84%	50,80%
18	MA-SingleDocSum	-1,65 %	-4,03 %	-1,68 %	-4,87 %
19	ESDS-GHS-GLO	-0,01 %	2,43%	-0,31 %	-1,51 %
20	ESDS-GHS-GLO-SNg	-1,68 %	-0,05 %	-1,50 %	-1,38 %
21	FPGAC	100,00%	100,00%	**-2,54 %**	-5,19 %
22	ESDS_SMODE	-2,45 %	**-10,79 %**	**-3,45 %**	**-10,80 %**
23	COSUM	-7,11 %	-3,92 %	**-3,37 %**	-6,02 %
24	ESDS_MGWO	15,92%	21,35%	11,86%	13,50%
25	ESDS_MWCA	16,84%	22,54%	11,96%	13,62%

Al comparar la métrica ROUGE-2 con el dataset DUC2002, LexRank-SNg supera dieciocho algoritmos (incluido LexRank con bolsa de palabras) y es superado por seis algoritmos del estado del arte, donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a LexRank-SNg en un 10,8% (diferencia de 0,02353). LexRank-SNg, por su parte, supera a otros algoritmos con mejoras entre un 1,36% y un 57,56% (ver **Tabla 5** y **Tabla 7**).

Al analizar el algoritmo LexRank Continuo, seleccionado para el trabajo de tesis, se alcanzó mejores resultados cuando se utilizó 2-gramas, 3-gramas y 4-gramas sintácticos (ver **Tabla 3**) en lugar de bolsa de palabras con términos sencillos (algoritmos base comparados en la **Tabla 5**).

La **Tabla 8** presenta los resultados obtenidos con LexRank Continuo-SNg el cual supera nueve algoritmos (incluido LexRank Continuo con bolsa de palabras) en la

métrica ROUGE-1 con el dataset DUC2001, y es superado por catorce algoritmos del estado del arte en los que los mejores resultados son obtenidos por los métodos metaheurísticos DE, que supera a LexRank Continuo-SNg en un 8,75% (sólo una diferencia de 0,03852) y por FEOM, que supera a LexRank Continuo-SNg en un 8,46% (sólo una diferencia de 0,03724). LexRank Continuo-SNg, por su parte, supera a otros algoritmos con mejoras entre un 1,33% y un 16,59% (ver **Tabla 5** y **Tabla 8**).

***Tabla 8**. Porcentaje de mejora obtenida por LexRank Continuo-SNg (%)*

#	Método	DUC2001		DUC2002	
		ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
1	LexRank-SNg	-0,29 %	-0,62 %	-0,20 %	-0,42 %
2	LexRank Continuo-SNg	0,00%	0,00%	0,00%	0,00%
3	LexRank Continuo	1,33%	9,63%	3,10%	9,67%
4	LexRank	4,05%	17,59%	4,67%	13,75%
5	SVM	-1,42 %	11,56%	8,76%	49,89%
6	FEOM	**-8,46 %**	3,60%	1,71%	42,41%
7	CollabSum	-0,08 %	15,65%	**0,41%**	7,32%
8	QCS	-1,93 %	3,74%	5,32%	13,47%
9	NetSum	-5,51 %	8,03%	5,11%	48,51%
10	CRF	-3,43 %	9,95%	7,13%	49,63%
11	Manifold Ranking	1,47%	13,55%	10,68%	50,77%
12	DE	**-8,75 %**	3,71%	1,46%	42,97%
13	UnifiedRank	-3,12 %	8,29%	-2,30 %	**1,04%**
14	DPSO-EDASum	9,26%	56,76%	11,96%	52,69%
15	0-1 non-linear	11,92%	**59,57%**	**13,54%**	**56,79%**
16	FSP-SNg	9,89%	26,60%	12,02%	48,87%
17	FSP	11,81%	51,93%	12,67%	50,59%
18	MA-SingleDocSum	-1,95 %	-4,68 %	-1,89 %	-5,32 %
19	ESDS-GHS-GLO	-0,31 %	**1,83%**	-0,51 %	-1,94 %
20	ESDS-GHS-GLO-SNg	-1,98 %	-0,67 %	-1,71 %	-1,81 %
21	FPGAC	100,00%	100,00%	**-2,74 %**	-5,64 %
22	ESDS_SMODE	-2,75 %	**-11,47 %**	**-3,65 %**	**-11,27 %**
23	COSUM	-7,42 %	-4,56 %	**-3,57 %**	-6,47 %
24	ESDS_MGWO	15,67%	20,86%	11,68%	13,14%
25	ESDS_MWCA	16,59%	22,06%	11,79%	13,26%

LexRank Continuo-SNg en la métrica ROUGE-2 con el dataset DUC2001, supera dieciocho algoritmos (incluido LexRank Continuo con bolsa de palabras) y es superado por cinco algoritmos del estado del arte en los que los mejores resultados son obtenidos por ESDS_SMODE, que supera a LexRank Continuo-SNg en un 11,47% (diferencia de 0,02208). LexRank Continuo-SNg, por su parte, supera a otros algoritmos con mejoras entre un 1,83% y un 59,57% (ver **Tabla 5** y **Tabla 8**).

LexRank Continuo-SNg supera dieciséis algoritmos (incluido LexRank Continuo con bolsa de palabras) y es superado por ocho algoritmos del estado del arte en la

métrica ROUGE-1 con el dataset DUC2002 donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a LexRank Continuo-SNg en un 3,65% (diferencia de 0,01731) y por COSUM, que supera a LexRank Continuo-SNg en un 3.57% (diferencia de 0,01694). LexRank Continuo-SNg, por su parte, supera a otros algoritmos con mejoras entre un 0,41% y un 13,54% (ver **Tabla 5** y **Tabla 8**).

Al comparar la métrica ROUGE-2 con el dataset DUC2002, LexRank Continuo-SNg supera diecisiete algoritmos (incluido LexRank Continuo con bolsa de palabras) y es superado por siete algoritmos del estado del arte donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a LexRank Continuo-SNg en un 11,27% (diferencia de 0,02445). LexRank Continuo-SNg, por su parte, supera a otros algoritmos con mejoras entre un 1,04% y un 56,79% (ver **Tabla 5** y **Tabla 8**).

Al analizar el algoritmo FSP, seleccionado para el trabajo de tesis, se alcanzó mejores resultados cuando se utilizó 1-gramas (ver **Tabla 3**) en lugar de bolsa de palabras con términos sencillos (algoritmos base comparados en la **Tabla 5**).

La **Tabla 9** presenta los resultados obtenidos con FSP-SNg el cual supera cuatro algoritmos (incluido FSP con bolsa de palabras) en la métrica ROUGE-1 con el dataset DUC2001, y es superado por diecinueve algoritmos del estado del arte en los que los mejores resultados son obtenidos por los métodos metaheurísticos DE, que supera a FSP-SNg en un 20,68% (diferencia de 0,08202) y por FEOM, que supera a FSP-SNg en un 20,36% (sólo una diferencia de 0,08074). FSP-SNg, por su parte, supera a otros algoritmos con mejoras entre un 2,14% y un 7,44% (ver **Tabla 5** y **Tabla 9**).

FSP-SNg en la métrica ROUGE-2 con el dataset DUC2001, supera tres algoritmos (incluido FSP con bolsa de palabras) y es superado por veinte algoritmos del estado del arte en los que los mejores resultados son obtenidos por ESDS_SMODE, que supera a FSP-SNg en un 51,88% (diferencia de 0,07327). FSP-SNg, por su parte, supera a otros algoritmos con mejoras entre un 34,5% y un 44,91% (ver **Tabla 5** y **Tabla 9**) pero al tener en cuenta los algoritmos que lo superan, se concluyó que no es tan representativo.

FSP-SNg supera únicamente a FSP con bolsa de palabras y es superado por veintidós algoritmos del estado del arte en la métrica ROUGE-1 con el dataset DUC2002 donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a FSP-SNg en un 17,82% (diferencia de 0,07429) y por COSUM, que supera a FSP-SNg en un 17.73% (diferencia de 0,07392). Al tener en cuenta los algoritmos que lo superan, se observa que no es tan representativo en ROUGE-1 para el dataset DUC2002.

***Tabla 9**. Porcentaje de mejora obtenida por FSP-SNg (%)*

#	Método	DUC2001		DUC2002	
		ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
1	LexRank-SNg	-11,30 %	-37,09 %	-13,90 %	-96,40 %
2	LexRank Continuo-SNg	-10,97 %	-36,25 %	-13,67 %	-95,57 %
3	LexRank Continuo	-9,50 %	-23,13 %	-10,15 %	-76,65 %
4	LexRank	-6,48 %	-12,28 %	-8,35 %	-68,67 %
5	SVM	-12,54 %	-20,50 %	-3,71 %	2,00%
6	FEOM	**-20,36 %**	-31,34 %	-11,72 %	-12,63 %
7	CollabSum	-11,06 %	-14,92 %	-13,20 %	-81,26 %
8	QCS	-13,11 %	-31,15 %	-7,62 %	-69,23 %
9	NetSum	-17,08 %	-25,31 %	-7,86 %	-0,70 %
10	CRF	-14,77 %	-22,69 %	-5,56 %	1,49%
11	Manifold Ranking	-9,34 %	-17,79 %	-1,53 %	3,72%
12	DE	**-20,68 %**	-31,19 %	-12,01 %	-11,53 %
13	UnifiedRank	-14,43 %	-24,95 %	-16,29 %	-93,54 %
14	DPSO-EDASum	-0,70 %	41,09%	-0,08 %	7,48%
15	0-1 non-linear	2,25%	44,91%	1,72%	15,50%
16	FSP-SNg	0,00%	0,00%	0,00%	0,00%
17	FSP	2,14%	34,50%	0,73%	3,36%
18	MA-SingleDocSum	**-13,13 %**	-42,62 %	-15,81 %	-105,97 %
19	ESDS-GHS-GLO	-11,31 %	-33,75 %	-14,25 %	-99,36 %
20	ESDS-GHS-GLO-SNg	-13,16 %	-37,15 %	-15,61 %	-99,12 %
21	FPGAC			-16,78 %	-106,60 %
22	ESDS_SMODE	-14,02 %	**-51,88 %**	**-17,82 %**	**-117,62 %**
23	COSUM	-19,21 %	-42,46 %	**-17,73 %**	-108,22 %
24	ESDS_MGWO	6,42%	-7,82 %	-0,39 %	-69,88 %
25	ESDS_MWCA	**7,44%**	-6,19 %	-0,27 %	-69,65 %

Al comparar la métrica ROUGE-2 con el dataset DUC2002, FSP-SNg supera seis algoritmos (incluido FSP con bolsa de palabras) y es superado por dieciocho algoritmos del estado del arte donde los mejores resultados son obtenidos por ESDS_SMODE, que supera a FSP-SNg en un 117,62% (diferencia de 0,13043). FSP-SNg, por su parte, supera a otros algoritmos con mejoras entre un 1,49% y un 7,48% (ver **Tabla 5** y **Tabla 9**).

Se analizaron los resultados con los métodos seleccionados del Framework del Grupo GTI basados únicamente en grafos y optimización como se presenta en la

Tabla 10.

ESDS-GHS-GLO-SNg supera a la mayoría de los algoritmos del estado del arte basados en grafos para la métrica ROUGE-1 en los datasets DUC2001 y DUC2002, con mejoras entre un 1,48% y un 6,27%, excepto UnifiedRank que lo supera entre 0,59% y 1.12% (diferencia de 0,00503). En cuanto a la métrica ROUGE-2, supera a todos los métodos basados en grafos con mejoras entre 0,05% y 18,13% para los datasets DUC2001 y DUC2002, por lo tanto, los N-gramas sintácticos obtienen resultados prometedores con métodos de optimización.

***Tabla 10**. Comparativo con grafos y Metaheurísticas no híbridas*

#	Método	Enfoque	DUC2001		DUC2002	
			ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
1	LexRank-SNg	Grafos	1,65%	0,05%	1,48%	1,36%
2	LexRank Continuo-SNg	Grafos	1,94%	0,66%	1,68%	1,78%
3	LexRank Continuo	Grafos	3,24%	10,23%	4,72%	11,28%
3	LexRank-SNg	Grafos	5,91%	18,13%	6,27%	15,29%
10	Manifold Ranking	Optimización	3,38%	14,12%	12,18%	51,64%
11	DE	Optimización	-6,65 %	4,35%	3,11%	43,99%
12	UnifiedRank	Grafos	-1,12 %	8,90%	-0,59 %	2,80%
13	DPSO-EDASum	Optimización	11,02%	57,05%	13,43%	53,53%
14	0-1 non-linear	Optimización	13,62%	59,83%	14,99%	57,56%
15	FSP-SNg	Optimización	11,63%	27,09%	13,50%	49,78%
15	FSP	Optimización	13,52%	52,25%	14,13%	51,47%
15	MA-SingleDocSum	Optimización	0,03%	-3,99 %	-0,18 %	-3,44 %
16	ESDS-GHS-GLO	Optimización	1,64%	2,48%	1,17%	-0,12 %

ESDS-GHS-GLO-SNg supera a la mayoría de los demás métodos de optimización del estado del arte, siendo superado por DE en un 6.65% en la métrica ROUGE-1 para el dataset DUC2001 y por MA-SingleDocSum en un 3,99% en la métrica ROUGE-2 del dataset DUC2001, en el dataset DUC2002, MA-SingleDocSum lo supera en las dos métricas.

LexRank-SNg, LexRank Continuo-SNg y ESDS-GHS-GLO-SNg ocuparon lugares privilegiados en las clasificaciones separadas (ver **Tabla 5**) Se utilizó clasificación unificada de todos los métodos acorde a la **Ecuación 8** adaptada de (Aliguliyev 2009b) para considerar la posición y el número de veces que cada método ocupa en cada medida, con el objetivo de tener una visión más general de los resultados.

$$Rank(Method) = 5 - \frac{\sum_{r=1}^{M}(r * Rr)}{TM}$$

***Ecuación 8**. Cálculo de la clasificación unificada*

Rr representa el número de veces que el método es clasificado en la posición r y TM es el número total de métodos que son comparados, para el trabajo de tesis, se calculó con 21 métodos debido a que el método FPGAC (Abbasi-ghalehtaki et al. 2016) solo presenta los resultados del dataset DUC2002 y esto afecta la clasificación.

La **Tabla 11** contiene los resultados de la clasificación. Se observa que los algoritmos ESDS_SMODE, COSUM, MA-SingleDocSum, y ESDS-GHS-GLO (basados en Metaheurísticas) con bolsa de palabras, tratan el problema de la generación automática de resúmenes de un documento como un problema de optimización, por lo tanto, se podría implementar el uso de N-gramas sintácticos en estos métodos para determinar si se superan los resultados reportados en el estado del arte como se experimentó con ESDS-GHS-GLO. Se alcanzaron lugares privilegiados al usar 2-gramas con todos los tipos de sintagmas en los métodos LexRank y LexRank Continuo (basados en grafos) pero aún se requieren más investigaciones para ser concluyentes.

***Tabla 11**. Clasificación unificada de los métodos para generación automática de resúmenes extractivos genéricos de un documento*

Método	*Rr*																										Ranking Final
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25		
ESDS_SMODE	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		4,60
COSUM	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		4,60
MA-SingleDocSum	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		4,20
ESDS-GHS-GLO-SNg	0	0	0	1	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		4,04
UnifiedRank	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0		3,76
ESDS-GHS-GLO	0	0	0	0	1	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0		3,76
LexRank-SNg	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		3,68
LexRank Continuo-SNg	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0		3,48
FEOM	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0		3,48
DE	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0		3,48
NetSum	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0		3,08
QCS	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0		3,00
CollabSum	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0		2,96
LexRank Continuo	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	1	0	0	0	0	0	0	0	0	0		2,88
CRF	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0		2,80
SVM	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0		2,40
LexRank	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	2	0	0	0	0	0	0	0		2,40
ESDS_MGWO	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0		2,04
Manifold Ranking	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0		2,00
ESDS_MWCA	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0		1,88
FSP-SNg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0		1,68
DPSO-EDASum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0		1,48
FSP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	1	0		1,44
0-1 non-linear	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2		1,16

6 Conclusiones y Trabajo futuro

Basado en los resultados obtenidos en la comparación de Freeling y NTLK, FreeLing es el analizador sintáctico recomendado para obtener N-gramas sintácticos por las siguientes ventajas que ofrece: i) reconoce 13 lenguajes, incluidos el inglés y español, ii) entrega el análisis de constituyentes y dependencias en formato de salida XML, iii) conserva la jerarquía de los elementos textuales en el árbol sintáctico, iv) incluye formatos y etiquetas que identifican la mayor cantidad de información sintáctica y semántica necesaria para crear N-gramas sintácticos y v) no requiere alto consumo de recursos informáticos para procesar datasets de entrenamiento utilizados en la generación automática de resúmenes extractivos de un documento.

La representación espacio vectorial por centroide convertida a una Matriz de Términos por Documento, permitió evaluar el uso de los N-gramas sintácticos no continuos a través de 4 de los métodos implementados en el Framework del Grupo GTI: 2 basados en grafos (LexRank y LexRank Continuo) y 2 basados en metaheurísticas (ESDS-GHS-GLO y FSP). Los resultados experimentales con todos los tipos de sintagmas (nominal, adjetival, verbal, preposicional y adverbial) permitieron mostrar que LexRank y LexRank Continuo obtuvieron resultados similares a bolsa de palabras con términos sencillos cuando se utilizó 1-grama sintáctico y superaron los resultados con 2-gramas sintácticos (ver **Tabla 3**). En el caso de LexRank Continuo, se alcanzaron mejores resultados cuando se utilizó 3-gramas y 4-gramas sintácticos (ver **Tabla 3**). Respecto al algoritmo ESDS-GHS-GLO, los mejores resultados se obtuvieron con 1-grama sintáctico no continuos usando como esquema de representación del documento, el centroide de todos los N-gramas. Esto último, permite concluir que la detección de entidades, por ejemplo: “President Bush”, ayuda al proceso de optimización realizado por ESDS-GHS-GLO. Respecto a FSP, no se alcanzaron resultados prometedores, aunque se utilizó 1-grama sintáctico no continuos usando como esquema de representación del documento, el centroide de todos los N-gramas.

El rango de los tamaños de los N-gramas adecuado para los N-gramas sintácticos es de 1 a máximo 4 gramas. Valores superiores a 4 hacen que la representación espacio vectorial sea aún más dispersa y convierten a los N-gramas sintácticos en oraciones que no se comparten entre las frases del documento. Además, los experimentos realizados demostraron que con 1-grama es necesario tener en cuenta todos los tipos de sintagmas (nominal, adjetival, verbal, preposicional y adverbial) para que se logre obtener mejores resultados que con bolsa de palabras. Para los N-gramas de tamaño 2 a 4 es probable que se pueden omitir algunos términos por las relaciones que se presentan entre ellos, pero para ser concluyentes en este aspecto se deben realizar más experimentos.

Los resultados obtenidos al Incluir la información sintáctica y semántica que tienen los N-gramas sintácticos en el modelo de representación espacio vectorial utilizado por 4 de los algoritmos del Framework del Grupo GTI (LexRank, LexRank Continuo, FSP y ESDS-GHS-GLO) permiten concluir que: i) La metaheurística de optimización basada en N-gramas sintácticos ESDS-GHS-GLO-SNg, logra mejoras entre un 0,05% y 18,13% respecto a los resultados alcanzados por los métodos basados en grafos que utilizan como características a la bolsa de palabras o N-gramas sintácticos, excepto UnifiedRank que lo supera entre 0,59% y 1.12% en la métrica ROUGE-1 para los datasets DUC2001 y DUC2002, ii) Respecto a otros algoritmos metaheurísticos de optimización del estado del arte que no utilizan otras técnicas (agrupamiento, difuso o autómata), ESDS-GHS-GLO-SNg supera a la mayoría y está cerca de los algoritmos que lo superan en la métrica ROUGE-1, MA-SingleDocSum y DE, con diferencias entre el 0,15% y 6,65% para los datasets DUC2001 y DUC2002, en cuanto a la métrica ROUGE-2 para el dataset DUC2001, solo lo supera en 3.99%.

Los algoritmos ESDS_SMODE, COSUM, MA-SingleDocSum, y ESDS-GHS-GLO (basados en Metaheurísticas) con bolsa de palabras (ver **Tabla 11**), tratan el problema de la generación automática de resúmenes de un documentos como un problema de optimización, por lo tanto, se podría implementar el uso de N-gramas

sintácticos en estos métodos para determinar si se superan los resultados reportados en el estado del arte como se experimentó con ESDS-GHS-GLO.

El Grupo de I+D en Tecnologías de la Información (GTI) de la Universidad del Cauca espera en futuras investigaciones, incluir nuevas implementaciones del top 10 del estado del arte en su Framework (la mayoría de estos basados en metaheurísticas) y adaptarles el modelo de N-gramas sintácticos para así incrementar las posibilidades de obtener resultados superiores al estado del arte.

Se propone además, analizar otros algoritmos del estado del arte como por ejemplo la propuesta memética de (Mendoza et al. 2014) o la propuesta multiobjetivo de (Saini et al. 2019) para evaluar el impacto de los N-gramas sintácticos en la calidad de los resúmenes generados.

Igualmente, se propone analizar dos fenómenos que inciden de forma distinta en el proceso de recuperación de información, por un lado, la variación lingüística, es decir, la posibilidad de utilizar diferentes palabras o expresiones para comunicar una misma idea o la posibilidad de expresar lo mismo pero cambiando el orden de la estructura sintáctica de la frase y por el otro, la ambigüedad lingüística que se presenta cuando una palabra o frase da lugar a más de una interpretación en el análisis sintáctico al crear N-gramas sintácticos, consecuencia de la posibilidad de asociar a una frase más de una estructura sintáctica por las relaciones entre las palabras para formar unidades superiores, sintagmas y frases.

7 Referencias

Abbasi-ghalehtaki, Razieh, Hassan Khotanlou, and Mansour Esmaeilpour. 2016. “Fuzzy Evolutionary Cellular Learning Automata Model for Text Summarization.” *Swarm and Evolutionary Computation* 30:11–26.

Abuobieda, Albaraa, Naomie Salim, Yogan Jaya Kumar, and Ahmed Hamza Osman. 2013a. “An Improved Evolutionary Algorithm for Extractive Text Summarization.” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Abuobieda, Albaraa, Naomie Salim, Yogan Jaya Kumar, and Ahmed Hamza Osman. 2013b. “Opposition Differential Evolution Based Method for Text Summarization.” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Alguliev, Rasim M, Ramiz M. Aliguliyev, Makrufa S. Hajirahimova, and Chingiz A. Mehdiyev. 2011. “MCMR: Maximum Coverage and Minimum Redundant Text Summarization Model.” *Expert Systems with Applications* 38(12):14514–22.

Alguliev, Rasim M., Ramiz M. Aliguliyev, and Nijat R. Isazade. 2013a. “CDDS: Constraint-Driven Document Summarization Models.” *Expert Systems with Applications*.

Alguliev, Rasim M., Ramiz M. Aliguliyev, and Nijat R. Isazade. 2013b. “Formulation of Document Summarization as a 0-1 Nonlinear Programming Problem.” *Computers and Industrial Engineering*.

Alguliev, Rasim M., Ramiz M. Aliguliyev, and Chingiz A. Mehdiyev. 2011a. “An Optimization Model and DPSO-EDA for Document Summarization.” *International Journal of Information Technology and Computer Science* 3(5):59–68.

Alguliev, Rasim M., Ramiz M. Aliguliyev, and Chingiz A. Mehdiyev. 2011b. “Sentence Selection for Generic Document Summarization Using an Adaptive Differential Evolution Algorithm.” *Swarm and Evolutionary Computation*.

Alguliyev, Rasim M., Ramiz M. Aliguliyev, Nijat R. Isazade, Asad Abdi, and Norisma Idris. 2019. “COSUM: Text Summarization Based on Clustering and Optimization.” *Expert Systems* 36(1):e12340.

Aliguliyev, Ramiz M. 2007. “A Novel Partitioning-Based Clustering Method and Generic Document Summarization.” in *Proceedings - 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2006 Workshops Proceedings)*.

Aliguliyev, Ramiz M. 2009a. “A New Sentence Similarity Measure and Sentence Based Extractive Technique for Automatic Text Summarization.” *Expert Systems with Applications* 36(4):7764–72.

Aliguliyev, Ramiz M. 2009b. “Performance Evaluation of Density-Based Clustering Methods.” *Information Sciences* 179(20):3583–3602.

Amancio, Diego R., Maria G. V Nunes, Osvaldo N. Oliveira Jr, and Luciano da F. Costa. 2012. “Extractive Summarization Using Complex Networks and Syntactic Dependency.” *Physica A: Statistical Mechanics and Its Applications* 391(4):1855–64.

Asgari, Hamed, Behrooz Masoumi, and Omid Sojoodi Sheijani. 2014. “Automatic Text Summarization Based on Multi-Agent Particle Swarm Optimization.” in *2014 Iranian Conference on Intelligent Systems, ICIS 2014*.

Baeza Yates, Ricardo and Berthier Ribeiro Neto. 1999. “Modern Information Retrieval.” in *ACM press*. Vol. 463.

Barzilay, Regina and Michael Elhadad. 1997. “Using Lexical Chains for Text Summarization.” Pp. 10–17 in *ACL/EACL Workshop on Intelligent Scalable Text Summarization, Madrid, Spain*.

Baxendale, P. B. 2010. “Machine-Made Index for Technical Literature—An Experiment.” *IBM Journal of Research and Development*.

Binwahlan, Mohammed Salem, Naomie Salim, and Ladda Suanmali. 2009. “Swarm Based Text Summarization.” in *2009 International Association of Computer*

Science and Information Technology - Spring Conference, IACSIT-SC 2009.

C. Aone, M. E. Okurowski, J. Gorlinsky, and B. Larsen. 1999. “A Trainable Summarizer with Knowledge Acquired from Robust Nlp Techniques.” *Advances in Automatic Text Summarization , I. Mani and M. T. Maybury*.

Carreras, Xavier, Isaac Chao, Lluís Padró, and Muntsa Padró. 2004. “FreeLing: An Open-Source Suite of Language Analyzers.” in *Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004*.

Chatterjee, Niladri, Amol Mittal, and Shubham Goyal. 2012. “Single Document Extractive Text Summarization Using Genetic Algorithms.” in *Proceedings - 2012 3rd International Conference on Emerging Applications of Information Technology, EAIT 2012*.

Conroy, John M. and Dianne P. O’leary. 2001. “Text Summarization via Hidden Markov Models.” in *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*.

Dehkordi, Pooya Khosraviyan and Farshad Kumarci. 2009. “Text Summarization Based on Genetic Programming.” *International Journal*.

Edmundson, H. P. 1969. “New Methods in Automatic Extracting.” *Journal of the ACM (JACM)* 16(2):264–85.

Erkan, Günes and Dragomir R. Radev. 2004. “LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization.” *Journal of Artificial Intelligence Research* 22(4):457–79.

Ferreira, Rafael, Frederico Freitas, Luciano De Souza Cabral, Rafael Dueire Lins, Rinaldo Lima, Gabriel França, Steven J. Simske, and Luciano Favaro. 2013. “A Four Dimension Graph Model for Automatic Text Summarization.” in *Proceedings - 2013 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2013*.

Gao, Jianfeng, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. 2004. “Dependence Language Model for Information Retrieval.” *Proceedings of the*

27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval 170–77.

García-Hernández, René Arnulfo and Yulia Ledeneva. 2013. “Single Extractive Text Summarization Based on a Genetic Algorithm.” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Gong, Y. and X. Liu. 2001. “Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis.” in *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*.

Hannah, M. Esther, T. V Geetha, and Saswati Mukherjee. 2011. “Automatic Extractive Text Summarization Based on Fuzzy Logic: A Sentence Oriented Approach.” *Proceedings of the Second International Conference on Swarm, Evolutionary, and Memetic Computing - Volume Part I* 530–38.

Hiemstra, Djoerd. 2009. “Information Retrieval Models.” Pp. 1–19 in *Information Retrieval: Searching in the 21st Century*.

Huston, Samuel and W. Bruce Croft. 2014. “A Comparison of Retrieval Models Using Term Dependencies.” *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM ’14* 111–20.

Ibrahim, Ahmed and Tarek Elghazaly. 2013. “Improve the Automatic Summarization of Arabic Text Depending on Rhetorical Structure Theory.” *Proceedings of the 2013 12th Mexican International Conference on Artificial Intelligence* 223–27.

Jiménez, Sabino Miranda, Alexander Gelbukh, and Grigori Sidorov. 2014. “Generación de Resúmenes Por Medio de Síntesis de Grafos Conceptuales.” *Revista Signos*.

Jones, Karen Sparck. 1999. “Automatic Summarising: Factors and Directions.” *Advances in Automatic Text Summarization* 1–12.

Kupiec, Julian, Jan Pedersen, and Francine Chen. 1995. “Trainable Document

Summarizer.” in *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*.

Kyoomarsi, Farshad, Hamid Khosravi, Esfandiar Eslami, Pooya Khosravyan Dehkordy, and Asghar Tajoddin. 2008. “Optimizing Text Summarization Based on Fuzzy Logic.” in *Proceedings - 7th IEEE/ACIS International Conference on Computer and Information Science, IEEE/ACIS ICIS 2008, In conjunction with 2nd IEEE/ACIS Int. Workshop on e-Activity, IEEE/ACIS IWEA 2008*.

Ledeneva, Yulia, René Arnulfo García-Hernández, and Alexander Gelbukh. 2014. “Graph Ranking on Maximal Frequent Sequences for Single Extractive Text Summarization.” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Ledeneva, Yulia, René García Hernández, Romyna Montiel Soto, Rafael Cruz Reyes, and Alexander Gelbukh. 2011. “EM Clustering Algorithm for Automatic Text Summarization.” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Lee, Ju Hong, Sun Park, Chan Min Ahn, and Daeho Kim. 2009. “Automatic Generic Document Summarization Based on Non-Negative Matrix Factorization.” *Information Processing and Management*.

Lin, C. Y. 2004. “Rouge: A Package for Automatic Evaluation of Summaries.” *Proceedings of the Workshop on Text Summarization Branches out (WAS 2004)* (1):25–26.

Lin, Chin-yew and Marina Rey. 2004. “Looking for a Few Good Metrics : ROUGE and Its Evaluation.” *NTCIR Workshop* (June):2–4.

Litvak, Marina, Mark Last, and Menahem Friedman. 2010. “A New Approach to Improving Multilingual Summarization Using a Genetic Algorithm.” Pp. 927–36 in *48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Uppsala, Sweden: Association for Computational Linguistics.

Lloret, Elena and Manuel Palomar. 2012. “Text Summarisation in Progress: A

Literature Review.” *Artificial Intelligence Review* 37(1):1–41.

López Condori, Roque Enrique and Thiago Alexandre Salgueiro Pardo. 2017. “Opinion Summarization Methods: Comparing and Extending Extractive and Abstractive Approaches.” *Expert Systems with Applications* 78:124–34.

Louis, Annie, Aravind Joshi, and Ani Nenkova. 2010. “Discourse Indicators for Content Selection in Summarization.” *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue* 147–56.

Lunh, H. P. 1958. “The Automatic Creation of Literature Abstracts.” *IBM Journal of Research Development*.

Mani, Inderjeet and Mark T. Maybury. 1999. *Advances in Automatic Text Summarization*. Vol. 26.

Marcu, D. 1998. “ Improving Summarization through Rhetorical Parsing Tuning.” Pp. 206–15 in *Sixth Workshop on Very Large Corpora. Montreal, Canada*.

Mashechkin, I. V., M. I. Petrovskiy, D. S. Popov, and D. V. Tsarev. 2011. “Automatic Text Summarization Using Latent Semantic Analysis.” *Programming and Computer Software*.

Meena, Yogesh Kumar and Dinesh Gopalani. 2015. “Feature Priority Based Sentence Filtering Method for Extractive Automatic Text Summarization.” *Procedia Computer Science* 48(0):728–34.

Mendoza, Martha, Susana Bonilla, Clara Noguera, Carlos Cobos, and Elizabeth León. 2014. “Extractive Single-Document Summarization Based on Genetic Operators and Guided Local Search.” *Expert Systems with Applications* 41(9):4158–69.

Mendoza, Martha, Carlos Cobos, and Elizabeth León. 2015. “Extractive Single-Document Summarization Based on Global-Best Harmony Search and a Greedy Local Optimizer.” Pp. 52–66 in *Lecture Notes in Computer Science*. Vol. 9414.

Mendoza, Martha and Elizabeth Leon Guzmán. 2013. “Una Revisión de La

Generación Automática de Resúmenes Extractivos.” *Revista UIS Ingenierías.* 12(1):7–27.

Mihalcea, Rada and Paul Tarau. 2004. “Text-Rank: Bringing Order into Texts.” P. 8 in *Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain.

Montiel Soto, Romyna, René Arnulfo García-Hernández, Yulia Ledeneva, and Rafael Cruz Reyes. 2009. “Comparación de Tres Modelos de Texto Para La Generación Automática de Resúmenes.” *Procesamiento de Lenguaje Natural* 43:303–11.

Muratore, Donatella, Markus Hagenbuchner, Franco Scarselli, and Ah Chung Tsoi. 2010. “Sentence Extraction by Graph Neural Networks.” *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III* 237–46.

Nenkova, Ani and Kathleen McKeown. 2011. “Automatic Summarization.” *Foundations and Trends in Information Retrieval*.

Ono, Kenji, Kazuo Sumita, and Seiji Miike. 1994. “Abstract Generation Based on Rhetorical Structure Extraction.” Pp. 344–48 in *15th conference on Computational linguistics*. Vol. 1. Kyoto, Japan: Association for Computational Linguistics.

Ozsoy, Makbule Gulcin, Ferda Nur Alpaslan, and Ilyas Cicekli. 2011. “Text Summarization Using Latent Semantic Analysis.” *J. Inf. Sci.* 37(4):405–17.

Pal, Alok Ranjan and Diganta Saha. 2014. “An Approach to Automatic Text Summarization Using WordNet.” in *Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014*.

Posadas Durán, Juan Pablo, Ilia Markov, Helena Gómez Adorno, Grigori Sidorov, Ildar Batyrshin, Alexander Gelbukh, and Obdulia Pichardo Lagunas. 2015. “Syntactic N-Grams as Features for the Author Profiling Task.” in *CEUR Workshop Proceedings*. Vol. 1391.

Qazvinian, Vahed, Leila Sharif Hassanabadi, and Ramin Halavati. 2008.

"Summarising Text with a Genetic Algorithm-Based Sentence Extraction." *International Journal of Knowledge Management Studies*.

Rijsbergen, C. J. Van. 1979. "Information Retrieval." *Information Retrieval* 208.

Robertson, Stephen and Steven Walker. 1994. "Some Simple Effective Approximations to the 2 – Poisson Model Probabilistic Weighted Retrieval." *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1):232–41.

Robertson, Stephen, Hugo Zaragoza, and Michael Taylor. 2004. "Simple BM25 Extension to Multiple Weighted Fields." *Proceedings of the 13th ACM Conference on Information and Knowledge Management* 42–49.

Saini, Naveen, Sriparna Saha, Anubhav Jangra, and Pushpak Bhattacharyya. 2019. "Extractive Single Document Summarization Using Multi-Objective Optimization: Exploring Self-Organized Differential Evolution, Grey Wolf Optimizer and Water Cycle Algorithm." *Knowledge-Based Systems* 164:45–67.

Salton, G., A. Wong, and C. S. Yang. 1975. "A Vector Space Model for Automatic Indexing." *Communications of the ACM* 18(11):613–20.

Shareghi, E. and Leila Sharif Hassanabadi. 2008. "Text Summarization with Harmony Search Algorithm-Based Sentence Extraction." Pp. 226–31 in *5th international conference on Soft computing as transdisciplinary science and technology* . Cergy-Pontoise, France: ACM.

Shen, Dou, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. "Document Summarization Using Conditional Random Fields." Pp. 2862–67 in *20th international joint conference on Artifical intelligence*. Hyderabad, India: Morgan Kaufmann Publishers Inc.

Sidorov, Grigori. 2013a. *Construcción No Lineal De N-Gramas En La Lingüística Computacional*. Vol. 1.

Sidorov, Grigori. 2013b. "N-Gramas Sintácticos No-Continuos." *Polibits* (48):69–78.

Sidorov, Grigori. 2013c. "N-Gramas Sintácticos y Su Uso En La Lingüística

Computacional.” *Vectores de Investigación* 6:13–27.

Sidorov, Grigori, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona Hernández. 2013a. “Syntactic Dependency-Based n-Grams: More Evidence of Usefulness in Classification.” *Lecture Notes in Computer Science* 7816:13–24.

Sidorov, Grigori, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona Hernández. 2013b. “Syntactic Dependency-Based n-Grams as Classification Features.” *Lecture Notes in Computer Science* 7630:1–11.

Sidorov, Grigori, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona Hernández. 2014. “Syntactic N-Grams as Machine Learning Features for Natural Language Processing.” *Expert Systems with Applications* 41(3):853–60.

Steinberger, Josef and Karel Ježek. 2006. “Sentence Compression for the LSA-Based Summarizer.” 141–148.

Svore, Krysta M., Lucy Vanderwende, and Christopher J. C. Burges. 2007. “Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources.” in *EMNLP-CoNLL 2007 - Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Tsarev, Dmitry, Mikhail Petrovskiy, and Igor Mashechkin. 2011. “Using NMF-Based Text Summarization to Improve Supervised and Unsupervised Classification.” in *Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems, HIS 2011*.

Uy, Nguyen Quang, Pham Tuan Anh, Truong Cong Doan, and Nguyen Xuan Hoai. 2012. “A Study on the Use of Genetic Programming for Automatic Text Summarization.” in *Proceedings - 4th International Conference on Knowledge and Systems Engineering, KSE 2012*.

Wan, Xiaojun. 2010. “Towards a Unified Approach to Simultaneous Single-

Document and Multi-Document Summarizations.” *Proceedings of the 23rd International Conference on Computational Linguistics* 1137–45.

Wong, Kam Fai, Mingli Wu, and Wenjie Li. 2008. “Extractive Summarization Using Supervised and Semi-Supervised Learning.” in *Coling 2008 - 22nd International Conference on Computational Linguistics, Proceedings of the Conference*.

Yazhini, R. and R. P. Vishnu. 2014. “Automatic Summarizer for Mobile Devices Using Sentence Ranking Measure.” Pp. 1–6 in *Recent Trends in Information Technology (ICRTIT), 2014 International Conference on*.

Yeh, Jen Yuan, Hao Ren Ke, Wei Pang Yang, and I. Heng Meng. 2005. “Text Summarization Using a Trainable Summarizer and Latent Semantic Analysis.” *Information Processing and Management*.

Yousefi-Azar, Mahmood and Len Hamey. 2017. “Text Summarization Using Unsupervised Deep Learning.” *Expert Systems with Applications* 68:93–105.

Zhao, Jiashu, Jimmy Xiangji Huang, and Zheng Ye. 2014. “Modeling Term Associations for Probabilistic Information Retrieval.” *ACM Transactions on Information Systems* 32(2):1–47.