UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Instituto de Investigación en Ciencias Básicas y Aplicadas

Centro de Investigación en Ciencias

Generación Automática de Resúmenes basada en representaciones embebidas y optimización multiobjetivo.

Tesis para obtener el grado de
Doctor en Ciencias

Presenta
Yanet Fors Isalguez

Director de Tesis
Dr. Jorge Hermosillo Valadez

Cuernavaca, Morelos Noviembre,2019

A mis padres Julio y Miriam,
por su amor incondicional, por absolutamente todo...

Agradecimientos

A mis padres por sus sabios consejos y por ser parte esencial de los logros que he alcanzado en la vida. Este será el primer logro que no podremos disfrutar juntos y sé que no sentiré la misma alegría que en los anteriores, aunque espero que donde quiera que estén se llenen de orgullo, ya que ha sido un camino muy difícil y de muchos sacrificios.

A mi esposo por su inmenso amor, por ser ese muro firme donde siempre puedo recostarme, por sacrificar su propia superación personal en aras de la mía y por nuestra familia. Gracias mi vida, espero poder recompensar tanto sacrificio.

A mis niñas Daniela y Vanessa, que fueron la luz cuando todo era oscuridad y desesperanzas, que me dieron la fuerza para levantarme y no ahogarme entre tanta tristeza.

A mi tutor, el Doctor Jorge Hermosillo, por la confianza, por las buenas propuestas, por entender cada una de mis situaciones, por su apoyo en todo momento y por la paciencia. Lamento no haber podido terminar en tiempo.

A Alicia, por todos los mensajes de aliento, por levantarme el ánimo, por las sesiones y las esencias de flores de Bash gratuitas, gracias por estar en los momentos que más lo necesité.

Al Dr. Manuel por los buenos consejos y sus acertadas ideas para conducir esta investigación. Al Dr. Daniel por ser parte de mi comité tutelar y al Dr. Manuel Rendón por su trato siempre amable.

A Francis porque gracias a ella pude iniciar este camino, por ser mi amiga desde hace años y por ser mi Pepe Grillo.

A Yasel por ser un amigo incondicional, porque a pesar de ser más joven aprendí mucho de él, por ser mi modelo de científico y profesional ideal. Por aceptar ser padrino de Dani a pesar de la competencia.

A Clau por todo su cariño, por la linda amistad que surgió en tan poco tiempo, gracias por la sopa, por las presentaciones en latex y por evitar que comprara el pasaje de regreso a Cuba. En verdad esta travesía fue más llevadera gracias a ustedes.

A Damiris, porque en menos tiempo aún se convirtió en una amiga indispensable, gracias por estar y apoyar aunque no te lo pidiera, por estar en los malos momentos a pesar de tus dolores, y gracias por las alegrías y sacarnos de la rutina.

A Luis y Luisa, por permitirnos ser parte de su hermosa familia, por todo el apoyo, y por aceptar ser padrinos de Vanessa.

A mi familia y amigos entrañables en Cuba, a todos los amigos cubanos y mexicanos que de un modo u otro han sido parte del camino.

A CONACYT, por el apoyo económico recibido durante el desarrollo de esta investigación. A la Red de Tecnologías del Lenguaje por el apoyo recibido en mi superación:por las estancias, conferencias y eventos científicos.

Resumen

La tarea de generación automática de resúmenes contribuye a mejorar el manejo del creciente exceso de información digital existente. Se han propuesto muchas soluciones a la tarea desde la perspectiva de la optimización de una función de un solo objetivo, para encontrar encontrar el óptimo global. Esta es una meta poco realista ya que cuando se consideran múltiples objetivos, una solución que optimiza uno de los objetivos puede inducir el efecto contrario en los demás. Recientemente se han propuesto otras soluciones que involucran objetivos múltiples y en conflicto, pero que eventualmente se agregan en una función lineal, lo que resulta en un problema de optimización de un solo objetivo. Además, a menudo se utiliza un modelo típico de bolsa de palabras y se ha hecho poco esfuerzo para incluir las relaciones semánticas entre oraciones para mejorar el rendimiento.

En el presente trabajo de tesis se propone un modelo para la generación automática de resúmenes multidocumentos orientados a consultas. La investigación se centra en las formas de representación de palabras y oraciones que capturen mayor información semántica, en particular las representaciones embebidas. Se recomienda el uso de un método de representación de oraciones modeladas como subespacios y que es aplicado por primera vez en una tarea de resúmenes. También se plantea abordar la tarea como un problema de optimización multiobjetivo donde se necesitan optimizar ciertos aspectos tales como: maximizar la relevancia y minimizar la redundancia y longitud del resumen.

En general propone un método novedoso para el resumen orientado a consultas como un problema de optimización multiobjetivo que tiene en cuenta el frente de Pareto y se basa en una representación embebida de oraciones. Los resultados experimentales muestran que el enfoque contribuye a mejorar el rendimiento en la tarea de resúmenes. Según el conocimiento de los autores, el método es el primer intento de incluir representaciones embebidas de oraciones en una solución de optimización multiobjetivo, que aplica el enfoque de Pareto al resumen orientado a consultas.

Abstract

Text Summarisation task helps to improve the management of the growing excess of existing digital information. Many solutions to the task have been proposed from the perspective of optimizing a single objective function, to find the global optimum. This is an unrealistic goal because when considering multiple objectives, a solution that optimizes one of the objectives can induce the opposite effect on the others. Recently, other solutions have been proposed that involve multiple and conflicting objectives, but that are eventually added into a linear function, which results in a single objective optimization problem. In addition, a typical bag of word model is often used and little effort has been made to include semantic relationships between sentences to improve performance.

In this thesis a model is proposed for the automatic generation of multi-document summaries query oriented. The research focuses on the words and sentences representation that capture more semantic information, particularly embedded representations. We recommend to use a method of sentences representation as subspaces and that is applied for the first time in a summary task. It is also proposed to address the task as a multiobjective optimization problem where certain aspects such as maximizing the relevance and minimizing the redundancy and length of the summary need to be optimized.

In general, it proposes a novel method for query-oriented summary as a multiobjective optimization problem that takes into account the Pareto front and is based on an embedded representation of sentences. Experimental results show that the approach helps improve performance in summary task. According to the authors' knowledge, the method is the first attempt to include embedded representations of sentences in a multi-objective optimization solution, which applies the Pareto approach to the query-oriented summary.

Índice general

Capítulo 1

Introducción

1.1 Generación Automática de Resúmenes

En Inteligencia Artificial (IA), el Procesamiento del Lenguaje Natural (PLN) representa uno de los ámbitos más retadores de las ciencias computacionales por su carácter multidisciplinario, al integrar conocimientos y herramientas de lingüística, aprendizaje de máquina, estadística y teoría de grafos.

En este ámbito, uno de los principales problemas de investigación consiste en desarrollar algoritmos y modelos computacionales para procesar grandes volúmenes de información textual, proveniente de múltiples documentos. Los sistemas de generación automática de resúmenes constituyen una alternativa muy útil para procesar abundante información en el menor tiempo posible. La generación de resúmenes es considerada una de las tareas más complejas dentro de la disciplina del Procesamiento del Lenguaje Natural, debido al amplio número de tareas que implícitamente conlleva.

Las primeras investigaciones en generación automática de resúmenes datan de finales de la década de los 50 y comienzos de la década de los 60, de la mano de (Luhn, 1958) y Edmundson (1969), dos autores de influencia decisiva en el desarrollo posterior de la disciplina. Durante esta primera etapa, las limitaciones impuestas por los recursos de hardware disponibles y por la rigidez de los lenguajes de programación existentes frenaron el avance de las investigaciones; y durante las dos décadas posteriores fueron pocas las aportaciones de cierta relevancia.

La mejora de los computadores, el aumento de la información en la web y los avances en ingeniería lingüística y procesamiento de lenguaje natural despertaron en la década de los 90 el interés por investigar en procesamiento automático de textos. Desde entonces, la dedicación de la comunidad científica a la generación automática de resúmenes ha ido en aumento.

Según (Jones et al., 1999), un resumen consiste en la transformación de un texto fuente, a través de la reducción de su contenido, ya sea por selección o por generalización de lo que es importante. Para Mani (Mani, 2001), en la elabora-

ción de un resumen se han de tener en cuenta tres tipos de factores de contexto. En primer lugar, se deben considerar las propias características del documento que se desea resumir; es decir, su estructura, tamaño, o la especificidad del contenido tratado. En segundo lugar, se ha de considerar la aplicación o uso que se pretende dar al resumen. Así, si su propósito es servir para seleccionar documentos interesantes para una lectura posterior, sin duda la longitud del resumen será menor que si se espera utilizarlo como sustituto del documento original. Por último, el formato en que se presenta el resumen al usuario es también, en ciertos contextos, importante. Así, por ejemplo, un sistema web de noticias que, además del resumen, presente los enlaces a las distintas noticias de las que procede la información, proporciona, sin duda, un valor añadido. Todos estos elementos fueron previamente identificados por (Jones et al., 1999), quien los denominó, respectivamente, factores de entrada, propósito y salida.

Los trabajos en generación de resúmenes pueden ser clasificados en cuanto a diversos criterios, así por ejemplo si hablamos de resúmenes con base en la cantidad de documentos que agrupan, pueden ser resúmenes mono-documentos o multi-documentos. Si son resúmenes aplicados a diferentes idiomas se pueden clasificar en multilingüe o monolingüe para el caso de un solo idioma. En dependencia del tipo de resumen a generar se pueden clasificar también en resúmenes indicativos, que proporcionan una función de referencia al usuario para una lectura más profunda; resúmenes informativos, que cubren toda la información esencial de los textos de entrada, expresando el punto de vista de la persona que realiza el resumen y resúmenes agregativos que incluyen información adicional no presente en el documento original con el objetivo de completar o matizar dicha información.

Sin embargo los enfoques más generales en los que se suelen clasificar las soluciones de generación de resúmenes son extracción y abstracción. Los resúmenes extractivos son aquellos que seleccionan las oraciones más importantes del documento basándose en determinados criterios y conforman el resumen con dichas oraciones. En algunos trabajos se enfocan en tratar de eliminar incoherencias y redundancia. Su principal ventaja es que es un enfoque muy independiente del dominio, y la desventaja es que los resúmenes pueden resultar inconexos y de baja calidad en cuanto a la relevancia del documento. Por otra parte las técnicas de abstracción construyen, en la fase de síntesis, una representación semántica del texto fuente, mediante la identificación de conceptos genéricos y relaciones entre ellos, generalmente haciendo uso de alguna plantilla o esquema que marca la información que se considera importante de acuerdo con el contexto particular en que se genera el resumen. Luego en la fase de síntesis se utiliza generación de lenguaje natural para reescribir el texto que conformará el resumen final. Su principal inconveniente es que se trata de técnicas aplicables a dominios acotados y usan recursos lingüísticos externos.

La Generación de Resúmenes conlleva varias tareas de PLN, entre las que podemos mencionar:

1. Detección de Temas: Identificar los distintos temas tratados en el texto de entrada.

2. Desambiguación Léxica: Resolver la ambigüedad del texto en función del contexto.

3. Resolución de referencias: Resolver las referencias anafóricas y pronominales presentes en el texto, así como identificar los respectivos referentes.

4. Resolución de acrónimos: Resolver posibles acrónimos y abreviaturas.

5. Simplificación y fusión o concatenación de oraciones, con el fin de reducir la información relevante a su mínima expresión.

6. Detección de redundancia: Identificar la información que se repite a lo largo del texto.

De manera general no se abordan todas las tareas en todos los trabajos. (Steinberger et al., 2007) realizan resolución de referencias como paso previo a la selección de oraciones para el resumen. (Filippova and Strube, 2008) simplifican y condensan las oraciones seleccionadas para reducir la longitud del resumen final. (Zhao et al., 2009) y (Plaza et al., 2010) detectan redundancias en sistemas de resúmenes multi-documentos y en (Morales et al., 2008) se analizan los efectos de la desambiguación y de la resolución de acrónimos y abreviaturas sobre la generación de resúmenes.

También se pueden diferenciar las técnicas de generación de resúmenes en función de la profundidad del análisis acometido y del conocimiento empleado. En este caso se puede distinguir 3 enfoques:

- Superficiales.
- Basados en la estructura del discurso.
- Enfoques en Profundidad.

Enfoques superficiales: Fase de Análisis: el texto de entrada se escanea, calculando para cada unidad (frase, oración o párrafo) su importancia por un peso o puntuación indicativa. Para ello se evalúan un conjunto de características para cada unidad, se normalizan y se suman. Fase de Síntesis: se extraen las unidades mejor puntuadas y se hace el resumen mediante simple concatenación. Algunos trabajos además realizan eliminación de redundancia u otro procesamiento para mejorar la coherencia. La principal diferencia entre unas técnicas y otras radica en las características ponderadas para calcular las puntuaciones. Las principales heurísticas que se utilizan son (Paice, 1980):

- Frecuencia de palabras.
- Estructura del documento.
- Localización (posición de la oración)
- Palabras o expresiones indicadoras: Algunas palabras o sintagmas aportan pistas sobre si la oración es relevante o no. En este caso se entrenan modelos sobre corpus para aprender las palabras pero dependen mucho del dominio.

- Uso de aprendizaje en la selección de oraciones. Se usa aprendizaje automático para determinar el conjunto de atributos que mejor se comportan en la extracción de oraciones. Esta técnica es muy dependiente del corpus.

Las técnicas anteriores presentan la ventaja de su relativa sencillez y su bajo coste, ya que apenas hacen uso de conocimiento o de complejas técnicas de procesamiento lingüístico. Además, son relativamente independientes del dominio, lo que sin duda constituye un argumento a su favor. Sin embargo, no resultan apropiadas para todos los tipos de resúmenes. Si se trata, por ejemplo, de resumir textos muy extensos, la taza de compresión que se necesita es muy elevada, y resulta imposible de alcanzar sin utilizar cierto grado de abstracción. Además, los resúmenes generados frecuentemente adolecen de falta de cohesión y coherencia.

Enfoques basados en la estructura del discurso: Constituyen un análisis más sofisticado del lenguaje natural, analizan las relaciones entre palabras o la estructura del discurso. Los estudios plantean que los humanos crean un modelo mental de lo que esperan sea la estructura del documento. Este modelo es lo que las técnicas discursivas aspiran a capturar. Entre los métodos basados en la estructura del discurso existen 2 grupos de técnicas:

1- Análisis de la cohesión del documento. En (Halliday and Hasan, 1996) definen la cohesión textual en términos de las relaciones entre palabras, que determinan qué tan estrechamente conectado está el texto. Distinguen entre:

- Cohesión gramatical: ciertas relaciones lingüísticas como la anáfora, la elipsis y la conjunción.

- Cohesión léxica: relaciones de reiteración, sinonimia y homonimia.

Los trabajos relacionados en esta área se centran alrededor de dos propuestas fundamentales:

- Cadenas léxicas: Se vinculan o relacionan determinadas palabras o expresiones de un texto por su significado.

- Grafos de cohesión: Es la representación más aceptada para la representación de la cohesión textual. Los vértices son los elementos textuales, típicamente oraciones y los arcos representan las relaciones entre ellos.

2- Análisis de la coherencia. La coherencia textual representa la estructura general o super-estructura de un texto, visto como un conjunto de oraciones, y en términos de las relaciones de alto nivel que se establecen entre ellas. Se han propuesto muchas teorías para el análisis de la estructura argumentativa de un texto:

- La teoría de la Estructura Retórica (Mann and Thompson, 1988).

- Las Gramáticas Discursivas (Longacre, 1979).

- Las Macro-estructuras (Dijk, 1988).

- Las Relaciones de Coherencia (Hobbs, 1985).

La Teoría de la Estructura Retórica (RST por sus siglas en inglés) proporciona un análisis de la argumentación de los textos, dirigiendo la organización del discurso a través de las relaciones que se establecen entre las distintas partes del texto. Una de las aportaciones más interesantes es la definición del concepto de la relación retórica para referirse a un tipo de relación asimétrica que se establece entre 2 segmentos de texto a los que se denominan núcleo y satélite. El núcleo contiene la información que es central en el documento y el satélite aporta información que completa o complementa al núcleo. Las relaciones en la RST se definen en 4 campos: restricciones sobre el núcleo, restricciones sobre el satélite, restricciones en la combinación del núcleo y el satélite, y efecto sobre el texto; y relaciones de circunstancia, motivación, propósito y solución. Una vez definidas las relaciones y los segmentos, el texto se representa como un árbol donde los nodos internos representan las relaciones y los nodos hojas representan los segmentos. Se ha aplicado en numerosos sistemas, muchos de los cuales son variantes del sistema original. (Marcu, 2000) se basa en la misma teoría del árbol para representar la estructura retórica y luego calcula la relevancia de los términos que actúan como nodos del árbol y que permite la composición de resúmenes a distintos niveles de detalle. (Radev et al., 2004) desarrollan, la llamada Teoría de la Estructura Inter-Documento, la diferencia subyace en que las relaciones se establecen entre distintos documentos, en lugar dentro de un mismo documento. Está diseñado para la generación de resúmenes multi-documentos.

Enfoques en profundidad: Las técnicas o enfoques en profundidad generalmente realizan resúmenes mediante abstracción. Abstraer implica realizar inferencias sobre el contenido del texto o incluso hacer inferencia a conceptos previos o a un conocimiento que se presupone. De este modo es posible conseguir un mayor grado de compresión en el resumen. Se distinguen 3 etapas en el proceso de generación de un resumen por abstracción:

1. Construcción de una representación semántica de las oraciones del documento.

2. Realización de operaciones de selección, agregación y generalización sobre estas representaciones.

3. Traducción al lenguaje natural.

En las técnicas de abstracción se pueden distinguir 2 líneas:

- Extracción de información: Estos enfoques recorren el texto en busca de un conjunto de información predefinida para incluir en el resumen. Por ello, a pesar de producir resúmenes de alta calidad, su validez se restringe únicamente a dominios muy concretos. Algunos enfoques combinan la extracción de plantillas con técnicas de análisis estadístico (Riloff et al., 1999)

- Compresión: Los enfoques basados en compresión abordan el problema desde el punto de vista de la generación del lenguaje y realizan operaciones de selección, agregación para reescribir el resumen. (Mani, 2001) introduce una aproximación al problema que denomina reescritura de texto.

1.1.1 Resúmenes orientados a consultas

Una de las clasificaciones principales en resúmenes, es cuando se refiere a resúmenes genéricos o resúmenes orientados a consultas. Los resúmenes genéricos permiten obtener un extracto del contenido general o central de los documentos. Los resúmenes orientados a consultas parten del conjunto de documentos y una consulta del usuario, por lo que el resumen generado debe reflejar la información condensada relacionada con la consulta dada. En el resumen centrado en la consulta, la información relacionada con un tema o consulta determinada debe incorporarse en los resúmenes, y deben extraerse las oraciones que satisfagan la necesidad de información declarada del usuario. Muchos métodos para el resumen genérico pueden ampliarse para incorporar la información de la consulta.

En los inicios de la tarea de resúmenes, se abordaban con más frecuencia los resúmenes genéricos. Sin embargo para un usuario no siempre es útil este tipo de resúmenes, pues pudiera estar interesado en algún aspecto específico, que no se aborda como tema central. Por ejemplo, si se tienen un conjunto de noticias con respecto a un evento meteorológico, algunas noticias pudieran hacer más énfasis en los daños humanos y materiales, otras en las acciones del gobierno para la recuperación y otras en analizar cómo el cambio climático está provocando que estos fenómenos ocurran con más frecuencia, o incluso es posible que todas las noticias hagan referencia a un poco de cada cosa. En un resúmen genérico lo más probable es que se refleje la información de los daños, ya que esta es la información más frecuente y abundante, dejando de lado otros temas que pudieran ser el principal interés de un usuario determinado.

De manera general los trabajos de resúmenes orientados a consultas se centran en obtener las oraciones más similares a la consulta para conformar el resumen, ya sea teniendo en cuenta el número de palabras que coinciden en ambas, identificando sinónimos o aplicando alguna medida de similitud (Gong and Liu, 2001) (Pembe and Güngör, 2007) (Wan, 2008) (Ouyang et al., 2011)

Actualmente los resúmenes orientados a consultas han cobrado mayor importancia, ya que son más flexibles y adapatables a las necesidades de cada persona. Un reflejo evidente de ello son las competiciones de resúmenes (DUC[1] y TAC[2]), que comenzaron sólo con resúmenes genéricos y a partir del quinto año en adelante solo han convocado la realización de resúmenes orientados a consultas.

[1]Document Undertanding Conference: Primera competición patrocinada por NIST para la evaluación de resúmenes

[2]Text Analysis Conference: Competición que aborda varias tareas de PLN, incluida la tarea de resúmenes

1.2 Descripción del problema

Aunque la tarea de generación de resúmenes ha evolucionado significativamente, aún existen aspectos sin resolver que siguen requiriendo atención de la comunidad científica.

Uno de los elementos claves que debe tener un resumen, es que abarque totalmente la información requerida, es decir que cuente con la máxima cobertura. Este requisito se torna más complejo cuando se tienen varios documentos a resumir, pues aparece el problema de la redundancia de información, por lo que se deben aplicar técnicas para minimizar este aspecto. Por otra parte, si los resúmenes son orientados a consultas se debe garantizar la obtención de las oraciones que sean más relevantes a la misma. Todo ello aunado a una restricción de longitud, pues los resúmenes deben ser textos cortos.

Como se puede notar, existen varios requisitos que se deben cumplir, y que a la vez se contraponen. Una forma de abordar estos problemas es asumiendo la tarea de resúmenes como un problema de optimización, donde se tengan en cuenta diversos criterios a optimar, que incluso pueden ser contradictorios. Varios trabajos se han propuesto en este sentido Carbonell and Jade (1998) (Mittal and Callan, 2000) (Huang et al., 2010) (García-Hernández and Ledeneva, 2013) , sin embargo hemos analizado que finalmente se resuelve como un problema de un sólo objetivo, pues deciden formular una única ecuación que englobe los difrentes criterios y encontrar la mejor solución a partir de ella. Cuando se tienen criterios que son contradictorios, es decir que minimizando un objetivo, aumenta el otro, es muy difícil que una única solución pueda satisfacer la optimzación.

La optimización multiobjetivo, en su máxima expresión, permite la obtención de diferentes soluciones que cumplen con los criterios de optimalidad, en un rango más amplio, donde es posible seleccionar las soluciones que más se adapaten al problema en cuestión, o incluso a las necesidades específicas del usuario. Dichas soluciones son las que conforman el llamado frente de Pareto. Hasta el momento del desarrollo de la presente investigación no se encontraron trabajos que abordaran la optimización de esta forma.

Otro desafío que aún enfrenta la generación automática de resúmenes y en general el área de PLN, es referente a la representación del texto. Mucho se ha avanzado en este aspecto, pues al día de hoy se cuenta con diferentes métodos y algoritmos que permiten una representación que incorpora mayor semántica. Todo ello gracias a la introducción de técnicas de *"machine learning"*, sobretodo la incorporación de redes neuronales para el aprendizaje de las representaciones. La representación del texto tiene como unidad principal la palabra, por lo que muchas de las propuestas están enfocadas en cómo representar cada término presente en los documentos. En particular, las representaciones vectoriales de oraciones y palabras se han utilizado ampliamente en las tareas de PLN en general. Más recientemente, las representaciones embebidas han cobrado impulso, ya que se ha demostrado que capturan las regularidades lingüísticas y semánticas de manera efectiva y eficiente Mikolov et al. (2013)

Sin embargo para la tarea de resúmenes, y sobretodo los extractivos, es muy importante obtener una representación a nivel de oraciones, para realizar las

operaciones necesarias y poder seleccionar las más apropiadas para conformar el resumen. A pesar de que existen varias propuestas Le and Mikolov (2014) Hill et al. (2016) Socher et al. (2011) Kiros et al. (2015), y se han realizado estudios comparativos que permiten determinar el mejor desempeño de algunas sobre otras, la representación de oraciones sigue siendo un problema abierto, pues no se tienen resultados concluyentes, ya que en la mayoría de los casos, el mejor desempeño de una técnica u otra depende mucho del contexto y de los parámetros que se utilicen.

Debido a los planteamientos anteriores vemos la necesidad de proponer métodos que procuren tener en cuenta múltilples objetivos, reflejando los criterios de lo que se podría considerar un buen resumen, y que a su vez cuenten con una representación del texto que capture la mayor semántica posible para aportar un mejor desempeño en la optimización.

1.3 Preguntas de Investigación e Hipótesis

A partir de los problemas sin resolver, nos planteamos las siguientes preguntas de investigación:

1. ¿ Cómo representar estructuras linguísticas de composición de palabras como frases de oraciones que capturen su relación semántica en estructuras similares?

2. ¿Puede un algoritmo de optimización multiobjetivo, basado en el principio de Pareto, resumir de manera extractiva, considerando criterios antagónicos de optimalidad y mejorar métricas de rendimiento?

3. ¿Cómo integrar estas técnicas de representación y búsqueda de soluciones en un método de resumen extractivo basado en consultas?

Teniendo en cuenta las preguntas de investigación, se formula la siguente hipótesis:

Si se aborda la tarea de Generación automática de Resúmenes Extractivos Multi-documentos como un problema de Optimización Multi-Objetivo donde se optimizacen las funciones de relevancia, cobertura y longitud, y adicionalmente se empleen modelos que aporten mayor semántica en la representación de los textos, se puede obtener un resumen de mayor calidad.

1.4 Objetivos

Con el propósito de dar solución a las preguntas de investigación planteadas, definimos como objetivo general:

- Proponer un modelo para la generación de resúmenes extractivos de múltiples documentos basado en optimización multiobjetivo con enfoque de Pareto y representaciones embebidas que mejore las métricas de los resúmenes con respecto al estado del arte.

y como objetivos específicos:

- Proponer un método de Optimización Multiobjetivo que incluya los principales criterios para generar un resumen de calidad.
- Proponer un método para el modelado semántico de oraciones.
- Desarrollar el modelo propuesto.
- Validar el modelo mediante experimentaciones.

1.5 Contribuciones

La constribuciones de la presente tesis son las siguientes:

- Se propone abordar la tarea de resumen automático como un problema de optimización multiobjetivo (MOO), que consiste en encontrar un conjunto de compensaciones óptimas, el llamado conjunto óptimo de Pareto Zitzler et al. (2000). En este trabajo, se investiga la viabilidad de este tipo de enfoque comparando su desempeño con el método linealizado.
- Se propone el uso de un modelo de representación de oraciones que explote las relaciones semánticas entre ellos, ya que esto podría ayudar a mejorar la relevancia, la cobertura y el rendimiento sin redundancia.
- Aunque la optimización multiobjetivo se ha aplicado durante la última década a una serie de problemas en la minería de datos Freitas (2004), según nuestro conocimiento, no hay ningún estudio en el área de resumen automático centrado en la consulta que compare los resultados obtenidos con un enfoque de optimización de objetivo único contra un método de optimización multiobjetivo (en el sentido de Pareto), que además utiliza representaciones embebidas de palabras para modelar las oraciones.

Como parte del desarrollo de la presente investigación, se logró la participación en un evento, un congreso y la publicación de los resultados en una revista:

- Yanet Fors Isalguez, Jorge Hermosillo Valadez. Algoritmo Word2Vec para la representación vectorial de oraciones en generación de resúmenes. 12 Taller de Tecnologías del Lenguaje Humano, Ciudad de México,2015.
- Yanet Fors Isalguez, Jorge Hermosillo Valadez, Manuel Montes y Gómez. Query-oriented Text Summarization based on Multiobjective Evolutionary Algorithms and Word Embeddings. 5Th International Symposium on Language & Knowledge Engineering, LKE'2017. Faculty of Computer Science at the Benemérita Universidad Autónoma de Puebla (BUAP), México. 2017

- Yanet Fors-Isalguez, Jorge Hermosillo Valadez, Manuel Montes-y-Gómez. Query-oriented text summarization based on multiobjective evolutionary algorithms and word embeddings. Journal of Intelligent and Fuzzy Systems 34(5): 3235-3244 (2018)

1.6 Estructura de la tesis

En el capítulo 2 se presenta una revisión sobre el estado del arte de los principales modelos de representación de palabras, comenzando con modelos clásicos y más sencillos como el de Bolsa de Palabras y explorando otros modelos más complejos, basados en redes neuronales pero que aportan mayor semántica en la representación. Finalmente se describen algunos trabajos de generación de resúmenes que utilizan los modelos revisados.

En el capítulo 3 se presentan las principales técnicas de optimización, haciendo especial énfasis en los algoritmos para la optimización multi-objetivo tales como los algoritmos genéticos. De manera general se explican conceptos importantes para comprender como se trabaja con esas técnicas o algoritmos.

En el capítulo 4 se presenta el modelo propuesto para la Generación de resúmenes multi-documentos, así como los diferentes experimentos que se llevaron a cabo para demostrar su validez.

En el capítulo 5 se presentan las conclusiones, a partir del desarrollo y experimentación del modelo propuesto, se resumen las contribuciones y se proponen alternativas de trabajo a futuro.

Capítulo 2

El problema de la representación semántica de texto

Las investigaciones en Psicología e Inteligencia Artificial han acumulado mucha evidencia sobre la importancia de representaciones apropiadas, tanto para humanos como para sistemas artificiales inteligentes (Fink, 2007; Russell and Norvig, 2016). En el ámbito del Procesamiento de Lenguaje Natural (PLN) (Martin and Jurafsky, 2009) sin embargo, la noción de "buena" representación de texto sigue siendo un problema abierto en general.

A continuación presentamos una síntesis del estado del arte, resaltando las técnicas pioneras o los trabajos relacionados más relevantes en torno al problema de la representación de texto.

2.1 Representación de texto basada en grafos

Los métodos basados en grafos, representan los documentos como un grafo conectado. Las oraciones forman los vértices del grafo y las aristas entre las oraciones indican cuán similares son dos oraciones. Una técnica común empleada para conectar dos vértices es medir la similitud de dos oraciones y si es mayor que un umbral están conectadas. Con esta representación basada en grafos se obtienen dos resultados importantes. Primero, las particiones (sub-grafos) incluidos en el grafo, crean temas discretos cubiertos en los documentos. El segundo resultado es la identificación de las oraciones importantes en el documento. Las oraciones que están conectadas a muchas otras oraciones en la partición son posiblemente el centro del grafo y son las más probables a incluir en el resumen.

2.1.1 Trabajos relacionados

(Erkan and Radev, 2004) presentan LexRank, uno de los métodos más aceptados para calcular la centralidad de un grafo aplicado a la generación automática de resúmenes multi-documento. Se construye un grafo para el conjunto de documentos a resumir con un vértice por cada oración. Para los enlaces entre los vértices, las oraciones se representan por sus vectores de frecuencias *tf-idf* y se calcula la similitud léxica entre ellos por la métrica del coseno, obteniendo una matriz de similitudes. Los pares de oraciones con similitud mayor a un umbral se entrelazan entre sí en el grafo, partiendo de la hipótesis de que las relaciones 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. Se investigan distintas definiciones de centralidad léxica en múltiples documentos: centralidad basada en el grado y centralidad basada en vectores propios. (Mihalcea and Tarau, 2004) utilizan el algoritmo TextRank para generación de resúmenes mono documentos. Los nodos del grafo representan a las oraciones y la medida de similitud entre 2 oraciones es basada en el número de palabras que tienen en común.

(Cohn and Lapata, 2009) proponen un método de compresión de oraciones a partir de la creación de árboles sintácticos por oración. En esencia el objetivo es lograr reescribir las oraciones a partir de un conjunto de reglas gramaticales, las cuales también son representadas con una estructura arborea. Estas reglas definen cual puede ser el árbol fuente y como quedaría el árbol final una vez aplicada la regla. Cada regla gramatical generalmente tiene una puntuación de la cual se puede derivar la puntuación global de una compresión y para la oración x. Para ello adoptan el formalismo de la gramática de sustitución de árbol sincrónica (STSG) de (Eisner, 2003) que puede modelar estructuras de árbol no isomorfas mientras se tienen algoritmos de inferencia eficientes. Muestran cómo se puede inducir tal gramática a partir de un corpus paralelo y proponen un modelo discriminativo para la tarea de reescritura que puede verse como un transductor ponderado de árbol a árbol.

Así por ejemplo (Miranda-Jiménez et al., 2013) proponen la creación de grafos conceptuales para una representación semántica completa del texto en la generación automática de resúmenes de un solo documento. Luego obtienen el resumen aplicando distintas operaciones sobre los grafos, tales como: generalización, unión o asociación, ponderación y poda. Para ambos procesos, el de generación de los grafos y el de síntesis, se apoyan en recursos linguísticos como WordNet(para obtener jerarquías de conceptos) y VerbNet(para obtener reglas heurísticas basadas en los patrones semánticos). Finalmente, como el resultado es un grafo conceptual resumido, diseñaron sus propias medidas de evaluación, teniendo en cuenta los conceptos del grafo y los conceptos que aparecen en los resúmenes humanos, concluyendo que el método es adecuado para resúmenes cortos. (Ferreira et al., 2014) propusieron un nuevo algoritmo de agrupamiento de oraciones utilizando un modelo de grafos que adapta los principios de similitud estadística y tratamiento lingüístico. En el algoritmo utilizan la similitud estadística de la frecuencia de las palabras, la similitud semántica de los sinóni-

mos y el hipónimo de las palabras en WordNet[1] y también toman en cuenta la correlación y las relaciones de discurso entre oraciones.

Otras propuestas son las presentadas en (Mendoza et al., 2014) en resúmenes de un solo documento basado en grafos,(Glavaš and Šnajder, 2014) proponen el uso de grafos de eventos, y un enfoque de indexación de palabras basado en el contexto con respecto a los modelos de grafos (Goyal et al., 2013)

Las representaciones basadas en estructuras han sido muy utilizadas sobretodo en la generación de resúmenes abstractivos para lograr una representación semántica del texto, y en muchos casos obtener una compresión del texto que permita la generación del resumen final con cierto nivel de abstracción.

2.2 Representación vectorial

La representación vectorial constituye uno de los métodos más utilizados para la representación del texto, que ha evolucionado desde modelos muy sencillos como el bolsa de palabras hasta modelos que buscan capturar la mayor información semántica posible como el word2vec. Muchos de los algoritmos de aprendizaje autómatico utilizados en PLN prefieren que la información de entrada y salida sean datos de longitud fija bien definidos, por lo que muchos no pueden funcionar con texto directamente sin ningún formato o estructura. De ahí que se hace necesario representar el texto como vectores de números específicamente. Los vectores se derivan de datos textuales para reflejar diversas propiedades lingüísticas del texto, lo que se conoce como extracción de características o codificación de características.

2.2.1 Bolsa de Palabras

Un método clásico y simple de extracción de características con datos de texto se denomina modelo de texto de bolsa de palabras. El modelo de bolsa de palabras fue propuesto inicialmente para la tarea de recuperación de información y luego se aplicó en otras tareas de PLN e incluso en el campo de visión por computador (Sivic and Zisserman, 2009). Es un modelo clásico que en el caso de la recuperación de información parte de una colección de documentos en la cual se tienen n términos no repetidos(que conforman el vocabulario) $w_1, ..., w_n$ y cada documento es representado por un vector $n-$dimensional, cuyo componente i es la frecuencia de la palabra w_i en el documento. En recuperación de información el documento es el contexto, pero esto pudiera cambiar en dependencia del problema a resolver, por ejemplo para la tarea de generación de resúmenes el contexto suele ser la oración.

A estos vectores, se les conoce como vectores de características, y la característica a representar puede variar. En sus inicios solo se representaban como un vector binario con un 1 si el término aparecía y 0 si no. Luego se pasó a

[1]Base de datos léxica del idioma inglés que agrupa las palabras en conjuntos de sinónimos y establece relaciones entre ellos. https://wordnet.princeton.edu/

representar el número de veces que ocurría el término (tf)[2], pues esto daba una medida de la importancia del mismo dentro del texto. La representación de el número de ocurrencias del término tuvo otra variante conocida como tf-idf (Schütze et al., 2008) que es una medida de peso que permite discriminar aquellos términos que aparecen con demasiada frecuencia, ya que pudiera tratarse de palabras muy comunes que no aporten una gran relevancia al contexto como los artículos y preposiciones. Otra mejora para este modelo fue con respecto a la conformación del vocabulario, en vez de hacer un análisis de la aparición de cada término, se propuso un modelo de n-gramas (n-términos) y lo que se analizaba eran pares de términos o más en dependencia del número de gramas n.

Aunque es un modelo sencillo, fácil de implementar y ha sido utilizado con éxito en varios problemas de modelado del lenguaje, presenta algunas desventajas importantes como el hecho de que no considera el orden de las palabras o cómo estas co-ocurren entre sí, por lo que deja a un lado la semántica y solo se centra en contabilizar. Por otra parte los vectores tienden a ser dispersos, poco densos, lo cual puede representar un problema desde el punto de vista computacional en cuestiones de espacio y tiempo, ya que se requiere de una representación espacial muy grande para almacenar poca información.

2.2.2 Representación latente de texto

2.2.2.1 Principales métodos

Los espacios vectoriales semánticos se basan en la idea que el significado de una palabra puede ser aprendido de un entorno lingüístico y poseen dos enfoques: la semántica distribucional y la semántica composicional. El primer enfoque analiza el significado de palabras individuales y el segundo enfoque el significado de frases, oraciones y párrafos. La mayoría de los modelos propuestos en este tipo de espacio vectorial se basan en el principio de la semántica distribucional (Harris, 1954) que plantea que palabras usadas en el mismo contexto tienden a tener significados similares. A continuación se explican algunos de estos modelos.

LSA: El análisis semántico latente fue propuesto por (Deerwester et al., 1990) en 1990, también para el área de Recuperación de Información, es una teoría y un método para extraer y representar el significado contextual de las palabras mediante cálculos estadísticos aplicados a un gran corpus de texto. El modelo consiste en conformar una matriz de términos contra documentos, con la frecuencia de aparición de cada término en cada documento. Luego se propone reducir la dimensionalidad de dicha matriz mediante una técnica de factorización de matrices, llamada descomposición de valores singulares (Singular Value Descomposition; SVD), para encontrar un espacio semántico latente. En esta técnica la matriz de término-documento se descompone en un conjunto de factores ortogonales, a partir de los cuales la matriz original puede aproximarse por una combinación lineal. La idea es que SVD induce relaciones entre filas o entre columnas, que son similares a otras filas o columnas en la matriz

[2]term-frecuency: medida utilizada para indicar la frecuencia del término

de coocurrencia original y de esta forma LSA agrupa palabras que ocurren en contextos similares.

PLSA: El análisis semántico latente probabilístico se propuso en 1999 por Hofmann como una versión probabilística de LSA (Hofmann, 1999). Hofmann declara que LSA posee varias limitaciones debido a que no tiene una base estadística. Por lo que PLSA es un modelo generativo probabilístico, basado en un modelo de aspectos y fue desarrollado para el análisis estadístico del texto. Este modelo se utiliza para descubrir la semántica de tópicos ocultos en documentos usando la representación de bolsa de palabras (Ren and Han, 2014).

LDA: El modelo de asignación latente de Dirichlet fue introducido por primera vez por Blei et al. 2003. LDA es un modelo probabilístico generativo para colecciones de datos discretos como es el caso de un corpus textual donde se utiliza una representación de bolsa de palabras. Es un método popular en el modelado de tópicos. Específicamente, es un modelo bayesiano jerárquico de tres niveles (documento, palabra y tópico), el cual considera a un tópico como "una distribución sobre un vocabulario fijo" . El modelo toma previamente una cantidad de tópicos predefinida para toda la colección y se definen las palabras que pertenecen a esos tópicos. El procesamiento del modelo consiste básicamente en identificar en qué medida esos tópicos se presentan en los documentos. Primero se escoge una distribución sobre los tópicos, es decir, el conjunto de tópicos predefinidos con sus palabras más probables; luego, para cada palabra del documento se escoge una asignación de tópicos y se selecciona la palabra para el tópico correspondiente.

PCA: El Análisis de Componentes Principales constituye un procedimiento matemático que permite transformar un número de variables posiblemente correlacionadas en un número menor de variables no correlacionadas (ortogonales), llamadas componentes principales. El primer componente (eje) absorbe la mayor cantidad de variabilidad posible del conjunto de datos y cada uno de los componentes restantes absorbe el resto. Es un método tradicionalmente utilizado para reducir la dimensionalidad de un conjunto de datos multivariados o identificar nuevas variables subyacentes al conjunto de datos que permitan una mejor interpretación de los mismos. Es de mucha utilidad cuando se quiere obtener una representación visual de los datos de alta dimensionnalidad en aras de poder identificar patrones en los mismos. Este método también ha sido empleado en el área de resúmenes para la representación textual.

2.2.2.2 Trabajos relacionados

El modelo LSA se ha aplicado en varios trabajos en el área de resúmenes, así por ejemplo en (Gong and Liu, 2001) proponen un método basado en LSA para generar resúmenes genéricos de un solo documento. Conforman la matriz de términos por oración $A = [A_1, A_2, \ldots, A_n]$, donde cada columna A_i representa un vector con el peso de la frecuencia de términos de la oración i en el documento. Si hay m términos y n oraciones en el documento, entonces la matriz A para el documento será de $m \times n$ (donde $m >= n$). Esta matriz A es dispersa porque cada término aparece esporádicamente en cada oración. El siguiente pa-

so consiste en aplicar SVD a la matriz A, así,

$$A = U\Sigma V \tag{2.1}$$

donde:
$U = [u_{ij}]$ es una matriz de columnas ortonormales de $m \times n$ cuyas columnas son llamadas vectores singulares de izquierda,
$\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_n)$ es una matriz diagonal de $n \times n$, cuyos elementos diagonales son valores singulares no negativos en orden descendente
$(\sigma_1 >= \sigma_2 .. >= \sigma_r > \sigma_{r+1} = .. = \sigma_n = 0)$ y $V = [vij]$ es una matriz ortonormal de $n \times n$, cuyas columnas se denominan vectores singulares derechos.
La dimensionalidad de las matrices es reducida a las r dimensiones más importantes y, por tanto, U' es $m \times r$, Σ' es $r \times r$ y V^T es una matriz de $r \times n$.
Si un patrón de combinación de palabras es destacado y recurrente en un documento, este patrón es capturado y representado por uno de los vectores singulares, la magnitud de este vector indica el grado de importancia de este patrón dentro del documento. Las oraciones que contengan este patrón de combinación de palabras serán proyectadas en este vector singular, y la oración que mejor represente este patrón tendrá el valor del índice más grande dentro del vector. Partiendo de que cada patrón de combinación de palabras describe un tópico en el documento, cada vector singular representa cada tópico y la magnitud de su valor singular representa el grado de importancia de este tópico. Para el resumen, este método selecciona las oraciones cuya representación vectorial sea mayor, escogiendo la oración con el ponderado más grande a través de todos los tópicos. Entre las desventajas que se reportan de este método se encuentran el hecho de que solo permite seleccionar una oración por tópico, aunque pudieran existir otras relevantes.

Los siguientes trabajos en resúmenes que se propusieron utilizando LSA, aplican de forma similar los primeros pasos de crear la matriz inicial y aplicar SVD y cambian la forma de selección de las oraciones para el resumen. Un enfoque similar es presentado por (Steinberger and Jezek, 2004) en resúmenes de un solo documento, pero cambiando el criterio de selección para incluir en el resumen las oraciones cuya representación vectorial en la matriz tengan la longitud más grande, en lugar de las oraciones que contienen el mayor valor del índice para cada tópico, permitiendo incluir más de una oración relacionada con un tópico importante, en lugar de una oración para cada tópico. Además proponen una extensión del mismo método para resúmenes multi-documentos en 2007 (Steinberger and Křišt'an, 2007).

Por otra parte en 2011, los autores de (Ozsoy et al., 2011) proponen dos métodos para la selección de las oraciones: el método cruzado y el método por tópico. El método cruzado es una extensión del enfoque de (Steinberger and Jezek, 2004) e incluye un paso intermedio entre el cálculo de los valores singulares y la selección de las oraciones.

(Hennig, 2009) propone un método orientado a consulta basado en PLSA, el cual permite representar las oraciones y las consultas como distribuciones de probabilidad sobre tópicos latentes. PLSA permite modelar los documentos

como una mezcla de tópicos. El resumen se produce en tres pasos: (1) Crear la matriz de términos por oración y entrenar el modelo PLSA sobre esta matriz; (2) Calcular las diferentes características a nivel de oración basado en la similitud de las distribuciones de las oraciones y de la consulta sobre los tópicos latentes; (3) Calcular el puntaje de la oración como la combinación lineal de los puntajes de las características y ordenar las oraciones de acuerdo al puntaje, luego utilizan Máxima relevancia marginal(MMR) Carbonell and Jade 1998 para seleccionar las oraciones y penalizar las oraciones candidatas basado en su similitud con el resumen parcial.

En el año 2012, (Nagesh and Murty, 2012), usan un modelo de tópicos basado en la asignación latente de Dirichlet (Latent Dirichlet allocation, LDA), identificando los temas que mejor describen el documento (solo pocos tópicos tienen alta probabilidad en la distribución tópico-documento). Construyen una matriz de similitudes de los párrafos por cada tópico identificado, que es usada para puntuar los tópicos y seleccionar los de mayor puntaje como los "tópicos resumen". Luego agrupan los párrafos en cada "tópico resumen" para ponderar cada tópico, por último usando el teorema de Bayes obtienen el peso de cada oración de los "tópicos resumen" y las oraciones con pesos más altos forman parte del resumen.

En (Lee et al., 2003) utilizan PCA para la generación de resúmenes de un solo documento. La propuesta toma ventaja de las relaciones de co-ocurrencia entre palabras en el documento para identificar las oraciones más relevantes. La técnica de PCA se utiliza para extraer las palabras claves del documento. Solo toman en cuenta los sustantivos que ocurren más de dos veces y estos son definidos como variables $(X1, ...Xn)$. Las oraciones serían las observaciones y conforman una matriz de observaciones por variables donde ubican la frecuencia acumulativa de cada variable en el documento, es decir de cada palabra sobre las oraciones. A dicha matriz le aplican PCA, obteniéndose los eigenventores y eigenvalores. Como los coeficientes determinan el grado de relación entre las variables y un componente principal, seleccionan las palabras con coeficiente mayor que 0.5 para representar el componente principal. Una vez obtenidas las palabras temáticas, le dan una puntuación a cada oración del documento según el número de palabras temáticas que aparezcan en la misma, ordenan las oraciones de acuerdo a dicha puntuación y seleccionan las oraciones de mayor puntaje para conformar el resumen. Los resultados experimentales usando artículos de noticias en koreano mostraron que el método propuesto es superior a los métodos que usan frecuencia de palabras y cadenas léxicas usando tesauros.

Por último en (Vikas et al., 2008) proponen un método de resumen de texto multidocumentos que utiliza semántica de datos para formar un resumen eficiente y relevante. El resumen se genera mediante la construcción de un Modelo de espacio de vectores estadístico y luego se modifica utilizando el concepto de palabras de acción para formar un Modelo de espacio de vectores semánticos. Las palabras de acción se identifican utilizando el clasificador de palabras de acción que utiliza Wordnet para analizar la semántica de la palabra. Luego se aplica el análisis de componentes principales en el modelo de espacio vectorial semántico para reducir la dimensión de los conjuntos de datos multidimensio-

nales. La descomposición del valor singular se lleva a cabo en SVSM como parte de PCA para producir valores singulares y vectores propios. Después realizan una retroproyección para proyectar los documentos en el espacio propio, lo que arroja valores proyectados de los documentos que, en lo sucesivo, se comparan con los valores singulares para generar el documento/tema más relevante. A la extracción de oraciones de varios conjuntos de documentos se le asigna un peso de acuerdo con las palabras claves obtenidas del documento/tema más importante. Las frases con mayor peso se toman para formar un resumen.

2.3 Representación semántica usando redes neuronales

2.3.1 El concepto de representación con redes neuronales

En los últimos años han surgido varias investigaciones y propuestas para representar texto mediante vectores aprendidos a partir de las propiedades estadísticas de la distribución de las palabras usando redes neuronales. A estos vectores se les conoce como *representaciones distribuidas*, *representaciones vectoriales densas* o *representaciones embebidas* de palabras—*Word Embeddings* (WE).

Los modelos de representación distribuidos se basan en la hipótesis de la semántica distribucional. Bajo este supuesto, las estadísticas de co-ocurrencia de las palabras extraídas de un corpus de textos, sirven como base para obtener representaciones semánticas. La hipótesis es que el grado de similitud semántica entre dos expresiones lingüísticas es una función de la similitud de los contextos lingüísticos en los que aparecen ambas. Por lo tanto, estas técnicas capturan las relaciones semánticas globales entre los términos de un corpus: la semántica se modela por la cercanía de un término a otro en este espacio vectorial medido por la similitud del coseno. Estas representaciones embebidas de texto son un mecanismo adecuado para modelar fenómenos semánticos del lenguaje y analizar propiedades lingüísticas, como lo han demostrado varios trabajos de investigación en la literatura.

El cálculo de representaciones distribuidas utilizando redes neuronales se remonta a Hinton et al. (1984), donde *"cada entidad está representada por un patrón de actividad distribuida en muchos elementos computacionales, y cada elemento computado está involucrado en la representación de muchas entidades diferentes."*. En otras palabras, las capas ocultas en las redes neuronales son codificadores de datos. En el contexto del modelado de lenguaje estadístico (Manning et al., 1999), el primer intento de construir un modelo de lenguaje por medio de una red neuronal se debe a Bengio et al. (2003).

Para comprender qué son estos vectores y cómo se calculan, presentamos las ideas principales detrás de los principios del algoritmo word2vec de Mikolov et al. (2013).

2.3.2 Principio de operación del algoritmo word2vec

El documento original de Mikolov et al. (2013) es muy conciso con respecto a los detalles técnicos. Se puede encontrar una explicación más detallada en (Rong, 2014).

En sus orígenes, el cálculo de representaciones embebidas buscaba modelar la distribución de probabilidad de las palabras en un corpus de textos. En el caso de las palabras, estos métodos toman una palabra y su contexto (típicamente algunas palabras antes y después) para producir un vector de una dimensión fija, mucho más pequeña que el tamaño del vocabulario, que representa el término en función de sus propiedades de distribución en el corpus.

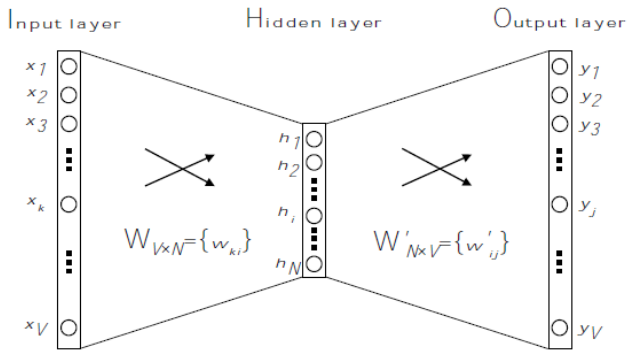

Figura 2.1: Modelo simple de CBOW model con una sola palabra en el contexto

Un modelo de lenguaje estadístico es esencialmente una distribución conjunta del vocabulario V de un corpus de documentos. Si las palabras están representadas por w_i y el tamaño del vocabulario es $|V|$, el modelo de lenguaje se expresa como:

$$\begin{aligned} P(w_1, w_2, w_3, \cdots, w_{|V|}) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)\cdots P(w_{|V|}|w_1^{|V|-1}) \\ &= \prod_{i=1}^{|V|} P(w_i|w_1^{(i-1)}) \end{aligned}$$

La distribución conjunta se descompone en *n-gramas*; el término $P(w_i)$ es un unigrama. Los términos condicionales como $P(w_i|w_j)$ se llaman bigramas, $P(w_i|w_j, w_{j-1})$ trigramas, etc.

La idea principal de Bengio et al. (2003), era aproximar la distribución conjunta mediante el cálculo de n-gramas, donde n es un parámetro de contexto (ventana). Dado un *contexto* $:= (w_{i-1}, w_{i-2}, \cdots, w_{i-m})$, el modelo proporciona las probabilidades de cada palabra en V de modo que la palabra correcta w_i maximiza $P(w_i|contexto)$. Una simplificación muy utilizada de la distribución conjunta es el modelo *Bag-Of-Words* (BOW), que consiste en usar solo unigramas. Aunque conceptualmente sólido, el enfoque de Bengio et al. (2003) fue computacionalmente ineficiente. En los últimos años, (Mikolov et al., 2013) propusieron el modelo word2vec que atrajo una gran cantidad de atención.

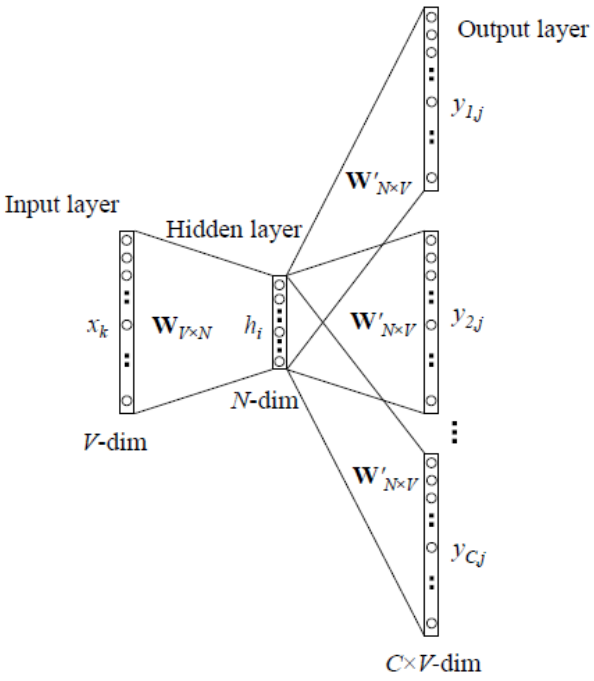

Figura 2.2: Modelo Skip-gram

Básicamente, hay dos modelos que han sido una opción estándar entre la comunidad de lingüística computacional. El modelo continuo de bolsa de palabras (CBOW) se muestra en la Figura 2.1. La imagen muestra un modelo de unigrama, donde la red recibe un único vector, el contexto de entrada, y produce un vector de probabilidades utilizando una capa jerárquica softmax (Morin and Bengio, 2005). Una capa softmax calcula una distribución multinomial, donde la probabilidad de cada palabra del vocabulario se calcula como:

$$P(w_j|w_i) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})}$$

donde u_j es el valor de la función de activación de la j-ésima unidad en la capa de salida. Se puede ver fácilmente que a medida que aumenta el tamaño del vocabulario, la complejidad de la capa de salida aumenta proporcionalmente. Softmax jerárquico (Morin and Bengio, 2005) implementa la idea de agrupar palabras en clases, de modo que la probabilidad de salida consistiría primero en estimar la clase, y luego, dada la clase, calcular la probabilidad de la palabra de salida en esa clase. Este esquema mejora drásticamente el rendimiento al reducir la complejidad.

El otro modelo es SKIP-gram que se muestra en la Figura 2.2. Es lo opuesto al modelo CBOW. La palabra objetivo ahora está en la capa de entrada,

y las palabras de contexto están en la capa de salida: su objetivo es calcular $P(context|w_i)$. La ventaja de este modelo es que diferentes contextos producen distintas representaciones. Algunos autores han aprovechado esta propiedad para producir WE de calidad variada, lo que ha llevado a mejores clasificadores en diversas tareas de PLN.

2.4 Representación de Oraciones

Con las propuestas mencionadas anteriormente logramos obtener una representación embebida de cada una de las palabras del vocabulario, sin embargo en muchas tareas de PLN, incluida la de generación de resúmenes es muy común trabajar a nivel de oraciones. La representación de oraciones es también una línea de investigación dentro del área. Una de las representaciones más sencillas pero que ha logrado resultados aceptables es la suma o promedio de los vectores de palabras de una oración, con los que se obtienen vectores de oraciones de la misma dimensionalidad que los de palabras. Muchas otras representaciones se han propuesto, sobretodo basadas en redes neuronales.

Paragraph Vector o Doc2Vec: Le and Mikolov (2014) proveen un método no supervisado basado en el modelo de word2vec que predice vectores de bloques de textos (oraciones, párrafos o documentos) prediciendo las palabras que estos contienen. Para muchas tareas como similitud/ analogía a nivel de documentos y en agrupación de documentos este método provee una representación eficiente, sin embargo en tareas de similitud de oraciones, existen otros métodos con mejores resulatdos.

Las redes auto-encoders son una propuesta intuitiva hacia las representaciones semánticas de oraciones. Estas redes son entrenadas de forma no supervisada para obtener la oración de entrada como un vector de salida. En Hill et al. (2016) proponen SDAE (sequential denoising auto-encoder), una red neuronal recurrente para comprimir una secuencia corrupta de palabras (es decir con cierto ruido) con respecto al orden y poder reproducir la oración original. También se han propuesto redes neuronales recurrentes basadas en la secuencia, igual con una arquitectura de auto-encoder. Socher et al. (2011) propusieron RAE(unfolding recursive auto-encoder), que es guiada por un árbol sintáctico binario de la oración. Recursivamente combina vectores de palabras para finalmente codificar la oración completa. Luego decodifica el vector de oración usando el mismo árbol.

Otro modelo encoder-decoder usado para producir oraciones embebidas es Skip Thought model (Kiros et al., 2015). Este modelo codifica las oraciones usando RNN y es entrenado para predecir las palabras en las oraciones vecinas. Usan la hipótesis distribucional a nivel de oración, construyen una distribución sobre las palabras en las oraciones circundantes dada la oración actual. Como el codificador y decodificador se basan en una red LSTM con un número relativamente alto de unidades ocultas, el entrenamiento y la inferencia son lentos.

Varios trabajos de resúmenes han aprovechado las ventajas de las anteriores propuestas de representación de palabras y oraciones. A continuación se describen algunos de ellos.

Los autores de Kågebäck et al. (2014) se basan en la propuesta para generación de resúmenes de Lin and Bilmes (2011), quienes formularon la tarea como un problema de optimización utilizando un conjunto de funciones submodulares monótonas no decrecientes. Una función submodular F en el conjunto de oraciones V debe satisfacer la siguiente propiedad, que para cualquier $A \subseteq B \subseteq V$ $v, F(A + v - F(A)) \geq F(B + v) - F(B)$ donde $v \in V$. Esta propiedad refiere intuitivamente que adicionar una oración a un conjunto pequeño de oraciones hace una mayor contribución que añadirla a un conjunto grande de oraciones. El objetivo es encontrar un resumen que maximice la diversidad de las oraciones y la cobertura del texto de entrada. La función objetivo es formulada como:

$$F(S) = L(S) + \lambda R(S) \tag{2.2}$$

donde S es el resumen, $L(S)$ la cobertura, $R(S)$ una función de recompensa de la diversidad y λ un coeficiente que permite definir la importancia de la cobertura sobre la diversidad del resumen. Este tipo de optimización es un problema NP-hard, sin embargo si la función es submodular existe un algortimo escalable y rápido que devuelve una aproximación con una garantía. En Lin and Bilmes (2011) utilizaron la siguiente función para el cálculo de la cobertura:

$$L(S) = \sum_{i \in V} \min \sum_{j \in S} Sim(i, j), \alpha \sum_{j \in V} Sim(i, j) \tag{2.3}$$

La similitud entre las oraciones se calcularon por la medida del coseno, y la representación de las oraciones usando $tf - idf$ (Salton and McGill, 1986). En Kågebäck et al. (2014) proponen varios experimentos cambiando la representación de oraciones sobre la propuesta original de Lin and Bilmes (2011). Las representaciones utilizadas fueron:

- Adición de vectores de palabras representadas con Word2Vec
- Adición de vectores de palabras representadas con el método de Collobert y Weston Collobert and Weston (2008)
- Representación de oraciones aprendidas utilizando una red recursiva de autoencoder Kågebäck et al. (2014)

Realizaron varias combinaciones y compararon contra el método original con la representación con $tf - idf$. Para las pruebas utilizaron el corpus de Opinosis Ganesan et al. (2010), que contiene reseñas cortas de usuarios en 51 tópicos diferentes(ejemplo: características de un hotel,de un carro, un producto, etc), cada tópico puede tener alrededor de 500 oraciones o menos. El corpus puede ser usado para la tarea de resúmenes porque cada oración se considera un documento y cuenta con 4 o 5 resúmenes de referencia hechos por humanos para cada tópico. Para la evaluación se utilizó el estandar de ROUGE con las medidas ROUGE-1, ROUGE-2 y ROUGE-SU4. Los resultados arrojaron que las representaciones embebidas se comportaron mejor que la representación con $tf - idf$, los mejores valores de recall fueron en la adición de palabras con

la representación de Collobert y Weston usando medida del coseno, y para la precisión y la medida promedio también ganó la combinación anterior pero con la medida euclideana.

En Khosla (2015) realizan resúmenes de opiniones para diferentes negocios, utilizando un conjunto de datos de Yelp. Exploran 3 tipos de representaciones: estadística con el modelo de Bolsa de Palabras, el modelo de Word2Vec y el modelo de vector de párrafo (Doc2Vec) para tratar de aprender representaciones vectoriales de oraciones; estas representaciones aprendidas se usan luego para agrupar y extraer oraciones relevantes del superconjunto utilizando el agrupamiento k-means. El objetivo es tener una mayor valor de precisión por lo que deciden entrenar los modelos en función de cada negocio y no sobre el corpus completo. Para la representación de las opiniones, que son las oraciones, utilizan la suma de los vectores de palabras y para seleccionar las oraciones que conformarán el resumen aplican el método de agrupamiento k-means y obtienen la oración más cercana de cada cluster. Para los experimentos, prueban con diferentes dimensiones, número de clusters, medidas de distacia(euclideana y coseno), y épocas de entrenamiento. Crearon una medida propia para evaluar su propuesta ya que el corpus de Yelp no cuenta con etiquetado para ello. Un humano seleccionó las oraciones clave que le gustaría que se incluyeran en un resumen de las vistas de un negocio. Luego, las oraciones generadas por el sistema se comparan con estas oraciones clave, y la puntuación de precisión es el número de oraciones verdaderamente claves, según lo escogido por el ser humano, dividido por el número total de oraciones clave devueltas por el algoritmo . La optimización de esta métrica se corresponde con el concepto de precisión, ya que revela el porcentaje de frases devueltas que son relevantes para el resumen general. El modelo que obtuvo mejores valores de precisión fue Doc2Vec alcanzando un máximo de 58%, el modelo Word2Vec con la suma de vectores tuvo mejor precisión que el de bolsa de palabras, pero no con una diferencia muy marcada.

En Templeton and Kalita (2018) utilizan funciones vectoriales y funciones de selector. Las funciones vectoriales son tipos de representaciones que utilizan para hacer las pruebas y compararlas. Para ello utilizan las siguientes variantes:

- SIF: Es un promedio de vectores ponderado propuesto por Arora et al. (2016)
- Arora: Es un método equivalente al anterior, solo que añaden el paso de eliminar componentes comunes (Arora et al., 2016)
- Paragraph Vectors: método propuesto por Mikolov (Le and Mikolov, 2014)
- Skip-Thought Vectors (Kiros et al., 2015)

Las funciones de selección que utilizaron fueron:

- Random: Se seleccionan las oraciones de forma aleatoria hasta que cumpla con el número de palabras límite.

- Near: Selecciona las oraciones que tienen mayor similitud (según la medida del coseno) con el vector del documento.
- Cluster: Ejecutan un algoritmo de agrupamiento aglomerado y seleccionan la oración más cercana al promedio de cada grupo para agregarla al resumen.
- Greedy: El selector codicioso, en cada paso, selecciona la oración que maximiza la similitud de coseno del nuevo resumen (incluidas las oraciones previamente seleccionadas). Esto es sutilmente diferente del selector Near para funciones de vector basadas en el promedio, pero significativamente diferente para los vectores de párrafo.
- PCA: Este selector realiza análisis de componentes principales en el conjunto de vectores de oraciones en un documento. Luego el algoritmo selecciona una oración cercana al primer componente, una segunda cercana al segundo componente y así hasta llegar al limite de longitud.
- LexRank: Se basa en el algoritmo clásico de LexRank (Erkan and Radev, 2004). Luego utilizan PageRank (Page et al., 1999) para identificar las oraciones más importantes. Los pesos entre los vértices los determinan usando la similitud del coseno entre los vectores embebidos que representan las oraciones.

Para las pruebas combinaron todas las funciones vectoriales con los selectores. Utilizaron los documentos de la colección DUC, una muestra para el conjunto de entrenamiento y otra para el conjunto de prueba. El análisis experimental arrojó que el selector de mejor rendimiento fue *Greedy*. Los vectores de párrafo (Paragraph Vectors) funcionan mucho peor con los algoritmos de *Clustering* y *Greedy*, y funcionan mucho mejor con *Near*. Muchas combinaciones de la función selectora y la función vectorial no funcionan por encima del nivel de posibilidad aleatoria. En general, a pesar de su sofisticación, los vectores de párrafos y los vectores de *Skip-Thought* se comportan peor que muchos de los enfoques más básicos.

2.4.1 Oraciones como sub-espacios de bajo rango

Por último se hará referencia a la propuesta presentada por (Mu et al., 2017), donde explican un enfoque para la representación de oraciones usando PCA y una propiedad que definen como geometría de oraciones. Su propuesta se apoya en los trabajos presentados en (Gong et al., 2017) y en (Mu et al., 2016), este último de los propios autores, donde se observa un fenómeno interesante y es que el contexto local de una palabra/frase dada puede ser bien representada por un subespacio de bajo rango. Proponen que se genere esta misma observación a oraciones: no solo los vectores de palabras en un fragmento de una oración se encuentran en un subespacio de bajo rango, sino que la oración completa también sigue esta propiedad geométrica. Las observaciones indican que el subespacio contiene la mayor parte de la información sobre una oración y, por lo tanto

se puede definir un método de representación de una oración en un espacio de subespacios, en vez de en un espacio vectorial. Formalmente definen que una oración puede representarse por un sub-espacio de bajo rango que abarca sus representaciones de palabras. Los primeros experimentos que realizan son para demostrar que la representación propuesta sí capta la mayor parte de la información contenida. Partiendo de la representación vectorial embebida de cada palabra con Word2Vec y con Glove, plantean:
Sea $v(w) \in \mathbb{R}^d$ un vector d-dimensional de una palabra, tal que $w \in V$ y $s = (w_1, \cdots, w_n)$ una oración dada, donde n es el número de palabras, se observa que la matriz $d \times n$ captura la mayor parte de la energía (80% para GloVe y 72% con word2vec) en un subespacio de rango N que es menor que n. La experimentación se relaizó partiendo desde N=4. En la Figura 2.3 se puede ver este concepto de geometría, donde llevaron las representaciones a dos dimensiones para poderlas graficar.

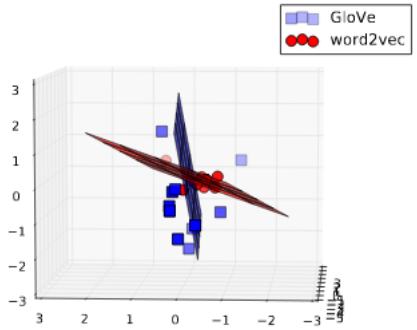

Figura 2.3: Geometría de Oraciones. Extraído de (Mu et al., 2017)

Con la demostración anterior sentaron las bases para justificar la representación propuesta. Una vez conformada la matriz se realiza la reducción del rango aplicando Análisis de Componentes Principales. En la figura 2.4 se muestra el algoritmo extraído del artículo.

Algorithm 1: The algorithm for sentence representations.

Input : a sentence s, word embeddings $v(\cdot)$, and a PCA rank N.

Compute the first N principle components of samples $v(w'), w' \in c$,

$$u_1, ..., u_N \leftarrow \text{PCA}(v(w'), w' \in s),$$

$$S \leftarrow \left\{ \sum_{n=1}^{N} : \alpha_n u_n, \alpha_n \in R \right\}$$

Output: N orthonormal basis $u_1, ..., u_N$ and a subspace S.

Figura 2.4: Algoritmo "Low-Rank SubSpace". Extraído de (Mu et al., 2017)

Una vez que se aplica el algoritmo se contará con una representación de las oraciones como subespacios de bajo rango, sin embargo, para las tareas en las cuales se aplicarán, es necesario tener en cuenta cómo se medirá la similitud entre oraciones. Los autores proponen una fórmula para el cálculo de la similitud motivados por la medida del coseno, generalizando esta métrica a subespacios. La similitud entre los subespacios es la norma l_2 de los valores singulares entre los dos subespacios, ya que los valores singulares representan el coseno de los ángulos principales, véase la fórmula (2.4)

$$CosSim_{(s1,s2)} = \sqrt{\sum_{t=1}^{N} \sigma t^2} \tag{2.4}$$

siendo $u_1(s), \cdots, u_N(s)$ la base ortonormal de tamaño N de una oración s, y
$U(s) = u_1(s), \cdots, u_N(s)$ la matriz de dimensiones $d \times N$ obtenida del algoritmo propuesto.

Realizaron las pruebas en la tarea de Similitud de Oraciones con 19 corpus diferentes de SemEval[3] y compararon la propuesta contra el promedio de vectores y otras representaciones basadas en redes neuronales como Doc2Vec, skipthoughts y Glove. Demostraron que esta representación mejora el método del promedio de vectores en un 14% y el método por redes neuronales(Doc2Vec y Glove) en un 15%

[3]Competición para la evaluación de sistemas de análisis semántico, incluye tareas como similitud de oraciones, análisis de sentimientos, entre otras

2.5 Conclusiones

Tal como se describió en el capítulo, existen diferentes propuestas para la representación del texto. Las propuestas basadas en grafos, proveen una representación muy intuitiva y las relaciones entre los nodos brindan un nivel de asociación entre palabras u oraciones que son útiles para determinar las oraciones más relevantes a la hora de conformar el resumen, sin embargo dicha representación está sujeta solamente al texto con el que se esté trabajando y para obtener términos u oraciones semánticamente similares que no aparezcan en los documentos se necesita del uso de recursos linguísticos como por ejemplo WordNet. Por otra parte la construcción de los grafos puede ser computacionalmente costosa.

Las representaciones vectoriales permiten la aplicación de un mayor número de algoritmos de aprendizaje automático en PLN. La representación clásica de Bolsa de Palabras representa un espacio vectorial caracterizado por el número de ocurrencias de cada palabra del vocabulario. Es un método fácil de aplicar debido a su simplicidad, sin embargo solo modelan un enfoque frecuentista, que no tiene en cuenta el orden de las palabras o frases, ni la semántica de las mismas.

La representación latente de texto constituye un paso importante en el intento de modelar las distribuciones entre palabras, con el fin de encontrar relaciones semánticas entre las mismas. Una de las desventajas principales de este método es el alto costo computacional (en espacio y tiempo de procesamiento), dependiendo de la dimensionalidad de la matriz que se necesite conformar, sin embargo métodos como PCA permiten reducir dicha dimensionalidad.

La aplicación de las redes neuronales en el aprendizaje de palabras para su representación como vectores embebidos ha sido de gran utilidad en el área y los trabajos muestran que dichas redes logran encontrar relaciones semánticas y linguísticas cuando entrenan sobre un amplio corpus. La desventaja radica justamente en el alto volumen de información que se necesita para su aprendizaje y los recursos computacionales.

Varios métodos se han propuesto también para la representación de oraciones como vectores embebidos, sin embargo no se ha llegado a ningún consenso en la comunidad científica, sobre algún método significativamente superior a otro. Dicha valoración se hace en dependencia de los experimentos que se llevan a cabo para una tarea y corpus específicos. Aún cuando pudiera pensarse que una representación aprendida a partir del uso de redes neuronales debe ser superior a un simple promedio de vectores de palabras, hay trabajos donde este último método ha dado mejores resultados.

Capítulo 3

El problema de la Optimización Multiobjetivo

En el presente capítulo se describe en qué consiste la optimización multiobjetivo, así como los principales conceptos que se deben conocer para dar solución a problemas de este tipo. Se explican los algoritmos evolutivos, especialmente los algoritmos genéticos, ya que constituyen un tipo de solución efectiva dentro de la optimización, y por último se detallan algunos trabajos que han afrontado la tarea de resúmenes desde el enfoque de optimización.

3.1 Optimización

La "optimización" es una técnica que juega un papel fundamental en la Inteligencia Artificial. Se pudiera plantear que muchos de los problemas de aprendizaje automático son en el fondo problemas de optimización. Así por ejemplo los algoritmos de clasificación, agrupación y regresión intentan resolver un tipo de problema de optimización denominado ajuste de datos y los algoritmos de redes neuronales se basan en métodos de optimización como el gradiente descendiente. En las últimas décadas, se introdujeron diferentes enfoques en los problemas de optimización para encontrar las mejores y más satisfactorias soluciones. Básicamente se trata de resolver problemas de búsqueda de soluciones en espacios de estado cuya topología se desconoce y son generalmente muy grandes. El avance de la programación y desarrollo de la tecnología, han permitido el desarrollo de propuestas más sofisticadas para afrontar la tarea de resúmenes, como la utilización de algoritmos de aprendizaje de máquina (Dunlavy et al., 2007) (Wong et al., 2008)(Bollegala et al., 2010).

3.1.1 Clasificación

De manera general se pueden clasificar las técnicas de optimización en **Exactas** y **Aproximadas**. Las técnicas **exactas** son aquellas que garantizan encontrar

la solución óptima para cualquier problema en un tiempo determinado y se dividen en técnicas basadas en Cálculo y técnicas enumerativas. La mayoría de los problemas que pueden resolver son de tipo NP-hard de dificultad no polinómica, por lo que el tiempo de resolución y la memoria computacional necesaria crecen exponencialmente, aún cuando estén acotados (Gary and Johnson, 1979). Al no poder encontrar una solución en un tiempo razonable en muchos casos el uso de técnicas exactas es inviable. Por lo tanto, las técnicas aproximadas para resolver estos problemas están recibiendo una atención cada vez mayor por parte de la comunidad científica en los últimos tiempos.

Por otra parte los métodos aproximados (algoritmos de aproximación o heurísticos) obtienen una solución a los problemas en un tiempo razonable sin ofrecer garantías de que la solución encontrada sea la óptima.

En el campo de la optimización matemática se define como técnica heurística de resolución de problemas aquella que no garantiza que la solución que encuentren sea óptima. Así la definición recogida por (Zanakis and Evans, 1981) dice que los algoritmos heurísticos son: procedimientos simples, a menudo basados en el sentido común, que se supone ofrecerán una buena solución (aunque no necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido.

Los algoritmos heurísticos no garantizan obtener una solución óptima, pero ofrecen en un tiempo razonable de ejecución una buena solución al problema planteado (Cunquero, 2003). Otra definición formal de los métodos heurísticos fue la enunciada por Lieberman en (Hillier et al., 1997) donde los define como *"...un procedimiento que trata de describir una solución factible muy buena, pero no necesariamente una solución óptima, para el problema específico bajo consideración".*

En algunas ocasiones no es necesario desarrollar heurísticas de propósito específico para enfrentar los problemas, por lo que surgen las metaheurísticas, que brindan un enfoque flexible para enfrentar problemas complejos. Los algoritmos metaheurísticos deben su nombre a Fred Glover quien lo define en 1986 (Glover, 1986): *" Una metaheurística es referida a la estrategia maestra que guía y modifica otras heurísticas para producir soluciones más allá de aquellas que son normalmente generadas en la búsqueda de un óptimo local. Las heurísticas guiadas por una estrategia meta pueden ser un procedimiento de alto nivel o pueden contener una descripción de movimientos permitidos para trasformar una solución en otra, conjuntamente con una regla asociada de evaluación. "*

Hillier et al. (1997) definen las metaheurísticas como "...un método de solución general que proporciona tanto una estructura general como criterios estratégicos para desarrollar un método heurístico específico que se ajuste a un tipo particular de problema". El prefijo meta se utiliza para indicar que estos algoritmos solo especifican la estrategia general para guiar aspectos específicos de la búsqueda pero no especifican todos los detalles de la búsqueda, lo cual le permiten ser adaptados por una heurística local a una aplicación específica.

El término heurístico deriva de la palabra griega *heuriske* que significa encontrar o descubrir y se usa en el ámbito de la optimización para describir una clase de algoritmos de resolución de problemas (Cunquero, 2003). La tendencia de las metaheurísticas para solucionar los problemas es comenzar por obtener

una solución o conjunto de soluciones e iniciar una búsqueda de mejoramiento guiada por ciertos principios.

Las metaheurísticas se pueden clasificar en métodos de búsqueda basados en trayectoria y métodos de búsqueda basados en población (Gendreau et al., 2010).

Los métodos de búsqueda basados en trayectoria son aquellos que partiendo de un punto, buscan la vecindad y actualizan la solución actual en función de esta, formando una trayectoria, de punto a punto. Entre estos algoritmos se encuentran: Escalador de Colinas, Recocido Simulado (Kirkpatrick et al., 1983), Búsqueda Tabú (Glover, 1986), GRASP (Feo and Resende, 1995), entre otros.

Los algoritmos basados en poblaciones, a diferencia de los basados en trayectoria, trabajan con un conjunto de soluciones en cada iteración, y su resultado está determinado por la forma en que se manipula la población. Entre estos algoritmos se encuentran: los algoritmos evolutivos (Coello et al., 2007) y los sistemas de enjambre de partículas (Cagnina et al., 2005), entre otros.

3.2 Optimización MultiObjetivo

La aplicación de la optimización multiobjetivo a dominios externos a la economía comenzó con el trabajo de (Koopmans, 1951) en teoría de la producción. La primera aplicación de ingeniería reportada en la literatura fue un artículo a principios de los años sesenta (Zadeh, 1963). Sin embargo, el uso de la optimización multiobjetivo se generalizó hasta la década de los 70 (Coello et al., 2007).

Un problema de optimización multiobjetivo (POM) es aquel que incluye un conjunto de funciones objetivo a optimizar (dos o más funciones). El objetivo de este tipo de problema es encontrar los parámetros necesarios que optimicen el vector de funciones objetivo y satisfagan las restricciones (Coello et al., 2007) Un problema de optimización multiobjetivo es definido como la minimización (o maximización) de:

$$F(x) = (f_1(x), \cdots, f_k(x)), i = 1, .., m \tag{3.1}$$

sujeto a

$$g_i(x) \leq 0 \tag{3.2}$$

y

$$h_j(x) = 0, j = 1, \ldots, p, x \in \Omega; |\Omega| = n \tag{3.3}$$

Este tipo de problema consiste en k objetivos reflejados en las k funciones objetivo, $m + p$ restricciones de las funciones objetivos y n variables de decisión. El objetivo que se persigue es encontrar el vector $x = (x_1, \ldots, x_n)$ que satisfaga las restricciones del problema y optimice la función vectorial F(x).

Los problemas de optimización multiobjetivo no obtienen como resultado de la búsqueda una única solución, sino un conjunto de soluciones, por lo que requieren del decisor para elegir una de las soluciones del conjunto. El concepto

de óptimo cambia, ya que lo que se pretende es encontrar buenos compromisos entre los diferentes objetivos. A continuación se definen algunos conceptos necesarios asumiendo un problema de minimización:

- Óptimo de Pareto: Una solución $x \in \Omega$ es llamada óptimo de Pareto con respecto a Ω si y solo si, no existe $x' \in \Omega$ para el cual $v = F(x') = f_1(x'), \cdots, f_2(x')$ domina a $u = F(x) = f_1(x), \cdots, f_2(x)$. Es decir x' es un óptimo de Pareto si no existe ningún vector factible x que disminuya algún criterio sin causar un aumento simultáneo en al menos uno de los otros criterios.

- Dominancia de Pareto: Un vector $u = (u_1, \ldots, u_k)$ se dice que domina a otro vector $v = (v_1, \ldots, v_k)$ si y solo si u es parcialmente menor que v (denotado por $u \preceq v$). En otras palabras, x domina a y si x es mejor que y en al menos una de las funciones objetivos y no es peor en ninguna de las restantes.

- Conjunto óptimo de Pareto: Para un problema de optimización multiobjetivo dado $F(x)$, el conjunto óptimo de Pareto P^* , es definido como

$$P^* = \{x \in \Omega \mid \neg\exists x' \in \Omega \quad F(x') \preceq F(x)\} \tag{3.4}$$

- Frente de Pareto: Para un problema de optimización multiobjetivo dado, F(x), y el conjunto óptimo de Pareto, $P*$, el frente de Pareto PF^* es definido como

$$PF^* = \{u = F(x) \mid x \in P^*\} \tag{3.5}$$

En los problemas de optimización multiobjetivo lo que se pretende es encontrar el conjunto de soluciones no dominadas del espacio de soluciones visitadas. Un ejemplo de esto se muestra en la Figura 3.1. Asumiendo un problema de optimización multiobjetivo que propone maximizar dos funciones objetivo f1 y f2, en la Figura 3.1 se muestran tres puntos (X,Y,Z) que son todas las soluciones posibles de las funciones a maximizar. Como se observa, X y Y son soluciones no dominadas entre sí y Z está dominada por X, por lo que X y Y son parte de las soluciones óptimas de Pareto (soluciones no dominadas).

En los problemas de optimización multiobjetivo existen diversas maneras de representar las preferencias del decisor. Algunas de las técnicas más simples proponen transformar el problema multiobjetivo en un problema escalar, ponderando las funciones objetivo, como es el caso del método combinación lineal de pesos. Otros proponen, tratar el problema como monobjetivo seleccionando solo una de las funciones objetivos y tratando las restantes como restricciones o considerar varios objetivos simultáneamente.

Combinación lineal de pesos: La combinación lineal de pesos, también conocido como método de factores ponderados (Caballero and Hernández, 2003) se basa en la idea de convertir el problema multiobjetivo en un problema escalar, construyendo una única función objetivo que sea la suma de las funciones objetivos iniciales, ponderadas según un peso que se le asigna a cada una de

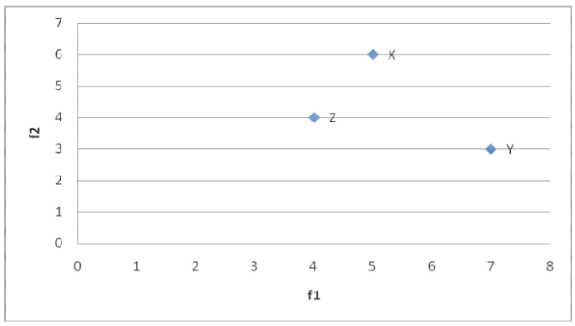

Figura 3.1: Dominancia asumiendo maximización.

ellas. De aquí que para cada ponderación posible, se obtenga un problema escalar consistente en minimizar o maximizar la función resultante, sujeta a las restricciones del problema original. El problema se plantea como sigue (Sunar and Kahraman, 2001):

$$\min(\sum_{i=1}^{n} w_i f_i(x)), \sum_{i=1}^{n} w_i = 1$$

Donde w_i es una constante que indica el peso asignado a f_i. El principal inconveniente de esta técnica es el hecho de asignarle valores a los pesos, lo cual puede resultar difícil para el decisor. Además, pueden obtenerse iguales soluciones para diferentes combinaciones de pesos. Por otra parte esta técnica no refleja la realidad multiobjetivo del problema.

Ordenamiento Lexicográfico: En la técnica de ordenamiento lexicográfico, el decisor asigna a cada objetivo una prioridad según la importancia del mismo (de mejor a peor). La solución óptima es obtenida con la minimización (o maximización) de la función objetivo de mayor prioridad, y luego de optimizar este valor obtenido para la función inicial, se convierte en una restricción del problema y se pasa a optimizar la próxima función objetivo. Esto se realiza de manera iterativa y de acuerdo al orden de importancia de los objetivos. La representación del método es la siguiente. Se minimiza la función objetivo de mejor prioridad: $\min f_1(x)$ sujeto a $g_j(x) \leq 0; \quad j = 1, 2, \cdots, m$ Y se obtiene la solución: x_1^* y $f_1^* = f(x_1^*)$.

Entonces el segundo problema es: $\min f_2(x)$ sujeto a $g_j(x) \leq 0; \quad j = 1, 2, \cdots, m$ y $f_1(x) \geq f_1^*$ Y se obtiene la solución: x_2^* y $f_2^* = f(x_2^*)$ Y así sucesivamente hasta que los k objetivos han sido considerados. Esta técnica presenta el inconveniente de cómo definir las prioridades de los objetivos.

MultiObjetivo Puro: Otra técnica de solución a este tipo de problema es tratarlo como multiobjetivo puro, y para ello debe encontrarse el conjunto de soluciones óptimas de Pareto ((Coello et al., 2007). Se dice que una solución óptima de Pareto es aquella tal que no existe ninguna otra solución alcanzable que la domine. Esta solución óptima generalmente produce más de una solución,

conocidas como soluciones no dominadas. Esta técnica resulta conveniente en tanto no necesita que el decisor asigne pesos o prioridades a los objetivos a diferencia de las técnicas anteriores y modela de forma más natural la realidad del problema.

La complejidad de algunos problemas de optimización multiobjetivo han hecho inviables el uso de las soluciones tradicionales. En este sentido el uso de algoritmos evolutivos (EA) ha sido una óptima solución, ya que su propia naturaleza, basada en poblaciones, permiten la generación de varios elementos del conjunto óptimo de Pareto en una sola ejecución. En las siguientes secciones se describen los algoritmo evolutivos.

3.3 Algoritmos Evolutivos

El próposito principal de los algoritmos evolutivos consiste en resolver problemas donde los algoritmos deterministas son demasiado costosos. Los problemas de optimización multiobjetivo pueden afrontar retos como espacios de búsqueda muy grandes, incertidumbre, ruido, curvas de Pareto disjuntas, etc, que sólo con el uso de algoritmos evolutivos es posible resolver.

En la década de 1960 surgieron las primeras ideas del uso de los algoritmos evolutivos en la optimización multiobjetivo, siendo uno de los pioneros Rosenberg con su tesis doctoral en 1967, pero no fue hasta la década de 1980 en que se realizó la primera implementación con la propuesta de (Schaffer, 1985). En los años posteriores el campo se mantuvo inactivo hasta que comenzó a crecer en los años 90, en que se desarrollaron varias aplicaciones y técnicas.

Existen varias técnicas de programación matemática para la optimización multiobjetivo, sin embargo la mayoría son muy sensibles a la forma del frente de Pareto y tienden a generar el conjunto óptimo de Pareto de uno en uno.

Los Algoritmos Evolutivos (Evolutionary Algorithms - EAs) han demostrado ser adecuados para la optimización multiobjetivo. La literatura actual reporta un gran número de Algoritmos Evolutivos que han sido utilizados en la resolución de problemas de múltiples objetivos.

Pueden considerarse, en general, dos tipos principales de algoritmos evolutivos multiobjetivos:

1. Los algoritmos que no incorporan el concepto de óptimo de Pareto en el mecanismo de selección del algoritmo evolutivo (p.ej., los que usan funciones agregativas lineales).

2. Los algoritmos que jerarquizan a la población de acuerdo a si un individuo es dominado o no. Ejemplos: MOGA, NSGA, NPGA, etc.

La Computación Evolutiva se refiere al estudio de los fundamentos y aplicaciones de ciertas técnicas heurísticas de búsqueda basadas en los principios naturales de la evolución. Una gran variedad de algoritmos evolutivos han sido propuestos pero principalmente pueden clasificarse en: Algoritmos Genéticos,

Programación Evolutiva, Estrategias Evolutivas, Sistemas Clasificadores y Programación Genética. Esta clasificación se basa sobre todo en detalles de desarrollo histórico más que en el hecho de un funcionamiento realmente diferente; de hecho las bases biológicas en las que se apoyan son esencialmente las mismas.

3.3.1 Algoritmos Genéticos

El algoritmo genético es una técnica de búsqueda basada en la teoría de la evolución de Darwin, que ha cobrado una gran popularidad en todo el mundo durante los últimos años (Koza, 1997). Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Hoy en día se sabe que estos cambios se efectúan en los genes de un individuo (unidad básica de codificación de cada uno de los atributos de un ser vivo), y que sus atributos más deseables (i.e., los que le permiten adaptarse mejor a su entorno) se transmiten a sus descendientes cuando éste se reproduce sexualmente.

Una definición bastante completa de un algoritmo genético es la propuesta por John Koza:

Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

Algunas de las ventajas de estos algortimos son:

Los algoritmos genéticos son intrínsecamente paralelos. La mayoría de los algoritmos de optimización que son en serie sólo pueden explorar el espacio de soluciones en una dirección al mismo tiempo. Y si la solución a la que convergen es un extremo local, no pueden hacer otra cosa que abandonar la búsqueda y empezar de nuevo. Por otro lado, ya que los algoritmos genéticos tienen descendencia múltiple, pueden explorar múltiples direcciones a la vez.

Otra de las ventajas es que las técnicas evolutivas no necesitan conocimientos específicos sobre el problema que intentan resolver. Así, al depender únicamente los algoritmos genéticos de la función objetivo la convergencia de la solución óptima no está vinculada a requisitos de continuidad, diferenciabilidad, convexidad, lo que proporciona a los algoritmos genéticos una mejora sustancial con respecto a los métodos clásicos.

Los algoritmos genéticos son realmente efectivos en la resolución de problemas no lineales, cuyo espacio de soluciones potenciales es realmente grande, demasiado vasto para hacer una búsqueda exhaustiva en un tiempo razonable.

A pesar de sus ventajas, también presentan un serie de limitaciones, tales como:

Debido a que desarrollan varios caminos en paralelo a la vez, son demasiado lentos en encontrar un extremo local una vez que están en sus proximidades, siendo frecuente tener que utilizar técnicas híbridas que con técnicas de optimización local, producen una mejora considerable en la búsqueda de la solución y en la disminución de los tiempos de cálculo.

También puede alcanzar lo que se conoce como convergencia prematura. Si un individuo que es más apto que la mayoría de sus competidores emerge muy pronto en el curso de la ejecución, se puede reproducir tan abundantemente que disminuya la diversidad de la población demasiado pronto, provocando que el algoritmo converja hacia el óptimo local que representa ese individuo, en lugar de rastrear el espacio de búsqueda lo suficiente para encontrar el óptimo global.

Sin embargo, los problemas antes mencionados se pueden superar por distintas vías y ninguno afecta a la validez y al desarrollo de la evolución artificial. Así, las ventajas de los algoritmos genéticos en múltiples y diversos problemas superan con creces sus desventajas, lo que ha impulsado su enorme desarrollo en las últimas décadas.

Las principales fases que debe seguir un algoritmo genético de manera general son:

1. Codificar la información del problema: Se debe elegir una forma de representación de las posibles soluciones del problema, con la cual el algoritmo genético pueda trabajar. En la siguiente subsección se explican las variantes existentes.

2. Generar aleatoriamente la población inicial: La población es el conjunto de posibles soluciones al problema, cada solución representa un cromosoma. Una vez que se determina cómo se codificará la información se necesita partir de un conjunto inicial de soluciones que típicamente se inicializa de forma aleatoria, aunque pueden existir casos en que se tenga un conocimiento previo de cuales sean los valores iniciales de dónde partir, con lo cual el algoritmo pudiera encontrar una mejor solución en un tiempo más corto.

3. Evaluación de la población: Para evaluar a cada individuo, previamente se debe haber definido la función de aptitud(fitness). Esta función es aplicada a cada solución en aras de conocer cuales son más aptas para que de ellas se puedan generar nuevas soluciones (individuos)

4. Selección de la nueva población: En esta fase se seleccionan los individuos más aptos por medio de algún método de selección. Algunos de estos métodos se explican en 3.3.1.3

5. Generación de la nueva población: Una vez definidos los individuos más aptos se realiza la reproducción, que no es más que aplicar operadores genéticos (cruzamiento y mutación) sobre la misma para obtener nuevos descendientes.

6. Se repiten los pasos del 3 al 5 hasta que se cumplan las condiciones de parada del algoritmo, y se elige la mejor solución.

3.3.1.1 Métodos de Representación de soluciones

Existen varias formas para realizar la representación de las posibles soluciones en los algoritmos genéticos, y así poder aplicar las operaciones que les permiten evolucionar. Cada posible solución hace analogía a un cromosoma, que a su vez está conformado por genes.

Una de las variantes más comunes es codificar las soluciones como cadenas binarias: secuencias de 1's y 0's, donde el dígito de cada posición(gen) representa el valor de algún aspecto de la solución.

Otro método consiste en codificar las soluciones como cadenas de enteros o números decimales, donde cada posición, igual, representa algún aspecto particular de la solución. Un tercer método consiste en representar a los individuos como cadenas de letras. Por último es posible realizar la representación de las soluciones mediante estructuras de árboles.

En las tres primeras representaciones aplicar los operadores de evolución es mucho más sencillo: cambiar un 0 por un 1 o viceversa, sumar o restar al valor de un número una cantidad elegida al azar, o cambiar una letra por otra y en la última los cambios aleatorios pueden generarse cambiando el operador, alterando el valor de un cierto nodo del árbol o sustituyendo un subárbol por otro.

3.3.1.2 Operadores Genéticos

Para garantizar la evolución de los individuos y seguir explorando el espacio de búsqueda teniendo en cuenta las mejores soluciones(individuos con mejor aptitud) es necesario aplicar una serie de operadores a la población que va quedando en cada iteración del algoritmo. A continuación se explican de manera sencilla los principales operadores genéticos.

Operador de Cruzamiento: Este operador permite la combinación entre las soluciones encontradas(padres) dando lugar a nuevos descendientes(hijos). La aplicación de dicho operador consiste en seleccionar de manera aleatoria pares de soluciones, luego se dividen ambos padres en un punto y se crean los hijos intercambiando las colas, véase Figura 3.2. Generalmente el operador de cruzamiento no se aplica a todos los pares de individuos que han sido seleccionados para emparejarse, sino que se aplica de manera aleatoria, normalmente con una probabilidad comprendida entre 0.5 y 1.0. Si el operador de cruce no se aplica, la descendencia se obtiene duplicando los padres. Existen otras variantes del operador de cruzamiento que se realizan en más de un punto. Aunque se debe tener en cuenta la investigación realizada por (De Jong, 1975) donde concluyó que el cruce basado en dos puntos, representaba una mejora mientras que añadir más puntos no benefeciaba el comportamiento del algoritmo. Tener más de un punto de cruce beneficia en que el espacio de búsqueda se puede explorar más fácilmente, sin embargo aumenta la probabilidad de división de buenos pedazos genéticos.

Operador de Mutación: El operador de mutación consiste en la alteración aleatoria (con cierta probabilidad) de cada gen componente del cromosoma.

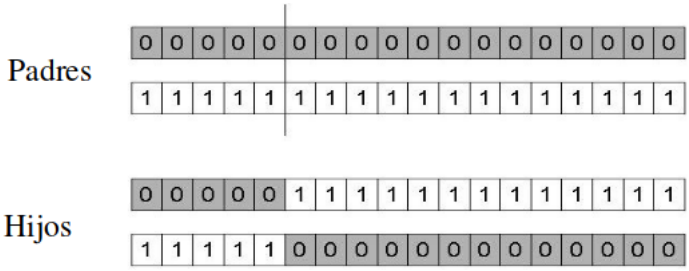

Figura 3.2: Ejemplo de operador de cruzamiento

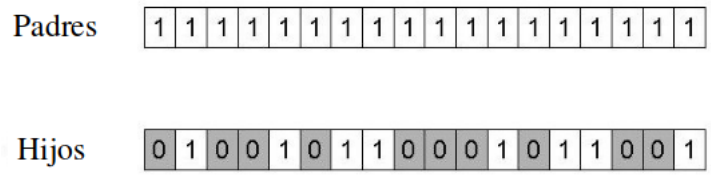

Figura 3.3: Ejemplo de operador de mutación

Dicha probabilidad se conoce como probabilidad de mutación y típicamente se encuentra entre $[\frac{1}{\text{tamañoPoblacion}}, \frac{1}{\text{tamañoCromosoma}}]$ Es muy común pensar que el operador de cruzamiento tiene más importancia que el operador de mutación, al permitir una exploración en el espacio de búsqueda más rápida, sin embargo la mutación garantiza que ningún punto tenga probabilidad cero de ser analizado, y es de gran importancia para asegurar la convergencia de los algoritmos genéticos

3.3.1.3 Métodos de Selección

Uno de los principales pasos del algortimo genético es la selección de los individuos para la reproducción, es decir seleccionar las soluciones sobre las cuales se aplicarán los operadores genéticos para crear la nueva población. Algunos de los métodos de selección son:

- Selección por ruleta: Consiste en asignar un segmento de la ruleta a los individuos con base en la aptitud de éstos y la aptitud total de la población actual, y girar la ruleta tantas veces como selecciones se requieran.
- Muestreo Estocástico Universal: Este método utiliza un único giro de la ruleta siendo los sectores circulares proporcionales a la función objetivo. Los individuos son seleccionados a partir de marcadores, igualmente espaciados y con comienzo aleatorio (Baker, 1987).
- Selección por Torneo: Opera eligiendo aleatoriamente algún número de individuos q de la población generacional y seleccionando los mejores a sobrevivir en la próxima generación por medio de hacerles competir entre ellos. El torneo binario (q = 2) es la más común (Coello et al., 2007).

- Selección por Rango: A cada individuo de la población se le asigna un rango numérico basado en su aptitud, y la selección se basa en este ranking, en lugar de las diferencias absolutas en aptitud. La ventaja de este método es que puede evitar que individuos muy aptos ganen dominancia al principio a expensas de los menos aptos, lo que reduciría la diversidad genética de la población y podría obstaculizar la búsqueda de una solución aceptable.

- $(\mu + \lambda)$ y (μ, λ): μ representa el número de soluciones padre y λ el número de soluciones hijas. El primero selecciona a los μ mejores individuos que toma tanto de los padres como de los hijos. El segundo selecciona los μ individuos solo de la población infantil.

3.3.2 Métodos basados en el frente de Pareto

Estos métodos cuentan con un esquema de selección que se basa en el concepto de optimalidad de Pareto. Dichos métodos se subdividen en dos generaciones. La primera generación se distingue por estar basada en el uso de la denominada jerarquización de Pareto (Pareto ranking) propuesta por (Goldberg, 1989). La idea es obtener a las soluciones no dominadas en la población y asignarles una jerarquía (la más alta). Posteriormente, se excluyen estas soluciones y se obtienen las siguientes soluciones no dominadas asignándoles una jerarquía menor, y así sucesivamente hasta jerarquizar a toda la población.

Los algoritmos más representativos de estos métodos son los siguientes:

- Non-dominated Sorting Genetic Algorithm (NSGA) propuesto por (Srinivas and Deb, 1994)

- Niched-Pareto Genetic Algorithm (NPGA) propuesto por (rey Horn et al., 1994)

- Multiobjective Genetic Algorithm (MOGA) propuesto por (Fonseca et al., 1993)

Luego surgen los métodos de segunda generación, denominados de esa forma porque incorporan el concepto de elitismo. El elitismo incluye un mecanismo de preservación de la soluciones dominantes a lo largo de varias generaciones. Los algoritmos de segunda generación enfatizan la eficiencia computacional. Se busca vencer la complejidad de la jerarquización de Pareto $O(kM^2)$ donde k es el número de funciones objetivo y M el tamaño de la población. Entre los algoritmos de la segunda generación podemos encontrar: SPEA, SPEA2, NSGA-II, MOMGA, MOMGA-II, PAES, PESA, PESA II, etc.... Desde finales de los 90 los algoritmos de segunda generación son considerados el estado del Arte en el área. A continuación describimos el algoritmo NSGA-II por ser uno de los algortimos más populares y con buenos resultados en las diferentes áreas donde se ha aplicado.

NSGA-II

Este algoritmo se propuso como una mejora del NSGA (Srinivas and Deb, 1994), incorporando el concepto de elitismo, lo que significa que el conjunto de individuos más adecuados tiene garantizado un lugar en la próxima generación.

El algoritmo NSGA-II comienza con una población inicial de N individuos (P_t), de los cuales se obtienen N descendientes (Q_t). Con esta población de tamaño 2N (R_t) el algoritmo procede a clasificar cada una de las soluciones vectoriales (vectores de decisión de Pareto-óptimo) en diferentes frentes (conjuntos locales de Pareto-óptimo). Las soluciones no dominadas del conjunto se obtienen y se incluyen en el primer frente. Este mismo proceso se repite con los vectores de decisión restantes hasta que todos se clasifican de alguna manera en diferentes frentes. Los pasos principales del algoritmo NSGA-II se describen a continuación:

1. Combinar las poblaciones de padres y descendientes para crear $R_t = P_t \cup Q_t$
2. Realizar el ordenamiento no dominado en R_t e identificar los frentes $Fi, i = 1, 2, ..., etc.$
3. Hacer $P_{t+1} = \emptyset$.
4. Mientras $|P_{t+1}|+|F_i| < Ndo|P_{t+1}| = |P_{t+1}|\cup|F_i|$
5. Ordenar por distancia de apilamiento (crowding distance) [1] e incluir en P_i la $N - |P_{t+1}|$ soluciones que son más dispersas usando los valores de distancia de apilamiento asociados al frente F_i.
6. Crear población de descendientes Q_{i+1} a partir de P_{i+1} usando una selección de torneo binario, cruce y mutación.

NSGA-II además de contar con el uso de elitismo es mucho más eficiente (computacionalmente) que NSGA y es un algoritmo altamente competitivo en convergencia al pareto. Por estas razones se eligió como el algoritmo de búsqueda para la propuesta presentada en esta tesis.

3.4 Trabajos Relacionados

La idea de abordar el resumen de texto como un problema de optimización no es nueva. Las primeras propuestas se centraron en la optimización de un solo objetivo, sugiriendo diferentes métodos para su resolución. Carbonell and Jade (1998) propusieron un procedimiento codicioso aproximado que incluye oraciones que maximizan la función de Relevancia Marginal Máxima, cuyo objetivo es minimizar la redundancia e incluir las oraciones más relacionadas con la consulta. McDonald (2007) estudió varios algoritmos de inferencia globales para el

[1] distancia de apilamiento es una medida de cuán cerca está un individuo de sus vecinos. Una gran distancia promedio de apilamiento dará como resultado una mejor diversidad en la población (Deb et al., 2002).

resumen de múltiples documentos: un método codicioso aproximado, un enfoque de programación dinámica y un algoritmo que utiliza la Programación Lineal Integral.

Gillick et al. (2008) trabajaron asumiendo que los documentos de entrada tienen un conjunto de conceptos, cada uno con un valor, por lo que una colección de conceptos tendrá un valor total igual a la suma de los valores de los conceptos contenidos. Se propone un método para encontrar la colección de conceptos con el valor máximo sujeto a una restricción de longitud. La solución al problema se enmarca en un algoritmo de Programación Lineal Integral de optimización global basado en el problema de la mochila.

En la última década, han surgido propuestas que abordan la tarea de resumen como un problema de optimización multiobjetivo, pero más bien con un enfoque *linealizado*. Huang *et al.* Huang et al. (2010) propusieron optimizar 4 funciones: cobertura de información, importancia, redundancia y coherencia. Estas funciones se definen sobre la base de una selección de términos principales y su agrupación en grupos. Los términos principales se eligen del cálculo de las estadísticas de co-ocurrencia con respecto a los términos de la consulta. Prueban con el enfoque *linealizado* y con el *algoritmo lexicográfico*, sin embargo, no tienen en cuenta el enfoque de Pareto.

Zhao and Tang (2010) proponen la resolución del problema de la optimización multiobjetivo por medio de un algoritmo genético. Definieron una sola función de aptitud como el promedio ponderado de tres factores: *característica enfocada en la consulta*, *característica de importancia*, y *característica de no redundancia*. Los valores de ponderación en la función deben ser definidos por el usuario en un rango entre 0 y 1.

Los trabajos más recientes se han centrado en el uso de diferentes algoritmos evolutivos. Un ejemplo revelador es Alguliev et al. (2013) quien propuso un algoritmo evolutivo diferencial adaptado para optimizar una función única que se formula como la cobertura dividida por la diversidad. Para la definición de estas dos propiedades (cobertura y diversidad) partieron de una representación vectorial de oraciones, cuyas características son todas las palabras en el vocabulario y donde cada valor de característica representa el peso de la palabra del vocabulario calculado por $tf - isf$. Los mismos autores propusieron en Alguliyev et al. (2015) pruebas con diferentes medidas para establecer la similitud entre oraciones combinando medidas simétricas como la distancia del coseno y el coeficiente de Jaccard, así como medidas asimétricas como la superposición. Demostraron que la medida de similitud utilizada influye en los resultados finales.

Los autores Saleh and Kadhim (2016) proponen dos modelos de optimización, uno para modelar la cobertura del contenido y otro para modelar la diversidad. Los dos modelos se acoplan y se definen como un problema de optimización de múltiples objetivos. Adoptan un algoritmo evolutivo multiobjetivo, específicamente "Decomposition Based Multi-objective Evolutionary Algorithm (MOEA/D)". Este algoritmo, es relativamente nuevo con respecto a NSGA-II y otros algoritmos evolutivos. Se basa en descomponer el problema multiobjetivo en N subproblemas de optimización escalar, que se optimizan simultáneamente mediante la evolución de una población de N soluciones Zhang and Li (2007).

La población en cada generación consiste en la mejor solución establecida hasta el momento para cada subproblema de optimización escalar. La definición de relaciones de vecindad entre subproblemas toma en consideración las distancias calculadas entre sus vectores de coeficientes agregados asociados. Para la construcción de las funciones de agregación en MOEA/D utilizan el método de Tchebycheff. La representación de las oraciones las realizan mediante vectores de peso con el valor de *tf-isf* por cada término existente en la colección de documentos. Para calcular las similitudes necesarias usan la medida del coseno. En los experimentos realizados utilizaron los documentos de la competición DUC-2000 y se evaluaron contra otros baselines como LexRank, el mejor de DUC, el método del centroide, entre otros. Las medidas evaluadas fueron de ROUGE-2 y ROUGE-L y en todos los casos el modelo propuesto estuvo por encima del resto de los métodos. De acuerdo con nuestra revisión de la literatura, esta es la primera investigación que aborda la tarea como un problema multiobjetivo real. Demostraron la efectividad del modelo propuesto basado en MOO sobre otros modelos de vanguardia. Sin embargo, aplicaron el modelo solo para resúmenes genéricos.

Más recientemente, un método de combinación fue propuesto por Mashreghi and Esmaeili (2017). Combinan la minería de texto y el algoritmo de optimización para reducir la redundancia y obtener la oración más relevante para el resumen de varios documentos. Los autores evalúan las oraciones según los criterios cuantitativos (longitud de las oraciones, posición de las oraciones y peso total de las palabras) y los criterios de similitud (oraciones, similitud con el título, oraciones que se superponen a los criterios). Esta evaluación se utiliza para la selección de las mejores oraciones utilizando un algoritmo de optimización de enjambre de partículas.

3.5 Conclusiones

Muchos de los enfoques actuales de optimización multiobjetivo para resolver la tarea de resumen se basan en el uso de algoritmos evolutivos, siendo el algoritmo implementado la variación más importante. Sin embargo, la mayoría de las propuestas se formulan como el problema de encontrar la mejor solución, que podría ser el valor mínimo o máximo de una función objetivo única que reúne todos los objetivos en uno. Esta perspectiva es útil para que eventualmente se pueda obtener una solución al problema, el óptimo global. De hecho, se han propuesto muchas soluciones para la tarea de resumen desde la perspectiva de la optimización de una función de un solo objetivo. Sin embargo, este es un objetivo poco realista en el escenario descrito anteriormente, ya que el problema se simplifica en exceso. En un problema de optimización multiobjetivo real con objetivos en conflicto, no hay una solución óptima única. La interacción entre diferentes objetivos da lugar a un conjunto de soluciones comprometidas, en gran parte conocidas como soluciones de compensación, no dominadas u óptimas de Pareto. La complejidad aumenta porque cuando se consideran objetivos múltiples (un frente óptimo de Pareto) una solución que optimiza uno de los ob-

jetivos puede inducir el efecto contrario en los otros. Por lo tanto, los métodos de un solo objetivo no pueden proporcionar un conjunto de soluciones alternativas que intercambien objetivos diferentes entre sí. Recientemente se han propuesto otras soluciones que involucran múltiples objetivos en conflicto, pero que eventualmente se agregan a una función escalar, lo que resulta en un problema de optimización de un solo objetivo.

En una tarea de resumen orientada a la consulta, uno de los principales objetivos a optimizar sería la relevancia de la consulta, a fin de obtener el resumen que mejor responda a las necesidades de información del usuario. Los trabajos anteriores se han dedicado principalmente a la producción de resúmenes genéricos, por lo que se centran únicamente en las propiedades de cobertura, redundancia y longitud. Algunos de los trabajos analizados incluyen la relevancia para la consulta, pero como parte de una función de un solo objetivo. La definición de una función correcta para la relevancia de la consulta debe garantizar un resumen más cercano a las necesidades del usuario.

Capítulo 4

Optimización Multi-Objetivo con Representación Semántica

En este capítulo se presenta la solución propuesta al problema planteado. Se describe el modelo propuesto, así como los experimentos realizados en aras de validar la hipótesis planteada. Finalmente se discuten los resultados y se presentan las conclusiones.

4.1 Modelo de Optimización Multiobjetivo

El presente trabajo se centra en una propuesta de solución al problema de *resumen extractivo de múltiples documentos orientado a la consulta*. La estrategia principal para el resumen extractivo consiste, en primer lugar, en obtener las oraciones más relevantes al calificarlas y clasificarlas de acuerdo con una serie de características. Luego se seleccionan las primeras N oraciones para construir el resumen. Sin embargo, esta estrategia no garantiza que colectivamente estas oraciones formen el mejor resumen. En general, deben cumplir, al menos, con las siguientes propiedades (VS, 2016):

- Cobertura: los resúmenes deben contener unidades de texto que cubran la mayor cantidad de información relevante posible. Esta propiedad normalmente se aplica a los resúmenes genéricos.
- Relevancia: los resúmenes deben contener unidades de texto (oraciones) que sean relevantes para el usuario. Esta propiedad tiene sentido en presencia de resúmenes orientados a consultas.
- Sin redundancia: los resúmenes no deben contener varias unidades de texto que transmitan información similar.

- Los resúmenes están limitados por una longitud.

Claramente, estas propiedades pueden ayudar a formular criterios u objetivos de diseño de un problema de optimización que debe resolverse con múltiples objetivos. Encontrar una solución a este problema equivale a elegir la mejor combinación de oraciones que cumpla con las propiedades mencionadas anteriormente (posiblemente en conflicto).

Otro aspecto fundamental de la tarea de resúmen automático es la forma en que se representan las oraciones para facilitar su procesamiento, pero principalmente en un intento de capturar relaciones sintácticas y semánticas. Las representaciones embebidas de texto, calculadas mediante redes neuronales, constituyen el estado del arte en PLN. Sin embargo, a pesar del vasto trabajo que existe, no hay un consenso claro sobre qué representaciones son las más útiles para modelar fenómenos semánticos de complejidad creciente y la calidad intrínseca de las representaciones embebidas ha sido poco estudiada en general. En nuestro trabajo decidimos adoptar una estrategia para el modelado de oraciones basada en el algoritmo word2vec, complementado mediante el uso de un método geométrico, lo que nos permitió proponer un nuevo método de generación de resúmenes extractivos eficaz, sin tener que aumentar excesivamente la complejidad del modelo.

4.1.1 Funciones de aptitud

A continuación se describen y se formulan las funciones objetivos que consideramos necesarias optimizar, a partir de los criterios principales que debe cumplir un buen resumen.

Función de Cobertura:

$$Cov(x) = \sum_{i=1}^{n} sim(S_i, O)x_i \tag{4.1}$$

donde $S = \{S_1, \cdots, S_n\}$ representa el conjunto de oraciones en la colección, S_i es la representación embebida de la oración i, n es el número de oraciones en la colección, O representa el vector medio de la colección S y x_i $in\{0, 1\}$, tomará valor 1 si la oración i será incluida en el resumen y 0 si no.

La función de cobertura tiene como objetivo maximizar la similitud entre cada oración del resumen con el vector promedio de la colección. En Radev et al. (2004) demostraron que el vector promedio es el vector que mejor representa el contenido central de la colección.

Función de Longitud:

$$Lgt(x) = \sum_{i=i}^{n} l_i x_i - L \tag{4.2}$$

donde l_i es el número de palabras en la oración i y L es la longitud máxima del resumen. Esta limitación se representa minimizando dicha función.

Función de Redundancia:

$$Red(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sim(S_i, Sj) x_i x_j \tag{4.3}$$

El objetivo que se persigue con esta función es minimizar la similitud de cada oración del resumen con respecto al resto de las oraciones.

Función de Relevancia:

$$Rel(x) = \sum_{i=1}^{n} sim(S_i, Q) x_i \tag{4.4}$$

donde Q es la representación vectorial de la consulta. El objetivo de la función de relevancia es encontrar las oraciones que sean más relevantas a la consulta dada por el usuario.

4.1.2 Modelo Propuesto

En esta subsección se describe el modelo que se propone para resolver el problema planteado en la presente investigación. La figura 4.1 muestra el modelo propuesto para obtener un resumen extractivo multi-documentos y orientado a consultas, aplicando optimización multiobjetivo y representaciones embebidas.

La colección representa un conjunto de documentos de un tópico específico que se requieren resumir. Estos documentos pasan a la fase de *Preprocesamiento* donde se realizan los pasos necesarios para la preparación del texto, entre las cuales se encuentran:

1. Segmentación de oraciones.
2. Obtención de tokens.
3. Eliminación de palabras de parada [1]

[1] Las palabras de parada, en inglés "stop-words"son aquellos términos que no se consideran relevantes en las tareas de PLN para su procesamiento, porque pueden generar ruidos en las métricas obtenidas, como por ejemplo artículos y preposiciones.

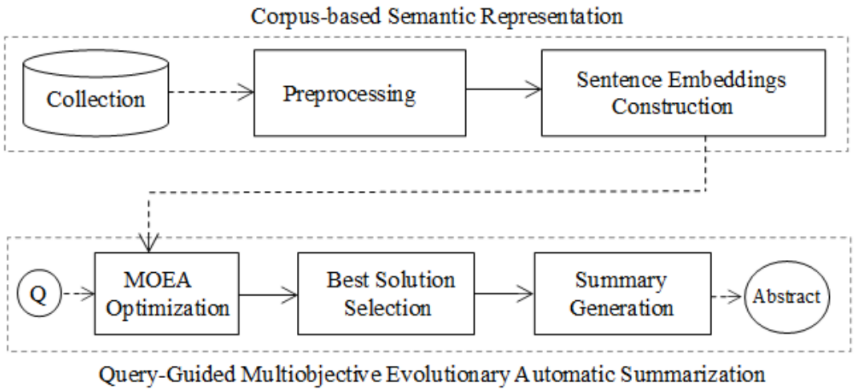

Figura 4.1: Modelo propuesto para la Generación de Resúmenes Multi-Documentos

Para la etapa **Representación embebida de oraciones** se usó un subconjunto de vectores pre-entrenados con el modelo `Word2Vec`. Los vectores se obtuvieron de parte del conjunto de datos de Google Noticias. La colección de palabras pre-entrenadas, que tiene un total de 3 millones de palabras y frases, se obtuvo directamente de Google [2]. Cada vector es una palabra embebida de 300 dimensiones. Para la representación de oraciones se utilizaron dos métodos:

1. Promedio de los vectores correspondientes a cada palabra de la oración.
2. Representación de las oraciones como subespacios de bajo rango.

La salida de esta etapa son los vectores que representan a cada una de las oraciones de los documentos.

En la fase de **Optimización evolutiva multiobjetivo** se procede a la ejecución del algoritmo de optimización, que en el caso de nuestras pruebas se propuso el algoritmo NSGA-II, aunque sería válido ejecutar cualquier otro algoritmo que se considere pueda realizar una optimización adecuada. Se aplica a las 4 funciones objetivo definidas anteriormente. Una solución está representada por un vector binario y el número de bits corresponde con el número total de oraciones de la colección. Los bits con valor 1 significan que corresponden a una oración que será incluida en el resumen. Como ya se explicó, el algoritmo partirá de un conjunto de soluciones aleatorias, es decir se tendrán diferentes propuestas de conjuntos de oraciones para el resumen, y en la medida que se ejecute el algoritmo, las soluciones se irán adecuando al cumplimiento de las funciones objetivos propuestas. De manera que al final de la etapa tendremos el conjunto óptimo de Pareto, que no son más que diferentes variantes de soluciones.

Selección de la mejor solución En esta etapa, el algoritmo selecciona una solución del conjunto óptimo de Pareto obtenido en el paso anterior. Como primer enfoque, se propone seleccionar la primera solución obtenida del primer

[2] https://code.google.com/archive/p/word2vec/

frente. Podría decirse que este criterio de selección está lejos de ser la mejor alternativa. Sin embargo, la selección de la mejor solución del conjunto Pareto-óptimo está lejos de ser obvia, como se señalará más adelante.

Generación del Resumen: Una vez que se obtiene la solución óptima en forma binaria, el resumen se conforma con las oraciones correspondientes extraídas del texto original.

4.2 Experimentación

En general, la mayoría de las pruebas se desarrollaron con el corpus de TAC-2009 (Dang and Owczarzak, 2009), que se compone de noticias de diferentes agencias de prensa como Agence France Presse, New York Times, etc., en el período de 2004 a 2006. Las noticias se agrupan en 44 temas, y hay 20 noticias por tema. Además, se cuenta con 4 resúmenes por tópico escritos por humanos, que son los resúmenes de referencia para la evaluación. El enfoque linealizado hace referencia a la optimización multiobjetivo utilizando una única función, en nuestro caso solo es la suma de las funciones objetivos definidas anteriormente. Para la evaluación de los resúmenes se utilizaron las mismas métricas que en la competición. La tarea de evaluación de resúmenes tiene como estándar el uso de las métricas definidas en ROUGE(Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004). En el siguiente apartado se explicarán brevemente dichas métricas.

4.2.1 Evaluación de Resúmenes

ROUGE (Lin, 2004), es un paquete automático para la evaluación de resumenes de texto que utiliza co-ocurrencias de unigram entre pares de resumen. Por lo general, los resúmenes producidos automáticamente se comparan con los escritos por humanos, utilizando medidas estandarizadas para determinar la calidad de un resumen. La medida de evaluación utilizada en los primeros experimentos de este trabajo fue ROUGE-N, N $\in \{1, 2\}$, donde N es el número de n-gramas utilizados para la métrica. Esta medida es bien aceptada por la comunidad científica para evaluar la calidad del resumen porque calcula la coincidencia de N-gramas entre el resumen generado y un resumen de referencia (escrito por un humano). De acuerdo con los estudios en profundidad basados en diversas métricas estadísticas y la comparación con los resultados de la Conferencia de Comprensión de Documentos(DUC) 2002, este método de evaluación se correlaciona sorprendentemente bien con la evaluación humana (Lin and Hovy, 2003)

ROUGE-N se calcula como:

$$\frac{\sum_{S \in RefSum} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in RefSum} \sum_{gram_n \in S} Count(gram_n)} \tag{4.5}$$

donde $RefSum$ es el conjunto de Resúmenes de Referencia; $Count_{match}(gram_n)$ es el número de n-gramas que coinciden entre un resumen del sistema y los resúmenes de referencia y $Count(gram_n)$ es el total de n-gramas en el resumen de referencia.

Otras dos medidas se utilizaron en el resto de los experimentos:

ROUGE-L: Esta medida emplea el concepto de subsecuencia más común (LCS) entre las dos secuencias de texto. La intuición es que cuanto más largo es el LCS entre dos oraciones de resumen, más similares son. Aunque esta métrica es más flexible que la anterior, tiene la desventaja de que todos los n-gramas deben ser consecutivos.

ROUGE-SU: Esta métrica llamada skip bi-gram o uni-gram considera tanto los bi-gramas como los uni-gramas y permite la inserción de palabras entre la primera y la última palabra de los bi-gramos, por lo que no es necesario que sean secuencias consecutivas de palabras.

4.2.2 Experimentos de Optimización Multiobjetivo

Para validar la propuesta, inicialmente se realizaron tres experimentos:

1. Optimización de cobertura y longitud para resumen genérico.
2. Optimización de Relevancia y Redundancia.
3. Selección de la mejor solución para QOS-MOO-WE

El primer experimento apunta a evaluar el desempeño del enfoque propuesto para una tarea de resumen genérico. El segundo experimento se diseñó para una tarea de resumen orientada a la consulta. Finalmente, un tercer experimento compara el rendimiento del método de selección utilizado en la versión actual del sistema con la mejor solución del conjunto de Pareto. El propósito de este tercer experimento es mostrar los beneficios potenciales de buscar la mejor solución en el conjunto de Pareto, que está fuera del alcance del trabajo de Tesis.

Para comprender los resultados mostrados, se explica la notación utilizada.

- Para referirnos al método de optimización:
  - LIN: representa el método de optimización linealizado.
  - MOO: representa la optimización multiobjetivo con enfoque de Pareto (para estas pruebas con el algoritmo de NSGA)
- Para referirnos al método de representación:
  - TFISF: se refiere a la representación de bolsa de palabras, que utilza la métrica de $tf - isf$.
  - WE: se refiere a la representación embebida de oraciones.

Utilizando la notación anterior fueron nombrados los diferentes sistemas de generación de resúmenes que se desarrollaron para la experimentación:

- LIN-TFISF: Es un sistema que optimiza las funciones objetivo mediante el método linealizado y la representación vectorial de oraciones se basa en la medida de $tf - isf$

- LIN-WE: Es un sistema que aplica el método linealizado en la optimización, pero utiliza la representación embebida de oraciones. Para estos primeros experimentos se utilizó el promedio de los vectores de las palabras que conforman una oración, siendo estos vectores de palabras extraídos de word2vec.

- MOO-TFISF: Sistema que aborda la tarea de optimización con el enfoque de Pareto, es decir obtiene el conjunto de las posibles soluciones óptimas, y la representación de las oraciones se basa en el modelo vectorial con la medida $tf - isf$

- MOO-WE: Este sistema, al igual que el anterior, aborda el paso de la optimización con el enfoque de Pareto, pero la representación de oraciones está basada en una representación embebida.

Dado que en la revisión de la literatura, no se encontraron propuestas de generación de resúmenes multi-documentos orientados a consultas utilizando el enfoque de Pareto, se decide utilizar como baseline el sistema **LIN-TFISF**. Ello debido a que sí existen trabajos que utilizan el enfoque linealizado, pero no orientados a consultas.

Optimización de cobertura y longitud para resumen genérico: Se decidió realizar un primer experimento para obtener un resúmen de múltiples documentos de tipo genérico, es decir que no se cuenta con ninguna consulta introducida por el usuario. Esto debido a que es un resumen más sencillo y permitía que nos pudiéramos enfocar en comparar los resultados obtenidos entre la propuesta de optimización linealizada y la propuesta mutiobjetivo.

Las tablas 4.1 - 4.4 muestran los resultados obtenidos usando ROUGE-1 y ROUGE-2 respectivamente. En las tablas 4.2 y 4.4 se evalúan las mismas medidas pero eliminando las palabras de parada (stopwords). Los valores marcados con asterico indican el mejor resultado de los cuatro sistemas. Estos resultados son estadísticamente mejores que los otros de acuerdo al Wilcoxon Signed Rank test. (Wilcoxon, 1945) con un nivel de significancia del 95

Tabla 4.1: Rouge-1 Evaluación para Experimento 1

Model	Recall	Precision	F-Score
LIN-TFISF	0.30534	0.30646	0.30339
LIN-WE	**0.32497**	**0.31601**	**0.31914**
MOO-TFISF	0.30681	0.30913	0.30347
MOO-WE	**0.32636***	**0.32101***	**0.32052***

Tabla 4.2: Rouge-1 en Experimento 1 (Sin palabras de parada)

Model	Recall	Precision	F-Score
LIN-TFISF	0.22143	0.22082	0.22040
LIN-WE	**0.23195***	**0.22740***	**0.22859***
MOO-TFISF	0.19601	0.20790	0.20001
MOO-WE	**0.21756**	**0.22334**	**0.21768**

Tabla 4.3: Rouge-2 Evaluación para Experimento 1

Model	Recall	Precision	F-Score
LIN-TFISF	0.07295	0.07181	0.07205
LIN-WE	**0.07776**	0.**07370**	**0.07544**
MOO-TFISF	0.06296	0.06396	0.06278
MOO-WE	**0.07835***	**0.07412***	**0.07593***

Tabla 4.4: Rouge-2 para Experimento 1 (Sin palabras de parada)

Model	Recall	Precision	F-Score
LIN-TFISF	0.05937	0.05877	0.05886
LIN-WE	**0.06412***	**0.06320***	**0.06337***
MOO-TFISF	0.04991	0.05219	0.05053
MOO-WE	**0.06005**	**0.06318**	**0.06028**

En este experimento se presta especial atención a la longitud del resumen de la solución. En la competencia TAC2009, el resumen se limitó a un máximo de 100 palabras, por lo que se analizó si los resúmenes obtenidos cumplían con esta restricción. Usando la ecuación 4.4, el parámetro L se estableció en 100 en una primera ejecución, sin embargo, se observó que los resúmenes no tenían exactamente el mismo número de palabras que L, por lo que estos valores se cambiaron por diferentes valores. Los resultados se muestran en la Figura 4.2.

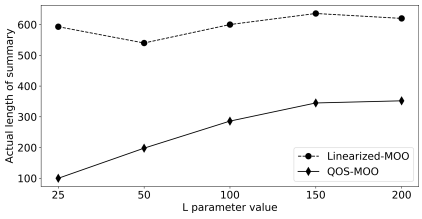

Figura 4.2: Longitud real de resúmenes obtenida en función del parámetro L

Como se puede apreciar en la Figura 4.2, en ambos casos el número de palabras obtenidas de cualquiera de los métodos es mucho mayor de lo esperado. Sin embargo, comparado con su contraparte *linealizado*, el método MOO produce resúmenes cuya longitud se comporta de forma más lineal. Como no fue posible cumplir con la restricción de longitud exacta, el criterio de longitud no se incluyó en el siguiente experimento.

Optimización de Relevancia y Redundancia: Como se mencionó anteriormente, este segundo experimento tiene como objetivo evaluar el desempeño del enfoque propuesto en una tarea de resumen enfocada en la consulta. Es importante tener en cuenta la redundancia en este caso, pues de lo contrario un resumen con alta relevancia probablemente estaría conformado por oraciones muy parecidas con una alta similitud con respecto a la consulta. Estos objetivos aparecen en conflicto, a saber, minimizar la redundancia y maximizar la relevancia. Los resultados se muestran en las tablas 4.5 a 4.8.

Tabla 4.5: Rouge-1 Evaluación en Experimento 2

Model	Recall	Precision	F-Score
LIN-TFISF	0.33436	0.31863	0.32492
LIN-WE	**0.34279***	**0.32801**	**0.33465***
MOO-TFISF	0.31251	0.31267	0.31086
MOO-WE	**0.33706**	**0.33166***	**0.33315**

Tabla 4.6: Rouge-1 en Experimento 2 (Sin palabras de paradas)

Model	Recall	Precision	F-Score
LIN-TFISF	0.22114	0.22421	0.22158
LIN-WE	**0.22164***	**0.23164**	**0.22653***
MOO-TFISF	0.19736	0.21263	0.20592
MOO-WE	**0.20483**	**0.23699***	**0.20751**

Tabla 4.7: Rouge-2 Evaluación en Experimento 2

Model	Recall	Precision	F-Score
LIN-TFISF	0.07727	0.07386	0.07521
LIN-WE	**0.08370***	**0.08080***	**0.08208***
MOO-TFISF	0.06548	0.06452	0.06469
MOO-WE	**0.06557**	**0.07889**	**0.06798**

Tabla 4.8: Rouge-2 en Experimento 2 (Sin palabras de parada)

Model	Recall	Precision	F-Score
LIN-TFISF	0.06087	0.06085	0.06057
LIN-WE	**0.07114***	**0.07095***	**0.07082***
MOO-TFISF	0.05339	0.05407	0.05364
MOO-WE	**0.05376**	**0.06671**	**0.05620**

Para cumplir con la restricción de longitud la estrategia seguida fue ordenar las oraciones obtenidas como parte de la solución por su similitud con respecto a la consulta y seleccionar las primeras oraciones hasta que se cumpliera con la longitud del resumen especificada.

Mejor solución para MOO-WE: El objetivo de este experimento es demostrar que el método propuesto tiene beneficios potenciales contra los enfoques del tipo MOO-LIN. En este experimento se realizó la evaluación de todas las soluciones que arrojó el algoritmo de optimización, y por cada tópico se seleccionó el resumen que mejor evaluación obtuvo. Los resultados se muestran en las tablas 4.9 a 4.12.

Tabla 4.9: Rouge-1 Evaluación en Experimento 3

Model	Recall	Precision	F-Score
LIN-MOO-TFISF	0.33436	0.31863	0.32492
LIN-MOO-WE	**0.34279**	**0.32801**	**0.33465**
QOS-MOO-TFISF	0.37486	0.36151	0.36711
QOS-MOO-WE	**0.39290***	**0.37250***	**0.38135***

Tabla 4.10: Rouge-1 en Experimento 3 (Sin palabras de parada)

Model	Recall	Precision	F-Score
LIN-MOO-TFISF	0.22114	0.22421	0.22158
LIN-MOO-WE	**0.22164**	**0.23164**	**0.22653**
QOS-MOO-TFISF	0.26452	0.26980	0.26621
QOS-MOO-WE	**0.28571***	**0.28761***	**0.28529***

Tabla 4.11: Rouge-2 Evaluación en Experimento 3

Model	Recall	Precision	F-Score
LIN-MOO-TFISF	0.07727	0.07386	0.07521
LIN-MOO-WE	**0.08370**	**0.08080**	**0.08208**
QOS-MOO-TFISF	0.10017	0.09879	0.09914
QOS-MOO-WE	**0.11927***	**0.11815***	**0.11834***

Tabla 4.12: Rouge-2 Evaluación en Experimento 3

Model	Recall	Precision	F-Score
LIN-TFISF	0.06087	0.06085	0.06057
LIN-WE	**0.07114**	**0.07095**	**0.07082**
MOO-TFISF	0.08954	0.08968	0.08908
MOO-WE	**0.09423***	**0.10184***	**0.09510***

4.2.3 Experimentos con Expansión de Consulta

En esta sección se describen nuevos experimentos realizados con el propósito de mejorar los resultados alcanzados en las primeras pruebas de concepto. El principal experimento está centrado en enriquecer la consulta dada por el usuario, este método se ha aplicado en Recuperación de Información con buenos resultados y se conoce como expansión de la consulta. Además se hicieron algunas variaciones como probar con diferentes medidas para establecer la similitud entre la consulta y una oración; y se propone una variante para conformar el vocabulario con el que se trabaja.

La expansión de la consulta consiste en añadirle un conjunto de términos que puedan darle una mayor profundidad desde el punto de vista semántico a la misma, generalmente se añaden términos relacionados con los términos de la consulta, como puede ser sinónimos. Este proceso puede hacerse de forma manual, interactuando con el usuario o puede realizarse de forma automática. Existen numerosas propuestas para enriquecer la consulta: (Moral et al., 2014) con un enfoque basado en agrupamiento, (Sanderson and Lawrie, 2002) y (Lang et al., 2010) con propuestas basadas en jerarquía de conceptos, (Smeaton and Van Rijsbergen, 1983) y (Vechtomova et al., 2003) toman en cuenta la co-ocurrencia de los términos, entre muchos otros. En nuestro experimento se decide utilizar el método propuesto en (Fernández-Reyes et al., 2018) ya que utilizan representaciones embebidas, específicamente el modelo word2vec que es también la representación que se utiliza en este trabajo de tesis.

Se parte de la conformación del vocabulario V con las representaciones embebidas de cada palabra. Tal y como se propone en (Fernández-Reyes et al., 2018) se conforma un grafo bipartito entre cada una de los términos de la consulta y

los términos del vocabulario, donde el peso de cada arista será la similitud entre ambas representaciones embebidas de las palabras, véase 4.3.

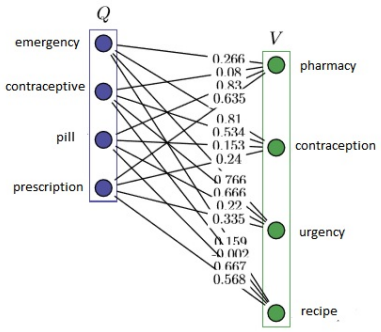

Figura 4.3: Ejemplo subgrafo bipartito. Tomado de (Fernández-Reyes et al., 2018)

Formalmente se construye el grafo bipartito (Q,V,E) partiendo de cada término de la consulta original, que estará conformado por:

$$\begin{aligned} Q &= \{q_1, q_2, \cdots, q_n\} \\ V &= \{t_1, t_2, \cdots, t_m\} \\ E &= \{\cos(q_1, t_1), \cos(q_1, t_2), \cdots, \cos(q_2, t_1), \cos(q_2, t_2), \cdots\} \end{aligned} \tag{4.6}$$

Donde Q es el conjunto de términos de la consulta, y W el conjunto de todos los términos del vocabulario, a partir de W se define $V = W - Q$. Siendo entonces V el subconjunto de términos del vocabulario que no son los términos de la consulta y por tanto son todos los candidatos posibles a expansión. $E = \{w(q_i, t_j) \mid w_{ij} \neq 0\}$ donde $w(q_i, t_j) = cos(q_i, t_j)$ es el peso de cada enlace en el grafo (Fernández-Reyes, 2017).

Una vez obtenido el grafo es necesario establecer una estrategia para la selección de los términos que conformarán parte de la consulta expandida. En este caso se utiliza la propuesta V2Q (Prospect Guided), en donde las asignaciones de peso se interpretan como que cada término del vocabulario(candidato) le asigna un voto a cada término de la consulta. Con esta interpretación se intuye que se pueden recuperar más términos que estén relacionados a la consulta como un todo y no con un solo término de manera independiente. Para determinar el peso final que cada término candidato tendrá, se proponen dos métodos: seleccionar el voto máximo($Rank_{max}$) o promediar todos los votos($Rank_{avg}$), y se definen de la siguiente manera:

El método para construir $Rank_{max}$:

$$sim_{max}(Q, t) := \max\{sim(q, t) \ : \ \forall q \in Q, t \in V\} \tag{4.7}$$

$$Rank_{max} := \{(t, sim_{max}(Q, t)) \ : \ \forall t \in V\} \tag{4.8}$$

El método $Rank_{avg}$:

$$w(q,t) = sim(q,t) \tag{4.9}$$

$$avg(Q,t) := \frac{\sum_{q \in Q} w(q,t)}{\sum_{q \in Q} f(q,Q)} \tag{4.10}$$

$$Rank_{avg} := \{(t, avg(Q,t)) \ : \ \forall t \in V\} \tag{4.11}$$

En la propuesta original la similitud es definida como el coseno, en nuestro caso lo definimos general ya que como se verá más adelante se proponen otras medidas de similitud. Una vez obtenida la lista de términos candidatos ordenados en forma descendente, se seleccionan los primeros top N para la expansión de la consulta, en este caso N es igual al número de términos de la consulta.

Una de las variantes que se proponen a este método es la reducción del vocabulario original, pues consideramos que se pudieran estar ponderando muchos términos que no aportan tanto valor pero que influyen en el voto final. La propuesta consiste en aplicar el algoritmo de agrupamiento K-medias sobre todos los vectores del vocabulario, una vez obtenido los clusters, se seleccionan los clusters con mayor similitud a la consulta, y se conforma el vocabulario con las palabras que están dentro de dichos clusters. La similitud entre cada cluster y la consulta se calcula como:

$$Sim(Q, Clust_n) := \cos(Q, Cent) \tag{4.12}$$

donde $\cos(Q, Cent)$ es la similitud del coseno entre el centroide del cluster $Clust_n$ y la consulta Q. El vector de consulta Q se calculó promediando las reprentaciones embebidas de cada término de la misma.

En las tareas de PLN la similitud del coseno es una medida de referencia popular por su simplicidad, eficiencia y efectividad, pero muchos trabajos se han propuesto donde se demuestra que el uso de otras medidas puede mejorar los resultados. Al depender mucho del contexto, no está definido de manera explícita cuándo una medida es mejor que otra, por lo que sólo a través de la vía experimental es posible llegar a algunas conclusiones, y aún así es muy específico de la tarea que se esté desarrollando. Para el presente experimento también se decicidió probar con varias medidas de similitud:

- Coseno: Es una medida de similitud que devuelve el valor del coseno del ángulo comprendido entre dos vectores, sus valores están en el intervalo [-1,1]. Cuando el ángulo entre los dos vectores es 0, el coseno tiene valor 1, lo cual indica que la similitud es máxima, por lo que mientras más cercanos sean los valores a 1 más similares serán los vectores. Se calcula utilizando el producto punto y la magnitud:

$$\cos(q,t) = \frac{q \cdot t}{\|q\|\|t\|} = \frac{\sum_{i=1}^{n} q_i t_i}{\sqrt{\sum_{i=1}^{n} (q_i)^2} \sqrt{\sum_{i=1}^{n} (t_i)^2}} \tag{4.13}$$

- Euclideana: Representa la longitud del segmento que une los dos vectores, por tanto mientras mayor sea esta distancia menos similares serán los vectores. Se formula como:

$$Euclid\_Dist(q,t) = \sqrt{\sum_{i=1}^{n}(q_i - t_i)^2} \tag{4.14}$$

- Correlación de Pearson: El coeficiente de correlación de Pearson busca determinar si existe un grado de correlación lineal entre dos variables. Sus valores se encuentran entre [-1,1], donde 1 indica una correlación lineal positiva total, 0 que no existe correlación lineal, y -1 una correlación lineal negativa. Su fórmula es:

$$\rho q, t = \frac{cov(q,t)}{\sigma q \sigma t} \tag{4.15}$$

donde $cov(q,t)$ es la covarianza entre q y t, que se calcula a partir de la construcción de la matriz de covarianza entre los dos vectores; y σq es la desviación estándar de q, así como σt es la desviación estándar de t. En este segundo experimento se aplican las medidas de similitud explicadas anteriormente, y para cada una se realiza la expansión de la consulta con el vocabulario original y con el vocabulario reducido según la propuesta que se explicó, además se aplicaron los métodos de $Rank_{max}$ y $Rank_{avg}$, por lo que para esta experimentación se cuenta con un total de 12 variantes.

Para la evaluación se decidió incluir las variantes que normalmente se presentan en las competiciones, es decir:

- Evaluación con stemming: Stemming es una técnica que se aplica en las tareas de PLN donde la palabra se reduce a su raíz, que es una forma igual o menor que la palabra. De esta forma algunos grupos de palabras serán tomados como la misma. Ejemplo para las palabras retrieval", retrieved", retrieves"su stemming sería retrieve".

- Evaluación sin stemming.

- Evaluación sin stemming y sin palabras de parada(stop words)

También se añadieron otras medidas que típicamente se usan en los trabajos para la evaluación de resúmenes: ROUGE-L, ROUGE-SU4

En las tablas 4.13 y 4.14 se muestran los resultados con la similitud del coseno. En la primera tabla se muestran los resultados aplicando stemming y en la segunda los resultados sin aplicar stemming y eliminando las palabras de paradas. Las primeras combinaciones (ExpCosine-Avg y ExpCosine-Max) se refieren al experimento de expansión utilizando los métodos $Rank_{avg}$ y $Rank_{max}$ respectivamente y con el vocabulario original, las otras dos combinaciones (ExpCosine-Avg-Clust y ExpCosine-Max-Clust) se refieren a los mismo pero con el vocabulario reducido, donde aplicamos el método de agrupamiento por clusters.

Tabla 4.13: Rouge Eval - Stemming

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
ExpCosine-Avg	0.38374	0.10941	0.33893	0.14370
ExpCosine-Max	0.39066	0.11257	0.34672	0.14644
ExpCosine-Avg-Clust	**0.39182**	**0.11713***	**0.34682**	**0.14820**
ExpCosine-Max-Clust	0.38841	0.10899*	0.34282	0.14405

Tabla 4.14: Rouge Eval - No-Stopwords

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
ExpCosine-Avg	0.27161*	0.08815*	0.24976*	0.09377*
ExpCosine-Max	0.27806	0.09517	0.25950	0.09911
ExpCosine-Avg-Clust	**0.28415***	**0.09707***	**0.26051***	**0.10217***
ExpCosine-Max-Clust	0.27540	0.09008	0.25294	0.09603*

Analizando los resultados de las tablas anteriores se evidencia que en ambos casos y para todas las medidas de evaluación la variante de Expansión de consulta usando $Rank_{avg}$ con el vocabulario reducido fue la que obtuvo mejores resultados, aunque no en todos los casos las diferencias fueron estadísticamente significativas. Partiendo de esto en las tablas 4.15 4.16 solo se muestran las comparativas con el resto de las medidas de similitud: euclideana y Pearson para esta variante.

Tabla 4.15: Rouge Eval - Stemming

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
ExpEuclid-Avg-Clust	0.37466	0.10303	0.32856	0.13721
ExpPearson-Avg-Clust	0.38626*	0.10445	0.34238*	0.14084*
ExpCosine-Avg-Clust	**0.39182***	**0.11713**	**0.34682***	**0.14820***

Tabla 4.16: Rouge Eval - Sin Stemming

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
ExpEuclid-Avg-Clust	0.35056	0.09772	0.31124	0.12620
ExpPearson-Avg-Clust	0.36235*	0.09864	0.32528*	0.13048*
ExpCosine-Avg-Clust	**0.36845***	**0.11054**	**0.32912***	**0.13705***

Como se puede apreciar de la comparativa entre las medidas de similitud la del coseno fue la que obtuvo mejores resultados, y la Pearson se comportó mejor que la distancia Euclideana. Para finalizar este experimento se presenta la comparación con el experimento de la sección 4.2.2, los resultados se reflejan en las tablas 4.17 4.18 4.19

Tabla 4.17: Rouge Eval - Stemming

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
MOO-WE	0.39033	0.11345	**0.34744**	0.14496
ExpCosine-Avg-Clust	**0.39182**	**0.11713**	0.34682	**0.14820**

Tabla 4.18: Rouge Eval - Sin Stemming

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
MOO-WE(Artículo)	0.36668	0.10758	0.32863	0.13437
ExpCosine-Avg-Clust	**0.36845**	**0.11054**	**0.32912**	**0.13705**

Tabla 4.19: Rouge Eval - Sin Stopwords

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
MOO-WE(Artículo)	0.27788	0.09498	0.25575	0.09893
ExpCosine-Avg-Clust	**0.28415**	**0.09707**	**0.26051**	**0.10217**

4.2.4 Experimentos de oraciones como subespacios

Como se describió en el marco teórico existen varios métodos para la representación de oraciones, que se pueden agrupar en aquellos que utilizan redes neuronales para aprender las representaciones y los que usan operaciones algebráicas sobre los vectores de palabras que conforman las oraciones como la suma y el promedio. Los experimentos anteriores se han realizado con la representación a través del promedio de los vectores, tomados de word2vec. En este apartado proponemos el uso del método de representación de oraciones propuesto en (Mu et al., 2017).

Teniendo en cuenta los reultados alcanzados en los experimentos, se decide probar este método de representación para la tarea de resúmenes con el enfoque multiobjetivo abordado en esta tesis. El nuevo sistema, nombrado **MOO-LowRank**, aplicará el enfoque de Pareto en la optimización y la representación de oraciones como subespacios de bajo rango, especificamente de dimensión 4. Una comparación de los resultados obtenidos con el enfoque de representación mediante el promedio se muestra en las tablas 4.20 4.21 4.22

Tabla 4.20: Rouge Evaluación-Stemming (Low-Rank vs WE)

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
MOO-WE	0.39033	0.11345	0.34744	0.14496
MOO-LowRank	**0.40724***	**0.12585***	**0.36262***	**0.15838***

Tabla 4.21: Rouge Evaluación-Sin Stemming (Low-Rank vs WE)

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
MOO-WE	0.36668	0.10758	0.32863	0.13437
MOO-LowRank	**0.38124***	**0.11968***	**0.34199***	**0.14649***

Tabla 4.22: Rouge Evaluación-Sin StopWords (Low-Rank vs WE)

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
MOO-WE	0.36668	0.10758	0.32863	0.13437
MOO-LowRank	**0.38124***	**0.11968***	**0.34199***	**0.14649***

Un segundo experimento se realizó, pero en este caso con otro corpus, el corpus de la competición TAC 2010, que tiene una constitución similar al del TAC 2009, con 46 tópicos con noticias de diferentes agencias de prensa. Los resultados obtenidos se compararon con el mejor de esa competición, véase Tabla 4.23

Tabla 4.23: Rouge Evaluación-Sin StopWords (TAC-2010)

Model	Rouge-1	Rouge-2	ROUGE-L	ROUGE-SU4
ID22(BEST)	0.37174	0.09574	0.32398	0.13024
MOO-LowRank	**0.38412**	**0.10853**	**0.33421**	**0.13125**

4.3 Discusión de los resultados

El primer exprimento de la subsección 4.2.2 consistió en obtener un resumen de tipo genérico a partir de múltiples documentos, optimizando la cobertura y la longitud. Los resultados obtenidos en esta primera evaluación muestran que la representación embebida es mejor que con $tf - isf$ en todos los casos y que la propuesta multiobjetivo(MOO) es mejor que el método *linealizado*(LIN) para las medidas ROUGE-1 y ROUGE-2 incluyendo las stopwords.

El segundo experimento de la misma subsección ya se centra en un resumen orientado a consulta, por lo que se optimiza la relevancia respecto a la consulta, tratando de minimizar la redundancia. Con respecto al tipo de representación de oraciones utilizado es posible afirmar que la representación embebida es mejor que el método clásico $tf - isf$, al igual que ocurrió en el primer experimento. Sin embargo, los resultados no son concluyentes a favor del método propuesto con respecto a la optimización multiobjetivo con enfoque de Pareto, ya que no es cierto para todos los casos que el enfoque MOO-WE es mejor que el método LIN-WE. Consideramos que el problema se deriva del procedimiento de selección de la solución del conjunto de Pareto.

El tercer experimento (4.2.2) se llevó a cabo con el objetivo de determinar si en el conjunto de resultados del Frente de Pareto existen buenas soluciones,

en especial si existía alguna solución mejor a la seleccionada en el segundo experimento, la cual consistió en seleccionar la primera solución del primer frente. Los resultados obtenidos permitieron validar que sí existe una solución de mayor calidad dentro del conjunto de soluciones del Frente de Pareto, véanse las tablas 4.9 a 4.12. Aún con los resultados obtenidos, se tenía la interrogante de cómo identificar donde se encuentra la mejor solución. Para ello decidimos graficar las mejores soluciones de cada uno de los tópicos con el objetivo de identificar algún patrón de ubicación en el Frente. Sin embargo, no es obvio encontrar la mejor solución, como se puede observar en la Figura 4.4. En esta figura, los resultados del conjunto de Pareto se normalizaron utilizando un enfoque típico de *min-max*. La mejor solución del conjunto de Pareto de cada tema se traza para ubicarla en la frontera de Pareto. A este respecto, se debe tener en cuenta que los objetivos representados son conflictivos, ya que el objetivo es maximizar la relevancia y minimizar la redundancia. La figura muestra que el espacio de las mejores soluciones está esparcido y no es obvio a partir de la información disponible, definir un patrón de optimalidad para todos los casos. Este problema por tanto requiere mayor análisis y trabajo de investigación.

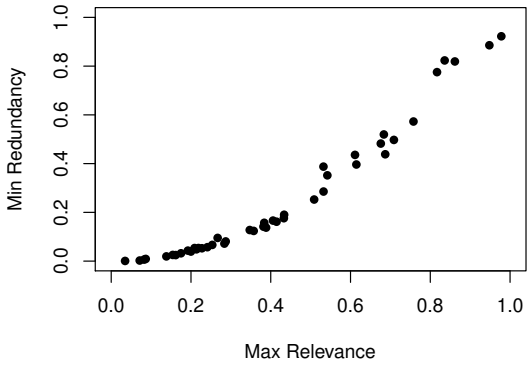

Figura 4.4: Mejores soluciones en QOS-MOO-WE

La expansión de consulta, en general, no produjo mejoras significativas en los resultados obtenidos. Esto pudiera deberse al hecho de que las consultas de entrada utilizadas, son consultas bien formuladas, ya que son las propuestas para las competiciones, y no se está en presencia de consultas que pudieran estar incompletas como suele ocurrir en el caso de los Sistemas de Recuperación de Información, cuando dependen de lo que entra cualquier usuario. Por otra parte, al tener una representación embebida es posible que ya se esté capturando toda la semántica necesaria y no se requiera darle mayor significado a la consulta, puesto que de alguna forma en dicha representación ya se encuentran embebidas las relaciones sintánticas y semánticas con otras palabras y oraciones.

Por último, las pruebas realizadas con la representación de oraciones como subespacios de bajo rango (4.2.4) permiten concluir que este tipo de representación captura de una manera adecuada la semántica y garantiza un mejor rendimiento dentro de nuestro modelo, por lo que su uso se considera factible.

Capítulo 5

Conclusiones

A partir de la inventigación realizada y el estudio del estado del arte pudimos arribar a las siguientes conclusiones:

- La mayoría de las propuestas en generación de resúmenes que abordan la tarea como un problema de optimización parten de una representación vectorial con enfoque frecuentista
- Muy pocos trabajos han tratado la semántica desde la representación de las oraciones.
- La mayoría de los trabajos resuelven el problema multi-objetivo convirtiéndolo a un problema de un solo objetivo.
- No se ha explorado la resolución del POM desde el enfoque de Pareto, en resúmenes orientados a consultas.

Es por ello que en el presente trabajo de tesis se propuso abordar la tarea de resúmen automático centrado en la consulta como un problema de optimización multiobjetivo, que consiste en encontrar el conjunto de soluciones óptimo de Pareto. Los enfoques de vanguardia proponen una función objetivo única, lo cual no es realista. En el mejor de los casos, se han propuesto soluciones que involucran múltiples objetivos en conflicto, que eventualmente se agregan a una función lineal, lo que resulta en un problema de optimización *linealizado*.

Este trabajo pone evidencia sobre los beneficios de las representaciones embebidas de palabras para la representación de oraciones al definir medidas de redundancia, cobertura y relevancia en un problema de optimización para una tarea de resumen. En este caso, la representación de una oración se obtuvo a partir de representaciones de palabras pre-entrenadas que fueron promediadas y se exploró otra representacion que aumentó el rendimiento general del sistema.

Los experimentos demostraron los beneficios del enfoque, lo que nos permite concluir lo siguiente:

- Una oración con representación semántica produce un mejor rendimiento que una representación de $tf - isf$

- El método de resumen orientado a la consulta basado en el conjunto óptimo de Pareto introducido en este documento funciona mejor que un enfoque de optimización multiobjetivo con una sola función que combina los objetivos a optimizar.
- La técnica de expansión de consulta utilizada no aporta cambios importantes en el modelo propuesto.
- El método de representación de oraciones como sub-espacios de bajo rango aportó mejoras en el modelo propuesto.

Una ventaja adicional del método propuesto en este trabajo es que, dependiendo del usuario y la aplicación, se podrían elegir diferentes soluciones según los resúmenes apropiados. De hecho, por un lado, puede darse el caso de que la redundancia sea tolerada en la medida en que se incluya todo lo relevante. Por otro lado, otro escenario podría ser una preferencia por resúmenes muy concisos con muy poca redundancia, aunque no incluye toda la información. El enfoque presentado en este documento tiene esta flexibilidad, ya que buscar la mejor solución del conjunto de Pareto permite modelar estos diferentes escenarios de una manera intuitiva.

Como se mostró, el problema sigue siendo cómo identificar automáticamente la mejor solución del Frente de Pareto, por lo que se considera un tema abierto para investigación futura.

Índice de tablas

Índice de figuras

Bibliografía

Alguliev, R. M., Aliguliyev, R. M., and Isazade, N. R. (2013). Multiple documents summarization based on evolutionary optimization algorithm. *Expert Systems with Applications*, 40(5):1675–1689. 45

Alguliyev, R. M., Aliguliyev, R. M., and Isazade, N. R. (2015). An unsupervised approach to generating generic summaries of documents. *Applied Soft Computing*, 34:236–250. 45

Alshargi, F., Shekarpour, S., Soru, T., and Sheth, A. (2018). Concept2vec: Metrics for evaluating quality of embeddings for ontological concepts. *arXiv preprint arXiv:1803.04488*.

Arora, S., Liang, Y., and Ma, T. (2016). A simple but tough-to-beat baseline for sentence embeddings. 27

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, volume 206, pages 14–21. 42

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155. 22, 23

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022. 19

Bollegala, D., Okazaki, N., and Ishizuka, M. (2010). A bottom-up approach to sentence ordering for multi-document summarization. *Information processing & management*, 46(1):89–109. 33

Caballero, R. and Hernández, M. (2003). El método de las ponderaciones en el problema fraccional lineal multiobjetivo. *Rect*, 11(1):1–11. 36

Cagnina, L., Esquivel, S. C., and Coello Coello, C. (2005). A particle swarm optimizer for multi-objective optimization. *Journal of Computer Science & Technology*, 5. 35

Canhasi, E. and Kononenko, I. (2014). Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. *Expert Systems with Applications*, 41(2):535–543.

Carbonell, J. and Jade, G. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval.*, pages 235–336. 11, 21, 44

Cheng, J. and Lapata, M. (2016). Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer. 35, 37, 42

Cohn, T. A. and Lapata, M. (2009). Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674. 16

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM. 26

Cunquero, R. M. (2003). Algoritmos heurísticos en optimización combinatoria. *Valencia: Universodad de Valencia. Retrieved*, 11(01):2012. 34

Dang, H. and Owczarzak, K. (2009). Overview of the tac 2009 summarization track. In *In Proceedings of the Second Text Analysis Conference (TAC2009).* 53

De Jong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, Ann Arbor, MI, USA. AAI7609381. 41

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197. 44

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407. 18

Dijk, T. A. v. (1988). News as discourse. 8

Dunlavy, D. M., O'Leary, D. P., Conroy, J. M., and Schlesinger, J. D. (2007). Qcs: A system for querying, clustering and summarizing documents. *Information processing & management*, 43(6):1588–1605. 33

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285. 5

Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 2*, pages 205–208. Association for Computational Linguistics. 16

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479. 16, 28

Feo, T. A. and Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133. 35

Fernández-Reyes, F. C. (2017). *Fusión de Datos basada en Representaciones Embebidas y Cópulas para la Recuperación de Información.* PhD thesis, Centro de Investigaciónń en Ciencias. Universidad Autónoma del Estado de Morelos. 60

Fernández-Reyes, F. C., Hermosillo-Valadez, J., and Montes-y Gómez, M. (2018). A prospect-guided global query expansion strategy using word embeddings. *Information Processing & Management*, 54(1):1–13. 59, 60, 72

Ferreira, R., de Souza Cabral, L., Freitas, F., Lins, R. D., de França Silva, G., Simske, S. J., and Favaro, L. (2014). A multi-document summarization system based on statistics and linguistic treatment. *Expert Systems with Applications*, 41(13):5780–5787. 16

Filippova, K. and Strube, M. (2008). Sentence fusion via dependency graph compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 177–185. Association for Computational Linguistics. 7

Fink, E. (2007). Evaluation of representations in ai problem solving. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, pages 349–353. IEEE. 15

Fonseca, C. M., Fleming, P. J., et al. (1993). Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *Icga*, volume 93, pages 416–423. Citeseer. 43

Freitas, A. A. (2004). A critical review of multi-objective optimization in data mining: a position paper. *ACM SIGKDD Explorations Newsletter*, 6(2):77–86. 13

Ganesan, K., Zhai, C., and Han, J. (2010). Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. 26

García-Hernández, R. A. and Ledeneva, Y. (2013). Single extractive text summarization based on a genetic algorithm. In *Mexican Conference on Pattern Recognition*, pages 374–383. Springer. 11

Gary, M. R. and Johnson, D. S. (1979). Computers and intractability: A guide to the theory of np-completeness. 34

Gawalt, B., Zhang, Y., and El Ghaoui, L. (2010). Sparse pca for text corpus summarization and exploration. In *NIPS 2010 Workshop on Low-Rank Matrix Approximation.*

Gendreau, M., Potvin, J.-Y., et al. (2010). *Handbook of metaheuristics*, volume 2. Springer. 35

Gillick, D., Favre, B., and Hakkani-Tür, D. (2008). The icsi summarization system at tac 2008. In *TAC*. 45

Glavaš, G. and Šnajder, J. (2014). Event graphs for information retrieval and multi-document summarization. *Expert systems with applications*, 41(15):6904–6916. 17

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549. 34, 35

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition. 43

Gong, H., Bhat, S., and Viswanath, P. (2017). Geometry of compositionality. In *Thirty-First AAAI Conference on Artificial Intelligence*. 28

Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. ACM. 10, 19

Goyal, P., Behera, L., and McGinnity, T. M. (2013). A context-based word indexing model for document summarization. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1693–1705. 17

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

Haghighi, A. and Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics.

Halliday, M. and Hasan, R. (1996). Cohesion in text. 8

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162. 18

Hennig, L. (2009). Topic-based multi-document summarization with probabilistic latent semantic analysis. In *Proceedings of the International Conference RANLP-2009*, pages 144–149. 20

Hill, F., Cho, K., and Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*. 12, 25

Hillier, F. S., Lieberman, G. J., and Osuna, M. A. G. (1997). *Introducción a la Investigación de Operaciones*, volume 1. McGraw-Hill. 34

Hinton, G. E., McClelland, J. L., Rumelhart, D. E., et al. (1984). *Distributed representations*. Carnegie-Mellon University Pittsburgh, PA. 22

Hobbs, J. R. (1985). On the coherence and structure of discourse. 9

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc. 19

Huang, L., He, Y., Wei, F., and Li, W. (2010). Modeling document summarization as multi-objective optimization. In *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*, pages 382–386. IEEE. 11, 45

Jones, K. S. et al. (1999). Automatic summarizing: factors and directions. *Advances in automatic text summarization*, pages 1–12. 5, 6

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Kågebäck, M., Mogren, O., Tahmasebi, N., and Dubhashi, D. (2014). Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 31–39. 26

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Khosla, N. K. (2015). Learning sentence vector representations to summarize. 27

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680. 35

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302. 12, 25, 27

Koopmans, T. (1951). Analysis of production as an efficient combination of activities. *Activity analysis of production and allocation, TC Koopmans, editor, Wiley, New York*. 35

Koza, J. R. (1997). Genetic programming. 39

Lang, H., Metzler, D., Wang, B., and Li, J.-T. (2010). Improved latent concept expansion using hierarchical markov random fields. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 249–258. ACM. 59

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196. 12, 25, 27

Lee, C. B., Kim, M. S., and Park, H. R. (2003). Automatic summarization based on principal component analysis. In *Portuguese Conference on Artificial Intelligence*, pages 409–413. Springer. 21

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain. 53

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78. Association for Computational Linguistics. 53

Lin, H. and Bilmes, J. (2011). A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics. 26

Longacre, R. E. (1979). The paragraph as a grammatical unit. *Syntax and semantics*, 12:115–134. 8

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165. 5

Mani, I. (2001). Summarization evaluation: An overview. 5, 10

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281. 8

Manning, C. D., Manning, C. D., and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press. 22

Marcu, D. (2000). The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational linguistics*, 26(3):395–448. 9

Martin, J. H. and Jurafsky, D. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River. 15

Mashreghi, Z. and Esmaeili, M. (2017). A combination method for improving text summarization. *International Journal of Computer Applications*, 159(9). 46

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. *Advances in Information Retrieval*, pages 557–564. 44

Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., and León, E. (2014). Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications*, 41(9):4158–4169. 17

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 16

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119. 11, 22, 23

Miranda-Jiménez, S., Gelbukh, A., and Sidorov, G. (2013). Summarizing conceptual graphs for automatic summarization task. In *International Conference on Conceptual Structures*, pages 245–253. Springer. 16

Mittal, J. G. V. and Callan, J. C. J. (2000). Creating and evaluating multi-document sentence extract summaries. In *Proceedings of the International Conference on Information and Knowledge Management: CIKM*, volume 9, page 165. ACM Press. 11

Moral, C., de Antonio, A., Imbert, R., and Ramírez, J. (2014). A survey of stemming algorithms in information retrieval. *Information Research: An International Electronic Journal*, 19(1):n1. 59

Morales, L. P., Esteban, A. D., and Gervás, P. (2008). Concept-graph based biomedical automatic summarization using ontologies. In *Proceedings of the 3rd textgraphs workshop on graph-based algorithms for natural language processing*, pages 53–56. Association for Computational Linguistics. 7

Morin, F. and Bengio, Y. (2005). Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, pages 246–252. Citeseer. 24

Mu, J., Bhat, S., and Viswanath, P. (2016). Geometry of polysemy. *arXiv preprint arXiv:1610.07569*. 28

Mu, J., Bhat, S., and Viswanath, P. (2017). Representing sentences as low-rank subspaces. *arXiv preprint arXiv:1704.05358*. 28, 29, 30, 64, 72

Murray, G., Renals, S., and Carletta, J. (2005). Extractive summarization of meeting recordings.

Nagesh, K. and Murty, M. N. (2012). Obtaining single document summaries using latent dirichlet allocation. In *International Conference on Neural Information Processing*, pages 66–74. Springer. 21

Nayeem, M. T. and Chali, Y. (2017). Extract with order for coherent multi-document summarization. *arXiv preprint arXiv:1706.06542*.

Ouyang, Y., Li, W., Li, S., and Lu, Q. (2011). Applying regression models to query-focused multi-document summarization. *Information Processing & Management*, 47(2):227–237. 10

Ozsoy, M. G., Alpaslan, F. N., and Cicekli, I. (2011). Text summarization using latent semantic analysis. *Journal of Information Science*, 37(4):405–417. 20

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab. 28

Paice, C. D. (1980). The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In *Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 172–191. Butterworth & Co. 7

Pembe, F. C. and Güngör, T. (2007). Automated querybiased and structure-preserving text summarization on web documents. In *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications, İstanbul*. 10

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.

Plaza, L., Lloret, E., and Aker, A. (2010). Improving automatic image captioning using text summarization techniques. In *International Conference on Text, Speech and Dialogue*, pages 165–172. Springer. 7

Radev, D. R., Jing, H., Styś, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938. 9, 50

Ren, P., Wei, F., Zhumin, C., Jun, M., and Zhou, M. (2016). A redundancy-aware sentence regression framework for extractive summarization. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 33–43.

Ren, W. and Han, K. (2014). Sentiment detection of web users using probabilistic latent semantic analysis. *Journal of Multimedia*, 9(10):1194–1200. 19

rey Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence*, volume 1, pages 82–87. Citeseer. 43

Riloff, E., Jones, R., et al. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479. 9

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*. 23

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,. 15

Saleh, H. H. and Kadhim, N. J. (2016). Extractive multi-document text summarization using multi-objective evolutionary algorithm based model. *Iraqi Journal of Science*, 57(1C):728–741. 45

Salton, G. and McGill, M. J. (1986). Introduction to modern information retrieval. 26

Sanderson, M. and Lawrie, D. (2002). Building, testing, and applying concept hierarchies. In *Advances in information retrieval*, pages 235–266. Springer. 59

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms, Pittsburgh, PA, USA, July 1985*, pages 93–100. 38

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press. 18

Sivic, J. and Zisserman, A. (2009). Efficient visual search of videos cast as text retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):591–606. 17

Smeaton, A. F. and Van Rijsbergen, C. J. (1983). The retrieval effects of query expansion on a feedback document retrieval system. *The Computer Journal*, 26(3):239–246. 59

Socher, R., Huang, E. H., Pennin, J., Manning, C. D., and Ng, A. Y. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems*, pages 801–809. 12, 25

Socher, R., Huval, B., Manning, C. D., and Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211. Association for Computational Linguistics.

Srinivas, N. and Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248. 43, 44

Steinberger, J. and Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4:93–100. 20

Steinberger, J. and Křišt'an, M. (2007). Lsa-based multi-document summarization. In *Proceedings of 8th International PhD Workshop on Systems and Control*, volume 7. 20

Steinberger, J., Poesio, M., Kabadjov, M. A., and Ježek, K. (2007). Two uses of anaphora resolution in summarization. *Information Processing & Management*, 43(6):1663–1680. 7

Sunar, M. and Kahraman, R. (2001). A comparative study of multiobjective optimization methods in structural design. *Turkish Journal of Engineering and Environmental Sciences*, 25(2):69–78. 37

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Tang, J., Yao, L., and Chen, D. (2009). Multi-topic based query-oriented summarization. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 1148–1159. SIAM.

Templeton, A. and Kalita, J. (2018). Exploring sentence vector spaces through automatic summarization. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 55–60. IEEE. 27

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep):1453–1484.

Vargas-Ocampo, E., Roman-Rangel, E., and Hermosillo-Valadez, J. (2018). Learning word and sentence embeddings using a generative convolutional network. In *Mexican Conference on Pattern Recognition*, pages 135–144. Springer.

Vechtomova, O., Robertson, S., and Jones, S. (2003). Query expansion with long-span collocates. *Information Retrieval*, 6(2):251–273. 59

Vikas, O., Meshram, A. K., Meena, G., and Gupta, A. (2008). Multiple document summarization using principal component analysis incorporating semantic vector space model. *International Journal of Computational Linguistics & Chinese Language Processing, Volume 13, Number 2, June 2008*, 13(2):141–156. 21

VS, R. K. (2016). A survey on text summarization using optimization algorithm. *ELK Asia Pacific Journal of Computer Science and Information Systems*. 49

Wan, X. (2008). Using only cross-document relationships for both generic and topic-focused multi-document summarizations. *Information Retrieval*, 11(1):25–49. 10

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83. 55

Wong, K.-F., Wu, M., and Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 985–992. Association for Computational Linguistics. 33

Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control*, 8(1):59–60. 35

Zanakis, S. H. and Evans, J. R. (1981). Heuristic "optimization": Why, when, and how to use it. *Interfaces*, 11(5):84–91. 34

Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731. 45

Zhao, L., Wu, L., and Huang, X. (2009). Using query expansion in graph-based approach for query-focused multi-document summarization. *Information Processing & Management*, 45(1):35–41. 7

Zhao, X. and Tang, J. (2010). Query-focused summarization based on genetic algorithm. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, volume 2, pages 968–971. IEEE. 45

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195. 13

DR. VICTOR BARBA LÓPEZ
COORDINADOR DEL POSGRADO EN CIENCIAS
PRESENTE

Atendiendo a la solicitud para emitir DICTAMEN sobre la revisión de la TESIS titulada **Generación Automática de Resúmenes basada en representaciones embebidas y optimización multiobjetivo** que presenta la alumna **Yanet Fors Isalguez (5920150301)** para obtener el título de **Doctor en Ciencias.**

Nos permitimos informarle que nuestro voto es:

NOMBRE	DICTAMEN	FIRMA
Dr. Markus Franziskus Müller CINC-UAEM	Aprobado	
Dr. Antonio Daniel Rivera López CINC-UAEM	Aprobado	
Dr. Manuel Montes y Gómez INAOE	Aprobado	
Dr. Edgar Francisco Román Rangel ITAM	Aprobado	
Dra. Lorena Díaz González CINC-UAEM	Aprobado	
Dr. Juan Manuel Rendón Mancha CINC-UAEM	Aprobado	
Dr. Jorge Hermosillo Valadez CINC-UAEM	Aprobado	