Universidad de La Habana
Facultad de Matemática y Computación

Recuperación Semántica de Información: Un enfoque integrado.

Autor:
Laura Victoria Riera Pérez

Tutores:

Lic. Carlos León González
Dra. C. Lucina García Hernández

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

8 de enero de 2024

A mis padres, que me dieron el cielo para volar.
A todos aquellos que, con su apoyo,
han sido el viento bajo mis alas.

Agradecimientos

Estos cuatro años de estudio de las Ciencias de la Computación han sido igualmente duros y hermosos. En este laberinto de algoritmos y lógica, he encontrado no solo la base para una carrera profesional, sino un camino que ha moldeado mi manera de pensar y ver el mundo.

Quiero agradecer en primer lugar a mis padres, pilares fundamentales de mi vida. Gracias por siempre creer en mí, incluso cuando yo no lo he hecho. Son ustedes, con su ejemplo y amor incondicional, quienes me han inspirado a esforzarme y superar cada desafío. Este sueño, hoy convertido en logro es también suyo. No puedo expresar con palabras lo mucho que los amo.

A mi facultad, MATCOM, y a toda su gente linda. Han creado un ambiente donde reina la paz, el aprendizaje y el trabajo duro. A los profesores, cuyo esfuerzo y dedicación han ayudado a hacer de mi la persona que soy hoy. En especial quiero agradecer a mis tutores, Carlos León González y Lucina García Hernández. Su cálida acogida en el departamento y su apoyo constante han sido muy valiosos para mí. Aprecio enormemente su comprensión y orientación durante este tiempo.

Por último, pero ciertamente muy importante, quiero agradecer a mi familia y amigos, los que siempre han estado para apoyarme, tanto cerca como en la distancia. Cada risa compartida y cada palabra de ánimo han sido faros de luz en mi camino. Este logro también lleva impreso el sello de su amor y confianza.

Opinión de los tutores

Opiniones de los tutores

Resumen

Resumen en español

Abstract

Resumen en inglés

Índice general

Índice de figuras

Ejemplos de código

Introducción

A lo largo de los siglos, mentes ilustres como las de Descartes, Newton y Bacon han tejido la noción cautivadora de que *el conocimiento es poder*. En la construcción de este conocimiento, la información desempeña un papel fundamental, siendo el material primario del cual se extraen ideas, conceptos y comprensiones profundas. En la sociedad contemporánea, este papel adquiere una relevancia sin precedentes, consolidándose la información como una fuerza motriz esencial que impulsa los engranajes del progreso y facilita la toma de decisiones cruciales.

El origen de la Recuperación de Información, ese arte y ciencia de extraer conocimiento de vastos conjuntos de datos, estuvo impulsado por la necesidad de superar desafíos en el acceso a información. La gestión manual de registros, especialmente en el ámbito de las publicaciones científicas y los archivos de bibliotecas [1], implicaba una labor intensiva y propensa a errores.

A medida que las computadoras comenzaron a desarrollarse y evolucionar, se reconocieron sus capacidades para manejar grandes volúmenes de datos y facilitar la recuperación de información. El uso de computadoras con este propósito se remonta a mediados del siglo XX. Durante la Segunda Guerra Mundial, Alan Turing y las computadoras británicas Colossus fueron fundamentales para procesar y descifrar mensajes encriptados nazis [2]. En los 1950s, se implementaron sistemas como el General Electric, que buscaba más de 30,000 resúmenes de documentos, representando un hito inicial en el uso de computadoras para gestionar grandes conjuntos de información [3]. Durante la década de 1960, se destacaron avances en la formalización de algoritmos para clasificar documentos en relación con una consulta. Un enfoque destacado consideraba documentos y consultas como vectores en un espacio N-dimensional [3]. Durante los años 1970, se produjeron avances significativos, como la complementación de los pesos de frecuencia de término de Luhn, basados en la ocurrencia de palabras dentro de un documento, con el trabajo de Spärck Jones sobre la ocurrencia de palabras en el conjunto de documentos de una colección [3]. Con el auge de las computadoras personales en los 1980s [3] y la irrupción de la World Wide Web en 1991 [4], se transformó radicalmente el panorama, con hitos como Yahoo! en 1994 [5] y el algoritmo PageRank de Google en 1996 [3], marcando un nuevo paradigma orientado a la web. En los 2000 se presenció una transición hacia la personalización

y la búsqueda semántica, con eventos destacados como el lanzamiento del algoritmo Hummingbird por Google en 2013 [6].

A partir de 2020, las tendencias fundamentales en la evolución de la Recuperación de Información han impulsado una exploración profunda en áreas como la Inteligencia Artificial, el aprendizaje de máquinas y el procesamiento del lenguaje natural, con el objetivo de perfeccionar la precisión de los resultados de búsqueda. Este período ha sido testigo del surgimiento de enfoques innovadores como la búsqueda conversacional y la generalización a la gran mayoría de aplicaciones de la personalización y recomendación de contenido, mejorando directamente la experiencia del usuario; así como la representación semántica y relacional, integrando la esencial comprensión contextual. Además, se han explorado métodos para la extracción de patrones y relaciones complejas, y la clasificación y organización temática.

La transformación significativa de la Recuperación de Información, ha pasado de ser un ámbito exclusivo de profesionales a involucrar a cientos de millones de personas en la búsqueda diaria de información, provocando un impacto profundo en diversas esferas. En el ámbito académico, los Sistemas de Recuperación de Información (SRI) no solo agilizan la investigación, sino que también fomentan la colaboración entre investigadores y facilitan el acceso a recursos compartidos, promoviendo así la difusión eficiente de conocimientos. Paralelamente, en el entorno empresarial, estos sistemas contribuyen a la productividad simplificando la búsqueda de información esencial y respaldando decisiones estratégicas. Asimismo, en redes sociales, personalizan la experiencia del usuario y fortalecen la conexión con recomendaciones adaptadas, mientras que en la industria médica, agilizan diagnósticos y tratamientos. También, en el ámbito cultural, contribuyen a la preservación y acceso a archivos históricos y museos virtuales.

A pesar de estos avances notables, persisten desafíos significativos en los SRI. La ambigüedad semántica, consultas vagas que dificultan la comprensión precisa de la intención del usuario, junto con problemas de relevancia, como el ruido de información y la sensibilidad al contexto, plantean obstáculos a la eficacia de la recuperación de información. Limitaciones tecnológicas, como la dificultad para manejar información multimedia y desafíos en el procesamiento del lenguaje natural, también presentan retos. Además, las preocupaciones éticas, la privacidad del usuario y la adaptación a nuevos contenidos emergentes son áreas críticas a abordar.

En el colectivo de Sistemas de Información de la Facultad de Matemática y Computación (MATCOM) de la Universidad de La Habana, se evidencia una rica trayectoria de investigación y logros en diversas temáticas. Entre los antecedentes, se encuentran trabajos como el presentado por Quintana Wong, García Hernández, Guillot Jiménez y Amable Ambrós en COMPUMAT 2019, que abordó recomendaciones para la promoción de la salud mediante el uso de bases de datos NoSQL [7]. También, se resalta la investigación de Quintana Wong, García Garrido y García Hernández

en IWOR 2019 [8], donde integraron modelos de vecindario y factores latentes para obtener recomendaciones precisas. La participación en eventos internacionales, como la Escuela Latinoamericana de Verano en Investigación Operativa (ELAVIO) 2019 en Lleida, España, refleja el compromiso del colectivo con trabajos que exploraron la combinación de filtrado colaborativo y modelación de tópicos para la recomendación de información [9]. En ese mismo año también se destaca el trabajo de Leon González y García Garrido, el cual abordó modelos de generación de tópicos con word embedding, explorando modelos probabilistas como LDA y presentando el algoritmo lda2vec [10]. Además, Quintana-Wong, García Hernández, y colaboradores han contribuido con artículos en revistas especializadas, como el estudio sobre sistemas de recomendación en soluciones analíticas transaccionales para la atención médica [7]. Prado Romero y colaboradores, por su parte, han destacado en la predicción de la popularidad de temas en proveedores de noticias, presentando sus avances en ICOR 2020 y con un artículo aceptado en el "Intelligent Data Analysis Journal" [11]. Asimismo, se destaca la colaboración internacional en un proyecto conjunto con profesores de la Universidad de L'Aquila, Italia, para desarrollar un Sistema de Recuperación de Información relacionada con la COVID-19, llevando a cabo esta iniciativa de manera efectiva durante los meses de mayo a julio de 2020.

El presente trabajo se centrará en uno de los enfoques fundamentales de la clasificación y organización temática: los modelos de tópicos. Estos modelos matemáticos poseen la capacidad de descubrir patrones temáticos subyacentes y organizar documentos de manera automática según su tema, proporcionando información sobre las palabras que componen cada uno. Sin embargo, enfrentan desafíos actuales, como la adaptación a contextos cambiantes, ya que dependen de hiperparámetros cruciales, cuya configuración se basa en información extraída del corpus y que impacta directamente en la deducción de las temáticas. Este ajuste no es automático y depende del análisis experto para su determinación. En entornos dinámicos, donde la cantidad y naturaleza de los tópicos pueden cambiar con el tiempo, esta adaptación constante puede representar un problema. Otro desafío es la subjetividad en la interpretación de temas, ya que no proporcionan un nombre específico para los tópicos y requieren la intervención de expertos para su identificación. Presentan también problemas en la gestión efectiva de la polisemia, la cual implica distinguir entre los diversos significados de una palabra para una interpretación precisa del tópico. La mejora de la robustez y la capacidad para manejar la variabilidad del lenguaje son áreas cruciales de investigación para perfeccionar los modelos de tópicos.

Por tanto, los problemas científicos a abordar son la dificultad de los modelos de tópicos para resolver eficazmente el ajuste automático de hiperparámetros en entornos dinámicos, así como la limitación para la precisión semántica de los tópicos.

Considerando los desafíos anteriormente mencionados la pregunta científica a responder en esta tesis es, ¿puede la implementación de mejoras en los modelos de

tópicos, específicamente en la adaptación a contextos cambiantes y la gestión de la subjetividad en la interpretación de temas, fortalecer la eficacia de los Sistemas de Recuperación de Información?

El objetivo general de este trabajo consiste en concebir, diseñar e implementar una solución computacional que automatice los procesos de los modelos de tópicos, contribuyendo a la adaptación a contextos cambiantes y la gestión de la subjetividad en la interpretación de temas.

Para alcanzar el cumplimiento del objetivo general, se proponen los siguientes objetivos específicos:

1. Profundizar en el marco teórico y conceptual de los SRI, dando prioridad a la comprensión de modelos de tópicos y ontologías.

2. Realizar un estudio del estado actual en entornos dinámicos y en la literatura académica sobre SRI, identificando tendencias clave.

3. Concebir y diseñar estrategias para potenciar la adaptabilidad de los modelos de tópicos en entornos dinámicos y perfeccionar la interpretación semántica de temas.

4. Implementar y evaluar las estrategias concebidas, para medir efectividad en el contexto de adaptabilidad y precisión semántica.

El contenido restante de esta tesis se organiza en cuatro capítulos, abarcando las distintas etapas que constituyen el desarrollo del trabajo. En el Capítulo 1, “Marco Teórico-Conceptual”, se proporciona un análisis detallado del estado actual de la ciencia y tecnología en las áreas relevantes, sirviendo como fundamento esencial para la investigación y los resultados obtenidos. El Capítulo 2, “Concepción y Diseño”, aborda la caracterización general de la propuesta computacional, la arquitectura del sistema, la estructura del modelo analítico y los procesos vinculados a la precisión semántica. Detalles técnicos de la implementación del sistema se presentan en el Capítulo 3, “Detalles de Implementación”, y en el Capítulo 4 “Experimentación” se explora cualitativa y experimentalmente la validez de la solución implementada. En la parte del desenlace, se exponen las Conclusiones, destacando los logros clave en relación con los objetivos planteados, así como las Recomendaciones que señalan futuras direcciones de investigación. La Bibliografía utilizada para respaldar la base científica de la solución propuesta se incluye para facilitar la exploración de temas relacionados.

Capítulo 1

Marco Teórico - Conceptual

En la esfera contemporánea de la gestión de la información, los Sistemas de Recuperación de Información desempeñan un papel fundamental. Estos sistemas, que han evolucionado significativamente a lo largo de las últimas décadas, son cruciales para el manejo eficiente de la creciente marea de datos digitales. Estos sistemas no solo facilitan el acceso rápido a información relevante sino que también contribuyen a la organización y estructuración de grandes volúmenes de datos [1]. Esta relevancia se extiende a través de una variedad de campos, desde la bibliotecología y la informática hasta los negocios, donde la capacidad de recuperar y clasificar información de manera efectiva se ha convertido en una herramienta indispensable para la toma de decisiones y el conocimiento estratégico.

Los sistemas de recuperación de información se han diversificado en varias vertientes especializadas, reflejando la complejidad y diversidad de las necesidades en este campo. Un componente esencial en la recuperación de información textual es el Procesamiento del Lenguaje Natural (NLP). El NLP permite a las máquinas entender, interpretar y manipular el lenguaje humano, proporcionando una comprensión del significado y el contexto, crucial para superar desafíos como la ambigüedad y la variedad idiomática El NLP ha revolucionado la búsqueda y recuperación de información, permitiendo análisis y síntesis más precisos y rápidos, esenciales en campos como la investigación académica. Además, ha mejorado la accesibilidad, facilitando interfaces de usuario más naturales e intuitivas, como en sistemas de preguntas y respuestas y asistentes virtuales.

Dentro de la recuperación textual se pueden encontrar campos como la recuperación web abordando aspectos como el ranking de páginas y la optimización para motores de búsqueda [12], y la recuperación bibliográfica centrada en la organización y el acceso a la literatura académica y científica. Por otro lado, la recuperación de información multimedia, se ocupa de contenido que incluye imágenes, audio y video [13]. También está la recuperación de información geográfica, crucial en aplicaciones

como los sistemas de información geográfica como GEMET[1] y los servicios basados en la localización [14].

En el presente capítulo, se discuten brevemente componentes clave dentro del campo de la recuperación textual que constituyen la base teórica de esta investigación: los modelos de tópicos, ontologías y *embeddings*, así como los antecedentes relevantes para la automatización de la estimación del número de tópicos en un corpus y la asignación de nombres a los mismos.

1.1. Modelos de Tópicos

Los modelos de tópicos, fundamentales en NLP y la Recuperación de Información, se destacan por su capacidad para explorar y organizar grandes conjuntos de datos textuales. Estos modelos matemáticos operan identificando tópicos subyacentes en colecciones de documentos, extrayendo patrones significativos en el uso de palabras y revelando así las estructuras temáticas latentes. Esta funcionalidad los convierte en herramientas esenciales para comprender, categorizar y sintetizar información en grandes volúmenes de texto.

El desarrollo histórico de los modelos de tópicos es un viaje fascinante a través de la evolución del NLP. Comenzando con el Modelo de Análisis Semántico Latente (LSA) en 1990 [15], un método pionero que utilizaba la descomposición en valores singulares para identificar estructuras semánticas en grandes colecciones de texto. Esta técnica fue fundamental para sentar las bases de los modelos de tópicos. El Modelo de Análisis Semántico Latente Probabilístico (pLSA), propuesto por Hofmann en 1999 [16], representó un avance significativo, introduciendo un enfoque probabilístico para modelar la relación entre documentos y tópicos. Sin embargo, pLSA tenía limitaciones, especialmente en la generalización a documentos no vistos durante el entrenamiento.

El gran avance llegó con la introducción del Modelo de la Asignación Latente de Dirichlet (LDA) [17], considerando cada documento como una mezcla de tópicos latentes, donde cada tópico está definido por una distribución sobre los tokens del vocabulario (palabras). Formalmente, para cada documento d, LDA asume una distribución de tópicos θ_d que se extrae de una distribución a priori de Dirichlet. Para cada palabra en el documento, se elige un tópico z de la distribución de tópicos θ_d. Luego, se selecciona una palabra de una distribución de palabras asociada a ese tópico específico. Este proceso se repite a lo largo de todos los documentos y palabras, iterando para ajustar las distribuciones de tópicos y palabras hasta que el modelo refleje adecuadamente la estructura latente de tópicos en los documentos. Este modelo generativo probabilístico ofreció una mayor flexibilidad y capacidad de interpretación,

[1] https://www.eionet.europa.eu/gemet/en/about/

convirtiéndose en el estándar de oro para el modelado de tópicos.

Desde entonces, los modelos de tópicos han continuado evolucionando, integrando enfoques más sofisticados como el Modelo de Tópicos Correlacionados (CTM) [18] y el Modelo de Tópicos Dinámicos (DTM) [19]. El CTM, al incorporar correlaciones entre tópicos, ofrece una representación más matizada y realista de cómo se distribuyen los tópicos en documentos, mientras que el DTM introduce una perspectiva temporal, analizando cómo los tópicos evolucionan con el tiempo. Además, los modelos jerárquicos como el Hierarchical Latent Dirichlet Allocation (HLDA) [20] han proporcionado una estructura más compleja, permitiendo la detección de tópicos en distintos niveles de granularidad. Estos avances han enriquecido el análisis de tópicos, ofreciendo una visión más profunda y detallada de las estructuras temáticas en grandes volúmenes de texto.

1.2. Ontologías

Las ontologías, en el contexto de la informática y NLP, son estructuras de datos que representan conocimientos de manera organizada y jerárquica. Una ontología define un conjunto de conceptos y categorías que representan un dominio de conocimiento, así como las relaciones entre estos conceptos [21]. Computacionalmente, las ontologías se manejan frecuentemente como grafos, donde los nodos representan conceptos o entidades, y las aristas o bordes representan las relaciones entre ellos. Las ontologías permiten que las máquinas "comprendan" y procesen el significado de la información de manera más eficaz, fundamental para la extracción y clasificación de información, donde se requiere no solo identificar datos, sino también comprender su semántica.

Las ontologías son creadas principalmente por expertos en un dominio específico, con el objetivo de facilitar la comunicación y la comprensión común en diferentes campos [22]. Estos expertos utilizan las ontologías para establecer un marco conceptual compartido, lo que ayuda a superar las barreras de comunicación y a mejorar la interoperabilidad entre sistemas y organizaciones, lo cual es fundamental en áreas como la ingeniería de sistemas, la integración empresarial y el desarrollo de software.

Pueden ser clasificadas como ontologías de dominio específico y ontologías generales. Las ontologías de dominio específico se enfocan en áreas de conocimiento particulares, brindando un marco detallado para las entidades y relaciones dentro de ese ámbito específico. Por ejemplo, en el campo de la medicina, Medical Subject Headings (MeSH)[2] es una ontología de dominio específico que detalla la terminología, las relaciones y los procesos relacionados con diversos aspectos de la medicina y las ciencias de la salud. Este tipo de ontología es de gran utilidad en aplicaciones médicas

[2]https://www.nlm.nih.gov/mesh/meshhome.html

y de investigación, proporcionando una estructura precisa para la gestión y el análisis de datos complejos relacionados con enfermedades, tratamientos y la biomedicina en general.

En contraste, las ontologías generales abarcan un conocimiento más amplio, estableciendo un marco general para clasificar y relacionar conceptos de varios dominios. Son esenciales en aplicaciones que demandan un entendimiento generalizado del conocimiento humano. WordNet[3] es un ejemplo significativo en esta categoría; es una base de datos léxica en inglés que organiza palabras en conjuntos de sinónimos (synsets), definiendo relaciones como sinonimia, antonimia y jerarquías de hipónimos (especificaciones) e hiperónimos (generalizaciones).

La integración de las ontologías proporciona una mayor profundidad en el análisis de textos, permitiendo la creación de metadatos más ricos y relevantes, lo que mejora sustancialmente la recuperación y gestión de información en bases de datos y repositorios digitales. Mejoran la accesibilidad y gestión de información en la web, facilitando búsquedas más eficientes y una navegación intuitiva [23]. Se utilizan para la agrupación [24, 25] y la anotación de documentos [26], así como para la similitud semántica [27], las cuales mejoran la precisión en la categorización y búsqueda de documentos, al permitir a los sistemas informáticos identificar conexiones y similitudes en el contenido textual basándose en significados subyacentes.

1.3. Vectores con contenido semántico: *embeddings*

Los *embeddings*, en el contexto de NLP, son representaciones vectoriales de palabras o frases. Estos son vectores densos y distribuidos de longitud fija, construidos utilizando estadísticas de co-ocurrencia de palabras [28]. Estas representaciones codifican información sintáctica y semántica, transformando el texto en una forma manejable para los algoritmos, facilitando tareas como clasificación de texto, análisis de sentimientos y traducción automática.

Existen varios tipos de *embeddings*, cada uno con características únicas. Los *word embeddings*, con modelos como Word2Vec [29] y GloVe [30], representan palabras individuales como vectores en un espacio multidimensional, capturando su significado y relaciones sintácticas. Los *sentence embeddings*, por otro lado, representan oraciones enteras, permitiendo capturar el contexto más amplio de la oración. Los *contextual embeddings* como los generados por modelos como BERT [31], van un paso más allá, representando palabras en el contexto de frases o párrafos, lo que permite una comprensión más matizada del significado, especialmente para palabras con múltiples interpretaciones.

Los *embeddings* tienen amplias aplicaciones en el análisis de textos y relaciones

[3] https://wordnet.princeton.edu/

semánticas. Por ejemplo, se han utilizado para evaluar la coherencia de temas en datos de Twitter [32], lo que permite medir con precisión la coherencia y relevancia de los tópicos generados, crucial para la interpretación efectiva de grandes conjuntos de datos sociales. También, se aplican en la expansión de consultas en motores de búsqueda [33], ofreciendo una búsqueda más rica y contextualizada. Además, los *embeddings* de palabras son útiles para medir distancias entre documentos [34] y se emplean en el análisis de texto basado en relaciones semánticas para mejorar la comprensión de tópicos específicos en grandes volúmenes de datos [35].

1.4. Estado del arte

La sección del estado del arte explora desarrollos recientes y metodologías clave en los dos problemas centrales a abordar en este trabajo: la automatización de la identificación del número de tópicos en un corpus y de la asignación de nombres a tópicos.

1.4.1. Identificación del número de tópicos presentes en un corpus

La identificación del número de tópicos en un corpus es esencial en el análisis de datos y el NLP, ya que define la estructura y claridad en la interpretación de grandes volúmenes de texto. Hasta ahora, este proceso depende en gran medida de la intervención de expertos, lo que conlleva una subjetividad inherente y limita la escalabilidad y consistencia en el análisis. Un enfoque automatizado y objetivo podría adaptarse mejor a la naturaleza dinámica y variable de la información, permitiendo a los analistas centrarse más en la interpretación y aplicación de los resultados.

El enfoque de Arun et al. (2010) en "On Finding the Natural Number of Topics with Latent Dirichlet Allocation" [36] se centra en determinar el número óptimo de tópicos para modelos LDA. Proponen una medida basada en la divergencia simétrica de Kullback-Leibler de las distribuciones salientes de los factores de la matriz LDA. Esta medida identifica el número de tópicos observando un 'pico' en los valores de divergencia para el número correcto de tópicos.

Por otro lado, el artículo de Gan y Qi (2021) en "Selection of the Optimal Number of Topics for LDA" [37] se centra en una metodología integral para determinar el número óptimo de tópicos en LDA. Presentan un índice que evalúa varios factores: perplejidad, aislamiento de tópicos, estabilidad y coincidencia. Este índice tiene como objetivo lograr una alta capacidad predictiva, un buen aislamiento entre tópicos, evitar tópicos duplicados y asegurar la repetibilidad.

Vangara et al. (2021) en "Finding the Number of Latent Topics With Semantic

Non-Negative Matrix Factorization" [38] introduce SeNMFk, una metodología que combina la factorización de matrices no negativas (NMF) con información semántica para determinar el número óptimo de tópicos. SeNMFk utiliza la divergencia de Kullback-Leibler y se enfoca en la estabilidad de los tópicos a través de un ensamble aleatorio de matrices. Además, presentan el software pyDNMFk para facilitar la estimación del número de tópicos.

Los siguientes artículos, aunque no persiguen el objetivo de identificar automáticamente el número de tópicos presentes en el corpus, son relevantes ya que emplean razonamientos básicos, análogos a los utilizados en esta investigación. El enfoque de Jiang et al. (2011) en "A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification" [39] se basa en un algoritmo de agrupación de características autoconstruido difuso. Este método se enfoca en identificar estructuras latentes en datos de texto, utilizando técnicas de agrupación difusa para manejar la incertidumbre y la ambigüedad inherentes a los datos textuales. Permite una clasificación más flexible y adaptativa de los datos, lo que puede ser especialmente útil en aplicaciones donde las categorías no son claramente definidas o donde los datos pueden pertenecer a múltiples categorías simultáneamente, sin tener que especificar el número de grupos como parámetro de entrada. Por otro lado, el enfoque de Thompson y Mimno (2020) en "Topic Modeling with Contextualized Word Representation Clusters" [40] investiga el uso de agrupaciones de representaciones de palabras contextualizadas, como las de BERT, para el modelado de tópicos. Su metodología se basa en la hipótesis de que estas representaciones contextualizadas pueden capturar polisemia y proporcionar información sintáctica más rica, lo que resulta en una organización de documentos similar a la obtenida con modelos LDA tradicionales.

1.4.2. Asignación de nombres a grupos

La asignación automática de nombres a tópicos en un corpus es de gran importancia pues facilita la comprensión e interpretación de los resultados generados por los modelos de tópicos. Actualmente, esta tarea también se realiza principalmente a través de la intervención de expertos, un proceso que puede ser subjetivo y laborioso. La automatización de este proceso permitiría una identificación de tópicos más coherente y objetiva, reduciendo la carga de trabajo manual y aumentando la capacidad para manejar abundante información.

El documento "An Ontology-based Semantic Tagger for IE system" de Boufaden (2003) [41], detalla un sistema para etiquetar semánticamente conversaciones en el ámbito de búsqueda y rescate marítimo, y facilitar el proceso de la extracción de información. La metodología combina dos fuentes de conocimiento: una ontología de búsqueda y rescate, y el diccionario-tesauro Wordsmyth[4]. El proceso se divide en

[4]https://www.wordsmyth.net/

cuatro etapas: extracción de palabras candidatas, anotación semántica, filtrado contextual, y utilización de las palabras anotadas para la resolución de correferencias y el llenado de plantillas de extracción de información. Se ejemplifica con un diálogo donde se etiquetan palabras clave con etiquetas específicas del dominio, como el incidente (MISSING-VESSEL), la localización (LOCATION-TYPE) y las condiciones climatológicas (WEATHER-TYPE).

En "Ontology based Web Page Topic Identification" de Singh Rathore y Roy [42] se describe un método para identificar tópicos en páginas web usando una ontología de dominio específico desarrollada manualmente con este propósito. El proceso inicia con la extracción de palabras clave de tags HTML y la detección de co-ocurrencias de palabras en el texto. Primero, se mapean las palabras clave extraídas de la página web a conceptos en la ontología, utilizando la distancia de Levenshtein para evaluar su similitud. Las palabras clave se clasifican por relevancia, y se establece un umbral para determinar la adecuación del mapeo. En la primera fase, se consideran las palabras clave más relevantes, y si su correspondencia con un nodo de la ontología es significativa, se sugiere que la página pertenece a ese tema. Si no se alcanza este umbral, la segunda fase incluye todas las palabras clave. Si la combinación de todas las palabras clave mapeadas supera el umbral para un nodo, entonces se asigna ese tema al documento.

El trabajo de Saqlain et al. (2016) [43] sigue un enfoque que utiliza WordNet y TF-IDF para asignar nombres automáticamente a tópicos, combinando técnicas semánticas y estadísticas. El proceso inicia con una agrupación jerárquica de los textos y su posterior procesamiento, incluyendo la eliminación de palabras comunes y la estandarización de términos. Posteriormente, se identifican términos clave en cada grupo utilizando el cálculo de Term Frequency-Inverse Document Frequency (TF-IDF). Estos términos son procesados a través de WordNet para generar sus hiperónimos, y la frecuencia de estos hiperónimos se calcula dentro del grupo. Los hiperónimos que aparecen con mayor frecuencia se seleccionan como etiquetas para los grupos.

Desambiguación del sentido de las palabras

La desambiguación de sentidos de palabras (WSD) es un campo de la lingüística computacional y el NLP que se enfoca en asignar significados precisos a palabras en contextos específicos, diferenciando entre múltiples significados o sentidos. Esto se hace particularmente necesario al utilizar ontologías pursto que una palabra puede corresponder a múltiples conceptos en la ontología, cada uno con un significado distinto de acuerdo al contexto.

El algoritmo de Lesk (1986) [44], es una técnica clásica para WSD. El algoritmo opera comparando las definiciones de una palabra objetivo con las palabras presentes

en su contexto inmediato. La definición que tenga el mayor solapamiento es elegida como la más probable. A pesar de su simplicidad, el algoritmo de Lesk ha sido fundamental en el desarrollo de técnicas más avanzadas de WSD, y ha sido implementado en discímiles bibliotecas de NLP. Tiene limitaciones evidentes, como su dependencia de coincidencias exactas de palabras y la posibilidad de que el contexto no ofrezca suficiente información para un solapamiento significativo, lo que puede resultar en la elección de sentidos incorrectos para las palabras. Desde entonces han surgido modificaciones de este algoritmo para añadir semántica al proceso.

Ambos, Edmonds y Agirre (2008) [45], y Bevilacqua et al. (2021) [46], proporcionan un análisis de los algoritmos y aplicaciones en WSD. Los algoritmos basados en grafos son fundamentales dado que estas son las estructuras generalmente adoptadas en los enfoques basados en el conocimiento, como WordNet. Estos algoritmos ayudan a establecer conexiones entre diferentes sentidos de palabras basándose en sus relaciones semánticas. Por otro lado, en los enfoques supervisados, los modelos neuronales, especialmente aquellos que utilizan arquitecturas de Transformer preentrenadas, han demostrado ser altamente efectivos. Estos modelos aprenden a asociar palabras en contextos específicos con sus sentidos correspondientes, utilizando grandes conjuntos de datos anotados. Además, las técnicas que incorporan definiciones textuales de inventarios de sentidos, como SensEmBERT [47], también han mostrado un rendimiento sobresaliente en WSD. Estos métodos aprovechan las representaciones contextualizadas de palabras y los *embeddings* de sentido para mejorar la precisión de la desambiguación.

También se destaca el uso de inteligencia artificial y metaheurísticas para WSD. El artículo de AL-Saiagh et al. (2018) [48] introduce un enfoque híbrido que combina la optimización de enjambres de partículas (PSO) con el recocido simulado, mientras que "A Self-adaptive Genetic Algorithm for the Word Sense Disambiguation Problem" de Wojdan Alsaeedan y Mohamed El Bachir Menai (2015) [49] explora el uso de un algoritmo genético autoadaptativo que ajusta automáticamente las probabilidades de cruce y mutación optimizando el proceso.

Capítulo 2

Concepción y diseño de la solución

La arquitectura del programa se presenta como una combinación de componentes interconectados, cada uno contribuyendo a la extracción y comprensión de información significativa de conjuntos de datos textuales. Se concibe un sistema que comprende dos etapas: el Pre-procesamiento Semántico y la Recuperación por tópicos (ver figura 2.1).

La primera etapa, Pre-procesamiento Semántico, inicia con un pre-procesamiento léxico básico, estructurando el texto para las operaciones subsiguientes. La Identificación de la Cantidad de Tópicos adopta un enfoque de agrupación semántica y *embeddings*, con el objetivo de eliminar la necesidad de intervención de expertos para obtener información sobre la distribución del corpus. Esta fase se entrelaza con el modelo de Latent Dirichlet Allocation (LDA) para desentrañar patrones temáticos. La transición hacia la Asignación de Nombres a Tópicos conecta el descubrimiento de tópicos con una ontología de dominio general, seleccionando palabras clave basadas en las probabilidades de LDA, en busca de facilitar una asignación contextualizada de nombres de forma automática.

En pos de aplicación y visualización de la clasificación en la etapa anterior se concibe en la segunda etapa, Recuperación por Tópicos, un motor de búsqueda que, al recibir una consulta, devuelva resultados organizados según su relevancia y tópicos asociados (nombrados), permitiendo una exploración más interpretable.

Esta investigación centra su atención en la primera etapa del sistema, detallando el diseño de cada uno de sus componentes en este capítulo. Al final, también se aborda brevemente la idea de la segunda etapa.

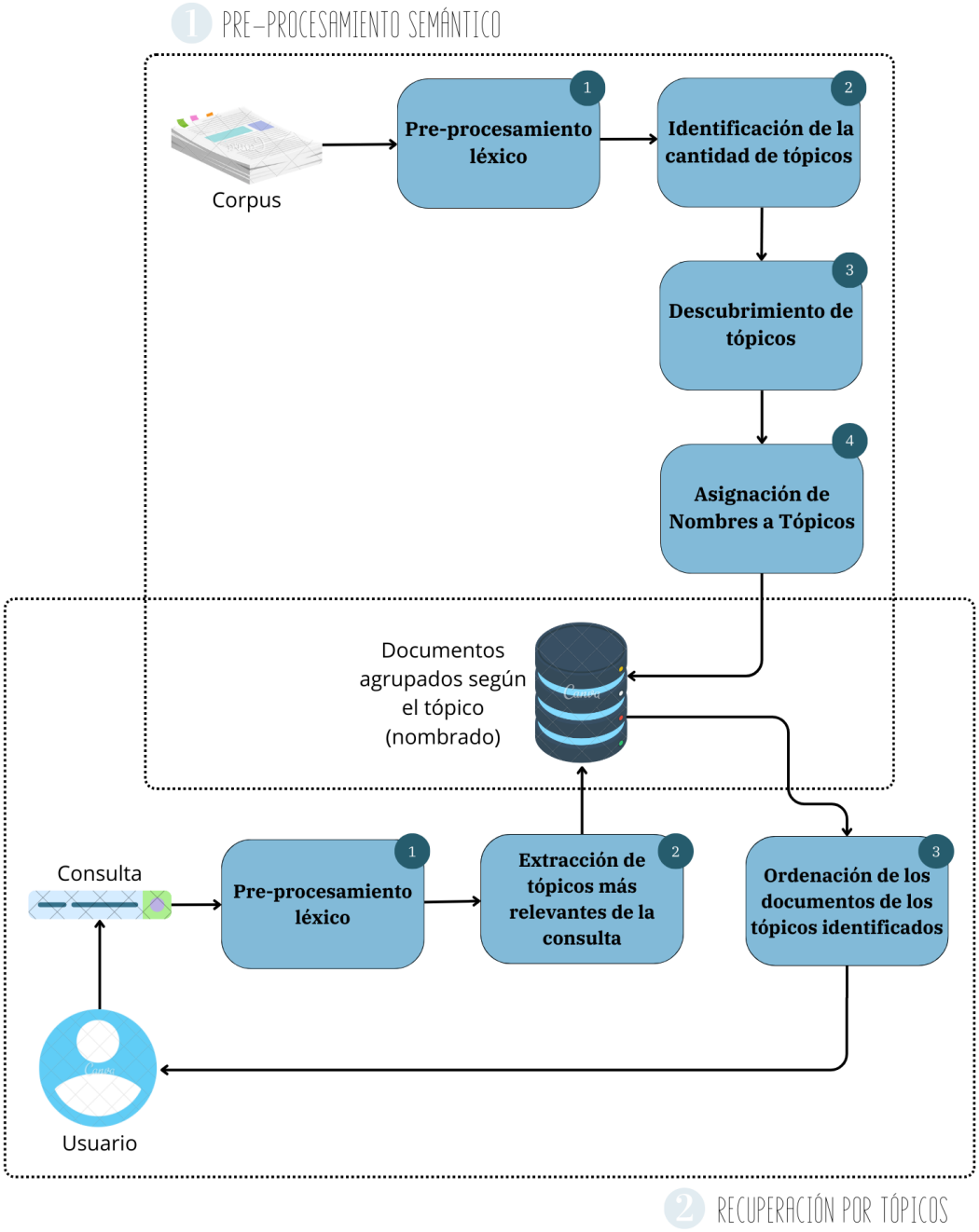

Figura 2.1: Arquitectura del sistema

2.1. Pre-procesamiento semántico

En la búsqueda por perfeccionar los modelos de tópicos en el procesamiento de texto, esta sección se enfoca en dos aspectos cruciales: la estimación automática del número de tópicos antes de la aplicación del modelo y la asignación automática de nombres a los tópicos identificados. Estos elementos desempeñan un papel fundamental en la mejora de la interpretación y utilidad de los resultados obtenidos mediante técnicas como Latent Dirichlet Allocation (LDA). La automatización de estos procesos no solo simplifica la implementación de modelos, sino que también enriquece la comprensión de patrones temáticos en grandes conjuntos de datos textuales, allanando el camino hacia un análisis más eficiente y accesible.

2.1.1. Pre-procesamiento léxico

En el ámbito del procesamiento de texto, el pre-procesamiento léxico desempeña un papel fundamental al garantizar la adecuada preparación del texto antes de su análisis. Este proceso es esencial en cualquier sistema de procesamiento de texto, ya que establece las bases para la comprensión y extracción de información significativa. En este contexto, se sigue un pre-procesamiento léxico básico (ver figura 2.2), un conjunto de pasos iniciales que buscan homogeneizar y organizar el texto de manera que facilite las tareas posteriores de análisis y procesamiento.

La primera etapa del pre-procesamiento léxico comienza con la tokenización, fragmentando el texto en unidades léxicas para facilitar la identificación y manipulación de palabras. Seguidamente, se lleva a cabo la eliminación de ruido, suprimiendo elementos redundantes como signos de puntuación o números, que podrían interferir con la interpretación precisa del contenido. A continuación, se realiza la exclusión de términos comunes que carecen de relevancia semántica significativa.

Para la reducción morfológica del vocabulario existen dos técnicas: *lemmatization* y *stemming*. La *lemmatization* simplifica la identificación de términos al reducir las palabras a sus formas base, mientras que el *stemming* halla formas truncadas al eliminar sufijos y prefijos. Se escoge la *lemmatization* debido al requisito de trabajar con palabras reales para modelos y ontologías en nuestro contexto específico, cosa no garantizada en el *stemming*. Posteriormente, se aplica un filtrado según la ocurrencia, eliminando términos poco frecuentes o excesivamente repetitivos.

Se construye el vocabulario del corpus y se representan vectorialmente los documentos, eligiendo en este estudio específicamente la representación de bolsa de palabras. Además, se elabora la matriz de co-ocurrencia de términos para capturar las correlaciones entre palabras.

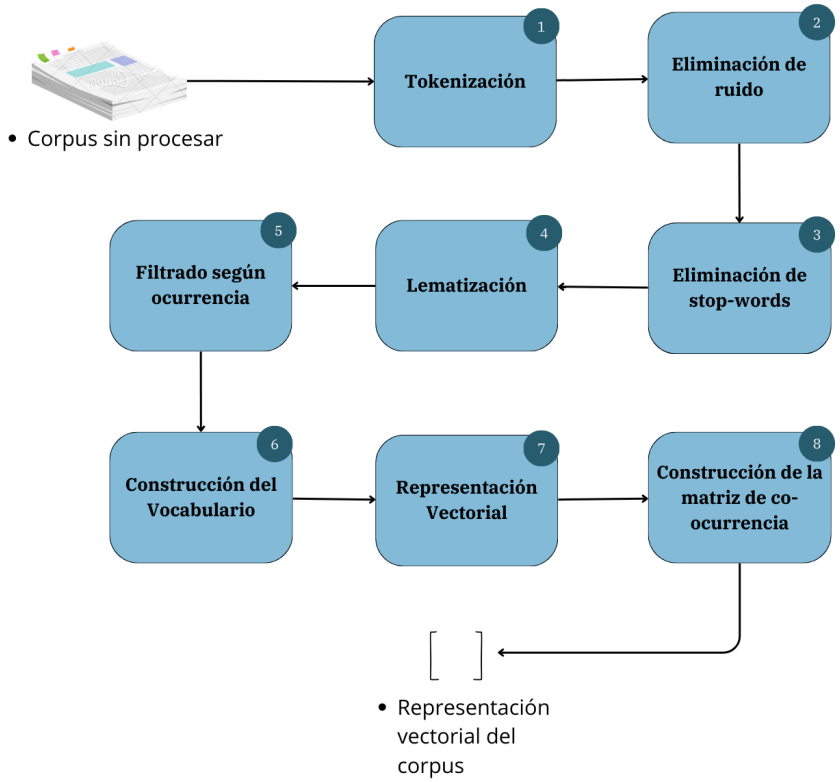

Figura 2.2: Pre-procesamiento léxico

2.1.2. Identificación de la Cantidad de Tópicos

Con el objetivo de identificar la cantidad de tópicos presentes en el corpus se propone un enfoque basado en la agrupación semántica de palabras, representadas por sus *embeddings* (ver figura 2.3). La hipótesis es que la cantidad de grupos formados, cada uno con una cantidad suficiente de palabras, proporcionará un indicador de la cantidad de tópicos presentes en el corpus. El algoritmo propuesto se basa en que las palabras que comparten significados similares o co-ocurren con frecuencia en documentos estarán asociadas en conjuntos semánticos, revelando así la presencia de tópicos específicos. Al realizar una agrupación flexible en el modelo, donde cada palabra puede pertenecer a varios grupos, se refleja la polisemia y la complejidad semántica del lenguaje.

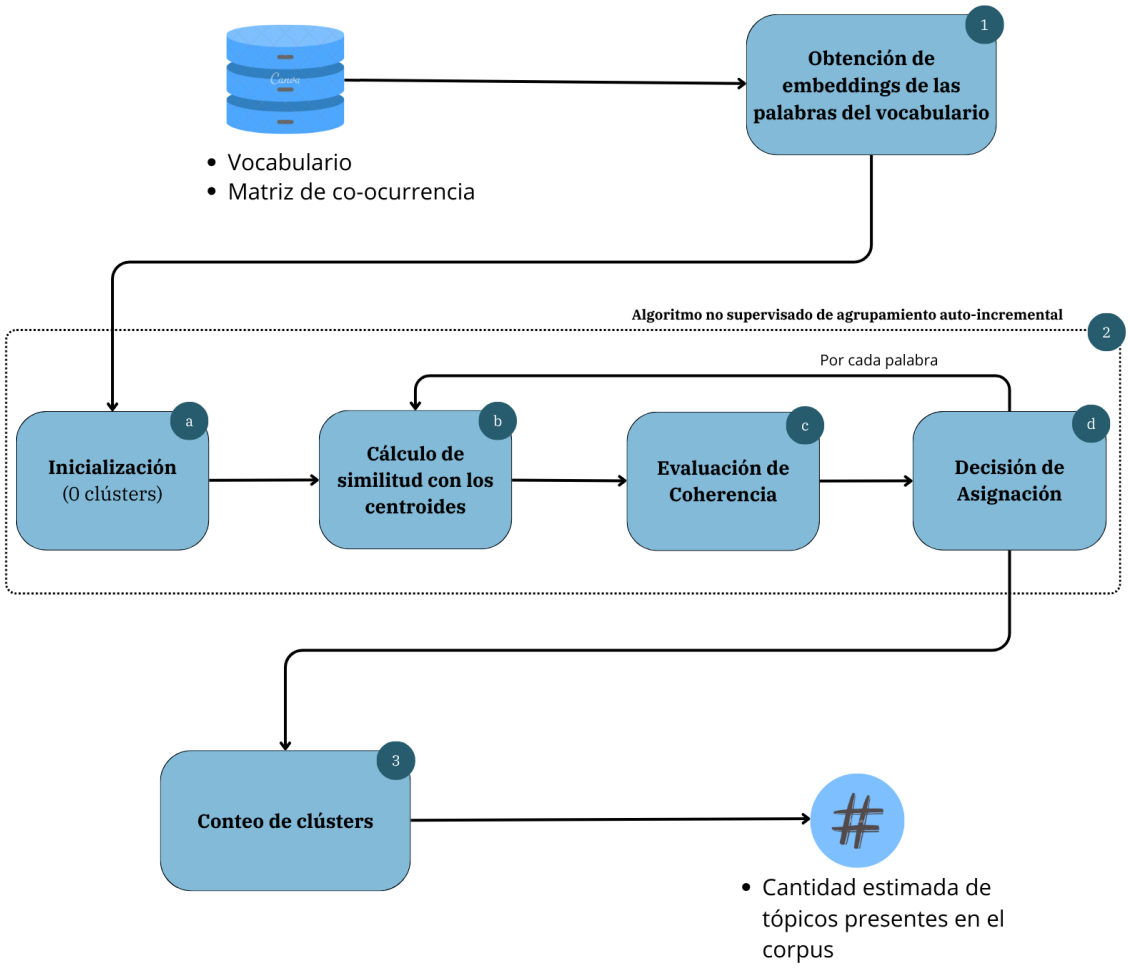

Figura 2.3: Identificación de la cantidad de tópicos

En la fase inicial del proceso, se obtienen los *word embeddings* para las palabras que conforman el vocabulario. Luego, se implementa un algoritmo de agrupamiento autoincremental sobre dichos *word embeddings*. A diferencia de los métodos convencionales que requieren una predefinición del número de grupos, este enfoque comienza con 0 grupos y estos se construyen dinámicamente a medida que es necesario.

La similitud semántica, basada en la proximidad en el espacio vectorial, desempeña un papel esencial en la creación de grupos al agrupar palabras con significados similares. Sin embargo, los *word embeddings*, debido a su entrenamiento centrado en co-ocurrencias locales, carecen de información contextual más amplia y no logran distinguir entre las diferentes acepciones de una palabra en distintos contextos. Esta limitación puede resultar en la agrupación errónea de palabras con significado

compartido pero usos contextuales diferentes. Para superar este desafío, el enfoque propuesto incorpora la matriz de co-ocurrencia del corpus, enriqueciendo la representación al capturar relaciones contextuales entre palabras. Esta integración mejora la precisión y significado en la formación de grupos.

El algoritmo evalúa la similitud entre el *embedding* a agregar y los centroides de los grupos, seleccionando aquellos cuya similitud sea igual o superior al umbral especificado. Si el vector no cumple con esta medida para ningún grupo existente, se crea uno nuevo. En caso contrario, se analiza la coherencia contextual con las palabras de cada grupo seleccionado, incorporándolo a aquellos en los que al menos la mitad de las palabras sean coherentes con el vector. Este proceso garantiza que las palabras con significados y contextos afines se agrupen de manera coherente.

En la etapa final, se cuentan los grupos con una cardinalidad superior a un umbral predefinido, que representa la cantidad mínima de palabras necesarias para que un grupo sea considerado como un tema. Esta contabilización proporciona una estimación de la cantidad de tópicos presentes en el corpus.

2.1.3. Descubrimiento de tópicos

La aplicación de modelos de tópicos desempeña un papel fundamental en el descubrimiento de patrones temáticos en grandes conjuntos de datos textuales. Estos modelos proveen una herramienta eficaz para organizar documentos relacionados y revelar las estructuras temáticas subyacentes en un corpus, facilitando así la extracción de información significativa.

Dentro de este contexto, destaca el algoritmo Latent Dirichlet Allocation (LDA), el asigna palabras a tópicos en documentos, utilizando un proceso probabilístico iterativo. Esto facilita la organización y análisis de grandes conjuntos de datos de texto al proporcionar distribuciones que describen la probabilidad de pertenencia de un documento a un tópico y la asociación de una palabra a un tópico específico. A pesar de la existencia de enfoques más complejos, la robustez, aplicabilidad general y estatus clásico de LDA, así como su amplia adopción en la literatura especializada, respaldan su confiabilidad. Esto lo posiciona como una elección sólida en el continuo desarrollo y mejora de modelos de tópicos para el descubrimiento temático.

2.1.4. Asignación de nombres a tópicos

En el ámbito del análisis de tópicos, conferir nombres de manera automática a los tópicos identificados no solo añade claridad interpretativa, sino que también facilita la comprensión y exploración de grandes conjuntos de documentos. Este proceso, esencial para dotar de significado a los patrones temáticos descubiertos, se llevará a cabo mediante la utilización de una ontología de dominio general. Al aprovechar la

riqueza semántica y la estructura jerárquica de esta ontología, se busca lograr una asignación de nombres precisa y contextualizada para cada tópico identificado. Este enfoque (ver figura 2.4) contribuirá a mejorar la interpretabilidad y utilidad de los resultados obtenidos en la fase de descubrimiento de tópicos mediante Latent Dirichlet Allocation (LDA).

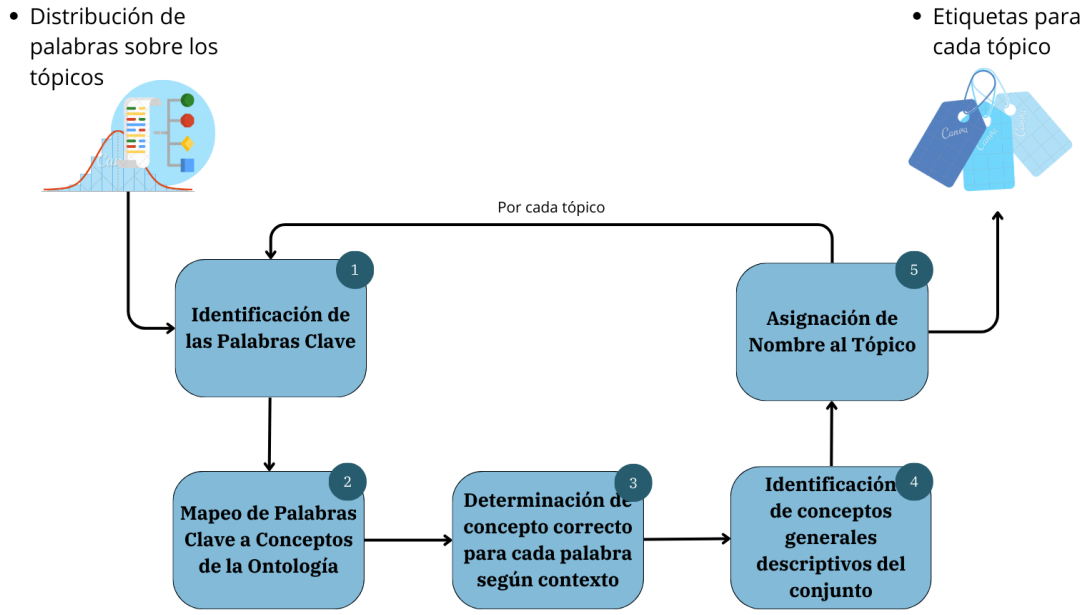

Figura 2.4: Asignación de nombres a tópicos

En la primera etapa, se seleccionan las palabras más probables para cada tópico a partir de las probabilidades proporcionadas por el modelo LDA. Este proceso incluye la aplicación de un umbral específico, lo que permite identificar de manera efectiva las palabras clave más representativas de cada temática. Estas palabras clave serán la base para la posterior asignación de nombres a los tópicos. Luego, cada palabra clave seleccionada se asigna a los conceptos pertinentes presentes en la ontología de dominio general. Es importante tener en cuenta que las ontologías están diseñadas para abordar la polisemia del lenguaje natural, lo que significa que, en la mayoría de los casos, una palabra puede estar asociada a más de un concepto en la ontología.

Luego, se hace necesario en la siguiente etapa, seleccionar el concepto adecuado para cada palabra de acuerdo con el contexto en el que aparece en el corpus. Con este propósito se aplican técnicas de Disambiguación del Sentido de las Palabras (WSD, por sus siglas en inglés). Este paso es fundamental para garantizar una asignación precisa y contextualizada de conceptos ontológicos que engloben a las palabras clave identificadas en los tópicos. Se exploran tres variantes para abordar este desafío.

La primera variante representa una mejora significativa sobre el algoritmo tradicional de Lesk utilizado en WSD. Mientras que el algoritmo de Lesk convencional se basa en coincidencias exactas entre las palabras del contexto y las presentes en las definiciones de los sentidos, la variante propuesta supera esta limitación. El enfoque propuesto aprovecha *word embeddings* para capturar la semántica y el significado contextual de las palabras, calculando la media de los *word embeddings* del contexto. Además, para cada definición en la ontología, se calcula la media aritmética de los *word embeddings* de las palabras asociadas a esa definición. La elección de la definición adecuada se realiza comparando las medias de *word embeddings*. Se selecciona la definición cuya media de *word embeddings* se aproxima más al contexto en términos de similitud del coseno. De esta forma se permite una mayor flexibilidad y capacidad para capturar relaciones semánticas más sutiles, mejorando así la precisión y el rendimiento del algoritmo en tareas de WSD.

Este problema de WSD se puede abordar también como un problema de optimización. Se plantea como la tarea de seleccionar, de una lista de n elementos (palabras), cada uno asociado a k características (definiciones), con k variable, n características exactamente, de modo que se maximice la similitud entre cada par de características seleccionadas. En este contexto, maximizar la similitud equivale a minimizar el camino entre dos definiciones en la ontología. Para resolver esta tarea, se utiliza el algoritmo SIMPLEX y se concibe un algoritmo genético, permitiendo así encontrar la mejor combinación de conceptos ontológicos asociados a las palabras clave en los tópicos identificados. Este enfoque optimizado pretende mejorar la precisión en la asignación de conceptos y contribuir a una representación semántica más refinada de los tópicos en el contexto ontológico.

Una vez se haya seleccionado la definición correcta en la ontología para cada palabra clave, se procede a identificar los conceptos más generales asociados a cada una. Se busca construir una lista de conceptos generales para cada palabra y, posteriormente, se aplica una función de peso híbrida para determinar cuáles de estos conceptos describen mejor al grupo de palabras. Esta función de peso considera diversos factores para evaluar la relevancia de cada concepto general. Entre los factores se encuentran: la profundidad en la jerarquía de la ontología, indicando cuán específico es el concepto, la información de contenido para obtener contexto, la similitud con las palabras presentes en el contexto y la generalidad del concepto. La función de peso híbrida permite asignar una puntuación a cada concepto general, tomando en cuenta la combinación de estos factores. Finalmente, se toman los conceptos generales que han obtenido la puntuación más alta, lo que asegura una selección de los términos más adecuados y representativos para denominar los tópicos identificados. Este enfoque garantiza una asignación de nombres que considere tanto la estructura jerárquica de la ontología como la riqueza semántica del contexto circundante.

2.2. Recuperación por tópicos

En esta sección se aborda el diseño de un motor de búsqueda basado en el modelo propuesto anteriormente. El propósito principal de este desarrollo es aprovechar los resultados obtenidos en la etapa anterior para mejorar la eficiencia y visualización de los resultados en la exploración.

Esta etapa se inicia con la recepción de consultas por parte de los usuarios. Utilizando el modelo de tópicos LDA previamente entrenado, se identifican los tópicos más relevantes para la consulta. Esta fase conlleva potencialmente la expansión de la consulta y su enriquecimiento semántico para poder obtener los tópicos correctos que esta abarque. Posteriormente, se lleva a cabo un filtrado de documentos asociados a tópicos no relevantes para la consulta, mejorando la eficiencia del sistema al centrar la búsqueda únicamente en la información contextualmente relacionada con los temas de interés del usuario. Se calcula la similitud entre los documentos de los tópicos relevantes y la consulta, y se presentan los resultados al usuario ordenados segun su relevancia, por tópicos.

Capítulo 3

Detalles de Implementación

En este capítulo, se lleva a cabo la materialización de la propuesta delineada en el capítulo anterior. Se detallan las herramientas utilizadas y la implementación concreta de la metodología concebida.

3.1. Herramientas y tecnologías utilizadas

3.1.1. Python

Python[1] es un lenguaje de programación de alto nivel, interpretado, con una sintaxis que enfatiza la legibilidad y la simplicidad. Es conocido por su versatilidad y eficiencia, lo que lo hace popular en diversas áreas como desarrollo web, análisis de datos, inteligencia artificial, aprendizaje automático y, por supuesto, recuperación de información.

Su amplia aceptación y uso en la comunidad académica, especialmente en áreas relacionadas con el procesamiento del lenguaje natural y el aprendizaje automático, proporcionan un soporte robusto, con una amplia disponibilidad de documentación, tutoriales y foros de discusión, así como una base sólida para el desarrollo y la colaboración en investigaciones.

La elección de Python se ve reforzada por su rica biblioteca de recursos, las cuales proporcionan funciones avanzadas para el procesamiento de datos, análisis estadístico, procesamiento del lenguaje natural y visualización de datos, elementos esenciales para la metodología propuesta en esta investigación. A continuación se detallan las principales bibliotecas utilizadas en esta investigación.

[1] https://www.python.org/

NLTK

NLTK (Natural Language Toolkit)[2] es una biblioteca integral para el procesamiento del lenguaje natural. Proporciona una amplia gama de herramientas como tokenizadores, etiquetadores gramaticales, analizadores sintácticos y reconocedores de entidades nombradas. Es especialmente valiosa por su facilidad de uso y la profundidad de sus recursos, como acceso a corpus y recursos léxicos, incluyendo WordNet.

Gensim

Gensim[3] se especializa en modelado semántico y recuperación de información, siendo particularmente fuerte en el modelado de tópicos y la indexación de documentos, con modelos como LDA y LSA. Además permite trabajar de forma sencilla con herramientas como Word2Vec y Doc2Vec. Ofrece también facilidades para acceder a corpus ampliamente utilizados, asi como procesar y manejar representaciones vectoriales de estos, convirtiéndose así un enfoque integral y eficiente en el análisis de grandes conjuntos de datos.

PuLP

PuLP[4] es una biblioteca para la programación lineal que permite a los usuarios formular y resolver problemas de optimización. Además de permitir el uso de solucionadores clásicos como el método SIMPLEX, PuLP también es compatible con una variedad de otros solucionadores, tanto de software libre como comerciales. Esto incluye solucionadores como CBC, GLPK, CPLEX y GUROBI, ofreciendo así una amplia flexibilidad en la elección del motor de optimización.

Sklearn

Sklearn[5] es una biblioteca de aprendizaje automático, la cual ofrece una variedad de herramientas simples y eficientes para análisis predictivo de datos y modelado estadístico. Incluye algoritmos de clasificación, regresión, agrupación y reducción de dimensionalidad. Además, es compatible con otras bibliotecas de Python como NumPy y SciPy, y se utiliza ampliamente tanto en la academia como en la industria debido a su facilidad de uso, eficiencia y documentación bien elaborada

[2] https://www.nltk.org/
[3] https://radimrehurek.com/gensim/index.html
[4] https://github.com/coin-or/pulp
[5] https://scikit-learn.org/stable/

NumPy

NumPy[6] NumPy ofrece soporte para grandes arrays y matrices multidimensionales, junto con una colección de funciones matemáticas para operar sobre estos arrays. Es la piedra angular para la mayoría de las bibliotecas de ciencia de datos en Python, proporcionando una eficiencia crítica en operaciones matemáticas y estadísticas, y es ampliamente utilizado en tareas que van desde simples cálculos hasta algoritmos complejos en diversos campos científicos y de ingeniería.

matplotlib

matplotlib[7] es una biblioteca de trazado que produce figuras de calidad de publicación en una variedad de formatos y entornos interactivos. Desde histogramas y gráficos de barras hasta gráficos en 3D, matplotlib maneja una amplia gama de visualizaciones de datos. Es ampliamente utilizado por científicos de datos para entender y presentar sus datos de manera clara y efectiva, siendo fundamental en la visualización y el análisis exploratorio de los mismos.

Wordcloud

Wordcloud[8] es una herramienta para la generación visual de nubes de palabras. Es una forma atractiva y a menudo reveladora de visualizar datos de texto, útil para resaltar las palabras más frecuentes o significativas en un conjunto de datos.

3.1.2. Word2Vec

Word2Vec [29] es un modelo para convertir palabras en *embeddings*, desarrollado por Tomas Mikolov y un equipo de investigadores de Google lanzado en 2013. Word2Vec utiliza dos arquitecturas principales para construir los *embeddings*, ambas técnicas basadas en redes neuronales poco profundas.

1. Continuous Bag of Words (CBOW): Se toman varias palabras circundantes (el contexto) y se utiliza esta información para predecir una palabra específica en medio de estas palabras de contexto. Por ejemplo, dada la secuencia de palabras "El gato come ...", CBOW intentaría predecir la palabra "pescado" basándose en las palabras "El", "gato" y "come". Esta arquitectura es particularmente eficaz para tratar palabras comunes, ya que promedia el contexto, lo que suaviza la distribución de información y ayuda a destacar las palabras más frecuentes.

[6] https://numpy.org/
[7] https://matplotlib.org/
[8] https://github.com/amueller/word_cloud

2. Skip-gram: Intenta predecir las palabras de contexto a partir de una palabra objetivo. Utilizando el mismo ejemplo anterior, dada la palabra "come", Skip-gram trataría de predecir las palabras "El", "gato", y "pescado". Esta técnica resulta ser más eficaz para aprender representaciones de alta calidad para palabras menos comunes, ya que se enfoca en predecir cada palabra de contexto desde la palabra objetivo, lo que permite captar patrones incluso en palabras menos frecuentes.

Los *embeddings* de Word2Vec permiten realizar una amplia gama de operaciones matemáticas y lingüísticas significativas. Entre estas, se encuentra la capacidad de calcular la similitud coseno entre dos palabras para determinar su cercanía semántica, y la habilidad para resolver analogías de palabras, como en el clásico ejemplo "rey es a reina como hombre es a mujer". Además, estos vectores pueden ser utilizados para agrupar palabras en categorías semánticas mediante técnicas de agrupamiento, así como para reducir su dimensionalidad y facilitar la visualización de relaciones lingüísticas. Otra operación común es la suma o promedio de vectores para representar el significado general de frases o documentos. También se pueden realizar operaciones aritméticas simples, como la famosa "Rey" - "Hombre" + "Mujer" que tiende a resultar en el vector de "Reina". Por último, los *embeddings* de Word2Vec son útiles en la detección de palabras atípicas dentro de un conjunto dado. Estas capacidades hacen de los *embeddings* de Word2Vec herramientas extremadamente poderosas en el campo del procesamiento del lenguaje natural, aplicables en sistemas de recomendación, búsqueda semántica, análisis de sentimientos y generación de texto.

Entre las limitaciones de este modelo se encuentra su incapacidad para capturar el significado de palabras fuera de su contexto inmediato, lo que resulta en una gestión a veces deficiente de palabras polisémicas. Además, como se basa en datos de entrenamiento, puede perpetuar y amplificar sesgos presentes en esos datos. Otra limitación importante es que Word2Vec no considera el orden de las palabras, lo cual puede ser crucial para comprender completamente el significado en ciertos contextos lingüísticos.

En esta investigación se trabaja específicamente con el modelo Word2Vec[9] de Google pre-entrenado con una porción del conjunto de datos de Google News. El modelo incluye un extenso vocabulario de 3 millones de palabras y frases, cada una representada por un vector de 300 dimensiones. Esta amplia gama de términos y la profundidad de sus representaciones vectoriales permiten capturar complejas relaciones semánticas y sintácticas. Además, el modelo destaca por incluir frases compuestas, ampliando su aplicabilidad más allá de las palabras individuales. Se escogió este específicamente puesto que es un modelo gratuito de dominio general, lo cual contribuye a la adaptabilidad del sistema a diferentes contextos. Su versatilidad y facilidad de

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

integración lo hacen ideal para una variedad de aplicaciones en NLP, desde el análisis de sentimientos hasta la generación de texto, ofreciendo así una herramienta robusta y eficaz para la comprensión y el análisis del lenguaje natural.

3.1.3. Wordnet

Wordnet[10] es una base de datos léxica para el idioma inglés que funciona como una red semántica. Fue desarrollada en el Cognitive Science Laboratory de la Universidad de Princeton bajo la dirección del profesor George A. Miller, uno de los pioneros en la psicología cognitiva. El desarrollo de WordNet comenzó en 1985 y su primera versión fue lanzada en línea en 1995. El objetivo principal de WordNet es proporcionar una estructura rica y utilizable para el procesamiento del lenguaje natural y la comprensión del lenguaje humano. Combina elementos de un diccionario tradicional con una red más avanzada de relaciones semánticas entre palabras y conceptos. Es de acceso público y gratuito, lo que la convierte en un recurso provechoso para investigadores, desarrolladores y estudiantes en todo el mundo. Además de su disponibilidad abierta, WordNet ofrece interfaces de programación de aplicaciones (API) que facilitan la integración de su vasta base de datos léxica en una variedad de aplicaciones y proyectos.

Cada palabra tiene un conjunto de conceptos correspondientes en Wordnet denominados *synsets*. Un *synset* representa el significado de una palabra en un contexto específico. De esta forma se captura la polisemia en la ontología. Los *synsets* están interconectados mediante múltiples tipos de relaciones semánticas, incluyendo la hiperonimia (identificación de términos más generales) e hiponimia (identificación de términos más específicos), así como la búsqueda de merónimos (partes de algo) y holónimos (el todo de una parte). Se pueden explorar antónimos para entender opuestos conceptuales. Además, es posible acceder a definiciones y ejemplos de uso para cada *synset*, proporcionando un contexto más claro del significado. También es posible investigar las relaciones de sinonimia entre palabras y examinar la pertenencia a diferentes categorías gramaticales (partes del discurso).

Entre dos *synsets*, se pueden realizar además otras operaciones interesantes para analizar y explorar sus relaciones semánticas y léxicas. Una operación común es calcular la distancia semántica entre los *synsets*. Se pueden explorar los diferentes caminos de conexión que unen los *synsets* en la red, revelando complejas relaciones semánticas. En los verbos, se puede explorar la troponimia, donde un verbo representa una manera más específica de realizar la acción descrita por otro verbo. Las relaciones morfosemánticas también son objeto de análisis, estudiando cómo los *synsets* se relacionan en términos de derivaciones morfológicas, como entre un verbo y su sustantivo relacionado o un adjetivo y su adverbio correspondiente.

[10] https://wordnet.princeton.edu/

En WordNet, los conceptos, especialmente dentro de la categoría de sustantivos, están organizados jerárquicamente. Esta jerarquía sigue el principio de que los conceptos más generales ocupan niveles superiores, mientras que los más específicos se encuentran en niveles más profundos. Por ejemplo, `animal` estaría en un nivel más alto que `mamífero`, y `mamífero` estaría a su vez en un nivel más alto que `perro`. Sin embargo, es importante destacar que WordNet, en su conjunto, no se conforma como un único árbol jerárquico, sino como una red compleja. En lo que respecta a los verbos, estos no forman una jerarquía lineal simple como en el caso de los sustantivos, sino que están organizados en estructuras que reflejan diferentes relaciones semánticas, como la troponimia, que muestra especificidad en la manera de realizar una acción. Esto refleja la complejidad de las acciones y estados en el lenguaje, donde los verbos pueden tener múltiples significados y usos. En consecuencia, los verbos en WordNet forman estructuras interconectadas que no siempre siguen una jerarquía lineal clara y, a menudo, no tienen una conexión directa con un nodo central o principal en la red

3.2. Implementación de la Metodología Propuesta

Esta sección ofrece una traducción de los conceptos y estrategias de diseño en código. Se centra en detallar la implementación de cada componente del pre-procesamiento semántico, incluyendo ejemplos de código y explicaciones sobre cómo se han utilizado las bibliotecas específicas en este proceso.

3.2.1. Pre-procesamiento léxico

Utilizando NLTK, se llevó a cabo la tokenización para descomponer el texto en unidades básicas, se aplicó el lematizador de WordNet para reducir las palabras a su forma base, y se empleó su lista predefinida de *stopwords* para eliminar términos comunes. Adicionalmente, se integró el corpus 'words'. de palabras reales del idioma inglés, para filtrar palabras no estándares.

Por otro lado, se utilizó Gensim para el proceso de filtrado de *tokens* según su ocurrencia, excluyendo aquellos presentes en menos de 5 documentos o en más de la mitad del total; así como para obtener la representación vectorial y el diccionario de los documentos.

Un aspecto importante del pre-procesamiento fue la construcción de la matriz de co-ocurrencia, que identifica y cuantifica la frecuencia con la que diferentes palabras aparecen juntas en el corpus, de manera global. Se utiliza CountVectorizer de Sklearn para transformar el corpus en una matriz de términos por documento, considerando sólo unigramas. Posteriormente, se calcula la matriz de co-ocurrencia multiplicando la matriz de términos-documentos por su transpuesta, lo que resulta en una matriz donde cada elemento indica cuántas veces dos palabras aparecen juntas en los documentos.

Después, establece en cero los elementos de la diagonal para excluir la co-ocurrencia de cada palabra consigo misma y normaliza la matriz resultante usando la normalización L2 por filas, lo que facilita comparar la co-ocurrencia en términos relativos en lugar de absolutos.

3.2.2. Identificación del número de tópicos

Primeramente, se realiza un filtrado del vocabulario de acuerdo a la parte del discurso (POS por sus siglas en inglés) que ocupe la palabra, excluyendo nombres propios, adverbios, verbos y adjetivos. Esta decisión se basa en que estos tipos de palabras tienden a generar agrupaciones que no aportan información relevante sobre un tema específico, entorpeciendo el funcionamiento del algoritmo.

Se utiliza entonces el modelo preentrenado Word2Vec para hallar los *word embeddings* de todos los sustantivos del vocabulario. Sobre estos *word embeddings*, se aplica un algoritmo de agrupamiento autoincremental (ver figura 3.1).

Algoritmo 1 AgrupamientoAutoIncremental

Require: *vocabulario*: Vocabulario del corpus, *matriz-coocurrencia*: Matriz de coocurrencia del corpus, *word_embeddings*: *Word embeddings* de las palabras del vocabulario, *min_sim*: Umbral de similitud mínima intra-clúster, *min_coh*: Umbral de coherencia mínima intra-clúster.
Ensure: Grupos de palabras semánticamente similares del vocabulario.

```
1:  function AUTO_INCREMENTAL CLUSTERING(vocabulario, matriz_coocurrencia, word_embeddings,
    min_sim, min_coh)
2:      grupos ← {}        ▷ Lista de grupos de la forma (centroide, word_embeddings, palabras).
3:      for cada word embedding w_e de las palabras del vocabulario do
4:          if no existen grupos then
5:              crear primer grupo con w_e
6:          else
7:              grupos_similares ← {C | C en grupos and sim(w_e, C) ≥ min_sim}
8:              grupos_coherentes ← {C | C en grupos_similares and coh(w, {w_C}) ≥ min_coh}
9:              if no existen grupos coherentes para w then
10:                 crear un nuevo grupo para w
11:             else
12:                 añadir w_e a los grupos coherentes
13:                 actualizar centroides de los grupos modificados
14:             end if
15:         end if
16:     end for
          return C
17: end function
```

Figura 3.1: Pseudo-código del Algoritmo de Agrupamiento Auto-Incremental.

Cada grupo tiene su centroide (media aritmética de los *word embeddings*), una lista con sus *word embeddings* y otra con las palabras correspondientes. La primera medida calculada para agrupar un nuevo *word embedding* es la similitud, utilizando la distancia coseno. En los grupos candidatos se halla también la coherencia. Por último,

se agrega el *word embedding* a los grupos resultantes y se recalculan los centroides o, en caso de que no haya ningún grupo coherente para el *word embedding*, se crea uno nuevo para él.

Luego, para definir el número de tópicos, se cuentan los grupos de palabras cuya cardinalidad sea superior a un umbral predefinido, que establezca la cantidad mínima de palabras necesarias para que un grupo sea considerado como tópico.

3.2.3. Descubrimiento de tópicos

En esta fase se utiliza el modelo de Asignación Latente de Dirichlet (LDA) de Gensim para el descubrimiento de tópicos. Esta implementación de LDA está optimizada para eficiencia, escalabilidad y manejo de grandes colecciones de texto, lo que la hace ideal para análisis de tópicos en grandes volúmenes de datos.

3.2.4. Asignación de nombres a tópicos

En esta última fase, se utiliza el módulo WordNet de NLTK para asignar nombres a los tópicos identificados. Para determinar las palabras más representativas de cada tópico, se ordenan según su relevancia, nuevamente tomando sólo los sustantivos. Se emplea un método que compara las diferencias en las probabilidades de palabras adyacentes, identificando el punto donde ocurre el primer descenso significativo (ver figura 3.2). Este enfoque se prefiere sobre la selección de las k palabras más probables, ya que reduce el riesgo de omitir términos relevantes o incluir otros que generen ruido.

Algoritmo 2 PalabrasRepresentativasTópico

Require: *tópico*: Número del tópico, k: Número inicial de palabras principales a recuperar
Ensure: Lista de las palabras más relevantes para el tópico especificado

function OBTENER_PALABRAS_PRINCIPALES(*topico*, k)
    palabras_del_topico ← Obtener las k palabras más probables para el tópico, que sean sustantivos, y sus probabilidades
    Ordenar *palabras_del_topico* en orden descendente por probabilidad
    cambios_de_probabilidad ← Calcular cambios en la probabilidad entre palabras consecutivas
    cambio_promedio ← Calcular el promedio de *cambios_de_probabilidad*
    desviacion_estandar ← Calcular la desviación estándar de *cambios_de_probabilidad*
    indice_caida_significativa ← Encontrar el primer índice donde el cambio es mayor que *cambio_promedio* + *desviacion_estandar*
    palabras_filtradas ← Filtrar las palabras hasta el punto de caída significativa
    return *palabras_filtradas*
end function

Figura 3.2: Pseudo-código del Algoritmo para hallar las palabras más relevantes del tópico.

La identificación del *synset* correcto para cada palabra se realiza mediante algoritmos de WSD, incluyendo una versión extendida de Lesk (ver figura 3.3), un algoritmo genético (ver figura 3.4) y el SIMPLEX de la biblioteca PuLP. Se descartan los conjuntos donde algún *synset* corresponda a un verbo, dada la limitación de estos en WordNet para analizar las relaciones de hiperónimos e hipónimos.

Luego, se examinan los hiperónimos de los *synsets* seleccionados, utilizando la función closure para identificar términos más generales con una profundidad mínima establecida en 4 para evitar términos demasiado generales. Posteriormente, se implementó una función de peso compuesta que integra varios factores clave: la frecuencia de aparición del hiperónimo en el conjunto analizado, su información de contenido (derivada del corpus de Wordnet), la relevancia semántica (calculada usando nuevamente el modelo Word2Vec preentrenado) y su especificidad. Este algoritmo selecciona los hiperónimos con los valores cercanos al máximo en esta la función de peso. El objetivo es priorizar aquellos términos que no sólo son frecuentes y relevantes en el contexto semántico, sino también informativos y con un nivel adecuado de especificidad.

Algoritmo 3 LeskExtendido

Require: *palabra*: La palabra a desambiguar, *contexto*: Contexto de la palabra, *synsets*: Lista de *synsets* de la palabra objetivo. *modelo*: Modelo preentrenado de *word embeddings*.
Ensure: *Synset* que mejor se ajusta al contexto.

```
function EXTENDED_LESK(palabra, contexto, synsets, modelo)
    context_embedding ← media aritmética de los word embeddings de las palabras del contexto.
    for cada synset de la palabra objetivo do
        if el synset no es un verbo then
            definition_embedding ← media aritmética de las palabras que conforman la definición del synset.
            similitud ← sim(context_embedding, definition_embedding)
            if similitud > maxima_similitud then
                maxima_similitud ← similitud
                mejor_synset ← synset
            end if
        end if
    end for
    return mejor_synset
end function
```

Figura 3.3: Pseudo-código del Algoritmo de Lesk Extendido.

Algoritmo 4 AlgoritmoGenéticoWSD

Require: *synsets*: Conjunto de synsets, *generaciones*: Número de generaciones, *tamaño_pob*: Tamaño de la población.
Ensure: Conjunto de *synsets* elegidos y su valor de fitness.

function GENETIC_ALGORITHM(*synsets*, *generaciones*, *tamaño_pob*)
    población ← INIT_POPULATION(*synsets*, tamaño_pob)
    mejor_solución ← ([], $-\infty$) ▷ Inicializar la mejor solución encontrada
    for cada generación **do**
        for cada individuo de la población **do**
            fitness ← EVALUATE(individuo)
            if fitness > *mejor_solución*[*fitness*] **then**
                mejor_solución ← (individuo, fitness)
            end if
        end for
        mejores_individuos ← SELECT_PARENTS(*población*, *valores_fitness*)
        población ← NEW_POPULATION(*mejores_individuos*)
    end for
    return *mejor_solucion*
end function
function EVALUATE(*synsets*)
    Calcular la similitud media entre los *synsets* seleccionados utilizando la medida de similitud según el camino (path) que ofrece wordnet
    Las combinaciones que contengan algún synset que actúe como verbo serán penalizadas.
    return Valor de similitud media
end function
function INIT_POPULATION(*context_synsets*, tamaño_pob)
    Inicializar *población* como lista vacía
    for cada iteración hasta *tamaño_pob* **do**
        Crear *individuo* como una lista de listas binarias, donde cada lista representa los synsets de una palabra. Asigna un '1' en una posición aleatoria de cada lista, correspondiendo al synset activo de la palabra que representa.
        Añadir *individuo* a *población*
    end for
    return *población*
end function
function SELECT_PARENTS(*población*, *valores_fitness*, *selection_proportion*)
    Seleccionar los mejores individuos como padres usando selección por ruleta
    return Lista de los mejores padres
end function
function NEW_POPULATION(*mejores_individuos*, *razón_cruce*, *razón_mutación*)
    Generar una nueva población usando operaciones de cruce y mutación
    return Nueva población
end function
function CROSSOVER(*padre_1*, *padre_2*)
    Seleccionar un punto de cruce aleatorio
    Crear dos hijos intercambiando los conjuntos de *synsets* en el punto de cruce
    return Lista con ambos hijos
end function
function MUTATE(*individuo*)
    Seleccionar un número aleatorio de conjuntos de *synsets* para mutar
    for cada conjunto seleccionado **do**
        Cambiar el *synset* activo por otro aleatorio
    end for
    return *individuo* mutado
end function

Figura 3.4: Pseudo-código del Algoritmo Genético.

Capítulo 4

Experimentación

Presentamos la estructura y diseño de los experimentos realizados para evaluar la eficacia de la metodología. Detallamos la selección de conjuntos de datos de prueba, establecemos métricas de evaluación pertinentes y explicamos las decisiones detrás de la configuración experimental. Este apartado proporciona un marco claro para la interpretación de los resultados y la validación de la propuesta.

4.1. Corpus utilizados

4.1.1. Brown

El Brown Corpus, desarrollado en la Universidad de Brown en la década de 1960, es uno de los primeros corpora electrónicos del idioma inglés y ha sido muy influyente en el campo del procesamiento del lenguaje natural. Está compuesto por 500 textos, cada uno de aproximadamente 2,000 palabras. Está dividido en 15 categorías, aunque estas pueden ser englobadas en 9 temas: prensa, religión, comercio y oficios, artes y literatura, ciencia, aprendizaje, ficción, humor, aprendizaje y miscelánea. Los textos provienen de una amplia gama de fuentes y estilos, incluyendo textos periodísticos, de ficción, y no ficción.

4.1.2. 20-Newsgroups

El corpus 20-Newsgroups[1] es un conjunto de datos, que consiste en aproximadamente 20.000 documentos, cada uno con alrededor de 1000 palabras, distribuidos en 20 grupos de noticias, cada uno correspondiente a un tema específico. Los documentos son en su mayoría mensajes o artículos de grupos de noticias, recopilados alrededor de 1995. Los temas cubren una variedad de áreas y son bastante diversos, apreciándose

[1] http://qwone.com/ jason/20Newsgroups/

siete temas principales: computación, vehículos, deportes, política, religión, ciencia y ventas; dentro de los cuales se encuentran los mencionados 20.

4.1.3. Reuters-21578

El corpus Reuters-21578[2] es otro conjunto de datos, compuesto por 21,578 artículos de noticias, recopilados de la agencia de noticias Reuters en 1987. Cada artículo tiene una longitud variable, generalmente oscilando entre unas pocas docenas a varios cientos de palabras. Los artículos en el corpus Reuters-21578 están clasificados en varias categorías, que representan una mezcla de temas relacionados con finanzas, economía, mercado de valores y eventos internacionales.

4.2. Experimento 1: Identificación del número de tópicos

Este experimento se centra en la identificación del número de tópicos presentes en tres corpus de texto distintos: Brown, 20-Newsgroups y Reuters-21578. El objetivo principal es evaluar la capacidad de nuestro algoritmo para hacer tal estimación y examinar la coherencia de los tópicos identificados, tanto internamente, como su correspondencia a los tópicos reales del corpus. Inicialmente, se realizan pruebas individuales para cada corpus con el fin de determinar los mejores valores de los hiperparámetros. A continuación, se calcula un valor medio de estos hiperparámetros y se utiliza este valor promedio para realizar una evaluación final en cada uno de los corpus.

4.2.1. Pruebas Individuales y Selección de Hiperparámetros

- **20-Newsgroups:** Detalles de las pruebas realizadas y el valor óptimo de hiperparámetros encontrado.
- **Reuters-21578:** Descripción similar para Reuters-21578.
- **Brown** Lo mismo para el dump de Wikipedia 2023.
- **Resumen de Hiperparámetros óptimos:** Discusión sobre los valores óptimos encontrados para cada corpus.

[2]https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

4.2.2. Evaluación con Valores Medios de Hiperparámetros

- **Valor Medio de Hiperparámetros:** Cálculo y justificación del valor medio de los hiperparámetros óptimos.
- **Evaluación en 20-Newsgroups:** Resultados y análisis utilizando el valor medio en 20-Newsgroups.
- **Evaluación en Reuters-21578:** Resultados y análisis similares para Reuters-21578.
- **Evaluación en Brown** Igual para el dump de Wikipedia 2023.

4.3. Experimento 2: Asignación de nombres a tópicos

Este experimento evalúa la coherencia en la asignación automática de nombres a tópicos, comparando los resultados obtenidos con los nombres reales de los tópicos. Además, se compararán los tres algoritmos de WSD concebidos en cada corpus para analizar su impacto en este proceso.

4.3.1. Comparación de Algoritmos de WSD

- **20-Newsgroups:** Detalles de las pruebas realizadas y el mejor algoritmo.
- **Reuters-21578:** Descripción similar para Reuters-21578.
- **Brown** Lo mismo para el dump de Wikipedia 2023.
- **Conclusiones del mejor** Presentación y análisis de los resultados obtenidos con cada algoritmo en cada corpus.

4.3.2. Comparación con Nombres Reales de Grupos

- **Comparación con Nombres de Grupos Reales:** Comparación entre los nombres asignados por el algoritmo y los nombres reales de los grupos en los corpus.
- **Comparación con investigación anterior**

4.4. Análisis de Resultados

Exponemos los resultados obtenidos a través de los experimentos y realizamos un análisis detallado de los mismos. Se comparan los rendimientos obtenidos con otros enfoques existentes (si aplicable) y se evalúa la robustez de la metodología en diferentes condiciones. Además, destacamos cualquier hallazgo inesperado o patrones significativos identificados durante la experimentación.

Conclusiones

En esta tesis, se han explorado desafíos importantes relacionados con los modelos de tópicos, poniendo especial énfasis en mejorar el ajuste automático de hiperparámetros en entornos dinámicos y en la precisión semántica de los tópicos. Este trabajo ha profundizado en el marco teórico y conceptual de los Sistemas de Recuperación de Información (SRI), con un enfoque particular en los modelos de tópicos y ontologías, estableciendo así una base sólida para las mejoras propuestas. Además, un análisis meticuloso del estado actual de los SRI en contextos dinámicos ha revelado tendencias clave, que han sido fundamentales en el desarrollo de la solución propuesta.

Basándose en este conocimiento, se han concebido y diseñado e estrategias orientadas a incrementar la adaptabilidad de los modelos de tópicos y a enriquecer la interpretación de sus resultados en una variedad de contextos. Estas estrategias fueron implementadas, proporcionando un marco práctico para evaluar su efectividad en los retos identificados.

El sistema fue evaluado de acuerdo a ..., obteniendo ... resultados

Recomendaciones

A partir de los desafíos surgidos en esta investigación y los resultados alcanzados, se identificaron áreas clave para futuras investigaciones que buscan mejorar y expandir el sistema desarrollado. A continuación se presentan recomendaciones derivadas de dichas experiencias y hallazgos, estableciendo direcciones para avanzar en este campo.

1. **Desarrollo de la Segunda Etapa del Sistema:** Como trabajo futuro, se sugiere la implementación completa de la segunda etapa del sistema propuesto. Esta etapa implica el desarrollo de un motor de búsqueda eficiente que utilice la clasificación y el análisis de tópicos realizados en la primera etapa para mejorar la recuperación y presentación de información relevante.

2. **Implementación de *contextual embeddings* para la Agrupación de Tópicos:** Se recomienda explorar el uso de *contextual embeddings* en lugar de *word embeddings* para la agrupación de tópicos. Esto implicaría mantener una lista de embeddings contextuales para cada palabra del vocabulario, basada en su aparición en diferentes documentos. Este enfoque podría implicar además un filtrado de dichos *embeddings*, seleccionando un representante para cada grupo semántico. Al incorporar contexto en estos *embeddings*, se espera mejorar la identificación del número de tópicos con grupos más coherentes.

3. **Ajuste para Uso con Ontologías de Dominio Específico:** Se recomienda adaptar y evaluar el sistema utilizando ontologías específicas de diferentes dominios. Esto podría mejorar significativamente la precisión y relevancia del análisis de tópicos en campos especializados.

4. **Experimentación con otros Modelos de Tópicos:** Se sugiere probar con otros modelos de tópicos que incorporen semántica, como el Correlated Topic Model. Estos modelos podrían ofrecer una comprensión más profunda y matizada de la estructura temática de los conjuntos de datos.

5. **Manejo de Palabras Ausentes en las herramientas utilizadas** Se recomienda desarrollar un enfoque para buscar palabras relacionadas con aquellas

que no aparecen en los modelos, con el objetivo de evitar su descarte y mejorar la cobertura del análisis de tópicos.

6. **Optimización de Hiperparámetros del Modelo:** Se propone investigar y desarrollar algoritmos específicos para la optimización de hiperparámetros, tal que se llegue a una buena opción general, o se intente algún enfoque para hallarlo dinámicamente.

Referencias

[1] C. Manning, P. Raghavan y H. Schuetze, «Introduction to information retrieval,» 2009 (vid. págs. 1, 5).

[2] A. Hodges, *Alan Turing: The Enigma The Centenary Edition*. Princeton: Princeton University Press, 31 de dic. de 2012, ISBN: 978-1-4008-4497-5. DOI: 10.1515/9781400844975. dirección: https://www.degruyter.com/document/doi/10.1515/9781400844975/html (visitado 06-12-2023) (vid. pág. 1).

[3] M. Sanderson y W. B. Croft, «The history of information retrieval research,» *Proceedings of the IEEE*, vol. 100, págs. 1444-1451, Special Centennial Issue mayo de 2012, ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2012.2189916. dirección: http://ieeexplore.ieee.org/document/6182576/ (visitado 06-12-2023) (vid. pág. 1).

[4] D. M. Sendall, «The world-wide web past present and future, and its application to medicine.,» (vid. pág. 1).

[5] R. Freeman, A. Wicks, P. Werhane y J. Mead, «Yahoo! And Customer Privacy (a),» *Darden Business Publishing Cases*, vol. 1, 21 de oct. de 2008. DOI: 10.1108/case.darden.2016.000356 (vid. pág. 1).

[6] C. O. Y. Lin y R. Yazdanifard, «How google's new algorithm, hummingbird, promotes content and inbound marketing,» *American Journal of Industrial and Business Management*, vol. 04, n.º 1, págs. 51-57, 2014, ISSN: 2164-5167, 2164-5175. DOI: 10.4236/ajibm.2014.41009. dirección: http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ajibm.2014.41009 (visitado 06-12-2023) (vid. pág. 2).

[7] C. Quintana-Wong, L. García, J. Guillot Jiménez y Z. Ambrós, *Recommendations for Healthcare using NoSQL*. 15 de jul. de 2019 (vid. págs. 2, 3).

[8] C. Quintana-Wong y L. García, *Integrating Neighborhood and Latent Factors Models to Obtain Accurate Recommendations*. 3 de sep. de 2019 (vid. pág. 3).

[9] C. Quintana-Wong, L. García y L. Garrido, *Latent Factors and Topic Models for Semantically Richer User Profiles*. 27 de ago. de 2019 (vid. pág. 3).

[10] C. L. González, «Modelos de generación de tópicos aplicando word embedding,» (vid. pág. 3).

[11] M. Prado-Romero, A. Coto-Santiesteban, A. Celi y G. Stilo, «A time-sensitive model to predict topic popularity in news providers1,» *Intelligent Data Analysis*, vol. 24, págs. 123-140, 4 de dic. de 2020. DOI: 10.3233/IDA-200012 (vid. pág. 3).

[12] W. B. Croft, D. Metzler y T. Strohman, «Search engines information retrieval in practice,» (vid. pág. 5).

[13] M. Lew, N. Sebe, C. Djeraba y R. Jain, «Content-based multimedia information retrieval: State of the art and challenges,» *TOMCCAP*, vol. 2, págs. 1-19, 1 de feb. de 2006. DOI: 10.1145/1126004.1126005 (vid. pág. 5).

[14] C. Jones y R. Purves, «Geographical information retrieval,» *International Journal of Geographical Information Science*, vol. 22, págs. 219-228, 1 de mar. de 2008. DOI: 10.1080/13658810701626343 (vid. pág. 6).

[15] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer y R. Harshman, «Indexing by latent semantic analysis,» *Journal of the American Society for Information Science*, vol. 41, n.º 6, págs. 391-407, sep. de 1990, ISSN: 0002-8231, 1097-4571. DOI: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9. dirección: https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9 (visitado 02-09-2023) (vid. pág. 6).

[16] T. Hoffman, «Probabilistic Latent Semantic Analysis,» (vid. pág. 6).

[17] D. Blei, A. Ng y M. Jordan, *Latent Dirichlet Allocation*. 1 de ene. de 2001, vol. 3, 601 págs., Journal Abbreviation: The Journal of Machine Learning Research Pages: 608 Publication Title: The Journal of Machine Learning Research (vid. pág. 6).

[18] D. M. Blei y J. D. Lafferty, «A correlated topic model of science,» *The Annals of Applied Statistics*, vol. 1, n.º 1, 1 de jun. de 2007, ISSN: 1932-6157. DOI: 10.1214/07-AOAS114. dirección: https://projecteuclid.org/journals/annals-of-applied-statistics/volume-1/issue-1/A-correlated-topic-model-of-Science/10.1214/07-AOAS114.full (visitado 02-09-2023) (vid. pág. 7).

[19] D. M. Blei y J. D. Lafferty, «Dynamic topic models,» en *Proceedings of the 23rd international conference on Machine learning - ICML '06*, Pittsburgh, Pennsylvania: ACM Press, 2006, págs. 113-120, ISBN: 978-1-59593-383-6. DOI: 10.1145/1143844.1143859. dirección: http://portal.acm.org/citation.cfm?doid=1143844.1143859 (visitado 02-01-2024) (vid. pág. 7).

[20] D. Blei, T. Griffiths y M. Jordan, «The nested Chinese restaurant process and Bayesian inference of topic hierarchies,» *Journal of the ACM*, vol. 57, 3 de oct. de 2007. DOI: 10.1145/1667053.1667056 (vid. pág. 7).

[21] N. Guarino, *Formal Ontologies and Information Systems*. 6 de jun. de 1998 (vid. pág. 7).

[22] M. Uschold y M. Grüninger, «Ontologies: Principles, methods and applications,» *The Knowledge Engineering Review*, vol. 11, 1 de ene. de 1996 (vid. pág. 7).

[23] T. Berners-Lee, J. Hendler y O. Lassila, «The Semantic Web: A New Form of Web Content That is Meaningful to Computers Will Unleash a Revolution of New Possibilities,» *ScientificAmerican.com*, 1 de mayo de 2001 (vid. pág. 8).

[24] A. Hotho, A. Maedche y S. Staab, «Ontology-based Text Document Clustering.,» *KI*, vol. 16, págs. 48-54, 1 de ene. de 2002 (vid. pág. 8).

[25] M. Batet, A. Valls y K. Gibert, «Improving classical clustering with ontologies,» *IASC*, 5 de ene. de 2009 (vid. pág. 8).

[26] O. Corcho, «Ontology Based Document Annotation: Trends and Open Research Problems,» *International Journal of Metadata, Semantics and Ontologies*, vol. 1, 1 de ene. de 2006. DOI: 10.1504/IJMSO.2006.008769 (vid. pág. 8).

[27] J. J. Lastra-Díaz, J. Goikoetxea, M. A. Hadj Taieb, A. García-Serrano, M. Ben Aouicha y E. Agirre, «A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art,» *Engineering Applications of Artificial Intelligence*, vol. 85, págs. 645-665, oct. de 2019, ISSN: 09521976. DOI: 10.1016/j.engappai.2019.07.010. dirección: https://linkinghub.elsevier.com/retrieve/pii/S0952197619301745 (visitado 05-10-2023) (vid. pág. 8).

[28] F. Almeida y G. Xexéo, *Word embeddings: A survey*, 1 de mayo de 2023. arXiv: 1901.09069[cs,stat]. dirección: http://arxiv.org/abs/1901.09069 (visitado 29-12-2023) (vid. pág. 8).

[29] T. Mikolov, K. Chen, G. Corrado y J. Dean, *Efficient estimation of word representations in vector space*, 6 de sep. de 2013. arXiv: 1301.3781[cs]. dirección: http://arxiv.org/abs/1301.3781 (visitado 02-01-2024) (vid. págs. 8, 24).

[30] J. Pennington, R. Socher y C. Manning, «Glove: Global vectors for word representation,» en *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, págs. 1532-1543. DOI: 10.3115/v1/D14-1162. dirección: http://aclweb.org/anthology/D14-1162 (visitado 02-01-2024) (vid. pág. 8).

[31] J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, «BERT: Pre-training of deep bidirectional transformers for language understanding,» (vid. pág. 8).

[32] A. Fang, C. Macdonald, I. Ounis y P. Habel, «Using word embedding to evaluate the coherence of topics from twitter data,» en *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa Italy: ACM, 7 de jul. de 2016, págs. 1057-1060, ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2914729. dirección: https://dl.acm.org/doi/10.1145/2911451.2914729 (visitado 05-10-2023) (vid. pág. 9).

[33] S. Kuzi, A. Shtok y O. Kurland, *Query Expansion Using Word Embeddings*. 24 de oct. de 2016, 1929 págs., Pages: 1932. DOI: 10.1145/2983323.2983876 (vid. pág. 9).

[34] M. J. Kusner, Y. Sun, N. I. Kolkin y K. Q. Weinberger, «From word embeddings to document distances,» (vid. pág. 9).

[35] X. Liu y W. B. Croft, «Cluster-based retrieval using language models,» en *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, Sheffield United Kingdom: ACM, 25 de jul. de 2004, págs. 186-193, ISBN: 978-1-58113-881-8. DOI: 10.1145/1008992.1009026. dirección: https://dl.acm.org/doi/10.1145/1008992.1009026 (visitado 02-09-2023) (vid. pág. 9).

[36] R. Arun, V. Suresh, C. E. Veni Madhavan y M. N. Narasimha Murthy, «On finding the natural number of topics with latent dirichlet allocation: Some observations,» en *Advances in Knowledge Discovery and Data Mining*, M. J. Zaki, J. X. Yu, B. Ravindran y V. Pudi, eds., red. de D. Hutchison et al., vol. 6118, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, págs. 391-402, ISBN: 978-3-642-13656-6 978-3-642-13657-3. DOI: 10.1007/978-3-642-13657-3_43. dirección: http://link.springer.com/10.1007/978-3-642-13657-3_43 (visitado 29-12-2023) (vid. pág. 9).

[37] J. Gan e Y. Qi, «Selection of the optimal number of topics for LDA topic model—taking patent policy analysis as an example,» *Entropy*, vol. 23, n.º 10, pág. 1301, 3 de oct. de 2021, ISSN: 1099-4300. DOI: 10.3390/e23101301. dirección: https://www.mdpi.com/1099-4300/23/10/1301 (visitado 27-12-2023) (vid. pág. 9).

[38] R. Vangara et al., «Finding the number of latent topics with semantic non-negative matrix factorization,» *IEEE Access*, vol. 9, págs. 117 217-117 231, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3106879. dirección: https://ieeexplore.ieee.org/document/9521777/ (visitado 29-12-2023) (vid. pág. 10).

[39] Y. Jiang, R.-J. Liou y S.-J. Lee, «A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification,» *IEEE Trans. Knowl. Data Eng.*, vol. 23, págs. 335-349, 1 de mar. de 2011. DOI: 10.1109/TKDE.2010.122 (vid. pág. 10).

[40] L. Thompson y D. Mimno, *Topic Modeling with Contextualized Word Representation Clusters*. 23 de oct. de 2020 (vid. pág. 10).

[41] N. Boufaden, «An ontology-based semantic tagger for IE system,» en *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03*, vol. 2, Sapporo, Japan: Association for Computational Linguistics, 2003, págs. 7-14, ISBN: 978-0-11-145678-1. DOI: 10.3115/1075178.1075179. dirección: http://portal.acm.org/citation.cfm?doid=1075178.1075179 (visitado 29-12-2023) (vid. pág. 10).

[42] A. SinghRathore y D. Roy, «Ontology based web page topic identification,» *International Journal of Computer Applications*, vol. 85, n.º 6, págs. 35-40, 16 de ene. de 2014, ISSN: 09758887. DOI: 10.5120/14849-3211. dirección: http://research.ijcaonline.org/volume85/number6/pxc3893211.pdf (visitado 29-12-2023) (vid. pág. 11).

[43] S. Saqlain, A. Nawaz, I. Khan, F. Shah y M. Ashraf, «Text Clusters Labeling using WordNet and Term Frequency- Inverse Document Frequency,» *Proceedings of the Pakistan Academy of Sciences*, vol. Vol. 53, págs. 281-291, 1 de sep. de 2016 (vid. pág. 11).

[44] M. Lesk, «Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone,» (vid. pág. 11).

[45] P. Edmonds y E. Agirre, *Word Sense Disambiguation—Algorithms and Applications*. 1 de ene. de 2008, ISBN: 978-1-4020-4808-1. DOI: 10.1007/978-1-4020-4809-8 (vid. pág. 12).

[46] M. Bevilacqua, T. Pasini, A. Raganato y R. Navigli, «Recent trends in word sense disambiguation: A survey,» en *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Montreal, Canada: International Joint Conferences on Artificial Intelligence Organization, ago. de 2021, págs. 4330-4338, ISBN: 978-0-9992411-9-6. DOI: 10.24963/ijcai.2021/593. dirección: https://www.ijcai.org/proceedings/2021/593 (visitado 29-12-2023) (vid. pág. 12).

[47] B. Scarlini, T. Pasini y R. Navigli, «SensEmBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation,» *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, n.º 5, págs. 8758-8765, 3 de abr. de 2020, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v34i05.6402. dirección: https://ojs.aaai.org/index.php/AAAI/article/view/6402 (visitado 02-01-2024) (vid. pág. 12).

[48] W. AL-Saiagh, S. Tiun, A. Alsaffar, S. Awang y A. S. Al-Khaleefa, «Word sense disambiguation using hybrid swarm intelligence approach,» *PLOS ONE*, vol. 13, e0208695, 20 de dic. de 2018. DOI: 10.1371/journal.pone.0208695 (vid. pág. 12).

[49] W. Alsaeedan y M. E. B. Menai, «A self-adaptive genetic algorithm for the word sense disambiguation problem,» en *Current Approaches in Applied Artificial Intelligence*, M. Ali, Y. S. Kwon, C.-H. Lee, J. Kim e Y. Kim, eds., vol. 9101, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, págs. 581-590, ISBN: 978-3-319-19065-5 978-3-319-19066-2. DOI: 10.1007/978-3-319-19066-2_56. dirección: https://link.springer.com/10.1007/978-3-319-19066-2_56 (visitado 31-12-2023) (vid. pág. 12).