

Universidad de La Habana
Facultad de Matemática y Computación



SRI-TOP: Sistema de Recuperación de Información basado en Tópicos

Autor: **Gelin Eguinosa Rosique**

Tutor: **Dr. Luciano García Garrido**

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



28 de noviembre del 2022

Agradecimientos

A mi querida madre Teresa, mi fuente de inspiración, gracias por tu amor, dedicación y sacrificios. A mi abuela Hilda, mi persona favorita, gracias por tu cariño y ternura, gracias por hacerme creer en mí. A mi padre Amaury, por brindarme siempre su apoyo. A mi primo Adrián, por ser el hermano que nunca tuve.

A mi tía Yamil y mi tío Alfonso, quienes me regalaron mi primera laptop. A mis tíos Hilda María, Tamara y Odalys, por venir siempre a mi rescate cuando más lo necesitaba. A mi tío Robert y mi primo Karl Lewis por su ayuda en los años finales de la carrera.

A mi tutor Luciano, por su infinita paciencia, por su apoyo y confianza, gracias por su guía y conocimientos. A la profesora Carmen por asegurarse de que terminara la carrera. A los profesores Wilfredo e Idania por sus lecciones sobre la carrera y la vida.

Bueno, tal vez el verdadero tesoro de mis estudios fueron los amigos que hice en el camino. A Wilber, Randy, Noel, Eduardo, Arturo, Linet, Marcos, Ricardo y Denise, amigos a los que admiro y respeto, gracias por estar siempre presentes y motivarme a ser la mejor versión de mí. A Yamilé y Yasmany, por ayudarme a regresar a la carrera y terminar mis últimos dos años. A Courtney y Celia, por estar siempre al tanto de como me iba en los estudios.

A toda mi familia, profesores de la facultad, amigos y compañeros de clase, gracias.

Opinión del tutor

Dada la explosión de información científica asequible en internet para la solución de problemas científicos, la búsqueda de procedimientos cada vez más efectivos y eficientes para su recuperación sigue siendo una tarea principal en Ciencia de la Computación y con imprescindible aplicación de los métodos de la Inteligencia Artificial.

El presente trabajo desarrolla una de las áreas más investigadas actualmente, la creación de sistemas de recuperación de información basados en tópicos. La importancia en la creación de tales sistemas es que los mismos mediante la descripción de documentos científicos según sus tópicos o temas, tiene en cuenta el contenido semántico conceptual de los documentos lo que hace la recuperación más orientada a los objetivos temáticos de la búsqueda.

El trabajo de diploma del estudiante Gelin Eguinosa Rosique constituye un modesto aporte en nuestro medio a la creación de los mencionados sistemas, habiendo el estudiante analizado una relevante y actualizada bibliografía al respecto. Los conocimiento y habilidades que muestra el estudiante en su trabajo de tesis hablan de su capacitación para su futuro trabajo profesional como Licenciado en Ciencias de la Computación.

Por lo anterior solicitamos al tribunal la calificación de excelente (5) para el estudiante Gelin Eguinosa Rosique.

Dr. Luciano García Garrido
Prof. Titular Consultante
Facultad de Matemática y Computación
Universidad de La Habana, Cuba

Resumen

La recuperación de información es una de las formas principales de acceder a información en la actualidad. Para garantizar el acceso a información de calidad cuando esta sea necesaria, se deben desarrollar técnicas novedosas que puedan adaptarse a las necesidades de los usuarios. En este trabajo desarrollamos un Sistema de Recuperación de Información basado en Tópicos (SRI-TOP). Para la construcción de este sistema, empleamos un Modelo con Representaciones Distribuidas de Tópicos (Top2Vec) usando los modelos SPECTER y Sentence-BERT en la representación de los documentos. Los Modelos de Tópicos construidos mejoran el estado del arte, obteniendo resultados más satisfactorios con respecto a modelos anteriores en cuanto a la calidad y capacidad descriptiva de sus tópicos. Como interfaz visual del SRI-TOP, creamos una aplicación que permite la realización de consultas y la interacción de los usuarios con los tópicos y documentos en el corpus.

Abstract

Information retrieval is one of the main ways to access information today. In order to guarantee access to quality information when necessary, innovative techniques must be developed that can be adapted to the needs of the user. In this work we develop a Topic-based Information Retrieval System (SRI-TOP). For the construction of this system, we used a Model with Distributed Representations of Topics (Top2Vec) using the SPECTER and Sentence-BERT models for the representation of the documents. The built Topic Models improve the state of the art, obtaining more satisfactory results compared to previous models in terms of the quality and descriptive capacity of their topics. As a visual interface for SRI-TOP, we created an application that allows queries and user interaction with the topics and documents in the corpus.

Índice general

Agradecimientos	I
Opinión del tutor	II
Resumen	III
Abstract	IV
Introducción	1
1. Recuperación de Información	3
1.1. Historia	4
1.2. Modelos	5
1.3. Operaciones	7
2. Representación de Documentos	9
2.1. One-Hot	9
2.1.1. Bolsa de Palabras	10
2.1.2. TF-IDF	11
2.1.3. Desventajas de los Modelos One-Hot	12
2.2. Doc2Vec	13
2.2.1. Word2Vec	13
2.2.2. Vector de Párrafo	14
2.2.3. Resumen Doc2Vec	15
2.3. Sentence-BERT	15
2.3.1. BERT	16
2.3.2. Modelo SBERT	18
2.3.3. Representación de Artículos con SBERT	19
2.4. SPECTER	20
2.4.1. SciBERT	20
2.4.2. Modelo SPECTER	21
3. Modelos de Tópicos	24
3.1. LDA	24
3.2. Top2Vec	26

3.2.1.	Representación Distribuida de Tópicos	27
3.2.2.	Creación del Espacio Semántico	28
3.2.3.	Encontrar el Número de Tópicos	28
3.2.4.	Calcular los Vectores de Tópicos	31
4.	Desarrollo del SRI-TOP	35
4.1.	Procesamiento del Corpus	36
4.1.1.	Corpus de Artículos Académicos CORD-19	36
4.1.2.	Creación del Vocabulario del Corpus	37
4.2.	Creación de los Modelos de Tópicos	38
4.2.1.	Modelo SBERT	39
4.2.2.	Modelo SPECTER-SBERT	44
4.3.	Evaluación de Modelos de Tópicos	48
4.3.1.	Ganancia de Información con Tópicos	48
4.3.2.	Comparación con Ganancia de Información	50
4.3.3.	Comparación utilizando Homogeneidad	56
4.4.	SRI-TOP	62
4.4.1.	Recuperación de Documentos	62
4.5.	Interfaz Visual	63
4.5.1.	Búsqueda de Información	64
4.5.2.	Explorar Tópicos	64
4.5.3.	Explorar Documentos	65
Conclusiones		67
Trabajos Futuros		68
Referencias		69

Índice de tablas

Desarrollo del SRI-TOP:	35
4.1. Tópicos del Modelo SBERT ordenados por cantidad de documentos	42
4.2. Tópicos del Modelo SPECTER-SBERT ordenados por cantidad de documentos	46
4.3. Tópicos del Modelo SBERT ordenados por su PWI-exact	52
4.4. Tópicos del Modelo SPECTER-SBERT ordenados por su PWI-exact	54
4.5. Tópicos del Modelo SBERT ordenados por Homogeneidad	57
4.6. Tópicos del Modelo SPECTER-SBERT ordenados por Homogeneidad	59

Índice de figuras

Representación de Documentos:	9
2.1. Procedimiento para el pre-entrenamiento y afinación de BERT	16
2.2. Arquitectura del modelo Sentence-BERT	18
2.3. Descripción general del modelo SPECTER	22
Modelos de Tópicos:	24
3.1. Ejemplo de un espacio semántico	29
3.2. Vectores de documentos generados a partir del <i>20 News Group</i>	30
3.3. Clústeres encontrados en el conjunto de datos <i>20 News Group</i>	32
3.4. Vector de tópico para un clúster de documentos	33
3.5. Vocabulario de un tópico en <i>Top2Vec</i>	34
Desarrollo del SRI-TOP:	35
4.1. Sistema de recopilación de artículos para CORD-19	37
4.2. Espacio Vectorial del Modelo de Tópicos SBERT	41
4.3. Espacio Vectorial SPECTER del Modelo de Tópicos SPECTER-SBERT	45
4.4. Comparación de Modelos con Ganancia de Información	51
4.5. Comparación de Modelos con Homogeneidad	56
4.6. Ejemplo de Tópicos con diferentes Homogeneidades	57
4.7. Pestaña <i>Search</i> para la realización de consultas	63
4.8. Pestaña <i>Topics</i> para la exploración de tópicos	64
4.9. Ventana para Vocabulario de un tópico en forma de texto	65
4.10. Visualización de Tópicos empleando Nube de Palabras	66
4.11. Pestaña <i>Documents</i> para la examinación de documentos	66

Introducción

La Recuperación de Información (RI) solía ser una actividad en la que solo participaban un pequeño grupo de personas: bibliotecarios, asistentes legales y buscadores de información profesionales. Ahora el mundo ha cambiado, y cientos de millones de personas se involucran todos los días en la recuperación de información cuando utilizan un motor de búsqueda web (como Google), una aplicación de RI (como Wikipedia) o simplemente realizan una búsqueda en su correo electrónico. La Recuperación de Información se ha convertido en la actualidad en la forma dominante de acceso a información.

Desde los inicios de la RI, los investigadores notaron lo difícil que era para los usuarios poder formular solicitudes de búsqueda efectivas. Una de las soluciones propuestas fue agregar a la consulta los sinónimos de las palabras en la solicitud realizada para mejorar la efectividad de la búsqueda. Las primeras investigaciones en RI se basaron en un diccionario de sinónimos para encontrar los sinónimos de las palabras [31], sin embargo, la creación de un buen tesauro de propósito general se consideró muy costosa. Los investigadores en lugar de utilizar un tesauro de propósito general, desarrollaron técnicas para generar automáticamente tesauros para su uso en la modificación de consultas. La mayoría de los métodos automáticos se basan en el análisis de la co-ocurrencia de palabras en los documentos (que a menudo produce una lista de palabras fuertemente relacionadas). Estas técnicas de expansión de consultas basadas en tesauros generados automáticamente, generalmente, tuvieron un éxito muy limitado en la mejora de la eficacia de la búsqueda [61]. La razón principal detrás de esto es la ausencia de una noción sobre el contexto de la consulta durante el proceso de expansión. No todos los sinónimos de una palabra de consulta son significativos en el contexto de la consulta realizada. Por ejemplo, aunque “máquina” es una buena alternativa para la palabra “motor”, esta expansión de la palabra no es significativa si la consulta es “motor de búsqueda”.

Nuevas técnicas para realizar una expansión sobre la consulta sin ninguna retroalimentación del usuario han sido desarrolladas. La más notable en este tipo de técnicas es la *pseudo-retroalimentación*, una variante de la retroalimentación de relevancia [13]. Dado que los documentos más relevantes dentro de los documentos recuperados por un Sistema de Recuperación de Información (SRI) representan a menudo el tópico general de la consulta, entonces la selección de términos relacionados con estos documentos debería generar nuevos términos útiles. En la *pseudo-retroalimentación*, el SRI asume que los primeros documentos recuperados en la consulta inicial del usuario son “relevantes” y realiza una retroalimentación de relevancia para generar una nueva consulta. Esta nueva consulta expandida es luego utilizada para crear el ranking de documentos que se presentará al usuario. La *pseudo-retroalimentación* ha demostrado ser una técnica muy efectiva, especialmente para consultas de pocas palabras [61].

A través de los años, muchas otras técnicas se han desarrollado con un éxito variable. Entre

ellas, la *hipótesis de clústeres* establece que los documentos agrupados juntos (muy similares entre sí) tendrán un perfil de relevancia similar para una determinada consulta [25]. Dada una colección de documentos, la *agrupación de documentos* se encarga de crear grupos de documentos similares en función de su contenido, similar a como se ordenarían los libros en un estante de una librería según su tópico. Las técnicas de *agrupación de documentos* siguen siendo un área activa de investigación. Aunque la utilidad de la *agrupación de documentos* para una mejor eficiencia en la búsqueda de información ha sido muy limitada, la *agrupación de documentos* ha permitido varios desarrollos en RI, por ejemplo, en la navegación de información y las interfaces de búsqueda.

Los *Modelos de Tópicos* son un enfoque muy popular en la representación del contenido de los documentos. Generalmente, la *modelación de tópicos* se utiliza cuando no es posible para una persona leer y organizar una gran colección de texto de forma razonable. Dado un corpus compuesto por muchos textos, denominados documentos, un *modelo de tópicos* descubrirá la *estructura semántica latente*, o *tópicos*, presentes en los documentos. Luego, los *tópicos* pueden ser usados para encontrar resúmenes de alto nivel en una colección grande de documentos, buscar documentos de interés y agrupar documentos similares [9, 8, 3]. Los *tópicos* se pueden ver como una descripción del contenido de la colección.

Una pregunta natural es si estos *tópicos* son útiles en la recuperación de documentos. Intuitivamente, los documentos relevantes para una consulta deben compartir el mismo tópico que la consulta, por lo que los *tópicos* serían una forma natural de aplicar *pseudo-retroalimentación* en el SRI y expandir las consultas de los usuarios. Además, los *tópicos* son una forma de *agrupación de documentos* por lo que la utilización de *modelos de tópicos* en la recuperación de información implica una mayor eficiencia en la búsqueda de documentos.

En este trabajo de tesis desarrollamos un SRI, para la recuperación de documentos científicos utilizando un Modelo de Tópicos con representación distribuida (Top2Vec) [3]. Empleamos los modelos Sentence-BERT y SPECTER para la representación de los documentos en el corpus, modelos con resultados vanguardias en la representación de texto y texto científico respectivamente. Obteniendo un sistema con resultados vanguardia en cuanto a la representación y la interpretabilidad de los tópicos.

En los siguientes capítulos primero introducimos la historia y características de los sistemas de recuperación de información. Luego, describimos las representaciones de documentos y los modelos de tópicos utilizados en el desarrollo del SRI. Después, introducimos el corpus de documentos científicos utilizados para desarrollar nuestro sistema. Mostramos los modelos de tópicos construidos y evaluamos sus rendimientos. Al final, vemos la aplicación visual utilizada para realizar las consultas en nuestro SRI empleando Modelos de Tópicos.

Capítulo 1

Recuperación de Información

La Recuperación de Información (RI) no comenzó con la Web, ya desde la década de 1940 el problema del almacenamiento y recuperación de información viene atrayendo de forma creciente mucha atención. El campo de RI surgió para proveer métodos efectivos en la búsqueda de diversos tipos de contenido, impulsado por la llegada de las computadoras y en respuesta a varios desafíos del acceso a la información con el objetivo de brindar un mejor servicio. Inicialmente, la RI se utilizó en publicaciones científicas y registros de bibliotecas, donde se crearon Sistemas de Recuperación de Información (SRI) para apoyar en tareas de indexación y acceso a colecciones de documentos, siendo extendido a otros campos paulatinamente [42]. Gran parte de la investigación científica sobre RI se ha producido en estos contextos, donde los SRI tratan de proporcionar acceso a información no estructurada en varios dominios corporativos y gubernamentales.

En los últimos años, uno de los principales impulsores de la innovación ha sido la World Wide Web, que ha provocado una explosión en la cantidad de información disponible a través de decenas de millones de creadores de contenido. La utilidad de este enorme conjunto de información publicada radica en la facilidad con que los usuarios puedan encontrarlas. Si un SRI ignora información relevante en la consulta de un usuario, esto puede conducir a la duplicación de trabajo y esfuerzo. Es por esto que en la actualidad la Recuperación de Información se ha convertido en uno de los principales campos investigativos de la Ciencia de la Computación y la Información, con el principal objetivo de brindar información con precisión y rapidez.

La Recuperación de Información consiste en encontrar material (generalmente documentos) de naturaleza no estructurada (texto fundamentalmente) que satisface una necesidad de información dentro de grandes colecciones (usualmente almacenadas en computadoras).

En principio, el almacenamiento y recuperación de información es simple. Supongamos que existe un almacén de documentos y una persona (usuario del almacén) formula una pregunta (consulta) a la que la respuesta es un conjunto de documentos que satisfacen la necesidad de información expresada por su pregunta. El usuario puede obtener este conjunto leyendo todos los documentos en el almacén, reteniendo los documentos relevantes y descartando todos los demás. Esto constituiría una recuperación “perfecta”, sin embargo, esta solución es claramente imprácticable, el usuario puede no tener el tiempo necesario o no desea perder el tiempo leyendo toda la colección de documentos. La RI se ocupa de la construcción de sistemas automáticos que permitan a los usuarios consultar datos textuales de cualquier tipo a través de consultas en lenguaje natural. La información recuperada de los SRI puede variar de una lista ordenada de elementos textuales relevantes de cualquier tipo,

como documentos completos o sus extractos, o pueden brindar resultados más elaborados, como resúmenes de documentos o respuestas a preguntas.

La recuperación de información es la ciencia de buscar información en un documento, buscar documentos en sí mismos y también buscar metadatos que describan objetos, como bases de datos de textos, imágenes, sonidos o videos. Un sistema de recuperación de información es un sistema de software que proporciona acceso a libros, revistas y otros documentos, en ocasiones también encargándose del almacenamiento y administración de estos documentos [41]. El proceso de recuperación de información comienza cuando un usuario ingresa una consulta en el sistema. Las consultas son declaraciones formales de necesidades de información, por ejemplo, **strings** en motores de búsqueda web. En la recuperación de información, una consulta no identifica de manera única un solo objeto de la colección, varios objetos pueden coincidir con la consulta, quizás con diferentes grados de relevancia. A menudo, los documentos en sí mismos no se guardan o almacenan directamente en el SRI, sino que se representan en el sistema mediante metadatos o sustitutos de documentos.

La mayoría de los SRI calculan una puntuación numérica representando cuánto cada objeto en la base de datos coincide con la consulta, y ordenan los objetos de acuerdo a este valor. Los objetos de mayor similitud a la consulta son entonces mostrados al usuario. Luego, si el usuario desea refinar la consulta, se puede realizar otra iteración en el proceso de recuperación de información, a esta última iteración se le llama generalmente retroalimentación.

Entre los principales desafíos actuales de los SRI está su capacidad de extraer información sintáctica y semánticamente relevante del texto de los documentos para decidir si un documento en particular es de interés o no con respecto a una consulta. La dificultad no está solo en saber cómo extraer la información, sino también en saber cómo utilizarla para decidir la relevancia del documento.

1.1. Historia de la Recuperación de Información

La práctica de archivar información escrita se remonta al tercer milenio a. C., cuando en Sumeria, la civilización más antigua conocida perteneciente a la región sur de Mesopotamia (centro-sur de Irak), se comenzaron a designar áreas especiales para almacenar tablillas de arcilla con inscripciones en cuneiforme, el sistema de escritura más antiguo de la humanidad. Incluso en esta etapa, los sumerios se dieron cuenta de que la organización e identificación adecuada de los archivos eran fundamentales para el acceso y uso eficiente de la información almacenada. Para esto, desarrollaron clasificaciones especiales para localizar e identificar cada tableta y su contenido [61].

La necesidad de almacenar y recuperar información escrita se volvió cada vez más importante a lo largo de los siglos, especialmente con inventos como el papel y la imprenta. Después de la invención de las computadoras, no se tardó mucho en reconocer la utilidad de estas para el almacenamiento y la recuperación mecánica de grandes cantidades de información. En 1945, Vannevar Bush publicó un artículo innovador titulado “As We May Think” que dio origen a la idea del acceso automático a grandes cantidades de conocimiento almacenado [14]. En la década de 1950, esta idea se materializó en descripciones más concretas de cómo se podían realizar búsquedas automáticas en los archivos de texto y varios trabajos surgieron a mediados de la década de 1950 que elaboraron sobre la idea básica de buscar texto empleando una computadora. Uno de los métodos más influyentes fue descrito por H.P. Luhn en 1957, en el que (de forma simple) propuso usar palabras como unidades de indexación para los documentos y emplear la superposición de palabras como criterio en la recuperación [40].

Varios avances clave en el campo de la recuperación de información ocurrieron durante la década de 1960. Una de las principales figuras que surgieron en este período fue Gerard Salton, quien trabajo

en el desarrollo del sistema SMART primero en la Universidad de Harvard y luego en Cornell [58]. Salton también trabajó en la formalización de algoritmos para crear ranking de documentos con respecto a una consulta. De particular interés fue su enfoque en el que los documentos y las consultas se veían como vectores dentro de un espacio con dimensión N , siendo N el número de términos únicos en la colección de documentos. Para medir la similitud entre un vector de documento y un vector de consulta Salton sugirió utilizar el coseno del ángulo entre los vectores [57, 59]. Otro de los avances en el campo fueron las evaluaciones de Cranfield realizadas por Cyril Cleverdon y su grupo en la Facultad de Aeronáutica de Cranfield [18]. Las pruebas de Cranfield desarrollaron una metodología de evaluación para los sistemas de recuperación que todavía es utilizada por los SRI en la actualidad. Los nuevos algoritmos junto con la nueva metodología de evaluación permitió un rápido progreso en el campo, y allanó el camino para muchos desarrollos críticos.

Las décadas de 1970 y 1980 vieron muchas innovaciones basadas en los avances de la década de 1960. Fueron desarrollados varios modelos para la recuperación de documentos y se lograron avances en todas las dimensiones del proceso de recuperación. Muchos de estos modelos demostraron ser efectivos en pequeñas colecciones de texto (varios miles de artículos), sin embargo, debido a la falta de disponibilidad de grandes colecciones de texto, la pregunta de que si estos modelos y técnicas podían ser adaptados a corpus más grandes quedaba sin respuesta. Esto cambió en 1992 con el inicio de la Conferencia de Recuperación de Texto, o TREC¹ [27]. TREC es una serie de conferencias de evaluación patrocinadas por varias agencias gubernamentales de los Estados Unidos bajo el auspicio del Instituto Nacional de Normas y Tecnologías (NIST²), con el objetivo fomentar la investigación en el campo de la RI utilizando grandes colecciones de textos.

Con la disponibilidad de grandes colecciones de texto en TREC, se modificaron muchas técnicas antiguas y se desarrollaron nuevas técnicas para realizar una recuperación eficaz en grandes colecciones. Con la introducción de los motores de búsqueda web a finales de 1993, aumentó la necesidad de sistemas de recuperación a gran escala empujando aún más el desarrollo nuevos métodos y técnicas en la RI.

1.2. Modelos de Recuperación de Información

Los modelos de Recuperación de Información se encargan de seleccionar y clasificar los documentos relevantes con respecto a la consulta de un usuario. Los textos de los documentos y las consultas de los usuarios utilizan el mismo tipo de representación, de forma tal que la selección y la clasificación de documentos se pueda formalizar mediante una función de coincidencia que devuelva un valor de estado de recuperación para cada documento de la colección [36].

Generalmente, los SRI representan los contenidos de los documentos y las consultas mediante un conjunto de descriptores, denominados términos, pertenecientes a un vocabulario V . Los modelos de RI utilizan varios enfoques para definir la función de coincidencia entre una consulta y los documentos, entre ellos están:

- Utilizar una función de similitud para calcular la distancia entre consultas y documentos en un espacio vectorial.

$$Sim(d, q) \quad (1.1)$$

¹del inglés Text REtrieval Conference

²del inglés National Institute of Standards and Technology

- Estimar la probabilidad de la relevancia rel del documento d para el usuario dada la consulta q y el conjunto de documentos D en el corpus.

$$P(rel|d, q, D) \quad (1.2)$$

Para poder recuperar documentos relevantes de manera efectiva utilizando alguna de las estrategias de RI mencionadas, los documentos deben ser transformados a una representación adecuada para el enfoque seleccionado. Cada estrategia de recuperación incorpora un modelo específico para sus propósitos en la representación de documentos.

Los modelos de Recuperación de Información pueden ser clasificados en tres tipos clásicos: Modelos de Teoría de Conjuntos, Modelos Algebraicos y Modelos Probabilísticos.

Los modelos de teoría de conjuntos representan documentos como conjuntos de palabras o frases. Las similitudes son generalmente derivadas de operaciones teóricas de conjuntos sobre sus conjuntos. Los modelos más comunes son:

- Modelo Estándar Booleano
- Modelo Booleano Extendido
- Modelo de Recuperación Difusa

Los Modelos Algebraicos generalmente representan los documentos y las consultas como vectores, matrices o tuplas. El vector de la consulta y el vector de los documentos se utiliza para encontrar la similitud entre ellos, siendo la función coseno uno de los métodos más utilizados en este tipo de modelo. Entre los modelos algebraicos más comunes están:

- Modelo de Espacio Vectorial
- Modelo de Espacio Vectorial Generalizado
- Modelo de Espacio Vectorial Basado en Tópicos
- Modelo Booleano Extendido
- Modelo de Semántica Latente

Los Modelos Probabilísticos tratan el proceso de recuperación de documentos como una inferencia probabilística. Las similitudes se calculan como las probabilidades de que un documento sea relevante para una consulta determinada. Los teoremas probabilísticos como el teorema de Bayes son utilizados a menudo en este tipo de modelos. Entre los más comunes están:

- Modelo de Independencia Binaria
- Modelo de Relevancia Probabilística
- Modelos de Lenguaje
- Latent Dirichlet Allocation (LDA)

1.3. Operaciones de Recuperación de Información

Un sistema de recuperación de información para colecciones de documentos con texto no estructurado puede verse como un conjunto de módulos de procesamiento, que comienza con el análisis léxico de los documentos y que conduce a un proceso final de recuperación en el que las consultas de los usuarios se comparan con los documentos. Los procesos que componen el sistema se denominan Operaciones de Recuperación de Información. En algunos casos, el objetivo final no es la recuperación de los documentos que coinciden con la consulta dentro de la colección, sino un objetivo auxiliar, como la categorización, el resumen o el filtrado de los documentos provenientes de un flujo de información. Si bien las operaciones básicas en los sistemas de recuperación de información son similares, puede haber una variabilidad considerable en el modelo o los detalles de la implementación, por lo que la evaluación de la efectividad del sistema es también un paso importante.

Las operaciones básicas en la construcción de un sistema de recuperación de información son el preprocessamiento del texto, la creación de los archivos resultado de la indexación de los documentos y el procesamiento de las consultas para proporcionar (generalmente) una lista ordenada con los documentos potencialmente relevantes en la colección.

El preprocessamiento de los documentos implica analizar sus textos para la creación de una lista de términos de índice (léxico) que se almacenarán para su recuperación. Durante la creación del índice, algunas palabras pueden identificarse como *palabras vacías*³, palabras comunes que no se consideran significativas por lo que no es necesario indexarlas. A las palabras con sufijos comunes se les encuentra su raíz a través de una operación llamada *stemming* y son representadas bajo una sola entrada del índice por su palabra raíz. Si bien es posible seleccionar un subconjunto de las palabras en un documento para crear los términos del índice, generalmente se emplean todas las palabras en el texto (excepto las *palabras vacías*), por lo que a este tipo de representación se le llama *bolsa de palabras*⁴.

Desde un inicio, las investigaciones sobre RI demostraron el valor de usar pesos diferenciales para los términos del índice asociados a un documento, estableciendo de esta forma la relevancia de los términos en los documentos, y entre los documentos y las solicitudes de los usuarios. Uno de los enfoques más antiguo y efectivo es *tf-idf*⁵, o Frecuencia del Término - Frecuencia Inversa en Documentos, en el que a cada término en un documento se le asigna un peso proporcional a su frecuencia en el documento e inversamente proporcional a su frecuencia en el corpus. La investigación de diferentes modelos para la asignación de pesos a los términos sigue siendo un área activa en la recuperación de información.

Luego de asignar su peso a cada término, los valores de la matriz documento-término resultante están esparcidos y no son eficientes para su almacenamiento y procesamiento por el sistema, por lo que el siguiente grupo de operaciones consiste en crear el índice para el almacenamiento de los términos y los pesos asociados a los términos en cada documento. Generalmente se utiliza el *índice invertido* (inverted index), donde por cada término se crea una lista con los documentos en los que aparecen y el peso del término para cada uno de estos documentos. El propósito del *índice invertido* es permitir búsquedas rápidas de texto completo, a cambio de un mayor procesamiento cuando se agrega un documento a la base de datos. Para colecciones de datos grandes, también se puede aplicar alguna forma de compresión de índice [42, 70].

³también llamadas Stop-Words

⁴del inglés Bag-of-words

⁵del inglés term frequency-inverse document frequency

Finalmente, un grupo de operaciones sobre la consulta del usuario dan como resultado los documentos que se entregan en la salida del sistema. La consulta se entrega al SRI a través de una interfaz de usuario, y se somete al mismo procesamiento que se empleó sobre la colección de documentos para identificar los términos del índice que contiene la consulta. El orden con el que se procesan los términos de la consulta puede variar con el objetivo tener un procesamiento más eficiente [36]. La consulta se compara con los documentos de la colección mediante la función de similitud del modelo para crear un ranking de los documentos por el orden de similitud. La función de similitud empleada depende del modelo de recuperación de información utilizado, por ejemplo, en los Modelos de Espacio Vectorial la función de similitud suele ser la función coseno que mide el ángulo entre el documento y el vector de la consulta.

Más operaciones sobre la consulta pueden realizarse con el objetivo de mejorar los resultados. Utilizando *retroalimentación de relevancia* (relevance feedback), la información sobre los documentos en el ranking de los que se sabe que son relevantes (a partir de la actividad del usuario) o se supone que son relevantes, se utiliza para volver a ponderar los términos de la consulta y ofrecer potencialmente una mejora en los resultados. En la *expansión de consulta* (query expansion), se agregan términos a la consulta, automáticamente o con ayuda del usuario, para mejorar el rendimiento de la recuperación.

El proceso de recuperación descrito anteriormente realiza lo que a veces se denomina tarea de recuperación *ad hoc*, en la que la consulta se compara con una colección relativamente estática de documentos. En algunas situaciones, es la consulta la que es (relativamente) estática y son los documentos los que cambian. Hacer coincidir una consulta con un flujo de documentos se denomina *filtrado de información* y es útil en situaciones en las que se debe monitorear información que es creada de forma constante (por ejemplo, en un periódico).

En algunos casos, el objetivo final puede no ser producir documentos en respuesta a una consulta, sino realizar alguna otra tarea a través del procesamiento especializado de la colección de documentos indexados, a menudo como apoyo a la recuperación de información. Un ejemplo es la *categorización de documentos*, mediante la cual los documentos que se agregan a una colección se asignan automáticamente a lo que se prevé que sea la categoría más apropiada para ellos, basado en las características observadas en la categoría (como los patrones de los términos dentro de la categoría). Normalmente, primero se clasifica manualmente un subconjunto de los documentos de la colección, para luego clasificar automáticamente el resto de la colección (o nuevos documentos). En la *agrupación de documentos* (document clustering), la similitud documento-documento se utiliza para dividir el corpus en grupos de documentos similares. La *agrupación* de documentos en clústeres difiere de la *categorización* en que los clústeres no se conocen a priori. La *agrupación de documentos* se puede emplear en la salida del SRI, para proporcionar al usuario documentos organizados por temas. De manera similar, en el *resumen de documentos* (document summarization), la información sobre un documento se emplea para identificar las palabras, oraciones o conceptos más importantes que contiene para construir un resumen del mismo, por ejemplo, para presentarlo como un sustituto del documento en el buscador. Otra operación posterior a la recuperación es la *visualización de la información* [72], en la que se genera una visualización en lugar de, o además de, el ranking de los documentos. En algunos casos, la presentación visual se puede manipular para proporcionar una interpretación adicional a los resultados de la recuperación.

Capítulo 2

Representación de Documentos

El documento es uno de los conceptos más importantes en las Ciencias de la Información y el Procesamiento del Lenguaje Natural (NLP). Todos los tipos de textos son documentos. Un artículo, un libro, un párrafo o incluso una oración pueden ser tratados como un documento.

Los avances en las tecnologías de la información, la comunicación y el internet han provocado un aumento exponencial del número de documentos disponibles en línea. Si bien cada vez hay más información textual disponible electrónicamente, la recuperación y minería de información efectiva se vuelven cada vez más difíciles sin la organización, el resumen y la indexación del contenido de los documentos de forma eficiente. Es por esto que la representación de documentos juega un papel fundamental en las ciencias de la información, teniendo muchas aplicaciones en el mundo real, por ejemplo, en las búsquedas web, la recuperación de documentos y el filtrado de spam [37].

Cuando se trabaja con algoritmos para el Procesamiento de Lenguaje Natural (NLP) en Machine Learning los documentos deben ser representados como vectores. Esta representación debe describir de forma efectiva y con la menor complejidad posible el contenido de los documentos. A continuación veremos varios de los modelos utilizados para la representación de documentos.

2.1. One-Hot

Entre los modelos tradicionales de representación de documentos se encuentra el modelo One-Hot, donde el algoritmo de encoding One-Hot es utilizado para generar un vector de longitud igual al número de palabras contenidas en el vocabulario.

En una representación One-Hot, dado un vocabulario fijo de n palabras para la colección de documentos $V = \{w_1, w_2, \dots, w_n\}$, las palabras son codificadas con un vector n-dimensional w , donde las dimensiones de w tienen como valor 0 o 1. Solo una dimensión de w puede ser 1, el resto debe tener 0 como valor, y la dimensión de valor 1 corresponde a la palabra del vocabulario que el vector está representando. Formalmente, cada dimensión del vector w se representa de la siguiente forma:

$$w_i = \begin{cases} 1 & \text{si } w = w_i, \text{ donde } w_i \in V \\ 0 & \text{cualquier otro caso.} \end{cases} \quad (2.1)$$

En esencia, los vocabularios de las representaciones One-Hot mapean cada palabra del vo-

cabulario a un índice en específico. Debido a esto, esta representación es muy eficiente para el almacenamiento y la computación de palabras y documentos.

Los Modelos One-Hot se usan comúnmente en métodos de clasificación de documentos donde la ocurrencia (y frecuencia) de cada palabra se usa como característica para entrenar un clasificador. Debido a su simplicidad y eficiencia, este modelo ofrece muy buenos resultados en muchas de las tareas de clasificación y agrupación de documentos [37].

2.1.1. Bolsa de Palabras

El modelo Bolsa de Palabras o Bag of Words (BoW), es el modelo de representación de documentos más simple y común. En este modelo, un documento se representa como la bolsa (multi-conjunto) de sus palabras, sin tener en cuenta la gramática o el orden de las palabras, pero manteniendo el número de ocurrencias que tiene cada palabra (multiplicidad).

Para un documento d , se puede utilizar la representación en Bolsa de Palabras $d = [w_1, w_2, \dots, w_m]$ para representar el documento, utilizando un vocabulario fijo $V = \{w_1, w_2, \dots, w_n\}$ y las representaciones one-hot de las palabras $w_i = [0, 0, \dots, 1, \dots, 0]$ donde w_i tiene valor 1 en la posición i y 0 en el resto de las posiciones. El documento puede ser representado en Bolsa de Palabras de la siguiente forma:

$$d = \sum_{k=1}^m w_i \quad (2.2)$$

donde m es la longitud del documento d . Veamos un ejemplo modelando dos documentos sencillos:

Doc 1: Laura encuentra la comida deliciosa. Carlos también encuentra la comida deliciosa.

Doc 2: Carlos encuentra la televisión aburrida.

Dado los documentos, el vocabulario sería:

$$V = [\text{Laura, encuentra, la, comida, deliciosa, Carlos, también, television, aburrida}] \quad (2.3)$$

utilizando este vocabulario obtenemos las representaciones One-Hot de las palabras, por ejemplo, el vector de “comida” es $w_4 = [0, 0, 0, 1, 0, 0, 0, 0, 0]$, y la representación de los documentos en Bolsas de Palabras quedaría de la siguiente forma:

$$\begin{aligned} d_1 &= [1, 2, 2, 2, 2, 1, 1, 0, 0] \\ d_2 &= [0, 1, 1, 0, 0, 1, 0, 1, 1] \end{aligned} \quad (2.4)$$

En esta representación, cada dimensión d_i de un vector d indica el número de ocurrencias que tiene en el documento la palabra correspondiente a esta dimensión, w_i . El orden de los valores en los vectores se corresponden con la ubicación de las palabras en el vocabulario, por lo que los vectores no preservan el orden original de las palabras en los documentos que están representando.

2.1.2. TF-IDF

La frecuencia de los términos utilizada en Bolsa de Palabras, no es necesariamente la mejor forma de modelar un documento, ni valorar la importancia que tiene cada palabra dentro del texto. Palabras comunes como ‘de’, ‘la’, ‘el’, ‘que’, entre otras, son casi siempre los términos de mayor frecuencia en un documento, y estas brindan muy poca información con respecto al contenido y los temas abordados en el documento. Es por esto que tener un valor alto de frecuencia no significa necesariamente que la palabra correspondiente tiene una mayor relevancia. Para solucionar este problema, una de las formas más populares de *normalizar* el peso de los términos es multiplicando la frecuencia de la palabra por su frecuencia inversa en el corpus para obtener lo que llamamos la Frecuencia del Término - Frecuencia Inversa en Documentos o TF-IDF¹.

TF-IDF tiene como intención indicar cuán importante es una palabra dentro de un Documento con respecto a la colección de documentos o corpus. El valor TF-IDF de una palabra aumenta proporcionalmente al número de ocasiones que una palabra aparece en un documento, y disminuye dependiendo del número de documentos en el corpus que contengan la palabra, mientras más documentos contengan la palabra menor será su valor. Esta representación disminuye significativamente la relevancia de aquellas palabras que aparecen en el corpus de forma frecuente, por lo que palabras comunes dentro del corpus tendrían mucha menor relevancia (o peso) dentro de la representación de un documento [34].

Siendo TF-IDF el producto de dos estadísticas, la frecuencia del término y su frecuencia inversa en documentos, veamos como calcular estos valores.

La frecuencia del término, $tf(t, d)$, es la frecuencia relativa del término t dentro del documento d ,

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (2.5)$$

donde $f_{t,d}$ es la cantidad de veces que aparece el término t en el documento d . El denominador no es más que el número total de términos que contiene el documento d , contando las ocurrencias de cada término de forma separada.

La frecuencia inversa en documentos es una medida de cuanta información es proporcionada por la palabra, es decir, si es un término común o raro de encontrar en el resto de los documentos. Su valor es la fracción inversa de los documentos que contienen la palabra escalada logarítmicamente, obtenida al dividir el número total de documentos por el número de documentos que contienen el término, para luego calcular el logaritmo de este cociente:

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (2.6)$$

con N siendo el número total de documentos en el corpus $N = |D|$, y con $|\{d \in D : t \in d\}|$ siendo el número de documentos donde el término t aparece. Si el término no esta en el corpus, se realizará una división por cero, por lo que es común ajustar el denominador a $1 + |\{d \in D : t \in d\}|$

Una vez que tenemos los valores de la frecuencia del término y su frecuencia inversa en documentos, podemos calcular su TF-IDF utilizando la siguiente formula:

$$tf-idf(t, d, D) = tf(t, d) \cdot idf(t, D) \quad (2.7)$$

¹del inglés Term Frequency-Inverse Document Frequency

Un término tiene un peso alto de TF-IDF, si tiene una alta frecuencia en el documento dado y poca presencia en el resto de los documentos del corpus; por lo tanto, este modelo tiende a filtrar las palabras comunes.

Recordando los documentos d_1 y d_2 vistos anteriormente:

Doc 1: Laura encuentra la comida deliciosa. Carlos también encuentra la comida deliciosa.

Doc 2: Carlos encuentra la televisión aburrida.

y el vocabulario V generado por estos documentos *Fórmula 2.3*,

$$V = [\text{Laura, encuentra, la, comida, deliciosa, Carlos, también, television, aburrida}]$$

podemos generar la representación TF-IDF para estos documentos:

$$\begin{aligned} d_1 &= [0.06, 0, 0, 0.13, 0.13, 0, 0.06, 0, 0] \\ d_2 &= [0, 0, 0, 0, 0, 0, 0, 0.14, 0.14] \end{aligned} \tag{2.8}$$

En los vectores generados podemos observar que las palabras *comida*, *deliciosa* son las de mayor relevancia para el documento d_1 , y las palabras *televisión*, *aburrida* tienen la mayor relevancia en el documento d_2 . Esto muestra una pequeña mejora con respecto a la representación en bolsa de palabras, ya que en esta representación palabras como *la* tenían la misma relevancia que *comida* y *deliciosa*.

La representación de documentos utilizando TF-IDF refleja con más facilidad la importancia que tiene una palabra dentro de un documento con respecto a un corpus, en este aspecto supera al modelo de Bolsa de Palabras. Por esto, TF-IDF es uno de los esquemas de ponderación de términos más populares en la actualidad. [37]

2.1.3. Desventajas de los Modelos One-Hot

La Representación de Documentos con One-Hot tiene varias desventajas. Cuando un documento es llevado a esta representación, el orden de las palabras se pierde, por lo que diferentes documentos pueden estar representados por el mismo vector si utilizan las mismas palabras. También, debido a que las palabras son representadas por vectores de n dimensiones, donde n es el tamaño fijo del vocabulario, este modelo no es capaz de trabajar con documentos que contengan palabras fuera del vocabulario. Esta inflexibilidad puede afectar su rendimiento en escenarios del mundo real [37].

En esencia, la representaciones de palabras One-Hot asigna cada palabra a un índice del vocabulario. Esto puede ser muy eficiente para el almacenamiento y el cálculo, pero no guarda ninguna de las relaciones semánticas y sintácticas entre las palabras. Con esta representación existe tanta diferencia entre las palabras “gato” y “perro”, como en las palabras “gato” y “cama”. Por lo tanto, las distancias en el espacio vectorial creado no reflejan la distancia en significado de las palabras [37].

A pesar de sus desventajas, este modelo sigue siendo uno de los más empleados debido a su eficiencia en almacenamiento y computación.

2.2. Doc2Vec

Como hemos mencionado anteriormente, a pesar de su simplicidad y eficiencia, los modelos One-Hot tienen varias desventajas. Entre ellas está que no captan las relaciones semánticas entre las palabras, lo que significa que en el espacio de vectores palabras como “grande”, “enorme” y “aerolínea” tienen la misma distancia a pesar de que “enorme” debería estar mucho más cerca de “grande” que de “aerolínea”. Word2Vec y Doc2Vec son una propuesta de solución a este problema [49, 33].

Word2Vec es un modelo reciente del año 2013, donde las palabras se colocan en un espacio vectorial utilizando una red neuronal poco profunda. Como resultado se obtiene un conjunto de vectores donde las palabras cercanas en el espacio vectorial tienen significados similares, y los vectores de palabras distantes entre sí tienen significados diferentes.

Con el modelo Word2Vec, podemos calcular los vectores para cada palabra en un documento. Para calcular el vector de todo el documento podríamos promediar los vectores de cada palabra en el documento. Este sería un método rápido y simple que puede ser de utilidad, pero existe una mejor opción: Doc2Vec.

El Vector de Párrafo (Doc2Vec) [33], es un sistema no supervisado para aprender representaciones vectoriales distribuidas continuas de segmentos de texto. El nombre, Vector de Párrafo, es para enfatizar el hecho de que este método puede ser aplicado a textos de distintas longitudes, cualquier documento sin importar su tamaño (grande o pequeño) puede ser representado utilizando Doc2Vec, desde oraciones hasta documentos extensos.

En este modelo, los vectores son entrenados para ser útiles en la predicción de palabras en un párrafo. De forma más precisa, el vector del párrafo es concatenado con varios vectores de palabras en un párrafo para predecir que palabra viene a continuación en un contexto dado. Los vectores de párrafos son únicos entre documentos, pero los vectores de las palabras son compartidos. Las palabras también utilizan una representación vectorial distribuida, Word2Vec [49].

Una vez que el modelo es entrenado, los vectores de las palabras y los documentos son asignados a un espacio vectorial de modo que las palabras semánticamente similares tienen representaciones vectoriales similares, por ejemplo, la palabra “enorme” quedará cerca de “grande”.

2.2.1. Word2Vec

Antes de introducir los detalles del modelo Doc2Vec, primero veamos el modelo de vectores utilizado para representar las palabras. Word2Vec [49], es un sistema para aprender la representación vectorial de una palabra dadas las palabras que la rodean en un contexto.

Word2Vec utiliza una red neuronal poco profunda para crear las representaciones vectoriales de las palabras. Existen dos versiones de este modelo:

1. Skip-gram Continuo (SG)
2. Bolsa de Palabras Continuas (CBOW²)

En ambas arquitecturas, Word2Vec considera una palabra individual w_i dentro de una ventana deslizante de palabras de contexto $w_{i-k}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+k}$ que rodean a la palabra individual, siendo k el tamaño de la ventana de texto. Con estas palabras se crean los pares de palabras

²del inglés *Continuous Bag of Words*

(w_i, w_j) , donde w_i es una palabra individual y w_j es una de las palabras que rodea a w_i en el contexto dado.

En el modelo SG, se emplean los pares de palabras (w_i, w_j) para entrenar una red neuronal de una capa oculta utilizando la tarea de dada la palabra de entrada w_i predecir las palabras w_j que la rodean dentro de su contexto. Luego, el número de neuronas empleadas en la capa oculta se usan para crear los embeddings de las palabras. Por ejemplo, si la capa oculta tiene 300 neuronas, esta red nos dará embeddings de palabras de 300 dimensiones.

El modelo CBOW es similar al modelo SG, también empleando una red neuronal de una capa oculta. Pero en este caso, para entrenar la red neuronal se emplea la tarea de predecir la palabra central w_i dada las palabras w_j que la rodean dentro de su contexto. La capa oculta resultante se emplea para crear los embeddings de las palabras.

La arquitectura SG le da un mayor peso a las palabras cercanas dentro del contexto que a las palabras más alejadas. CBOW es más eficiente, mientras que SG tiene un mejor rendimiento y representa mejor las palabras poco frecuentes.

Como resultado, los embeddings de las palabras con un significado similar son asignadas a una posición similar en el espacio vectorial. Por ejemplo, “grande” y “enorme” quedarían cerca, mientras que “grande” y “aerolínea” quedarían más distantes. La diferencia entre los vectores de palabras también tendrían significado, y los vectores de las palabras se podrían usar para responder preguntas de analogía usando álgebra vectorial simple, por ejemplo, en este espacio vectorial Rey – Hombre + Mujer = Reina [52].

Estas propiedades hacen que los vectores de palabras sean atractivos para muchas tareas del Procesamiento del Lenguaje Natural, como el Modelado de Lenguaje [6, 48], Comprensión del Lenguaje Natural [20, 73], la Traducción Automática [50, 75] y la Comprensión de Imágenes [23].

2.2.2. Vector de Párrafo

El enfoque para aprender *vectores de párrafo* está inspirado en los métodos utilizados para aprender vectores de palabras. La inspiración viene de la forma en que se requiere que los vectores de palabras contribuyan en la tarea de predecir cuál es la próxima palabra en una oración. Por lo que, a pesar del hecho de que los vectores son inicializados aleatoriamente, estos pueden eventualmente capturar la semántica de las palabras como un resultado indirecto de la tarea de predicción. Esta idea es usada de forma similar en los *vectores de párrafo*. En el caso de los vectores de párrafo, a estos se les pide que contribuyan en la tarea de predicción de la siguiente palabra dado un contexto extraído aleatoriamente del párrafo [33].

Existen dos implementaciones de este modelo:

1. Vector de Párrafo - Memoria Distribuida (PV-DM³)
2. Vector de Párrafo - Bolsa de Palabras Distribuida (PV-DBOW⁴)

PV-DM trabaja de forma similar al modelo CBOW de *Word2Vec*. Donde los vectores de los documentos se obtienen al entrenar una red neuronal con la tarea de predecir el vector del documento y la palabra central dadas las palabras del contexto.

PV-DBOW es análogo al modelo SG de *Word2Vec*. Las representaciones vectoriales de los documentos se obtienen entrenando una red neuronal con la tarea de predecir palabras aleatorias

³del inglés Paragraph Vector - Distributed Memory

⁴del inglés Paragraph Vector - Distributed Bag of Words

extraídas de los documentos empleando solamente el vector del documento. Para esto, en cada iteración del proceso de entrenamiento, se selecciona una ventana de texto y se extrae una palabra aleatoria dentro de la ventana de texto, creando una tarea de clasificación dado el vector de párrafo del documento.

2.2.3. Resumen Doc2Vec

Los vectores de párrafo solucionan varias de las debilidades presentes en los modelos One-Hot. Primero, heredan una importante propiedad de los vectores de palabra: la semántica de las palabras. En este espacio, “grande” está más cerca de “enorme” que de “aerolínea”. Otra ventaja es que los vectores de párrafo tienen en cuenta el orden de las palabras [33], al menos en un contexto pequeño, de la misma forma que lo haría un modelo N-gram con N grande. Y en comparación con un modelo N-gram de documentos es mejor, porque un modelo N-gram crea vectores de dimensiones muy altas, mientras Doc2Vec es capaz de representar los documentos de una forma mucho más simple.

Después de ser entrenados, los vectores de párrafos pueden ser usados como características de documentos, por ejemplo, en lugar de o además de la bolsa de palabras. Estas características se pueden utilizar como entrada en otros sistemas de Machine Learning.

De los dos modelos, PV-DM tiene el mejor rendimiento y por si solo suele funcionar bien en la mayoría de las tareas, aunque su combinación con PV-DBOW suele ser más consistente.

2.3. Sentence-BERT

Después de la introducción de los modelos Word2Vec y Doc2Vec [49, 33], el uso de Redes Neuronales en el pre-entrenamiento de Modelos de Lenguajes se intensificó, siendo un tema de investigación muy activo en el Procesamiento del Lenguaje Natural. Nuevos enfoques en la representación de texto fueron creados, entre ellos, los embeddings de oraciones [32, 39]. Para entrenar las representaciones de oraciones se utilizaron nuevos objetivos de entrenamiento, como dado un grupo de oraciones candidatas seleccionar cuál es la próxima oración [30, 39], o la generación de izquierda-a-derecha de las palabras de la próxima oración dada la representación de la oración anterior [32].

Para adaptar mejor los modelos a las tareas donde serán utilizados, se emplearon nuevos enfoques no supervisados de afinación⁵. Para la afinación de los modelos, los codificadores de oraciones y documentos son primero entrenados utilizando texto sin etiquetar, y luego de forma supervisada se ajustan los parámetros de codificación según el objetivo final del modelo [21, 29, 54]. La ventaja de estos enfoques es que pocos parámetros deben ser aprendidos desde cero.

Con la introducción de BERT: Representación Bidireccional de Codificadores de Transformers [22], se logró avanzar mucho más el estado del arte. BERT logra generar representaciones de mayor calidad utilizando un “Modelo de Lenguaje Enmascarado” (MLM), inspirado en la tarea Cloze [64]. Este modelo enmascara aleatoriamente algunos de los tokens en el texto de entrada, con el objetivo de predecir cuáles fueron las palabras enmascaradas basándose únicamente en su contexto. BERT utiliza Transformers [66], un modelo de aprendizaje profundo que adopta el mecanismo de auto-atención. En su arquitectura se emplean un número variable de capas codificadoras y cabezas de auto-atención, lo que le permite tener distintas representaciones vectoriales para una palabra dependiendo de su contexto.

⁵del inglés fine-tuning

BERT estableció nuevos resultados de vanguardia en varias tareas del Procesamiento del Lenguaje Natural (NLP), incluyendo preguntas-respuestas, clasificación de oraciones y la Evaluación General de Comprensión del Lenguaje (GLUE) [22, 67].

Con todas sus ventajas, una gran desventaja de la estructura del sistema BERT es que no se calculan los embeddings de oraciones independientes, lo que dificulta la tarea de representación de texto utilizando BERT. Para resolver esta limitación, usaremos el modelo Sentence-BERT: Embedding de Oraciones usando Redes BERT Siamesas (SBERT) [56]. Este sistema utiliza un modelo BERT previamente entrenado con una estructura de Red Siamesa y de Tripletes [60], para deducir embedding de oraciones semánticamente significativos que se puedan comparar empleando la similitud de coseno. Este modelo reduce el esfuerzo para encontrar el par más similar de 65 horas con BERT a unos 5 segundos con SBERT, manteniendo la misma precisión en las representaciones vectoriales del texto [56].

Antes de introducir el modelo Sentence-BERT, veamos con más detalles el funcionamiento del sistema BERT.

2.3.1. BERT

En esta sección introducimos BERT y los detalles de su implementación. Hay dos pasos para la creación del sistema: el pre-entrenamiento y la afinación. Durante el pre-entrenamiento, el modelo se entrena con datos no etiquetados en diferentes tareas de pre-entrenamiento. Para la afinación, primero se inicializa el modelo BERT con los parámetros previamente entrenados, y todos estos parámetros son ajustados utilizando datos etiquetados para que cumplan los objetivos de la tarea final. Cada tarea final utiliza modelos diferentes para su afinación, aunque sean inicializados con los mismos parámetros de pre-entrenamiento [22]. El ejemplo en la *Figura 2.1* muestra el proceso para crear un modelo BERT en una tarea final de preguntas-respuestas.

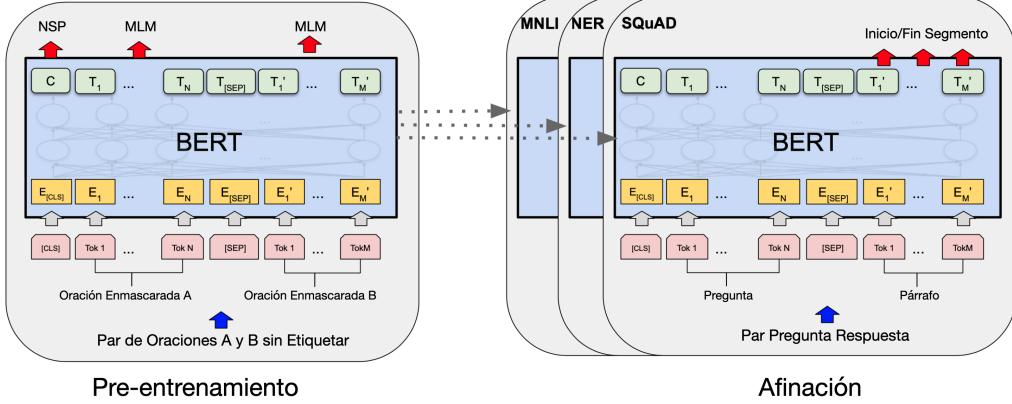


Figura 2.1: Procedimiento para el pre-entrenamiento y afinación de BERT. Con la excepción de las capas de salida, se emplea la misma arquitectura (Transformers) tanto en el pre-entrenamiento como en la afinación. Los mismos parámetros del modelo generados durante el pre-entrenamiento se utilizan para inicializar los diferentes modelos de tareas finales. Los parámetros son ajustados durante la afinación. [CLS] es un símbolo especial agregado al inicio de cada ejemplo de entrada, y [SEP] es un token separador especial, por ejemplo, para separar preguntas-respuestas u oraciones consecutivas.

Una de las características distintivas de BERT es su arquitectura unificada independientemente de la tarea objetivo. Hay pocas diferencias entre la arquitectura de pre-entrenamiento y la arquitectura final creada para la tarea. En su arquitectura, BERT utiliza un codificador multi-capa Bidireccional de Transformers basado en la implementación original de Transformers [66].

Para el pre-entrenamiento de BERT se utilizan dos tareas no supervisadas: Modelado de Lenguaje Enmascarado (MLM) y Predicción de la Próxima Oración (NSP).

Modelado de Lenguaje Enmascarado: Para entrenar una representación bidireccional del lenguaje, un porcentaje de los tokens de entrada son enmascarados al azar, para luego predecir estos tokens enmascarados. Los vectores ocultos finales correspondientes a los tokens enmascarados son entonces utilizados en el softmax de salida sobre el vocabulario, como en un Modelo de Lenguaje estándar. Generalmente, enmascarar un 15 % de las palabras brinda los mejores resultados [22].

Predicción de la Próxima Oración: El objetivo de muchas tareas, como Preguntas-Respuestas (QA⁶) y Comprensión del Lenguaje Natural (NLI⁷), se basa en la comprensión de la relación existente entre dos oraciones, relación que no es capturada directamente mediante la Modelación del Lenguaje. Con el fin de entrenar un modelo capaz de entender las relaciones entre oraciones, BERT pre-entrena el modelo para una tarea binaria de predicción de la siguiente oración que puede generarse de forma trivial para cualquier corpus monolingüe. En el momento de escoger las oraciones *A* y *B* como entrada en el pre-entrenamiento, el 50 % de las veces *B* es la siguiente oración real que sigue a *A* (etiquetando este ejemplo como IsNext), y el 50 % de las veces es una oración aleatoria del corpus (etiquetado como NotNext). Como se muestra en la *Figura 2.1*, *C* se usa para la predicción de la siguiente oración (NSP). Este entrenamiento es beneficioso tanto para el control de la calidad del modelo, como para la NLI.

Para el pre-entrenamiento del modelo BERT en inglés se utilizó el BooksCorpus (800 millones de palabras) [74] y la Wikipedia en inglés (2,500 millones de palabras). En el caso de la Wikipedia, solo se extraen los pasajes de texto y se ignoran las listas, tablas y encabezados.

Afinación de BERT: Para cada tarea, simplemente conectamos las entradas y salidas deseadas de cada ejemplo y ajustamos todos los parámetros del modelo de principio a fin para obtener los resultados esperados. Dependiendo del objetivo de la tarea, en la entrada las oraciones *A* y *B* son análogas, por ejemplo, a la pregunta (*A*) y el pasaje conteniendo la respuesta (*B*) en Preguntas-Respuestas (QA), o a la hipótesis (*A*) y la premisa (*B*) en Inferencia del Lenguaje Natural (NLI). En la salida, las representaciones de los tokens son introducidos en la capa de salida donde los token son etiquetados dependiendo del objetivo de la tarea, por ejemplo, en Preguntas-Respuestas (SQuAD en la *Figura 2.1*), se marcan los tokens donde inicia (T'_1) y termina (T'_M) la respuesta a la pregunta (*A*) dentro del pasaje (*B*). La representación [CLS] es utilizada por una capa de salida dedicada a la clasificación, por ejemplo, para determinar si la hipótesis (*A*) se cumple en la premisa (*B*) en un modelo de Inferencia del Lenguaje Natural. [CLS] también puede ser utilizado para el Análisis de Sentimiento en un texto [22].

Comparado con el pre-entrenamiento, la afinación del modelo es relativamente sencilla y poco costosa, ya que el mecanismo de auto-atención de Transformers permite que BERT sea modelado correctamente para diversos objetivos de tareas, sin importar si se trabaja con pares de texto o

⁶del inglés Question-Answering

⁷del inglés Natural Language Inference

solo un documento, siendo capaz de inferir las salidas adecuadas para cada tipo de entrada. La codificación de un par de textos concatenados con auto-atención incluye efectivamente la atención cruzada bidireccional entre dos oraciones.

2.3.2. Modelo SBERT

Como mencionamos anteriormente, con BERT solo podemos obtener de forma directa los embeddings de los tokens de la oración. Para inferir el embedding de una oración, Sentence-BERT (SBERT) agrega una operación de Pooling [24] a la capa de salida de BERT. En esta operación se pueden utilizar tres estrategias de pooling: usar la salida del token [CLS], calcular la media de todos los vectores de salida (MEAN-strategy) o calcular el máximo-a-través-del-tiempo de los vectores de salida (MAX-strategy) de los tokens de la oración. Por defecto, SBERT utiliza la MEAN-strategy.

Para la afinación del sistema SBERT, hay que informarle que pares de oraciones son similares y deberían estar cerca, y que pares son disímiles y deberían estar lejos en el espacio vectorial. La forma más sencilla de lograr esto es utilizando pares de oraciones anotadas con una puntuación que indique su similitud, por ejemplo, en una escala de 0 a 1. Luego el sistema puede ser entrenado con una arquitectura de Redes Siamesas [60, 56], utilizando regresión o tripletes para actualizar los pesos del modelo, de modo que los embeddings de las oraciones sean semánticamente significativos y se puedan comparar empleando la similitud de coseno.

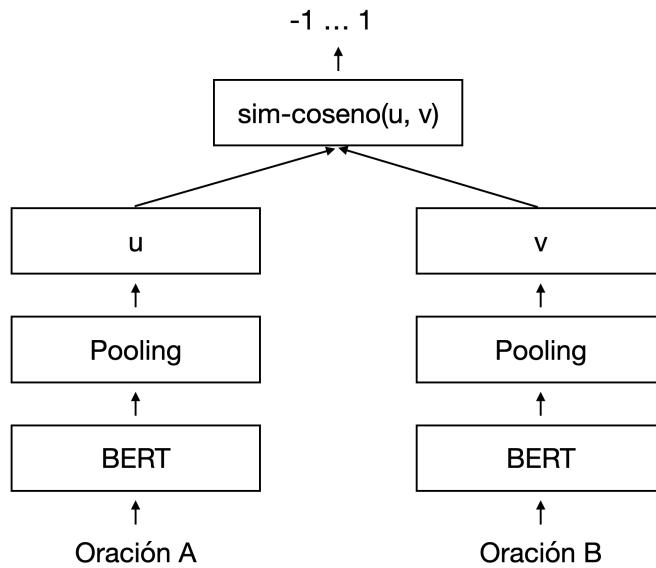


Figura 2.2: Arquitectura de SBERT para calcular la similitud entre oraciones utilizando el coseno. Es también la arquitectura utilizada para poner a punto el modelo utilizando la pérdida cuadrático-medio (mean-squared-loss) como la función objetivo.

La función objetivo utilizada depende de los datos disponibles para entrenar el modelo.

Regresión como Función Objetivo: La similitud de coseno entre los embeddings de dos oraciones se calcula, para luego usar la pérdida del error cuadrático medio como la función objetivo del modelo.

Utilizando regresión en la *Figura 2.2*, para cada par de oraciones pasamos la oración A y la oración B a través del sistema, lo que produce los embeddings u y v . La similitud de estos embeddings se calcula empleando la similitud de coseno y el resultado es comparado con la puntuación ideal para este par. Esto permite realizar la afinación del sistema SBERT y poder reconocer la similitud entre oraciones.

Triplete como Función Objetivo: Dada una oración a , una oración positiva p y una oración negativa n , la pérdida del Triplete ajusta el sistema de tal forma que la distancia entre a y p sea menor que la distancia entre a y n . Matemáticamente se minimiza la siguiente función de pérdida:

$$\max(\|s_a - s_p\| - \|s_a - s_n\| + \epsilon, 0) \quad (2.9)$$

con s_x siendo el embedding de las oraciones $a/n/p$, $\|\cdot\|$ la distancia métrica y ϵ el margen. El margen ϵ asegura que s_p esté al menos ϵ más cerca de s_a que de s_n . Como métrica se utiliza la distancia euclíadiana y estableciendo $\epsilon = 1$ por defecto.

En la práctica, para poner a punto el modelo con Triplete se escogen un par de oraciones similares como anclaje a y oración positiva b , y una oración con significado semántico diferente como la oración negativa n .

SBERT se entrena con una combinación del conjunto de datos Multi-Genre NLI [69] y el corpus de Stanford para la Comprensión del Lenguaje Natural (SNLI⁸) [11]. Multi-NLI contiene 430.000 pares de oraciones y cubre una variedad de géneros de texto hablado y escrito. SNLI es una colección de 570.000 pares de oraciones con etiquetas de vinculación, contradicción y neutral.

Sentence-BERT ofrece resultados vanguardia en la generación de embeddings para oraciones, texto e imágenes [56]. Este sistema se puede emplear para generar embeddings en más de 100 idiomas. Dado que estos embeddings se pueden comparar utilizando la similitud de coseno para encontrar oraciones con significado similar, este modelo puede ser de gran utilidad para la búsqueda por similitud textual semántica.

2.3.3. Representación de Artículos con SBERT

En modelos como BERT, que utilizan Transformers, el tiempo de ejecución y el espacio que ocupa el sistema en memoria crece cuadráticamente con la longitud de la entrada. Esto limita la longitud de los textos que pueden ser codificados con BERT. Generalmente, BERT acepta como máximo 512 tokens de palabras, que corresponde a unas 300-400 palabras⁹. Los textos con una longitud mayor a la aceptada son truncados, tomando solo las primeras X palabras del texto.

Sentence-BERT tiene un comportamiento similar, por lo que para resolver esta limitación y poder representar documentos utilizando este tipo de modelo, solo se puede utilizar el texto de los títulos y los resúmenes. El título y resumen de un documento debe contener en una corta longitud la sinopsis de los temas más importantes abordados en un artículo, libro, tesis, u otro tipo de documento, por lo que los embeddings obtenidos utilizándolos deben representar correctamente la semántica de los documentos, ubicando más cerca a los documentos con contenido similar.

⁸del inglés Stanford Natural Language Inference

⁹para un texto en inglés

2.4. SPECTER

Modelos de Lenguaje con Transformers como BERT [22] y Sentence-BERT [56] aprenden representaciones textuales poderosas, pero estos modelos tienen como objetivo la representación de texto a nivel de token y oración, y no son capaces de capturar la relación que puede existir entre documentos, lo que limita su poder de representación. El título y resumen de un artículo brindan una buena representación del contenido semántico del artículo, pero utilizar simplemente el texto de los documentos con estos modelos de lenguaje no siempre resulta en la representación más precisa para el artículo. Los objetivos de modelado de lenguaje empleados en el pre-entrenamiento de estos modelos no son los más adecuados para generar representaciones que sean útiles en tareas a nivel de documento, como la clasificación de documentos por tópicos o la recomendación de documentos similares.

En este segmento, introducimos un nuevo método para aprender representaciones vectoriales de documentos científicos. SPECTER [19], incorpora el contexto entre documentos a los modelos de lenguaje Transformer [66] (por ejemplo, BERT, SBERT, o SciBERT [5]), para aprender representaciones de documentos que sean efectivas en una amplia variedad de tareas finales, sin la necesidad de ninguna afinación con respecto al objetivo de la tarea del modelo de lenguaje pre-entrenado. Las referencias son utilizadas como una característica entre documentos que indica cuáles son los artículos más relacionados, y son utilizadas como objetivo en el pre-entrenamiento empleando la pérdida Triplete [60] como la función objetivo.

SPECTER utiliza un modelo Transformer pre-entrenado como base, sobre el que construye un nuevo modelo para incorporar las relaciones existentes entre documentos a través de las referencias. En específico, utiliza el modelo SciBERT [5], un modelo de lenguaje pre-entrenado basado en BERT [22], para mejorar la representación vectorial de textos científicos. SciBERT es pre-entrenado con un gran corpus de publicaciones científicas de múltiples dominios. Veamos primero este modelo (SciBERT), antes de continuar con los detalles del modelo SPECTER.

2.4.1. SciBERT

Como vimos anteriormente, la arquitectura del modelo BERT [22] se basa en un Transformer Bidireccional Multi-capas [66], y en lugar del objetivo tradicional de modelado de lenguaje de izquierda a derecha, BERT se entrena en dos tareas: predecir tokens enmascarados al azar y predecir si dos oraciones se suceden. SciBERT [5] sigue la misma arquitectura y pre-entrenamiento que BERT, pero en cambio, se entrena utilizando texto científico.

SciBERT está entrenado con una muestra aleatoria de 1,14 millones de artículos de Semantic Scholar [38, 2], un sistema para organizar la literatura científica publicada y facilitar su descubrimiento. Este corpus consta de un 18 % de artículos del dominio de las ciencias de la computación y un 82 % del amplio dominio biomédico. Se utiliza el texto completo de los artículos, no solo los resúmenes. La longitud promedio de un artículo es de 154 oraciones (2,769 tokens), resultando en un corpus con 3,17 mil millones de tokens como tamaño, similar a los 3,3 mil millones de tokens con los que se entrenó BERT. Las oraciones son divididas empleando Sci-SpaCy [53], un modelo optimizado para el procesamiento de texto científico, biomédico, o clínico.

SciBERT utiliza el código original de BERT para entrenar su modelo en el corpus Semantic Scholar, con la misma configuración y tamaño que BERT-Base [22].

SciBERT supera a BERT-Base en tareas relacionadas con las Ciencias de la Computación y la Biomédica, siendo su entrenamiento con un corpus científico el mayor beneficio para este sistema.

[5].

2.4.2. Modelo SPECTER

El objetivo de SPECTER [19] es aprender representaciones de artículos académicos que sean de utilidad, independientemente del objetivo para el cual se quieran utilizar. Este modelo se inspira en el reciente éxito de los modelos de lenguaje pre-entrenados con Transformers para varias tareas del Procesamiento del Lenguaje Natural (NLP).

La arquitectura de los modelos Transformers se utiliza como base para codificar el documento de entrada y, para aprender representaciones de alta calidad a nivel de documentos, SPECTER emplea las referencias como la característica que señala una relación entre documentos, usando las referencias como objetivo de aprendizaje del modelo en la pérdida Triplete [60]. Empleando este objetivo, SPECTER es pre-entrenado en un corpus grande con referencias, alentándolo a generar representaciones vectoriales más similares para los artículos que comparten una misma referencia, y representaciones más distantes para los artículos que no tengan ninguna referencia en común. A diferencia de otros modelos de lenguajes como BERT, que deben realizar una afinación del modelo antes de ser utilizados en una tarea final, los embeddings generados por SPECTER pueden ser aplicados a cualquier tarea final sin necesidad de ajustes adicionales específicos para la tarea en cuestión.

Para la inicialización SPECTER, se utiliza SciBERT como modelo base inicial, debido a su especialización en la representación de texto científico, pero cualquier modelo de lenguaje con Transformers puede ser usado en lugar de SciBERT.

SPECTER crea los embeddings a partir del título y el resumen de los artículos. Intuitivamente, estos campos deberían ser suficientes para producir buenos embeddings, ya que están escritos para proporcionar un resumen sucinto y completo del artículo. El título y el resumen son concatenados y codificados utilizando el Transformer inicializado con SciBERT, actualizado los parámetros del Transformer a medida que se entrena el modelo. El resultado final del token [CLS] en la capa de salida del Transformer se emplea como la representación vectorial final v del artículo P :

$$\mathbf{v} = \text{Transformer}(\text{input})_{[\text{CLS}]} \quad (2.10)$$

Una referencia de un documento a otro sugiere que están relacionados. Para codificar esta señal de relación en las representaciones generadas, se emplea una función de pérdida usando Tripletes. La descripción general de alto nivel del modelo se muestra en la *Figura 2.3*.

De forma más específica, cada instancia de entrenamiento es un triplete de documentos: \mathcal{P}^Q documento de consulta, \mathcal{P}^+ documento positivo y \mathcal{P}^- documento negativo. El artículo positivo \mathcal{P}^+ es un artículo que es citado por el artículo de consulta \mathcal{P}^Q , y el artículo negativo \mathcal{P}^- es un artículo que no es citado por el artículo de consulta \mathcal{P}^Q (pero que puede ser citado por \mathcal{P}^+). El modelo se entrena minimizando la siguiente función de pérdida Triplete:

$$\mathcal{L} = \max\{d(\mathcal{P}^Q, \mathcal{P}^+) - d(\mathcal{P}^Q, \mathcal{P}^-) + m, 0\} \quad (2.11)$$

donde d es una función de distancia y m es el hiperparámetro del margen de pérdida deseado (generalmente $m = 1$). Aquí, usamos la distancia normal $L2$:

$$d(\mathcal{P}^A, \mathcal{P}^B) = \|\mathbf{v}_A - \mathbf{v}_B\| \quad (2.12)$$

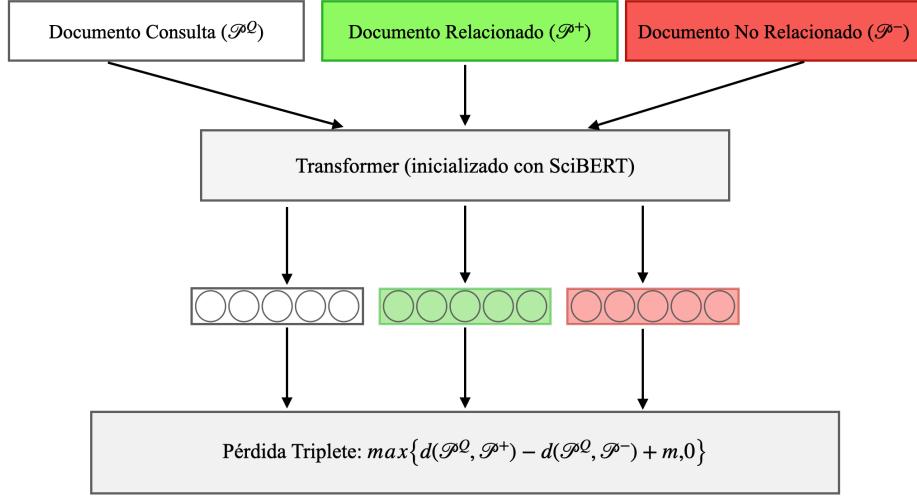


Figura 2.3: Descripción general del modelo SPECTER.

donde v_A es el vector correspondiente a la representación del artículo A, después de haber realizado Pooling [24] al encoding generado para este documento (*Fórmula 2.10*). De esta forma se entrena el Transformer del modelo para capturar las relaciones por referencias entre los documentos.

Selección de Artículos Negativos: La selección de los artículos negativos P^- es de gran importancia. Para el entrenamiento del modelo, dos conjuntos de ejemplos negativos son creados. El primer conjunto consiste de artículos seleccionados del corpus al azar. Dado un artículo de consulta, se espera que el modelo pueda distinguir intuitivamente entre artículos citados y artículos no citados seleccionados aleatoriamente. Este enfoque de la inferencia es muy eficaz para aplicaciones de recomendación de referencias basadas en el contenido [7], pero los negativos aleatorios pueden ser fáciles de distinguir para el modelo con respecto a los positivos. Para proporcionar una señal de entrenamiento más pronunciada, a los negativos aleatorios se les agrega un segundo conjunto de ejemplos negativos más desafiante, los *negativos duros*. Se denotan como *negativos duros* los artículos que no son citados por el artículo de consulta, pero sí son citados por un artículo citado por el artículo de consulta, es decir, si $\mathcal{P}^1 \xrightarrow{\text{cita}} \mathcal{P}^2$ y $\mathcal{P}^2 \xrightarrow{\text{cita}} \mathcal{P}^3$, pero $\mathcal{P}^1 \not\xrightarrow{\text{cita}} \mathcal{P}^3$ entonces \mathcal{P}^3 es un ejemplo de *negativo duro* candidato para \mathcal{P}^1 . Se espera que los negativos duros estén algo relacionados con el documento de consulta, pero por lo general menos relacionados que los documentos citados. Incluir negativos duros en el entrenamiento da como resultado embeddings mucho más precisos que si solo se incluyeran negativos aleatorios.

Para la inferencia, SPECTER requiere solo el título y el resumen del artículo de entrada. El sistema obtiene la representación vectorial, realizando Pooling [24] sobre la capa de salida del Transformer, una vez que se haya pasado el título y resumen como entrada. El modelo no necesita ninguna información sobre las referencias del documento de entrada. Esto significa que SPECTER puede producir embeddings incluso para artículos nuevos que aún no hayan sido citados, lo que puede ser muy útil en tareas donde se trabaje con artículos científicos recientes.

Para entrenar el modelo, se utiliza un subconjunto del corpus Semantic Scholar [2] que consta con alrededor de 146,000 artículos de consulta (alrededor de 26,7 millones de tokens) con sus correspondientes referencias. 32,000 artículos adicionales son usados para la validación del modelo. Por cada artículo de consulta en el corpus, se construyen hasta 5 tripletes de entrenamiento compuestos por un documento consulta, un documento positivo y un documento negativo. Los artículos positivos son muestreados a partir de las citas directas de la consulta, mientras que los artículos negativos se seleccionan al azar y de citas directas, como explicamos anteriormente. Los mejores resultados se obtienen cuando se entrena utilizando dos negativos duros (citas de citas) y 3 negativos fáciles (artículos seleccionados al azar) por cada artículo de consulta. Este proceso genera alrededor de 684,000 tripletes de entrenamiento y 145,000 tripletes de validación.

Las representaciones vectoriales generadas por SPECTER superan sustancialmente el estado del arte en una variedad de tareas a nivel de documento, incluyendo la clasificación de artículos por tópicos, la recomendación de documentos y la predicción de referencias.

Capítulo 3

Modelos de Tópicos

En el Procesamiento del Lenguaje Natural (NLP), un *modelo de tópicos* es un modelo de tipo *estadístico o distribuido* para descubrir los *tópicos* (o temas) abstractos que ocurren en una colección de documentos. El modelado de tópicos es una herramienta de minería de texto utilizada con frecuencia para descubrir *estructuras semánticas ocultas* en el cuerpo de un texto. Intuitivamente, dado que un documento trata sobre un tema en particular, ciertas palabras deben aparecer con mayor o menor frecuencia dependiendo del tópico del documento: “perro” y “hueso” aparecerán con mayor frecuencia en documentos sobre perros, “gato” y “miau” aparecerán en documentos sobre gatos, y “la”, “para” y “es” aparecerán aproximadamente por igual en ambos documentos.

Un documento generalmente se refiere a múltiples temas en proporciones diferentes, por lo que un documento donde se habla un 10 % sobre gatos y un 90 % sobre perros, probablemente tendrá 9 veces más palabras sobre perros que palabras sobre gatos. Los tópicos producidos por el modelo de tópicos son el grupo de palabras por las que se puede identificar el tema del que se habla en un tópico específico. Un modelo de tópicos captura esta conexión entre palabras, lo que permite examinar un conjunto de documentos y descubrir los tópicos y el balance de cada tópico para cada documento.

3.1. LDA: Modelos de Tópicos Probabilísticos

Latent Dirichlet Allocation (LDA) [9], es el *modelo de tópicos* más simple. LDA es un modelo probabilístico, donde los datos se tratan como si surgieran de un proceso generativo que incluye *variables ocultas*. El modelo LDA fue creado utilizando la intuición de que cada documento exhibe múltiples *tópicos*. Este modelo asume que para toda la colección existe un número específico de *tópicos*, y que cada tópico es una distribución sobre palabras.

Por ejemplo, supongamos que tenemos el siguiente grupo de documentos:

Doc 1: Los mangos y los plátanos son muy sabrosos.

Doc 2: Me tomé un batido de mango y guayaba en el desayuno.

Doc 3: Los gatos y perros son las mascotas más populares.

Doc 4: Mi hermana rescató un gato ayer.

Doc 5: Mira el hamster masticando un pedazo de plátano.

Podemos emplear LDA para encontrar los 2 tópicos (A y B) dentro de la colección de documentos. Como resultado el modelo puede producir los siguientes tópicos:

Tópico A: 30 % mango, 20 % plátano, 15 % guayaba, 10 % desayuno, ...

Tópico B: 20 % gatos, 15 % perros, 10 % hamster, 10 % mascotas, ...

Donde los tópicos en cada documento son los siguientes:

Doc 1: 100 % Tópico A

Doc 2: 100 % Tópico A

Doc 3: 100 % Tópico B

Doc 4: 100 % Tópico B

Doc 5: 55 % Tópico A, 45 % Tópico B

Observando ambos tópicos, podemos interpretar que el tópico A es sobre comida y que el tópico B es sobre mascotas. Por supuesto, la pregunta es cómo LDA descubre los tópicos. El modelo LDA puede ser descrito por su proceso generativo, el proceso aleatorio imaginario por el cual el modelo asume que surgieron los documentos.

Formalmente, un tópico (o tema) se define como una distribución sobre un vocabulario fijo. Por ejemplo, el tópico de A tiene una alta probabilidad para palabras sobre comida y el tópico B tiene una alta probabilidad para palabras sobre mascotas. Se asume que los tópicos son especificados antes de generar cualquier dato. Para cada documento en la colección, las palabras son generadas en un proceso de dos etapas.

1. Elija aleatoriamente una distribución sobre los tópicos.
2. Por cada Palabra en el documento:
 - a) Elija aleatoriamente uno de los tópicos utilizando la distribución creada en el paso #1.
 - b) Elija aleatoriamente una palabra del vocabulario utilizando la distribución sobre el vocabulario del tópico seleccionado.

Este modelo estadístico refleja la intuición de que los documentos exhiben múltiples tópicos. Cada documento exhibe los tópicos con diferentes proporciones (paso #1); cada palabra en cada documento se extrae de uno de los tópicos (paso #2b), donde el tópico seleccionado se elige de la distribución sobre tópicos de cada documento (paso #2a). Una de las características distintivas de LDA es que todos los documentos en la colección comparten el mismo conjunto de tópicos, pero cada documento exhibe estos tópicos con diferente proporción.

El objetivo principal del modelado de tópicos es descubrir automáticamente los temas de una colección de documentos. Los documentos en sí son observados, mientras que la estructura de los tópicos (los tópicos, las distribuciones de tópicos por documento y las distribuciones de palabras por tópicos), es una estructura oculta [8]. El problema computacional central para el modelado de tópicos es utilizar los documentos observados para inferir la *estructura de tópicos oculta*. El problema se puede ver como “revertir” el proceso generativo: ¿cuál es la estructura oculta que probablemente generó la colección observada?

Es importante resaltar, que los algoritmos no tienen ninguna información sobre los posibles tópicos, y los artículos no están etiquetados con tópicos ni palabras clave. Las distribuciones de

tópicos interpretables surgen al calcular la estructura oculta que probablemente generó la colección de documentos observada.

La utilidad de los modelos de tópicos viene de la propiedad de que la *estructura oculta inferior* se asemeja a la *estructura temática* de la colección. Esta estructura oculta interpretable anota cada documento de la colección, una tarea laboriosa de realizar a mano, y estas anotaciones se pueden usar para ayudar en tareas como la recuperación de información, la clasificación y la exploración del corpus. De esta forma, el modelado de tópicos proporciona una solución algorítmica para administrar, organizar y anotar grandes archivos de textos.

3.2. Top2Vec: Representación Distribuida de Tópicos

Como ya hemos visto, un tópico es el tema, materia o sujeto de un texto; es de lo que se discute o se habla en un documento. Los tópicos a menudo son considerados valores discretos, como la política, la ciencia y la religión. Sin embargo, este no es el caso, ya que cualquiera de estos *tópicos* se puede subdividir en muchos otros *sub-tópicos*. Además, un tópico como la política puede coincidir en parte con otros tópicos, como el tópico de la salud, ya que ambos comparten el sub-tópico de la atención médica. Estos tópicos, sus combinaciones y variaciones pueden ser descritos mediante un conjunto único de *palabras ponderadas*. Por esto, se puede asumir que los tópicos son continuos, ya que hay infinitas combinaciones de *palabras ponderadas* que se pueden usar para representar un tópico [3]. Además, se puede asumir que cada documento tiene un tópico propio con un valor en este continuo. De esta forma, el tópico de un documento se puede considerar como el conjunto de palabras ponderadas que son más informativas de su tópico único, que puede ser una combinación de los temas coloquiales discretos.

LDA discretiza el espacio de tópicos continuo con t tópicos, y modela los documentos como una mezcla de estos t tópicos, asumiendo un número de tópicos t a conocer. En este modelo, la discretización de los tópicos es necesaria para poder modelar la relación entre documentos y palabras. Esta es una de las mayores debilidades del modelo LDA, ya que rara vez se conoce el número de tópicos o la forma de estimarlos, especialmente para conjuntos de datos muy grandes o desconocidos [12, 63].

Cada tópico generado por LDA es una distribución de probabilidades de palabras. Debido a esto, las palabras de mayor probabilidad dentro de un tópico suelen ser palabras como “la”, “para”, “es” (“the”, “to”, “is” en inglés), y otras palabras comunes del idioma. Estas palabras comunes, también llamadas *palabras vacías*¹, a menudo necesitan ser filtradas para que los tópicos puedan ser interpretables, y poder obtener palabras informativas sobre el tópico. Encontrar el conjunto de *palabras vacías* que deben eliminarse no es un problema trivial, ya que es específico tanto para el corpus como el lenguaje con el que se esté trabajando [71]; un modelo de tópicos entrenado con textos sobre perros probablemente tratará “perro” como una *palabra vacía*, ya que para estos documentos no es una palabra muy informativa.

El objetivo de los modelos generativos probabilísticos como LDA es encontrar tópicos que puedan emplearse para recrear las distribuciones de palabras del documento original con errores mínimos. Sin embargo, es común que los textos contengan una proporción grande de palabras que son poco informativas, que pueden no considerarse temáticas. Estos modelos no diferencian entre palabras informativas y no informativas, ya que su objetivo es simplemente recrear las distribuciones de palabras del documento. Por lo tanto, las palabras de alta probabilidad en los tópicos encontrados

¹en inglés se les llama stop-words

no necesariamente se corresponden con las palabras que un usuario identificaría intuitivamente como un tópico.

También, otra debilidad del modelo LDA es que utiliza Bolsa de Palabras (BOW²) para la representación de documentos, representación que ignora la semántica de palabras. En la representación BOW, las palabras “Canadá” y “canadiense” son tratadas como palabras diferentes, a pesar de su similitud semántica. Las técnicas de *stemming* y *lematización* tienen como objetivo solucionar estos problemas, pero a menudo hacen que los tópicos sean más difíciles de comprender. Además, el *stemming* y la *lematización* no reconocen la similitud entre palabras como “grande” y “enorme”, ya que no comparten una palabra raíz.

El modelo de tópicos Top2Vec trata de solucionar los problemas presentes en los modelos de tópicos probabilísticos utilizando una representación distribuida de tópicos.

3.2.1. Representación Distribuida de Tópicos

Un *espacio semántico* es una representación espacial en la que la distancia representa una asociación semántica [26]. Mucha atención se le ha prestado a los embeddings semánticos de palabras. Específicamente, a los vectores distribuidos de palabras generados por modelos como Word2Vec o BERT, que han demostrado su capacidad de capturar regularidades sintácticas y semánticas del lenguaje [49, 52, 56].

Otros modelos como Doc2Vec y Sentence-BERT, son capaces de aprender representaciones vectoriales de documentos y palabras con *embeddings conjuntos* en el mismo espacio vectorial. Estos vectores se aprenden de tal manera que los embeddings de documentos quedan cerca de los embeddings de palabras que sean semánticamente similares. Esta propiedad se puede emplear en la recuperación de información, ya que los vectores de palabras se pueden usar para consultar documentos similares. También se puede utilizar para encontrar qué palabras son las más similares o más representativas de un documento. Además, el vector de párrafo o documento actúa como una memoria del tópico del documento [33]. Por lo tanto, los vectores de palabras más similares al vector de un documento son probablemente los más representativos del tópico del documento. Estos *embeddings conjuntos* de documentos y palabras constituyen un *espacio semántico* (o *embedding semántico*), ya que la distancia en el espacio vectorial mide la similitud semántica entre los documentos y las palabras.

A diferencia de los métodos tradicionales de modelado de tópicos con BOW, el *embedding semántico* tiene la ventaja de aprender la asociación semántica entre palabras y documentos. Por esto, se puede considerar que el *espacio semántico* en sí mismo es una representación continua de tópicos [3], en la que cada punto es un tópico diferente que es mejor descrito por sus palabras más cercanas. En el *espacio semántico* de documentos y palabras de *embeddings conjuntos*, un área densa de documentos puede interpretarse como muchos documentos que tienen un tópico similar. Utilizando esta intuición se crea Top2Vec [3], un vector de tópicos distribuido que se calcula a partir de áreas densas de vectores de documentos, y donde el número de áreas densas de documentos encontradas en el *espacio semántico* se asumen como el número de tópicos destacados en la colección de documentos. Los vectores de tópicos se calculan como los centroides de cada área densa de vectores de documentos. Un área densa es un área de documentos muy similares, y el centroide, o vector del tópico, puede considerarse como el documento promedio más representativo de esa área. El *embedding semántico* es aprovechado para encontrar las palabras que son más representativas de cada tópico, buscando los vectores de palabras más cercanos a cada vector de tópico.

²del inglés Bag of Words

El modelo **Top2Vec** produce vectores de palabras, documentos y tópicos con embeddings conjuntos, de modo que la distancia entre ellos represente una similitud semántica. La eliminación de *palabras vacías*, la *lematización*, el *stemming*, y un conocimiento a priori de la cantidad de tópicos no son necesarios para que **Top2Vec** aprenda buenos vectores de tópicos. Esto le da a **Top2Vec** una gran ventaja sobre los métodos tradicionales. El vector de tema se puede emplear para encontrar documentos similares y las palabras se pueden usar para encontrar temas similares. La misma álgebra vectorial demostrada con **Word2Vec** [49, 51] se puede utilizar entre los vectores de palabras, documentos y tópicos. La representación vectorial de los tópicos se pueden emplear para calcular los tamaños de los tópicos en función del vector de tema más cercano a cada vector de documento. Además la reducción del número de tópicos se puede realizar con los vectores de tópicos para agrupar jerárquicamente tópicos similares y reducir la cantidad de tópicos descubiertos.

La mayor diferencia entre **Top2Vec** y los *modelos probabilísticos generativos* es cómo cada uno modela los tópicos. LDA modela los tópicos como distribuciones de palabras, que se emplean para recrear las distribuciones de palabras del documento original con error mínimo, requiriendo que las palabras no informativas de alta presencia en el documento tengan altas probabilidades en los tópicos. Un vector de tópicos de **Top2Vec**, por el contrario, representa un tema destacado compartido entre documentos dentro del embedding semántico, donde las palabras más cercanas a un vector de tema describen mejor el tema y los documentos que lo rodean. Esto se debe al aprendizaje conjunto de los embeddings de documentos y palabras, que permite predecir qué palabras son más descriptivas para un documento, y causa que los vectores de tópicos estén lo más cerca posible de sus palabras más informativas [3].

3.2.2. Creación del Espacio Semántico

Para poder extraer tópicos con **Top2Vec**, se requieren embeddings conjuntos de documentos y palabras con ciertas propiedades. Específicamente, embeddings donde la distancia entre los vectores de documentos y los vectores de palabras representen una asociación semántica. Los documentos semánticamente similares deben estar agrupados juntos en el espacio vectorial, y los documentos diferentes deben estar más separados entre sí. Además, las palabras deben estar cerca de los documentos a los que describan mejor. Cualquier modelo de representación de documentos y palabras con estas propiedades puede ser utilizado con **Top2Vec**. Esta representación espacial de palabras y documentos se denomina *espacio semántico* [26]. Un *espacio semántico* con las propiedades descritas anteriormente es una *representación continua de tópicos* [3]. La Figura 3.1 muestra un ejemplo de un *espacio semántico*.

Con este modelo de tópicos se pueden emplear modelos de lenguaje pre-entrenados como BERT [22], Sentence-Bert [56], o Universal Sentence Encoder (UNE) [16]. Para aprender vectores de documentos y palabras para un corpus específico se puede usar **Doc2Vec** [33, 55].

3.2.3. Encontrar el Número de Tópicos

Los embeddings semánticos tienen la ventaja de aprender una representación continua de tópicos. En el espacio vectorial conjunto de documentos y palabras, con las propiedades descritas en la **Sección 3.2.1**, los documentos y las palabras se representan como posiciones en el espacio semántico. En este espacio, cada vector de documento puede verse como una representación del tema del documento [33].

En el espacio semántico, un área densa de documentos puede interpretarse como un área de

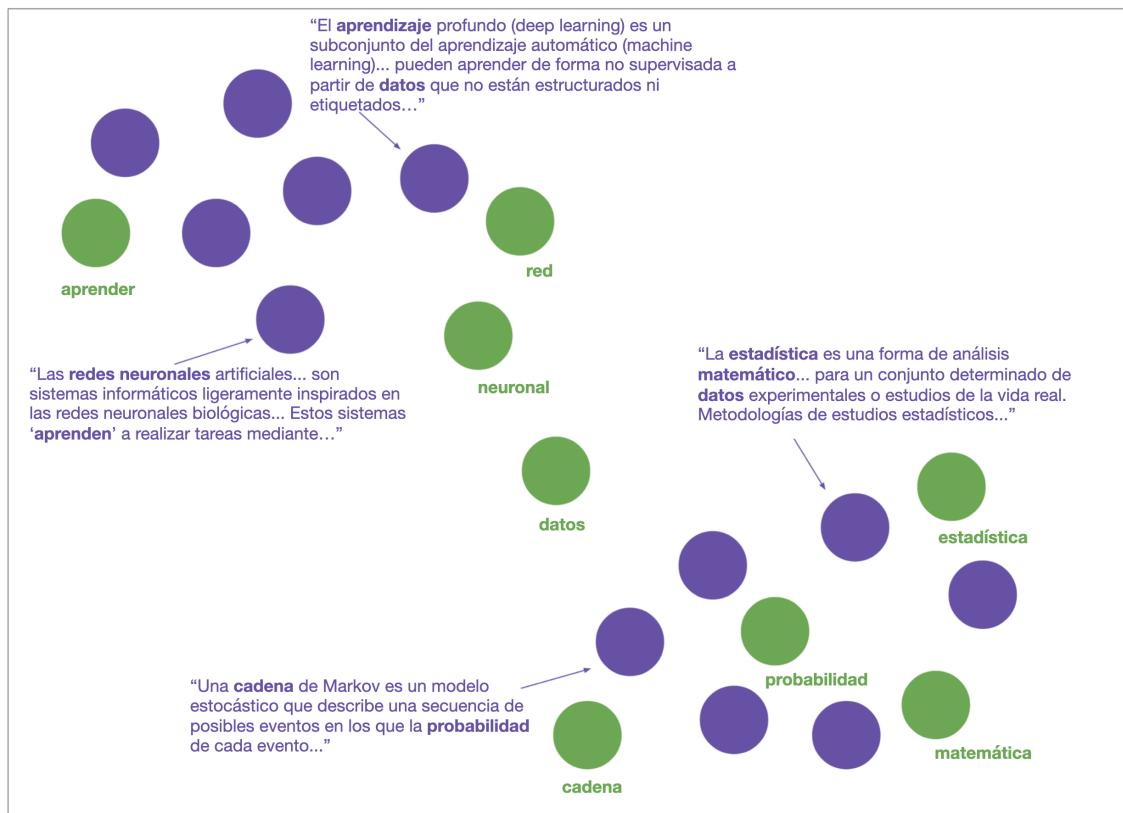


Figura 3.1: Ejemplo de un *espacio semántico*. Los puntos morados son documentos y los puntos verdes son palabras. Las palabras están más cerca de los documentos que mejor representan y los documentos similares están agrupados.

documentos muy similares. Esta área densa de documentos es indicativa de un tópico subyacente que es común para los documentos. Dado que los vectores de documentos representan los tópicos de los documentos, se puede calcular el centroide o promedio de estos vectores. Este centroide sería el vector de tópico más representativo del área densa de documentos a partir de los cuales se calculó, y las palabras más cercanas al vector son las palabras que mejor lo describen semánticamente. Como ya mencionamos, la suposición principal detrás de Top2Vec es que la cantidad de áreas densas de vectores de documentos es igual a la cantidad de tópicos destacados. Esta es una forma natural de discretizar los tópicos, ya que se encuentra un tópico para cada grupo de documentos que comparten un tópico destacado.

Para encontrar las áreas densas de documentos en el espacio semántico, se utiliza el *agrupamiento basado en la densidad* (*density based clustering*) con los vectores de los documentos, específicamente el algoritmo "Hierarchical Density-Based Spatial Clustering of Applications with Noise" (HDBSCAN) [15, 44, 45]. Sin embargo, las altas dimensiones de los vectores de documentos provocan dos problemas grandes. En la alta dimensionalidad del espacio semántico, generalmente de 300 dimensiones o más, los vectores de los documentos están muy espaciados. El esparcimiento de los vectores

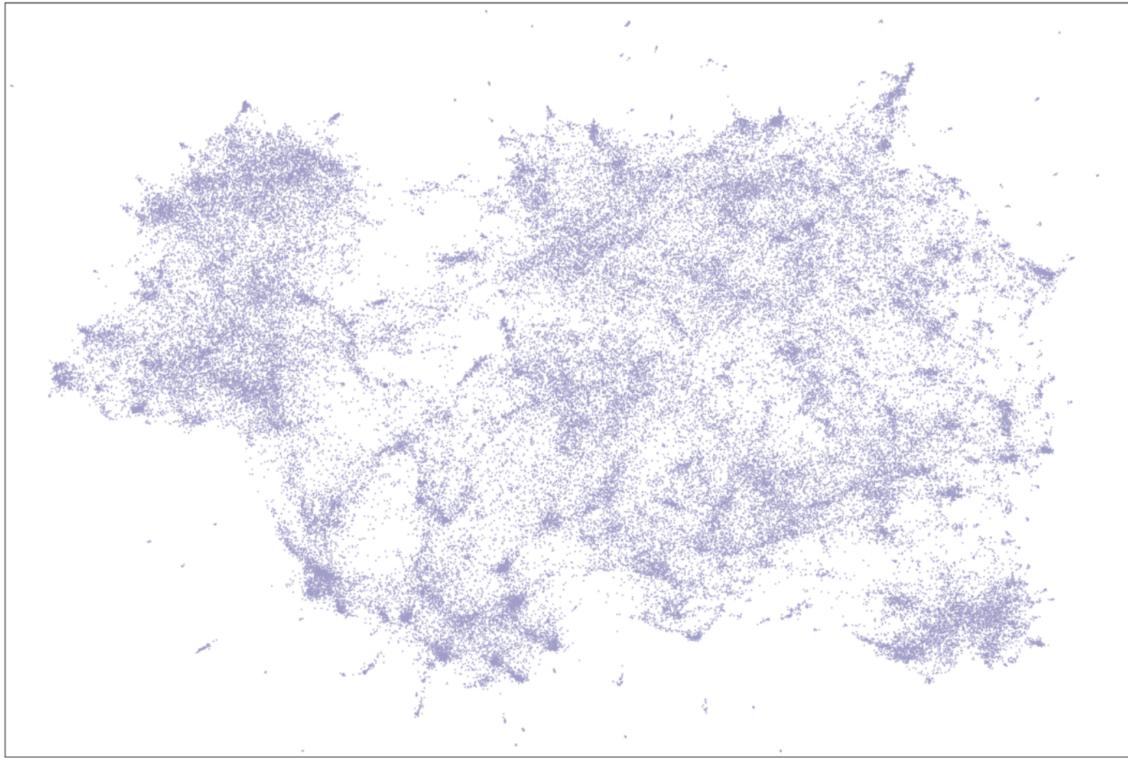


Figura 3.2: Vectores de documentos con 300 dimensiones del conjunto de datos *20 News Group*, a los cuales se les redujo las dimensiones de sus embeddings a 2 empleando UMAP.

de documentos en este espacio, provoca que sea muy difícil encontrar grupos densos y realizar este tipo de búsqueda tiene un alto costo computacional [43]. Para aliviar estos dos problemas, se realiza una reducción de dimensiones a los vectores de documentos empleando el algoritmo “Uniform Manifold Approximation and Projection for Dimension Reduction” (UMAP) [46, 47]. En el espacio de dimensiones reducidas, se puede usar HDBSCAN para encontrar los grupos densos de documentos.

Embedding de Documentos con Baja Dimensionalidad: La reducción de dimensiones permite encontrar los grupos densos de documentos en una forma más eficiente y precisa en el espacio vectorial reducido. UMAP es una técnica de aprendizaje múltiple para la reducción de dimensiones con fuerte fundamento teórico [46, 47]. Otro algoritmo muy usado en la reducción dimensional es “T-distributed Stochastic Neighbor Embedding” (t-SNE), pero t-SNE no conserva la estructura global tan bien como UMAP y no tiene una buena adaptación para grandes conjuntos de datos. Por este motivo, UMAP es la técnica elegida para la reducción de dimensiones en *Top2Vec*, ya que conserva la estructura local y global, y se puede escalar a conjuntos de datos muy grandes. La *Figura 3.2* muestra vectores de documentos reducidos empleando UMAP; se puede observar que los embeddings conservan en gran medida la estructura global y local.

UMAP tiene varios hiperparámetros que determinan cómo se realiza la reducción de dimensión. Uno de los parámetros más importantes es el *número de vecinos más cercanos*, que controla el equi-

librio entre la conservación de la estructura global frente a la estructura local para los embeddings de bajas dimensiones. Valores altos de este parámetro ponen más énfasis en la preservación de la estructura global sobre la estructura local. Dado que el objetivo es encontrar áreas densas de documentos que están cerca en el espacio dimensional alto, la estructura local es más importante en esta aplicación. Establecer 15 como el *número de vecinos más cercanos* brinda los mejores resultados, ya que este valor da más énfasis en la estructura local. Otro parámetro de importancia es la métrica de distancia, que se emplea para medir la distancia entre puntos en el espacio dimensional alto. La métrica de distancia utilizada más a menudo con los vectores de documentos es la similitud de coseno [49, 51], porque mide la similitud de los documentos independientemente de su tamaño. Por último, se debe elegir la dimensión de los embeddings finales, Top2Vec obtiene los mejores resultados empleando vectores de 5 dimensiones en la tarea posterior de agrupamiento basado en densidad [3].

Encontrar Grupos Densos de Documentos: El objetivo del *agrupamiento basado en densidad* es encontrar áreas de documentos muy similares en el espacio semántico, que indiquen un tema subyacente. Los agrupamientos se realizan sobre los vectores de documentos reducidos con UMAP. El desafío está en que los vectores de los documentos tienen una densidad variable dependiendo de su ubicación en el espacio semántico. Además, existen áreas espaciadas donde hay mucha diferencia entre un documento y otro. Estas áreas pueden verse como ruido, ya que no tienen un tópico subyacente destacado. Para superar estos desafíos y encontrar las áreas densas de documentos se utiliza HDBSCAN, ya que fue diseñado para manejar tanto el ruido como los clústeres de densidad variable [44]. HDBSCAN asigna una etiqueta a cada grupo denso de vectores de documentos y asigna una etiqueta de ruido a todos los vectores de documentos que no están en un grupo denso. Las áreas densas de los vectores de documentos identificados se utilizarán para calcular los vectores de los tópicos. Los documentos clasificados como ruido pueden verse como no descriptivos de un tópico destacado. La *Figura 3.3* muestra un ejemplo de áreas densas de documentos identificados por HDBSCAN.

El hiperparámetro principal que debe elegirse para HDBSCAN es el *tamaño mínimo de clúster*; este parámetro está en el centro de cómo el algoritmo encuentra grupos de densidad variable [44], y representa el tamaño mínimo que debe tener un grupo de vectores de documentos para que el algoritmo pueda considerarlo un clúster. Con 15 como el *tamaño mínimo de clúster* se obtienen los mejores resultados, ya que los valores más grandes tienen una mayor probabilidad de fusionar clústeres de documentos no relacionados [3].

3.2.4. Calcular los Vectores de Tópicos

Los grupos densos de documentos y los documentos clasificados como ruido por HDBSCAN en el espacio de vectores UMAP-reducido corresponden a ubicaciones en el espacio de embedding semántico original. La utilización de UMAP y HDBSCAN puede verse como un proceso para etiquetar cada documento en el espacio de embedding semántico con una etiqueta de ruido o una etiqueta con el grupo denso al que pertenece el documento.

Dadas las etiquetas para cada grupo de documentos densos en el espacio de embedding semántico, se pueden calcular los vectores de tópicos. Hay varias formas de calcular los vectores de tópicos a partir de los vectores de documento. El método más simple es calcular el centroide, es decir, la media aritmética de todos los vectores de documentos en el mismo grupo denso. Existen otras opciones para calcular los vectores, como la media geométrica, pero estas técnicas resultan en vectores de tópicos muy similares y con vectores de palabras vecinos casi idénticos a los obtenidos

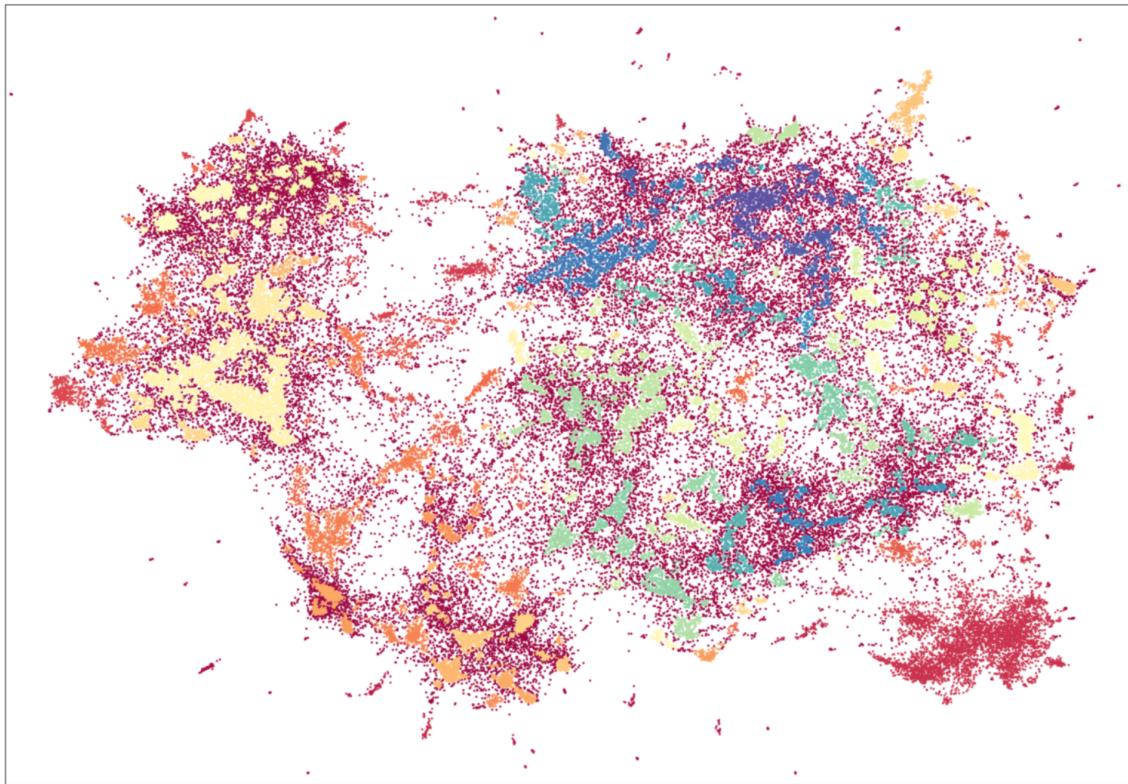


Figura 3.3: Vectores de documentos UMAP-reducidos del conjunto de datos *20 News Group*. Cada área coloreada de puntos es un área densa de documentos identificados por HDBSCAN, los puntos rojos son documentos que HDBSCAN ha etiquetado como ruido.

empleando el centroide. Esto se debe principalmente al esparcimiento de los vectores en los espacios de alta dimensión. Por lo tanto, se utiliza el método del centroide por ser el más simple de calcular [3]. La *Figura 3.4* muestra un ejemplo visual de un vector de tópico calculado a partir de un área densa de documentos.

El centroide se calcula para cada conjunto de vectores de documentos que pertenecen a un grupo denso, generando un vector de tópico para cada conjunto. El *número de áreas densas* encontradas es el *número de tópicos destacados* identificados en el corpus.

Palabras de los Tópicos: En el espacio semántico, cada punto representa un tópico que es mejor descrito semánticamente por los vectores de palabras más cercanos. Por lo tanto, los vectores de palabras que están más cerca de un vector de tópico son los que mejor lo representan semánticamente. La distancia de cada vector de palabra al vector de tópico indica cuanta similitud semántica existe entre la palabra y el tópico. Las palabras más cercanas al vector de tópico pueden ser vistas como las palabras más similares a los documentos pertenecientes al área densa con la que se creó el tópico. Por esto, estas palabras se pueden utilizar para resumir el tema común entre los documentos del área densa. La *Figura 3.5* muestra un ejemplo de un vector de tópico y las palabras más cercanas.

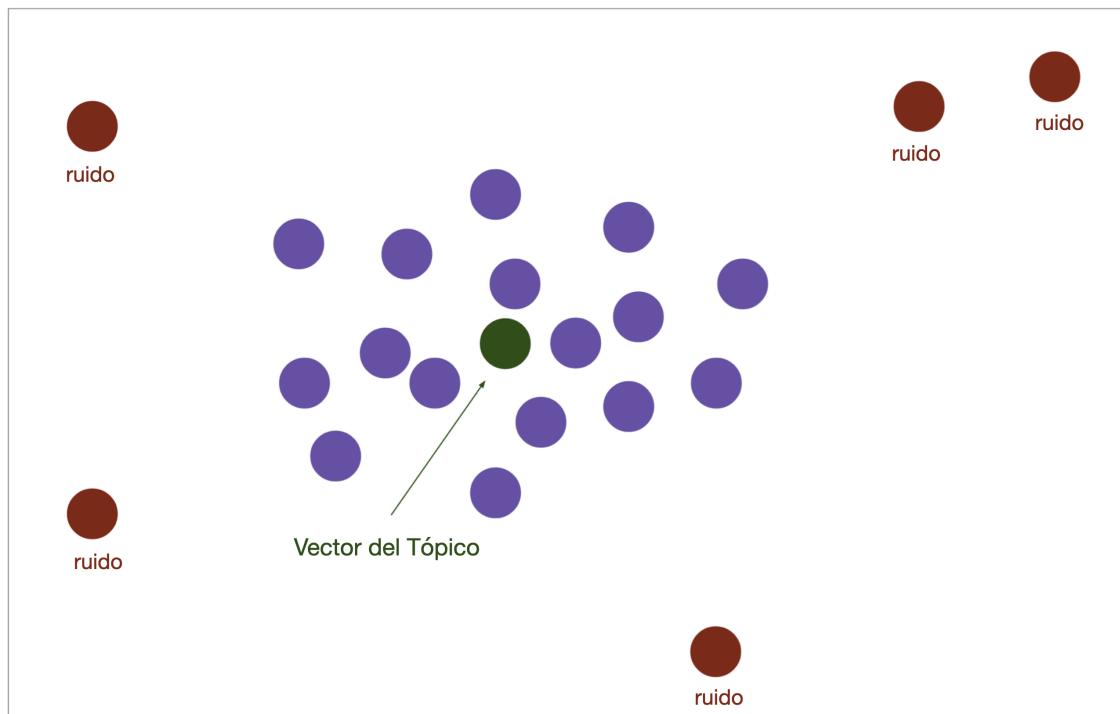


Figura 3.4: El vector de tópico es el centroide del área densa de documentos identificada por HDBSCAN, que son los puntos morados. Los documentos identificados por HDBSCAN como ruido no se utilizan para calcular el centroide.

Las *palabras vacías* aparecen en la mayoría de los documentos y, como tales, generalmente se encuentran en una región del espacio semántico que está igualmente distante de todos los documentos. Como resultado, las palabras más cercanas a un vector temático en muy raras ocasiones son *palabras vacías*. Por lo tanto no hay necesidad de escanear los documentos para eliminar las *palabras vacías*.

Tamaño de los Tópicos: Los vectores de tópicos y documentos permiten el cálculo del tamaño de los tópicos. Los vectores de tópicos se pueden utilizar para dividir los vectores de documentos de tal manera que cada vector de documento pertenezca a su vector de tópicos más cercano. Esto asocia cada documento a un tópico, el tópico más semánticamente similar al documento. El tamaño de cada tópico se mide por el número de documentos que le pertenecen.

Reducción Jerárquica de Tópicos Una ventaja de los vectores de tópicos y la *representación continua de tópicos* en el *espacio semántico* es que el número de tópicos encontrados por Top2Vec se puede reducir jerárquicamente a cualquier número de tópicos menor que el número encontrado inicialmente. Esto se hace uniendo iterativamente el tópico más pequeño a su tópico más similar semánticamente hasta alcanzar el número de tópicos deseados. Para esto, se toma la media aritmética ponderada del vector de tópico más pequeño y su vector de tópico más cercano, cada uno

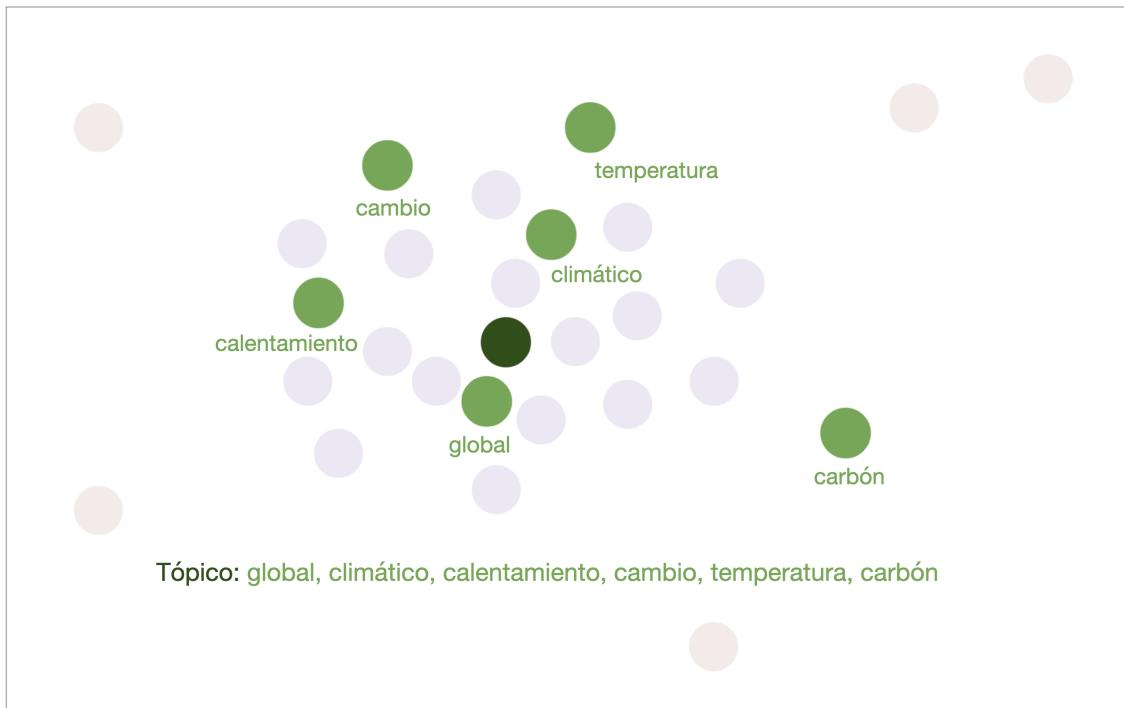


Figura 3.5: Las palabras del tópico son los vectores de palabras más cercanos al vector del tópico.

siendo ponderado por su tamaño de tópico. Después de cada combinación, los tamaños de tópicos se vuelven a calcular para cada tópico. Esta reducción jerárquica de tópicos tiene la ventaja de encontrar los temas más representativos del corpus, ya que tiende a proteger los tópicos de mayor tamaño [3].

Capítulo 4

Desarrollo del SRI-TOP: Sistema de Recuperación de Información basado en Tópicos

En los últimos años se han dado grandes pasos en la representación distribuida de palabras y documentos, representaciones que son capaces de capturar las relaciones semánticas y sintácticas entre palabras y documentos. En este trabajo de tesis construimos un Sistema de Recuperación de Información basado en Tópicos (SRI-TOP) utilizando dos de los Modelos de Lenguaje más recientes Sentence-BERT (SBERT) y SPECTER, modelos con muy buenos resultados en la representación, de texto y texto científico respectivamente. Con estos Modelos de Lenguaje construimos un Modelo de Tópicos con Representaciones Distribuidas de Tópicos (*Top2Vec*, ver **Sección 3.2**) para mejorar y facilitar las tareas de organización, exploración y recuperación de información dentro del SRI-TOP.

Para la implementación del SRI-TOP se desarrollaron dos Modelos de Tópicos. Un modelo emplea SBERT para la creación de los embeddings de las palabras, documentos y tópicos. El otro modelo emplea ambos Modelos de Lenguaje, SPECTER para la creación de los embeddings de los documentos y tópicos, y SBERT para la creación de los embeddings de las palabras y el vocabulario de los tópicos. Para comprobar y evaluar el rendimiento de los Modelos de Tópicos construidos, primero empleamos la *ganancia de información con tópicos* para comparar el rendimiento de nuestros modelos con los modelos de tópicos LDA y Top2Vec con Doc2Vec. Luego empleamos la *Homogeneidad de Tópicos* para comparar de forma más precisa el rendimiento de los modelos desarrollados, uno con respecto al otro.

Al final de este capítulo introducimos la aplicación visual desarrollada para la interacción de los usuarios con el SRI-TOP. Una interfaz de usuario que permite la realización de consultas sobre los documentos en el corpus utilizando tópicos, ver la cantidad de tópicos encontrados en la colección de documentos, disminuir el tamaño del modelo (número de tópicos) para el procesamiento de las consultas, y poder examinar el contenido de los documentos en el corpus.

4.1. Procesamiento del Corpus

Para la creación del SRI-TOP y los Modelos de Tópicos se utilizó el conjunto de datos 2020-05-31 de CORD-19 publicado el día 31 de mayo del 2020 [68]. Esta versión de la colección contiene alrededor de 139 mil documentos, con artículos publicados desde el año 1870 hasta la fecha de publicación del conjunto de datos.

En la creación de los modelos de tópicos se utilizan los modelos de texto SBERT y SPECTER. Ambos modelos se basan en BERT (ver **Sección 2.3.1**), por lo que como se explica en la **Sección 2.3.3** solo podemos utilizar el título y resumen de los documentos para generar sus embeddings. Por este motivo, solo utilizamos esta parte de los documentos durante el proceso de indexación.

El título y resumen de los documentos ya vienen pre-procesados con los metadatos del corpus en el archivo `metadata.csv`. Por lo que solo se necesita cargar este archivo para guardar los títulos y resúmenes de cada uno de los artículos. Para una correcta indexación y organización de los documentos, necesitamos tanto el título como el resumen de los artículos, por lo que todos los documentos a los que les falte al menos uno de estos campos deben ser descartados. De los 139 mil documentos en la colección, 30 mil no contienen información sobre su título o resumen, reduciendo el tamaño del corpus utilizable a 109 mil. El texto completo de los documentos también se extrae de la colección y se guarda, en caso de que un usuario quiera acceder a este durante el proceso de recuperación de información. De los 109 mil documentos restantes en el corpus, alrededor del 50 % tiene su texto completo disponible.

En el corpus, la mayoría de los documentos están en inglés, pero existe un grupo significativo de documentos en otros idiomas. Estos documentos provocan un esparcimiento innecesario en el espacio vectorial de los documentos, ya que tanto SPECTER como SBERT [19, 3] interpretan su texto como semánticamente distante del resto de los artículos en el corpus. Un esparcimiento extremo en el espacio vectorial puede afectar el rendimiento del algoritmo HDBSCAN para encontrar los clústeres de documentos adecuados, y esto puede disminuir la calidad de los tópicos creados por los Modelos de Tópicos y el rendimiento posterior de los modelos en el Sistema de Recuperación de Información (SRI) [45, 3]. Por este motivo, empleando **FastText** detectamos los documentos en otros idiomas para eliminarlos [10]. De los 109 mil documentos restantes en el corpus, alrededor de 5 mil estaban en otros idiomas y fueron descartados, por lo que nos quedamos con un total de 105 mil documentos para la construcción de los modelos y el sistema.

Dentro de los datos más importantes utilizados durante la creación del Modelo de Tópicos **Top2Vec** está la representación vectorial de los documentos. Una de las ventajas que ofrece el conjunto de datos CORD-19 es que los embeddings SPECTER de los artículos vienen junto con los metadatos de los documentos, lo que facilita el proceso de creación de los modelos, ya teniendo acceso a esta información solo necesitamos crear las representaciones vectoriales SBERT. Los embeddings SPECTER se guardan en un diccionario, utilizando los identificadores especiales de los documentos (`cord_uid`) dentro de la colección como llaves de este diccionario de vectores [68].

4.1.1. Corpus de Artículos Académicos CORD-19

El Conjunto de Datos de Investigación sobre la COVID-19 (CORD-19) es una colección de artículos científicos relacionados con la COVID-19, el SARS, el MERS y la investigación del coronavirus. CORD-19 está diseñado para facilitar el desarrollo de Sistemas de Recuperación de Información, la minería de texto y la investigación del Procesamiento del Lenguaje Natural. Desde su lanzamiento

en marzo del 2020 con alrededor de 40 mil artículos, este corpus se ha ido actualizando hasta incluir más de 1 millón de artículos en su versión final de junio del 2022, incluyendo el texto completo de casi 370,000 artículos. Este corpus de artículos académicos es curado y mantenido por el equipo de Semantic Scholar en el Instituto Allen de Inteligencia Artificial [68].

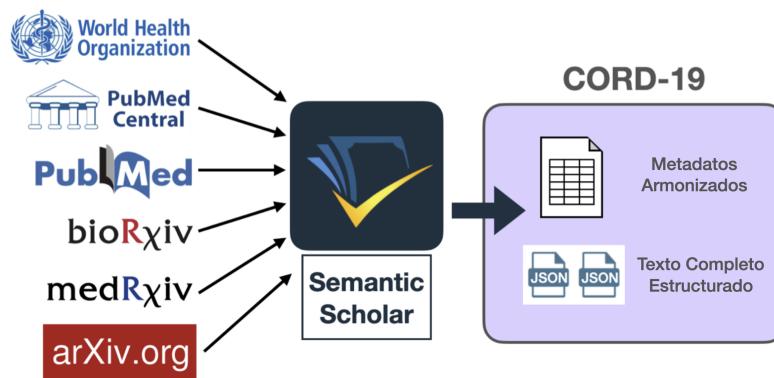


Figura 4.1: Los artículos y las publicaciones son recopilados a través de diferentes fuentes y procesados por Semantic Scholar para extraer los metadatos y el texto de los documentos.

Para la creación del corpus académico CORD-19 el Instituto Allen trabajó en colaboración con Microsoft Research, la Biblioteca Nacional de Medicina de los EE.UU., la Universidad de Georgetown, y la Oficina de Política Científica y Tecnológica de la Casa Blanca. Los artículos y las publicaciones son recopiladas y procesadas por el motor de Búsqueda Semantic Scholar [38], se armonizan los metadatos de los documentos y se eliminan los duplicados. También, los PDFs de los artículos son analizados para extraer el texto completo de los documentos.

4.1.2. Creación del Vocabulario del Corpus

De los artículos académicos, no solo necesitamos el texto de sus títulos y resúmenes o sus representaciones vectoriales, también necesitamos crear un vocabulario del corpus, necesario para poder encontrar las palabras que mejor describen los tópicos creados por los modelos. Para el procesamiento de los documentos utilizamos `spaCy`, una biblioteca para el procesamiento avanzado del lenguaje natural en Python y Cython creado empleando investigaciones de última generación [28].

El primer paso en la creación del vocabulario del corpus es la *tokenización* de los documentos, donde el texto del título y resumen de cada documento se transforma en la lista de las palabras contenidas dentro del texto. Luego de transformar los documentos en listas de palabras procedemos a eliminar las *palabras vacías*¹.

Las *palabras vacías* son aquellas palabras que carecen de significado como los artículos, pronombres y preposiciones. Estas palabras además de contener poca información, tienen una alta frecuencia en los textos y pueden distorsionar el análisis de las frecuencias de las palabras, por esto deben ser eliminadas. Ejemplos de *palabras vacías* en el lenguaje español son “para”, “el”, “en” y “aquel”. En el lenguaje inglés las palabras vacías pueden ser “the”, “are”, “but”, y “they”.

¹en inglés *stop-words*

Una vez eliminadas las palabras vacías comenzamos el proceso de *lematización*. La *lematización* es el proceso de reducir las palabras a su raíz, reduciendo de este modo el número de palabras diferentes en el vocabulario, pero manteniendo el significado de las palabras. Por ejemplo, “organiza”, “organizando” y “organizó” son diferentes conjugaciones de “organizar”, por lo que “organizar” se puede utilizar para representar estas tres palabras.

Después de terminar el proceso de *lematización*, guardamos las frecuencias de las palabras dentro del corpus y sus frecuencias por cada documento donde aparecen. Esta información será luego utilizada para analizar la calidad de los tópicos construidos por los modelos, y comparar nuestros Modelos de Tópicos con otros.

En el proceso de creación del vocabulario se analizaron 105,548 documentos, con un tamaño total de 12,677,221 palabras², y 193,448 palabras únicas dentro del corpus. La palabra de mayor frecuencia en el corpus es “patient” con 136,600 apariciones en 36,537 documentos.

4.2. Creación de los Modelos de Tópicos

Para la implementación del SRI-TOP empleamos el Modelo de Tópicos con Representación Distribuida Top2Vec, por su ya probada ventaja sobre los Modelos de Tópicos Probabilísticos [3]. En la versión original del Modelo de Tópicos Top2Vec, se emplea el modelo Doc2Vec para la representación vectorial de los documentos y Word2Vec para la representación de las palabras. Estos son modelos ligeros y rápidos que pueden ser entrenados sobre cualquier corpus con cierta facilidad³, y que no necesitan ajustes posteriores sobre los modelos obtenidos para obtener resultados satisfactorios, además de que no tienen limitaciones en cuanto a la longitud de los documentos en el momento de crear sus embeddings. Sin embargo, el SRI-TOP no solo debe ser rápido y eficaz, también debe devolver resultados de calidad y precisión ante la consulta de un usuario. Por este motivo decidimos utilizar modelos mucho más avanzados y potentes que Word2Vec y Doc2Vec como Sentence-BERT (SBERT) y SPECTER [3, 19].

Como en el SRI-TOP trabajamos con un corpus de artículos científicos, entonces podemos representar los documentos utilizando el texto de sus títulos y resúmenes, y con estos textos crear las representaciones vectoriales de los documentos. De esta forma solucionamos uno de los principales problemas presentes en modelos basados en Transformers como BERT, SBERT y SPECTER, el límite en cuanto a la longitud de los documentos aceptados (512 tokens).

De los Modelos de Lenguaje escogidos para el desarrollo de nuestro Modelo con Representación Distribuida de Tópicos, SBERT es un modelo con resultados vanguardias en la generación de embeddings para texto, oraciones y palabras [56]. Por otro lado, SPECTER es un modelo con resultados vanguardias en la representación de texto científico, pero con resultados pobres en cuanto a la representación de palabras, por lo que este no sería un Modelo de Lenguaje capaz de crear un espacio vectorial conjunto de palabras, documentos y tópicos de alta calidad. Por este motivo, decidimos desarrollar dos Modelos de Tópicos, comparar sus resultados y utilizar el modelo que brinde los mejores resultados. Un Modelo de Tópicos con SBERT para crear el espacio vectorial conjunto de palabras, documentos y tópicos. Y un Modelo de Tópicos que utilice de forma conjunta los modelos SPECTER y SBERT, el modelo SPECTER para el espacio vectorial conjunto de documentos y tópicos, y el modelo SBERT para el espacio vectorial conjunto de palabras y tópicos.

En el modelo de tópicos SPECTER-SBERT, el espacio vectorial SPECTER se utiliza para la

²después de haber eliminado las palabras vacías

³comparado con modelos mucho más avanzados como BERT o SPECTER

creación de los tópicos y la asignación de los documentos a sus tópicos más cercanos, mientras que el espacio vectorial SBERT se utiliza para la creación del vocabulario de los tópicos.

Una vez construidos ambos Modelos de Tópicos, primero utilizamos el método de evaluación *Ganancia de Información con Tópicos* para comparar sus rendimientos con otros Modelos de Tópicos. Luego, empleamos la *Homogeneidad de Tópicos* para comparar el rendimiento entre ellos. Antes, veamos el proceso de construcción de cada modelo.

4.2.1. Modelo de Tópicos SBERT

El primer paso en la creación del Modelo de Tópicos SBERT es la creación de los embeddings de los documentos con el Modelo de Lenguaje Sentence-BERT (SBERT). Estas representaciones vectoriales de los documentos serán utilizadas para encontrar los tópicos presentes en el corpus, y luego, para determinar el tamaño de los tópicos.

Después de haber generado los embeddings de los documentos, procedemos a la creación de los tópicos, la determinación del tópico más representativo para cada documento y la creación del vocabulario de los tópicos utilizando las palabras de sus documentos. Además, es posible reducir el número de tópicos en el corpus si se desean tópicos que sean más generales y que agrupen más documentos. La reducción del número de tópicos se realiza de forma jerárquica e iterativa. A continuación explicamos como realizamos cada uno de estos pasos.

Creación de los Tópicos

Para determinar el número de tópicos en el corpus, se emplean los embeddings de los documentos para encontrar las áreas densas de documentos en el espacio vectorial, pues cada área densa en este espacio semántico representa un grupo de documentos con un tópico subyacente importante. Como explicamos en la [Sección 3.2.3](#), antes de poder encontrar las áreas densas de documentos debemos reducir el número de dimensiones en los vectores de los documentos, dado que las altas dimensiones de los vectores⁴ provocan un elevado esparcimiento de los documentos en el espacio vectorial, además que trabajar con vectores de tantas dimensiones tiene un alto costo computacional. La reducción de dimensiones permite encontrar los clústeres de documentos en una forma más eficiente y precisa. Para reducir las dimensiones de los embeddings empleamos el algoritmo “Uniform Manifold Approximation and Projection for Dimension Reduction” (UMAP) [46, 47]. UMAP es una técnica de aprendizaje múltiple para la reducción de dimensiones capaz de conservar la estructura local y global de los espacios vectoriales, y que puede ser utilizado en conjuntos de datos muy grandes.

UMAP tiene varios parámetros de importancia. El *número de vecinos más cercanos* controla el equilibrio entre la conservación de la estructura global frente a la estructura local durante la reducción de dimensiones. Durante el desarrollo de nuestros modelos de tópicos encontramos que 15 como el *número de vecinos más cercanos* genera los mejores resultados, ya que este valor da el énfasis necesario en la estructura local del espacio vectorial. Como *métrica de distancia* en UMAP utilizamos la *similitud de coseno*, ya que puede medir la similitud de los documentos independientemente del tamaño de sus vectores. Como tamaño de dimensión final seleccionamos 5 dimensiones, porque en este valor encontramos el mejor balance en cuanto al costo computacional de encontrar los clústeres de documentos y la calidad de los clústeres encontrados.

⁴generalmente 300 dimensiones o más

Una vez que hayamos reducido las dimensiones de las representaciones vectoriales de los documentos, podemos proceder a encontrar los grupos densos de documentos. Para esto utilizamos “Hierarchical Density-Based Spatial Clustering of Applications with Noise” (HDBSCAN), que puede encontrar agrupamientos de vectores basados en la densidad de los vectores en el espacio vectorial en general. HDBSCAN para crear los clústeres de vectores, genera varios agrupamientos utilizando diferentes valores de distancia mínima entre los vectores de los clústeres, y evalúa estos agrupamientos para encontrar el valor de distancia mínima que genera los mejores resultados. Esto permite que HDBSCAN genere agrupamientos de vectores con varias densidades. Para encontrar los clústeres de documentos en el espacio vectorial utilizamos HDBSCAN con el parámetro *tamaño mínimo de clúster* con valor 15, ya que valores más grandes tienden a fusionar grupos densos de documentos no relacionados.

Con estos grupos densos de documentos ya podemos generar las representaciones vectoriales de los tópicos, pues cada clúster encontrado por HDBSCAN representa un tópico importante dentro del corpus. Para generar el embedding de los tópicos, por cada grupo denso de documentos encontrado, utilizamos el embedding original de estos documentos en el espacio SBERT, siendo el vector del tópico la media aritmética de los vectores de los documentos en el clúster. Así, por cada clúster encontrado en el espacio vectorial generamos un tópico.

El siguiente paso es determinar el tópico de cada documento. Para esto asignamos a cada documento el tópico más cercano en el espacio vectorial, utilizando la similitud de coseno como la métrica de distancia. El número de documentos asignados a cada tópico determina su tamaño, siendo los tópicos más grandes los más representativos del corpus.

Reducción del Número de Tópicos

Los tópicos generados originalmente representan los temas de mayor importancia encontrados en la colección de documentos, pero un usuario puede desear tópicos más generales o que agrupen más documentos. El proceso de reducción del tamaño del Modelo de Tópicos se realiza de forma iterativa y jerárquica, donde el modelo se reduce de un tamaño n a $n - 1$.

En una iteración de la reducción del número de tópicos se selecciona el tópico de menor tamaño⁵ y este se mezcla con su tópico más cercano en el espacio vectorial. En el proceso de unificación del tópico más pequeño con su tópico más cercano se genera una nueva representación vectorial para la unión de estos tópicos. El embedding del tópico resultante de la unión se obtiene a partir de la media ponderada de los vectores de ambos tópicos, usando como peso el tamaño de cada tópico. Siendo t_{peq} , t_{cer} las representaciones vectoriales del tópico más pequeño y su tópico más cercano, y n_{peq} , n_{cer} el número de documentos en cada tópico, entonces el vector del nuevo tópico t_{uni} resultante de la unión es:

$$t_{uni} = \frac{(n_{peq} \times t_{peq}) + (n_{cer} \times t_{cer})}{n_{peq} + n_{cer}} \quad (4.1)$$

Después de la unión de ambos tópicos, se vuelve a calcular el tamaño de los tópicos asignando nuevamente cada documento a su tópico más cercano. Esta reducción del tamaño del modelo de forma jerárquica tiene la ventaja de encontrar los tópicos más importantes dentro del corpus, ya que tiende a proteger los tópicos más representativos y de mayor tamaño durante el proceso de reducción del número de tópicos. En la *Figura 4.2* se muestra el modelo de tópicos SBERT luego de haber reducido su tamaño a 20 tópicos.

⁵El tópico con la menor cantidad de documentos

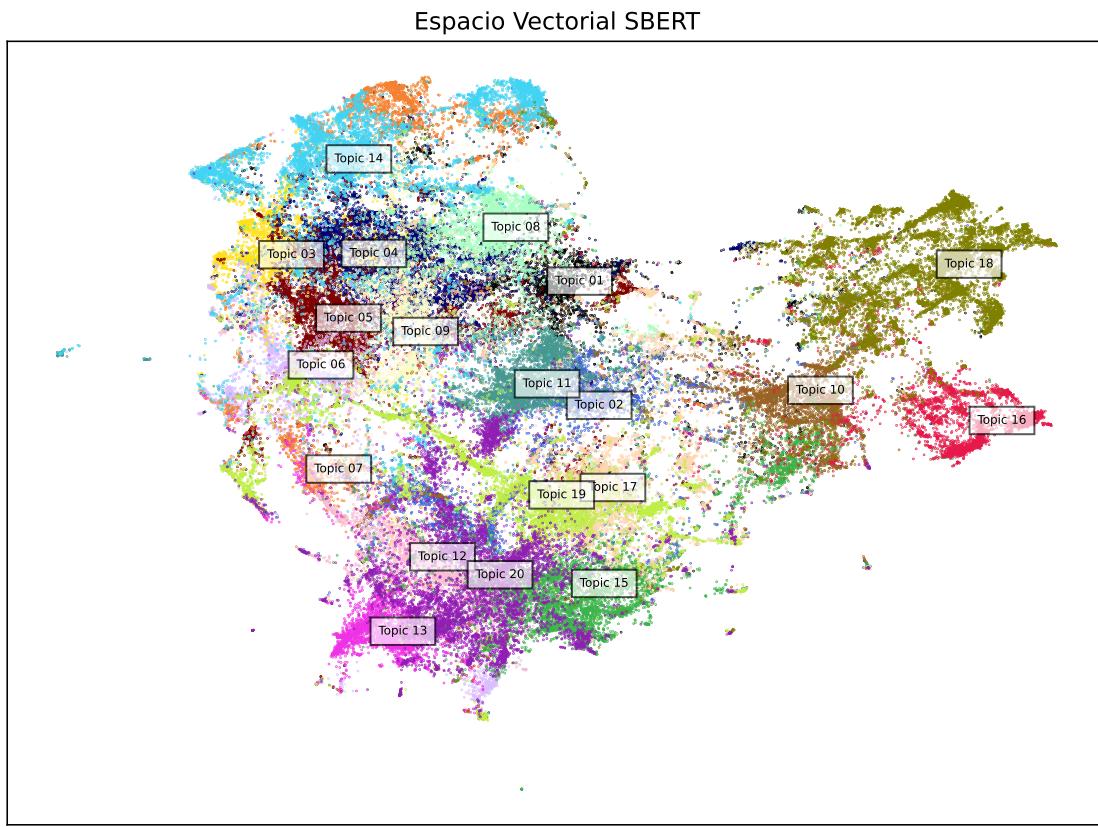


Figura 4.2: Representación en 2 dimensiones del espacio vectorial resultado de la aplicación del Modelo de Tópicos SBERT sobre el corpus CORD-19 con un tamaño de 20 tópicos.

Vocabulario de los Tópicos

El proceso de creación del vocabulario de un tópico consiste en encontrar las palabras más representativas del tópico, o sea, las palabras más cercanas al embedding del tópico en el espacio semántico. Primero extraemos las palabras de cada uno de sus documentos (los documentos del corpus más semánticamente similares al tópico). Luego, empleando el Modelo de Lenguaje SBERT creamos las representaciones vectoriales de estas palabras. Finalmente, para crear un vocabulario de tamaño n para el tópico, escogemos las n palabras más cercanas al embedding del tópico en el espacio vectorial. De esta forma creamos una descripción del tema al que se refiere el tópico.

Tabla 4.1: Los 20 tópicos encontrados por el Modelo de Tópicos SBERT ordenados por su tamaño, que también representa su importancia en el corpus. Las palabras en el vocabulario de los tópicos están ordenadas de más a menos similares a su tópico.

ID	Tamaño	Vocabulario
Topic 18	9,557 docs	surgery-specific, laparoscopic-guided, intraoperative-sutured, surgery-use, laparoscopy-guide, post-surgery, re-laparoscopy, surgery-analysis, surgery-expert, laparoscopically-assisted
Topic 14	7,187 docs	virus-antibody, antibody-to-virus, coronaviruses, coronavirus-associated, virus-animal, viruses-porcine, enterovirus-specific, virus-receptor, enteroaphthovirus, nonarboflaviviruse
Topic 15	6,968 docs	pandemic-specific, pandemic-adjusted, pandemic-response, health-care-related, COVID-19-related, patient-related, non-hospital-based, hospitalizations, clinical-oriented, post-hospitalization
Topic 04	6,773 docs	virus-protein, structure-antiviral, pathogen-targeting, phyto-antiviral, anti-lyssaviral, viral-cell, glycan-protein, pathogen-binding, pathogen-targeted, virus-carbohydrate
Topic 20	6,695 docs	influenza-pandemic, pandemic-flu, pandemic-specific, influenza-specific, pandemic-associated, pandemic-threat, intra-pandemic, influenza-like-illness, influenza-virus, influenza-associated
Topic 19	6,413 docs	coronavirus-associated, coronavirus-MERS-CoV, SARS-CoV-2-related, SARS-Coronavirus, SARS-CoV-related, coronavirus-229E, coronavirus-like, coronavirus-ENT, COVID-19-related, respiratory-virus
Topic 17	5,692 docs	COVID-19-related, coronavirus-associated, patients-infected, MERS-coronavirus, post-COVID, COVID-SAFER, HCoV-infected, HCoV-19, 2019-ncov-infected, COVID-positive
Topic 05	5,381 docs	coronavirus-associated, coronavirus, coronaviruses-S1, SARS-CoV-2-related, virus-ligand, anti-SARV-CoV-2, SARS-CoV-related, viral-receptor-dependent, virus-receptor, coronavirus-EMC
Topic 02	5,107 docs	respiratory-infection, pneumonia-risk, non-SARS-pneumonia, respiratory-illness, pneumonia-induced, pneumonia-associated, non-pneumonia, pneumonia-in-plan, pneumoniae-positive, pneumonia-relate
Topic 10	4,901 docs	ventilation-to-perfusion, ventilation-perfusion, bronchoscope-assisted, cardio-respiratory, volume-to-respiratory, respiratory-device, ventilator-dependent, non-ventilatory, post-cardiorespiratory, post-intubation

ID	Tamaño	Vocabulario
Topic 11	4,797 docs	respiratory-virus, respiratory-viruse, viral-asthma, respiratory-infected, respiratory-pathogen, rhinovirus-positive, rhinovirus-affected, rhinovirus-associate, enterovirus-rhinovirus, RSV-Rhinovirus
Topic 03	4,530 docs	RNA-virus, RNA-virus-triggered, viral-protein-associated, HCV-RNA, VP3-RNA, RNAs-transfected, WNV-RNA, HTNV-RNA, RNA-binding, RNA-launched
Topic 13	4,303 docs	epidemic-phase, epidemic-infected, epidemic-control, pandemic-spread, epidemic-curve-like, epidemic-suppression, disease-forecasting, outbreak-severity, endemic-epidemic, sub-epidemic
Topic 08	4,265 docs	innate-immunity, immune-modulation, cell-immune, innate-immune, immuno-pathogenetic, immune-deficient, neuro-immune, immune-enhance, immune-modulator, immune-driven
Topic 07	4,158 docs	wildlife-pathogen, animal-infected, zoonoses-pathogens, host-pathogen-environment, vector-host-pathogen, pathogen-contaminated, pathogen-transmission, rotavirus-bovine, pathogenicity, cattle-disease
Topic 16	4,136 docs	angio-embolization, angioembolization, stent-for-stroke, re-embolization, guidelines-stroke, thrombectomy-assisted, aneurysm-vessel, post-intra-arterial, stroke-specific, post-thrombolytic
Topic 09	3,936 docs	pathogen-targeting, nanoparticle-vaccinated, immunoinfectomic, specific-pathogen-free, vaccine-vector, pathogenomic, viral-antibody, immuno-assay, DNA-Vaccines, antibody-based
Topic 12	3,722 docs	emerging-disease, infectious-disease, pandemic-related, ebola-welfare, disease-endemic, disease-prevention, ebola-virus-disease, epidemiological, health-policy, public-health
Topic 01	3,523 docs	inflammation-mediated, inflammation-based, pro-inflammation, inflammation-modulating, anti-inflammation, pro-inflammatory, cytokines-induced, cytokine-induced, micro-inflammation, sepsis-mediated
Topic 06	3,504 docs	virus-diversity, viral-transcript, bat-vector-virus, mammal-virus, virus-bioinformatic, viral-encoded, RNA-viruses, virus-associated, virus-bacterium, inter-viral

4.2.2. Modelo de Tópicos SPECTER-SBERT

Como mencionamos anteriormente el Modelo de Lenguaje SPECTER genera muy buenas representaciones vectoriales para documentos con texto científico, pero pobres embeddings para las palabras en el vocabulario de estos documentos. Para resolver esta deficiencia en SPECTER, y crear un modelo de tópicos que pueda aprovecharse de los beneficios de sus representaciones de documentos, decidimos crear el modelo de tópicos con dos espacios vectoriales. Un espacio vectorial SPECTER utilizado para crear los embeddings de los documentos y determinar el tamaño de los tópicos, y un espacio vectorial SBERT para encontrar las palabras que mejor describen estos tópicos. Ambos espacios vectoriales se crean y se manipulan de forma paralela.

Creación de los Tópicos

El primer paso en la creación de los tópicos es generar los embeddings de los documentos en el espacio vectorial SPECTER y el espacio vectorial SBERT. El corpus CORD-19 utilizado para la creación del SRI-TOP tiene la ventaja de que junto con los textos y metadatos de los documentos también viene sus embeddings SPECTER. Por lo que solo necesitamos crear los embeddings SBERT de los documentos.

El espacio vectorial SPECTER es el que se va a utilizar para determinar el número de tópicos en el corpus. Para encontrar los tópicos, primero disminuimos a 5 las dimensiones de los vectores en este espacio vectorial utilizando el algoritmo UMAP, de forma similar a como se realizó en el Modelo de Tópicos SBERT. Luego de haber reducido las dimensiones, empleando el algoritmo HDBSCAN encontramos los grupos densos de documentos en el espacio vectorial SPECTER. Cada grupo denso de documentos representa un tópico de importancia en el corpus, siendo el número de grupos densos el número de tópicos presentes en la colección de documentos.

Una vez que tenemos los grupos densos de documentos comenzamos el proceso de creación de los embeddings de los tópicos de forma paralela en ambos espacios vectoriales SPECTER y SBERT. Para generar los embeddings de los tópicos en el espacio vectorial SPECTER se calcula la media aritmética de los embeddings SPECTER de los documentos dentro del grupo denso que este tópico representa. De igual forma, para generar el embedding SBERT de un tópico, utilizamos las representaciones vectoriales SBERT de los documentos en su grupo denso. Siguiendo este procedimiento por cada tópico obtenemos su representación en los espacios vectoriales SPECTER y SBERT.

Para determinar a que tópico pertenece cada documento, y el tamaño de los tópicos, empleamos solamente el espacio vectorial SPECTER. Cada documento en el corpus se asigna a su tópico más cercano, utilizando la similitud de coseno como la métrica de distancia. El número de documentos asignados a cada tópico determina su tamaño, siendo los tópicos más grandes los más representativos de los temas hablados en el corpus.

Reducción del Número de Tópicos

La reducción del número de tópicos en el Modelo de Tópicos SPECTER-SBERT, al igual que en el Modelo de Tópicos SBERT, se realiza de forma jerárquica e iterativa. En cada iteración se selecciona el tópico de menor tamaño, y este, se unifica con su tópico más cercano en el espacio vectorial SPECTER. La unificación se realiza de forma paralela en ambos espacios vectoriales, utilizando los embeddings SBERT de ambos tópicos para generar el embedding de la unificación. Sea S el tópico más pequeño con $S_{specter}$, S_{sbert} sus embeddings, N_S su tamaño, y C su tópico más cercano en el espacio vectorial SPECTER con $C_{specter}$, C_{sbert} sus embeddings, N_C su tamaño,

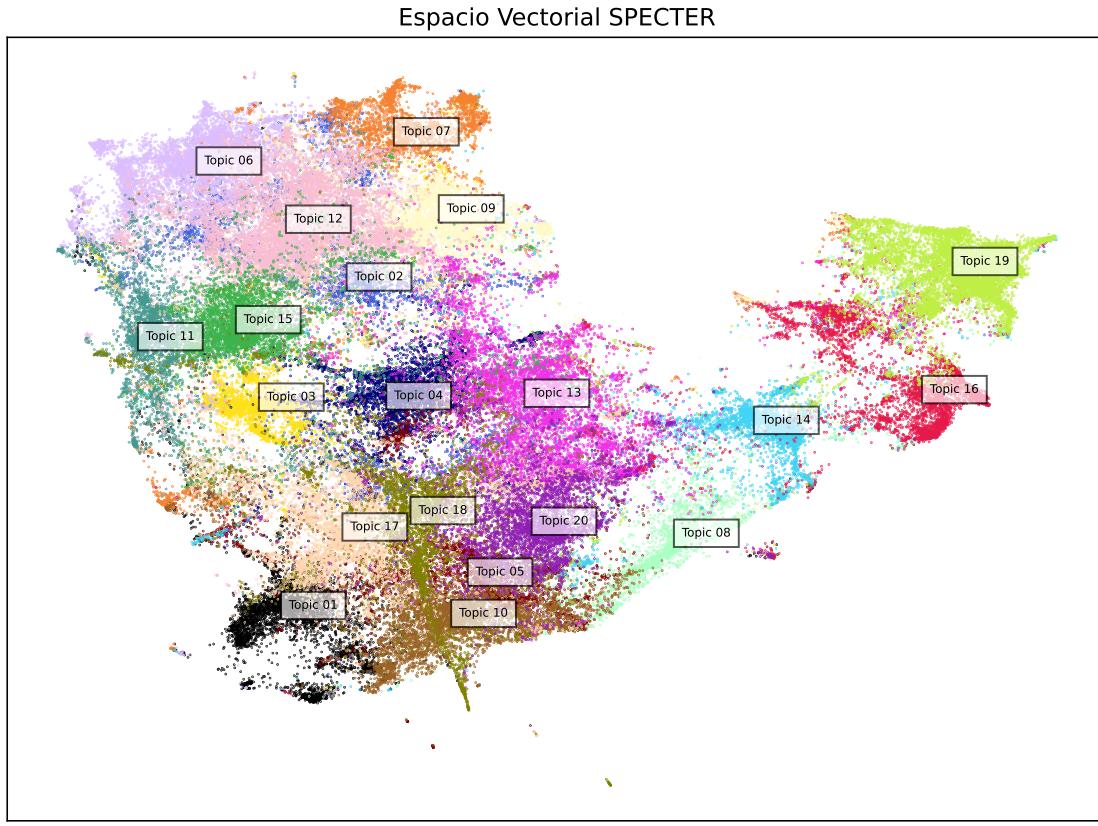


Figura 4.3: Representación en 2 dimensiones del espacio vectorial resultado de la aplicación del Modelo de Tópicos SPECTER-SBERT sobre el corpus CORD-19 con un tamaño de 20 tópicos.

entonces el tópico U resultado de la unificación de estos tópicos tiene las siguientes representaciones vectoriales:

$$\begin{aligned} U_{specter} &= \frac{(N_S \times S_{specter}) + (N_C \times C_{specter})}{N_S + N_C} \\ U_{sbert} &= \frac{(N_S \times S_{sbert}) + (N_C \times C_{sbert})}{N_S + N_C} \end{aligned} \quad (4.2)$$

Después de la unión del tópico más pequeño con su tópico más cercano, se vuelve a calcular el tamaño de los tópicos asignando cada documento a su tópico más cercano en el espacio vectorial SPECTER. En la *Figura 4.3* se muestra el modelo de tópicos SPECTER luego de haber reducido su tamaño a 20 tópicos.

Vocabulario de los Tópicos

El espacio vectorial SBERT es el empleado para encontrar las palabras más representativas de los tópicos. Para encontrar las palabras que mejor describen tópico utilizamos los embeddings SBERT de las palabras en el vocabulario de sus documentos, y seleccionamos las n palabras más cercanas al tópico en el espacio vectorial SBERT. En la *Tabla 4.2* mostramos el vocabulario generado para los tópicos en el modelo SPECTER-SBERT.

El modelo de tópicos SPECTER-SBERT se beneficia de las propiedades temáticas y semánticas que tiene el modelo de lenguaje SPECTER para el texto científico, además de mantener la interpretabilidad de sus tópicos empleando el modelo de lenguaje SBERT. De esta forma pudimos construir un modelo de tópicos robusto que sirva en tareas de recuperación de información.

Tabla 4.2: Los 20 tópicos encontrados por el Modelo de Tópicos Top2Vec con SPECTER. Este modelo tiene 0.017 de *ganancia de información con tópicos*, un incremento de un 20% con respecto al modelo Top2Vec con Sentence-BERT.

ID	Tamaño	Vocabulario
Topic 13	8,793 docs	coronavirus-associated, COVID-19-related, patients-infected, respiratory-virus, pathogen-therapy, pneumonia-induced, corona-virus-infecte, HBoV-pneumonia, non-pneumonia, pathogen-specific
Topic 12	8,757 docs	virus-immunity, virus-immune, antibody-to-virus, viral-receptor-dependent, antiviral-signaling, virus-expressed, viral-induce, virus-ligand, reovirus-mediated, viral-intracellular
Topic 19	8,310 docs	laparoscopic-surgery, laparoscopic-guided, laparoscopically-assisted, laparoscopy-guide, laparoscopy-experienced, preserving-laparoscopic, re-laparotomy, laparoscopic-converted, laparoscopic-ELAPE, intraoperative-sutured
Topic 06	7,472 docs	viral-protein-associated, proteins-virus, virus-bioinformatic, virus-membrane, viral-derived, antiviral-signaling, protein-pathogen, calicivirus-protein, VPg-proteinase, VP3-RNA
Topic 17	5,954 docs	coronavirus, coronavirus-associated, MERS-coronavirus, coronavirus-MERS-CoV, non-pandemic-flu, outbreak-associated, outbreak-related, post-pandemic, infectious-disease, COVID-19
Topic 16	5,349 docs	angio-embolization, angioembolization, angiography-predicted, stent-for-stroke, neovascularization, stroke-thrombolytic, CT-perfusion, thrombectomy-assisted, sub-angiographic, post-intra-arterial

ID	Tamaño	Vocabulario
Topic 01	5,178 docs	pandemic-spread, epidemic-infected, epidemic-spreading, epidemic-systematic, disease-forecasting, epidemic-control, COVID-pandemic, outbreak-severity, multi-outbreak, outbreaks-including
Topic 09	5,023 docs	cytokine-mediated, inflammation-mediated, innate-immunity-driven, innate-immunity, inflammation-promoting, cytokine-targeted, cytokines-induced, pro-inflammation, interleukin-1a, IL-2-mediated
Topic 10	4,792 docs	post-pandemic, pandemic-related, pandemic-driven, COVID-CARE, health-policy, disaster-health-politic, health-threat, health-care-related, public-health, disease-avoidance
Topic 02	4,747 docs	coronavirus-associated, anti-coronaviral, coronaviruses-S1, SARS-CoV-2-related, coronavirus-229E, coronavirus-EMC, SARS-CoV-related, virus-receptor, coronaviral, anti-viral
Topic 04	4,678 docs	respiratory-virus, respiratory-viruse, influenza-pneumonia, RSV-influenza, influenza-infection, rhinovirus-positive, rhinovirus-related, rhinovirus-associated, viral-infection, RSV-Rhinovirus
Topic 15	4,466 docs	pig-infectious-dose, virus-antibody, virus-intestinal, rabbit-infectious, virus-serum-toxin, pathogen-specific, FECV-infected, ebolavirus-specific, zoonoticus-infected, rotavirus-bovine
Topic 07	4,410 docs	structure-antiviral, phyto-antiviral, nano-antimicrobial, compounds-targets-disease, anti-coronaviral, medicinal-chemistry, virus-inhibitory, anti-plant-viruse, anti-influenza-virus, anti-microbial
Topic 20	4,334 docs	COVID-19-related, intra-pandemic, pandemic-specific, COVID-Activated, non-COVID-related, post-pandemic, COVID-CT, pandemic-adjusted, COVID-suspected, CT-COVID
Topic 11	4,296 docs	mammal-virus, species-astrovirus-type, coronavirus-associated, rhinovirus-specific, hantaviruses, influenza-virus, nonarbovirose, turdivirus, MERS-coronavirus, criterion-intervirus
Topic 05	4,002 docs	epidemiology-infection, influenza-pandemic, health-surveillance, disease-reporting, infection-prevention, disease-prevention, pandemic-threat, epidemiology, inter-pandemic, outbreak-related
Topic 18	3,885 docs	coronavirus, coronavirus-associated, COVID-19, coronavirus-NL63, post-pandemic, intra-pandemic, ebolavirus, SARS-CoV-related, anti-pandemic, SARS-CoV-2-related

ID	Tamaño	Vocabulario
Topic 14	3,832 docs	ventilation-to-perfusion, ventilation-perfusion, cardio-respiratory, oxygenation-monitoring, anesthetic-conserving, post-intubation, ventilator-dependent, post-cardiorespiratory, respiratory-device, intubation-induced
Topic 03	3,795 docs	virus-antibody, pathogen-detection, single-pathogen-test, coronavirus-associated, viral-testing, coronavirus-229E, rNPiBV-ELISA, respiratory-virus, rhinovirus-positive, rPDCoV-N-ELISA
Topic 08	3,475 docs	meta-analyses, clinical-trial, systematic-review, symptom-reporting, cohort-study, pharmacovigilance, non-pharmacologic, delirium-prevention, medication-weight, health-related

4.3. Evaluación de Modelos de Tópicos

Normalmente, los modelos de tópicos se evalúan de la siguiente manera. Primero, se extrae un subconjunto del corpus como conjunto de prueba. Después, se entrena un grupo de modelos de tópicos con el resto del corpus para examinar sus rendimientos con el subconjunto de prueba. Por último, se elige el modelo con el que se obtuvieron los mejores resultados [8]. Pero, los modelos de tópicos son a menudo empleados para organizar, resumir y ayudar a los usuarios en la exploración de grandes corpus, y no existe ninguna razón técnica para suponer que la precisión retenida sobre el conjunto de prueba corresponde a una mejor organización o a una interpretación más fácil de los documentos.

Una forma más natural de evaluar los modelos de tópicos es evaluar qué tan bien los tópicos describen los documentos. Esta evaluación permite ver que tan informativos pueden ser los tópicos para un usuario. Para esto podemos usar la *información mutua*⁶ [65], que puede medir la información obtenida sobre los documentos cuando son descritos por sus palabras temáticas. Al método para evaluar la información recibida sobre los documentos cuando estos son descritos por sus tópicos lo llamamos *ganancia de información con tópicos*⁷[3].

4.3.1. Ganancia de Información con Tópicos

Los métodos tradicionales de modelado de tópicos, como LDA, discretizan el espacio de tópicos y describen los documentos como una mezcla de estos tópicos. Para poder evaluar un conjunto de estos T tópicos generados a partir de los documentos D , se calcula la información total obtenida para cada documento cuando son descritos por las proporciones de tópicos dadas por el modelo de tópicos.

Por otro lado, Top2Vec aprende una representación continua de tópicos y coloca los documentos en este espacio en correspondencia con sus tópicos. Un vector de tópicos encontrado por Top2Vec

⁶del inglés mutual information

⁷del inglés topic information gain

representa el tópico común entre un grupo de documentos, o el promedio de sus tópicos individuales. Para evaluar un conjunto de tópicos T generados por Top2Vec a partir de una colección de documentos D , los documentos se dividen en subconjuntos, correspondiendo cada subconjunto a los vectores de documento con el mismo vector de tópico más cercano. Por lo tanto, cada documento se asigna exactamente a un tópico. Para evaluar estos tópicos, la *información total obtenida* se mide para cada uno de los subconjuntos de documentos descritos por las palabras más cercanas a su vector de tópico.

La *ganancia de información total*, o información mutua, de todos los documentos D cuando son descritos por todas las palabras W , viene dada por:

$$I(D, W) = \sum_{d \in D} \sum_{w \in W} P(d, w) \log \left(\frac{P(d, w)}{P(d)P(w)} \right) \quad (4.3)$$

La contribución de cada co-ocurrencia entre un documento d y una palabra w al cálculo de la ganancia de información puede verse como la *cantidad de información con probabilidad-ponderada* (PWI⁸) que d y w contribuyen a la ganancia de información total [1], dada por:

$$\text{PWI}(d, w) = P(d, w) \log \left(\frac{P(d, w)}{P(d)P(w)} \right) \quad (4.4)$$

Los tópicos son distribuciones sobre todo el vocabulario W . Sin embargo, para poder evaluar su utilidad para un usuario, tenemos que evaluarlos usando las n palabras principales del tópico (las n palabras de mayor probabilidad en el tópico con LDA, o las n palabras más cercanas al tópico en el espacio semántico con Top2Vec).

Para evaluar modelos de tópicos donde cada documento se asigna a solo un tópico, cada tópico $t \in T$, tendrá un conjunto de n palabras $W_t \subset W$ y m documentos $D_t \subset D$. La *ganancia de información* sobre todos los documentos cuando son descritos por su tópico correspondiente está dada por:

$$\begin{aligned} \text{PWI}(T) &= \sum_{t \in T} \sum_{d \in D_t} \sum_{w \in W_t} P(d, w) \log \left(\frac{P(d, w)}{P(d)P(w)} \right) \\ &= \sum_{t \in T} \sum_{d \in D_t} \sum_{w \in W_t} P(d, w) P'(w) \log \left(\frac{P(d, w)}{P(d)P(w)} \right) \end{aligned} \quad (4.5)$$

En la *Fórmula 4.5*, $P(w)$ es la probabilidad marginal de la palabra w en todos los documentos D , y se utiliza para calcular el término logarítmico que es la *información mutua de pares* entre w y d (pairwise mutual information)[1, 17]. $P'(w)$ es la probabilidad de la palabra w en el tópico t , que se utiliza para calcular la información mutua esperada [65], o la *ganancia de información* sobre el documento d dada la palabra w del tópico t . Como estamos midiendo la *ganancia de información* sobre cada documento dadas las palabras de un tópico especificado a priori, entonces $P'(w)$ es 1 y puede omitirse [1, 3], lo que da lugar a:

$$\text{PWI}(T) = \sum_{t \in T} \sum_{d \in D_t} \sum_{w \in W_t} P(d, w) \log \left(\frac{P(d, w)}{P(d)P(w)} \right) \quad (4.6)$$

⁸del inglés probability-weighted information

Alternativamente, la ecuación se puede generalizar para el caso donde cada documento está representado por varios tópicos. Para este tipo de modelo, se sustituye $P'(w)$ por $P(t)$, que es la proporción de la distribución de palabras del tópico t utilizadas para representar el documento d :

$$\text{PWI}(T) = \sum_{d \in D} \sum_{t \in T} \sum_{w \in W_t} P(d, w) P(t) \log \left(\frac{P(d, w)}{P(d)P(w)} \right) \quad (4.7)$$

Usando la *Fórmula 4.6* y la *Fórmula 4.7*, se pueden comparar diferentes grupos de tópicos. Un valor mayor de *ganancia de información* indica que los tópicos $t \in T$ son más informativos de sus documentos correspondientes. Si los tópicos contienen palabras como “la”, “para”, “es” u otras palabras no informativas, recibirán valores más bajos de *ganancia de información*. Esto se debe en gran parte a la presencia del término $P(d|w)$ en la operación, ya que la probabilidad de cualquier documento dada una palabra común es muy baja. Por lo tanto, la *ganancia de información* también es baja. Las palabras que están presentes mayoritariamente en el subconjunto de documentos pertenecientes al tópico conducen a una mayor *ganancia de información*, ya que son palabras relevantes para estos documentos. Además, se obtendrán valores bajos de ganancia de información si el modelo de tópicos asigna tópicos a los documentos equivocados. La *ganancia de información con tópicos* mide la calidad de las palabras que describen los tópicos con respecto a los documentos con los que están asociados. Por lo que, la *Fórmula 4.6* y la *Fórmula 4.7* dan valores que se corresponden con lo que es intuitivamente más informativo. Por lo antes descrito, la *ganancia de información con tópicos* puede considerarse un buen método para la evaluación de modelos de tópicos.

4.3.2. Comparación con Ganancia de Información

Existen dos interpretaciones de la *ganancia de información* una empleando la intuición detrás del algoritmo de búsqueda tf-idf llamada PWI_{tf-idf}, y otra que emplea las frecuencias exactas de los documentos y las palabras en el corpus llamada PWI_{exact}⁹.

La fórmula para calcular cada uno de estos valores de *ganancia de información* es la siguiente:

$$\text{PWI}_{\text{tf-idf}}(w_i, d_j) = \frac{f_{ij}}{F} \log \frac{N}{N_i} \quad (4.8)$$

$$\text{PWI}_{\text{exact}}(w_i, d_j) = \frac{f_{ij}}{F} \log \frac{F f_{ij}}{f_{w_i} f_{d_j}} \quad (4.9)$$

donde N es el número total de documentos, N_i el número de documentos que contienen w_i , f_{ij} la frecuencia de la palabra w_i en el documento d_j , f_{w_i} la frecuencia de la palabra w_i en el corpus, f_{d_j} la suma de todas las frecuencias f_{ij} de las palabras w_i en el documento d_j (la longitud del documento), y F la suma de todas las frecuencias f_{d_j} de los documentos d_j en el corpus.

Estas fórmulas las podemos utilizar para obtener la ganancia de información obtenida por cada tópico T :

$$\text{PWI}(T) = \sum_{d \in D_t} \sum_{w \in W_t} \text{PWI}(w_i, d_j) \quad (4.10)$$

donde D_t son los documentos pertenecientes al tópico y W_t las palabras utilizadas para describirlo. La ganancia de información del Modelo de Tópicos M queda como:

⁹PWI viene del inglés Probability-Weighted amount of Information.

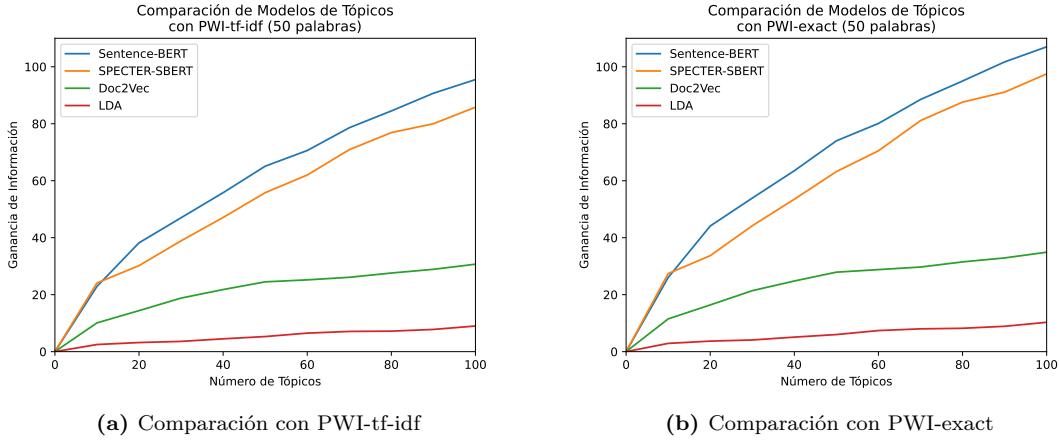


Figura 4.4: Comparación de los Modelos de Tópicos SBERT y SPECTER-SBERT utilizando la *ganancia de información con tópicos*.

$$\text{PWI}(M) = \sum_{T \in M} \text{PWI}(T) \quad (4.11)$$

Con estas fórmulas podemos comprobar el valor informativo de los modelos construidos, y comprobar sus valores con otros Modelos de Tópicos construidos sin importar que estos sean modelos LDA o Top2Vec. En la *Figura 4.4* se puede observar la comparación de los modelos de tópicos creados con otros modelos.

A simple vista se puede observar que los tópicos generados por los modelos de tópicos SBERT y SPECTER-SBERT son superiores a modelos de tópicos anteriores. Pero la comparación específica entre SBERT y SPECTER-SBERT carece de objetividad cuando se emplea la *ganancia de información*, ya que este método de evaluación emplea las palabras en el vocabulario de los tópicos para descubrir que tan informativos estos son. El problema de esta comparación con nuestros modelos radica en que ambos utilizan el espacio vectorial SBERT para generar sus vocabularios, por lo que los resultados pueden estar parcializados para un modelo que emplea solamente el espacio vectorial SBERT, comparado con un modelo que emplea SPECTER y SBERT de forma conjunta. Es por esto que decidimos utilizar la homogeneidad de tópicos para determinar finalmente que modelo de tópicos es el más robusto.

Tabla 4.3: Los 20 tópicos del Modelo de Tópicos SBERT ordenados por su *ganancia de información* utilizando PWI-exact.

ID	PWI	Vocabulario
Topic 19	4.8	coronavirus-associated, coronavirus-MERS-CoV, SARS-CoV-2-related, SARS-Coronavirus, SARS-CoV-related, coronavirus-229E, coronavirus-like, coronavirus-ENT, COVID-19-related, respiratory-virus
Topic 18	4.5	surgery-specific, laparoscopic-guided, intraoperative-sutured, surgery-use, laparoscopy-guide, post-surgery, re-laparoscopy, surgery-analysis, surgery-expert, laparoscopically-assisted
Topic 20	4.3	influenza-pandemic, pandemic-flu, pandemic-specific, influenza-specific, pandemic-associated, pandemic-threat, intra-pandemic, influenza-like-illness, influenza-virus, influenza-associated
Topic 17	4.2	COVID-19-related, coronavirus-associated, patients-infected, MERS-coronavirus, post-COVID, COVID-SAFER, HCoV-infected, HCoV-19, 2019-ncov-infected, COVID-positive
Topic 05	3.9	coronavirus-associated, coronavirus, coronaviruses-S1, SARS-CoV-2-related, virus-ligand, anti-SARV-CoV-2, SARS-CoV-related, viral-receptor-dependent, virus-receptor, coronavirus-EMC
Topic 03	3.8	RNA-virus, RNA-virus-triggered, viral-protein-associated, HCV-RNA, VP3-RNA, RNAs-transfected, WNV-RNA, HTNV-RNA, RNA-binding, RNA-launched
Topic 15	3.1	pandemic-specific, pandemic-adjusted, pandemic-response, health-care-related, COVID-19-related, patient-related, non-hospital-based, hospitalizations, clinical-oriented, post-hospitalization
Topic 13	2.4	epidemic-phase, epidemic-infected, epidemic-control, pandemic-spread, epidemic-curve-like, epidemic-suppression, disease-forecasting, outbreak-severity, endemic-epidemic, sub-epidemic
Topic 09	2.2	pathogen-targeting, nanoparticle-vaccinated, immunoinfectomic, specific-pathogen-free, vaccine-vector, pathogenomic, viral-antibody, immuno-assay, DNA-Vaccines, antibody-based
Topic 02	1.9	respiratory-infection, pneumonia-risk, non-SARS-pneumonia, respiratory-illness, pneumonia-induced, pneumonia-associated, non-pneumonia, pneumonia-in-plan, pneumoniae-positive, pneumonia-relate

ID	PWI	Vocabulario
Topic 14	1.6	virus-antibody, antibody-to-virus, coronaviruses, coronavirus-associated, virus-animal, viruses-porcine, enterovirus-specific, virus-receptor, enteroaphthovirus, nonarboflaviviruse
Topic 04	1.3	virus-protein, structure-antiviral, pathogen-targeting, phyto-antiviral, anti-lyssaviral, viral-cell, glycan-protein, pathogen-binding, pathogen-targeted, virus-carbohydrate
Topic 11	1.2	respiratory-virus, respiratory-viruse, viral-asthma, respiratory-infected, respiratory-pathogen, rhinovirus-positive, rhinovirus-affected, rhinovirus-associate, enterovirus-rhinovirus, RSV-Rhinovirus
Topic 01	1.1	inflammation-mediated, inflammation-based, pro-inflammation, inflammation-modulating, anti-inflammation, pro-inflammatory, cytokines-induced, cytokine-induced, micro-inflammation, sepsis-mediated
Topic 12	1.1	emerging-disease, infectious-disease, pandemic-related, ebola-welfare, disease-endemic, disease-prevention, ebola-virus-disease, epidemiological, health-policy, public-health
Topic 16	1.0	angio-embolization, angioembolization, stent-for-stroke, re-embolization, guidelines-stroke, thrombectomy-assisted, aneurysm-vessel, post-intra-arterial, stroke-specific, post-thrombolytic
Topic 07	0.7	wildlife-pathogen, animal-infected, zoonoses-pathogens, host-pathogen-environment, vector-host-pathogen, pathogen-contaminated, pathogen-transmission, rotavirus-bovine, pathogenicity, cattle-disease
Topic 08	0.5	innate-immunity, immune-modulation, cell-immune, innate-immune, immuno-pathogenetic, immune-deficient, neuro-immune, immune-enhance, immune-modulator, immune-driven
Topic 10	0.3	ventilation-to-perfusion, ventilation-perfusion, bronchoscope-assisted, cardio-respiratory, volume-to-respiratory, respiratory-device, ventilator-dependent, non-ventilatory, post-cardiorespiratory, post-intubation
Topic 06	0.3	virus-diversity, viral-transcript, bat-vector-virus, mammal-virus, virus-bioinformatic, viral-encoded, RNA-viruses, virus-associated, virus-bacterium, inter-viral

Tabla 4.4: Los 20 tópicos del Modelo de Tópicos SPECTER-SBERT ordenados por su *ganancia de información* utilizando PWI-exact.

ID	PWI	Vocabulario
Topic 19	5.1	laparoscopic-surgery, laparoscopic-guided, laparoscopically-assisted, laparoscopy-guide, laparoscopy-experienced, preserving-laparoscopic, re-laparotomy, laparoscopic-converted, laparoscopic-ELAPE, intraoperative-sutured
Topic 20	4.0	COVID-19-related, intra-pandemic, pandemic-specific, COVID-Activated, non-COVID-related, post-pandemic, COVID-CT, pandemic-adjusted, COVID-suspected, CT-COVID
Topic 17	3.7	coronavirus, coronavirus-associated, MERS-coronavirus, coronavirus-MERS-CoV, non-pandemic-flu, outbreak-associated, outbreak-related, post-pandemic, infectious-disease, COVID-19
Topic 02	2.7	coronavirus-associated, anti-coronaviral, coronaviruses-S1, SARS-CoV-2-related, coronavirus-229E, coronavirus-EMC, SARS-CoV-related, virus-receptor, coronaviral, anti-viral
Topic 01	2.7	pandemic-spread, epidemic-infected, epidemic-spreading, epidemic-systematic, disease-forecasting, epidemic-control, COVID-pandemic, outbreak-severity, multi-outbreak, outbreaks-including
Topic 13	2.0	coronavirus-associated, COVID-19-related, patients-infected, respiratory-virus, pathogen-therapy, pneumonia-induced, corona-virus-infecte, HBoV-pneumonia, non-pneumonia, pathogen-specific
Topic 18	1.7	coronavirus, coronavirus-associated, COVID-19, coronavirus-NL63, post-pandemic, intra-pandemic, ebolavirus, SARS-CoV-related, anti-pandemic, SARS-CoV-2-related
Topic 10	1.5	post-pandemic, pandemic-related, pandemic-driven, COVID-CARE, health-policy, disaster-health-politic, health-threat, health-care-related, public-health, disease-avoidance
Topic 04	1.4	respiratory-virus, respiratory-viruse, influenza-pneumonia, RSV-influenza, influenza-infection, rhinovirus-positive, rhinovirus-related, rhinovirus-associated, viral-infection, RSV-Rhinovirus
Topic 16	1.3	angio-embolization, angioembolization, angiography-predicted, stent-for-stroke, neovascularization, stroke-thrombolytic, CT-perfusion, thrombectomy-assisted, sub-angiographic, post-intra-arterial

ID	PWI	Vocabulario
Topic 06	1.3	viral-protein-associated, proteins-virus, virus-bioinformatic, virus-membrane, viral-derived, antiviral-signaling, protein-pathogen, calicivirus-protein, VPg-proteinase, VP3-RNA
Topic 11	1.2	mammal-virus, species-astrovirus-type, coronavirus-associated, rhinovirus-specific, hantaviruses, influenza-virus, nonarboflaviviruse, turdivirus, MERS-coronavirus, criterion-intervirus
Topic 15	1.1	pig-infectious-dose, virus-antibody, virus-intestinal, rabbit-infectious, virus-serum-toxin, pathogen-specific, FECV-infected, ebolavirus-specific, zooepidemicus-infected, rotavirus-bovine
Topic 09	1.0	cytokine-mediated, inflammation-mediated, innate-immunity-driven, innate-immunity, inflammation-promoting, cytokine-targeted, cytokines-induced, pro-inflammation, interleukin-1a, IL-2-mediated
Topic 07	0.9	structure-antiviral, phyto-antiviral, nano-antimicrobial, compounds-targets-disease, anti-coronaviral, medicinal-chemistry, virus-inhibitory, anti-plant-viruse, anti-influenza-virus, anti-microbial
Topic 08	0.7	meta-analyses, clinical-trial, systematic-review, symptom-reporting, cohort-study, pharmacovigilance, non-pharmacologic, delirium-prevention, medication-weight, health-related
Topic 03	0.5	virus-antibody, pathogen-detection, single-pathogen-test, coronavirus-associated, viral-testing, coronavirus-229E, rNpIBV-ELISA, respiratory-virus, rhinovirus-positive, rPDCoV-N-ELISA
Topic 05	0.4	epidemiology-infection, influenza-pandemic, health-surveillance, disease-reporting, infection-prevention, disease-prevention, pandemic-threat, epidemiology, inter-pandemic, outbreak-related
Topic 12	0.3	virus-immunity, virus-immune, antibody-to-virus, viral-receptor-dependent, antiviral-signaling, virus-expressed, viral-induce, virus-ligand, reovirus-mediated, viral-intracellular
Topic 14	0.3	ventilation-to-perfusion, ventilation-perfusion, cardio-respiratory, oxygenation-monitoring, anesthetic-conserving, post-intubation, ventilator-dependent, post-cardiorespiratory, respiratory-device, intubation-induced

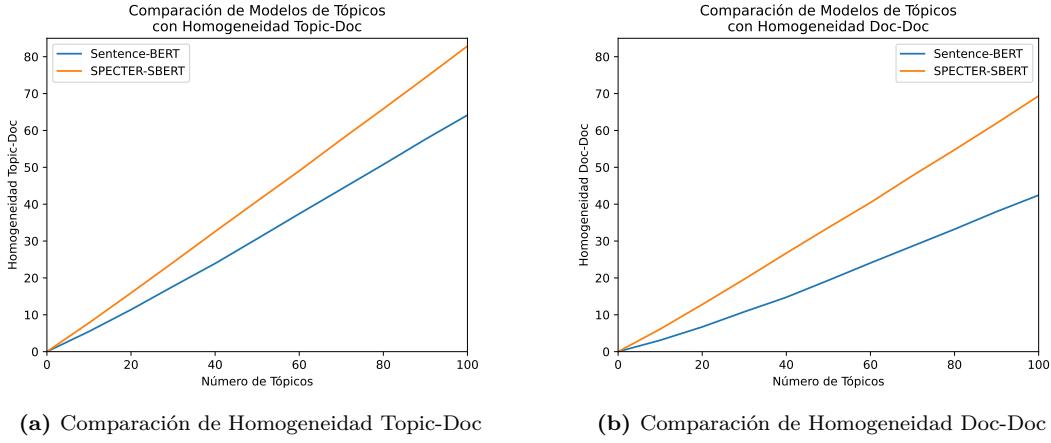


Figura 4.5: Comparación de los Modelos de Tópicos SBERT y SPECTER-SBERT utilizando Homogeneidad de Tópicos.

4.3.3. Comparación utilizando Homogeneidad

La homogeneidad es una característica que puede ser empleada para evaluar que tan bien agrupados están un grupo de vectores en un espacio vectorial. No es un concepto nuevo, este ya ha sido utilizado de forma exitosa para la detección de rumores en Twitter [35], y para evaluar la relación existente entre los documentos en un corpus y determinar si comparten un tópico en común [62]. Mientras más elevada sea la homogeneidad de un grupo de vectores, mayor será la relación existente entre ellos.

En los espacios vectoriales SBERT y SPECTER, las distancias entre los vectores de los documentos y tópicos representan la similitud semántica y temática entre ellos, por lo que la homogeneidad se puede utilizar para determinar que tópicos tienen una alta similitud con los documentos que representan, y en qué tópicos existe una mayor interconexión dentro del tema abordado por sus documentos.

La homogeneidad en Modelos de Tópicos tiene dos interpretaciones. La Homogeneidad Topic-Doc ($H_{topic-doc}$) donde se evalúa si un tópico es una buena representación para sus documentos, calculando el promedio de las similitudes del tópico con sus documentos. La Homogeneidad Doc-Doc ($H_{doc-doc}$) evalúa la interconexión existente entre los documentos de un tópico y la existencia de un tópico subyacente en los documentos, calculando el promedio de las similitudes entre los documentos. Las fórmulas para calcular ambos valores son las siguientes:

$$H_{topic-doc}(T) = \frac{\sum_{i=1}^N sim(d_i, T)}{N} \quad (4.12)$$

$$H_{doc-doc}(T) = \frac{\sum_{j=2}^N \sum_{i=1}^{j-1} sim(d_i, d_j)}{N^2/2} \quad (4.13)$$

dónde T es el tópico del que estamos calculando su homogeneidad, N es el tamaño del tópico y d_i los documentos del tópico. La función $sim(d_i, T)$ calcula la similitud coseno entre los embeddings del documento d_i y el tópico T , la función $sim(d_i, d_j)$ calcula la similitud coseno entre los documentos

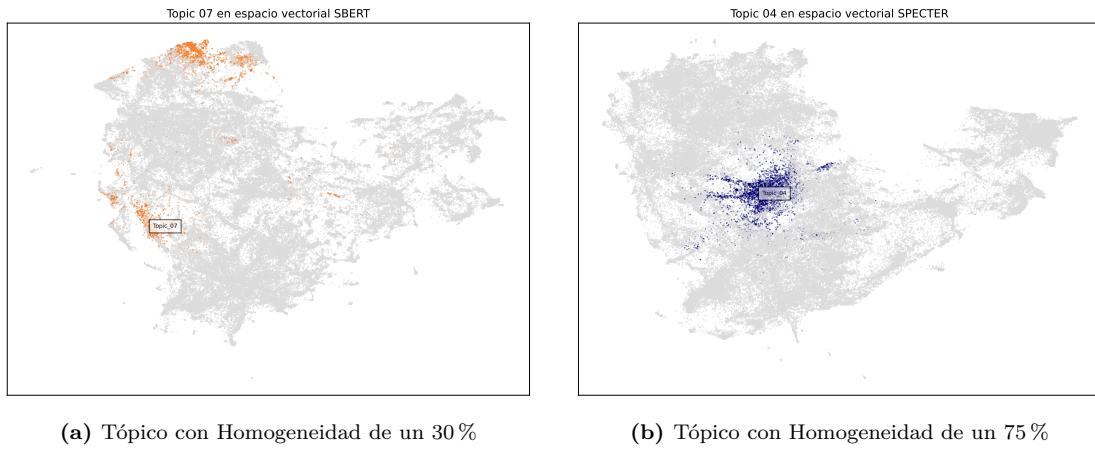


Figura 4.6: Ejemplo de Tópicos con diferentes homogeneidades en un espacio vectorial de 2 dimensiones.

d_i y d_j . En la función de Homogeneidad Doc-Doc, para poder calcular el promedio entre la similitud de los documentos, también tenemos que calcular el número de permutaciones posibles entre los documentos del tópico, este valor está dado por $N^2/2$.

En la *Figura 4.5* podemos observar el resultado de comparar los valores Homogeneidad Topic-Doc y Doc-Doc entre los Modelos de Tópicos SBERT y SPECTER-SBERT. El modelo SPECTER-SBERT supera al modelo SBERT en ambos tipos de homogeneidad. En la Homogeneidad Topic-Doc, SPECTER-SBERT supera como promedio en un 31.4 % al modelo SBERT, lo que demuestra que los tópicos en el modelo SPECTER-SBERT son mucho más representativos de sus documentos. Con respecto a la Homogeneidad Doc-Doc, SPECTER-SBERT es superior al modelo SBERT en un 68.8 % como promedio, mostrando la superioridad de este modelo para encontrar tópicos subyacentes de calidad entre los documentos. La diferencia de calidad entre un tópico con valor bajo de homogeneidad y un tópico con valor alto de homogeneidad también pueden ser observados en las presentaciones 2D de los espacios vectoriales, como podemos ver en la *Figura 4.6*.

Tabla 4.5: Los 20 tópicos encontrados por el Modelo de Tópicos SBERT ordenados por su Homogeneidad Doc-Doc.

ID	Homog. (%)	Vocabulario
Topic 19	50.6	coronavirus-associated, coronavirus-MERS-CoV, SARS-CoV-2-related, SARS-Coronovirus, SARS-CoV-related, coronavirus-229E, coronarvirus-like, coronavirus-ENT, COVID-19-related, respiratory-virus
Topic 05	49.0	coronavirus-associated, coronavirus, coronaviruses-S1, SARS-CoV-2-related, virus-ligand, anti-SARV-CoV-2, SARS-CoV-related, viral-receptor-dependent, virus-receptor, coronavirus-EMC

ID	Homog. (%)	Vocabulario
Topic 11	46.4	respiratory-virus, respiratory-viruse, viral-asthma, respiratory-infected, respiratory-pathogen, rhinovirus-positive, rhinovirus-affected, rhinovirus-associate, enterovirus-rhinovirus, RSV-Rhinovirus
Topic 14	40.6	virus-antibody, antibody-to-virus, CORONAVIRUSES, coronavirus-associated, virus-animal, viruses-porcine, enterovirus-specific, virus-receptor, enteroaphthovirus, nonarboflaviviruse
Topic 13	39.4	epidemic-phase, epidemic-infected, epidemic-control, pandemic-spread, epidemic-curve-like, epidemic-suppression, disease-forecasting, outbreak-severity, endemic-epidemic, sub-epidemic
Topic 03	39.4	RNA-virus, RNA-virus-triggered, viral-protein-associated, HCV-RNA, VP3-RNA, RNAs-transfected, WNV-RNA, HTNV-RNA, RNA-binding, RNA-launched
Topic 20	38.9	influenza-pandemic, pandemic-flu, pandemic-specific, influenza-specific, pandemic-associated, pandemic-threat, intra-pandemic, influenza-like-illness, influenza-virus, influenza-associated
Topic 17	38.1	COVID-19-related, coronavirus-associated, patients-infected, MERS-coronavirus, post-COVID, COVID-SAFER, HCoV-infected, HCoV-19, 2019-ncov-infected, COVID-positive
Topic 02	32.1	respiratory-infection, pneumonia-risk, non-SARS-pneumonia, respiratory-illness, pneumonia-induced, pneumonia-associated, non-pneumonia, pneumonia-in-plan, pneumoniae-positive, pneumonia-relate
Topic 16	32.0	angio-embolization, angioembolization, stent-for-stroke, re-embolization, guidelines-stroke, thrombectomy-assisted, aneurysm-vessel, post-intra-arterial, stroke-specific, post-thrombolytic
Topic 08	31.5	innate-immunity, immune-modulation, cell-immune, innate-immune, immuno-pathogenetic, immune-deficient, neuro-immune, immune-enhance, immune-modulator, immune-driven
Topic 07	29.6	wildlife-pathogen, animal-infected, zoonoses-pathogens, host-pathogen-environment, vector-host-pathogen, pathogen-contaminated, pathogen-transmission, rotavirus-bovine, pathogenicity, cattle-disease

ID	Homog. (%)	Vocabulario
Topic 04	27.2	virus-protein, structure-antiviral, pathogen-targeting, phyto-antiviral, anti-lyssaviral, viral-cell, glycan-protein, pathogen-binding, pathogen-targeted, virus-carbohydrate
Topic 06	27.1	virus-diversity, viral-transcript, bat-vector-virus, mammal-virus, virus-bioinformatic, viral-encoded, RNA-viruses, virus-associated, virus-bacterium, inter-viral
Topic 09	26.4	pathogen-targeting, nanoparticle-vaccinated, immunoinfectomic, specific-pathogen-free, vaccine-vector, pathogenomic, viral-antibody, immuno-assay, DNA-Vaccines, antibody-based
Topic 18	26.1	surgery-specific, laparoscopic-guided, intraoperative-sutured, surgery-use, laparoscopy-guide, post-surgery, re-laparoscopy, surgery-analysis, surgery-expert, laparoscopically-assisted
Topic 12	25.8	emerging-disease, infectious-disease, pandemic-related, ebola-welfare, disease-endemic, disease-prevention, ebola-virus-disease, epidemiological, health-policy, public-health
Topic 15	25.5	pandemic-specific, pandemic-adjusted, pandemic-response, health-care-related, COVID-19-related, patient-related, non-hospital-based, hospitalizations, clinical-oriented, post-hospitalization
Topic 10	23.7	ventilation-to-perfusion, ventilation-perfusion, bronchoscope-assisted, cardio-respiratory, volume-to-respiratory, respiratory-device, ventilator-dependent, non-ventilatory, post-cardiorespiratory, post-intubation
Topic 01	21.6	inflammation-mediated, inflammation-based, pro-inflammation, inflammation-modulating, anti-inflammation, pro-inflammatory, cytokines-induced, cytokine-induced, micro-inflammation, sepsis-mediated

Tabla 4.6: Los 20 tópicos encontrados por el Modelo de Tópicos SPECTER-SBERT ordenados por su Homogeneidad Doc-Doc.

ID	Homog. (%)	Vocabulario
Topic 04	74.9	respiratory-virus, respiratory-viruse, influenza-pneumonia, RSV-influenza, influenza-infection, rhinovirus-positive, rhinovirus-related, rhinovirus-associated, viral-infection, RSV-Rhinovirus

ID	Homog. (%)	Vocabulario
Topic 02	73.8	coronavirus-associated, anti-coronaviral, coronaviruses-S1, SARS-CoV-2-related, coronavirus-229E, coronavirus-EMC, SARS-CoV-related, virus-receptor, coronaviral, anti-viral
Topic 17	70.3	coronavirus, coronavirus-associated, MERS-coronavirus, coronavirus-MERS-CoV, non-pandemic-flu, outbreak-associated, outbreak-related, post-pandemic, infectious-disease, COVID-19
Topic 11	69.8	mammal-virus, species-astrovirus-type, coronavirus-associated, rhinovirus-specific, hantaviruses, influenza-virus, nonarboflavivirus, turdivirus, MERS-coronavirus, criterion-intervirus
Topic 12	69.7	virus-immunity, virus-immune, antibody-to-virus, viral-receptor-dependent, antiviral-signaling, virus-expressed, viral-induce, virus-ligand, reovirus-mediated, viral-intracellular
Topic 15	69.2	pig-infectious-dose, virus-antibody, virus-intestinal, rabbit-infectious, virus-serum-toxin, pathogen-specific, FECV-infected, ebolavirus-specific, zoonotic-epidemicus-infected, rotavirus-bovine
Topic 03	68.1	virus-antibody, pathogen-detection, single-pathogen-test, coronavirus-associated, viral-testing, coronavirus-229E, rNPiBV-ELISA, respiratory-virus, rhinovirus-positive, rPDCoV-N-ELISA
Topic 19	67.5	laparoscopic-surgery, laparoscopic-guided, laparoscopically-assisted, laparoscopy-guide, laparoscopy-experienced, preserving-laparoscopic, re-laparotomy, laparoscopic-converted, laparoscopic-ELAPE, intraoperative-sutured
Topic 06	67.0	viral-protein-associated, proteins-virus, virus-bioinformatic, virus-membrane, viral-derived, antiviral-signaling, protein-pathogen, calicivirus-protein, VPg-proteinase, VP3-RNA
Topic 20	65.5	COVID-19-related, intra-pandemic, pandemic-specific, COVID-activated, non-COVID-related, post-pandemic, COVID-CT, pandemic-adjusted, COVID-suspected, CT-COVID
Topic 13	65.3	coronavirus-associated, COVID-19-related, patients-infected, respiratory-virus, pathogen-therapy, pneumonia-induced, corona-virus-infecte, HBoV-pneumonia, non-pneumonia, pathogen-specific

ID	Homog. (%)	Vocabulario
Topic 16	63.1	angio-embolization, angioembolization, angiography-predicted, stent-for-stroke, neovascularization, stroke-thrombolytic, CT-perfusion, thrombectomy-assisted, sub-angiographic, post-intra-arterial
Topic 09	62.0	cytokine-mediated, inflammation-mediated, innate-immunity-driven, innate-immunity, inflammation-promoting, cytokine-targeted, cytokines-induced, pro-inflammation, interleukin-1a, IL-2-mediated
Topic 05	61.1	epidemiology-infection, influenza-pandemic, health-surveillance, disease-reporting, infection-prevention, disease-prevention, pandemic-threat, epidemiology, inter-pandemic, outbreak-related
Topic 18	58.0	coronavirus, coronavirus-associated, COVID-19, coronavirus-NL63, post-pandemic, intra-pandemic, ebolavirus, SARS-CoV-related, anti-pandemic, SARS-CoV-2-related
Topic 14	57.8	ventilation-to-perfusion, ventilation-perfusion, cardio-respiratory, oxygenation-monitoring, anesthetic-conserving, post-intubation, ventilator-dependent, post-cardiorespiratory, respiratory-device, intubation-induced
Topic 07	56.4	structure-antiviral, phyto-antiviral, nano-antimicrobial, compounds-targets-disease, anti-coronaviral, medicinal-chemistry, virus-inhibitory, anti-plant-viruse, anti-influenza-virus, anti-microbial
Topic 08	56.2	meta-analyses, clinical-trial, systematic-review, symptom-reporting, cohort-study, pharmacovigilance, non-pharmacologic, delirium-prevention, medication-weight, health-related
Topic 01	52.6	pandemic-spread, epidemic-infected, epidemic-spreading, epidemic-systematic, disease-forecasting, epidemic-control, COVID-pandemic, outbreak-severity, multi-outbreak, outbreaks-including
Topic 10	50.6	post-pandemic, pandemic-related, pandemic-driven, COVID-CARE, health-policy, disaster-health-politic, health-threat, health-care-related, public-health, disease-avoidance

4.4. SRI-TOP

El SRI-TOP desarrollado es un Sistema de Recuperación de Información flexible que puede ser aplicado tanto a artículos académicos, como a documentos de temas generales (con una longitud corta, menos de 512 tokens). En la construcción del sistema se debe seleccionar primero el corpus sobre el que se va a trabajar, para luego elegir el Modelo de Tópicos más adecuado para la colección de documentos empleada.

El sistema fue desarrollado y probado empleando el conjunto de datos CORD-19, siendo este un corpus organizado y pre-procesado específicamente para tareas relacionadas con el Procesamiento del Lenguaje Natural, pero el SRI-TOP también se puede utilizar en otras colecciones de documentos sin necesidad de un preprocessamiento.

Si se desea emplear un corpus específico, entonces se debe guardar el texto de los documentos en archivos de textos (.txt) y colocar todos los archivos dentro de una misma carpeta. Luego, la ubicación de este directorio debe ser pasada como parámetro al SRI-TOP, para que este pueda detectar el corpus. El sistema escanea los archivos de texto dentro de la carpeta, tomando el nombre de cada archivo como el identificador de los documentos. Los archivos de texto dentro de la carpeta deben contener solamente el título y resumen de los documentos, siendo el título la primera línea no vacía, y el resumen el resto del texto.

Con el SRI-TOP también se puede especificar los modelos de lenguaje SPECTER y SBERT que se desean emplear en la construcción del Modelo de Tópicos. Por ejemplo, un usuario puede querer utilizar un modelo SPECTER más actualizado o entrenado solamente con artículos sobre Bio-Medicina, o un modelo SBERT multilingüe para usarlo en una colección de documentos en español. Ambos Modelos de Lenguaje deben ser especificados para poder crear el Modelo de Tópicos.

Además, como ya vimos anteriormente, el SRI-TOP tiene dos opciones en cuanto al Modelo de Tópicos empleado, se puede utilizar el modelo SBERT o el modelo SPECTER-SBERT. El modelo SPECTER-SBERT se recomienda para corpora de artículos académicos. Para documentos sobre temas generales (que no sean artículos académicos), se recomienda el modelo SBERT, ya que el modelo de lenguaje SPECTER no fue entrenado para crear representaciones vectoriales empleando este tipo de texto, y con el modelo SBERT se obtienen mejores resultados.

4.4.1. Recuperación de Documentos

Cuando se realiza una consulta en el sistema, SRI-TOP crea una representación vectorial del texto de la consulta utilizando el modelo de lenguaje principal en el modelo de tópicos empleado. Luego, este embedding es usado para encontrar los tópicos más cercanos a la consulta en el espacio vectorial. De esta forma obtenemos los tópicos más relevantes con respecto a la consulta.

Para encontrar los documentos más relevantes, empleamos los documentos dentro de los tópicos más similares a la consulta. El sistema permite especificar cuáles de los tópicos más relevantes pueden ser utilizados durante la búsqueda de los documentos, por defecto solo se emplea el tópico más similar a la consulta. También, es posible emplear todos los documentos dentro del corpus, para extraer de forma absoluta los documentos más relevantes en la colección. Esta opción solo se recomienda cuando no se han obtenido resultados satisfactorios empleando los tópicos relevantes. Con las presentaciones vectoriales de la consulta y los documentos dentro de los tópicos especificados, procedemos a recuperar los documentos más similares a la consulta. Luego, con los resultados de esta recuperación creamos una lista con los documentos más relevantes.

4.5. Interfaz Visual

El SRI-TOP fue construido con una interfaz visual que permite a los usuarios realizar consultas, examinar el contenido de los documentos, y explorar los tópicos encontrados en el corpus por el SRI-TOP. En la aplicación desarrollada para nuestro sistema, se construyeron pestañas especializadas para cada una de las tareas que se pueden realizar en sistema. Las pestañas son *Search*, *Topics* y *Documents*.

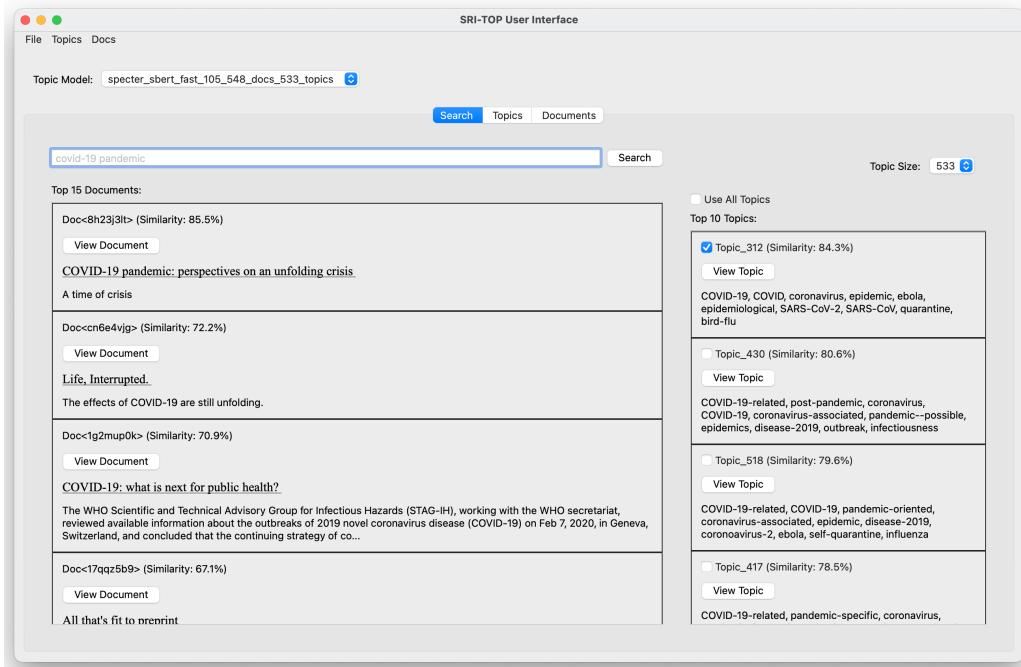


Figura 4.7: Pestaña *Search* de la aplicación visual del SRI-TOP para la realización de consultas.

Antes de realizar una consulta el usuario puede seleccionar el Modelo de Tópicos que desea emplear con el SRI-TOP, teniendo la opción también de controlar el número de tópicos. De esta forma, el usuario puede navegar con más facilidad los tópicos en la colección de documentos, pudiendo seleccionar tópicos más generales o más específicos dependiendo de sus necesidades.

El proceso de creación del Modelo de Tópicos SBERT empleando el corpus CORD-19 tarda alrededor de 2 horas y 30 minutos utilizando una computadora con un microprocesador de 8 núcleos, encontrando 745 tópicos en la colección de documentos. La creación del modelo SPECTER-SBERT tarda 1 hora y 40 minutos, encontrando 533 tópicos. Para reducir el número de tópicos, el modelo SBERT se demora 4 horas y 30 minutos reduciendo su tamaño de 745 tópicos a 2 tópicos; el modelo SPECTER-SBERT tarda 2 horas y 30 minutos en el proceso de reducción de 533 tópicos a 2 tópicos.

Debido al tiempo necesario para la creación de los Modelos Tópicos, y para mejorar la interactividad de la interfaz visual, decidimos utilizar por defecto modelos pre-procesados en las actividades de recuperación de información con el corpus CORD-19. En caso de que el usuario desee utilizar un corpus propio, este lo puede hacer, pero se debe tener en cuenta que el procesamiento de la colección de documentos y la creación del modelo de tópicos puede tardar varias horas dependiendo

del tamaño del corpus.

4.5.1. Búsqueda de Información

Para la realización de consultas en el SRI-TOP construimos la pestaña *Docs* mostrada en la *Figura 4.7*. En esta pestaña el sistema le muestra al usuario los tópicos más similares a la consulta realizada por el usuario, y este, puede seleccionar los tópicos donde desea encontrar los documentos más relevantes. El usuario tiene las opciones de realizar la búsqueda documentos en un tópico, varios tópicos, o todos los tópicos presentes en el corpus. Tanto los tópicos como los documentos en los resultados son ordenados con respecto a su similitud con la consulta.

La búsqueda de los documentos más relevantes se realiza empleando los documentos correspondientes a los tópicos seleccionados. Por defecto, se selecciona el tópico más similar a la consulta.

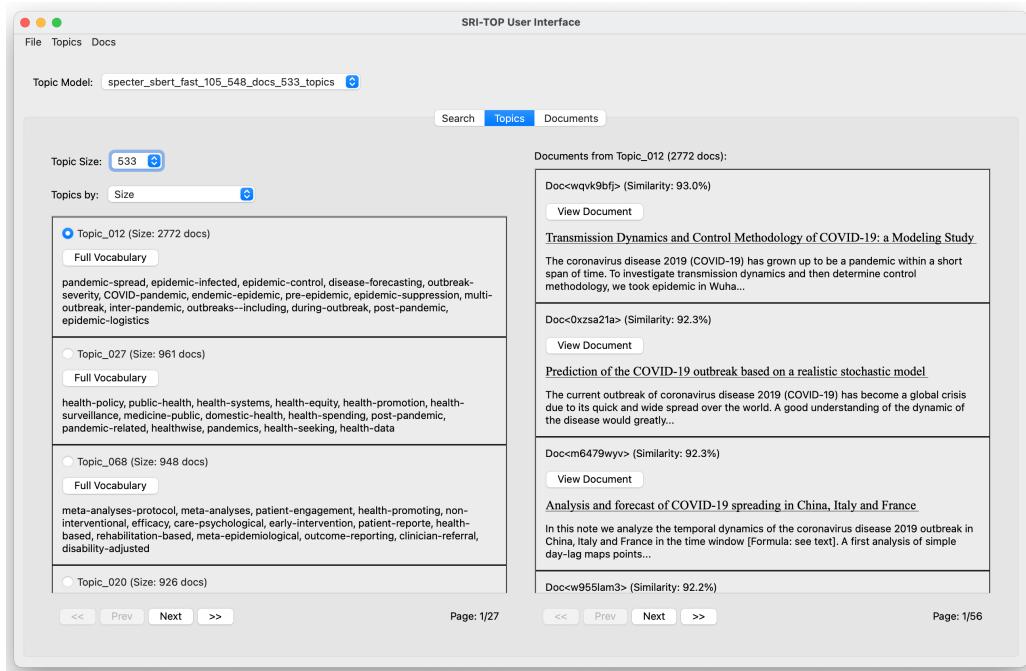


Figura 4.8: Pestaña *Topics* de la aplicación visual para la exploración de los tópicos presentes en el corpus. Los tópicos pueden ser ordenados por su tamaño, homogeneidad o su ganancia de información (PWI).

4.5.2. Explorar Tópicos

El usuario, si lo desea, puede explorar los tópicos dentro de la colección de documentos abriendo la pestaña *Topics*, mostrada en la *Figura 4.8*. Los tópicos pueden ser ordenados por su tamaño, su homogeneidad o su ganancia de información (PWI). Por cada tópico se muestran los documentos que contienen, con los documentos ordenados de más a menos similar.

Otra opción que brinda la pestaña de tópicos es poder ver el vocabulario completo de un tópico. El vocabulario completo de un tópico se puede ver en forma de texto, con la opción de incluir o no

las similitudes de las palabras con el tópico, como se muestra en la *Figura 4.9*.

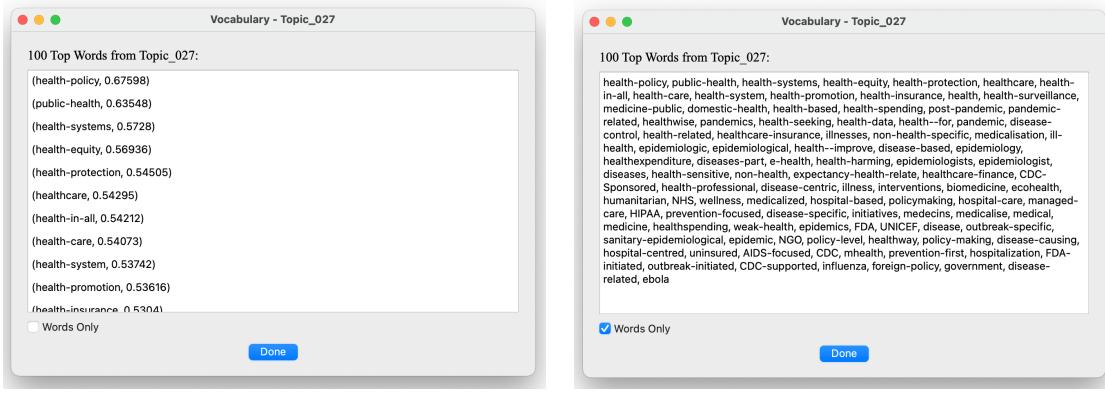


Figura 4.9: Ventana para ver el vocabulario de un tópico en forma de texto. El vocabulario puede ser mostrado con o sin la similitud de las palabras con el tópico.

Las palabras de los tópicos también se pueden mostrar en forma de nube de palabras, en una imagen. Para activar esta opción, se debe activar *Word Clouds* en el menú *Topics*. Las nubes de palabras permiten observar las palabras más importantes dentro de un tópico de una forma más interactiva y dinámica, donde las palabras de mayor tamaño son las más similares al tópico. En la *Figura 4.10* se muestra el ejemplo de una nube de palabras generadas para un tópico.

4.5.3. Explorar Documentos

Una de las características más importantes en un Sistema de Recuperación de Información es, por supuesto, la posibilidad de explorar el contenido de sus documentos. En la pestaña *Documents* es donde podemos observar el título y resumen de los documentos.

En esta pestaña a la izquierda podemos ver los tópicos más similares al documento en el espacio vectorial. A la derecha, se muestran los documentos más similares al documento dentro de los tópicos seleccionados a la izquierda. Están listados de tópicos y documentos relevantes con respecto al documento siendo examinado, permite al usuario expandir fácilmente su búsqueda una vez que se encuentra un documento de interés. En la *Figura 4.11* se muestra la pestaña para acceder al contenido de un documento.

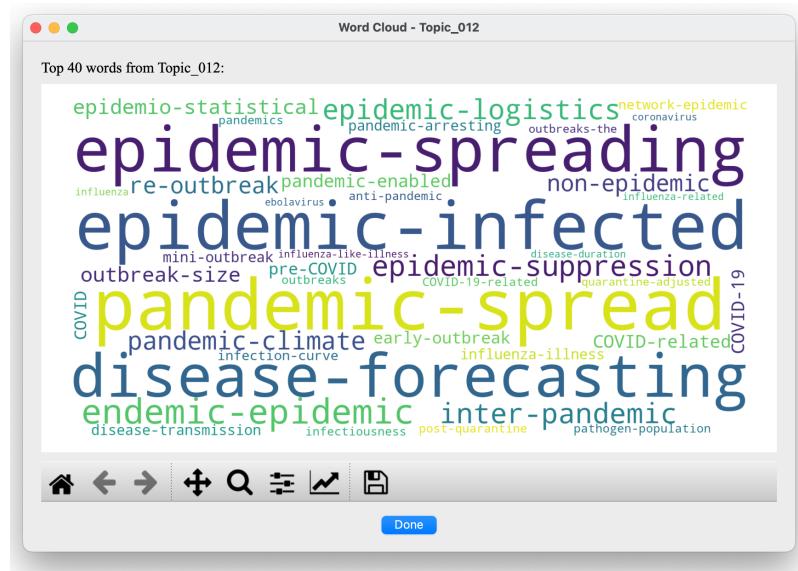


Figura 4.10: Ventana utilizada para ver el vocabulario de un tópico en el formato de nube de palabras. El tamaño de las palabras representa su similitud al tópico.

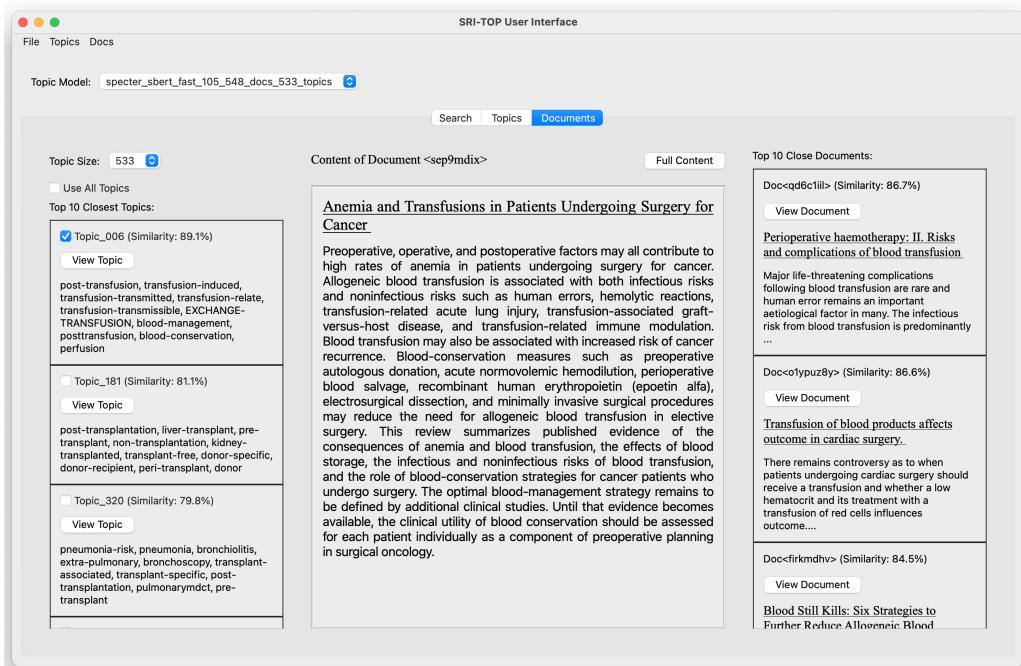


Figura 4.11: Pestaña *Documents* de la aplicación visual para poder examinar el contenido de los documentos del corpus.

Conclusiones

En este trabajo de tesis construimos un Sistema de Recuperación de Información basado en Tópicos (SRI-TOP). Dentro del sistema, empleamos el Modelo de Tópicos Top2Vec con los Modelos de Lenguaje SPECTER y Sentence-BERT. La utilización de un modelo de tópicos en el SRI-TOP permite incluir de forma automática la *agrupación de documentos* y la *pseudo-retroalimentación* para las consultas realizadas por los usuarios, técnicas que mejoran la efectividad y eficiencia del sistema.

Durante el desarrollo del SRI-TOP creamos dos Modelos de Tópicos SBERT y SPECTER-SBERT. Ambos modelos demostraron su superioridad con respecto a modelos anteriores durante su evaluación con *ganancia de tópicos*. Para comparar ambos modelos entre sí empleamos la *homogeneidad de tópicos*, donde el modelo SPECTER-SBERT probó su superioridad en cuanto a la similitud de los embeddings de sus tópicos con respecto a los documentos que representan y la cercanía semántica y temática de los documentos dentro de un mismo tópico.

Para la utilización de nuestro Sistema de Recuperación de Información creamos una interfaz de visual donde los usuarios pueden realizar sus consultas, además de poder explorar de forma separada los tópicos y los documentos dentro del corpus. Esta aplicación permite seleccionar el tipo de Modelo de Tópicos con la cantidad de documentos deseada, cambiar el número de tópicos empleados durante las consultas, ver las descripciones completas de los tópicos y ordenar los tópicos por tamaño, ganancia de tópico u homogeneidad. También, la aplicación hace posible la utilización de corpus nuevos con el SRI-TOP y la actualización de los Modelos de Lenguaje utilizados por los Modelos de Tópicos del sistema.

Trabajos Futuros

Además de organizar los tópicos por su tópico principal, también puede ser de interés obtener los tópicos de los que se hablan dentro del documento, por ejemplo, creando representaciones vectoriales de sus párrafos o secciones. De esta forma, un usuario puede ir directamente al segmento del documento que le resulte de interés sin necesidad de examinar todo el documento.

Otro acercamiento con el que se podría mejorar el SRI-TOP y los Modelos de Tópicos empleados sería la utilización de un modelo de lenguaje Sentence-BERT que utilice como base un modelo BERT entrenado en texto científico como SciBERT. También se puede explorar la representación de los documentos utilizando todo el texto del documento, no solo el título y el resumen, ya existen modelos de lenguaje como **Longformer** que utilizan transformers para representar documentos de una larga longitud [4].

Adicionalmente, hasta ahora solo hemos probado los Modelos de Tópicos en un corpus científico sobre la COVID-19, otros tipos de artículos científicos deben ser explorados para verificar que tan robustos son los resultados de los modelos construidos. Semantic Scholar ofrece un gran número de artículos ya clasificados por la rama de la ciencia en las que fueron publicados [38].

Finalmente, un aspecto que debe ser explorado es la utilización de estos modelos con artículos en Español, pues debido a la poca disponibilidad de documentos en este idioma no se pudo comprobar el comportamiento del sistema y los modelos con artículos científicos en Español.

Referencias

- [1] Akiko Aizawa. An information-theoretic perspective of tf—idf measures. *Information Processing and Management*, 39(1):45–65, January 2003.
- [2] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the literature graph in semantic scholar. In *NAACL*, 2018.
- [3] Dimo Angelov. Top2Vec: Distributed representations of topics. *arXiv*, 2020.
- [4] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.
- [5] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: Pretrained language model for scientific text. In *EMNLP*, 2019.
- [6] Yoshua Bengio, Holger Schwenk, Jean-Sebastien Senecal, Frederic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.
- [7] Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. Content-based citation recommendation. In *NAACL-HLT*, 2018.
- [8] David M. Blei. Probabilistic topic models. *Commun. ACM*, 44(4):77–84, 2012.
- [9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- [10] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv*, 2016.
- [11] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, 2015. Association for Computational Linguistics.
- [12] Jordan L. Boyd-Graber and David M. Blei. Syntactic topic models. In *Advances in neural information processing systems*, pages 185–192, 2009.

- [13] Chris Buckley, James Allan, Gerard Salton, and Amit Singhal. Automatic query expansion using smart: Trec 3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 69–80. NIST Special Publication, April 1995.
- [14] Vannevar Bush. As we may think. *Interactions*, 3(2):35–46, mar 1996.
- [15] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.
- [16] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *arXiv*, 2018.
- [17] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.
- [18] C. W. Cleverdon. The Cranfield tests on index language devices. In *Aslib Proceedings*, volume 19, pages 173–192, 1967.
- [19] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*, 2020.
- [20] Ronan Collobert and Jason Weston. Proceedings of the 25th international conference on machine learning. In *A unified architecture for natural language processing: Deep neural networks with multitask learning*. ACM, 2008.
- [21] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *In Advances in neural information processing systems*, pages 3079–3087, 2015.
- [22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding, 2018.
- [23] Andrea Frome, Greg S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, 2013.
- [24] Hossein Gholamalinezhad and Hossein Khosravi. Pooling methods in deep neural networks, a review, 2020.
- [25] A. Griffiths, H.C. Luckhurst, and P. Willett. Using interdocument similarity in document retrieval systems. *Journal of the American Society for Information Science*, 37:3–11, 1986.
- [26] Thomas L. Griffiths, Joshua B. Tenenbaum, and Mark Steyvers. Topics in semantic representation. *Psychological Review*, 114, 2007.
- [27] D. K. Harman. Overview of the first text retrieval conference (trec-1). In *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 1–20, mar 1993.
- [28] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear (Unpublished), 2017.

- [29] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339. Association for Computational Linguistics, 2018.
- [30] Yacine Jernite, Samuel R. Bowman, and David Sontag. Discourse-based objectives for fast unsupervised sentence representation learning. *CoRR*, 2017.
- [31] K. Sparck Jones. *Automatic Keyword Classification for Information Retrieval*. Butterworths, London, 1971.
- [32] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.
- [33] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Beijing, China, 22–24 Jun 2014. PMLR.
- [34] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2nd edition, 2014.
- [35] Jiawen Li, Shiwen Ni, and Hung Yu Kao. Birds of a feather rumor together? exploring homogeneity and conversation structure in social media for rumor detection. *IEEE Access*, 8:212865–212875, 2020.
- [36] Ling Liu and M. Tamer Özsu, editors. *Encyclopedia of Database Systems*. Springer Reference. Springer, New York, 2009.
- [37] Zhiyuan Liu, Yankai Lin, and Maosong Sun. *Representation Learning for Natural Language Processing*. Springer Singapore, 2020.
- [38] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983. Association for Computational Linguistics, 07 2020.
- [39] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018.
- [40] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.*, 1(4):309–317, oct 1957.
- [41] Robert Wing Pong Luk. Why is information retrieval a scientific discipline? *Foundations of Science*, 27:427–453, 2020.
- [42] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.
- [43] RB Marimont and MB Shapiro. Nearest neighbour searches and the curse of dimensionality. *IMA Journal of Applied Mathematics*, pages 59–70, 1979.

- [44] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov 2017.
- [45] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2017.
- [46] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv*, 2018.
- [47] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 2018.
- [48] Tomas Mikolov. Statistical language models based on neural networks. In *PhD thesis, Brno University of Technology*, 2012.
- [49] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.
- [50] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation, 2013.
- [51] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of phrases and their compositionality. In *Advances on Neural Information Processing Systems*, 2013.
- [52] Tomas Mikolov, Scott Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *NAACL HLT*, 2013.
- [53] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, 08 2019. Association for Computational Linguistics.
- [54] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Technical Report, OpenAI*, 2018.
- [55] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.
- [56] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
- [57] Gerard Salton. *Automatic Information Organization and Retrieval*. McGraw Hill Text, 1968.
- [58] Gerard Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1971.

- [59] Mark Sanderson and W. Bruce Croft. The history of information retrieval research. *Proc. IEEE*, 100(Centennial-Issue):1444–1451, 2012.
- [60] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering, 2015.
- [61] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.
- [62] Jeffrey M. Stanton and Yisi Sang. Assessing topical homogeneity with word embedding and distance matrices. <https://surface.syr.edu/istpub/193>, 2020.
- [63] S. Syed and M. Spruit. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 165–174, 2017.
- [64] Wilson L. Taylor. Cloze procedure: A new tool for measuring readability. *Journalism Bulletin*, pages 415–433, 1953.
- [65] M. Cover Thomas and A. Thomas Joy. *Elements of information theory*. Wiley, New York, 1991.
- [66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [67] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.
- [68] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. CORD-19: The COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Association for Computational Linguistics, jul 2020.
- [69] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018.
- [70] I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. San Francisco: Morgan Kaufmann, 2nd edition, 1999.
- [71] Kai Yang, Yi Cai, Zhenhong Chen, Ho-fung Leung, and Raymond Lau. Exploring topic discriminating power of words in latent dirichlet allocation. In *In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2238–2247, 2016.

- [72] J. Zhang. *Visualization for Information Retrieval*. Springer, New York, 2008.
- [73] A. Zhila, W.T. Yih, C. Meek, G. Zweig, and T. Mikolov. Combining heterogeneous models for measuring relational similarity. In *NAACL HLT*, 2013.
- [74] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.
- [75] Will Zou, Richard Socher, Daniel Cer, and Christopher Manning. Bilingual word embeddings for phrase-based machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2013.