UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Departamento de Lingüística Aplicada
Departament de Lingüística Aplicada

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Lingüística Aplicada

Modelado de temas en documentos de texto: análisis comparativo de LSA, PLSA y LDA

Trabajo Fin de Máster

Máster Universitario en Lenguas y Tecnología

AUTOR/A: Jiang, Linxi

Tutor/a: Periñán Pascual, José Carlos

CURSO ACADÉMICO: 2022/2023

Departamento de Lingüística Aplicada
Departament de Lingüística Aplicada

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

MÁSTER EN LENGUAS Y TECNOLOGÍA

Curso Académico: 2022 / 2023

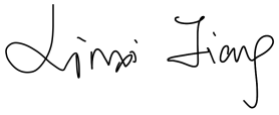
TÍTULO TRABAJO FIN DE MÁSTER:

MODELADO DE TEMAS EN DOCUMENTOS DE TEXTO: ANÁLISIS COMPARATIVO DE LSA, PLSA Y LDA

AUTORA:

LINXI JIANG

Declaro que he redactado el Trabajo de Fin de Máster "Modelado de temas en documentos de texto: análisis comparativo de LSA, PLSA Y LDA" para obtener el título de Máster en Lenguas y Tecnología en el curso académico 2022-2023 de forma autónoma, y con la ayuda de las fuentes consultadas y citadas en la bibliografía (libros, artículos, tesis, etc.). Además, declaro que he indicado claramente la procedencia de todas las partes tomadas de las fuentes mencionadas.

Firmado: 

DIRIGIDO POR:
DR. CARLOS PERIÑÁN PASCUAL

AGRADECIMIENTOS

Este trabajo no hubiera sido posible sin el apoyo de varias partes a las que la autora desea expresar su sincero agradecimiento.

En primer lugar, al Dr. Periñán Pascual por toda su ayuda, experiencia y dedicación en la creación de este trabajo.

Al profesorado del Departamento de Lingüística Aplicada por las enseñanzas, actividades prácticas y comprensión ofrecida en todo momento al alumnado.

Especialmente a mi hija Xuanyan Li por el acompañamiento durante esta etapa académica.

RESUMEN

Este trabajo se centra en los modelos teóricos clásicos más representativos que han marcado el desarrollo del modelado de temas en la minería textual, razón por la cual se ha puesto el foco en el análisis de semántica latente, el análisis probabilístico de semántica latente y la asignación latente de Dirichlet. Siendo una rama de investigación en el ámbito del procesamiento de lenguaje natural, el modelado de temas proporciona una solución automatizada para tareas como la categorización de textos y la elaboración de resúmenes, captando el interés de los investigadores por su capacidad de descubrir estructuras semánticas latentes en los documentos. En este contexto, el estudio aborda principalmente un análisis cuantitativo y cualitativo en dos modelos probabilísticos, i.e. análisis probabilístico de semántica latente y asignación latente de Dirichlet. El objetivo es evaluar y comparar la efectividad de ambos modelos cuando se aplican a corpus de distintos tamaños. Para ello, se crearon tres corpus a partir de títulos de noticias en *Wall Street Journal* y *Nature*. Basándonos en los datos obtenidos, concluimos que PLSA proporciona mejores resultados que LDA en la agrupación de los textos según los temas latentes. Asimismo, se ha notado una mejora considerable en el rendimiento de PLSA a medida que aumenta el tamaño del corpus. Este estudio también analiza algunas cuestiones críticas que pueden afectar a la efectividad de estos modelos.

Palabras clave: Modelado de temas; LSA; PLSA; LDA; Minería de textos.

RESUM

Aquest treball se centra en els models teòrics clàssics més representatius que han marcat el desenvolupament del modelatge de temes en la mineria textual, raó per la qual s'ha posat el focus en l'anàlisi de semàntica latent, l'anàlisi probabilística de semàntica latent i l'assignació latent de Dirichlet. Sent una branca d'investigació en l'àmbit del processament de llenguatge natural, el modelatge de temes proporciona una solució automatitzada per a tasques com la categorització de textos i l'elaboració de resums, captant l'interés dels investigadors per la seua capacitat de descobrir estructures semàntiques latents en els documents. En aquest context, l'estudi aborda principalment una anàlisi quantitativa i qualitativa en dos models probabilístics, anàlisi probabilística de semàntica latent i assignació latent de Dirichlet. L'objectiu és avaluar i comparar l'efectivitat de tots dos models quan s'apliquen a corpus de diferents grandàries. Per a això, es van crear tres corpus a partir de títols de notícies a Wall Street Journal i Nature. Basant-nos en les dades obtingudes, es conclou que PLSA s'exerceix millor que LDA a classificar els textos segons els temes latents. Així mateix, s'ha notat una millora considerable en el rendiment de PLSA a mesura que augmenta la grandària del corpus. El treball també analitza algunes qüestions crítiques que poden afectar l'efectivitat d'aquests models.

Paraules clau: Modelatge de temes; LSA; PLSA; LDA; Mineria de textos.

ABSTRACT

This research focuses on the most representative classical theoretical models that have marked the development of topic modeling in text mining, which are latent semantic analysis, probabilistic latent semantic analysis and latent Dirichlet assignment. As a branch of research in the field of natural language processing, topic modeling provides an automated solution for text mining tasks, such as text categorization and summarization. Thus, it has captured researchers´ interest for the ability in discovering latent semantic structures in documents. In this context, the research mainly addresses a quantitative and qualitative analysis in two probabilistic models, i.e. probabilistic latent semantic analysis and latent Dirichlet assignment. The objective is to evaluate and compare the effectiveness of both models when applied to corpora of different sizes. For this purpose, three corpora were created from subheadings of articles of Wall Street Journal and Nature. Based on the results, we concluded that PLSA performed better than LDA in classifying texts according to the latent topics. Indeed, as the corpus size grew, there was a clear improvement in the performance of PLSA. This research also discusses some critical factors that may affect the effectiveness of these models.

Key words: Topic modeling; LSA; PLSA; LDA; Text mining

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE FÓRMULAS

1 INTRODUCCIÓN

Los datos electrónicos se están generando y acumulando a una velocidad acelerada. Las bases de datos tradicionales almacenan grandes colecciones de información en forma de registros estructurados. La minería de datos, también conocida como el descubrimiento de conocimientos a partir de bases de datos (KDD, del inglés *knowledge discovery in databases*), tiene el objetivo de extraer informaciones implícitas, previamente desconocidas y potencialmente útiles a partir de datos dados (Frawley et al., 1992). El análisis basado en minería de datos se centra en descubrir patrones desde datos extremadamente organizados, que pueden ser procesados por un lenguaje de programación.

Sin embargo, se estima que más del 80% de la información se almacena en formato de texto no estructurado. Teniendo en cuenta la dificultad del paradigma convencional de programación lógica en captar estructuras semánticas latentes en los documentos, se requieren métodos y algoritmos específicos para cumplir esta tarea. La minería de textos, sinónimo del descubrimiento de conocimientos a partir de textos (KDT, del inglés *knowledge discovery in text*), se refiere al proceso de extraer información significativa desde datos textuales (Feldman et al., 1998). Siendo un campo interdisciplinario de investigación, utiliza técnicas de recuperación de información, extracción de información, procesamiento del lenguaje natural y las conecta con los métodos y algoritmos de la minería de datos, el aprendizaje automático y la estadística para realizar análisis textual automatizado.

El modelado de temas es una disciplina de la minería de textos que abarca un conjunto diverso de técnicas. Se define como un tipo de modelo estadístico utilizado para descubrir los temas abstractos que ocurren en una colección de documentos (Blei, 2012). La idea principal del modelado de temas parte de la base de que, los documentos y las palabras son datos observables, mientras que los temas se consideran como las estructuras semánticas latentes en el cuerpo del texto. Las palabras de cada documento proceden de una mezcla de

temas, cada uno de los cuales es una distribución multinomial sobre el vocabulario más representativo. El modelado de temas analiza las palabras, descubre los temas latentes en la colección textual y anota los documentos con informaciones temáticas. Durante este proceso, no interviene ninguno de los recursos de referencia construidos por los humanos, i.e. diccionarios, tesauros u otro tipo de recursos de representación semántica.

Considerado como un potente sistema de análisis textual, el modelado de temas proporciona una solución automatizada para interactuar con los volúmenes masivos de archivos electrónicos. En otras palabras, nos permite comprender, organizar y resumir colecciones de documentos a una escala que sería imposible mediante anotación humana. Asimismo, Blei (2012) afirma que, aparte de los documentos, los algoritmos del modelado de temas pueden adaptarse a muchos tipos de datos. Entre otras aplicaciones, se han utilizado para encontrar padrones en datos genéticos, imágenes y contenidos de redes sociales (Blei y Lafferty, 2006; Zhou et al., 2017; Ostrowski, 2015; Feng y Lapata, 2010).

El análisis de semántica latente (LSA, del inglés *latent semantic analysis*) (Deerwester et al., 1990), el análisis probabilístico de semántica latente (PLSA, del inglés *probabilistic latent semantic analysis*) (Hofmann, 1999) y la asignación latente de Dirichlet (LDA, del inglés *latent Dirichlet allocation*) (Blei et al., 2003) son las técnicas más representativas para el modelado de temas.

Por una parte, LSA proporciona un fundamento teórico a las técnicas posteriores. Es capaz de extraer relaciones semánticas entre palabras a partir de los contextos de uso en los documentos. En concreto, esto es posible a través de la construcción de una matriz término-documento (TDM, del inglés *term-document matrix*) basado en una gran colección de textos, la cual permite descubrir las estructuras de similitud semántica (Dumais, 2004). Es una técnica útil en mejorar la recuperación de información y en resolver otras tareas relacionadas con la minería de textos.

Por otra parte, PLSA, también conocido como la indexación semántica

latente probabilística, principalmente en el ámbito de la recuperación de información (PLSI, del inglés *probabilistic latent semantic indexing*), se propuso como una alternativa a LSA. Hofmann (1999) argumentó que, a pesar de que PLSA está inspirado e influido en gran medida por LSA, tiene un fundamento estadístico más sólido por la integración de un modelo probabilístico. Por tanto, se ha aplicado a una variedad de tareas, p.ej. la categorización de textos, la recuperación de información y el filtrado de información.

Finalmente, LDA se describe como un modelo probabilístico generativo flexible para colecciones de documentos (Blei et al., 2003). Por su buen rendimiento en la reducción dimensional de los datos textuales, se ha convertido en una de las técnicas más populares en la práctica de la minería de textos.

Este trabajo tiene como objetivo evaluar y comparar la efectividad de dos enfoques probabilísticos en el modelado de temas, i.e. PLSA y LDA, al aplicarse a corpus de distintos tamaños. Con este fin, se crearon tres corpus de 200, 800 y 1600 textos recopilados de los subtítulos de los artículos del periódico *Wall Street Journal* y la revista *Nature*. En concreto, los textos se organizan entorno a cuatro temas, i.e. economía, política, ciencia y deportes, donde los textos de cada corpus se distribuyeron equitativamente entre los cuatro temas. A modo de ilustración, y tomando el corpus de tamaño más pequeño como ejemplo, sus 200 textos se organizan en cuatro grupos de 50 textos, donde cada grupo corresponde a uno de los temas. Estos corpus se utilizaron con PLSA y LDA a través de diversos experimentos desarrollados con la herramienta *CSharp Scripting*, la cual forma parte del programa TexMiLAB, el cual a su vez en una versión avanzada del sistema DAMIEN (Periñán-Pascual, 2017). Los dos modelos de nuestro estudio agruparon los textos de los corpus en cuatro clases, i.e. una por cada uno de los temas, además de realizar una distribución multinomial sobre los términos más representativos de cada clase.

Una vez interpretados los temas latentes a partir de los términos más representativos de cada clase, se han comparado los agrupamientos de textos generados automáticamente con las etiquetas que manualmente se asignaron al

recopilar los textos de los tres corpus. En esta fase de evaluación, se introdujeron las métricas intrínsecas (p.ej. perplejidad y coherencia) y métricas extrínsecas (p.ej. precisión, cobertura y valor F1). Por tanto, se analizaron los datos experimentales para comprobar cuál de los modelos había generado mejores resultados y cómo el tamaño de los corpus afectó los resultados generados por los modelos.

A partir de los resultados derivados del presente experimento, se ha concluido que PLSA funciona mejor en agrupar los textos según los temas latentes descubiertos. También se ha observado una mejora considerable en el rendimiento de PLSA a medida que aumenta el tamaño del corpus. En otras palabras, comparado con LDA, PLSA ha mostrado una mayor sensibilidad al tamaño del corpus. Aunque muchos estudios previos (Blei et al., 2003; Wang y McCallum, 2006; Wallach et al., 2009; Hu et al., 2014; Lee et al., 2016) han incidido el éxito del LDA en distintas tareas de la minería textual, es probable que exista una brecha entre la comprensión teórica de los modelos y la aplicación práctica. Por esta razón, tras obtener los datos, se ha realizado un análisis sobre las posibles variables en el experimento que podrían influir en el resultado final.

El resto de este trabajo escrito se estructura de la siguiente forma. La Sección 2 se centra principalmente en la descripción las tres técnicas más importantes del modelado de temas, i.e. LSA, PLSA y LDA. En la Sección 3, se explica el diseño y la implementación del experimento, describiendo con detalle las fases involucradas en el proceso de la investigación. Los datos del experimento se presentan y evalúan en la Sección 4, cuyos resultados se discuten en la Sección 5. Finalmente, la Sección 6 expone las conclusiones y sugiere futuros estudios relacionados con este ámbito de investigación.

2 MARCO TEÓRICO

2.1 Inteligencia artificial y aprendizaje automático

2.1.1 CONTEXTO

Las últimas dos décadas han visto avances cada vez más rápidos en el campo de la inteligencia artificial (IA) y el aprendizaje automático. McCarthy (2007) definió la disciplina de IA como "la ciencia e ingenio de hacer máquinas inteligentes". Sin embargo, décadas antes de esta definición, la prueba de Turing (Turing, 2009) se propuso como un experimento mental para eludir la vaguedad filosófica de la pregunta, "¿puede pensar una máquina?". En la prueba, un interrogador humano trata de distinguir si una respuesta es proporcionada por una máquina o por un humano para examinar si una máquina exhibe una inteligencia parecida a los humanos.

Russell y Norvig (2010) organizaron las diversas definiciones de IA en cuatro categorías: los sistemas que piensan como humanos, los que actúan como humanos, los que piensan racionalmente y los que actúan racionalmente. Plantearon que el procesamiento del lenguaje natural, la representación del conocimiento y el razonamiento, el razonamiento automático, el aprendizaje automático, la visión artificial, el reconocimiento del habla y la robótica son capacidades necesarias para que una máquina pase la prueba de Turing, entre las cuales el aprendizaje automático dota a los ordenadores de la capacidad de adaptarse a nuevas circunstancias y descubrir patrones.

Considerado como una rama de investigación en la IA, el aprendizaje automático se centra en el uso de datos y algoritmos para imitar la forma en que aprenden los humanos. Diferentes sectores se están beneficiando de la tecnología en optimizar sus productos y operaciones. Las aplicaciones más representativas del aprendizaje automático son: el reconocimiento del habla, el motor de recomendación, la atención automática a clientes con *chatbots*, la

visión artificial y el sistema de detección de fraudes. Un ejemplo concreto es *ChatGPT*, el *chatbot* desarrollado por *OpenAI*, que está atrayendo la atención del mundo por su rendimiento notable en proporcionar repuestas detalladas siguiendo una instrucción.

El procesamiento del lenguaje natural (PLN) es un subcampo de la IA y el aprendizaje automático que utiliza una serie de técnicas informáticas para aprender, comprender, analizar y producir contenidos en el lenguaje humano (Hirschberg y Manning, 2015). El carácter de irregularidad y ambigüedad del lenguaje natural hace difícil el estudio de PLN. Las técnicas de PLN y de la minería de textos se utilizan para transformar el abundante volumen de textos en lenguaje natural a un formato estructurado para realizar análisis y obtener conocimientos. La minería de textos, que esencialmente forma parte de la minería de datos, pretende descubrir relaciones ocultas y conceptos clave desde colecciones de materiales textuales. Tras años de esfuerzos por parte de los investigadores, se han conseguido logros en las áreas como la traducción automática, la recuperación de información, la extracción de información, los sistemas de respuestas a preguntas, la elaboración de resúmenes, la categorización de textos, el análisis de sentimiento, la minería de opiniones y la generación de contenidos (Chowdhary, 2020). En este contexto, este estudio ha implementado varios experimentos sobre el modelado de temas con el propósito de identificar las estructuras temáticas en diferentes corpus de documentos.

2.1.1.1 BREVE HISTORIA DE LA INTELIGENCIA ARTIFICIAL

El Proyecto de Investigación de Verano de Dartmouth celebrado en 1956 se considera el acontecimiento que marcó el nacimiento de la IA como una disciplina de investigación (Solomonoff, 1985; Moor, 2006). Desde entonces, generaciones de científicos no han dejado de explorar posibilidades de inventar máquinas que puedan simular la inteligencia humana, de las cuales algunos sistemas enfatizan las conductas y las consideran características externas de la

inteligencia, mientras que otros destacan los procesos cognitivos interiores de los humanos.

Durante los años 60, los investigadores se mostraron muy ilusionados en el desarrollo de sistemas que pudieran ejecutar tareas indicativas de la inteligencia de los humanos. Por esta razón, desarrollaron programas para resolver problemas como juegos, rompecabezas, matemáticas y pruebas de coeficiente intelectual.

En los primeros años de la década de los 70, se observó que los sistemas de la IA no marchaban tan bien como se esperaba, principalmente por dos razones: por un lado, a los ordenadores les faltaba un análisis minucioso de la tarea y una comprensión sobre qué deberían hacer para obtener un algoritmo que fuera capaz de producir soluciones fiables. Por otro lado, los científicos fracasaban al descubrir la insolubilidad de muchos de los problemas que la IA intentaba resolver.

Durante los años 80, el enfoque de investigación en la IA se centró en crear la búsqueda de encadenar pasos elementales de razonamiento para encontrar soluciones completas. No obstante, después de esa etapa, se dieron cuenta de que ese método no se adaptaba a problemas complejos y difíciles. Un enfoque más científico que incorporaba el razonamiento basado en la probabilidad, el aprendizaje automático y los resultados experimentales reemplazó la programación manual. Esta corriente sigue avanzando hasta el presente.

Desde 2001, la disponibilidad de los datos masivos ha permitido diseñar los algoritmos de aprendizaje automático especialmente orientados a aprovechar los grandes volúmenes de datos. A partir de 2011, los estudios en las redes neuronales han proporcionado el fundamento del aprendizaje profundo, que hace referencia a un subcampo del aprendizaje automático que permite a los modelos computacionales compuestos de múltiples capas de procesamiento a aprender representaciones de datos con múltiples niveles de abstracción (LeCun et al., 2015).

En 2017, el programa informático *AlphaGo*, desarrollado por la empresa

DeepMind Technologies, derrotó por primera vez al campeón mundial de *Go*[1] en aquel entonces y se convirtió en el mejor jugador de la historia. Debido a la alta complejidad, el *Go* se ha considerado como uno de los juegos clásicos más desafiantes para IA. Por tanto, el éxito de *AlphaGo* ha captado la atención del mundo sobre el inmenso poder de IA.

2.1.1.2 Relación entre inteligencia artificial y aprendizaje automático

Desde que nació la IA, existen dos opiniones enfrentadas sobre cómo construir una máquina inteligente: una visión, que predominó en el terreno durante varias décadas, se basaba en la programación manual, con la cual una serie de instrucciones se transmiten a ordenadores para que ejecuten una acción o un conjunto de acciones; la otra abogó por el aprendizaje directo a partir de los datos. Sin embargo, debido a la limitada potencia de cálculo de los ordenadores en la década de los 80, la última tardó más tiempo en madurarse (Sejnowski, 2018).

Los paradigmas del aprendizaje automático encajan con la segunda corriente. La IA y el aprendizaje automático son dos conceptos estrechamente relacionados e interconectados. IA se refiere a la habilidad de los sistemas informáticos en imitar las capacidades cognitivas humanas, como el aprendizaje y la resolución de problemas. El aprendizaje automático forma una parte importante de la IA. Es un método computacional que permite a una máquina resolver problemas sin una programación explícita (Samuel, 1959). En el ámbito de la programación, el tipo explícito significa escribir manualmente las instrucciones para que la máquina realice un cambio específico. Por el contrario, en el aprendizaje automático, no es necesario programar específicamente ninguna instrucción para la máquina. En este sentido, el aprendizaje automático comparte similitudes con el aprendizaje humano. Ambos

[1] *Go* es un juego de tablero de estrategia compleja entre dos personas cuyo objetivo es lograr un mayor territorio que el oponente. Los territorios se disputan entre la lucha entre las piedras opuestas.

implican un incremento de conocimientos y competencias para completar una tarea después de aprender de un conjunto de experiencias pasadas.

Gracias a los avances obtenidos en la computación de los ordenadores y los abundantes datos disponibles en la red, se han logrado importantes progresos en el campo del aprendizaje automático. Por una parte, según la Ley de Moore, la velocidad y capacidad de los ordenadores se duplican cada dos años como resultado del aumento del número de transistores que puede contener un microchip (Moore, 1965). Esta ley se ha constatado en el progreso computacional durante la segunda mitad del siglo XX y principios del XXI. En concreto, el número de transistores por chip pasó de 37,5 millones en 2000 a 904 millones en 2009. Como resultado, la velocidad de procesamiento aumentó de 1,3GHz a 2,8 GHz durante ese periodo (Rashid et al., 2016). Por otra parte, la compañía especializada en datos *Statista* afirma que en 2020 se generaron 64,2 zettabytes[2] de datos, y predice que esta cifra aumentará a 180 zettabytes en 2025.

2.1.1.3 Comparación entre aprendizaje automático y aprendizaje humano

El mecanismo del aprendizaje de los humanos y de otros animales se puede describir como el proceso continuo de adaptación a los estímulos procedentes del entorno (Moreno et al., 1994). Los individuos, con la ayuda de las neuronas que están dispersas en el cerebro, son capaces de almacenar y analizar las informaciones que reciben para tomar acciones orientadas por las experiencias pasadas en situaciones semejantes. Los tipos de aprendizaje humano incluyen la habituación, el aprendizaje asociativo, la imitación y la impronta.

Aunque la sinapsis del cerebro humano es un sistema complejo y muy potente en procesar informaciones, especialmente las perceptuales, existe una

[2] Zettabyte es una unidad de almacenamiento de información. Un zettabyte corresponde a 10^{21} bytes.

limitación obvia al enfrentarse con una cantidad masiva de datos. Ahora bien, los ordenadores, apoyados por la tecnología de la computación en nube, pueden rápidamente almacenar, transformar y analizar los datos (Dillon et al., 2010). A partir de esta ventaja, se espera que una máquina pueda aprender de los datos en vez de simplemente seguir las instrucciones establecidas por los humanos. De hecho, es imposible que los desarrolladores predigan todas las situaciones posibles en el futuro ni programar resolución aplicable a cada tarea. Así pues, se introducen los algoritmos del aprendizaje automático.

Sin embargo, esto no quiere decir que el aprendizaje automático de las máquinas es necesariamente superior al aprendizaje humano. Un ejemplo es el cerebro humano, el cual es mucho más avanzado en identificar y distinguir las representaciones visuales. Por el contrario, un ordenador necesita mucho entrenamiento para cumplir la misma tarea, e inevitablemente comete errores. El estudio sobre cómo funcionan las neuronas del cerebro humano lleva las investigaciones al campo del aprendizaje profundo.

Tomando como objetivo el crear sistemas computacionales que se optimicen con acumulación de experiencia y sin la intervención humana, el aprendizaje automático es un campo de convergencia de varias disciplinas. Según Jordan y Mitchell (2015), la disciplina se sitúa en la intersección de la ciencia informática y la estadística. La estadística desempeña un papel sustancial tanto para la comprensión teórica y práctica del aprendizaje automático como para su desarrollo en el futuro. Siendo una ciencia interdisciplinaria, sirve como base no solo para evaluar y analizar los datos, sino también para interpretar los resultados obtenidos. Las métricas para evaluar los algoritmos de aprendizaje automático, p.ej. exactitud, precisión, cobertura, valor F1 y error cuadrático medio, entre otras, tienen su raíz en la estadística (Friedrich et al., 2022). Otras disciplinas que han incluido en el aprendizaje automático incluyen y no se limitan a las siguientes: filosofía, lingüística, psicología, neurobiología y teoría de control (Russell y Norvig, 2010).

2.1.2 TIPOS DE APRENDIZAJE AUTOMÁTICO

Mitchell (1997: 2) definió el aprendizaje automático de forma más precisa y moderna: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"*. A partir de esta definición, es imprescindible confirmar tres aspectos para diseñar un programa del aprendizaje automático: una tarea de aprendizaje, los indicadores de desempeño y la fuente de datos para entrenar el sistema.

Tomemos como ejemplo la categorización de documentos de la minería de textos que se va discutiendo a lo largo de este estudio: la tarea de aprendizaje consiste en agrupar los textos según las estructuras semánticas latentes; el indicador de desempeño se refiere a las métricas extrínsecas (i.e. precisión, cobertura, valor F1) y las métricas intrínsecas (i.e. perplejidad, coherencia); la experiencia para entrenamiento consiste en los corpus que se utilizan con las técnicas del modelado de temas.

Los distintos paradigmas del aprendizaje automático se pueden clasificar de acuerdo con el tipo de selección y transformación de los datos en la fase de entrenamiento (Moreno et al., 1994). El paradigma de aprendizaje inductivo, i.e. el proceso de generalizar lo que se aprende desde un conjunto específico de observaciones a una regla general, es el más estudiado dentro del aprendizaje automático. Según la retroalimentación que acompaña a los datos de entrada, es decir, la ayuda que recibe el sistema de aprendizaje, otro tipo de clasificación consiste en categorizar el aprendizaje automático en aprendizaje supervisado, aprendizaje no supervisado y aprendizaje mediante refuerzos.

2.1.2.1 APRENDIZAJE SUPERVISADO

En el aprendizaje supervisado, un conjunto de entrenamiento que contiene *N* pares de entrada-salida se alimenta al sistema de aprendizaje para que descubra

un modelo que empareja la entrada con la salida (Russell y Norvig, 2010). Por ejemplo, en una tarea de clasificar correos electrónicos como spam o no spam, correos previamente etiquetados son ejemplos proporcionados como entrada. El sistema, a través de un proceso de cotejamiento, tendrá la habilidad de poner una etiqueta apropiada (i.e. spam / no spam) a los correos nuevos. Mediante un entrenamiento continuo con nuevos datos, el sistema va mejorando la precisión de la clasificación.

La clasificación y la regresión son dos tareas comunes dentro del aprendizaje supervisado. Por un lado, se considera una tarea de clasificación si se espera una salida desde un conjunto finito de valores. Los algoritmos se utilizan para asignar con precisión los datos a categorías específicas. Los tipos comunes de algoritmos de clasificación incluyen el clasificador lineal, la máquina de vector soporte, el árbol de decisión y el bosque aleatorio (Caruana et al., 2006). Por otro lado, los modelos de regresión sirven para predecir valores numéricos a partir de distintos datos. Los algoritmos populares de regresión son la regresión lineal, la regresión logística y la regresión polinómica.

2.1.2.2 Aprendizaje no supervisado

En el aprendizaje no supervisado, no tiene lugar un entrenamiento explícito con los datos (Moreno et al., 1994). Es decir, el sistema tiene que trabajar por su cuenta para descubrir la estructura inherente de los datos de entradas sin retroalimentación explícita. Por tanto, la diferencia principal entre el aprendizaje supervisado y el no supervisado consiste en que el último no utiliza datos etiquetados para ayudar a predecir resultados. Las tareas del aprendizaje no supervisado incluyen el agrupamiento (en inglés, *clustering*), la asociación y la reducción de la dimensionalidad de los rasgos, de las cuales el agrupamiento se considera la más común. Es una técnica para agrupar datos en diversas clases en función de las similitudes entre sí. Por ejemplo, a partir de una enorme base de datos de clientes, este tipo de algoritmo puede agrupar los compradores

automáticamente en segmentos, donde los miembros de cada segmento representan necesidades comunes y responden de forma similar a una acción de marketing.

2.1.2.3 APRENDIZAJE MEDIANTE REFUERZOS

El aprendizaje mediante refuerzos está a medio camino entre el supervisado y el no supervisado. A diferencia del aprendizaje supervisado, en el aprendizaje mediante refuerzos no son necesarios los pares de entrada y salida etiquetados. En cambio, se centra en aprender desde una serie de señales de refuerzo, en concreto, recompensas y castigos. De esta forma, se evalúa a sí mismo si ha resuelto correctamente el problema y modifica sus acciones para conseguir más recompensas en el futuro.

2.1.3 LINGÜÍSTICA DE CORPUS

En los últimos años ha crecido enormemente el interés en el ámbito de la creación y el análisis de corpus. Los investigadores en PLN coinciden en que los estudios basados en corpus forman una parte importante en la infraestructura del desarrollo de las aplicaciones avanzadas de PLN (Atkins et al., 1992). La demanda social en los productos de la industria lingüística ha conducido a la aparición del paradigma de la ingeniería lingüística. Esto, a su vez, requiere la disponibilidad de grandes conjuntos de recursos lingüísticos reutilizables para construir, entrenar y evaluar los sistemas de PLN, i.e. corpus escritos y hablados, lexicones y bases de datos terminológicas, entre otros.

Meyer (2002) definió un corpus como una colección de textos o partes de textos sobre los que se puede realizar algún análisis lingüístico general, mientras que Atkins et al. (1992) categorizaron las colecciones de textos en cuatro tipos: primero, un repositorio de archivos, que se refiere a un conjunto de textos en formato electrónico que no están vinculados de ninguna forma coordinada;

segundo, una biblioteca electrónica de textos (ETL, del inglés *electronic text library*), que se considera una colección de textos electrónicos normalizada con ciertas convenciones relativas al contenido, pero sin una restricción rigurosa de selección; tercero, un corpus, que es un subconjunto de ETL que se construye según ciertos criterios de diseño para un objetivo específico; por último, un subcorpus, el cual consiste en una combinación estática o dinámica de varios corpus completos de acuerdo con algún fin.

Indudablemente, los corpus de textos en formato electrónico tienen un papel esencial en los estudios actuales del lenguaje natural. El uso de los corpus informatizados proporciona una sólida base empírica para las descripciones lingüísticas, permitiendo realizar análisis de un alcanza que de otro modo que no sería posible (Biber, 1993). *Brown Corpus* (Francis y Kucera, 1964) es uno de los primeros corpus disponibles en formato electrónico, que contiene un millón de palabras en inglés estadounidense escrito.

La creación de un corpus se puede dividir en varias fases. En la fase de diseño, se debe tener en cuenta los siguientes aspectos: la tipología del corpus que se está construyendo; el tamaño de las muestras textuales que se deben incluir; el idioma, el género de textos y el periodo de tiempo que se va a muestrear; las fuentes de los textos electrónicos que se pueden utilizar de conformidad con el derecho de autor.

En la fase de muestreo de los datos, es imprescindible considerar la representatividad del corpus. Según Leech (1992), un corpus es considerado representativo de la variedad de una lengua si las conclusiones obtenidas de su contenido pueden generalizarse a dicha variedad lingüística. Para los corpus de propósitos generales, el equilibrio y el muestreo son dos criterios para evaluar su representatividad. El equilibro se refiere a la variedad de géneros de los textos y su correspondiente proporción. En el proceso de muestreo, se definen la unidad de la muestra, los límites de la población y la forma de seleccionar las muestras. Para garantizar la representatividad de un corpus, el número de muestras para cada género de texto debe ser proporcional a su peso en la

población.

En la fase de captura de los datos, los materiales impresos se pueden teclear manualmente. Sin embargo, a medida que mejora la tecnología del reconocimiento óptico de caracteres (OCR, del inglés *optical character recognition*), resulta más fácil convertir los textos impresos al formato electrónico mediante dispositivos OCR.

Finalmente, en la fase de procesamiento de los textos, se utilizan una serie de técnicas para obtener informaciones útiles en función del fin específico del análisis lingüístico. Las técnicas generales incluyen la frecuencia de palabras, las concordancias, la lematización, el etiquetado gramatical, el analizador sintáctico, la colocación, la desambiguación lingüística y la vinculación a bases de datos léxicas (Atkins et al.,1992).

2.2 Minería de textos y procesamiento del lenguaje natural

2.2.1 MINERÍA DE TEXTOS

Jo (2018) divide el campo de la minería de datos en cuatro tipos: la minería de datos relacionales, la minería de textos, la minería de contenido Web y la minería de datos masivos. En concreto, la minería de datos relacionales forma la base de la minería de textos. La minería de contenido Web se considera como un área extendida de la minería de textos, y la minería de datos masivos se ha introducido últimamente como un área nueva y desafiante. A continuación, explicamos en qué consiste cada tipo de la minería de datos y sus aplicaciones.

En primer lugar, la minería de datos relacionales se dedica a buscar patrones desde múltiples tablas en una base de datos relacional. Es relativamente fácil la conversión de estos elementos estructurados en vectores numéricos para ser procesados por los algoritmos del aprendizaje automático. Las tareas típicas de la minería de datos relacionales son la clasificación, la regresión y la agrupación. En una tarea de clasificación, se definen de antemano las categorías y se espera

clasificar los registros en una o varias de esas categorías. La regresión se refiere al proceso de calcular un valor de salida a partir de un análisis sobre los valores del registro dado y la agrupación al segmentar un grupo de elementos de datos en varios subgrupos por la similitud entre ellos.

En segundo lugar, la minería de textos tiene como objetivo descubrir conocimientos nuevos y significativos a través de la extracción automática de informaciones a partir de los datos textuales. Hay una diferencia destacada entre la minería de textos y la recuperación de información, que se refiere al proceso de identificar y recuperar informaciones de las bases de datos a partir de las consultas proporcionadas por usuarios o aplicaciones. Por una parte, como los conocimientos descubiertos por la minería de textos son implícitos, es decir, no se almacenan previamente en las bases de datos, se debe distinguir de las informaciones ya existentes que se pueden recuperar de forma directa. Por otra parte, los resultados generados por la minería de textos son valores predichos, mientras que la recuperación de información devuelve valores verdaderos del pasado o de la actualidad. Por ende, hay que tener en cuenta que no existe una certeza perfecta en la minería de textos. Al realizar las tareas de la minería de textos, cuanto más avanzados sean la computación y los algoritmos, mejor resultado producirán.

En tercer lugar, la minería de contenido Web se considera una expansión de la minería de textos ya que los documentos en la Web se amplían a partir de los textos añadiendo los elementos como la dirección URL, las palabras hipervinculadas y los registros de acceso. Así pues, Markov y Larose (2007) la definieron como la aplicación de las técnicas y los modelos de la minería de datos a una variedad de formatos de datos que existe en la Web, p.ej. textos, imágenes, audios y vídeos. En el contexto del procesamiento de los documentos textuales, es similar a la minería de textos. Por ejemplo, la categorización de cada documento Web según su tema es una tarea de clasificación. La elaboración automática de resúmenes de los documentos Web también se considera una tarea de la minería de contenido Web.

Finalmente, la minería de datos masivos recoge los datos de dispositivos como móviles, sensores y cámaras. Por lo tanto, los datos masivos se caracterizan por la variedad de formatos, la velocidad de actualización, la variabilidad (o inconsistencia) y la veracidad de calidad (Wu et al., 2003).

2.2.2 PROCESAMIENTO DEL LENGUAJE NATURAL

El PLN es un campo de investigación que estudia las técnicas computacionales con el objetivo de comprender, manejar y producir los textos y discursos en lenguaje natural (Hirschberg y Manning, 2015). En este sentido, podemos decir que el PLN está relacionado con diferentes teorías y técnicas que tratan el problema del lenguaje natural para realizar las tareas de la minería de textos.

El estudio en PLN empezó en los años 50 del siglo anterior como una disciplina a caballo entre la IA y la lingüística (Nadkarni et al., 2011). Siguiendo la corriente de la primera etapa de desarrollo de la IA, los investigadores en PLN adoptaron un método simplista, por el cual intentaron programar las reglas y los vocabularios de las lenguas naturales para que los ordenadores realizaran tareas como la traducción automática. Sin embargo, el análisis teórico de Chomsky (1956) sobre la estructura lingüística presentó la dificultad de tener gramáticas sencillas que pudieran generar todas las oraciones en inglés teniendo en cuenta las características del lenguaje natural: la variabilidad, la ambigüedad y la interpretación dependiente del contexto.

A partir de los años 80, los métodos del aprendizaje automático basados en la probabilidad empezaron a emplearse en los estudios de PLN. Grandes cantidades de datos empíricos, i.e. corpus, se utilizaban para entrenar los algoritmos de PLN. Los corpus anotados proporcionaban las entradas y salidas correctamente etiquetadas para evaluar la eficacia de los algoritmos. En esta época también nació el PLN estadístico. La estadística, como una disciplina sustancial para la IA, juega un papel significativo en el PLN. Algunos modelos analíticos de corpus emplean el modelo estadístico de N-grama, i.e. el modelo

que asigna probabilidades a frases y secuencias de palabras, mientras que muchos clasificadores de textos y sentimientos parten de la noción de “bolsa de palabras”, i.e. durante el procesamiento de los documentos se ignora el orden de las palabras (Wallach, 2006).

El estudio en PLN, el estudio se enfocaba en crear auxilio para la comunicación humano-humano y humano-máquina. En concreto, son las tecnologías que ayudan a las máquinas a leer y comprender los textos y conversaciones en lenguaje natural, p.ej. la traducción automática, el sistema de reconocimiento y síntesis de voz y el agente conversacional (Hirschberg y Manning, 2015). A medida que los investigadores van profundizando la comprensión en la estructura del lenguaje humano y su uso en los contextos sociales, más las premisas que se han discutido previamente: el aumento enorme en la potencia de cálculo de los ordenadores, la disponibilidad de gran cantidad de los datos lingüísticos en forma digital y el avance en los métodos de aprendizaje automático, se ha ampliado el alcance de estudio en PLN. Por ejemplo, la minería de textos se ha empleado para explorar informaciones útiles, la clasificación e identificación de sentimiento y emoción de los textos en redes sociales sirven para que las empresas tomen decisiones orientadas por datos en sus estrategias de marketing.

Hoy en día, el PLN ha mostrado su ventaja en realizar las tareas relacionadas con el análisis de textos. Algunas de las tareas investigadas en el PLN incluyen: la elaboración automática de resúmenes, la resolución de correferencia, el análisis del discurso, la traducción automática, la segmentación morfológica, el reconocimiento de entidades nombradas, el reconocimiento óptico de caracteres y el etiquetado gramatical.

2.3 Modelado de temas

2.3.1 Introducción al modelado de temas

El modelado de temas es una disciplina importante en la minería de textos. Dependiendo del contexto de aplicación, se utiliza para revelar, descubrir y anotar los conceptos clave, las características destacadas o las variables latentes en las grandes colecciones de documentos (Kherwa y Bansal, 2019). Los textos que forman el corpus pueden provenir de cualquier género, estando digitalizado y almacenado electrónicamente en bases de datos.

Las técnicas del modelado de temas proporcionan una potente herramienta automática para agrupar los materiales textuales en temas, de forma similar a cómo lo harían los humanos. El modelado de temas pertenece a la clase de aprendizaje no supervisado considerando que toma los textos brutos como entrada y no requiere ninguna anotación temática ni etiqueta de documentos en el proceso de entrenamiento. Generalmente, se clasifican estas técnicas en dos principales categorías: los modelos no probabilísticos y los modelos probabilísticos (Kherwa y Bansal, 2019). Por un lado, el enfoque no probabilístico, también conocido como el modelo algebraico, surgió a principios de los años 1990 con la implantación del análisis de semántica latente (LSA, del inglés *latent semantic analysis*) (Deerwester et al., 1990) y la factorización no negativa de matrices (NMF, del inglés *non-negative matrix factorization*) (Paatero y Tapper, 1994). Ambos se basan en el método de bolsa de palabras (Zhang et al., 2010; Zhao y Mao, 2017), donde el corpus es transformado en una TDM. En este marco, solo importa la frecuencia de las palabras y se ignora totalmente su orden. Por otro lado, los modelos probabilísticos se propusieron para mejorar los no probabilísticos adoptando un enfoque generativo. En esta categoría, los modelos más representativos son el análisis probabilístico de semántica latente (PLSA, del inglés *probabilistic latent semantic analysis*) (Hofmann, 1999) y la asignación latente de Dirichlet (LDA, del inglés *latent Dirichlet allocation*) (Blei et al., 2003).

Este estudio se enfoca en las tres técnicas que han aportado la mayor contribución al desarrollo del modelado de temas, i.e. LSA, PLSA y LDA. Las técnicas parten del fundamento de que, existe una jerarquía de tres niveles en cualquier texto, i.e. palabra, documento y tema, de los cuales las palabras y los documentos constituyen los datos observables, mientras que los temas se consideran una estructura semántica latente, es decir, una variable intuitivamente imperceptible que se oculta en los documentos. Cada documento se trata como una mezcla de temas y cada tema se puede representar con una distribución multinomial de palabras. En concreto, la distribución multinomial se refiere a una distribución de probabilidad que describe el resultado de un experimento en que tienen múltiples categorías posibles. Es una generalización de la distribución binomial, que solo tiene dos resultados posibles. En el contexto del modelado de temas, el número de categorías se especifica de antemano, así como las probabilidades asociadas a cada categoría. Por tanto, las estructuras semánticas latentes descubiertas por el modelado de tema se representan mediante una distribución multinomial sobre un vocabulario.

El modelado de temas ha demostrado su efectividad en numerosas investigaciones. Steyvers y Griffiths (2007) afirma que el modelado de temas ha proporcionado contribuciones importantes al análisis estadístico de los corpus grandes y ha profundizado en la comprensión sobre el aprendizaje y el procesamiento del lenguaje natural. Mediante un análisis sobre las palabras de los documentos originales, las técnicas del modelado de temas son capaces de revelar los temas que recorren los textos, cómo se conectan entre sí y cómo cambian a lo largo del tiempo (Blei, 2012).

Las aplicaciones del modelado de temas son variadas. Primero, es ampliamente utilizado en explorar las investigaciones científicas mediante categorización de los temas y la generalización automática de resúmenes. Además, puede analizar las tendencias de investigación en los trabajos publicados e identificar las cuestiones candentes. Segundo, comparado con los métodos tradicionales de agrupar o clasificar las informaciones bioinformáticas,

se ha comprobado que el modelado de temas ayuda a los profesionales en interpretar estas informaciones (Liu et al., 2016; Heo el al., 2017). Por último, se puede aplicar para analizar los contenidos en las redes sociales, p.ej. la detección de noticias de última hora, la recomendación personalizada de mensajes y el análisis de sentimientos (Hong y Davison, 2010).

Sin embargo, a pesar de los usos extendidos del modelado de temas, es susceptible a los problemas de inestabilidad, optimización y sensibilidad al ruido. Un ejemplo notable es el “efecto del orden”. Los modelos pueden generar temas diferentes cuando se modifica el orden de los datos de entrenamiento y como consecuencia, se introduce un error sistemático en el análisis (Agrawal et al. 2018).

En este trabajo se utiliza “palabra” , “término” y “N-grama” para representar la unidad fundamental de los datos textuales; un “documento” o un “texto” es una cadena compuesta por palabras y un “corpus” se refiere a una colección formada por documentos que abarcan todo el conjunto de datos.

2.3.2 Análisis de semántica latente

LSA es una de las técnicas más básicas del modelado de temas. Es un método estadístico basado en corpus para inducir y representar vectorialmente la asociación semántica de las palabras y los textos reflejada en sus usos (Berry et al., 1995).

2.3.2.1 Modelo de espacio vectorial

LSA tiene su raíz en el concepto del modelado de espacio vectorial (VSM, del inglés *vectorial space model*), que al principio se empleó para abordar el límite de capacidad de los ordenadores en entender el lenguaje natural. VSM fue el primer modelo algebraico basado en la TDM para extraer información semántica del uso de las palabras (Salton et al., 1975).

Un vector se refiere a un objeto geométrico que tiene magnitud y dirección. Un espacio vectorial es un conjunto de vectores que pueden sumarse o multiplicarse por números llamados escalares. La idea básica de VSM es que un documento se puede convertir en un vector y así una colección de documentos forma un espacio de vectores. En otras palabras, cada documento de un corpus se representa como un punto en un espacio (i.e. un vector en un espacio vectorial). Los puntos cercanos en este espacio son semánticamente similares mientras que los alejados son semánticamente distantes (Turney y Pantel, 2010).

En sus inicios, VSM se desarrolló para el sistema de recuperación de información SMART (del inglés *System for the Mechanical Analysis and Retrieval of Text*) (Salton, 1971). Como funcionaba bien en las tareas que implican la medición de la similitud semántica entre palabras, frases y documentos, se utilizó ampliamente en los buscadores modernos para cuantificar la similitud entre una consulta y un documento (Manning et al., 2008).

A modo de ilustración, se supone que existe un espacio vectorial compuesto por documentos D_i , y cada documento es identificado por un vector de características T_j (Mitchell, 1997; Witten et al., 2005). Los vectores de características se consideran como las unidades lingüísticas indivisibles más pequeñas en VSM: en este caso, forman una serie de términos indexados. Por tanto, un documento se representa a través de un conjunto de términos indexados:

$$d_i = (t_1, t_2, t_3 \dots, t_n),\ 1 \leq j\ \leq n$$

Los términos indexados se pueden ponderar según su importancia en el documento, lo cual se marca como el peso de los términos W_j. De esta forma, un documento puede representarse mediante los términos indexados y sus pesos correspondientes:

$$d_i = (t_1, w_1;\ t_2, w_2;\ t_3, w_3; \dots;\ t_n, w_n),\ 1 \leq j \leq n$$

Se simplifica como $d_i = (w_1, w_2, w_3, \dots, w_n)$, donde w_n equivale a la ponderación de t_n. El documento d_i satisface las condiciones de que no haya duplicados ni una relación secuencial entre los términos indexados. En este sentido, la colección D_i configura un espacio vectorial de n dimensiones donde cada documento es representado por un vector cuyo valor de coordenadas corresponde a las ponderaciones de los términos. Se ilustra un ejemplo de tres dimensiones con la Figura 1.

En realidad, los ordenadores se muestran incapaces de comprender el significado del lenguaje humano. VSM proporciona la posibilidad de analizar las informaciones semánticas a través de las frecuencias de palabras de un documento en un corpus (Turney y Pantel, 2010). La TDM (Salton et al., 1975), la matriz término-contexto (Deerwester et al., 1990), la matriz par-patrón (Lin y Pantel, 2001) son los tres tipos de matriz más usados en VSM. En LSA se utiliza la TDM.

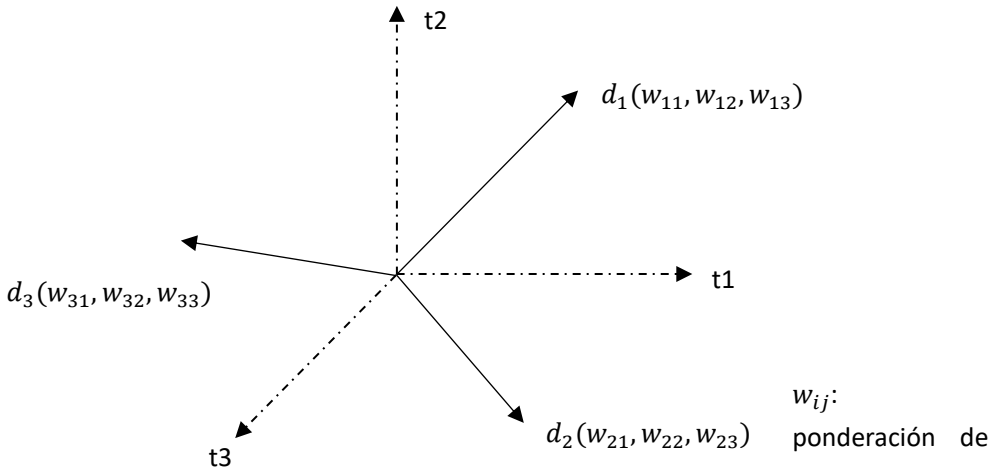


Figura 1. Representación vectorial de documentos

2.3.2.2 MATRIZ TÉRMINO-DOCUMENTO Y PUNTUACIÓN DE TF-IDF

En el marco de VSM y LSA, la información semántica puede derivarse de una

TDM. Según Steyvers y Griffiths (2007), la reducción de la dimensionalidad forma una parte esencial en el proceso de la derivación de información semántica. Para ello, los documentos se representan como vectores en el espacio euclídeo.

En la TDM, las filas representan los términos, donde se excluyen los *stopwords*, que se refieren a las palabras que llevan poca información temática (p.ej. *the*, *and*, *a*, *is* en inglés). Las columnas corresponden a los documentos. A modo de ilustración, a partir de un corpus que contiene *M* documentos y *N* términos, se puede crear una TDM de *M x N* donde cada documento es un vector de *n* dimensiones (Andrews y Fox, 2007). La matriz sirve para contar las veces de apariciones de los términos en los documentos. Cada celda corresponde a la frecuencia de un término en un cierto documento. Tomando como ejemplo un corpus con cuatro documentos, se ilustra el uso de la TDM en la Tabla 1:

Documento 1: “the big brown dog”
Documento 2: “the lazy yellow cat”
Documento 3: “the big write dog”
Documento 4: “the small yellow cat”

	Doc 1	Doc 2	Doc 3	Doc 4
big	1	0	1	0
brown	1	0	0	0
dog	1	0	1	0
cat	0	1	0	1
yellow	0	1	0	1
write	0	0	1	0
small	0	0	0	1
lazy	0	1	0	0

Tabla 1. Matriz término-documento a partir de los documentos de la muestra

La TDM es simple e intuitiva. Sin embargo, siendo una matriz dispersa de alta dimensión que incluye más valores de cero que de no cero, la computación es poco eficiente al procesar enormes cantidades de datos. Aparte, La TDM

ignora el hecho de que no todas las palabras de un documento tienen la misma importancia en la estructura semántica.

Una alternativa al simple recuento de palabras es la puntuación TF-IDF (del inglés *term frecuency-inverse document frecuency*). TF-IDF es una medición estadística utilizada para calcular la relevancia de una palabra para un documento en un corpus. Se calcula según la Fórmula 1:

$$W_{ij} = TF_{ij} \times \log\left(\frac{M}{DF_i}\right)$$

Fórmula 1. Puntuación TF-IDF

donde,

W_{ij}= peso del término *i* en el documento *j*;

TF_{ij}= número de ocurrencias del término *i* en el documento *j*;

M= número de documentos;

DF_i= número de documentos que contienen el término *i*.

La primera parte de la puntuación TF (frecuencia de término) calcula el número de veces que un término aparece en un documento. Cuanto más se use un término, más relevante será para definir el tema general de un documento. El peso de un término que aparece en un documento es simplemente proporcional a la frecuencia del término (Luhn, 1957). La segunda parte IDF (frecuencia inversa de los documentos) se introduce en el cálculo para cuantificar la intuición de que un término tendrá un peso más elevado cuando aparece con frecuencia en el documento, pero con poca frecuencia en el corpus.

Como consecuencia de la compensación, se reducirá el peso de los términos que aparecen muy a menudo en el corpus y se aumentará el peso de los más raros, que son más importantes para representar el tema del documento. Por ejemplo, la palabra “resolver” podría aparecer repetidamente en un documento, pero como es común en el resto del corpus, no tendrá una puntuación TF-IDF

alta. Sin embargo, si la palabra “álgebra” aparece a menudo en un documento y es rara en otros textos del corpus, tendrá una puntuación TF-IDF más alta. Intuitivamente, es muy probable que este documento trate de matemáticas. El esquema de TF-IDF consigue reducir los documentos a vectores de números reales y positivos y el corpus a una matriz de *M* documentos por *N* palabras.

2.3.2.3 DESCOMPOSICIÓN EN VALORES SINGULARES

Como la TDM es una matriz dispersa, LSA realiza la reducción de dimensionalidad partiendo de la idea de descomponerla en dos matrices: una matriz documento-tema y una matriz tema-término. En términos generales, LSA, PLSA y LDA difieren en cómo definen y consiguen el objetivo de la reducción de la dimensionalidad.

La técnica específica para realizar la descomposición en LSA es la descomposición en valores singulares (SVD, del inglés *singular value decomposition*) (Berry, 1992), que consiste en descomponer una gran matriz por documentos en un conjunto de factores ortogonales a partir de los cuales se puede aproximar la original TDM mediante una combinación lineal (Deerwester et al., 1990). SVD, como una técnica popular para la reducción de la dimensionalidad en el aprendizaje automático, se deriva del álgebra lineal y se utiliza en la fase de preparación de los datos. Sirve para crear una proyección de un conjunto de datos dispersos antes de ajustarse a modelo. En SVD, una matriz rectangular se factoriza en el producto de tres diferentes matrices con la Fórmula 2:

$$TDM = U \times \Sigma \times V^{t}.$$

Fórmula 2. Factorización de TDM mediante SVD

En esta fórmula, la TDM original se descompone en un conjunto de factores

ortogonales, a partir de los cuales la matriz original se puede aproximar mediante una combinación lineal (Torres López y Arco García, 2016) U se interpreta como la matriz documento-tema, en la cual las filas corresponden a los documentos y las columnas, a los temas latentes. V es la matriz término-tema, donde los términos y los temas se registran en las filas y las columnas, respectivamente. Σ forma la matriz diagonal tema-tema que contiene los valores de escalado.

En el proceso de SVD truncada, ilustrado en la Figura 2, las celdas en la matriz original se representan como una combinación lineal de rotación y estiramiento. SVD reduce la dimensionalidad a través de simplemente seleccionar los *t* valores más grandes y conservar las primeras *t* columnas de matrices U y V. En este caso, *t* es un hiperparámetro (también se denomina “el ritmo de aprendizaje” o coeficiente utilizado para controlar el proceso de aprendizaje automático) que sigue ajustándose para reflejar el número de temas latentes que se intenta encontrar. Se consigue la reducción de la dimensionalidad mediante la eliminación de los coeficientes de la matriz diagonal Σ, normalmente empezando por el más pequeño. LSA realiza la inducción de temas latentes basado en la idea matemática de que, si se cambia la entrada en cualquier celda original, los valores en las matrices reconstruidas podrán cambiar (Landauer et al., 1998).

Se define que la matriz diagonal Σ contiene los valores singulares de *N* con el resto de las celdas en ceros. LSA computa la aproximación de *N* mediante umbrales de cero en todos los valores singulares en la matriz Σ excepto los más grandes de *K*. Se conseguirá el rango *K* óptimo en el sentido de la norma de L_2 o la norma de Frobenius (Hofmann, 1999).

En resumen, LSA utiliza un modelo de espacio reducido de vectores para explorar la estructura semántica latente de los documentos basado en una serie de sofisticados algoritmos numéricos (Martin y Berry, 2007).

TDM (m x n)

Matriz Término-Documento

	Doc 1	Doc 2	Doc 3	Doc 4
Term 1				
Term 2				
Term 3				
Term 4				

U (n x t)

Matriz Documento-Tema

	Tema 1	Tema 2
Doc 1		
Doc 2		
Doc 3		
Doc 4		

Σ (t x t)

Matriz Tema-Tema

	Tema 1	Tema 2
Tema 1		
Tema 2		

V (m x t)

Matriz Término-Tema

	Tema 1	Tema 2
Term 1		
Term 2		
Term 3		
Term 4		

m= número de términos
n= número de documentos
t= número de temas

Figura 2. SVD truncada en matriz término-documento para la extracción de temas latentes

2.3.2.4 RESUMEN ANALÍTICO SOBRE LSA

En el modelo de LSA, los documentos se convierten en una matriz término-documento de alta dimensión en la que cada celda corresponde a la frecuencia de un término en un determinado documento. Tras una ponderación TF-IDF en las entradas de las celdas, se aplica la descomposición de valores singulares a la matriz (i.e. proceso de proyección lineal reductora de dimensión) y se construye un espacio semántico donde los términos y los documentos originales se representan como vectores. La representación de LSA está independiente del orden de los términos (Landauer y Dumais, 1997).

LSA se considera un modelo eficaz en la indexación automática y la

recuperación de información. Se basa en la afirmación de que los documentos que comparten más términos frecuentes tendrán una representación similar en el espacio semántico latente. Los temas extraídos de la TDM permiten una interpretación humana. En muchas aplicaciones, LSA ha demostrado una capacidad robusta en computar la similitud semántica entre textos.

En el ámbito de la educación, se han desarrollado muchas herramientas basadas en LSA. Se ha aplicado para predecir la comprensión, evaluar las estrategias generales de lectura e identificar las específicas del alumnado mediante un análisis sobre los protocolos verbales (Millis et al., 2007). McNamara et al. (2007) realizaron un estudio comparativo entre LSA y los algoritmos basados en palabras para evaluar la calidad de autoexplicación en iSTART (del inglés *interactive strategy training for active reading and thinking*).[3] El estudio concluyó que los sistemas que incorporaron LSA clasificaron con mayor precisión los tipos de autoexplicación que los basados en palabras. Graesser et al. (2004) demostraron que LSA puede obtener resultados satisfactorios como un sistema fundamental de representación para AutoTutor.[4]

Con la popularidad de la educación a distancia, se requieren aplicaciones que sean capaces de moderar y evaluar los grupos de discusión en línea. LSA hace posible visualizar a tiempo real los resultados de evaluación tras monitorizar los aspectos de la discusión. Al aplicar LSA a grandes colecciones de documentos, se consigue vincular los recursos electrónicos relevantes con las discusiones (Streeter et al., 2007). Además, se ha demostrado que las tecnologías basadas en LSA pueden mejorar la eficacia del aprendizaje gracias a sus características en optimizar la búsqueda entre distintas secciones, proporcionar enlaces instantáneos acerca de temas relevantes y automatizar la elaboración de resúmenes de los materiales (Foltz y Landauer, 2007).

[3] iSTART es una aplicación que proporciona el entrenamiento de autoexplicación y las estrategias de lectura.

[4] AutoTutor es un tutor informático que mantiene conversaciones en lenguaje natural con los aprendices.

LSA se introdujo por Dumais et al. (1988) como una técnica para mejorar el desempeño en la recuperación de información. En comparación con la técnica tradicional de concordancia de palabras, LSA tiene la ventaja de que los documentos pueden ser recuperados cuando no coinciden con ninguna palabra de la consulta siempre que tengan una similitud semántica entre sí. Muchas tareas vinculadas con la recuperación de información (p.ej. la clasificación y agrupación de textos) pueden abordarse mediante LSA, especialmente cuando se tienen entradas ruidosas (Dumais, 2007).

A pesar de los éxitos que obtienen LSA en distintos ámbitos, muestra carencias que principalmente se deben a su insatisfactorio fundamento estadístico (Hofmann, 1999). Por un lado, se observa un uso limitado de las informaciones estadísticas extraídas del espacio de LSA y de las informaciones dimensionales en la representación vectorial de los términos en el proceso de análisis (Hu et al., 2007). Por otro lado, la complejidad de computación de la descomposición de valores singulares puede conducir a una representación menos eficiente. Como es imposible tener un conocimiento previo de los temas, los componentes en las matrices documento-tema y término-tema pueden ser arbitrariamente positivos o negativos, lo cual planteará un problema de interpretabilidad. Finalmente, LSA muestra un punto débil en su incapacidad de dar un tratamiento adecuado a la polisemia de las palabras. La representación vectorial puede acabar siendo un promedio de todos los significados de la palabra en el corpus de muestra, lo cual dificultará la comparación entre documentos.

2.3.3 Análisis semántico latente probabilístico

LSA logra comprimir de manera significativa los datos textuales cuando se aplica a grandes corpus. Sin embargo, un método más directo y eficiente puede ser la introducción de un modelo generativo y un modelo probabilístico. Hofmann (1999) propuso PLSA como una versión probabilística de LSA (Blei, 2012). PLSA define

un modelo generativo para los datos textuales y se ajusta mediante un método de verosimilitud. Desde entonces, el modelado de temas se ha construido sobre la idea de que en un documento se exponen múltiples temas y un tema es una distribución de probabilidad sobre las palabras (Steyvers y Griffiths, 2007). Por consiguiente, comparado con LSA, se cree que PLSA tiene una base estadística más sólida por la integración de un modelo probabilístico.

2.3.3.1 MODELO GENERATIVO

Un modelo generativo se refiere a un modelo que genera datos observables al azar, típicamente dadas algunas variables latentes. Un modelo generativo de textos se describe como un proceso estadístico por el que el modelo supone cómo son generados los documentos (Steyvers y Griffiths, 2007). Un tema, como una variable latente, se define como una distribución probabilística sobre un vocabulario. Al generar un documento, se supone que los temas se especifican de antemano, y sobre los cuales se selecciona una distribución. A continuación, para cada palabra del documento, se elige un tema al azar a partir de esa distribución y se extrae una palabra de ese tema. Como explica Blei (2012):

Paso #1: Seleccionar al azar una distribución sobre temas.
Paso #2: Para generar cada palabra del documento:

a. Seleccionar al azar un tema de la distribución sobre los temas del Paso #1;
b. Seleccionar al azar una palabra de la correspondiente distribución sobre el vocabulario.

El descubrimiento automático de temas dentro de una colección de documentos que hace el modelado de temas puede considerarse como un proceso inverso del generativo. Las técnicas estadísticas estandarizadas se emplean para inferir un conjunto de temas latentes que producen los documentos.

Se supone que el modelo generativo ha generado los datos observados, es decir, las palabras contenidas en los documentos. El objetivo es inferir qué modelo sería más probable que haya generado estos datos. El proceso de inferencia estadística implica encontrar la distribución de probabilidad sobre palabras asociadas con cada tema, en otras palabras, la estructura semántica latente dentro de los documentos (Steyvers y Griffiths, 2007).

2.3.3.2 MODELO DE ASPECTO

PLSA y LDA son las técnicas más representativas que se incorporan el modelo probabilístico en el modelado de temas para analizar el contenido de los documentos y las palabras (Hofmann, 1999; Blei et al., 2003). La diferencia principal entre las dos consiste en los distintos supuestos estadísticos en los que se basan.

En PLSA se introduce el modelo de aspecto (Hofmann y Puzicha, 1999), que es un modelo de variable latente donde los datos de coocurrencia (i.e. la presencia de una determinada palabra junto a otras en un documento concreto) asocian a una variable no observada (i.e. un tema latente). La notación se escribe del siguiente modo:

Variable de documento: $d \in D = \{d_1, d_2, \dots, d_i\}$;
Variable de palabra: $w \in W = \{w_1, w_2, \dots, w_j\}$;
Variable latente de tema: $z \in Z = \{z_1, z_2, \dots, z_k\}$;
Distribución probabilística sobre documentos: $P(d)$;
Distribución probabilística sobre temas dado el documento d : $P(z|d)$;
Distribución probabilística sobre palabras dado el tema z: $P(w|z)$.

PLSA modela la distribución conjunta de probabilidad $P(d,w)$ tomando el par (d,w) como una mezcla de distribuciones multinomiales condicionalmente independientes. El proceso generativo de los textos se puede interpretar con la

Fórmula 3:

$$P(d,w) = P(d)\,P(w|d),$$

$$P(w|d) = \sum_{i=1}^{k} P(w|z)P(z|d).$$

Fórmula 3. Proceso generativo de textos en PLSA

Desde las dos ecuaciones se puede conseguir la probabilidad de cada palabra de un documento suponiendo que, dado un tema, la distribución de palabras es condicionalmente independiente del documento. Es decir, $P(w|z,d) = P(w|z)$:

$$P(d,w) = P(d)\sum_{i=1}^{k} P(w|z)P(z|d)$$

Fórmula 4. Probabilidades de cada palabra en un documento

La distribución conjunta se puede utilizar para calcular la distribución condicional (también llamada la distribución posterior) de las variables latentes dadas las observadas. En el modelo de aspecto se supone que d, w están condicionados de forma independiente dada una variable latente. A través de una transformación de la representación asimétrica gráfica del modelo (Figura 3: a) a la simétrica (Figura 3: b), se obtiene una parametrización (Fórmula 5) equivalente de la probabilidad conjunta (Fórmula 3):

$$P(d,w) = \sum_{i=1}^{k} P(z)P(d|z)P(w|z)$$

Fórmula 5. Parametrización equivalente a la probabilidad conjunta

Desde esta expresión se puede establecer una analogía con la fórmula de

descomposición de valores singulares en la TDM: $TDM = U \times \Sigma \times V^t$, donde $P(d,w)$ corresponde a la TDM; $P(z)$ es análoga a la matriz diagonal Σ; y $P(d|z), P(w|z)$ corresponden a U y V, respectivamente.

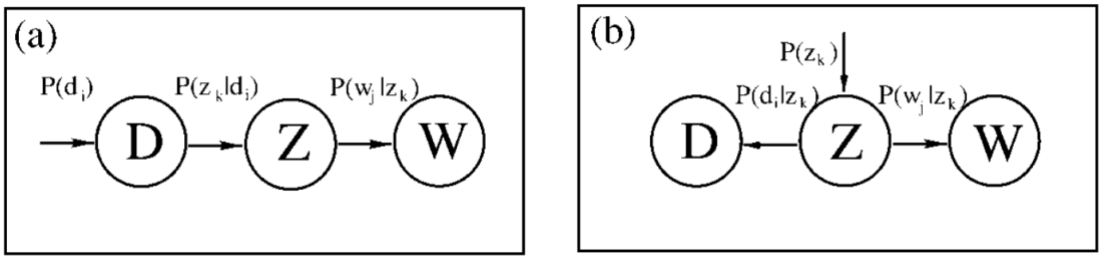


Figura 3. Representación gráfica del modelo de aspecto en la parametrización asimétrica (a) y simétrica (b) (Hofmann, 2001)

2.3.3.3 ALGORITMO DE MAXIMIZACIÓN DE EXPECTATIVAS

El objetivo de PLSA reside en maximizar la estimación del parámetro: $\theta = (P(w|z), P(z|d))$. El algoritmo de maximización de expectativas (EM, del inglés *expectation-maximization algorithm*) se aplica para computar la estimación de la máxima verosimilitud en los modelos con variables latentes (Dempster et al., 1977). El principio básico del algoritmo EM es que, cada iteración de computación se considera un paso de expectativa seguido por un paso de maximización. Primero, se elige al azar un valor $\theta^{(0)}$ para inicializar el parámetro pendiente de estimar. El algoritmo itera constantemente con el fin de encontrar un óptimo $\theta^{(n+1)}$, donde el resultado de la función de verosimilitud $L(\theta^{(n+1)})$ debe ser más grande que $L(\theta^{(n)})$. Por tanto, el algoritmo se divide en dos pasos, uno de expectativa (E) y otro de maximización (M). En el paso (E), se calculan las probabilidades posteriores para las variables latentes, en este contexto, los temas. Para computar la probabilidad posterior de temas dados los datos observados, i.e. los documentos y las palabras, se aplica la fórmula del teorema Bayes (Fórmula 6):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Fórmula 6. Teorema de Bayes

Se aplican las probabilidades de la parametrización de la Fórmula 3 a la Fórmula 6 y se consigue:

$$P(z|d,w) = \frac{P(w|z)P(z|d)}{\sum_{i=1}^{k} P(w|z)P(z|d)}$$

Fórmula 7. Probabilidades de tema aplicando el teorema de Bayes

En el paso (M), el algoritmo iterativo EM va ajustando un corpus de entrenamiento D maximizando la función de similitud logarítmica $\mathcal{L}$:

$$\mathcal{L} = \sum_{d \in D} \sum_{w \in d} f(d,w) log P(d,w)$$

Fórmula 8. Función de similitud logarítmica

donde $f(d,w)$ representa la frecuencia de palabras w en el documento d. Los parámetros $P(w|z), P(z|d)$ se actualizan de forma iterativa para maximizar $\mathcal{L}$ (Brants et al., 2002). El proceso se repite hasta que el resultado converja.

2.3.3.4 RESUMEN ANALÍTICO SOBRE PLSA

PLSA es una variante estadística de LSA construyendo una distribución multinomial basada en el modelo generativo de los documentos. Las dos

técnicas difieren en algunos aspectos.

Por una parte, se basan en diferentes funciones de objetivo. Para LSA, se emplea la norma de L_2, mientras que en PLSA se utiliza la función de verosimilitud como el criterio de optimización para maximizar explícitamente el poder predictivo del modelo. Así pues, se cree que PLSA posee una base estadística más sólida que LSA.

Por otra parte, LSA realiza la reducción de dimensiones mediante SVD, mientras que PLSA emplea una descomposición probabilística que está estrechamente relacionada con la descomposición no negativa de matrices (Cohn y Hofmann, 2000).

Finalmente, en cuanto a la complejidad de computación, SVD puede ser compleja pero su resultado saldrá exacto. Sin embargo, PLSA utiliza el algoritmo iterativo EM para conseguir las estimaciones de la distribución conjunta de tema-palabra y la distribución de tema para cada documento (Steyvers y Griffiths, 2007). El problema del método EM es que solo garantiza encontrar un máximo local de la función de verosimilitud, lo que ha motivado a los investigadores a explorar mejores algoritmos de estimación (Blei et al., 2003).

Siendo PLSA y LDA las técnicas que adoptan el enfoque probabilístico, incorporan muchos supuestos básicos de LSA. La mejora principal de PLSA en la base de LSA consiste en que utiliza un modelo generativo de clase latente para realizar la descomposición. Además, la introducción de los sofisticados métodos estadísticos facilita la representación matemática de las distribuciones de los documentos y las palabras. Los temas son interpretables mediante una distribución probabilística en lugar de un espacio vectorial semántico. Por tanto, comparado con LSA, PLSA trata mejor el problema de las palabras polisémicas.

A partir del modelo de PLSA, se han realizado investigaciones en las tareas relacionadas con el modelado de textos, el filtro colaborativo y la minería del uso Web. Por ejemplo, Hennig (2009) presenta un método de recapitulación de documentos basado en consultas a partir de PLSA y concluye que PLSA puede captar mejor la información dispersa contenida en un texto que LSA. Das et al.

(2007) plantean un método del filtro colaborativo basado en PLSA que tiene el objetivo de construir un modelo predictivo desde una base de datos de preferencias de usuarios. Comprueban que este método consigue una recomendación precisa y es extremadamente escalable y flexible. Según Jin et al. (2004), PLSA es útil en la minería del uso Web, i.e. el descubrimiento de patrones en el comportamiento de navegación de los usuarios en la Web, teniendo en cuenta de que puede descubrir las asociaciones semánticas latentes entre los usuarios y las páginas recuperadas a partir de las coocurrencias en las sesiones de usuario. En el ámbito académico, se ha desarrollado un modelo de citación que puede identificar la autoridad de un documento determinado entre un conjunto de documentos vinculados, p.ej. las páginas recuperadas de la Web, los artículos de investigación recuperados de un repositorio (Cohn y Chang, 2000). PLSA da un paso hacia adelante para abordar uno de los desafíos más acuciantes de la “era de información”, i.e. la localización de las informaciones útiles en un entorno semi-estructurado como la Web. Gracias a su buen rendimiento en la representación de los documentos en un espacio de dimensiones reducidas, permite obtener los factores relacionados potencialmente más significativos y estables entre los textos y los temas.

La principal deficiencia de PLSA es que no hace suposición sobre cómo se generan las ponderaciones de los temas mezclados en los documentos. Como resultado, será difícil comprobar si el modelo puede generalizar a los documentos nuevos (Steyvers y Griffiths, 2007). Además, como PLSA estaba diseñado explícitamente para los datos discretos y cada documento de entrenamiento se trata como una entidad independiente, el modelo puede tener gran cantidad de rasgos y causar el problema del sobreajuste (Blei et al., 2003). Por lo tanto, aunque PLSA es más eficaz que las técnicas anteriores, aún requiere optimización y ajuste para obtener resultados fiables.

2.3.4 Asignación de Dirichlet latente

La asignación de Dirichlet latente es una técnica estadística que intenta captar la intuición de que un documento es una mezcla de múltiples temas (Blei et al., 2003). Los documentos son modelados por una variable aleatoria Dirichlet que especifica su distribución probabilística en un espacio temático latente y de baja dimensionalidad. Dirigido a colecciones de datos discretos, LDA proporciona una completa semántica probabilística generativa para los documentos. Se cree que LDA consigue una mejora en la base de PLSA mediante la incorporación de la distribución de Dirichlet para estimar las distribuciones documento-tema y término-tema con un enfoque bayesiano. Asimismo, igual que sus predecesores, LDA se basa en el modelo de bolsa de palabras, es decir, un documento está compuesto por un conjunto de palabras sin relación secuencial entre sí. Además, comparte la idea de que un documento puede contener varios temas y cada palabra del documento se genera desde uno de estos temas.

2.3.4.1 Distribución Dirichlet

La distribución Dirichlet $Dir(\alpha)$ es una serie de distribuciones de probabilidad continuas multivariada parametrizada por un vector α de números reales positivos (Blei et al., 2003). Los supuestos estadísticos del proceso generativo de LDA se pueden codificar de forma matemática a través de una distribución conjunta; también se pueden ilustrar con el modelo gráfico probabilístico en la Figura 4. Se describe LDA con la siguiente notación:

D: número de documentos;
N: número de palabras en un documento;
K: número de temas;
α: parámetro de Dirichlet priori sobre θ;
η: parámetro de Dirichlet priori sobre β;

θ_d: distribución de temas para el documento d;

β_k: distribución de palabras para tema k;

$z_{d,n}$: el tema para la palabra con número n en el documento con número d;

$w_{d,n}$: la palabra con número n en el documento con número d;

La distribución de temas θ y la distribución de palabras β son extraídas respectivamente desde α y η. Estas últimas se denominan Dirichlet priori. El valor α implica el número de temas dominantes que puede contener un documento. En concreto, cuanto más grande sea α, más temas se incluirían en el documento. Igualmente, un valor grande o pequeño de η significa que un tema tiene más o menos palabras dominantes. A partir de la distribución Dirichlet $Dir(\alpha)$, se extrae una muestra aleatoria a través del muestreo Gibbs que representa la distribución de temas θ para un documento. θ también se considera como la proporción de temas en el documento. Basado en esta distribución mezclada θ, se puede obtener un tema z. Asimismo, se observa otra distribución Dirichlet $Dir(\eta)$, que es el parámetro de Dirichlet priori sobre las distribuciones de palabras para cada tema. Desde $Dir(\eta)$, se selecciona una muestra de la misma manera para representar la distribución de palabras β dado un tema específico. En otras palabras, las distribuciones Dirichlet generan otras distribuciones. Por lo tanto, θ y β se llaman distribución sobre distribuciones. El proceso generativo para LDA se puede representar con la siguiente distribución conjunta de las variables observadas y latentes:

$$P(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} P(\beta_i) \prod_{d=1}^{D} P(\theta_d) \left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n}|\beta_{1:K}, z_{d,n}) \right)$$

Fórmula 9. Proceso generativo de textos en LDA

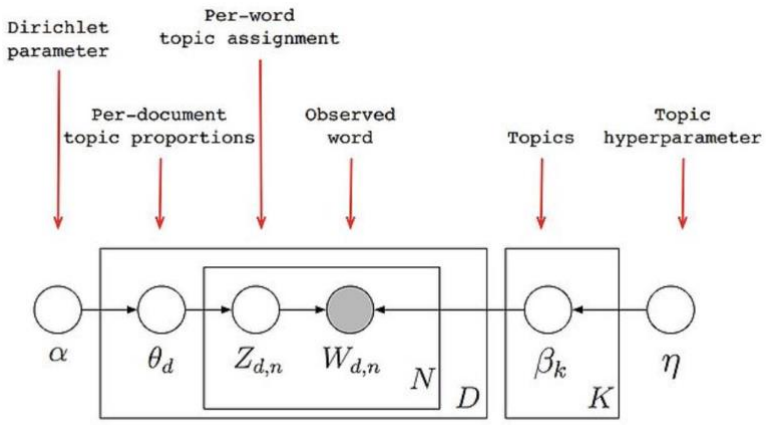


Figura 4. Representación gráfica del LDA (Blei, 2012)

2.3.4.2 MÉTODO DE INFERENCIA Y ESTIMACIÓN DE PARÁMETRO

En el modelo de LDA, las palabras $w_{d,n}$ constituyen los datos observados mientras que las distribuciones de temas y palabras son ocultas. α y η son hiperparámetros. El objetivo es inferir las distribuciones y los hiperparámetros. Se trata de un problema difícil de computar. Normalmente, tres técnicas son disponibles para la inferencia y la estimación de los parámetros: muestreo de Gibbs, algoritmo de maximización de expectativas y algoritmo variacional.

El muestreo de Gibbs (en inglés, *Gibbs Sampling*) es un algoritmo de Monte Carlo basado en cadenas de Markov para realizar el muestreo desde una distribución conjunta cuando las distribuciones condicionales de temas y palabras se pueden computar eficazmente. El método Monte Carlo basado en cadenas de Markov se refiere a un conjunto de técnicas iterativas de aproximación para muestrear desde una distribución de alta dimensionalidad (Gilks et al., 1996). Una cadena Markov es una secuencia de variables aleatorias. Cada una es independiente de la anterior y su distribución de límite es la posterior. El algoritmo ejecuta la cadena Markov iterativamente, recoge muestras desde la distribución de límite y gradualmente aproxima la distribución de las muestras elegidas.

Hemos discutido el algoritmo de maximización de expectativas en la Sección

2.3.3.3. Es una técnica útil para la estimación de parámetros mediante la verosimilitud máxima.

El algoritmo variacional es una alternativa a los algoritmos basados en el muestreo. Plantea un conjunto parametrizado de distribuciones sobre la estructura latente e intenta encontrar el elemento óptimo que se aproxima a la posterior. La tarea se transforma en un problema de optimización para buscar los mejores parámetros variacionales. El proceso se realizar a través de minimizar la divergencia de Kullback-Leibler entre la distribución aproximada y la posterior verdadera.

2.3.4.3 Resumen analítico sobre LDA

LDA empieza por asignar al azar un tema a cada palabra en cada documento. A continuación, las distribuciones de temas y de palabras son calculables. Estas distribuciones se utilizan en la siguiente iteración para reasignar temas y el proceso se repite hasta que converjan los resultados. Una vez obtenidas las distribuciones durante el entrenamiento, se pueden identificar los temas de los documentos de prueba mediante su ubicación en el espacio temático. Al igual que PLSA, LDA también se basa en las distribuciones de probabilidad. Considera cada documento como una distribución de temas, y trata cada tema como una distribución de palabras.

En la actualidad, LDA es una de las técnicas más populares en el modelado de temas por las ventajas que tiene comparado con LSA y PLSA. En primer lugar, LSA es simplemente una técnica de reducción de dimensionalidad y carece de un enfoque probabilístico sólido. Aunque proporciona un buen rendimiento al modelar la sinonimia, fracasa en procesar la polisemia. Por otra parte, PLSA remedia el problema de LSA parcialmente a través de la construcción de un modelo probabilístico generativo. A modo de ilustración, los temas se recogen desde una distribución probabilística $P(z)$ y luego se seleccionan los documentos y las palabras desde las probabilidades $P(d|z)$ y $P(w|z)$, respectivamente. Sin

embargo, PLSA no ofrece un modelo probabilístico en el nivel de documento. Como consecuencia, el número de parámetros en PLSA crecerá de modo lineal cuando se incrementa el tamaño del corpus, lo que provocará el problema del sobreajuste. Además, PLSA no es capaz de asignar probabilidades de tema a los nuevos documentos. Por el contrario, la distribución Dirichlet de LDA permite asignar probabilidades a nuevos documentos.

No obstante, LDA comparte algunas deficiencias con sus predecesores. Por una parte, se debe conocer de antemano el número de temas. Por otra parte, la hipótesis de bolsa de palabras no tiene en cuenta la representación semántica de las palabras en un corpus. Diversos trabajos de investigación han concluido que LDA puede agrupar los documentos en temas demasiado generales (Zhou et al., 2019) o irrelevantes (Alnusyan et al., 2020), o incluso puede dar resultados incoherentes entre distintas ejecuciones (Egger et al., 2021).

3 METODOLOGÍA

3.1 Evaluación del modelado de temas

Esta investigación se centra en evaluar y comparar desde un enfoque cuantitativo y cualitativo la tarea de agrupación de textos a través de dos técnicas probabilísticas del modelado de temas, i.e. PLSA y LDA.

El modelo con LSA, que utiliza el álgebra lineal para descomponer un corpus en los temas que lo construyen, se considera la base de la formulación probabilística del PLSA. Las primeras evaluaciones con LSA mostraron buenos resultados al replicar los juicios humanos (Chang et al., 2009). En la recuperación de información, se utiliza para vincular las consultas a los documentos.

Las dos técnicas alternativas, PLSA y LDA, que pertenecen a los modelos generativos, tienen un fundamento estadístico más sólido que LSA. Son modelos muy populares para realizar diferentes tipos de análisis de textos. En definitiva, se puede considerar LDA como una versión Bayesiana de PLSA y éste a su vez como una versión probabilística de LSA.

Después de aplicar una técnica de modelado de temas a una colección de textos, es importante identificar si un modelo entrenado es objetivamente bueno o malo, así como comparar los distintos modelos. En esta investigación, también pretendemos averiguar si el tamaño de un corpus tiene una influencia significativa en el rendimiento del modelo.

Desde la perspectiva cuantitativa, las tareas extrínsecas, como la recuperación de información, la agrupación y clasificación de textos, disponen de las métricas extrínsecas para su evaluación. Desde la perspectiva cualitativa, típicamente se supone que el espacio latente de temas descubierto por el modelado de temas es semánticamente significativo. La presentación de las distribuciones multinomiales sobre las palabras asociadas a los temas sirve de una evaluación cualitativa del espacio latente. A continuación, presentamos los métodos que se utilizan habitualmente para la evaluación del modelado de temas.

3.1.1 MÉTODO DE LA EXPLORACIÓN VISUAL

Desde un punto de vista computacional, el lenguaje natural es irregular y ambiguo. La minimización de la ambigüedad para facilitar su procesamiento con los ordenadores reduce el lenguaje a una forma menos natural. A través de la vectorización de las palabras, los ordenadores son capaces de realizar una tarea de modelización del lenguaje natural. Por el contrario, los humanos somos mejores en detectar de forma intuitiva los patrones a partir de un gráfico que por medio de una simple secuencia de números (Christopoulos et al., 2000).

El método de la exploración visual es una técnica subjetiva de evaluación basada en la observación manual de los resultados. Mediante una percepción e interpretación visual sobre las palabras más frecuentes extraídas de los corpus por el modelado de temas, nos permite realizar una valoración rápida sobre las informaciones sin utilizar los métodos cuantitativos.

El proceso de inspeccionar visualmente los resultados es útil para tener una idea aproximada de los datos. Sin embargo, la percepción puede ser influida por prejuicios personales e interpretaciones subjetivas, lo cual suele llevar a una conclusión sesgada. Además, evaluar la calidad de un modelo examinando las palabras más frecuentes de cada tema requiere mucho trabajo y es bastante difícil para extensos modelos del lenguaje.

3.1.2 MÉTODOS DE EVALUACIÓN INTRÍNSECA

El método cuantitativo para evaluar el modelado de temas implica múltiples métricas estadísticas. Las métricas pueden clasificarse a grandes rasgos en dos categorías: las métricas intrínsecas y las métricas extrínsecas. Las métricas intrínsecas sirven para estimar la viabilidad de los modelos diseñados para generar representaciones del lenguaje contextualizado. Estos modelos pueden ayudar a mejorar la precisión y el rendimiento de las aplicaciones posteriores de PLN. Las métricas extrínsecas para el modelado de temas son utilizadas para

evaluar el rendimiento de un modelo con una tarea posterior, como la clasificación de textos y la recuperación de información (Cao et al., 2022).

La perplejidad y la coherencia son dos métricas de evaluación intrínseca para el modelado de temas. Por un lado, la perplejidad mide el rendimiento de una distribución probabilística o de un modelo de probabilidad al predecir una muestra. Por lo tanto, la métrica es una forma de captar el grado de incertidumbre que tiene un modelo al predecir un texto sobre el conjunto de pruebas y sirve de parámetro para comparar distintos modelos. Cuanto más baja sea la perplejidad, más alta será la probabilidad que asigne el modelo al conjunto de pruebas y mejor será el modelo en predecir la muestra. La perplejidad se puede calcular como la probabilidad inversa del conjunto de pruebas normalizada por el número de palabras. Por ejemplo, dado un conjunto de pruebas que contiene una secuencia de palabras $W = w1, w2, \dots, wN$, donde N es el número de palabras en el conjunto, su perplejidad se calcula como $perplejidad(W) = P(w1, w2 \dots wN)^{-1/N}$. Sin embargo, algunos estudios han demostrado que, sorprendentemente, los modelos que logran una mejor perplejidad predictiva suelen tener espacios latentes menos interpretables (Chang et al., 2009).

Por otro lado, la coherencia temática es una métrica para evaluar la interpretabilidad de los temas descubiertos por los modelos. Su puntuación para un tema determinado mide el grado de similitud semántica entre las palabras más frecuentes del tema. Por lo tanto, puede reflejar el desempeño del modelo en captar el concepto subyacente del tema. Uno de los métodos más comunes para calcular la coherencia temática es el punto de información mutua (PIM, del inglés *pointwise mutual information*), que mide el grado de asociación entre dos palabras mediante el cociente logarítmico entre su probabilidad conjunta y sus probabilidades individuales. Una puntuación PIM más alta indica una asociación más fuerte entre dos palabras.

3.1.3 MÉTODOS DE EVALUACIÓN EXTRÍNSECA

Ante una tarea de clasificación en el aprendizaje automático, la matriz de confusión es una herramienta para visualizar el rendimiento de un algoritmo. Es una tabla que resume el número de predicciones correctas e incorrectas realizadas por el modelo para cada clase del conjunto de datos. Cada columna de la matriz representa las instancias de una clase predicha mientras que cada fila indica las de una clase real (Powers, 2020).

En las tareas de clasificación, el verdadero positivo (TP, del inglés *true positive*), el verdadero negativo (TN, del inglés *true negative*), el falso positivo (FP, del inglés *false positive*) y el falso negativo (FN, del inglés *false negative*) son cuatro términos que se emplean en una matriz de confusión para comparar los resultados del modelo con los juicios externos fiables realizados por las personas. El TP ocurre cuando el modelo identifica correctamente una instancia positiva en comparación con la interpretación humana. El FN aparece al predecir correctamente una instancia negativa mientras que el FP ocurre cuando el modelo identifica incorrectamente una condición como presente, pero en realidad está ausente. El FN corresponde a las situaciones en las que el modelo predice una instancia como negativa y en realidad es positiva.

A partir de la matriz de confusión, se puede calcular una serie de métricas como precisión, cobertura y valor F1 para valorar la efectividad de los modelos:

- La precisión mide la proporción de los casos verdaderamente positivos en relación con los casos clasificados como positivos por el modelo. Es una métrica muy útil especialmente cuando el coste de los positivos falsos es elevado. En la tarea de agrupar los textos según temas, la precisión refleja la proporción de los textos correctamente clasificados en el grupo de un tema determinado (TP) con respecto a la totalidad de los que tienen asignada la etiqueta de ese tema, aunque se incluyen los erróneamente agrupados (TP+FP). Se calcula con la Fórmula 10:

$$Presición = \frac{TP}{TP + FP}$$

Fórmula 10. Métrica de evaluación Precisión

- La cobertura es la medida que evalúa cómo de bien un modelo identifica los casos positivos en relación con todos los casos positivos reales. Es la métrica ideal para seleccionar el mejor modelo cuando hay un coste alto asociado a los negativos falsos. En el contexto del modelado de temas, mediante la cobertura se obtiene el porcentaje de los textos bien identificados en sus temas (TP) con respecto a todos los textos que realmente pertenecen a esas clases (TP+FN). Para calcular la cobertura, se usa la Fórmula 11:

$$Cobertura = \frac{TP}{TP + FN}$$

Fórmula 11. Métrica de evaluación Cobertura

- El valor F1 es la media armónica de precisión y cobertura. Es práctico para encontrar un equilibrio entre las dos métricas. A diferencia de otras tareas en las que las consecuencias de los falsos positivos o los falsos negativos son más graves (por ejemplo, la detección de spam en el correo electrónico y la detección de fraudes), consideramos que el coste de los dos casos de clasificación errónea en el modelado de temas debería tener un peso similar. Por lo tanto, el valor F1 es ideal para evaluar los modelos. Se calcula con la Fórmula 12:

$$Valor\ F1 = 2 * \frac{presición * exhaustividad}{presición + exhaustividad}$$

Fórmula 12. Métrica de evaluación Valor F1

3.2 Implementación de los modelos

3.2.1 Creación de corpus

Con el fin de comparar el rendimiento de las técnicas del modelado de temas y analizar cómo el tamaño del corpus puede afectar a los resultados que generan, creamos tres corpus recopilando los subtítulos de las noticias publicadas entre enero y diciembre de 2022 en *Wall Street Journal* y *Nature Journal*. Los textos están en inglés y fueron seleccionados aleatoriamente entre los cuatro temas, i.e. economía, política, ciencia y deportes. Los textos de economía, política y deportes fueron extraídos de las correspondientes secciones de *Wall Street Journal* y los de ciencia provinieron de *Nature Journal*. Los corpus originales se almacenaron en una base de datos SQLite, donde cada corpus se guardó en una tabla cuyos campos representan: un identificador único para cada texto, la fuente, el propio texto de la noticia y las anotaciones temáticas verificadas por la interpretación humana. Teniendo en cuenta que se utilizaron cuatro categorías temáticas (i.e. economía, política, ciencia y deportes), la distribución de los corpus fue la siguiente:

- Corpus #1: contiene 50 textos para cada tema, haciendo un total de 200 textos.
- Corpus #2: contiene 200 textos para cada tema, haciendo un total de 800 textos (incluidos los del Corpus #1).
- Corpus #3: contiene 400 textos para cada tema, haciendo un total de 1600 textos (incluidos los del Corpus #2).

La misma colección de textos se utilizará con PLSA y LDA. Para cada modelo, el número de temas T es determinado previamente, i.e. T = 4.

3.2.2 Procesamiento de corpus

En esta investigación utilizamos *CSharp Scripting*, desarrollado por el Dr.

Periñán-Pascual para implementar y evaluar PLSA y LDA. *CSharp Scripting* es una herramienta integrada en TexMiLAB, la cual se desarrolló a partir de la arquitectura del sistema DAMIEN (Periñán-Pascual, 2017), una aplicación en línea diseñada para la implementación de la investigación lingüística experimental basada en corpus que integra técnicas y métodos de la minería textual y el PLN. Una serie de métodos de máxima verosimilitud truncada son integrados en la aplicación y son utilizados en el experimento. Se trata de métodos para estimar los parámetros de los modelos de acuerdo con una muestra de datos truncados maximizando la verosimilitud de los datos observados. En el resto de este apartado, explicamos las funciones de los bloques de código utilizados en cada paso de la implementación de los modelos. El código completo se ha incluido en el Anexo 1.

En el primer paso, manipulamos los archivos y creamos una matriz documento N-grama a partir del Corpus #1, repitiendo el mismo proceso para el Corpus #2 y #3). En el código de la Figura 5, observamos que las líneas 1, 2 declaran dos variables de tipo "`string`", p.ej. a la variable "`fileName`" se le asigna la ruta completa de los archivos de los corpus. La ruta se construye concatenando "`path`" con el nombre del archivo. La línea 3 declara una variable llamada "`stopwordsFile`" y se le asigna un archivo de texto que contiene las palabras vacías o funcionales. Son palabras como artículos, preposiciones, pronombres etc. que no afectan la categorización temática de los textos, razón por la cual se filtran desde el corpus. La línea 4 declara la ruta de un directorio de recursos necesarios para el procesamiento del corpus. En la línea 6 empleamos el método "`Files.ReadText`" para leer el archivo de texto que contiene el Corpus #1, el cual se genera automáticamente a partir de una sentencia SQL que permite recuperar los valores del campo "texto" que contiene nuestra base de datos, y lo asigna a la variable "`corpus`". En la línea 8 usamos el método "`Ngrams.GetDocNgramMatrix`", el cual incluye varios argumentos: "`corpus`" como el texto de entrada, "eng" como el idioma, "`multiple`" como el tipo de entrada (i.e. se trata de un único archivo que contiene múltiples

documentos, más concretamente, uno por cada línea) , “unigram” como el tipo de N-grama, “lexeme” como el tipo de unidad de análisis con la que se va a trabajar, “frecuency” como el valor inicial que se va a asignar a cada unidad de análisis, “stopwordsFile” como el archivo que contiene las palabras funcionales en inglés, “startlistFile” como el archivo de palabras que se deben incluir obligatoriamente en el análisis (en este caso, no es pertinente, razón por la cual utilizamos el valor “null”) y “resourcesPath” para acceder a los recursos necesarios de PLN, los cuales están integrados en la propia aplicación. El resultado se almacenará en la variable “matrix”, definida en la línea 7, como un parámetro de salida que contiene el método “Ngrams.GetDocNgramMatrix” en la línea 8.

```
1 string path = @"C:\Users\DLA\Desktop\advanced\";
2 string fileName = path + "corpus_small.txt";
3 string stopwordsFile = @"C:\Users\DLA\Desktop\experiment\eng_functional.txt";
4 string resourcesPath =
@"C:\Users\DLA\Desktop\CSharpScripting\Release\resources\";
5 string corpus;
6 Files.ReadText(fileName, out corpus);
7 DataTable matrix;
8 Ngrams.GetDocNgramMatrix(corpus, "eng", "multiple", "unigram", "lexeme",
"frequency", stopwordsFile, null, resourcesPath, out matrix);
```

Figura 5. Código para el procesamiento de los corpus

3.2.3 APLICACIÓN DE LOS MODELOS

En esta fase aplicamos LDA y PLSA a la matriz de N-gramas que hemos creado en la fase anterior. En la línea 9 de la Figura 6 declaramos la variable “numTopic” de tipo “int” y le asignamos el valor 4. Este valor representa el número de temas que se utilizarán en el análisis. En la línea 11 empleamos el método “LDA.Apply” para aplicar LDA al contenido de la variable “matrix”. De hecho, los argumentos de este método incluyen “matrix” como la matriz de N-gramas, “ID” como el

nombre de la columna de identificación de textos, “numTopic” como el número de temas y los parámetros de salida “topicNgramMatrix”, “docTopicMatrix”, “topicProbability” y “evaluation” para almacenar los resultados generados por el modelo. Las matrices de salida corresponden a la matriz tema-Ngrama y la matriz documento-tema, respectivamente. También se devuelven las probabilidades de temas y las puntuaciones de evaluación. En la línea 12, el código para aplicar PLSA está comentado con una barra doble oblicua. Funciona como la opción alternativa que se ejecutará para PLSA, pero que no se aplica con LDA.

```
9 int numTopic = 4;
10 string topicNgramMatrix, docTopicMatrix, topicProbability, evaluation;
11 LDA.Apply(matrix, "ID", numTopic, out topicNgramMatrix, out docTopicMatrix,
out topicProbability, out evaluation);
12 //PLSA.Apply(matrix, "ID", numTopic, out topicNgramMatrix, out docTopicMatrix,
out topicProbability, out evaluation);
```

Figura 6. Código para la aplicación de los modelos

3.2.4 VISUALIZACIÓN DE LOS RESULTADOS

En la siguiente fase, realizamos una serie de procesamiento sobre los resultados generados por LDA o PLSA a partir de los tres corpus. En la línea 14 de la Figura 7, usamos el método “ConvertCSVToTable” de clase “DataMisc” para convertir el archivo CSV “docTopicMatrix” en una tabla de datos y almacenarlo en “dt1”. En la línea 16 se utiliza el modelo “GetBestTopicForDocs” de la clase “TopicModel” para obtener los mejores temas para cada documento en “dt1” y se almacenan los resultados en “topics”, que es definido como un nuevo objeto “Dictionary” en la línea 15. Guardamos el listado de los identificadores de textos y los temas predichos al archivo CSV “topics.csv”, que se ubica en la ruta especificada por “path”. El listado se usará para evaluar los modelos posteriormente. Con el código en la línea 21 guardamos el contenido de la matriz

tema-Ngramas en el archivo CSV “topicNgramMatrix.csv”. Según la línea 23, lo convertimos en una tabla y la almacenamos en “dt2”. Para obtener los N-gramas más representativos para cada tema, creamos un bucle que se itera desde 1 hasta el valor de “numTopic”, que es 4 en este experimento (línea 26-33). La línea 27 declara una variable “topic” de tipo “string” que se inicializa concatenando la cadena “topic” con el valor de la variable “x”.

```
13 DataTable dt1;
14 DataMisc.ConvertCSVToTable(docTopicMatrix, out dt1);
15 Dictionary<string, string> topics;
16 TopicModel.GetBestTopicForDocs(dt1, out topics);
17 Editor.Show(topics);
18 Editor.Show(Environment.NewLine + Environment.NewLine);
19 Editor.Save("id" + "\t" + "predicted", path + "topics.csv");
20 Editor.AppendToFile(topics, path + "topics.csv");
21 Editor.Save(topicNgramMatrix, path + "topicNgramMatrix.csv");
22 DataTable dt2;
23 DataMisc.ConvertCSVToTable(topicNgramMatrix, out dt2);
24 Dictionary<string, double> ngrams;
25 for(int x = 1; x <= numTopic; x++)
26 {
27    string topic = "topic" + x.ToString();
28    TopicModel.GetBestNgramsForTopic(dt2, topic, 20, out ngrams);
29    Editor.Show(topic.ToUpper());
30    Editor.Show(ngrams);
31    Editor.Show(Environment.NewLine);
32    Editor.Save(ngrams, path + topic + ".csv");
33 }
34 Editor.Show(topicProbability + Environment.NewLine + Environment.NewLine +
evaluation);
```

Figura 7. Código para la visualización de los resultados

En la línea 28, empleamos el método “GetBestNgramsForTopic” de la clase “TopicModel” y le pasamos los siguientes argumentos: la tabla “dt2” como la matriz tema-Ngrama, la variable “topic” previamente creada y el número de los mejores N-gramas que queremos que devuelva. Guardamos los resultados de

los mejores N-gramas de cada tema en archivos CSV para una posterior exploración. Con el código que se indica en la línea 34, obtenemos los resultados de las probabilidades de los temas y de la evaluación intrínseca, que incluye la perplejidad y la coherencia.

3.2.5 Evaluación de los resultados

La evaluación extrínseca de los resultados se divide en varios pasos. En primer lugar, realizamos una exploración visual de los N-gramas más importantes de cada tema, los interpretamos según nuestro juicio como humanos y pusimos a cada uno una etiqueta del mejor tema. La matriz de confusión se puede utilizar para extraer los valores de TP, TN, FP y FN para la clasificación binaria y multiclase. En nuestro caso, como los corpus son multitemáticos, pertenece a la última. Por lo tanto, fue necesario calcular la precisión, la cobertura y el valor F1 para cada uno de los cuatro temas en los corpus de tres tamaños. Una vez que se obtuvo el listado de los temas asignados para cada corpus, se realizó un análisis manual por cada clase individualmente. Aplicamos el valor 1 a una clase en concreto y al resto el valor 0. Realizamos la misma transformación con las etiquetas originales de tema y obtuvimos los archivos "`corpus_small_modificado.csv`" y "`topics_modificado.csv`".

Se utilizó el método "`Files.ReadDataTable`" para cargar los dos archivos CSV en las tablas de "`finalDataset`" y "topics" que se declaran en línea 35 y 37 en la Figura 8. El método "`Datatable.GetFieldValues`" en la línea 40 extrae los valores del campo "`predicted`" en la tabla "topics" y los guarda en el vector "`newValues`" que se ha declarado en la línea 39. El método en la línea 41 "`Datatable.InsertFieldandValues`" inserta un nuevo campo "`predicted`" en la tabla "`finalDataset`" y le asigna los valores del vector "`newValues`". Para comparar los resultados predichos por los modelos con los esperados, en la línea 43 aplicamos el método "`GetValues`" de la clase "`ConfusionMatrix`" para obtener los valores de los campos "`predicted`" y "`expected`" en la tabla

"finalDataset" y guardarlos en los vectores de números enteros "predicted" y "expected" que se han declarado en la línea 42. A través del mismo método "GetValues" calculamos los valores de TP, FN, FP y TN basados en los vectores de predicción y valores esperados. En la línea 46, declaramos las variables de tipo double "precision", "recall" y "F1". Al final, de la línea 47 a 49, utilizamos los métodos "GetPrecision", "GetRecall" y "GetFScore" de la clase "ConfusionMatrix", los cuales contienen los parámetros "TP", "FN", "FP" y "TN". De esta forma, obtuvimos los resultados de las métricas de evaluación: precisión, cobertura y valor F1.

```
35 DataTable finalDataset;
36 Files.ReadDataTable(path + "corpus_small_modificado.csv", out finalDataset);
37 DataTable topics;
38 Files.ReadDataTable(path + "topics_modificado.csv", out topics);
39 string[] newValues;
40 Datatable.GetFieldValues(topics, "predicted", out newValues);
41 Datatable.InsertFieldandValues(finalDataset, "predicted", newValues, out
finalDataset);
42 int[] predicted, expected;
43 ConfusionMatrix.GetValues(finalDataset, "predicted", "expected", out
predicted, out expected);
44 int TP, FN, FP, TN;
45 ConfusionMatrix.GetValues(predicted, expected, out TP, out FN, out FP, out
TN);
46 double precision, recall, F1;
47 ConfusionMatrix.GetPrecision(TP, FN, FP, TN, out precision);
48 ConfusionMatrix.GetRecall(TP, FN, FP, TN, out recall);
49 ConfusionMatrix.GetFScore(TP, FN, FP, TN, out F1);
50 Editor.Show("precision: " + precision.ToString() + Environment.NewLine +
"recall: " + recall.ToString() + Environment.NewLine + "F1: " + F1.ToString());
51 Editor.Show(Environment.NewLine);
52 Editor.Show("TP: " + TP.ToString() + Environment.NewLine + "FN: " +
FN.ToString() + Environment.NewLine + "FP: " + FP.ToString() +
Environment.NewLine + "TN: " + TN.ToString());
```

Figura 8. Código para la evaluación de los resultados

4 ANÁLISIS DE RESULTADOS

En el experimento, aplicamos respectivamente PLSA y LDA a los tres corpus y obtuvimos las clases de agrupación de temas y los listados de las palabras más representativas de cada tema. Después, interpretamos los temas basados en las palabras con mayor probabilidad y comparamos los resultados de los modelos con los cuatro temas determinados de antemano. A partir de eso, calculamos y analizamos las métricas de evaluación de los distintos modelos en cada corpus.

4.1 Resultados de LDA

Al examinar visualmente las listas de los N-gramas con mayor probabilidad generadas por los modelos, no resultó fácil interpretar los temas a primera vista, especialmente al distinguir entre la economía y la política debido a que las noticias de economía suelen tratar contextos políticos y las de política hablan frecuentemente de asuntos económicos. Al identificar a qué tema se refiere cada una de las agrupaciones de palabras, la estrategia fue revisar tanto los N-gramas más frecuentes como los más específicos que se suelen utilizar en un campo determinado.

En primer lugar, del listado de las palabras más representativas de cada tema después de aplicar LDA al Corpus #1 (Tabla 2), deducimos que el Tema 1 debería ser economía por la alta probabilidad de “*bank*”, “*inflation*”, “*economic*” y “*growth*”. En segundo lugar, inferimos que el Tema 2 es política. Las palabras que encabezan el listado “*republican*”, “*democrat*”, “*country*”, “*president*”, “*election*” suelen aparecer en los textos políticos. En tercer lugar, la interpretación sobre el Tema 3 es difusa porque intuitivamente las palabras que se obtienen parecen una mezcla de diferentes temas. Después de una exploración sobre el listado completo de todas las palabras y los corpus originales, hemos descubierto que las noticias de ciencias que provienen de la revista *Nature* durante el año 2022 trataron a menudo sobre asuntos relacionados con el cambio climático, la

subida del precio de la energía y la investigación en alternativas a los combustibles fósiles. De esta forma, concluimos que palabras como "*price*", "*new*", "*rise*", "*change*" y "*energy*" justamente reflejan esos asuntos. Igualmente, se encontraron términos como "*gravitational*", "*radiation-blocking*" y "*alternative-energy*". Por tanto, es muy probable que el Tema 3 corresponda a ciencia. Por último, el Tema 4 debería ser deportes por la alta frecuencia en la presencia de palabras como "*sport*", "*athlete*", "*game*" y "*olympic*". También hemos detectado que en este último grupo se encuentran algunas palabras con probabilidades elevadas relacionadas con la economía y la política. Eso se debe a que la prensa suele informar sobre la influencia del macroentorno en la industria deportiva.

TEMA 1 (economía)		**TEMA 2 (política)**		**TEMA 3 (ciencia)**		**TEMA 4 (deportes)**	
new	0.01483	republican	0.0172	price	0.01174	sport	0.01543
bank	0.01059	sport	0.01505	year	0.01071	college	0.00771
global	0.01059	inflation	0.01075	new	0.01006	decline	0.00771
major	0.01059	year	0.01075	rise	0.01006	demand	0.00771
year	0.00978	democrat	0.0086	change	0.00991	help	0.00771
central	0.00847	high	0.0086	show	0.00838	slow	0.00629
high	0.00847	labor	0.0086	democrat	0.00836	activity	0.00579
inflation	0.00847	make	0.0086	company	0.00671	american	0.00579
league	0.00847	package	0.0086	growth	0.00671	athlete	0.00579
researcher	0.00847	seek	0.0086	image	0.00671	find	0.00579
slow	0.00847	spend	0.0086	push	0.00671	game	0.00579
ukraine	0.00847	ukraine	0.0086	service	0.00671	government	0.00579
economic	0.00635	big	0.00645	study	0.00671	job	0.00579
far	0.00635	energy	0.00645	survey	0.00671	olympic	0.00579
former	0.00635	expect	0.00645	republican	0.00669	output	0.00579
growth	0.00635	former	0.00645	allow	0.00503	pro	0.00579
make	0.00635	household	0.00645	attorney	0.00503	program	0.00579
move	0.00635	include	0.00645	capitol	0.00503	region	0.00579
rate	0.00635	market	0.00645	department	0.00503	risk	0.00579
record	0.00635	new	0.00645	energy	0.00503	take	0.00579

Tabla 2. Las 20 palabras más representativas y sus probabilidades de cada tema aplicando LDA al Corpus #1

Después de identificar los cuatro temas manualmente, calculamos

automáticamente los valores de TP, FN, FP, TN en cada uno (Tabla 3). A partir de estos valores, conseguimos los valores de las métricas extrínsecas de evaluación para cada agrupamiento de documentos (Tabla 4). Por ejemplo, desde los resultados podemos ver que, LDA ha agrupado correctamente 15 textos de economía (TP). No obstante, ha puesto 36 textos (FP) en este grupo del tema de economía, aunque realmente son de otros temas. La precisión, la cobertura y el valor F1 del rendimiento del modelo en agrupar los textos económicos en el Corpus #1 corresponden a 0,294, 0,3 y 0,297, respectivamente. Para evaluar la agrupación multitemática con una métrica global, calculamos la macromedia de los valores de precisión, cobertura y F1 de los cuatro temas porque todas las clases contribuyen por igual a la métrica final promediada.

Corpus #1: Temas	**TP**	**FN**	**FP**	**TN**
Tema 1 Economía	15	35	36	114
Tema 2 Política	16	34	28	122
Tema 3 Ciencia	16	34	42	108
Tema 4 Deportes	16	34	31	119

Tabla 3. Valores de TP, FN, FP, TN de los cuatro temas aplicando LDA al Corpus #1

Corpus #1: Temas	**Precisión**	**Cobertura**	**Valor F1**
Tema 1 Economía	0.294	0.3	0.297
Tema 2 Política	0.364	0.32	0.34
Tema 3 Ciencia	0.276	0.32	0.296
Tema 4 Deportes	0.34	0.32	0.33
Macromedia	0.3185	0.315	0.31575

Tabla 4. Métricas extrínsecas de evaluación de los cuatro temas aplicando LDA al Corpus #1

En la Tabla 5 se presentan los resultados de las métricas intrínsecas de evaluación. Como hemos discutido en la Sección 3.1.2, la perplejidad es una medida sobre cuánto se sorprende el modelo cuando se le introducen datos que

no había visto anteriormente. Cuanto menor sea el valor de perplejidad, mejor será el modelo. Por otra parte, la coherencia evalúa cómo de coherentes son los temas generados por el modelo. Cuanto más distintas sean las palabras de los diferentes temas entre sí, menos relacionados estarán los temas y más coherente será el modelo. Cuanto mayor sea el valor de coherencia, mejor será el modelo. En la tabla vemos que al aplicar LDA al Corpus #1, la perplejidad y la coherencia promediada de los cuatro temas son 633,37822 y -118,30751, respectivamente.

Perplejidad	Coherencia	Coherencia-T1	Coherencia-T2	Coherencia-T3	Coherencia-T4
633.37822	-118.30751	-125.86030	-108.65531	-115.44930	-123.26513

Tabla 5. Métricas intrínsecas de evaluación de los cuatro temas aplicando LDA al Corpus #1 [T1=Economía; T2=Política; T3=Ciencia;T4=Deportes]

A partir de los mejores N-gramas extraídos por LDA del Corpus #2 (Tabla 6), repetimos el mismo proceso de exploración visual. En primer lugar, interpretamos el Tema 1 como economía y el Tema 3 como política. Aunque comparten muchas palabras con alta probabilidad, hemos decidido asignar el tema de política al agrupamiento de documentos donde "*republican*" y "*democrat*" aparecen frecuentemente. En segundo lugar, a primera vista el Tema 2 debería ser ciencia porque las noticias científicas pueden tratar asuntos sobre avances innovadores en distintos ámbitos académicos e industriales. La existencia de palabras como "*galaxy*", "*ecosystem*" ha verificado nuestra intuición. Finalmente, la identificación del Tema 4 es relativamente más fácil por la presencia de palabras como "*league*", "*race*", "*game*", "*NFL*" etc.

TEMA 1 (economía)		TEMA 2 (ciencia)		TEMA 3 (política)		TEMA 4 (deportes)	
business	0.00989	inflation	0.01153	inflation	0.00843	team	0.01245
year	0.00865	spend	0.00957	growth	0.00682	new	0.00964
increase	0.00596	new	0.00864	new	0.00682	set	0.00737
concern	0.00556	show	0.00826	rate	0.00662	former	0.00681
see	0.00556	year	0.00789	economy	0.00639	league	0.00592
recession	0.00494	price	0.00718	year	0.00631	change	0.00567
risk	0.00494	sport	0.00688	slow	0.00551	race	0.00567
rate	0.00479	ukraine	0.00685	make	0.00512	state	0.00533
league	0.00447	rise	0.00616	interest	0.00507	allow	0.00511
spend	0.00442	high	0.00598	call	0.00504	sport	0.00511
test	0.00435	play	0.00596	high	0.00494	ukraine	0.00511
return	0.00433	energy	0.00543	bank	0.0048	decision	0.00485
show	0.00433	far	0.00505	come	0.00469	athlete	0.00454
struggle	0.00433	war	0.00505	republican	0.00426	game	0.00454
rise	0.00425	use	0.00503	space	0.00426	mission	0.00454
world	0.00384	slow	0.005	sport	0.00417	high	0.00411
inflation	0.00376	growth	0.00459	labor	0.00384	image	0.00397
change	0.00371	pandemic	0.00457	market	0.00384	nfl	0.00397
come	0.00371	household	0.00455	president	0.00384	record	0.00397
economy	0.00371	economic	0.00413	central	0.00369	rule	0.00397

Tabla 6. Las 20 palabras más representativas y sus probabilidades de cada tema aplicando LDA al Corpus #2

Los valores de TP, FN, FP, TN y las métricas extrínsecas para cada tema se han resumido en Tabla 7 y Tabla 8. Considerando que el Corpus #2 es cuatro veces más grandes que el Corpus #1, habíamos supuesto que el modelo podría dar resultados más precisos al aplicarse al Corpus #2. Sin embargo, los datos indican que las macromedias de la precisión, la cobertura y el valor F1 del segundo corpus son 6,9%, 9,1%, 8,3%, respectivamente, más bajos que el primero. Al comparar los valores de evaluación entre cada agrupamiento de documentos, LDA genera peor resultado en la tarea de agrupación de los temas de economía, política y ciencia. No obstante, la precisión del tema de deportes del Corpus #2 ha mejorado en un 35%, pasando de 0,34 a 0,439.

Corpus #2: Temas	TP	FN	FP	TN
Tema 1 Economía	50	150	120	480
Tema 2 Ciencia	50	150	173	427
Tema 3 Política	54	146	182	418
Tema 4 Deportes	75	125	96	504

Tabla 7. Valores de TP, FN, FP, TN de los cuatro temas aplicando LDA al Corpus #2

Corpus #2: Temas	Precisión	Cobertura	Valor F1
Tema 1 Economía	0.294	0.25	0.27
Tema 2 Ciencia	0.224	0.25	0.236
Tema 3 Política	0.229	0.27	0.248
Tema 4 Deportes	0.439	0.375	0.404
Macromedia	0.2965	0.28625	0.2895

Tabla 8. Métricas extrínsecas de evaluación de los cuatro temas aplicando LDA al Corpus #2

En cuanto a las métricas intrínsecas, los resultados que genera LDA en el Corpus #2 corresponden a una perplejidad de 1099,65974 y una coherencia de -173,31525. En comparación con los valores del Corpus #1, hemos observado un 73,62% y un 46,49% de aumento en la perplejidad y la coherencia. Eso significa que el modelo da peores resultados en el Corpus #2.

Perplejidad	Coherencia	Coherencia-T1	Coherencia-T2	Coherencia-T3	Coherencia-T4
1099.65974	-173.31525	-183.76582	-175.48630	-160.68842	-173.32048

Tabla 9. Métricas intrínsecas de evaluación de los cuatro temas aplicando LDA al Corpus #2 [T1=Economía; T2=Ciencia; T3=Política;T4=Deportes]

A medida que crece el corpus, hemos percibido que es cada vez más fácil identificar los temas a través del método de la exploración visual . Determinamos que los cuatro temas corresponden respectivamente a economía, ciencia,

deportes y política (Tabla 10). Calculamos los valores de TP, FN, FP, TN (Tabla 11) y las métricas de evaluación (Tabla 12). Aunque el Corpus #3 es dos veces más grande que el #2, los resultados de los dos conjuntos son similares. Entre los tres corpus, el modelo funciona mejor con el más pequeño. Podemos llegar a la misma conclusión comparando los valores de las métricas intrínsecas. Obtenemos una perplejidad de 1646,77424 y una coherencia de -190,26235, que corresponden a una subida de 49,75% y 9,78% en comparación con el Corpus #2.

TEMA 1 (economía)		**TEMA 2 (ciencia)**		**TEMA 3 (deportes)**		**TEMA 4 (política)**	
year	0.01055	new	0.0098	republican	0.0093	inflation	0.01076
rise	0.00636	team	0.00705	state	0.00823	president	0.0082
high	0.00566	year	0.00604	sport	0.00637	rate	0.00633
spend	0.00551	republican	0.00495	win	0.00441	high	0.00592
change	0.00512	rate	0.00494	race	0.00414	growth	0.00574
new	0.00482	use	0.00483	democrat	0.00389	former	0.00563
inflation	0.00478	reveal	0.00475	show	0.00386	slow	0.00563
economic	0.00448	increase	0.0043	cause	0.00381	bank	0.00547
market	0.00395	make	0.00416	new	0.00368	new	0.00541
use	0.00393	show	0.00416	trump	0.0036	ukraine	0.00486
rate	0.0038	star	0.0041	year	0.0036	policy	0.00463
show	0.00379	person	0.00395	president	0.00348	economy	0.00452
economy	0.00377	effort	0.00379	record	0.00341	central	0.00448
challenge	0.00376	rule	0.00371	former	0.00339	call	0.00435
find	0.00366	come	0.00361	major	0.00332	increase	0.00424
labor	0.00366	game	0.00335	star	0.00332	interest	0.00417
plan	0.00341	month	0.00324	report	0.00321	price	0.00397
datum	0.00334	price	0.00321	athlete	0.00304	sport	0.00397
official	0.00317	bank	0.00315	candidate	0.00304	change	0.00389
make	0.00313	leave	0.00309	good	0.00285	set	0.00387

Tabla 10. Las 20 palabras más representativas y sus probabilidades de cada tema aplicando LDA al Corpus #3

Corpus #3: Temas	TP	FN	FP	TN
Tema 1 Economía	121	279	298	902
Tema 2 Ciencia	110	290	273	927
Tema 3 Deportes	113	287	231	969
Tema 4 Política	121	279	333	867

Tabla 11. Valores de TP, FN, FP, TN de los cuatro temas aplicando LDA al Corpus #3

Corpus #3: Temas	Precisión	Cobertura	Valor F1
Tema 1 Economía	0.289	0.302	0.295
Tema 2 Ciencia	0.287	0.275	0.281
Tema 3 Deportes	0.328	0.282	0.304
Tema 4 Política	0.267	0.302	0.283
Macromedia	0.29275	0.29025	0.29075

Tabla 12. Métricas extrínsecas de evaluación de los cuatro temas aplicando LDA al Corpus #3

Perplejidad	Coherencia	Coherencia-T1	Coherencia-T2	Coherencia-T3	Coherencia-T4
1646.77424	-190.26235	-205.74257	-202.10538	-203.51763	-149.68382

Tabla 13. Métricas intrínsecas de evaluación de los cuatro temas aplicando LDA al Corpus #3 [T1=Economía; T2=Ciencia; T3=Deportes;T4=Política]

4.2 Resultados de PLSA

Aplicamos PLSA a los mismos corpus de tres tamaños. Al examinar el listado de los mejores N-gramas de cada tema, hemos descubierto que, a diferencia del modelo LDA, el PLSA produce muchas palabras con una probabilidad de 1, seguidas por otras con distintas probabilidades ordenadas de mayor a menor. En la Tabla 14 se presenta una parte del listado al aplicar PLSA al Corpus #1. Hemos de destacar que los N-gramas con probabilidad 1 que se encuentran delante de otros con la misma probabilidad no implica que sean más representativos, sino que simplemente están ordenados alfabéticamente. En

realidad, el hecho de que un cuantioso grupo de palabras tenga una misma probabilidad dificulta nuestro proceso de exploración visual. Por lo tanto, hacemos que el modelo devuelva todas las palabras con una probabilidad de valor 1 y exploramos la totalidad para interpretar el tema.

TEMA 1 (deportes)		**TEMA 2 (economía)**		**TEMA 3 (política)**		**TEMA 4 (ciencia)**	
abortion	1	accept	1	abandon	1	above-normal	1
account	1	accord	1	abate	1	accountable	1
acknowledge	1	accuse	1	academy	1	accumulation	1
afghanistan	1	adapt	1	achieve	1	activism	1
aim	1	adopt	1	acid	1	administrator	1
…		…		…		…	
economy	0.83929	see	0.79299	inflation	0.82556	federal	0.76763
demand	0.8001	warn	0.76663	official	0.7777	image	0.76763
include	0.8001	coach	0.7418	health	0.75802	olympic	0.75413
campaign	0.78413	concern	0.7418	effort	0.7579	team	0.72931
package	0.75012	rise	0.73987	voter	0.73978	criticize	0.68787
europe	0.73664	ease	0.66652	oil	0.6947	quarter	0.68787
labor	0.7315	gas	0.66652	datum	0.67621	research	0.68787
pandemic	0.72952	start	0.66652	call	0.67607	china	0.68773
loss	0.70046	bring	0.65698	milestone	0.67607	capitol	0.67851
face	0.69744	cup	0.65698	moon	0.67607	identify	0.67851
risk	0.66681	day	0.65698	pressure	0.67607	map	0.67851
temperature	0.66681	open	0.65698	energy	0.67082	potential	0.67851
williams	0.66681	program	0.65698	clean	0.67013	suggest	0.67851
leader	0.65713	raise	0.65698	attorney	0.65461	disruption	0.6785
market	0.65713	swiss	0.65698	earth	0.65461	week	0.6549
domestic	0.64492	threat	0.65698	gop	0.65461	political	0.62297

Tabla 14. Una parte del listado de las palabras y sus probabilidades de cada tema aplicando PLSA al Corpus #1

En el experimento con el Corpus #1, asignamos los temas de deportes, economía, política y ciencia a los cuatro agrupamientos de documentos. Durante la interpretación de temas nos encontramos con un problema similar al experimento con LDA. La distribución irregular de los N-gramas representativos en los grupos generados por el modelo dificulta la determinación de los temas.

Por ejemplo, muchas palabras relacionadas con la economía, como "*economy*", "*labor*", "*market*" y "*recession*", aparecen con alta probabilidad. Sin embargo, anotamos que en el listado se incluyen dos asociaciones deportivas: "*NCAA*" (en inglés *National Collegiate Athletic Association*) y "*NFL*" (en inglés *National Football League*), que normalmente solo se mencionan en las noticias de deportes.

La misma estrategia se usó al identificar el tema de la ciencia. En vez de explorar las palabras más frecuentes, los términos específicos del campo ayudaron más a identificar el tema, aunque cuentan con una frecuencia mucho más baja en los textos. Estos resultados también están en la misma línea que el concepto de la puntuación de TF-IDF, que hemos discutido en la Sección 2.3.2. Las palabras del tema de economía y política siguen estando mezcladas. Hemos decidido que el tema 3 sea la de política por la alta frecuencia de "*voter*", "*attorney*" y "*justice*", entre otras palabras.

Los resultados de las métricas extrínsecas de evaluación de los resultados producidos por PLSA tras aplicarse al Corpus #1 se presentan en Tabla 15, en la cual vemos que las macromedias de las métricas precisión, cobertura y valor F1 son 0,2745, 0,275 y 0,27475, respectivamente. Los valores resultan más bajos que los obtenidos por el modelo LDA en el mismo corpus. Por ejemplo, tomando el valor F1 como una referencia, el PLSA da un rendimiento de 14,92% peor que el LDA.

Corpus #1: Temas	**Precisión**	**Cobertura**	**Valor F1**
Tema 1 Deportes	0.224	0.22	0.222
Tema 2 Economía	0.308	0.32	0.314
Tema 3 Política	0.26	0.26	0.26
Tema 4 Ciencia	0.306	0.3	0.303
Macromedia	0.2745	0.275	0.27475

Tabla 15. Métricas extrínsecas de evaluación de los cuatro temas aplicando PLSA al Corpus #1

Se presentan los valores de las métricas intrínsecas de PLSA en Corpus #1 en Tabla 16. Vemos que existe una gran diferencia entre estos valores con los obtenidos en la prueba con LDA. Asimismo, se obtienen las coherencias del tema 2 y 3 con valor 0.

Perplejidad	Coherencia	Coherencia-T1	Coherencia-T2	Coherencia-T3	Coherencia-T4
2.22899	1.00000	0.00000	0.00000	1.00000	3.00000

Tabla 16. Métricas intrínsecas de evaluación de los cuatro temas aplicando PLSA al Corpus #1 [T1=Deportes; T2=Economía; T3=Política;T4=Ciencia]

La identificación de los temas en el Corpus #2 es más clara. Las palabras como "*sun*", "*telescope*" y "*species*" sirven de ayuda para definir el tema de la ciencia para el primer agrupamiento de documentos. En el tema 2, las palabras que encabezan la lista están relacionadas con la política. Por ejemplo, "*defense*", "*vote*" y "*campaign*" suelen aparecer en textos políticos. "*Inflation*", "*economy*", "*market*" y "*job*" de la tercera lista nos dan pista para determinar que el tema 3 es economía. Por último, las palabras "*football*", "*soccer*" y "*mile*" indican que es muy probable que el tema 4 sea deportes (Tabla 17).

Tras calcular la precisión, la cobertura y el valor F1 del modelo PLSA aplicado al Corpus #2 (Tabla 18), descubrimos un aumento obvio en comparación con la prueba realizada con el Corpus #1. Las métricas de precisión, cobertura y Valor F1 promediadas han llegado a 0,40325, 0,4125 y 0,4075, respectivamente. En comparación con el Corpus #1, suben un 46,9%, 50% y 48,32%, respectivamente. Entre los cuatro temas, el tema de economía produce los mejores resultados.

TEMA 1 (ciencia)		TEMA 2 (política)		TEMA 3 (economía)		TEMA 4 (deportes)	
above-normal	1	abortion	1	abate	1	abandon	1
absent	1	acceleration	1	abu	1	abruptly	1
abuse	1	accept	1	accelerate	1	accumulation	1
academy	1	acceptance	1	accelerator	1	accuse	1
acquire	1	accessible	1	achieve	1	acid	1
…		…		…		…	
live	0.87362	defense	0.99752	globally	0.99997	zero-tolerance	0.99999
justice	0.85025	begin	0.88099	gay	0.99951	glimpse	0.99996
advance	0.83158	focus	0.85983	inflation	0.96329	shanghai	0.99993
bob	0.83158	move	0.84943	economy	0.95601	legislature	0.99591
spread	0.83158	person	0.84103	yield	0.92651	social	0.9912
fail	0.82846	vote	0.83429	labor	0.92585	astronaut	0.87295
winner	0.79725	delay	0.82413	market	0.91823	democrat	0.85911
career	0.7944	campaign	0.80693	regional	0.91798	force	0.85867
triple	0.7944	paris	0.80549	job	0.90735	football	0.84074
investigation	0.78899	wnba	0.80549	decline	0.8944	mile	0.82869
sun	0.78033	ask	0.80351	global	0.87862	emerge	0.82142
telescope	0.77308	national	0.80351	economic	0.87573	concern	0.81001
species	0.76072	slowdown	0.80351	demand	0.86483	age	0.80138
multiple	0.75157	speak	0.80351	feed	0.84892	soccer	0.79644
approach	0.74764	yellow	0.80351	risk	0.84416	woman	0.79599
confirm	0.74764	measure	0.79409	chinese	0.81492	coronavirus	0.78527

Tabla 17: Una parte del listado de las palabras y sus probabilidades de cada tema aplicando PLSA al Corpus #2

Corpus #2: Temas	Precisión	Cobertura	Valor F1
Tema 1 Ciencia	0.348	0.345	0.347
Tema 2 Política	0.33	0.315	0.322
Tema 3 Economía	0.644	0.715	0.678
Tema 4 Deportes	0.291	0.275	0.283
Macromedia	0.40325	0.4125	0.4075

Tabla 18: Métricas extrínsecas de evaluación de los cuatro temas aplicando PLSA al Corpus #2

Perplejidad	Coherencia	Coherencia-T1	Coherencia-T2	Coherencia-T3	Coherencia-T4
2.40861	-11.66993	-2.16993	-7.00000	-15.50978	-22.00000

Tabla 19: Métricas intrínsecas de evaluación de los cuatro temas aplicando PLSA al Corpus #2 [T1=Ciencia; T2=Política; T3=Economía;T4=Deportes]

En el experimento con el Corpus #3, asignamos los temas de política, ciencia, economía, deportes a los cuatro grupos (Tabla 20). El proceso de interpretación ha sido más fácil con el último corpus. PLSA produce los mejores resultados entre las seis pruebas (Tabla 21). Las macromedias de las métricas extrínsecas de precisión, cobertura y Valor F1 alcanzan 0,539, 0,541 y 0,538, respectivamente. Corresponden a un aumento de 33,66%, 25,7% y 32,02%, respectivamente, comparando con los resultados del Corpus #2.

TEMA 1 (política)		**TEMA 2 (ciencia)**		**TEMA 3 (economía)**		**TEMA 4 (deportes)**	
above-normal	1	abnormal	1	abate	1	abraham	1
abrupt	1	abnormality	1	able	1	absence	1
abundant	1	abramovich	1	abruptly	1	abundance	1
accelerator	1	academic	1	absent	1	accept	1
accessible	1	actin	1	abu	1	accidental	1
…		…		…		…	
measure	0.90514	authorization	0.99999	tax	0.9999	write	0.99996
turn	0.90406	pentagon	0.99997	stability	0.99974	stage	0.99716
emission	0.89972	structural	0.99382	charles	0.99819	team	0.96869
arizona	0.89081	aid	0.97354	inflation	0.98377	game	0.96205
pick	0.88774	characterize	0.95059	cool	0.98332	discussion	0.92206
congressional	0.88554	successful	0.94582	economy	0.97151	bill	0.90362
attorney	0.87384	brain	0.91302	price	0.97126	play	0.89969
performance	0.85375	research	0.90044	membership	0.96491	break	0.8958
tree	0.84484	age	0.89894	economist	0.93462	title	0.87937
supreme	0.83779	variant	0.8932	slow	0.93202	serve	0.87892
announce	0.83112	insight	0.88809	raise	0.9297	olympic	0.87121
department	0.81911	mission	0.88043	central	0.92067	pro	0.86973
name	0.81632	molecule	0.87363	quarter	0.90804	majority	0.86928
legal	0.81156	leg	0.87126	package	0.90087	win	0.86409
massive	0.81138	dynamics	0.86163	survey	0.89986	professional	0.8625
relative	0.80839	protein	0.86117	trade	0.8938	fan	0.85956

Tabla 20: Una parte del listado de las palabras y sus probabilidades de cada tema aplicando PLSA al Corpus #3

Corpus #3: Temas	Precisión	Cobertura	Valor F1
Tema 1 Política	0.393	0.375	0.384
Tema 2 Ciencia	0.615	0.515	0.561
Tema 3 Economía	0.662	0.762	0.708
Tema 4 Deportes	0.486	0.512	0.499
Macromedia	0.539	0.541	0.538

Tabla 21: Métricas extrínsecas de evaluación de los cuatro temas aplicando PLSA al Corpus #3

Perplejidad	Coherencia	Coherencia-Tema 1	Coherencia-Tema 2	Coherencia-Tema 3	Coherencia-Tema 4
2.54301	-25.86074	-10.00000	-22.58496	-22.09474	-48.76327

Tabla 22: Métricas intrínsecas de evaluación de los cuatro temas aplicando PLSA al Corpus #3 [T1=Política; T2=Ciencia; T3=Economía;T4=Deportes]

5 DISCUSIÓN

5.1 Tamaño de los corpus y efectividad de LDA y PLSA

Analizamos por separado el efecto de las dos variables en este experimento en los resultados finales: el tamaño del conjunto de datos para el entrenamiento y las distintas técnicas del modelado de temas.

En primer lugar, extraemos los valores de macromedias de LDA y PLSA para analizar los cambios de tendencia de sus rendimientos al aumentar el tamaño de corpus (Figura 9). Hemos visto que, en este experimento, el modelo PLSA es mucho más sensible al tamaño del corpus que LDA. Examinando los resultados de LDA, se observa que la precisión, la cobertura y el valor F1 han descendido ligeramente al crecer el conjunto de datos. En cambio, descubrimos una correlación positiva entre los valores de las métricas y el tamaño del corpus en la prueba con PLSA. Aunque el rendimiento de PLSA en el Corpus #1 queda inferior al LDA, obtiene una mejora evidente al aplicarse en el Corpus #1 y #2.

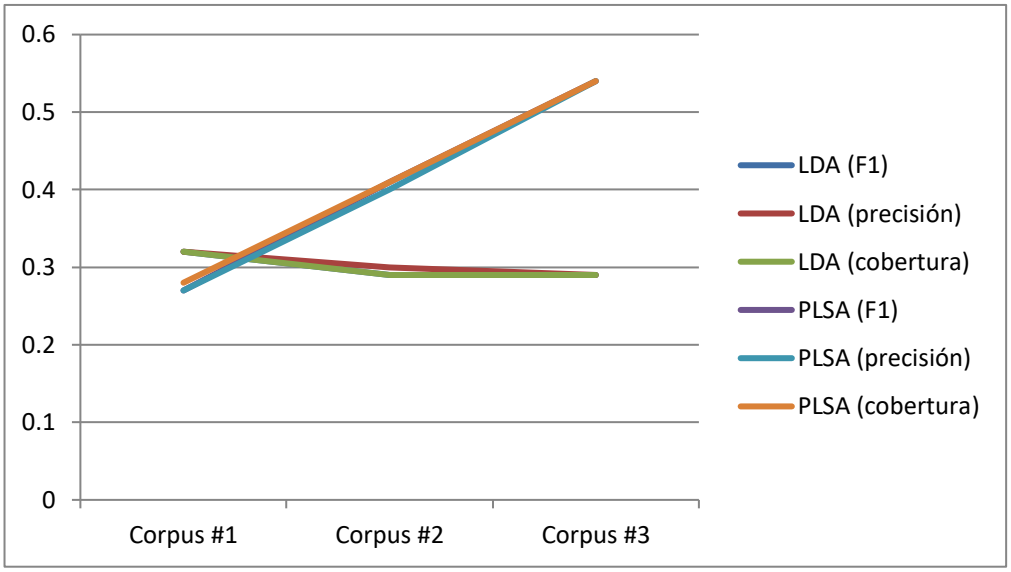


Figura 9. Valores obtenidos de las macromedias

En segundo lugar, representamos los valores de las métricas intrínsecas de

LDA y PLSA en la Figura 10. Vemos que la perplejidad de LDA muestra un aumento considerable al crecer el tamaño del corpus mientras la coherencia casi mantiene en el mismo nivel. Por el contrario, los valores de la perplejidad y la coherencia de PLSA oscilan muy ligeramente alrededor de 0. En este caso, estas dos métricas no aportan mucho conocimiento para comparar la eficacia de los modelos.

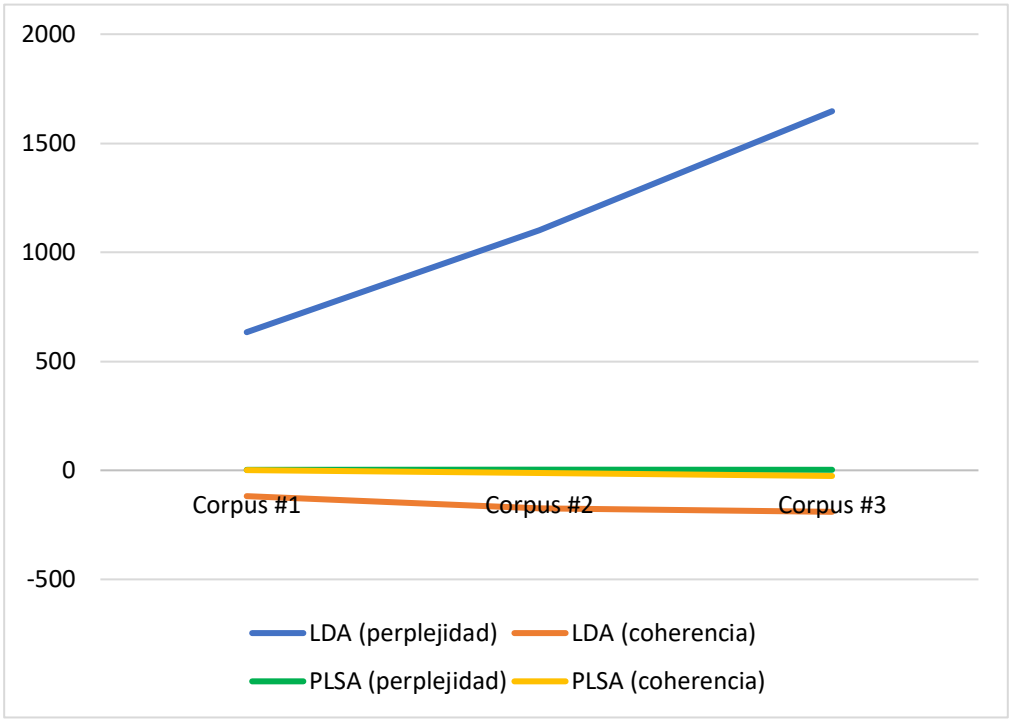


Figura 10. Valores obtenidos de perplejidad y coherencia

Por lo tanto, basado en los datos experimentales, principalmente en las métricas extrínsecas, podemos concluir que el tamaño del corpus ejerce mayor influencia en la efectividad de PLSA que LDA.

5.2 Efectividad de LDA y PLSA en un mismo corpus

Comparamos las métricas de evaluación de LDA y PLSA al aplicarse a los mismos corpus. A partir de los resultados de las métricas extrínsecas,

observamos que, excepto la prueba con el Corpus #1, PLSA supera a LDA en las otras dos colecciones de datos. En la tarea de modelado de textos, suponemos que el coste de los falsos positivos es similar al coste de los falsos negativos. Es decir, las consecuencias que causan los documentos erróneamente agrupados en un tema y los perdidos que deberían estar en este grupo son parecidas. Por eso, nos concentramos en comparar las macromedias del valor F1 de los modelos. Aplicados al Corpus #1, LDA obtiene un valor F1 de 0,31575 mientras que PLSA devuelve un valor de 0,27475. En las pruebas con el Corpus #2 y #3, los valores de F1 de PLSA son 40,76% y 85,04% más elevados que los de LDA, respectivamente.

En resumen, basándonos en los resultados de este experimento, concluimos que, comparado con LDA, PLSA demuestra una mejora considerable en el rendimiento a medida que aumenta el tamaño del corpus. También, PLSA produce una categorización más precisa de textos que LDA.

5.3 Comparación con estudios previos

En Blei et al. (2003), donde se propuso por primera vez el modelo LDA, se indicó que, desde los principios matemáticos, a PLSA le falta un modelo probabilístico generativo para procesar los listados de números que representan los documentos. En cambio, LDA muestra una ventaja competitiva en el cálculo de una representación semántica de baja dimensión con granularidad fina mediante la eliminación del problema de sobreajuste del PLSA. Estos autores realizaron tres experimentos de modelado de temas, clasificación de documentos y filtración colaborativa. En la tarea de clasificación, a diferencia del experimento que se efectuó en el presente trabajo, hicieron una clasificación binaria utilizando el conjunto de datos Reuters-21578, que contiene 8.000 documentos y 15.818 palabras. En vez de comparar directamente el rendimiento de PLSA y LDA, estimaron los parámetros de un LDA en todos los documentos sin aportar las verdaderas etiquetas de clase. Luego, entrenaron una máquina de vectores de

soporte (SVM, del inglés *support vector machine*) en las representaciones de baja dimensión proporcionadas por LDA y compararon esta SVM con otra SVM entrenada en todas las características de palabras. Los resultados indicaron una reducción ligera en el rendimiento de clasificación al usar las características basadas en LDA. No obstante, también manifestaron que estos resultados necesitarían más comprobación porque en la mayoría de las pruebas el rendimiento se había mejorado por las características de LDA. En los experimentos del modelado de temas y de filtración colaborativa, LDA funcionó mejor que PLSA, dando una perplejidad más baja.

Desde entonces, se han publicado estudios teóricos que analizan las conexiones entre el LDA y el PLSA. Sin embargo, la comparación en la mayoría de los estudios se basa en la probabilidad de los datos retenidos (Blei et al., 2003; Blei y Lafferty, 2005; Steyvers y Griffiths, 2007; Wallach et al., 2009). Se han efectuado pocas comparaciones empíricas entre las implementaciones de los dos algoritmos. En Chang et al. (2009), se centraron en cuantificar la interpretabilidad de los temas y en la comparación de manera cuantitativa de los significados semánticos inferidos por LDA y PLSA. En concreto, entrenaron los modelos PLSA, LDA y CTM (del inglés *correlated topic model*) en dos corpus: una colección de 8.447 artículos de *New York Times* desde el año 1987 hasta el 2007, que cuenta con 8.269 palabras únicas, y un conjunto de textos de *Wikipedia* con 15.273 palabras únicas. Comparando dos métricas predictivas, la verosimilitud predictiva logarítmica y el rango predictivo, CTM obtuvo los mejores resultados, seguido de LDA y PLSA.

En un estudio comparativo sistemático entre PLSA y LDA, Lu et al. (2011) analizaron el rendimiento de ambos en tres tareas representativas en la minería de textos: el modelado de temas, en el que cada texto pertenece exactamente a una clase, la categorización de textos, donde cada texto se clasifica para una de las categorías predeterminadas y la recuperación de información, que consiste en devolver informaciones relacionadas con una consulta. Adoptaron dos corpus: TDT2 y Reuters-21578. El primero consta de 11.201 documentos que se

clasifican en 96 categorías semánticas. El segundo contiene 21.578 documentos que se agrupan en 135 categorías. Sus resultados experimentales mostraron que, cuando se optimiza, LDA es mejor que PLSA en la tarea de categorización, donde requiere una representación semántica latente de granularidad fina que es generalizable desde los datos de entrenamiento hasta los de prueba. En las tareas de modelado de temas y de recuperación de información, los dos modelos tienden a comportarse de manera similar.

Después de comparar estos estudios previos con el presente trabajo, la diferencia entre los resultados experimentales se puede deber a los siguientes factores:

a) El tamaño de los corpus. Observamos que, en el experimento de Lu et al. (2011), la tasa de error en la clasificación binaria se reduce a medida que aumenta el tamaño de los datos de entrenamiento. Se considera que la riqueza de las características de las palabras está correlacionada positivamente con el tamaño del corpus. Los estudios mencionados anteriormente utilizaron los grandes corpus normalizados que están disponibles en bases de datos públicas. Por el contrario, el tamaño limitado de los corpus que hemos creado para este trabajo puede afectar al resultado final de los modelos. Hipotetizamos que LDA tendería a proporcionar mejores resultados que PLSA si los corpus fueran mucho más extensos.
b) Número de iteraciones. Los modelos producen resultados ligeramente diferentes cada vez que se ejecuta el código. En nuestro experimento hemos mostrado los datos que se devolvieron en la primera ejecución. Sin embargo, no podemos garantizar que los modelos proporcionen constantemente un rendimiento estable. Será interesante hacer una investigación en el futuro sobre cuántas iteraciones son adecuadas para lograr un rendimiento estable. También podremos comparar la estabilidad de los modelos ejecutándolos con distintos números de iteraciones en las pruebas.

c) Tipo de tareas. Aunque PLSA, LDA y sus extensiones se han aplicado con éxito en muchas tareas de la minería de textos, hay pocos conocimientos sobre cómo afecta el tipo de tareas al rendimiento y cómo se pueden optimizar los modelos para las diversas tareas.

d) Sesgos ocurridos al interpretar los temas. LDA y PLSA son capaces devolver las palabras más importantes para cada agrupamiento de documentos, ordenadas por sus probabilidades. Además, PLSA ha detectado algunas palabras con una probabilidad de 1, las cuales hemos ordenado alfabéticamente. Eso se debe al poco poder discriminatorio del modelo cuando se emplea con corpus de estos tamaños. Como la máquina solo se encarga de agrupar los textos en cuatro clases sin comprender realmente el significado de las etiquetas predeterminadas de antemano (i.e. economía, política, ciencia y deportes), se requiere una labor manual que relacione las clases 1, 2, 3, 4 con los listados de las palabras más representativas en cada clase. Percibimos cierta dificultad y confusión en este proceso por varias razones. Por un lado, sabemos que los algoritmos de aprendizaje automático requieren una cantidad masiva de datos para el entrenamiento. En el caso del modelado de temas, si se prueban con corpus muy extensos, se generarán las palabras más representativas desde las cuales sería más fácil detectar patrones. En nuestro experimento, creamos diversos corpus relativamente pequeños. Como resultado, en ocasiones observamos que las distintas clases comparten palabras representativas. Por otro lado, como cada texto se considera una mezcla de temas, cuanto más diferentes sean los temas, más lejos se encontrarán en el espacio de distribución temática y, por tanto, menos confusa será la interpretación. En nuestro caso, las cuatro etiquetas que seleccionamos reflejan una cierta relación entre los temas. De hecho, la línea que separa los artículos de temática económica de los de temática política es a menudo bastante difusa.

6 CONCLUSIÓN

Para este estudio, se ha llevado a cabo una investigación sistemática de las tres técnicas más representativas en el modelado de temas, i.e. LSA, PLSA y LDA. En el ámbito de la minería de textos y PLN, el modelado de temas es una tarea frecuentemente utilizada para descubrir las estructuras semánticas latentes en una colección de textos.

En la sección del marco teórico, se ha revisado brevemente la historia y el estado actual de la inteligencia artificial en general, y del aprendizaje automático, la minería de textos y el PLN en particular.

Con respecto a nuestro estudio comparativo, se han analizado las teorías matemáticas básicas sobre el modelado de temas, tras lo cual se han evaluado las ventajas y los inconvenientes de cada una de ellas. El modelado de temas parte de una hipótesis distribucional, suponiendo que cada documento se compone de una mezcla de temas y cada tema se compone de un conjunto de palabras. Por una parte, LSA proporciona un fundamento teórico a PLSA, LDA y sus extensiones. La idea es descomponer la matriz término-documento en la matriz documento-tema y la matriz tema-término con la técnica de descomposición en valores singulares. Los vectores de documentos, términos y temas permiten capturar la similitud semántica entre los distintos documentos. Por otra parte, a partir de PLSA, se empieza a introducir un método probabilístico para abordar el problema de la alta dimensión de la matriz término-documento original. El objetivo es encontrar un modelo probabilístico con temas latentes que sea capaz de generar los datos observados en la matriz. Finalmente, LDA se considera una versión bayesiana de PLSA. Adopta el priori Dirichlet en las distribuciones documento-tema y tema-término, que permite una mejor generalización del modelo cuando se enfrenta con nuevos documentos.

Se ha realizado un experimento de agrupamiento de documentos según sus temas en los dos modelos probabilísticos, PLSA y LDA. Los resultados se han evaluado con métricas extrínsecas (i.e. precisión, cobertura y valor F1) y

métricas intrínsecas (i.e. perplejidad y coherencia). Los datos experimentales obtenidos han mostrado que PLSA produce mejores resultados que LDA y es más sensible al crecimiento del tamaño del corpus. Es decir, se ha notado una mejora de rendimiento en PLSA a medida que crece el tamaño del corpus mientras que LDA no muestra una reacción obvia a esta variable. Pese a que la conclusión no coincide con la mayoría de las afirmaciones publicadas en estudios previos sobre el rendimiento de los dos modelos, hemos reflexionado sobre los posibles factores que podrían haber afectado al resultado de la evaluación. En este sentido, el tamaño reducido de los corpus utilizados en las pruebas, el número de iteraciones ejecutadas, el tipo de tarea a la que se aplican los modelos y los sesgos que podrían haber sucedido durante la fase de interpretación de los temas son las variables potenciales que podrían haber influido en la evaluación.

La limitación principal de nuestro estudio radica en el reducido tamaño de las colecciones de los textos utilizadas en la evaluación de los modelos. Así pues, un importante trabajo futuro sería seguir comprobando las observaciones en experimentos donde intervengan corpus a mayor escala.

Concluyo este trabajo indicando las principales líneas futuras de investigación que se pueden derivar. Por ejemplo, este estudio se ha concentrado exclusivamente en LSA, PLSA y LDA, las tres técnicas más básicas del modelado de temas. En los trabajos futuros, sería interesante investigar y evaluar otros modelos variados o extendidos, por ejemplo, NMF (del inglés *Non-negative Matrix Factorization*), BERTopic y Top2Vec.

Además, nuestro experimento se centra en la comparación del rendimiento de estos tres modelos. El proceso interno de cómo tratan y calculan los datos es invisible. Otra línea sería descomponer los modelos a analizar a través de sus hiperparámetros, donde los algoritmos de estimación afectarán al resultado.

REFERENCIAS BIBLIOGRÁFICAS

Agrawal, A., Fu, W. & Menzies, T. (2018). What is wrong with topic modeling? And how to fix it using search-based software engineering. *Information and Software Technology, 98*, 74-88.

Alnusyan, R., Almotairi, R., Almufadhi, S., Shargabi, A. A. & Alshobaili, J. (2020). A semi-supervised approach for user reviews topic modeling and classification. In *2020 International Conference on Computing and Information Technology*, 1-5.

Andrews, N. O. & Fox, E. A. (2007). Recent developments in document clustering. *Computer Science Technical Reports*.

Atkins, S., Clear, J. & Ostler, N. (1992). Corpus design criteria. *Literary and linguistic computing, 7*(1), 1-16.

Berry, M. W. (1992). Large-scale sparse singular value computations. *The International Journal of Supercomputing Applications, 6*(1), 13-49.

Berry, M. W., Dumais, S. T. & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM review, 37*(4), 573-595.

Biber, D. (1993). Representativeness in corpus design. *Literary and linguistic computing, 8*(4), 243-257.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM, 55*(4), 77-84.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of machine learning research, 3*, 993-1022.

Blei, D.M. & Lafferty, J. (2006). Correlated topic models. *Advances in neural information processing systems*, *18*, 147.

Brants, T., Chen, F. & Tsochantaridis, I. (2002). Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the eleventh international conference on Information and knowledge management*, 211-218.

Cao, Y. T., Pruksachatkun, Y., Chang, K. W., Gupta, R., Kumar, V., Dhamala, J. &

Galstyan, A. (2022). On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. *arXiv:2203*.13928 [cs.CL].

Caruana, R. & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, 161-168.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory, 2*(3), 113-124.

Chowdhary, K. R. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649.

Christopoulos, A. & Lew, M. J. (2000). Beyond eyeballing: fitting models to experimental data. *Critical Reviews in Biochemistry and Molecular Biology, 35*(5), 359-391.

Cohn, D. & Chang, H. (2000). Learning to probabilistically identify authoritative documents. In *ICML*, 167-174.

Cohn, D. & Hofmann, T. (2000). The missing link-a probabilistic model of document content and hypertext connectivity. *Advances in neural information processing systems, 13*.

Das, A. S., Datar, M., Garg, A. & Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, 271-280.

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

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological), 39*(1), 1-22.

Dillon, T., Wu, C. & Chang, E. (2010). Cloud computing: issues and challenges. In *24th IEEE international conference on advanced information networking and applications*, 27-33.

Dumais, S. T. (2004). Latent semantic analysis. *Annual review of information science and techonology, 38*(1), 188-230.

Dumais, S. T. (2007). LSA and information retrieval: Getting back to basics. In Landauer, T. K., McNamara, D.S., Dennis, S. & Kintsch, W (Eds.), *Handbook of latent semantic analysis* (pp. 305-334). Psychology Press.

Egger, R. & Yu, J. (2022). A topic modeling comparison between LDA, NMF, Top2Vec, and BERTopic to demystify twitter posts. *Frontiers in Sociology, 7*(886498).

Feldman, R., Dagan, I., & Hirsh, H. (1998). Mining text using keyword distributions. *Journal of Intelligent Information Systems, 10*, 281-300.

Feng, Y. & Lapata, M. (2010). Topic models for image annotation and text illustration. In *Human Language Technologies: The 2010 annual conference of the north American chapter of the association for computational linguistics*, 831-839.

Foltz, P. W. & Landauer, T. K. (2007). Helping People Find and Learn from Documents: exploiting Synergies Between Human Andcomputer Retrieval with Supermanual. In Landauer, T. K., McNamara, D.S., Dennis, S. & Kintsch, W (Eds.), *Handbook of latent semantic analysis* (pp. 335-356). Psychology Press.

Francis, W. N. & Kucera, H. (1979). Brown corpus manual. *Letters to the editor, 5*(2), 7.

Frawley, W. J., Piatetsky-Shapiro, G. & Matheus, C. J. (1992). Knowledge discovery in databases: An overview. *AI magazine*, 13(3), 57.

Friedrich, S., Antes, G., Behr, S., Binder, H., Brannath, W., Dumpert, F., Ickstadt, K., Kestler, H. A., Lederer, J., Leitgöb, H., Pauly, M., Steland, A., Wilhelm, H. & Friede, T. (2022). Is there a role for statistics in artificial intelligence? *Advances in Data Analysis and Classification, 16*(4), 823-846.

Gilks, W. R., Richardson, S. & Spiegelhalter, D. J. (1995). Introducing Markov chain Monte Carlo. In Gilks, W. R., Richardson, S. & Spiegelhalter, D. J. (Eds.), *Markov chain Monte Carlo in practice* (pp. 1-16). CRC Press.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A. & Louwerse, M. M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, y Computers, 36*, 180-192.

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

Heo, G. E., Kang, K. Y., Song, M. & Lee, J. H. (2017). Analyzing the field of bioinformatics with the multi-faceted topic modeling technique. *BMC bioinformatics*, *18*, 45-57.

Hirschberg, J. & Manning, C. D. (2015). Advances in natural language processing. *Science, 349*(6245), 261-266.

Hofmann, T. & Puzicha, J. (1999). Latent class models for collaborative filtering. In *Proceedings of IJCAI, 99*(1999).

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*, 50-57.

Hong, L. & Davison, B. D. (2010). Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, 80-88.

Hu, X., Cai, Z., Wiemer-Hastings, P., Graesser, A. C. & McNamara, D. S. (2007). Strengths, limitations, and extensions of LSA. In Landauer, T. K., McNamara, D.S., Dennis, S. & Kintsch, W (Eds.), *Handbook of latent semantic analysis* (pp. 413-438). Psychology Press.

Hu, Y., Boyd-Graber, J., Satinoff, B. & Smith, A. (2014). Interactive topic modeling. *Machine learning*, *95*, 423-469.

Jin, X., Zhou, Y. & Mobasher, B. (2004). Web usage mining based on probabilistic latent semantic analysis. In *Proceedings of the tenth ACM SIGKDD*

international conference on Knowledge discovery and data mining, 197-205.

Jo, T. (2018). *Text mining.* Springer.

Jordan, M. I. & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.

Kherwa, P. & Bansal, P. (2019). Topic modeling: a comprehensive review. *EAI Endorsed transactions on scalable information systems, 7*(24).

Landauer, T. K. & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review, 104*(2), 211.

Landauer, T. K., Foltz, P. W. & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes, 25*(2-3), 259-284.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444.

Lee, K. D., Han, K. & Myaeng, S. H. (2016). Capturing word choice patterns with LDA for fake review detection in sentiment analysis. In *Proceedings of the 6th international conference on Web intelligence, mining and semantics*, 1-7.

Leech, G. (1992). Corpora and theories of linguistic performance. *Directions in corpus linguistics*, 105-122.

Lin, D. & Pantel, P. (2001). DIRT @SBT@discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining,* 323-328.

Liu, L., Tang, L., Dong, W., Yao, S. & Zhou, W. (2016). An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus, 5*(1608), 1-22.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development, 1*(4), 309-317.

Manning, C. D., Raghavan, P. & Schütze, H. (2008). *Introduction to information*

retrieval. Cambridge University Press.

Markov, Z. & Larose, D. T. (2007). *Data mining the Web: uncovering patterns in Web content, structure, and usage*. John Wiley y Sons.

Martin, D. I. & Berry, M. W. (2007). Mathematical foundations behind latent semantic analysis. In Landauer, T. K., McNamara, D.S., Dennis, S. & Kintsch, W (Eds.), *Handbook of latent semantic analysis* (pp. 35-55). Psychology Press.

McCarthy, J. (2004). *What is artificial intelligence?* [PDF file]. https://www.diochnos.com/about/McCarthyWhatisAI.pdf

McNamara, D. S., Boonthum, C., Levinstein, I. B. & Millis, K. (2007). Evaluating self-explanations in iSTART: Comparing word-based and LSA algorithms. In Landauer, T. K., McNamara, D.S., Dennis, S. & Kintsch, W (Eds.), *Handbook of latent semantic analysis* (pp. 227-241). Psychology Press.

Meyer, C. F. (2002). *English Corpus Linguistics: An Introduction*. Cambridge University Press.

Millis, K., Magliano, J., Wiemer-Hastings, K., Todaro, S. & McNamara, D. S. (2007). Assessing and improving comprehension with latent semantic analysis. In Landauer, T. K., McNamara, D.S., Dennis, S. & Kintsch, W (Eds.), *Handbook of latent semantic analysis* (pp. 207-225). Psychology Press.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill Science.

Moor, J. (2006). The Dartmouth College artificial intelligence conference: The next fifty years. *AI Magazine, 27*(4), 87-87.

Moore, G. E. (1998). Cramming more components onto integrated circuits. *Proceedings of the IEEE, 86*(1), 82-85.

Moreno, A., Armengol, E., Béjar Alonso, J., Belanche Muñoz, L.A., Cortés García, C.U., Gavaldà Mestre, R., Gimeno, J.M., Martín Muñoz, M. & Sànchez-Marrè, M. (1994). *Aprendizaje automático*. Edicions UPC.

Nadkarni, P. M., Ohno-Machado, L. & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics*

Association, 18(5), 544–551.

Ostrowski, D. A. (2015). Using latent Dirichlet allocation for topic modelling in twitter. In *Proceedings of the 2015 IEEE 9th international conference on semantic computing*, 493-497.

Paatero, P. & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics, 5*(2), 111-126.

Periñán-Pascual, C (2017). Bridging the gap within text-data analytics: A computer environment for data analysis in linguistic research. *Revista de lenguas para fines específicos 23*(2), 111-132.

Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology, 2*(1), 37-63.

Rashid, S., Shakeel, R., Bashir, H., Malik, K. & Wajib, K. (2016). Moore's Law Effect on Transistors Evolution. *International Journal of Computer Applications Technology and Research, 5*(7), 495 – 499.

Russell, S. & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach (4th ed.)*. Pearson.

Salton, G. (1971). The SMART system. *Retrieval Results and Future Plans,* 260.

Salton, G., Wong, A. & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613-620.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development, 3*(3), 210-229.

Sejnowski, T. J. (2018). *The Deep Learning Revolution*. MIT Press.

Solomonoff, R. J. (1985). The time scale of artificial intelligence: Reflections on social effects. *Human Systems Management, 5*(2), 149-153.

Steyvers, M. & Griffiths, T. (2007). *Probabilistic topic models.* In Handbook of latent semantic analysis, 439-460. Psychology Press.

Streeter, L. A., Lochbaum, K. E., LaVoie, N. & Psotka, J. E. (2007). Automated Tools for Collaborative Learning Environments. In Landauer, T. K.,

McNamara, D.S., Dennis, S. & Kintsch, W (Eds.), *Handbook of latent semantic analysis* (pp. 554-575). Psychology Press.

Torres López, C. & Arco García, L. (2016). Representación textual en espacios vectoriales semánticos. *Revista Cubana de Ciencias Informáticas*, *10*(2), 148-180.

Turing, A. M. (2009). *Computing machinery and intelligence*. Springer, Dordrecht.

Turney, P. D. & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research, 37*, 141-188.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. *Proceedings of the 23rd international conference on machine learning*, 977-984.

Wallach, H. M., Murray, I., Salakhutdinov, R. & Mimno, D. (2009). Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, 1105-1112.

Wang, X. & McCallum, A. (2006). Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*, 424-433.

Witten, I. H., Frank, E. & Hall, M. A. (2005). Practical machine learning tools and techniques. *Data Mining, 2*(4).

Wu, X., Zhu, X., Wu, G. Q. & Ding, W. (2013). Data mining with big data. *IEEE transactions on knowledge and data engineering, 26*(1), 97-107.

Zhang, Y., Jin, R. & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, *1*, 43-52.

Zhao, R. & Mao, K. (2017). Fuzzy bag-of-words model for document representation. *IEEE transactions on fuzzy systems, 26*(2), 794-804.

Zhou, S., Zhao, Y., Rizvi, R., Bian, J., Haynos, A. F. & Zhang, R. (2019). Analysis of Twitter to identify topics related to eating disorder symptoms. In *2019 IEEE international conference on healthcare informatics*, 1-4.

Zhou, X., Tao, X., Rahman, M. M. & Zhang, J. (2017). Coupling topic modelling in opinion mining for social media analysis. In *Proceedings of the international conference on web intelligence*, 533-540.

ANEXOS

Anexo 1: Códigos para implementación de PLSA, LDA y evaluación de resultados

```
string path = @"C:\Users\DLA\Desktop\advanced\";
string fileName = path + "corpus_small.txt";
string stopwordsFile = @"C:\Users\DLA\Desktop\experiment\eng_functional.txt";
string resourcesPath =
@"C:\Users\DLA\Desktop\CSharpScripting\Release\resources\";
string corpus;
Files.ReadText(fileName, out corpus);
DataTable matrix;
  Ngrams.GetDocNgramMatrix(corpus,  "eng",  "multiple",  "unigram",  "lexeme",
"frequency", stopwordsFile, null, resourcesPath, out matrix);
int numTopic = 4;
string topicNgramMatrix, docTopicMatrix, topicProbability, evaluation;
LDA.Apply(matrix, "ID", numTopic, out topicNgramMatrix, out docTopicMatrix, out
topicProbability, out evaluation);
//PLSA.Apply(matrix, "ID", numTopic, out topicNgramMatrix, out docTopicMatrix, out
topicProbability, out evaluation);
DataTable dt1;
DataMisc.ConvertCSVToTable(docTopicMatrix, out dt1);
Dictionary<string, string> topics;
TopicModel.GetBestTopicForDocs(dt1, out topics);
Editor.Show(topics);
Editor.Show(Environment.NewLine + Environment.NewLine);
Editor.Save("id" + "\t" + "predicted", path + "topics.csv");
Editor.AppendToFile(topics, path + "topics.csv");
Editor.Save(topicNgramMatrix, path + "topicNgramMatrix.csv");
DataTable dt2;
DataMisc.ConvertCSVToTable(topicNgramMatrix, out dt2);
Dictionary<string, double> ngrams;
for(int x = 1; x <= numTopic; x++)
{
   string topic = "topic" + x.ToString();
   TopicModel.GetBestNgramsForTopic(dt2, topic, 20, out ngrams);
   Editor.Show(topic.ToUpper());
   Editor.Show(ngrams);
   Editor.Show(Environment.NewLine);
   Editor.Save(ngrams, path + topic + ".csv");
}
```

```
34  Editor.Show(topicProbability + Environment.NewLine + Environment.NewLine +
evaluation);
35 DataTable finalDataset;
36 Files.ReadDataTable(path + "corpus_small_modificado.csv", out finalDataset);
37 DataTable topics;
38 Files.ReadDataTable(path + "topics_modificado.csv", out topics);
39 string[] newValues;
40 Datatable.GetFieldValues(topics, "predicted", out newValues);
41   Datatable.InsertFieldandValues(finalDataset,   "predicted",   newValues,   out
finalDataset);
42 int[] predicted, expected;
43 ConfusionMatrix.GetValues(finalDataset, "predicted", "expected", out predicted,
out expected);
44 int TP, FN, FP, TN;
45 ConfusionMatrix.GetValues(predicted, expected, out TP, out FN, out FP, out TN);
46 double precision, recall, F1;
47 ConfusionMatrix.GetPrecision(TP, FN, FP, TN, out precision);
48 ConfusionMatrix.GetRecall(TP, FN, FP, TN, out recall);
49 ConfusionMatrix.GetFScore(TP, FN, FP, TN, out F1);
50 Editor.Show("precision: " + precision.ToString() + Environment.NewLine + "recall:
" + recall.ToString() + Environment.NewLine + "F1: " + F1.ToString());
51 Editor.Show(Environment.NewLine);
52 Editor.Show("TP: " + TP.ToString() + Environment.NewLine + "FN: " + FN.ToString()
+ Environment.NewLine + "FP: " + FP.ToString() + Environment.NewLine + "TN: " +
TN.ToString());
```