TRABAJO DE FIN DE MÁSTER

exTRAE: Clasificación de tuits dirigidos a la RAE mediante las herramientas de modelado de tópicos LSA y LDA

Por

Jose María Hernández de la Cruz
Bárbara Saiz Escobar

UNIVERSIDAD COMPLUTENSE MADRID

Máster Universitario en Letras Digitales: Estudios Avanzados en Textualidades Electrónicas

Facultad de De Filología Y Facultad de Informática

Dirigido por
Rafael Caballero Roldán
Adrián Riesco Rodríguez

Curso académico: 2020-2021
Convocatoria: julio 2021
Calificación: 9.5

exTRAE: Clasificación de tuits dirigidos a la RAE mediante las herramientas de modelado de tópicos LSA y LDA

Memoria que se presenta para el trabajo de fin de máster

Jose María Hernández de la Cruz
Bárbara Saiz Escobar

Tutores:

Rafael Caballero Roldán
Adrián Riesco Rodríguez

En colaboración con:

REAL ACADEMIA ESPAÑOLA

Facultad de Filología
Facultad de Informática
Universidad Complutense de Madrid

Madrid, a 14 de julio de 2021

Resumen

Este trabajo pretende ofrecer un método informático que clasifique los tuits recibidos por la Real Academia Española (RAE) en su cuenta de Twitter @RAEinforma. Dada la enorme cantidad de tuits que muestran sus dudas sobre cuestiones lingüísticas, es necesario implementar métodos informáticos que ayuden al manejo de tales datos. A lo largo de este trabajo, trataremos de vislumbrar cuál es la actual situación de las instituciones en el ámbito digital y pondremos especialmente el foco en la RAE y su labor en Twitter. Luego, explicaremos *topic modeling* y dos de sus métodos: Latent Semantic Allocation y Latent Dirichlet Allocation. Ambos serán empleados para la clasificación de un corpus de más de nueve mil tuits. Concluiremos llevando a cabo un *test* con el que comprobar el éxito de los resultados.

Palabras clave: comunicación institucional; LDA; LSA; Real Academia Española; *topic modeling*; Twitter.

Abstract

This dissertation aims to provide a programming method to classify the tweets received by the Spanish Royal Academy (RAE) throughout its Twitter account @RAEinforma. Due to the enormous quantity of tweets wondering about linguistic inquiries, it is necessary to implement computer methods that help to manage these data. Throughout the paper, we will shed some light on institutions' current digital situation, specifically focusing on RAE and its role on Twitter. Then, we will get immersed into topic modeling and the two of its methods: Latent Semantic Allocation and Latent Dirichlet Allocation. We will apply them to a corpus of more than nine thousand tweets. A final test using added tweets will be performed to check if our results turn out to be successful.

Keywords: institutional communication; LDA; LSA; Royal Spanish Academy; topic modeling; Twitter.

Índice general

Página

Capítulo 1

Introducción

La incesante cantidad de información que ocupa nuestra vida es, desde el punto de vista humano, inabarcable. Los dispositivos digitales que usamos a diario nos abren las puertas a un campo infinito de información mostrada de diversas maneras: textos, imágenes, vídeos, etc. Una de las características principales de estos dispositivos es facilitarnos el acceso a la redes sociales, un tipo de páginas web que promueven las interacciones entre usuarios. Estas gozan de un número sumamente elevado de participantes. Ellos crean y consumen altas cantidades de información que pueden resultar relevantes a empresas e instituciones. El manejo de tales cantidades de datos supone un reto para estos organismos, por lo que poseer herramientas automáticas resulta ineludible. Es en esta clasificación y cuantificación de datos donde se enmarca nuestro Trabajo de Fin de Máster.

A lo largo del apartado introductorio, nos esforzaremos en explicar en qué consiste nuestro trabajo, qué nos ha llevado a escribirlo, cuáles son sus objetivos, de qué manera lo hemos llevado a cabo, cómo se divide y dónde encaja en la imperiosa necesidad de manejar la ingente cantidad de información.

1.1. Motivación

Enmarcado dentro del Reto Digital Consumer[1] de Telefónica y la Real Academia Española, la propuesta nos animaba a crear una herramienta informática que ayudara a clasificar y cuantificar el número de tuits recibidos por la Real Academia Española a través de su perfil en la red social Twitter. Dado el perfil del Máster Universitario en Letras Digitales de la Universidad Complutense de Madrid, nuestra ambición era poner al servicio de esta novedosa tarea nuestros conocimientos como filólogos hispánicos, así como aquel aprendizaje adquirido a lo largo del máster que resultara útil para el desarrollo del trabajo. Hemos tratado de hacer valer los talleres de Python proporcionados durante el curso académico, y de estudiar por cuenta propia, no sin la ayuda y guía de nuestros tutores, el uso de otras herramientas relevantes y necesarias. En el proceso de investigación y aprendizaje descubrimos nuevas bibliotecas y *software* que aumentan las posibilidades de obtener un trabajo profesional y, sobre todo, fructífero. De este modo, entusiasmados por

[1] *Telefónica* Telefónica. 22 de diciembre de 2020. «La Real Academia Española y Telefónica avanzan en su colaboración en inteligencia artificial con retos lingüísticos para dispositivos del hogar e investigación académica». Consultado: 2021-07-06.

haber sido seleccionados por dos organismos tan importantes y respetados, pusimos todo nuestro esfuerzo en la elaboración de este Trabajo de Fin de Máster.

1.2. Objetivos

El trabajo nace con el objetivo transversal de descubrir y clasificar los temas más comunes entre las consultas realizadas en Twitter a la cuenta oficial de la Real Academia Española. Partiendo de este objetivo central, nos planteamos el resto de metas que favorecían al logro del propósito principal. Estos objetivos se dividen en dos grupos: los de corte teórico y los de ejecución práctica.

Los objetivos teóricos se focalizan en conocer la situación actual de las instituciones en el plano de la comunicación digital. Un entorno que ha ganado un peso incalculable en los últimos años, haciendo que la adaptación a ellos sea una necesidad clave a la hora de justificar la relevancia del trabajo. Y, más concretamente, en la necesidad de describir la situación actual en el plano digital de la institución de referencia del español en el mundo: la Real Academia Española. La manera en la que este organismo se ha adaptado a los nuevos medios en los últimos años ha sido clave en la percepción de los hispanohablantes. Las consultas a las diversas secciones de sus webs, la creación de espacios como «Enclave» o su diccionario en línea resultan altamente relevantes. Así como su cuenta en Twitter, en la que ofrece un servicio único y personalizado para la resolución de dudas lingüísticas. El servicio, ampliamente utilizado según atestigua el número de interacciones recibidas y el número de seguidores del que disfrutan (un millón novecientos mil)[2], goza de un prestigio sin parangón. Su supremacía lingüística en este contexto se demuestra comparando sus datos con los de otras academias y diccionarios: Cambridge[3] cuenta con 221.1 mil seguidores en Twitter, mientras que el Trésor de la Langue Française Informatisé ni siquiera tiene presencia oficial en la plataforma. Por ende, la relevancia de la Real Academia Española en el panorama lingüístico digital es innegable.

A partir de estas interacciones y dudas trasladadas a la cuenta de @RAEinforma se configura el objetivo transversal de nuestro proyecto del que derivan los objetivos prácticos. Estos son los que constituyen la ambiciosa meta de nuestro trabajo: la clasificación de las consultas recibidas en forma de tuit por la Real Academia Española. Ante las incesantes dudas recibidas, el organismo precisa de una herramienta informática que le ayude a identificar sobre qué preguntan más los usuarios. Para lograr esa clasificación, necesitamos una técnica de procesamiento del lenguaje natural. En nuestro caso, optamos por el entrenamiento de dos algoritmos de *topic modeling* capaces de realizar la tarea. Para ello, elegimos los modelos LSA y LDA, sobre los que se perseguía el objetivo práctico final: clasificar y cuantificar las consultas recibidas por la Real Academia Española en Twitter para poder responder a preguntas como estas: ¿cuáles son las dudas más frecuentes?; ¿es posible agrupar las dudas de alguna manera para ofrecer respuestas automáticas? Estas cuestiones precisaban de un contexto teórico-práctico consolidado.

[2] *@RAEinforma* en Twitter. Consultado: 2021-07-06.
[3] *@CambridgeWords* en Twitter. Consultado: 2021-07-06.

1.3. Metodología

Ante un trabajo de gran envergadura, definir la metodología desde el principio resulta ser un paso clave. El primer estadio supuso realizar una extensa búsqueda sobre los diversos temas que íbamos a tratar: comunicación institucional, la Real Academia Española, Twitter, *topic modeling*, LSA y LDA. Con ello, formaríamos una idea de cuál era la situación actual, así como de los antecedentes y hacia dónde enfocar específicamente nuestra herramienta. Para la realización de esta primera tarea nos servimos del *software* Citavi6, una útil herramienta para el manejo bibliográfico. De manera paralela, iniciamos la programación en Python de nuestra herramienta. Para ello, usamos tanto el *software* Jupyter de Anaconda como Google Colab, entornos que permiten la codificación en Python de manera visual, siendo el segundo de ellos una herramienta colaborativa en línea. Para el manejo del lenguaje natural, la informática ofrece una serie de técnicas para su procesamiento. De entre todas ellas, decidimos emplear la técnica *topic modeling*, una herramienta conformada por distintos modelos que, a grandes rasgos, extraen automáticamente temas de un gran conjunto de documentos. De los modelos que *topic modeling* proporciona, optamos, instados por nuestros tutores, por Latent Semantic Allocation (LSA) y Latent Dirichlet Allocation (LDA). Ambos modelos han sido ampliamente usados para fines similares. Entre ellos, comparten características, aunque difieren en la implementación de ciertos pasos, lo que provoca algunas diferencias en los resultados.

La segunda etapa vino marcada por el refinamiento de las más de cuatrocientas citas que hasta el momento manteníamos alojadas en Citavi6: complementar los metadatos o añadir nuevos artículos fueron algunas de las tareas de este segundo estadio del proyecto. Simultáneamente, recopilamos los tuits y refinamos los códigos de LSA y LDA hasta obtener resultados que encajaran con nuestras expectativas y objetivos.

En el estadio final del proyecto, llevamos a cabo, en primer lugar, la elaboración de esta memoria en LaTeX usando Overleaf[4], donde se incluye el análisis de los resultados. Por otro lado, el segundo y último paso consistió en la elaboración de una página web HTML a partir de una plantilla Bootstrap en la que mostrar los resultados de manera profesional.

1.4. Organización del estudio

El estudio consta de varias partes. La presente introducción esboza cuál ha sido la motivación del trabajo, así como los objetivos y organización del mismo. Al tratarse de un trabajo colaborativo entre dos personas, concluir esta sección con el apartado 1.5 «Contribuciones» resulta ineludible. Añadimos el apartado «Enlaces de interés», en el que mostramos el hipervínculo a nuestra página web.

A partir de ese momento comenzamos a definir el estado de la cuestión. El trabajo aglutina dos campos bien diferenciados, por lo que decidimos crear dos grandes secciones para los dos estados de la cuestión. En el primero definimos cuál es la situación actual de las instituciones en el ámbito digital, focalizándonos especialmente en la labor de la Real Academia Española y su actuación en la red social Twitter. En el segundo estado de la cuestión aludimos a la técnica del procesamiento del lenguaje natural seleccionada para

[4] *Overleaf* https://www.overleaf.com/project/6061d127df8b0811c6079f6f

este trabajo, poniendo el foco en el aspecto teórico de los algoritmos utilizados: LSA y LDA.

En la tercera parte de nuestro trabajo desarrollamos nuestras herramientas informáticas. Haciendo uso de la codificación en Python, buscamos la manera de crear una herramienta provechosa y que cubriera los objetivos. Luego, ofrecemos una relación de trabajos afines con los que mostramos la amplitud de temas en los que se puede aplicar *topic modeling*.

Por último, resumimos el perfil institucional del trabajo así como mostramos una comparativa de los resultados y nuestras impresiones sobre los mismos a modo de conclusión.

1.5. Contribuciones

Realizado por Jose María Hernández de la Cruz y Bárbara Saiz Escobar, el trabajo ha sido dividido en dos partes complementarias. Para dejar claro quién hizo qué y de esta manera ajustarnos al reglamento de la Universidad Complutense de Madrid, ofrecemos la siguiente relación de tareas:

Jose María Hernández

El integrante del equipo colaboró de manera equitativa en la búsqueda bibliográfica y su incorporación a Citavi6 sobre todos los temas tratados en las dos secciones de «Estado de la cuestión», así como de los documentos relativos a LSA (sección 2 y sección 3). En los siguientes estadios del proceso, se encargó al cien por cien del manejo de Citavi6: el refinamiento de citas, la incorporación de metadatos restantes, nuevas entradas bibliográficas y depuración del texto de las citas antes de su exportación. En cuanto a programación, tal y como se decidió en un primer momento, el integrante asumió la responsabilidad de llevar a cabo el código relativo a LSA, al que dedicó la mayor parte del tiempo que se prolongó el proyecto. Hernández se encargó también de buscar la plantilla para la web final, a la que decidió llamar «exTRAE» y de la que creó su logotipo (véase 1.1) y estructura inicial. Esta plantilla Bootstrap fue una tarea posteriormente delegada en exclusiva a Saiz. En cuanto a la redacción de la memoria, ha redactado parte de la presente introducción y de las conclusiones, así como el capítulo 2, exceptuando el apartado relativo a Twitter. Del capítulo tercero escribió *topic modeling* y el apartado teórico de LSA. En el capítulo 4, dedicado a «Nuestra contribución» redactó la explicación del código LSA. Modificó la plantilla LaTeX en las siguientes secciones: cubierta, portada, estructuración de contenidos y bibliografía.

Bárbara Saiz

Al igual que su compañero, la integrante colaboró de manera equitativa en la búsqueda bibliográfica y su incorporación a Citavi6. Sus búsquedas recogieron igualmente todos los aspectos del «Estado de la cuestión», rescatando citas relativas al modelo LDA. En el apartado de programación, se decidió que se encargara del código LDA, en el que centró la mayor parte del tiempo dedicado al proyecto. Una vez su compañero hubo depurado la plantilla Bootstrap para la realización de la web, esta pasó a ser en exclusiva una tarea

propia. Su labor consistió en modificar ampliamente los archivos HTML y CSS, salvaguardando en todo momento el aspecto profesional de la misma. En su empresa, añadió, eliminó y modificó secciones y apartados. En cuanto a la redacción de esta memoria, se ha encargado de redactar el apartado 2.1 «Twitter» dentro del capítulo 2, así como de la parte teórica de LDA, complementada con diversas figuras. Asimismo, explicó su código LDA en el capítulo cuarto, investigó los distintos trabajos relaciones expuestos en el capítulo 5 y participó en labores de diseño del documento elaborado en LaTeX así como en parte de la presente introducción y de las conclusiones.

Puntos comunes

La corrección y depuración de la memoria que aquí se presenta ha sido realizada colaborativamente por ambos integrantes. Igualmente, las decisiones relativas a estructura y estilo del resultado final han sido compartidas y consensuadas. Del mismo modo, la introducción y la conclusión fueron redactadas a la par con el fin de plasmar la visión de ambos participantes sobre el proyecto, así como otros apartados presentes en esta redacción que no han sido mencionados como tarea exclusiva de ninguno de los dos estudiantes.

1.5.1. Enlaces de interés

Nuestra página web, previamente mencionada, está almacenada en un repositorio de Github y puede consultarse en el siguiente enlace: `https://josemh301.github.io/extrae/index.html`

Figura 1.1: Logotipo e hipervínculo de **exTRAE**

Su nombre, exTRAE, no es solo una referencia a la sustracción de tópicos a partir de un corpus, sino que contiene la «T» que alude a Telefónica, y RAE, acrónimo de «Real Academia Española». El logotipo, mostrado en la figura 1.1, se compone de múltiples puntos, inspirado en la nueva imagen de la compañía tecnológica.

Los enlaces a los dos códigos de programación se encuentran disponibles en la introducción a la sección 4 «Nuestra contribución». Ambos están alojados en cuadernos de Google Colab.

Capítulo 2

Estado de la cuestión: instituciones

2.1. Twitter

La presencia de las redes sociales es prácticamente ineludible en nuestros días. Entre todas las disponibles, Twitter presenta unas características muy marcadas que no provocan el rechazo de los usuarios, sino que mantienen su posición como la red social donde la información, el ocio, la política e incluso la lengua ven un altavoz con el que atraen a millones de tuiteros. Es pues, una red social caldo de cultivo para ver el sentir de la población. Por ello, es la plataforma seleccionada de la que obtener nuestra base de datos. En el siguiente apartado mostraremos algunas de sus peculiaridades.

2.1.1. Fenómeno de comunicación digital

Twitter nace en 2006 como consecuencia de varios proyectos cuyo fin último y fundamental era mejorar la comunicación y el intercambio de información en línea ya fuese por vía telefónica o web. En su creación participaron desde 2004 diversos programadores y creadores comenzando por Noah Glass, el fundador de Odeo: un recurso con el que buscaba dar la posibilidad de grabar mensajes en formato MP3 al llamar por teléfono y que estos se quedaran alojados en la Nube. Más tarde, el creador de Blogger[1], Evan Williams, se une al proyecto llegando a aportar, junto a Jack Dorsey, la clave para reconducir y modernizar la compañía: usar SMS para crear conversaciones para grupos pequeños de personas. Como consecuencia a estos proyectos, la red social Twitter nació con carácter de *microblogging* con la colaboración conjunta de Jack Dorsey, Noah Glass, Biz Stone y Evan Williams por las aportaciones de sus trabajos. La idea original tuvo lugar en la compañía Odeo que trabajaba en un servicio de radio en línea a modo de *podcast* cuyo éxito se vio truncado por productos de la competencia.

En 2006, la red social se establece como un servicio interno de la idea original de Glass con el nombre de «Twittr», pero no fue hasta 2007 cuando Twitter se convierte en una empresa independiente con Dorsey a la cabeza. Los usuarios y el tráfico de información evolucionaban positivamente y sus rasgos característicos se fueron definiendo con mayor nitidez: la etiqueta (*hashtag*) y el límite de 140 caracteres por publicación ya eran signos propios de la plataforma.

[1] *Blogger* `https://www.blogger.com/about/?bpli=1`

En un primer momento, este *microblogging* lo usaron los empleados de Odeo. En 2007 la popularidad de la Twitter consiguió que cambiara de categoría: fue premiada como blog. Ante el reconocimiento y la evidente evolución, Twitter se formó como empresa independiente.

En 2008 y con Jack Dorsey como presidente de la compañía, el equipo de Twitter estaba formado por una plantilla de dieciocho personas (Naveira). En 2009, el número de trabajadores se había multiplicado y el crecimiento de la empresa se orientaba hacia el panorama internacional: añaden publicidad y se crean las versiones en otros idiomas.

Su éxito fue tan notable que sufrieron ciertos desajustes, como colapsos en el funcionamiento de la red y más problemáticas en la dirección de la empresa. Pese a todo, la revolución informativa que conllevaba el surgimiento de Twitter hizo imparable su existencia que, actualmente, podemos catalogar como exitosa: «Hasta la actualidad . . . cuenta con 330 millones de usuarios activos mensuales en todo el mundo, de los que 4.9 millones son españoles» (Naveira).

El uso principal y clave de la plataforma por parte de los usuarios es el de fuente de información por su instantaneidad de acuerdo con el Décimo estudio de redes sociales en España llevado a cabo por Elogia (Naveira).

Esa instantaneidad ha conseguido que Twitter se establezca como una red social de referencia a nivel mundial a la que acuden millones de usuarios para comentar cualquier noticia, evento, programa de televisión o, incluso, para lo que en este trabajo nos compete, comunicarse con instituciones para realizar consultas, observaciones o expresar sus opiniones relacionadas con el trabajo de estas. Dado su éxito y crecimiento favorable, los estudios sobre Twitter son ya numerosos y enfocados desde múltiples disciplinas tales como «Twitter y la comunicación política» (Campos-Domínguez), «Uso de Twitter y Facebook por los medios iberoamericanos» (García de Torres) o «Twitter. Marketing personal y profesional» (Carballar). Entre otros a los que se suman los mencionados más adelante en el capítulo 5 dedicado a «Trabajos afines» sobre Twitter como entorno para el modelado de tópicos. Sin embargo, esos signos característicos de la plataforma han seguido limitando el trabajo analítico de la información que comparte y almacena.

Incluyendo las características topológicas de Twitter como red social o medio de prensa, los tuits también pueden también ser utilizados para seguir las impresiones compartidas sobre eventos en tiempo real o como indicadores del pronóstico de taquilla de nuevas películas según las opiniones de los usuarios, entre otros. No obstante, y a pesar de las distintas aplicaciones que puedan existir para el uso de esta fuente de información, el manejo de las grandes cantidades de datos en Twitter para el análisis y los distintos usos aún es limitado debido a las particularidades de la propia red social: mensajes cortos existentes en grandes cantidades, con lenguaje informal (Chandía Sepúlveda 2). Por esta razón se han llevado a cabo distintos trabajos, como el aquí presente y otros relacionados que se tratarán en la sección dedicada a investigaciones afines, que proponen entrenar las distintas técnicas de procesamiento y análisis del lenguaje natural para conseguir encontrar tópicos, clasificarlos y analizarlos para distintos fines, asegurando la relevancia de los resultados al trabajar con una de las redes sociales más populares entre los usuarios. Una red social que brinda al investigador una muestra útil, variada e importante a nivel de análisis del lenguaje desde dos perspectivas: la estrictamente lingüística y la fundamentalmente estadística y computacional.

2.1.2. Almacén de información

De acuerdo con el origen, la evolución y la relevancia adquirida por Twitter, el número de usuarios y el formato de comunicación digital han dado lugar a una plataforma cuyo funcionamiento establece sus publicaciones (tuits) como objeto de análisis. Para trabajar con los textos breves creados, publicados y almacenados en Twitter como datos e información hemos de atender a los distintos elementos relevantes para el escenario de nuestra investigación que forman parte del comportamiento de la plataforma.

Cuentas

La red social basa su funcionamiento en la publicación de un mensaje corto dirigido al resto de usuarios. Estos se identifican desde una cuenta anónima o no, pudiendo interactuar con su publicación a través de sus propias cuentas.

Tuits

El mensaje corto publicado desde la cuenta de un usuario permitía en los inicios una extensión máxima de 140 caracteres. Actualmente, esa extensión ha incrementado a 260, manteniendo su carácter de texto breve. La visibilidad de ese tuit depende del número de seguidores de la cuenta, del número de republicaciones (retuits) que alcance y las respuestas y menciones que aporte el creador desde su publicación a modo de hilos de conversación.

Etiquetas o *hashtags*

En esa visibilidad participan las etiquetas que se adjuntan al tuit. Estas se identifican con una almohadilla que precede a la palabra o frase más repetida en un conjunto de tuits cuyo flujo está teniendo lugar en un momento concreto, pudiendo llegar a convertirse en tendencia o *Trending Topic* (TT) dependiendo del número de participación de cuentas y tuits en la etiqueta en concreto.

Las etiquetas también pueden crearse manualmente para facilitar la identificación de un tuit con un determinado hilo de conversación o con un tema de discusión en concreto.

Estadísticas

Las estadísticas en Twitter muestran la relevancia de una cuenta: número de seguidores, correspondencia entre seguidores y seguidos, interacciones por tuit, menciones y ratio de evolución. Este ratio de evolución se refiere a lo que recibe también el nombre de ratio de interacción: un indicador incluido en el servicio de analíticas de Twitter que informa sobre el crecimiento de cuentas, sobre su alcance a nivel de usuarios dentro de la plataforma o sobre la cantidad de acciones que reciben sus tuits (me gustas, clics, respuestas, etc.). Entre otros datos relevantes, estos nos dan una visión de si la cuenta puede pertenecer a una institución, a una persona famosa o a un tuitero distinguido. Todas ellas son informaciones que pueden caber en un estudio analítico con Twitter como escenario principal de las muestras.

Tuits, usuarios y etiquetas pueden organizarse en listas o momentos. Las listas se crean como un grupo de cuentas de Twitter que van a conversar sobre una temática especí-

fica. Puede ser pública, dejando que cualquiera interesado en el título o la descripción pueda seguirla y participar o; privada, dándole así una función más propia de la idea originaria de Dorsey apoyada por Williams: pequeños grupos de personas intercambiando mensajes.

Por otro lado, los momentos de Twitter están diseñados para la consulta y la organización de tuits por temas. Según el día, la plataforma ofrece distintos momentos personalizados para cada usuario. Son temas populares o relevantes que favorecen a la instantaneidad informativa de la red social. El usuario puede también crearlos, incluir un tuit propio o eliminarlo. Para nuestra investigación, las cuentas, las etiquetas y, por supuesto, los propios tuits en cuestión han sido los objetos relevantes tanto para el preprocesamiento de los datos como para el análisis último de los resultados. Ambas cuestiones tratadas y fundamentadas en la sección dedicadas a los códigos, sus pasos y sus resultados.

2.1.3. Escenario para la detección de tópicos

Los elementos que conforman Twitter, así como su naturaleza de red social en la que prima la instantaneidad en la comunicación y la obtención de información obliga a que los tuits cuenten con unos actores que los determinan como textos de longitud corta, aún con la reciente extensión en el límite de caracteres. Entre estos actores se encuentran palabras con errores ortográficos y gramaticales, abreviaturas, emoticonos y sintaxis incorrecta o no convencional. Todos ellos hacen de la detección de tópicos un reto aún mayor, pues empeora la generación de coocurrencias entre los distintos términos (Alash y Al-Sultany 2).

Por estas condiciones, Twitter se plantea como un desafío respecto a la tarea de encontrar temas por medio de algoritmos de *topic modeling*. Por el carácter desestructurado de sus textos cortos y por la naturaleza de las palabras que lo componen, los datos que conforman la muestra para nuestro estudio deben ser limpiados hasta proporcionar una estructura y morfología más viable para la tarea a desempeñar por el algoritmo entrenado.

Además de los actores mencionados, los signos de puntuación, los signos de interrogación y exclamación, los signos de mención, etiqueta, los enlaces y demás información que se ha de catalogar como irrelevante para la detección de tópicos que atañe a nuestro análisis se plantean como retos a tratar cuando el estudio se embarca con Twitter como escenario para la recopilación de los datos que conformarán el corpus a trabajar por el algoritmo en cuestión.

Estos retos son relevantes en mayor medida si la detección de tópicos va encaminada a sacar conclusiones no solo informativas, sino también con cierto valor lingüístico, como es el caso de esta investigación, en la que los tópicos detectados definirán temas directamente relacionados con cuestiones sobre la gramática, el léxico o el uso del español en forma de consulta realizadas a través de Twitter a la cuenta oficial de la Real Academia Española.

La recolección de tuits para el corpus generado en nuestra investigación se ha basado en aquellos tuits con mención a la cuenta oficial de la Real Academia Española «@RAEinforma» que utilizan la etiqueta «#dudaRAE» o «#RAEconsultas» y que suponen las distintas dudas planteadas a la Real Academia Española y, en consecuencia, el texto corto objeto de estudio de la investigación aquí expuesta cuya recolección, procesamiento y consiguiente detección de tópicos será explicada a lo largo de las siguientes páginas,

detallando cómo se han atajado los retos que supone la plataforma como escenario para la detección de tópicos.

Como ya hemos indicado, este trabajo pone en común la labor de las instituciones en el marco digital, en concreto de la Real Academia Española, y el análisis de datos masivos empleando procedimientos informáticos. A lo largo de los dos siguientes grandes apartados (los que se refieren al estado de la cuestión), trataremos aspectos de ambos campos.

Veremos en primer lugar la alta relevancia que tiene para las instituciones su estrategia comunicativa. Permanecer actualizados y adaptados a las nuevas circunstancias es vital para su pervivencia. Buscan mantener o incrementar la confianza que el público deposita en ellos, así como ofrecer un contenido de calidad, respaldado e interesante. Estos son, en parte, los objetivos de la Real Academia Española. La institución vela desde sus inicios por el castellano, y ha sido a partir de las últimas décadas cuando se ha debido adaptar a las nuevas tecnologías para lograr ampliar los receptores de su mensaje. Para ello recurrió a Twitter, una red social con unas características muy limitadas que emplean tanto usuarios como empresas, medios de comunicación, gobiernos, etc. Este primer gran apartado dejará paso al segundo, en el que trataremos cuestiones informáticas relativas al *topic modeling*, del que adelantamos que juega un papel fundamental en la clasificación de un volumen masivo de datos y documentos.

2.2. Comunicación institucional

Las instituciones prestan especial atención, más allá de las actividades en las que se especializan, a la manera de difundir su papel y labor entre el grueso de la población:

> La comunicación institucional puede definirse como el tipo de comunicación realizada de modo organizado por una institución o sus representantes, y dirigida a las personas y grupos del entorno social en el que desarrolla su actividad. Tiene como objetivo establecer relaciones de calidad entre la institución y los públicos con quienes se relaciona, adquiriendo una notoriedad social e imagen pública adecuada a sus fines y actividades. (Ballester 210)

Confianza

La manera en la que una institución desempeña su labor comunicativa supone cambios en la percepción entre el público. En palabras de Salazar y Prieto, «el buen trato con el usuario permite que una institución se posicione de manera positiva en la mente del lector» (18). Las instituciones necesitan casi de manera obligatoria de este apartado comunicativo. La no comunicación va contra el carácter móvil y transformador, y tal inmovilidad supone un fracaso (Salazar y Prieto 12).

La relación entre el gran público y las instituciones tiene como base la confianza, definida por Fukuyama como «the expectation that arises within a community of regular, honest and cooperative behavior, based on commonly shared norms, on the part of members of that community» (26). Esta confianza es esencial para mantener el orden social en diversos ámbitos, y su fracaso daña a la institución (Blau 125). De esta manera, Warret et al. afirman que «individuals' willingness to rely on institutions is likely to reduce the uncertainty entailed in their decisions» (293). Es en esta confianza donde tiene lugar

la cooperación institución-público, intercambio con el que se reducen incertidumbres y críticas. La confianza es un factor que la institución presupone del público, explicándose de esta manera el uso previsible que los usuarios puedan hacer de la institución (Warren et al. 293).

Pautas estilísticas

Las instituciones siguen una serie de estrategias comunicativas a la hora de elaborar sus mensajes que mitiguen el perjuicio que se puedan hacer a sí mismas, entre las que se encuentran evitar las connotaciones negativas o el lenguaje coloquial. Además, procuran emplear un tono neutral que desvincule a la institución de ciertos intereses, mostrándose de manera transparente y evitando la polémica. Emplean un lenguaje sencillo sin abreviaturas, comprensible y con el que remiten y aluden a la fuente original de información. Esta estrategia lleva de la mano otra que procura que la institución permanezca en boca del público. El objetivo es ofrecer un contenido actualizado, de calidad, respaldado e interesante (Salazar y Prieto 22).

Etapas

La comunicación institucional se ha servido de los diferentes medios presentes en las distintas épocas históricas: de la imprenta a internet, «pasando por el periódico, la radio, la televisión o el cine» (Ballester 210). En los medios tradicionales como el periódico, «los papeles del comunicador y destinatario aparecen aislados, independientes de las relaciones sociales o situaciones culturales en las que se producen los procesos comunicativos» (Salazar y Prieto 13). Sin embargo, esta relación cambia.

En el año 2004 surge la Web 2.0, que trae consigo el fomento de la participación y la conexión entre usuarios. Esto le valió el sobrenombre de «web social» o «web participativa» (López 34). La nueva web resulta más interpersonal y conversacional, dejando atrás el esquema «emisor-mensaje-receptor» (Ballester 209). De esta semilla nacen años más tarde Twitter, Facebook y otras redes sociales. Las instituciones, conscientes del potencial de estos nuevos portales, trasladan a ellas parte de su estrategia comunicativa. Se abre un lugar donde poder comunicar de manera más segmentada, elaborando mensajes específicos (Castillo 2). Se trata de «un modelo comunicacional totalmente innovador fundado en las redes y la colaboración de los usuarios ... bajo la denominación de "comunicación digital interactiva"» (Franco Romo 167). La nueva web permite conectar a las instituciones con los usuarios en tiempo real, obteniendo ambas gran beneficio. Mediante el diálogo constante con el usuario también se conocen sus gustos y preferencias. «El *feedback* continuo y recíproco fortalece a ambos» (Salazar y Prieto 18). Este tipo de comunicación se establece con el sistema pregunta-respuesta propio de Stack Overflow[2] o la ya inexistente Yahoo Respuestas, aunque el caso que nos ocupa tendrá un matiz: es la institución la voz autorizada para responder, y no otro usuario.

En el medio en línea se produce un intercambio masivo, no solo entre usuarios y la institución, sino entre los propios usuarios. Tiene lugar un fenómeno al que Guy denomina «social overload», donde «a huge amount of information participates in vast amounts of interactions» (511). Sin embargo, para Salazar y Prieto, el volumen de información y usuarios no es importante a la hora de influir, sino que «lo importante es la relevancia

[2] `https://stackoverflow.com/`

y la facultad de influir en la comunidad . . . en definitiva, la capacidad de *engagement* y amplificación del mensaje» (20).

2.3. Real Academia Española

Dentro de las instituciones más relevantes y conocidas en el mundo hispanoparlante se encuentra la Real Academia Española. El organismo regula y describe el uso del español en el mundo. Desde su fundación limpian, fijan y dan esplendor al castellano. Para lograr su propósito, la Real Academia Española marca unos límites en su permisividad que han ido evolucionando.

Con el inicio del siglo XXI la institución busca reinventarse, y bajo el amparo del Estado español, la Real Academia Española inicia una nueva etapa (Rizzo 426). Lauría y López García indican que tras de siglos de prescriptivismo centralizado, la Real Academia Española pretende adaptarse a los nuevos tiempos a través de la denominada Nueva Política Lingüística Panhispánica (NPLP) publicada en 2004. Supone un acercamiento al modelo descriptivo que se combina con el prescriptivo. La norma surge del uso real la lengua por parte de los hablantes (50). Así, la Real Academia Española aspira a mantener su papel regulador. Sin embargo, este cambio parece meramente teórico y no ha bastado para que la institución siga siendo considerada monocéntrica y horizontalista, manteniendo su eje en el español del centro peninsular, y consecuentemente desplazando al resto de variantes hacia la periferia, privilegiando unas formas lingüísticas sobre otras (Bentivegna et al. 353).

La crítica a la institución da cabida a la intrusión de espacios normativos lingüísticos no oficiales surgidos en los últimos años. A este fenómeno ayudó el constante desarrollo de las tecnologías de la información (Rizzo 426). Espacios en línea como WordReference[3] promueven una descentralización institucional de la norma lingüística. Carecen de respaldo por parte de organismos gubernamentales, suponiendo un riesgo para las instituciones oficiales (Bentivegna et al. 351–52).

Frente a ellos, la defensa de la Real Academia Española se vincula directamente al Estado. Es en este contexto donde se crean «las condiciones de la creación de un mercado lingüístico unificado y dominado por la lengua oficial» (Narvaja de Arnoux y Del Valle 2–3). Gozando de tal respaldo, la NPLP aspira a perpetuar la Real Academia Española como la referencia institucional en materia de corrección lingüística. Como parte de este nuevo posicionamiento abierto a hablantes hasta ahora excluidos del español normativo, la Real Academia Española comienza a emplear para su función divulgativa no solo sus publicaciones, sino sus webs y su perfil en Twitter. La reputación se refleja en el alto seguimiento de la Real Academia Española en plataformas digitales, que respaldan a la institución como un lugar de prestigio, como autoridad normativa en el mundo hispanohablante (Lauria y López García 85). Además, en este marco también encontramos la actualización de la página electrónica oficial de la Real Academia Española y la ASALE, la digitalización de archivos, el acceso libre a recursos lingüísticos y la apertura de cuentas en las principales redes sociales (Rizzo 429).

[3] `https://www.wordreference.com/ES/`

2.3.1. Real Academia Española en Twitter

La cuenta oficial de la Real Academia Española en Twitter abre en 2011 y es en octubre de 2012 cuando inicia el servicio de consultas lingüísticas. Se trata de un servicio único respecto a la resolución de dudas lingüísticas que ninguna otra institución presta en Twitter. Surge durante una serie acciones que la Real Academia Española lleva a cabo apoyada por varias empresas privadas y fundaciones. La cuenta se pone a cargo del departamento de «Español al día», un equipo de trabajo creado en 1998 para modernizar la atención de consultas lingüísticas a través de medios electrónicos. De este modo, la Real Academia Española incluye Twitter como canal de comunicación. La apertura del perfil en esta red social no implica un simple cambio de soporte o formato. La Real Academia Española se adapta a un nuevo medio con características fijas, suponiendo un modo distinto de comunicación entre la institución y los usuarios. Implica una relación más cercana con los hablantes. La interacción Academia-usuario posibilita que este último pueda comentar los mensajes producidos por la institución. Hasta el momento, el usuario recibía la norma y era incapaz de proporcionar retroalimentación a la institución de manera rápida y eficaz. Si bien divulgar la norma ha sido la premisa básica de la institución desde sus comienzos en Twitter, no siempre se ha llevado a cabo de la misma manera. Encontramos, al menos, dos etapas comunicativas, entre las que entendemos hubo una transitoria (Rizzo 430).

Etapas

En una primera etapa, los mensajes provienen mayormente de la institución. Publican sobre diez tuits al día con los que difunden normas que mayoritariamente se refieren a cuestiones ortográficas: «uso de mayúsculas y minúsculas, prefijos, tildes, puntuación, abreviaturas, escritura de extranjerismos, de expresiones numéricas, homófonos» (Rizzo 432). Las preguntas son mínimas y la institución opta por la difusión de normas en lugar de la resolución de dudas. Ante las primeras consultas específicas, la institución insta al usuario a redirigir su consulta al Departamento del Español al Día a través de un formulario en la página web de la Real Academia Española. La interacción con los usuarios es prácticamente nula. Se trata de un sistema comunicativo de uno a muchos, implicando en palabras de Rizzo una «despersonalización de la respuesta» (432).

De manera paulatina, las comunicaciones iniciadas por la Real Academia Española se fueron reduciendo al tiempo que se multiplicaron las consultas de los usuarios. Se deja atrás el sistema comunicativo de uno a muchos. De esta manera se permite a los usuarios ser los que inicien el proceso comunicativo mediante el planteamiento de dudas lingüísticas. Los hablantes de distintos puntos geográficos piden información sobre su particular uso del lenguaje o sobre aquellos que desconocen. Se redefine así el discurso normativo. El proceso comunicativo es ahora de uno a uno, ofreciendo la institución un contenido personalizado y obteniendo una cierta cercanía, un carácter más personal (Rizzo 435-42).

Las consultas realizadas por los usuarios tienen como sello característico y obligatorio la etiqueta o *hashtag* #RAEconsultas. De no ser así, la Real Academia Española les indica que reformulen su duda incluyéndola. Por parte de la institución, «the Twitter public profile of the RAE offers a hashtag #dudaRAE ('doubtRAE') through which the RAE answers the questions of Twitter users» (Slemp et al. 11). Bajo estas dos etiquetas se agrupan toda la actividad de la Real Academia Española en Twitter. Su empleo se

debe a que «users who share the same hashtag at most discuss the same topic» (Alash y Al-Sultany 1).

La novedosa y laboriosa estrategia comunicativa adoptada por la Real Academia Española provoca el incremento del número de consultas. Dentro de ellas, muchas son recurrentes. Esto ha llevado a la Real Academia Española a ofrecer unas respuestas estructuradas con unas determinadas características: Para sus tuits de respuesta, @RAEinforma emplea la tercera persona y construcciones impersonales. De esta manera se presenta como discurso legitimado. En ocasiones emplea la fórmula de segunda persona «usted». Los tuits de la Real Academia Española ofrecen información normativa de manera breve, incluyendo en ocasiones ejemplos o explicaciones, además de posibles hipervínculos en los que se detalla la regla de manera extensa (Rizzo 438). No siempre es una sentencia categórica, sino que, matizando el efecto normativo, podemos encontrar expresiones como «se recomienda». La Real Academia Española no responde a todas las consultas, sino solo aquellas que se ajustan a la función que debe cumplir su cuenta en la red social, excusándose «de responder un pedido de información porque no es pertinente para la sección de consultas lingüísticas» (Niklison 20). La Real Academia Española mantiene un objetivo divulgativo orientado a un lector no especializado en temas lingüísticos. Por parte de los usuarios, las particularidades de la red social le obligan, al igual que a la institución, a condicionar su tuit: la limitación de caracteres obliga a sintetizar y recortar, omitiendo cierta información mientras a la vez que otra se pone en valor (Rizzo 445).

Motivo de respuestas

En la actualidad, existen tres motivos para que la Real Academia Española tuitee: responder a los usuarios que inician la interacción y plantean su duda a la institución; mantener las publicaciones originales de la Real Academia Española propias de su primera etapa, que ya no se centran únicamente en el aspecto ortográfico, sino que tiene como referente las dudas recibidas; el último caso se da cuando en la interacción entre dos usuarios que debaten sobre algún aspecto lingüístico, se menciona a la institución para que medie y resuelva la duda (Rizzo 433–34). De esta manera se crea un ecosistema del que ambos, usuarios e institución, se benefician (Szpektor et al. 1249).

Repercusión digital

Con todo ello logra la Real Academia Española incrementar su reputación y confianza a unos niveles realmente altos. A fecha de julio de 2021, @RAEinforma cuenta con un millón novecientos mil seguidores[4]. La Real Academia Española llega de esta manera a una amplia cantidad de hablantes de múltiples procedencias étnicas, culturales o geográficas que tienen en común la inquietud lingüística y los asuntos normativos. La importante diferencia entre la cantidad de seguidores y los usuarios a los que sigue (tan solo doscientos en la fecha mencionada) atestigua el gran alcance, influencia e importancia del perfil, y del acierto que supuso su creación.

[4] *@RAEinforma* en Twitter

Capítulo 3

Estado de la cuestión: procesos informáticos

Una vez hemos discernido cómo las instituciones interactúan en el ámbito digital y hemos valorado en su justa medida que la creación de documentos ha alcanzado un volumen y ritmo vertiginoso, entendemos la necesidad por parte de las instituciones de poseer herraminetas que ayuden al manejo de tales datos. El incremento de los datos derivados del crecimiento de internet presenta varias características: en primer lugar, observamos que ahora se producen de manera ininterrumpida, en mayor número y en medios que hasta hace escasos años no existían: «micro-bloggings such as Twitter are being used to measure ... consumer opinions and people's moods» (Jiménez-Zafra et al. 1). El valor de los datos ha pasado de estar en su contenido a estar en la manera en que se manejan. Por ello, surgen procedimientos informáticos dedicados al análisis de este maremágnum informativo: «machine learning (ML) approaches and natural language processing (NLP) techniques can be very useful to describe the amount of information being micro-blogged» (García y Bertón 2). Así, se nos hace ineludible esbozar en qué consisten los dos métodos enmarcados en *topic modeling* que hemos empleado: LSA y LDA. Ofrecer una visión global sobre los métodos informáticos seleccionados para el análisis de los datos resultará de utilidad a la hora de entender tanto el proceso de análisis como el resultado de los mismos. Definiremos los términos básicos que ayuden a comprender los algoritmos, y los supuestos clave para el análisis de tuits a partir de LSA y LDA. Procedemos pues a tratar de entender de manera teórica en qué consiste el *topic modeling* y sus dos modelos más característicos.

3.1. Topic Modeling

El procesamiento del lenguaje natural (NLP, por sus siglas en inglés) se ocupa de encontrar y aplicar mecanismos informáticos eficaces que permitan el análisis y la investigación del lenguaje natural, es decir, de los idiomas humanos. «Natural language processing (NLP) is a challenging research in computer science to information management, semantic mining, and enabling computers to obtain meaning from human language processing in text-documents» (Jelodar et al. 1). Con este fin, la tecnología ha desarrollado potentes métodos usados para el minado de datos (Shivam Bansal 1). En este trabajo, nos centraremos solo en una técnica del procesamiento del lenguaje natural, el *topic modeling*.

Definición

Topic modeling, al que nos referimos, salvo excepciones, por su voz inglesa debido a la escasez de referencias al tema entre los investigadores que emplean el español, es una técnica que permite la clasificación automática de documentos de manera no supervisada. Esto implica que, en el proceso, la presencia del ser humano es prácticamente innecesaria. No requiere de anotación o etiquetado previo, sino que posibilita organizar un número muy elevado de archivos casi sin intervención humana (Blei 78).

Este conjunto de técnicas justifica su existencia en tres hechos que tienen lugar en la sociedad global de nuestros días: el enorme volumen de datos; y la velocidad con la que nuevos datos se generan y trasmiten; la variedad de temas, formatos y procedencia de estos datos. Esto conforma el conocido *big data*, que se define como conjuntos de datos de tal tamaño que resultan imposibles de obtener, gestionar o analizar con herramientas tradicionales en un periodo de tiempo asumible (González Fernández 172). Para Blei, el desarrollo del modelado de tópicos probabilístico es la solución a la presencia de un sinfín de documentos que el ser humano por sí mismo no puede analizar (77).

El modelado de tópicos se emplea para extraer temas de un corpus (García y Bertón 3): «It is a process to automatically identify topics present in a text object and to derive hidden patterns exhibited by a text corpus» (Shivam Bansal 1). Sus algoritmos son métodos estadísticos que analizan las palabras halladas en vastas colecciones de documentos. La técnica pone su foco de atención en términos que se solapan o aparecen de manera coocurrente. Así, de un documento extenso, se extraen los que en general, en todo el corpus, son más importantes (Qiang et al. 7). *Topic modeling* identifica los temas y las palabras o *tokens* que los conforman (Alash y Al-Sultany 1–2). Según apunta Chandía Sepúlveda, este procedimiento informático ha sido empleado en las últimas décadas en «el tratamiento de textos en la web para inferir, concluir, analizar y comparar datos e información de distinta índole» (7). Para la obtención de estos documentos, existe cierta tendencia a emplear las APIs (características o contenidos de una web que pueden usar otras) (Blei 77).

Con *topic modeling* se pueden analizar un sinfín de diferentes tipos de documentos: correos electrónicos, capítulos de libro e incluso libros o una noticia periodística. Es importante entender que la técnica no comprende el significado de las palabras, sino que realiza su análisis a partir de la manera en las que estas se combinan. Es un proceso basado en la repetición constante hasta que halla de qué manera están colocadas las palabras para dividirlas en temas o «cestas» (Jelodar et al. 32).

Aplicación a textos largos vs. textos cortos

Las técnicas que conforman *topic modeling* han demostrado de manera satisfactoria su eficacia en el análisis de corpus compuestos por documentos largos, como por ejemplo novelas (Mottaghinia et al. 4). Con el nacimiento de las redes sociales y el alto volumen de información que en ellas se maneja, siendo un nicho en el que millones de usuarios vierten sus preferencias y gustos sobre casi cualquier aspecto de su vida diaria, *topic modeling* se ha trasladado al análisis de textos, casi siempre breves, encontrados en este tipo de webs (Qiang et al. 8). De las interacciones halladas en sitos web como Facebook o Twitter se extraen patrones fructíferos (Jelodar et al. 2). Parece existir cierto consenso en que las técnicas de *topic modeling* no están adaptadas a los *short texts*. Sus dos

métodos más representativos, LSA y LDA, fueron pensados en su origen para obtener temas desde documentos largos que pudieran albergar varios temas. Consecuentemente y según apuntan Alash y Al-Sultany, los resultados a la hora de aplicarlos a tuits son mejorables (4).

Los *short texts* comparten ciertas características que dificultan su procesado mediante *topic modeling*: presentan escasez de coocurrencia de términos; cada texto tiene un único tema y las palabras con mayor parecido tendrán una mayor probabilidad de pertenecer al mismo tema; la información es estática, por lo que el modelo tiene dificultades para capturar palabras semánticamente relacionadas. Además, *topic modeling* considera que dos palabras que aparezcan en posición similar en distintos documentos tienen un significado cercano, incrementando la posibilidad de que puedan pertenecer al mismo tema. Y estas premisas parten del hecho base de que un corpus se componga de varios temas bien diferenciados (Qiang et al. 1-2).

Tuits

Los tuits encajan en las características de los *short texts*. Tal y como apuntan García y Bretón, un análisis de estos documentos mediante *topic modeling* puede resultar incompleto:

> To find relevant information or topics is a difficult task since there are millions of daily tweets covering thousands of topics, there is noisy vocabulary (slangs, emoticons, grammar errors), the text is very short (140 characters), and multilingual tweets. Topic detection is a technique for discovering the main topics automatically. (1–2)

Para aumentar la eficacia, los autores proponen atender a tuits que compartan la misma etiqueta. Los tuits con esta característica comparten palabras similares de las que *topic modeling* obtiene los datos a partir de grupos de palabras. Así se percata de la existencia de los distintos temas, distinguiendo e indicando qué *tokens* lo gobiernan, teniendo estos un mayor peso que el resto de las palabras (Alash y Al-Sultany 3). Los *tokens* son aquellas palabras y caracteres almacenados en una cadena que resultan del proceso de *tokenización* explicado en el apartado «Pasos: preprocesado». Las particularidades de los tuits, como el empleo de palabras erróneamente escritas, caracteres irrelevantes, presencia de emoticonos o el uso de una sintaxis no convencional provocan que la coocurrencia de términos sea escasa. Los tuits incluyen puntuación, preposiciones y *stopwords* o palabras vacías. Estas son palabras cuyo significado no solo no es relevante para el procesamiento, sino que puede entorpecer los resultados del mismo. Estas palabras son ser artículos, pronombres, preposiciones, adverbios y algunos verbos que se almacenan en una lista para poder eliminar su aparición en el conjunto de documentos. La lista puede aplicarse en su forma predeterminada facilitada en el paquete utilizado o editada, añadiendo más términos al total. Una vez han sido eliminadas del corpus, reducen la longitud del tuit a aquellas palabras que sí se consideran relevantes para el procesamiento de ambos algoritmos.

Con todo ello y en contra de la opinión de los investigadores, es posible obtener resultados satisfactorios en la aplicación de LSA y LDA a tuits. Así lo intentamos demostrar con el presente trabajo.

Pasos

Si bien ya hemos indicado en qué consiste *topic modeling* y sus aparentes problemas a la hora de analizar textos breves, aún no hemos indicado los pasos comunes que sus modelos siguen. Un análisis basado en *topic modeling* consta de varios pasos: el preprocesado, la creación de un diccionario y calcular la coherencia son pasos comunes en la aplicación de las técnicas.

Pasos: preprocesado

El preprocesado consiste en eliminar las partes, *tokens*, signos de puntuación y todo aquello que obstruya la obtención de unos resultados sin inferencias de cada uno de los documentos (en este trabajo «documento» y «tuit» son sinónimos cuando nos referimos a los elementos que alberga el corpus) que conforman un corpus: «Cleaning is an important step before any text mining task, in this step, we will remove the punctuations, stopwords and normalize the corpus» (Shivam Bansal 3). De esta manera nos deshacemos del conocido como «ruido» (Deerwester et al. 391). En el caso de los tuits, su contenido se compone de diversos elementos anteriormente enumerados que dificultan la aplicación de un modelado de tópicos (Mottaghinia et al. 3). Así, resulta necesaria la supresión de menciones a cuentas, identificadas por @ en su comienzo; asimismo, se eliminan signos de puntuación como puntos, comillas o signos de interrogación; también números, símbolos, emoticonos y enlaces; además se transforma todos los *tokens* a minúscula; es necesario eliminar los saltos de línea, identificados normalmente con `\n`; por último, se eliminan las conocidas como *stopwords*, esa serie de palabras fijas pero ampliables que carecen de significado. El paso previo a esta limpieza es la *tokenización* (Alash y Al-Sultany 4). A grandes rasgos, la *tokenización* es un proceso consistente en la división de una cadena en palabras, almacenando cada uno de los *tokens* que la componen como elementos de una lista manejable, es «the process of splitting the text into "tokens," which can be individual words, sequences of words, or individual sentences» (Jacobucci et al. 56).

El resultado final del preprocesado es la reducción de un tuit a no más, en la mayoría de los casos, de tres o cuatro palabras relevantes para *topic modeling*. En ocasiones, la lista que recoge los *tokens* relevantes para cada tuit queda vacía, por lo que asumimos que ese tuit no tenía información relevante para nuestro análisis. El último paso del preprocesado es el *stemming*. Esta etapa consiste en reducir a la raíz las palabras restantes dentro de las listas. Esto implica unificar las formas que se diferencien en género y número bajo un mismo *token*. Aunque Jacobucci et al. indican que es un paso imprescindible para evitar la presencia de palabras erróneamente escritas en los resultados finales, nosotros hemos preferido mantenerlas, pues creemos que un fallo gramatical es ocasional, y que por lo tanto el *token* mal escrito correspondiente tendrá un peso ínfimo. Esto le impedirá formar parte de los resultados finales. Este paso es pues prescindible, ya que este aglutinamiento bajo el término raíz se puede perder información relevante a la hora de elaborar resultados (56). En el caso del preprocesado de tuits con temática lingüística, imaginemos uno que consulta sobre el punto, mientras que otro pregunta por los puntos suspensivos. Tras la *stemización*, ambos se aglutinarán bajo `punto`, perdiendo su rasgo característico y eliminando la diferencia de las consultas.

Pasos: Diccionario

Con este resultado, procedemos a la preparación de estos *tokens* para la aplicación del modelo. El modelo necesita de la creación de un diccionario. Este diccionario será igualmente clave al comprobar cómo un tuit que no formaba parte del corpus original encaja en los temas resultantes. El diccionario no es una lista alfabéticamente ordenada, sino una en la que a cada *token* se le asigna un valor numérico único y entero, un identificador. Si un *token* aparece en una lista posterior de la que apareció por primera vez, este tomará el valor entero que le fue asignado la primera vez. De esta manera cada tuit es transformado a un «appropriate format (features vector) for topic models. one of these features is a bag of words matrix (Bow) where each document (tweet) is represented by the number of times a word found in a tweet» (Alash y Al-Sultany 5).

Una vez se ha realizado esta correlación entre *token* y valor entero, así como se ha procedido con la creación del diccionario, es posible aplicar las técnicas de *topic modeling*. Teniendo en cuenta los dos modelos que nosotros implementaremos, los resultados pueden ser de dos tipos: LSA refleja un valor de dispersión. Esto quiere decir que una palabra tiene mayor peso dentro del tema cuanto más se aleja del eje, representado por `0`. Lo máximo, aunque nunca se alejará tal distancia, es `1` o `-1`. Esto implica que no es relevante si el valor es negativo o positivo, porque lo importante es la distancia con `0`. La suma de los valores no da `1`. En LDA es diferente, es probabilístico. Esto quiere decir que la suma de los valores asignados a las palabras pertenecientes a un mismo tema dará `1`, es decir, el cien por cien. LDA obtiene solo valores positivos, con un máximo establecido en el valor `1`.

Pasos: *Coherence score*

Ya en posesión de los resultados, podemos aplicar el marcador de coherencia o *coherence score*. Esta marca el grado de similitud entre dos palabras del tema. Es un valor que representa cómo de adecuado es que un determinado término o *token* pertenezca al tema asignado (Alash y Al-Sultany 6).

Estos son los tres pasos comunes que sigue la técnica *topic modeling*. Es decir, ambos modelos comparten dos pasos previos a la aplicación del modelo en sí: preprocesado y creación del diccionario. Asimismo, comparten un paso posterior: calcular la coherencia. Con esto se obtienen bien resultados similares, bien resultados diferentes.

3.2. Latent Semantic Allocation

El primer método de *topic modeling* que vamos a desglosar es Latent Semantic Allocation, una técnica anterior a LDA y de la que fue su precursora. Varias son las similitudes con su sucesor, pero ello no le exime de contar con características propias. Adelantamos que su actuaciones se basan en la descomposición de valores singulares, una técnica vectorial que estructura documentos mediante valores numéricos. El prestar atención a la manera en la que se distribuyen las palabras y no a su significado es otra de su características que desarrollaremos a continuación.

3.2.1. Definición y conceptos clave

> WHAT IS LSA? LSA is a fully automatic mathematical and statistical technique for extracting and inferring relations of expected contextual usage of words in passages of discourse. It is not a traditional natural language processing or artificial intelligence program; it uses no humanly constructed dictionaries, knowledge bases, semantic networks, grammars, syntactic parsers, morphologies, or the like, and it takes as its input only raw text parsed into words defined as unique character strings and separated into meaningful passages or samples such as sentences or paragraphs. (Landauer et al. 263)

El Latent Semantic Analysis o Análisis Semántico Latente, al que nos hemos y seguiremos refiriendo por sus siglas en inglés «LSA», es una de las técnicas más longevas empleadas para identificar las relaciones entre palabras y documentos dentro de un corpus. Es un método de indexación y recuperación de datos, además de «for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text» con el que obtener temas subyacentes (Landauer et al. 259-260). LSA basa su análisis en el principio de que las palabras que aparecen en contextos similares tienen significados parecidos, focalizándose de esta manera en el contenido relacionado. Asimismo, también se percata de esas palabras gráficamente parecidas, como por ejemplo «punto» y «puntos», que de acuerdo con Qiang et al. tienen una mayor probabilidad de referirse al mismo concepto (2). Según Deerwester, LSA es un modelo que trata de juntar ítems similares. Este tipo de modelos se componen de jerarquía, superposiciones y *clusterings* o agrupamientos (393). LSA captura información indirecta contenida en las innumerables limitaciones y las relaciones estructurales presentes en los documentos (Landauer et al. 262): «We assume there is some underlying latent semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval» (Deerwester et al. 391). Partiendo de estas características, LSA puede actuar de manera coherente a la hora de lidiar con sinónimos que describen o se refieren a la misma idea. LSA se caracteriza y se diferencia de otros métodos de *topic modeling* porque es capaz de extraer «contenido conceptual de un cuerpo de texto mediante el establecimiento de asociaciones entre los términos que aparecen en contextos similares» (Chandía Sepúlveda 18).

3.2.2. LSA vs. humanos

Algunos autores como Landaure et al. dictaminan que LSA es capaz de simular una variedad de fenómenos cognitivos humanos, como el desarrollo del proceso que van desde la adquisición de vocabulario a la categorización de palabras, o bien la identificación de temas por palabras más importante dentro de una cadena. Para ellos, LSA es el método más cercano a los procesos cognitivos propios del ser humano, con los que extrae información: «close resemblance between what LSA extracts and the way people's representations of meaning reflect what they have read and heard, as well as the way human representation of meaning is reflected in the word choice of writers» (Landauer et al. 260). Para estos autores, el poder de LSA sobrepasa la clasificación que un humano podría llevar a cabo. LSA está estrechamente relacionado con los modelos neuronales, con la diferencia de que el método informático emplea una matriz de descomposición. Sin embargo, los temas obtenidos se asemejan a la idea global que pueda tener el lector de un determinado número elevado de textos. En sus palabras, la similitud dada por LSA va más allá de frecuencias

de contigüidad o coocurrencias, ya que el algoritmo obtiene mediante un poderoso método matemático relaciones más profundas entre términos. Es este hecho el que les hace asegurar que el modelo sobrepasa el análisis humano (Landauer et al. 260–1).

3.2.3. Funcionamiento del algoritmo

El código de LSA tiene varias partes clave para poder funcionar de manera correcta. Su principal característica es la SVD, una técnica matemática descrita a continuación. Asimismo, cuenta con otros pasos que se parecen en mayor o menor medida a los que veremos con posterioridad en LDA.

Descomposición en valor singular

LSA toma como datos iniciales no solo «the summed contiguous pairwise (or tuple-wise) co-occurrences of words but the detailed patterns of occurrences of very many words over very large numbers of local meaning-bearing contexts, such as sentences or paragraphs» (Landauer et al. 261). A partir de este punto, implementa su rasgo más característico: la SVD.

La técnica matemática llamada Descomposición de Valor Singular (SVD por sus siglas en inglés) identifica patrones en las relaciones entre los *tokens* o palabras presentes en documentos que componen un corpus, así como entre los propios documentos a partir de tales términos. Su implementación permite hallar las relaciones entre palabras empleadas en el lenguaje natural, poniendo de manifiesto conexiones y relaciones entre distintos documentos y textos. La matriz no presta atención al orden de las palabras, sino a la estructura sintáctica y relaciones lógicas o morfológicas. La SVD refleja los patrones principales asociativos en los datos, ignorando las conexiones menores, las que tienen menor peso (Deerwester et al. 391). Según Landauer et al., «relationships inferred by LSA are also not logically defined, and they are not assumed to be consciously rationalizable» (269). La técnica SVD omite cómo las palabras producen el significado y presta mayor atención a la presencia de palabras en un mismo punto de múltiples documentos. En consecuencia, es capaz de relacionar la información similar que los creadores de los documentos compusieron de manera dispar (Deerwester et al. 392). Resulta importante remarcar que LSA prefiere palabras a grupos de palabras, por lo que el valor de la *tokenización* incrementa (Landauer et al. 260-1).

Esta matriz SVD es la implementación clave de la innovación de LSA. Funciona descomponiendo la matriz original de términos-contextos. Esto es que, para cada documento del corpus (por ejemplo, un tuit) existe en la descomposición una columna, y para cada *token* de ese documento, una fila (Chandía Sepúlveda 19):

> Each row stands for a unique word and each column stands for a text passage or other context. Each cell contains the frequency with which the word of its row appears in the passage denoted by its column. Next, the cell entries are subjected to a preliminary transformation... in which each cell frequency is weighted by a function that expresses both the word's importance in the particular passage and the degree to which the word type carries information. (Landauer et al. 263)

En el siguiente fragmento se nos ofrece una explicación más detallada del proceso:

> In SVD, a rectangular matrix is decomposed into the product of three other matrices. One component matrix describes the original row entities as vectors of derived orthogonal factor values, another describes the original column entities in the same way, and the third is a diagonal matrix containing scaling values such that when the three components are matrix multiplied, the original matrix is reconstructed. (Landauer et al. 263)

A través de la SVD se obtienen los temas y los *tokens* que los caracterizan. A estos temas, también llamados clases latentes, tópicos o categorías que componen el documento, se les asignan distribuciones de probabilidad que asocian las variables latentes con las palabras y documentos.

Pasos del modelo LSA

Para la división en temas anónimos, LSA sigue distintos pasos: el primero es tomar un corpus preprocesado. A pesar de que LSA presta alta atención a la relación entre palabras, el preprocesado omite este hecho y suprime una gran número de *tokens*, lo que resulta en una mejora considerable de los resultados, ya que estas palabras carecen de importancia a la hora de proceder con la implementación del modelo. Las que se mantiene y sí que serán relevantes se almacenan en listas. Luego, asigna a cada elemento de las respectivas listas (cada uno será un *token* relevante para el estudio) una probabilidad de aparición (estas palabras han sido previamente almacenadas en un diccionario); en segundo lugar y en función de esta asignación de probabilidad, se le asigna un tema al documento; por último, se extrae la palabra o palabras que marcan cuál es el tema del documento (Chandía Sepúlveda 19–20). LSA no mide únicamente el peso de las palabras en sí mismas, sino que lo obtiene a partir de las relaciones palabra-documento y documento-documento basado en la similitud semántica. La idea subyacente es que a una palabra le viene dado su grado de similitud con otras a través de cuantos contextos comparten ambas. Para LSA, el significado de una palabra viene determinado por el significado de todos los documentos en los que aparece. Y también que el significado de un documento se obtiene del significado medio de las palabras que lo componen (Landauer et al. 259-61). En los resultados, es más probable ver cómo las palabras que aparecen con mayor frecuencia forman parte del núcleo de los distintos temas. Los términos con pocas apariciones son puntos débiles del modelo, por lo que un buen preprocesado resulta ineludible. El límite de cuántas veces es necesaria la aparición de un término para que sea considerado relevante es una decisión del analista o investigador (Shivam Bansal 7).

Tras el procesado de un corpus original y de haber hallado un modelo, LSA puede tomar un nuevo corpus o documento y posteriormente indicar en cuál de los temas anteriormente generados clasificar los nuevos documentos a partir del diccionario ya creado y del peso de las palabras dentro de los nuevos documentos (Landauer et al. 260).

3.2.4. LSA y Twitter: retos y oportunidades

Traspolado a la red social Twitter, Alash y Al-Sultany indican que teniendo como objetivo la obtención de unos resultados adecuados, es preferible seleccionar un corpus que comparta una característica visible común, como por ejemplo la presencia de una etiqueta o *hashtag*. Estas etiquetas ayudan a la detección de temas. Del mismo modo, ambos autores hacen hincapié en que el preprocesado de estos documentos debe ser exhausti-

vo, pues contienen demasiado ruido: menciones, etiquetas, signos de puntuación y otros rasgos presentes en estos *short texts* dificultan una elaboración precisa del modelo:

> After the preprocessing stage, each text (tweet) was converted into an appropriate format (features vector) for topic models. one of these features is a bag of words matrix (Bow) where each document (tweet) is represented by the number of times a word found in a tweet. (Alash y Al-Sultany 5)

Por último, hemos de indicar que no todo son bondades en la aplicación de este método. Entre las características lingüísticas que perjudican la técnica LSA encontramos la polisemia o la no distinción entre dos palabras referidas al mismo tema. Una palabra homógrafa que designa dos realidades distintas es indetectable para el modelo. Es decir, en el caso de "como", que tiene valor de adverbio, adverbio relativo, conjunción y preposición, además de ser la forma conjugada del verbo "comer", implica que a todas ellas LSA le asigne el mismo valor entero, por lo que el peso de este *token* vendrá dado por las distintas funciones que cumplan los "como" dentro de los documentos, provocando una clasificación altamente imprecisa de palabras polisémicas (Deerwester et al. 392). Por el contrario, dos palabras similares halladas en el mismo tuit, por ejemplo, «aún» y «aun», son guardadas como *tokens* diferentes, hecho que va en detrimento de los resultados.

3.3. Latent Dirichlet Allocation

Abordados *topic modeling* y la explicación teórica del algoritmo LSA, procedemos al desarrollo conceptual del algoritmo LDA. En esta sección, en la misma línea que LSA, se expondrán sus rasgos principales y definitorios, su composición y trabajo, sus aplicaciones y los retos y oportunidades que encuentra el modelo al trabajar con tuits. Todo ello con el fin de ofrecer un acercamiento suficiente para la futura comprensión del comportamiento del modelo en su ejecución.

3.3.1. Definición y conceptos clave

Latent Dirichlet Allocation (LDA) (Blei, Ng y Jordan) es un modelo de tópicos generativo cuya asignación de temas se basa en la frecuencia de palabras existente en un conjunto de documentos, resultando particularmente útil para encontrar una composición coherente y precisa de los temas existentes en un documento. Es decir, a partir de un conjunto de textos, LDA, como algoritmo implicado en el modelado de tópicos, encuentra de forma automática en qué temas se pueden dividir dichos textos, asigna cada texto a uno de esos temas o, al menos, da la probabilidad de que un término pertenezca a un tema y, finalmente, sugiere a qué tema de los que se han decidido asignaría un nuevo texto. Previo a su surgimiento, al modelo LDA le preceden dos modelos relacionados: Latent Semantic Analysis (LSA) y Probabilistic Latent Semantic Analysis (PLSA). Tras sus antecesores, en 2003, David M. Blei, Andrew Y. Ng y Michael I. Jordan presentaron el modelo definiendo sus principales características como modelo probabilístico generativo pensado para cantidades discretas de datos cuyo funcionamiento se establece en un modelo bayesiano compuesto por una jerarquía de tres niveles (1). El modelo se contextualiza en el campo del aprendizaje automático (*machine learning*): disciplina de las ciencias de la computación basada en el análisis de datos por medio de algoritmos para distintos fines computacionales. Y, más concretamente, en la técnica de aprendizaje automático de

modelado de tópicos. La combinación de ambos conceptos y funciones se esclarecen de la siguiente manera:

> Los modelos de tópicos son un conjunto de algoritmos que proporcionan una solución estadística al problema de la gestión de grandes archivos de documentos. Con los recientes avances científicos en apoyo de los componentes para el modelado de aprendizaje automático no supervisado para el modelado, algoritmos escalables para la inferencia posterior y con el mayor acceso a modelos de conjuntos de datos masivos (dataset), los modelos de tópicos prometen ser un componente importante para la síntesis y la comprensión de nuestros crecientes archivos digitalizados de la información. (Chandía Sepúlveda 12)

De modo que LDA, como técnica de modelado de tópicos de aprendizaje automático generativo se compone «por conceptos de los Modelos Bayesianos y se basa en el proceso probabilístico genérico que permite inferir tópicos de un documento en base a una distribución a posteriori obtenida por el LDA» (Chandía Sepúlveda 14).

Sin embargo, su funcionamiento establecido de acuerdo con determinados conceptos del Teorema de Bayes determina un proceso que parte de una distribución *a priori*. En el contexto del modelo LDA, esta distribución *a priori* se conoce como intuición y está basada en el hecho de que los documentos pueden exhibir múltiples temas (Blei 78). Por ejemplo, si el conjunto de documentos está formado por una colección de diferentes textos literarios y hay indicios de que varios temas surgen con frecuencia dentro del conjunto de documentos, el objetivo es determinar cuáles son esos temas frecuentes. Para ello, se llevan a cabo ciertas suposiciones clave: las palabras que aparecen juntas frecuentemente es posible que tengan un significado cercano, cada tema es una mezcla de diferentes palabras (ejemplo desarrollado para esta investigación en la figura 3.1) y cada documento es una mezcla de diferentes temas (ejemplo desarrollado para esta investigación en la figura 3.2). LDA trata de determinar cuál es el mecanismo que genera los documentos y los temas basándose en esa intuición. Así, crea un conjunto de máquinas con diferentes configuraciones y selecciona la que da los mejores resultados. Una vez que se encuentra el mejor, observamos su configuración y deducimos los temas (Serrano).

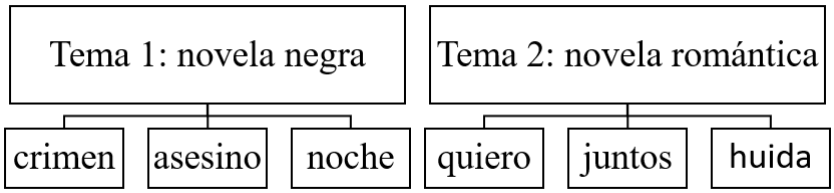

Figura 3.1: Ejemplo propio: Cada tema es una mezcla de diferentes palabras

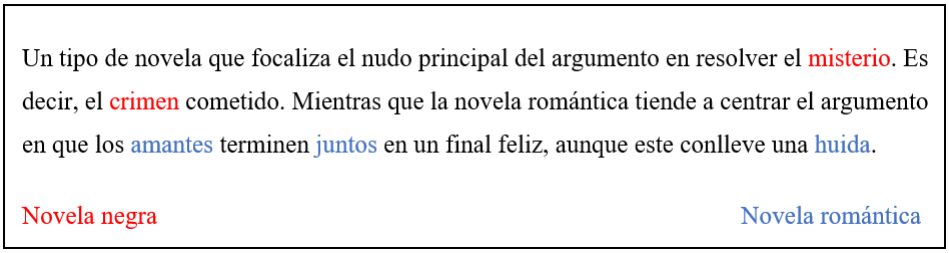

Figura 3.2: Ejemplo propio: Cada documento es una mezcla de diferentes temas

Las configuraciones parten del Dirichlet antes de la asignación de temas: el conjunto de distribuciones de probabilidad según las cuales se establece cómo se mezclan los documentos hasta conseguir asignar los temas. Como los temas son distribuciones, el modelo genera un documento compuesto por una serie de palabras que, dependiendo de la distribución definida para el algoritmo en sus parámetros e hiperparámetros, es decir, en los valores previos que componen el algoritmo antes de su entrenamiento (parámetros) y durante este (hiperparámetros), ofrece dos posibles distribuciones: unos porcentajes concretos de cada tema en un documento o de cada palabra en un tema. Por ejemplo, dentro del tema de novela negra la palabra «crimen» puede tener un porcentaje de `0.9`, la palabra «asesino» de `0.05` y la palabra «noche» de `0.03`. No obstante, el algoritmo no proporciona el título del tema «novela negra» o «novela romántica», sino solo las palabras que lo componen. A menudo, esas palabras vienen acompañadas de datos sobre la probabilidad de pertenencia a un tema. Y, es a partir de las informaciones ofrecidas, cómo se lleva a cabo la tarea del usuario de identificar ese título o concepto basándose en la relación de las palabras, pues un tema tenderá a ser una combinación de temas diferentes a ojos del usuario, por ejemplo «novela negra o de misterio».

El algoritmo relee cada documento hasta obtener la distribución más coherente para el conjunto de documentos dado que dará como resultado una lista de palabras con sus respectivos porcentajes de peso en cada documento, los cuales determinarán el tema. Es decir, de un conjunto de documentos trabajado por el algoritmo se consiguen varias listas de palabras en las cuales aquellas de mayor peso serán los temas. Con arreglo a la función de este algoritmo, Blei propuso su técnica de modelado de tópicos Latent Dirichlet Allocation o LDA. La propuesta nació con el propósito «de intentar recrear de manera inversa el modelo teórico por el que los textos se generan» (Calvo). Para ello, se basaba en la premisa de que los autores disponen de un conjunto delimitado de temas que, a su vez, contienen una serie de palabras con una distinta relevancia que el autor selecciona para componer su texto. De esta premisa nace la clave del modelo: la intuición, los supuestos.

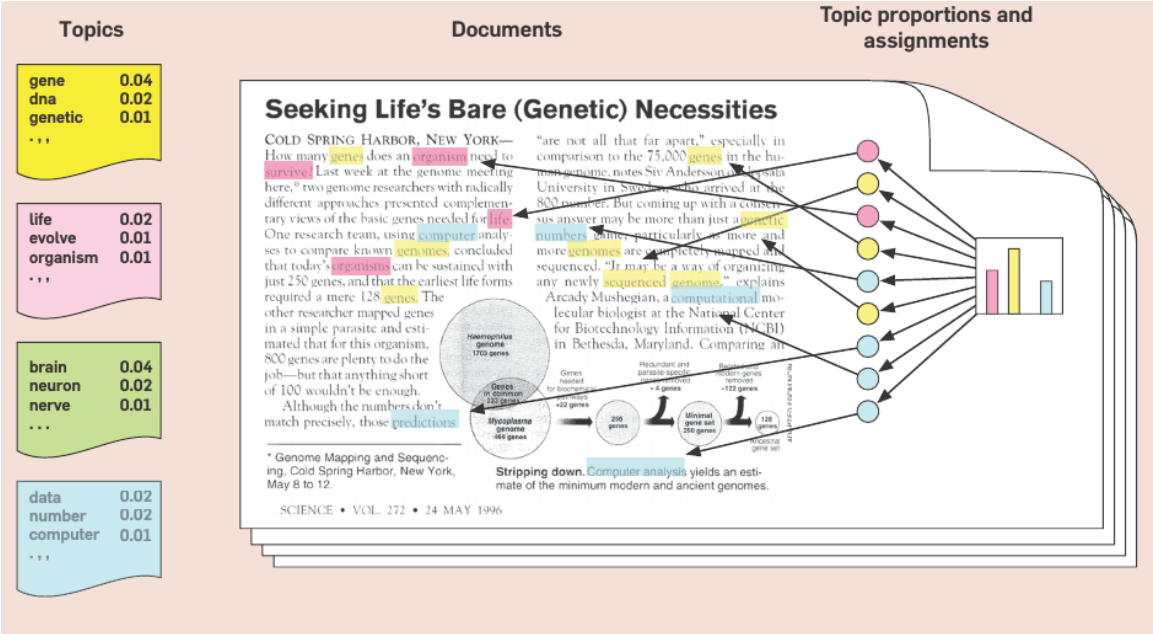

Figura 3.3: Palabras y porcentaje de notabilidad
Fuente: Blei 78

Así, en la figura 3.3 se muestra las listas de palabras con fondos de diferentes colores y, cada palabra, aparece con su porcentaje de notabilidad en el documento en cuestión. Estas palabras son los denominados temas o *topics* «a distribution over a fixed vocabulary» (Blei 78). Es decir, la distribución de temas conseguida al entrenar el algoritmo en un conjunto de documentos determinado. Técnicamente, el modelo asume que esos tópicos son generados en primer lugar, incluso antes que los propios documentos. Definiendo así el objetivo principal de LDA como técnica para el modelado de tópicos que extrae los temas que estructuran un determinado conjunto de documentos siempre asumiendo que esos tópicos existen previamente, aunque no estén precisados.

Por ejemplo, Blei toma diecisiete mil artículos de una revista de ciencia para entrenar el modelo. El algoritmo asume que hay 100 tópicos y, de acuerdo con esa cantidad asumida, configura la distribución de tópicos según el conjunto de palabras obtenido de los documentos. Con este resultado, se examinan los tópicos más probables o relevantes (79).

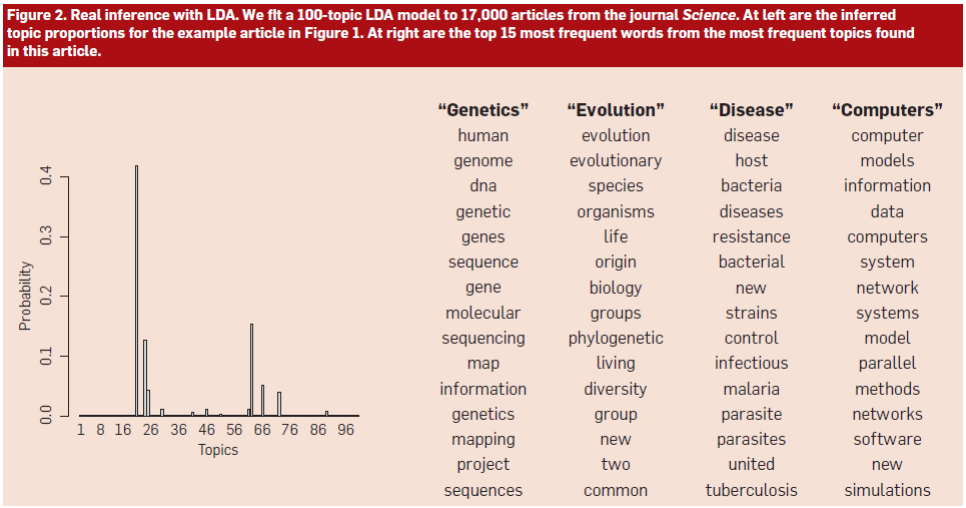

Figura 3.4: Distribución de términos y tópicos
Fuente: Blei 79

Observando las distintas distribuciones ofrecidas por el algoritmo en la figura 3.4 (Blei 79), la determinación de los tópicos ha de establecerse según algún criterio. En este caso, el criterio de interpretación para el modelo LDA y sus temas resultantes es la relevancia, propuesto y entrenado por Shirley y Slevert para solventar el problema que surgía en la implementación del modelo en cuestión:

> To interpret a topic, one typically examines a ranked list of the most probable terms in that topic, using anywhere from three to thirty terms in the list. The problem with interpreting topics this way is that common terms in the corpus often appear near the top of such lists for multiple topics, making it hard to differentiate the meanings of these topics. (Shirley y Slevert 64)

La relevancia la definen entonces como un método para determinar la frecuencia de un término dentro de un tema en concreto y, a su vez, la del término en todo el conjunto de documentos. La relevancia permite determinar si el término es frecuente porque pertenece a un tema específico o porque aparece repetidamente en la muestra, según qué datos se ingresen en sus valores.

De acuerdo con todo el diseño específico del algoritmo, como técnica de modelado o detección de tópicos, puede aplicarse prácticamente para cualquier tipo de colección que se quiera analizar. Aunque los resultados sean más beneficiosos para cierto tipo de documentos, la estructura estadística que subyace del lenguaje observado que infiere los temas a partir de la interacción del conjunto de documentos observado y los supuestos probabilísticos de LDA hace favorable el resultado de la aplicación de la técnica desarrollada por Blei.

3.3.2. Composición y trabajo del algoritmo

La teoría del desarrollo de LDA establece que el modelo infiere los temas latentes del conjunto de documentos con base en un marco bayesiano según las palabras que componen dicho conjunto a partir de probabilidades condicionales. Todo ello, como se ha mencionado previamente, parte de un modelo probabilístico generativo y de distribuciones de

Dirichlet. Para llevarse a cabo, hay que tener en cuenta qué hiperparámetros debemos definir antes de implementar y visualizar LDA. El algoritmo cuenta con una serie de elementos para su correcto funcionamiento. En primer lugar, ha de decidirse el número de temas, k. Ese elemento es una de las suposiciones clave del modelo pues establece que el conjunto de documentos a analizar se puede describir mediante un número de temas concreto antes de procesarlo. El conjunto de documentos sobre el que trabaja el algoritmo recibe el nombre de «corpus» y el resultado del análisis de dicho corpus es un conjunto de palabras que recibe el nombre de «vocabulario». Además del número de temas, LDA permite establecer previamente otros hiperparámetros que también forman parte de sus suposiciones: `alpha` y `beta`, vinculadas a las distribuciones de Dirichlet. Si no se especifican, el propio algoritmo asume valores predeterminados que pueden resultar igualmente útiles para el análisis que se quiera realizar. No obstante, si conocemos bien el corpus sobre el que trabajará LDA y los resultados que se quieren conseguir, la definición de esas suposiciones favorecerá a la interpretación de los temas obtenidos. Los valores de estos dos hiperparámetros influyen, por tanto, en las distribuciones generadas. Cuanto más alto sea este valor, más centrada será la distribución, y viceversa.

Una vez concretados los hiperparámetros principales y su teoría la cuestión es conocer cómo consigue el algoritmo que la técnica LDA funcione. El algoritmo para entrenar se basa en un proceso que se repite realizando las asignaciones de temas para cada palabra en cada documento del corpus. Así, detecta los temas latentes del corpus y genera la combinación de temas o la distribución de los temas en el documento, las dos distribuciones usadas por el Dirichlet de LDA. Obtiene entonces las diferentes mezclas de acuerdo con los k temas concretados. De modo que, una vez determinado ese valor k, el algoritmo LDA comienza sus pasos cuya explicación podemos dividir en cuatro partes.

En primer lugar, el modelo se inicializa asignando aleatoriamente un tema a cada palabra en cada documento. Después de esa asignación, ofrece las dos posibilidades de frecuencias según las distribuciones pudiendo calcular la frecuencia de palabra o la frecuencia de tema desde distintas interpretaciones: la notabilidad de cada tema en el corpus, la notabilidad de cada palabra en cada tema o probabilidades como la plausibilidad de que una palabra se asigne con un determinado tema.

En segundo lugar, el algoritmo detecta las funciones de las dos distribuciones de Dirichlet, es decir, la generación de probabilidades condicionales para cada uno de los temas, independientemente de si aparecen en todos los documentos del corpus o no, pudiendo así ser considerados en ambas distribuciones: temas por documento y palabras por tema. Ambas probabilidades son codependientes para la generación de la asignación de temas que determinarán los valores más importantes para dicha asignación: los recuentos de frecuencia y las distribuciones de Dirichlet. Estas asignaciones y distribuciones las basará, por tanto, en la frecuencia notable para las distribuciones no en la morfología semántica de las palabras. Así, las asignaciones iniciales pueden no ser óptimas, pero, la interpretación guiada por la relevancia, en combinación con el proceso iterativo del modelo, logrará un descubrimiento de temas y relaciones latentes.

En tercer lugar, aunque fuera del orden de pasos expresado, como técnica que trabaja con lenguaje natural, incluye en su funcionamiento una fase previa de procesamiento del corpus que va a trabajar el algoritmo. Algunas de las partes de este preprocesado son, principalmente, la *tokenización*, la *lematización* y la eliminación de *stopwords*. Todas ellas explicadas en el apartado correspondiente del capítulo 4.

Por último, una vez que el algoritmo termina su ejecución, el modelo está implementado y pueden interpretarse diversas formas de visualización de los resultados obtenidos siendo `pyLDAvis` (Sievert y Shirley) la herramienta de visualización para modelos LDA por excelencia por estar expresamente pensada para las necesidades de este modelo en cuestión. Un paquete diseñado por Carson Sievert y Kenny Shirley para la interpretación de los temas identificados por LDA. La herramienta desarrollada extrae la información obtenida por el algoritmo y la representa en un modo de visualización interactiva basada en el formato web en forma de gráfico cuya muestra se basa en el término de la relevancia, previamente referido en este trabajo.

Lo más favorable de la visualización presentada por Sievert y Shirley lo concretaban los propios autores en la presentación del paquete `pyLDAvis`: «Our visualization provides a global view of the topics (and how they differ from each other), while at the same time allowing for a deep inspection of the terms most highly associated with each individual topic» (63).

Este paquete ayuda a la interpretación del tema teniendo en cuenta las relaciones de relevancia mostrando sus probabilidades y distribuciones, siendo así el más adecuado para el modelo LDA por ir más allá de las relaciones entre los temas y las palabras del conjunto de documentos expresados en gráficos de barras, nubes de palabras o diagramas. Su visualización pretendía ser más compacta y específica, asegurando añadir una ventaja clara sobre *Termite*, el modelo que consideran su antecesor y que visualiza las distribuciones entre palabra y tema a partir de un diseño de matriz que se interpreta según los términos de distinción y prominencia (Sievert y Shirley 65-6). Estas medidas determinan cuánta información transmite un término sobre los temas a partir de determinados cálculos de distribución y frecuencia que decide qué temas se incluyen en la visualización y en qué orden aparecen para hacer notables las diferencias entre ellos.

La propuesta antecesora también era interactiva, compacta e intuitiva, pero solo permite visualizar los temas que tienen una alta notabilidad, mientras que Sievert y Shirley pretendían que el usuario pudiera profundizar en cada uno de los temas: analizándolos de manera independiente o en conjunto, atendiendo al carácter exploratorio de los resultados que marca el modelo LDA desde su identificación y asignación latente (véase figura 3.5).

De modo que `pyLDAvis` permite seleccionar un tema para mostrar sus términos más relevantes a la vez que tiene la capacidad de seleccionar un término para revelar la distribución condicional que tiene con respecto a los temas. Todo a través del cursor manejado por el usuario (Sievert y Shirley 66-8).

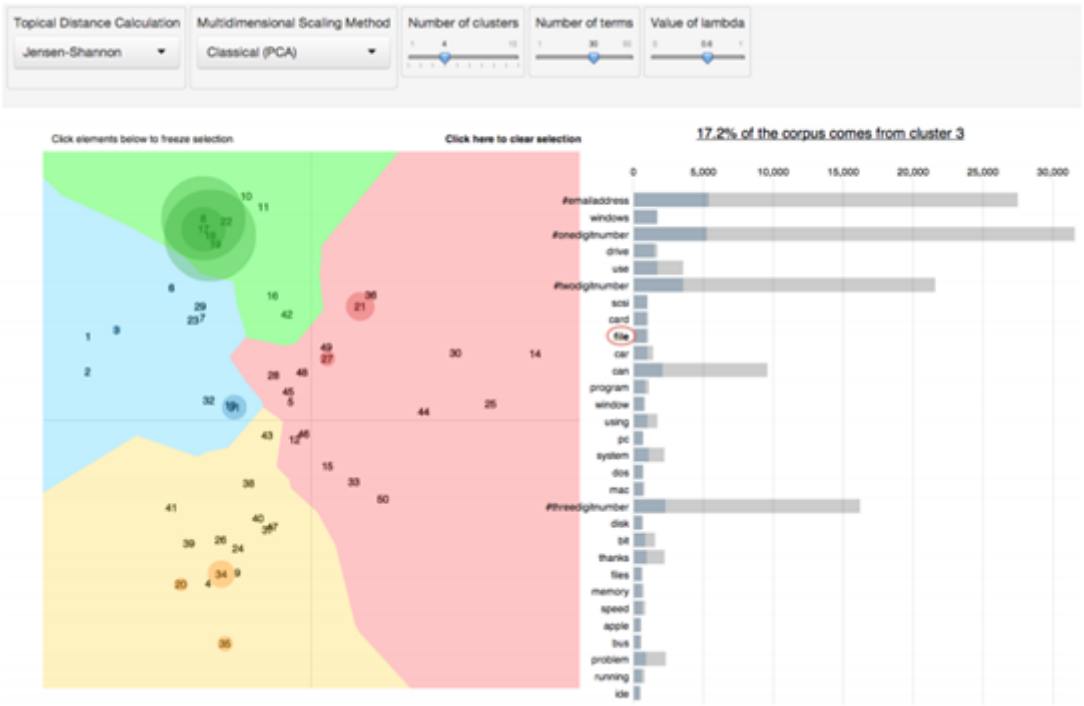

Figura 3.5: Visualización de temas `pyLDAvis`
Fuente: Sievert y Shirley 69

Considerados preprocesado, parámetros e hiperparámetros, asignación e identificación latente, distribución, visualización e interpretación, conocemos la composición y cómo se fundamenta el trabajo del algoritmo del modelo LDA para que su implementación ofrezca al usuario los resultados más productivos posibles en las diferentes aplicaciones de su entrenamiento.

3.3.3. Aplicaciones del modelo LDA

El enfoque que alberga el modelo LDA es popular dentro del modelado y la detección de tópicos, pues su asignación de temas latentes en conjuntos de documentos de texto, de acuerdo con un proceso probabilístico generativo e iterativo, suele resultar conveniente y relevante para análisis concretos sobre los temas más frecuentes que tienen lugar en documentos ricos para las ciencias informativas y de la comunicación. Su uso extendido suele deberse a su procedimiento latente, su implementación considerada sencilla y la calidad de sus resultados dejando a un lado la necesidad de contar con datos etiquetados como ocurre en el trabajo realizado en el contexto del aprendizaje supervisado («Modelado de temas con LDA: una explicación intuitiva»). La utilidad de los modelos de tópicos radica en que la estructura latente consigue asemejarse con la estructura temática del conjunto de documentos analizado. Esa estructura oculta por interpretar permite obtener clasificaciones del texto que compone el corpus con múltiples aplicaciones que van desde la recuperación, la búsqueda y la exploración de ese conjunto de información hasta el estudio del algoritmo y su comportamiento con distintos conjuntos de archivos.

Una vez que LDA determina la estructura latente de los tópicos y su posible distribución, inicializa el modelo y lleva a cabo el proceso iterativamente hasta concluir unas probabilidades *a posteriori* sacadas del proceso generativo que parte de la intuición capturada

por LDA en sus supuestos previos. Dichas probabilidades ofrecen diferentes distribuciones aplicables a investigaciones derivadas de la minería de datos y del tratamiento de la información. En general LDA, como herramienta dentro del modelado de tópicos, se utiliza para el tratamiento de la información digital o digitalizada y, más concretamente, como algoritmo con la capacidad de tratar la organización, la estructura, la disposición o la generación de la información que compone la web en la era de la comunicación masiva. Con LDA, el modelado de tópicos puede principalmente descubrir los temas fundamentales de una colección de documentos, clasificar ciertos documentos de acuerdo con los temas que aparezcan en cada uno de ellos, concluir métodos de generación de textos a partir de la frecuencia y distribución del vocabulario que los compone o determinar la lógica para las palabras clave de búsqueda de información. Su aplicación puede descubrir los tópicos del corpus conformado. Por ejemplo, determinar cuáles son los temas más relevantes en un conjunto de documentos formado por las noticias publicadas en un determinado territorio durante un año y sus palabras relacionadas. Además, con la posibilidad de actualización del corpus, puede concluir, por ejemplo, qué temas han sido más relevantes en las cartas al director de un periódico concreto durante años sucesivos.

No obstante, aunque sus aplicaciones son variadas y productivas, la extensión del corpus o el lenguaje utilizado en los textos que lo componen puede provocar que los resultados de la aplicación del modelo sean más difíciles de interpretar o, incluso, no concluir con resultados satisfactorios. Por ello, resulta interesante la posible aplicación del modelo elegido en esta investigación: el modelo LDA para la extracción de los temas más frecuentes en un conjunto de tuits. Esta aplicación supone una serie de retos concretos para el funcionamiento y los resultados del modelo de tópicos que podrían asemejarse con los que se encontrarían al entrenar LDA para extraer los temas de conversación más frecuentes en un chat, en los comentarios de un blog o de una publicación de Instagram.

3.3.4. LDA y Twitter: retos y oportunidades

El uso del algoritmo en cuestión ha sido entrenado en diversos trabajos, como se presentará en la sección «Investigaciones afines». En ellas, las conclusiones generales relatan problemáticas y conveniencias similares.

Como retos, destacan los ya presentados: registro informal en el lenguaje que conforma los textos del corpus, documentos pequeños en su forma original, entorno para la recolección de documentos con rasgos muy característicos como la disposición de las publicaciones, las etiquetas, el frecuente uso de enlaces o referencias a imágenes y menciones, nombres de usuario e informaciones extras como la fecha y la hora o el alcance del tuit, en otras especificaciones del entorno. Estos factores que pueden resultar un obstáculo en la investigación en cuestión también pueden llegar a ser oportunidades para explorar la aplicación de LDA si se consiguen superar los desafíos que presenta Twitter como entorno analítico.

Se ha de crear un documento con tuits que permita un trato de la información correcto tanto para el modelo como para la interpretación última de los datos concluidos. Una vez conseguido el documento, el preprocesado que precisa LDA se vuelve aún más clave para el análisis de datos recopilados: se necesita pulir el documento hasta conseguir un conjunto de textos con extensión, forma y recuperación de información idónea para el estudio. Solventado esto, la investigación del comportamiento del modelo de Blei en combinación con la multiplicidad y riqueza de datos disponibles en esta red social supone toda una

fuente de investigaciones fructíferas para los estudios de la información que tiene lugar en la web.

Twitter se constituye como un entorno repleto información publicada por usuarios. Por su parte, LDA brinda resultados a través de sus procesos de extracción, análisis y visualización de datos, ofreciendo posibilidades de investigación con trayectorias, al menos, prometedoras para el avance de las técnicas implicadas. No obstante, aún queda por resolver ciertas problemáticas que se escapan en el procesamiento del lenguaje natural como la polisemia y la acentuación, entre otros. Problemáticas que afectan al procesamiento que realiza LDA y, como se menciona previamente, también a LSA.

3.4. Comparativa teórica de los modelos

Siendo los dos modelos más usados del modelado de tópicos, LSA y LDA presentan diferencias sustanciales que serán, en parte, desarrolladas en este apartado:

Aunque su labor es similar, la comparación teórica de ambos modelos resulta compleja, pues solo podemos afirmar datos que, objetivamente, suelen destacarse en la bibliografía consultada como los grandes rasgos diferenciadores entre un algoritmo y otro. Entre esas características, lo más reseñable es lo relativo al trabajo del algoritmo. Ambas técnicas se basan en principios matemáticos, probabilísticos y estadísticos diferentes, y tales principios determinan su funcionamiento. LSA se basa en las descomposición de valores singulares y las relaciones entre palabras de distintos documentos dentro de un corpus. Por su parte, LDA pone el foco en distribuciones probabilísticas generativas que determinan los tópicos según el peso o la relevancia de los términos que conforman el corpus total. Son los factores en los que basan su funcionamiento los que determinan que los resultados, comúnmente, tiendan a coincidir en las siguientes líneas teóricas:

Según apuntan Alash y Al-Sultany, LSA obtiene mejores resultados que LDA a la hora de clasificar y extraer temas de un corpus de documentos, si bien ya hemos indicado con anterioridad que ninguno de los modelos resulta óptimo. Sin embargo, existe un número elevado de estudios basado en la extracción de tópicos latentes a partir de distintas redes sociales, muchos de ellos focalizados en la presencia de etiquetas, menciones o fuentes externas que puedan mejorar los resultados finales. Para ellos, tener en cuenta estos elementos a la hora de seleccionar el corpus logra reducir la diferencia en el *coherence score* de LSA respecto a LDA, no siendo para este último tan relevante la presencia de etiquetas en los tuits «because of LDA structure that tries to extract all topics in the text. While LSA is semantic structure method that tries to collect text share the same subject into the same topics. Therefore, the number of k topics in LSA is less than and coherence score values are less than LDA» (2-8).

Además de estos rasgos que tienden a ser comunes en las comparaciones de ambos modelos, es recurrente discutir cuál de los dos algoritmos funciona mejor para conjuntos de datos pequeños y grandes. Sobre esto, las opiniones son divergentes, pues lo cierto es dependiendo del estudio, resultan ser más útiles para según qué cantidades de datos. En lineas generales, LSA es más aceptado en textos breves, sin embargo, en palabras de Jacobucci et al., «LDA has been found to generally produce more coherent topics» (57). En contraposición, Kalepalli et al. aseguran que en su investigación LSA encontraba resultados más coherentes y concretos en cantidades de datos más grandes, mientras que LDA

funcionaba mejor con la cantidad más reducida (1250). Betak y Williams indican que la clave está en el uso de ambos algoritmos, pues se complementan para la interpretación de los resultados (102) y no atiende al carácter de los datos a procesar.

LSA y LDA ofrecen resultados similares. En el capítulo 5 se ofrece una muestra de cinco estudios en los que ambos modelos analizan corpus conformados por *short texts*. Así se demuestra que ambos métodos son capaces de clasificar conjuntos de textos breves como corpus compuestos por tuits cuya temática se centra en los hábitos de vida saludables o en temas de salud pública.

Teniendo en cuenta estas observaciones, lo más adecuado es determinar la evaluación comparativa del comportamiento de los modelos en la práctica. Dependiendo de los pasos implicados en sus ejecuciones, cada uno de los algoritmos conseguirá resultados de manera diferente. Por ello, la comparativa será completada una vez hayan sido interpretados los datos obtenidos por ambos modelos (véase 6.5 «Comparativa práctica»).

Capítulo 4

Nuestra contribución

Una vez hemos abarcado el aspecto teórico de *topic modeling*, procedemos con el apartado en el que ofreceremos dos códigos ajustados al objetivo del proyecto, que es la generación de dos herramientas informáticas capaces de discernir tópicos entre un corpus extenso de tuits con temática lingüística. En primer lugar, explicamos la creación del corpus. A continuación, enumeramos y explicamos las bibliotecas comunes utilizadas en los modelos LSA y LDA. Y, en la última sección del apartado, ejemplificamos los pasos comunes de los dos modelos: preprocesado, diccionario y *coherence score*.

La sección introductoria dará paso a la explicación de los códigos LSA y LDA, en los que la mayoría de los apartados serán comunes: explicación de las bibliotecas propias, desarrollo de las funciones empleadas para la generación del modelo, visualización de los datos y representación de los resultados. El punto final en el que comprobaremos la utilidad de sendos códigos vendrá dado por el análisis de tuits no pertenecientes al corpus original.

Enlaces a los códigos depositados en Google Colab:

- LSA: `https://colab.research.google.com/drive/1ErHAYVuxMBWM60h2uFeM7sHRJubHjNFd#scrollTo=znAuH0tPbrOR`
- LDA: `https://colab.research.google.com/drive/1RvsB1YoX36NLR52h1JkImu_3_J9jmAdt?usp=sharing`

4.1. Introducción

A lo largo de este primer apartado observaremos las características de los documentos que conforman el corpus, así como las bibliotecas comunes a los modelos LSA y LDA. Finalmente, desarrollaremos cuáles son los pasos comunes entre ambos.

4.1.1. Corpus

El entrenamiento de los algoritmos de *topic modeling* como técnica para la detección de tópicos conlleva la elaboración de un código de programación formado por distintas etapas que conducen hasta el procesamiento del conjunto de documentos.

El procesado de datos requiere de un corpus extenso. El conjunto de documentos se compone de tuits recibidos por la cuenta de la Real Academia Española desde el 2 de diciembre de 2020 hasta el 8 de febrero de 2021. En total, tal y como figura en el cuadro 4.1, el corpus que conforman el conjunto de textos a procesar por los dos modelos de *topic modeling* consta de 9.700 tuits que incluyen la etiqueta «#dudaRAE», los cuales fueron rescatados de un compendio de aproximadamente 30.000 tuits. En la recopilación inicial se incluían, además de los tuits que conforman el conjunto de textos a procesar, los retuits, las citas y las respuestas a estos tuits.

Información sobre los datos a procesar	
Total tuits en la base de datos	c. 30.000
Total tuits en conjunto de documentos	9.700

Cuadro 4.1: Corpus completo de tuits

Los tuits seleccionados para la base de datos que conforma el conjunto de documentos a trabajar por el algoritmo tienen, en general, la forma que muestran los siguientes ejemplos:

- «@RAEinforma ¿Ha aceptado a regañadientes? #dudaRAE»
- «¿Andadura o anduviese? #dudaRAE @dudaRAE»
- «#dudaRAE @RAEinforma @RAEinforma #dudaRAE tiempo de prepararse o tiempo para prepararse, ¿hay alguna diferencia?»

Sin embargo, algunos de ellos, además de la etiqueta y la mención común a todos los tuits de la base de datos, añaden otras distintas. Cuentan también con enlaces a recursos web o audiovisuales con el fin de complementar la consulta que están realizando.

Una vez conocida la naturaleza del conjunto de documentos, podemos centrarnos en los elementos comunes que compondrán el código: bibliotecas y pasos.

4.1.2. Bibliotecas

El primer paso para lograr que el código alcance cierto resultado es la importación de algunas bibliotecas. En informática entendemos por bibliotecas el conjunto de funciones que encapsulan funcionalidades en un área concreta.

Las bibliotecas Python son herramientas ya existentes, por regla general de código abierto, que ayudan a los programadores a lograr objetivos dentro de su código, reduciendo en tiempo y líneas la realización de sus programas. De esta manera, y con la certeza de que estas herramientas cuentan con el respaldo de la comunidad que previamente las ha puesto a prueba, aseguramos un mínimo de eficiencia en nuestro código.

A lo largo del presente apartado prestaremos únicamente atención a aquellas que comparten los modelos LSA y LDA. Las bibliotecas que usen específicamente alguno de los dos métodos serán desarrolladas en el apartado correspondiente dentro de la explicación de los códigos LSA y LDA. Las bibliotecas prestan diversas funcionalidades, por lo que dependiendo de su aplicación y objetivo, podemos distinguir los siguientes tipos:

1. Para el manejo de archivos;

2. Para el preprocesado de datos;
3. Para la generación del modelos;
4. Para la visualización de resultados.

Para la aplicación de los modelos entrenados en este trabajo, las principales bibliotecas comunes utilizadas han sido `urllib` para cargar archivos, y `NLTK` y `SpaCy` para el preprocesado de los datos. Además, `Gensim` ha sido empleada para ejecutar los modelos y `matplotlib` para visualizar resultados.

Manejo de ficheros: `urllib.request`

A la hora de afrontar un procesado de lenguaje natural, hemos de tener en cuenta que por regla general nos vamos a enfrentar a uno o varios archivos de gran volumen. Por ello, son necesarias bibliotecas y módulos que ayuden al manejo de ficheros.

Es importante en el procesamiento del lenguaje natural ser capaz de manejar los archivos dentro de nuestro código. Si dichos archivos se encuentran en línea, necesitamos el módulo `urllib.request` de la bibliotecas `urllib`, que permite utilizar recursos identificados por URLs. Con estos dos módulos podremos manejar archivos externos, bien se encuentren en línea, bien en local.

Preprocesado: `SpaCy` y `NLTK`

Para llevar a cabo el procesamiento del lenguaje natural es necesario un preprocesado. Llevarlo a cabo sin emplear ciertas bibliotecas resulta altamente desaconsejable, por eso hemos empleado varias para lograr un buen preprocesado de datos.

`SpaCy` es una biblioteca altamente eficaz en el ámbito de la lingüística computacional. Se trata de un *software* de código abierto focalizado en el procesamiento del lenguaje natural avanzado en Python. Cuenta con tres corpus por cada uno de los idiomas que soporta. Estos corpus, cuya diferencia principal reside en sus tamaños, fueron entrenados con noticias periodísticas. Decidir emplear uno u otro puede alterar el resultado final. Destaca por su rapidez y por sus procesos de asignación. Está diseñada para la creación y recopilación de datos reales, ofreciendo anotaciones lingüísticas, *tokenización*, etiquetado, análisis sintáctico y de dependencias existentes en el conjunto de documentos, identificar similitudes, entre otros pasos importantes del preprocesado.

Dentro de las bibliotecas empleadas para el procesamiento del lenguaje natural, quizá la más usada sea la biblioteca `NLTK` y sus módulos. La Natural Language Toolkit[1] es la plataforma líder para la construcción de programas en Python basados en el lenguaje humano. Incorpora más de cincuenta recursos de corpus y léxico en una interfaz sencilla. A través de ellos logramos procedimientos básicos en el preprocesado de datos: clasificación, *tokenización* o *stemmatización*, entre otros.

Entre estos procesamientos destaca el módulo dedicado a la eliminación de *stopwords*: aquellas palabras que pueden ser frecuentes en el conjunto de documentos a procesar pero que no han de tenerse en cuenta debido a su carencia de significado o relevancia para los resultados esperados. La biblioteca cuenta con distintas listas de *stopwords* según el

[1] `https://www.nltk.org/`

idioma que se indique, listas a las que se puede añadir manualmente términos para evitar problemas en los resultados del proceso.

Tras el preprocesado, resultado es la creación de una lista de palabras por cada uno de los tuits del corpus, generándose una gran lista de listas de gran tamaño que usaremos en pasos posteriores del procesado. A modo de ejemplo vemos como el siguiente tuit:

> @RAEinforma ¿Ha aceptado a regañadientes? #dudaRAE», «¿Andara o anduviese? #dudaRAE @dudaRAE

Se ve reducido a la siguiente lista de *tokens*:

```
['resultado', 'aceptado', 'regañadientes', 'andara', 'anduviese']
```

Generación de modelos: `Gensim`

`Gensim` es una biblioteca de Python utilizada para modelado de tópicos, indexación de documentos y minería de similitud de texto, entre otras funcionalidades. Es una biblioteca de código abierto ampliamente utilizada para el procesamiento del lenguaje natural que trabaja para el modelado de tópicos sin supervisión. Está diseñada para manejar grandes colecciones de texto con los algoritmos `Word2Vec`, `LSI`, `LSA` y `LDA`, detectando automáticamente la estructura semántica de los documentos procesados según los patrones de coocurrencia de *tokens*.

El hecho de que sea un sistema no supervisado implica que, salvo la elección del corpus y del parámetro, todo lo haga el modelo por sí mismo. Una vez encuentra patrones, cualquier documento en texto plano puede ser expresado con esta representación semántica. Para nuestra investigación, hacemos uso de las varias funcionalidades de `Gensim` aplicadas a LSA y LDA.

Las funcionalidades mencionadas en las que se basa su diseño son en sí mismas ventajas que explican la popularidad de la biblioteca cuando se trata del procesamiento del lenguaje natural y modelado de tópicos. Por ello, para la composición del código que permite el trabajo de los algoritmos de entrenados, `Gensim` proporciona las siguientes funcionalidades: preprocesamiento, la creación del diccionario y el corpus necesarios, la creación de los algoritmos y el cálculo de la coherencia del modelo.

Visualización de resultados: `matplotlib`

La manera en la que `Gensim` muestra su modelo es poco visual, por lo que es recomendable hacer uso de bibliotecas centradas en esa labor. Para cubrir esta carencia recurrimos las gráficas proporcionadas por la biblioteca `matplotlib`.

4.1.3. Pasos comunes: apertura y lectura de archivos, preprocesado, diccionario y coherence score

LSA y LDA comparten no solo su objetivo y algunas de las bibliotecas, sino que también tienen en común las fases de su ejecución. A lo largo del presente apartado veremos cuatro de sus pasos comunes: apertura y lectura de archivos, preprocesado, diccionario y *coherence score*.

Apertura y lectura de archivos

Una vez hemos desglosado las bibliotecas que nos ayudaran a continuación, podemos proceder con el el preprocesado. En nuestro caso lo hemos llevado a cabo en dos partes.

Antes de generar las funciones que limpien cada documento del código es necesario generar la que cargue el corpus con los más de nueve mil tuits. La función toma un documento de una URL y lo lee línea por línea, coincidiendo cada una de ellas con un tuit. Lo decodifica en `utf-8` para su posterior manejo.

Preprocesado

Como se ha mencionado previamente, el preprocesado de los documentos es imprescindible para trabajar el análisis del lenguaje natural. Tanto LSA como LDA llevan a cabo un preprocesado que limpia los tuits hasta reducirlos a su forma más refinada en aras de favorecer los resultados de los algoritmos. En este apartado explicaremos el preprocesado común a ambos modelos, pudiendo presentar mínimas diferencias. En ese supuesto, el preprocesado particular se explicará de manera concreta en el apartado posterior dedicado a cada uno de los dos códigos. Ilustraremos este proceso con un ejemplo práctico.

Partimos de este tuit, incluido en el conjunto de documentos:

> @RAEinforma #dudaRAE Estimados, está bien dicha la frase «aquellos muertos que fallecieron en la pandemia». Me parece que de pretendería recalcar un periodo de tiempo.\n#dudaRAE

Observamos que entre sus caracteres se incluyen espacios en blanco o saltos de línea «\n», menciones a la cuenta de la Real Academia Española que implican símbolos como «@», la etiqueta común a todos los tuits considerados para el conjunto a analizar marcada por la almohadilla (#), barras (\), signos de puntuación y `stopwords`. Todos estos caracteres, han de ser eliminados en el preprocesado para conseguir una limpieza del conjunto óptima. Por ello, el primer paso es eliminar saltos de línea, espacios en blanco, caracteres innecesarios entre los que se incluyen las comillas y las barras, además de eliminar la puntuación. En este momento, si el tuit en cuestión tuviera, se eliminarían también enlaces. Una vez ejecutada esta parte del preprocesado, el tuit tiene la siguiente apariencia:

> dudaRAE Estimados, está bien dicha la frase aquellos muertos que fallecieron en la pandemia Me parece que de pretendería recalcar un periodo de tiempo dudaRAE

Como se muestra, la etiqueta no ha desaparecido, tampoco las comillas, las barras y la puntuación. No obstante, el siguiente paso en la limpieza termina de eliminar aquellos caracteres irrelevantes para el procesamiento que llevará a cabo el algoritmo. Esta fase del preprocesado es la que incluye la *tokenización* acompañada de más limpieza. Como se ha mencionado previamente, la *tokenización* transforma el documento en una lista de *tokens*. Estos son etiquetados por la biblioteca `SpaCy`: cada documento está formado por *tokens* y a estos se les añade lo que se denomina «la token», es decir, «la documentación para obtener una idea de qué datos posee cada ficha antes de que la solicitemos y para obtener información específica» (Mayo). Esta información es lo que llamamos categorías.

SpaCy etiqueta cada palabra según su categoría gramatical favoreciendo el preprocesado y ofreciendo posibles utilidades para diferentes objetivos en el procesamiento del lenguaje natural. En el preprocesamiento, Gensim se sirve de sus utilidades para participar en la limpieza de los datos previa a la composición y a la ejecución del algoritmo, incluyendo por ejemplo código destinado a eliminar la puntuación innecesaria. El resultado es una lista de palabras por cada uno de los tuits que componen el corpus:

```
['esta', 'bien', 'dicha', 'la', 'frase', 'aquellos', 'muertos',
'que', 'fallecieron', 'en', 'la', 'pandemia', 'me', 'parece', 'que',
'de', pretendería', 'recalcar', 'un', 'periodo', 'de', 'tiempo',
'dudarae']
```

Después de la *tokenización*, se eliminan los caracteres innecesarios: la almohadilla, las comillas, la puntuación. Sin embargo, aquellas palabras consideradas stopwords forman todavía parte de la lista. Por consiguiente, el paso a seguir en el preprocesado es la eliminación de las stopwords. Para ello, importamos el paquete de NLTK que incluye las stopwords en español, lo convertimos en un conjunto de palabras y, a ese conjunto, el añadimos el fichero conformado por las stopwords seleccionadas manualmente como, por ejemplo: «dudarae». Y, finalmente, las eliminamos del conjunto total:

```
['aquellos', 'muertos', 'fallecieron', 'pandemia', pretendería',
'recalcar', 'periodo', 'tiempo']
```

Diccionario

Una vez preprocesado, podemos proseguir con la creación del diccionario y del corpus pertinentes para cada algoritmo. El módulo corpora se emplea para su creación mediante la identificación de las palabras relevantes de cada documento. El documento del que parte es la lista de listas mencionada con anterioridad. Almacena las palabras que considera relevantes en un orden no alfabético. A cada palabra le otorga un valor entero que corresponde a la posición dentro del corpus en la que apareció por primera vez. Para llevar a cabo esta asignación la función asocia a la primera palabra que cumple los requisitos arriba detallados el valor 0, a la segunda, el valor 1, y así sucesivamente. En otras palabras, le asigna un identificador que se corresponde con un valor entero. Si una palabra se repite más adelante en el corpus, no le asigna el valor entero consecutivo, sino el mismo que le asignó la primera vez que apareció. La función devuelve un diccionario alfabéticamente desordenado, y los valores enteros correspondientes a dichas palabras. En el ejemplo siguiente, es4 el identificador de «tiempo», y 19 de «pandemia». Los unos que aparecen tras las comas indican el número de veces que aparece la palabra en concreto en este tuit:

```
[(4, 1), (19, 1)]
[('tiempo', 1), ('pandemia',1)]
```

Los diccionarios son objetos que permiten la asignación de un identificador numérico a cada palabra única. Las funciones Gensim implicadas en la creación del diccionario son dos, doc2bow y id2bow, para LSA y LDA respectivamente. Los argumentos de ambas son el corpus, el diccionario y el conjunto de hiperparámetros pertinentes. Como decisión de

diseño, dentro del diccionario existe la posibilidad de eliminar las palabras con pocas apariciones, así como también aquellas demasiado frecuentes con la función `filter_extremes` que incluye en el diccionario aquellas palabras que aparecen, por ejemplo, en al menos 20 documentos y aquellas que aparecen en no más que el 50 % de los documentos. Esto se define con `no_below` y `no_above` respectivamente, parámetros editables según el criterio que quiera imponerse al diccionario. Esta posibilidad permite que el conjunto de términos que forman el diccionario quede refinado, de manera que favorece al procesamiento ejecutado por el algoritmo. Consecuentemente, se eliminan palabras que habían pasado el preprocesado, y por lo tanto no formarán parte de los resultados finales del modelo.

Con este último paso tenemos el diccionario y el corpus generados. El conjunto de documentos está listo para ser procesado por cada uno de los algoritmos, en los que emplearemos el siguiente y último de los pasos comunes, el *coherence score*.

Coherence score

Coherence score se usa para calcular la coherencia del modelo dentro de *topic modeling*. Consiste en aplicar cuatro pasos que sirven para la evaluación de *topic model*: segmentación, estimación probabilística, medida de confirmación y adicción. `Gensim` permite el cálculo de los datos de coherencia del modelo por número de tópicos. La función se lleva a cabo por la importación del modelo de coherencia de la biblioteca que implementa cuatro estados para la evaluación de la función llamada. Estos son segmentación, estimación de probabilidad, confirmación de la medida y agregación. La coherencia puede facilitar la elección del número de tópicos asignado al algoritmo según el valor (entre `0` y `1`) con el que puntúe la relación de los términos dada en cada distribución de tópicos. Cuanto más cercano sea el valor a `1`, mayor peso tendrá el *token* dentro del tópico.

4.2. LSA

Una vez tenemos claro los puntos en común entre LSA y LDA, podemos proceder con el desarrollo del primero de los modelos de *topic modeling*. A lo largo del siguiente apartado desmembraremos los pasos que hemos seguido antes de alcanzar un resultado satisfactorio a través de nuestro código. En primer lugar, veremos las bibliotecas propias del modelo que no estaban presentes en LDA. El procesado del lenguaje natural a través de la técnica de *topic modeling* LSA se compone, en gran parte, de pasos similares a los vistos en apartados anteriores: preprocesado, construcción de un diccionario, aplicación del modelo y obtención de resultados de dicho modelo. Al igual que ocurría en el apartado anterior, el paso final de un análisis de este estilo es comprobar los resultados. Por ello añadiremos un breve corpus con el que veremos cómo encajan algunos de sus documentos en el modelo ya creado.

4.2.1. Etapas de la programación y definición de funciones

Los pasos por los que atraviesa el código hasta su final son:

1. Importación de bibliotecas;
2. Lectura e importación del corpus;
3. Preprocesado: limpieza, *tokenización* y eliminación de `+stopwords+`;

4. Obtención del modelo LSA y resultados;
5. Visualización e interpretación de los resultados obtenidos;
6. Predicción de tópicos para nuevos tuits.

Bibliotecas

LSA usa las bibliotecas generales para la lectura de archivos (véase 4.1.3). Para el preprocesado, el módulo `re` facilita la posibilidad de emplear expresiones regulares. Las *regular expressions*, también conocidas como `regex`, son una secuencia de caracteres que conforma un patrón de búsqueda dentro de una cadena, es decir, encuentra un fragmento de una cadena que cumpla uno o varios requisitos. LSA emplea el método `regex` de `nltk.tokenize` para la identificación de expresiones regulares. La biblioteca `nltk` ya ha sido explicada en profundidad anteriormente. Para completar el las bibliotecas que ayudan al preprocesado de LSA incorporamos `spacy.lang.es`, el módulo específico de `SpaCy` para la lengua castellana.

La última de las bibliotecas, que no ha sido nombrada con anterioridad es `WordCloud`, una herramineta que muestra los resultados de manera visual, exponiendo con distintos tamaños y colores los *tokens* mas importantes de los resultados.

Apertura y lectura de archivos

Como también veremos en LDA, las dos primeras funciones se dedican a la primera etapa: la lectura e importación del fichero que contiene los documentos o tuits. La finalidad última de estas funciones es devolver el fichero correctamente cargado y leído habiendo procesado como argumentos el fichero y la dirección donde se alberga.

Preprocesado

Tras la lectura de archivos, comienza el preprocesado. La función `limpiar` se encarga de preprocesar el corpus. Para su implementación seguimos diversos pasos: creamos una serie de funciones auxiliares que serán utilizadas por `limpiar`. La creación parte del uso de expresiones regulares y `SpaCy`, herramientas que facilitan gran parte del preprocesado. Con ellas, se construye la función final `limpiar`. Dentro de todos los verbos que encuentre, nos interesan «saber», «querer» y «gustar». Estos verbos carecen de significado dentro del tuit, por lo que eliminarlos supone una mejora de los resultados finales. El problema que acarrearía no haber usado `SpaCy` para llevar a cabo esta tarea era que de haber usado una expresión regular para identificar palabras que empezaran, por ejemplo, por «sab-», aludiendo al verbo «saber», el programa podría haber identificado, y posteriormente eliminado, sustantivos como «sable» o «sabiduría». `SpaCy` identifica todas las esas palabras que empiezan por los caracteres marcados, pero distingue cuáles son verbos y cuáles otro tipo de palabras. Posteriormente prescindimos de los primeros. Del mismo modo, `SpaCy` identifica un gran número de signos de puntuación de los que también le indicamos que se deshaga. En el paso posterior cumplimentamos aquello que `SpaCy` no abarca empleando expresiones regulares. Así, eliminamos las menciones dentro de los tuits, identificando los *tokens* que mencionan a otras cuentas de Twitter y que consecuentemente empiezan por "@", los caracteres numéricos, los *tokens* que comienzan por «#» y se corresponden con *hashtags*. De esta manera conseguimos una serie de funciones que aglutinamos en una sola,

`limpiar`. Con ella conseguimos que reducir a los *tokens* más relevantes los documentos del corpus.

El segundo paso que implementaremos en el preprocesado del corpus consiste en aumentar la lista importada de `stopwords`. Para ello, importamos la lista de `stopwords` en español. Una vez importada y habiendo comprobado que carece de *tokens* que nos convendría incorporar, generamos un documento de texto con *tokens* añadidos al que ya hemos hecho referencia con anterioridad. En nuestro código aglutinamos las palabras de dicho documento con las `stopwords` importadas en español bajo una misma variable.

Una vez tenemos las `stopwords` cargadas y definida la función `limpiar`, podemos generar la función que lleve a cabo el preprocesado de todo el corpus. La función pone en minúsculas todas las palabras de los tuits, luego los *tokeniza* y añade cada uno de los *tokens* a las listas correspondientes. Estas listas ya están limpias y preprocesadas, gracias a la función `load_data`, que usaba la función `limpiar`.

Obtención del modelo LSA

Una vez ha finalizado el preprocesado pasamos a preparar el corpus para el modelo LSA. Se crea la función que ayudará a generar el modelo LSA. La función crea un diccionario con su matriz. Para que LSA funcione necesita asignar, como hemos indicado con anterioridad, valores numéricos a determinadas palabras. Para ello, la función toma los tuits preprocesados y genera un diccionario con las palabras que considera relevantes. Entre los parámetros configurados, limitamos la elección de *tokens* a aquellos que en todo el corpus aparecen no menos de 20 veces, y no más de la mitad de las veces. Con esto, podemos implementar el método `.doc2bow()`:

> The function `.doc2bow()` simply counts the number of occurrences of each distinct word, converts the word to its integer word id and returns the result as a sparse vector.

El `.doc2bow()` generado recibe el nombre de «vector de dispersión», un vector de gran tamaño en el que la mayor parte de sus elementos son cero. El número de ceros es lo que se considera el *sparse*. Este brinda a la siguiente función dos de los tres elementos que necesitará para generar finalmente el modelo. La nueva función incorpora el `lsamodel`, que tomará un número indicado de tópicos, y las palabras del diccionario.

El siguiente paso es la creación de otras dos funciones que concluyan con la generación del modelo. Como bien indicamos con anterioridad, LSA funciona bajo el paraguas de `Gensim`, y las herramientas que este proporciona son no supervisadas, por lo que el sistema generará el modelo de manera automática. Para ello toma las palabras del diccionario y el valor numérico asignado en el vector de dispersión, y con ello genera la función principal del modelo. El resultado es el *coherence score* de los distintos *tokens*.

Como ayuda al investigador, que es quien elige el número de tópicos que compondrán el estudio, se genera una última función que toma el diccionario, los valores asignados a sus palabras y la coherencia de la función anterior. La función es capaz de generar una gráfica (véase figura 4.1) que predice el número adecuado de tópicos para un determinado corpus. Con esto se abre el siguiente apartado, la visualización de datos.

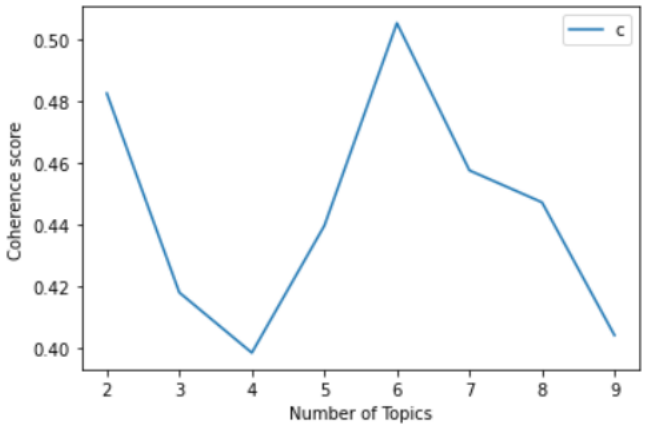

Figura 4.1: Predicción del número óptimo de tópicos en LSA
Fuente: Código LSA

Ahora ya contamos con todas las funciones necesarias definidas. El siguiente paso es generar el código que aglutine las funciones y genere el modelo. El modelo requiere al analista que elija el corpus, pero también que indique el número de tópicos y además, cuantos *tokens* sacar por tópico. Para ellos nos basamos en los resultados de la gráfica generada con `graph_plot` (véase figura 4.1). En esta elección debemos tener en cuenta que si el número de tópicos es muy alto, los tópicos serán muy parecidos, y también serán parecidas las palabras que en ellos aparezcan. Por el contrario, si elegimos escasos tópicos, la disparidad entre ellos será demasiado grande, y clasificaciones poco aceptadas podrían ocurrir. En cuanto a la elección del número de palabras que aparecen por tópico, hemos de indicar que al elegir un número bajo podemos perder palabras menos relevantes que sean clave a la hora de discernir los tópicos. En LSA se nos muestran las palabras que componen un tópico, pero no cuál es el propio tópico. Es trabajo del analista de datos indicar el tópico de un determinado grupo de *tokens* que ha aglutinado el modelo. Por ello, elegir un número excesivamente alto puede suponer la repetición de muchas de las palabras, lo que dificultaría el trabajo de identificar tópicos del analista.

El modelo se genera aplicando la función `create_gensim_lsa_model`, usando el `dictionary` y el valor entero asignado a cada uno de los *tokens* en `doc_term_matrix`. La función contiene los siguientes argumentos:

- `clean_text` es el resultado del preprocesado (función `preprocess_data`) del corpus de textos, encontrado en `document_list`. Por lo tanto, el primer argumento provee a `create_gensim_lsa_model` de los tuits limpios.
- `number_of_topics` establece el número de tópicos. Esta es una decisión tomada por el investigador, que se puede valer de la predicción de tópicos óptimos ofrecida por LSA (véase figura 4.1).
- `words` es un argumento que indica cuantas palabras queremos extraer de cada tópico. A partir de ellas, es nuevamente el investigador el que discierne el tópico bajo el que se encuentran.

```
model,dictionary,doc_term_matrix=create_gensim_lsa_model(
   clean_text,number_of_topics,words)
```

Listing 4.1: Código de creación del **Modelo LSA**

Resultados: visualización e interpretación

Los resultados se muestran de manera poco gráfica, por lo que hemos optado por componer unas tablas que reflejen los resultados disponibles en los siguientes cuadros:

Tópico 0	
0.725	Coma
0.297	Punto
0.255	Verbo
0.213	Oración
0.212	Después
0.174	Mayúsculas
0.159	Expresión
0.131	Puntos
0.141	Tras
0.104	Nombre

Cuadro 4.2: LSA: Tópico 0

Tópico 2	
0.838	Verbo
-0.318	Coma
-0.217	COVID
0.186	Expresión
0.118	Conjugación
-0.117	Punto
0.084	Oración
-0.075	Después
-0.073	Singular
-0.070	Pasado

Cuadro 4.4: LSA: Tópico 2

Tópico 4	
0.685	Mayúscula
-0.405	Expresión
-0.307	Minúscula
-0.273	Coma
0.199	Nombre
0.158	Inicial
0.102	Escriben
0.094	Puntos
0.090	Nombres
0.087	Escribirse

Cuadro 4.6: LSA: Tópico 4

Tópico 1	
-0.945	COVID
0.173	Coma
0.153	Expresión
-0.113	Virus
-0.110	Verbo
0.060	Punto
-0.051	Enfermedad
-0.044	Masculino
-0.037	Femenino
-0.037	Medios

Cuadro 4.3: LSA: Tópico 1

Tópico 3	
0.835	Expresión
-0.327	Verbo
0.230	Mayúscula
-0.209	Coma
-0.160	COVID
0.110	Minúscula
0.096	Tilde
0.064	Nombre
0.053	Siempre
0.049	Inicial

Cuadro 4.5: LSA: Tópico 3

Tópico 5	
0.797	Oración
0.385	Tilde
-0.233	Mayúscula
-0.190	Coma
-0.172	Verbo
-0.121	Expresión
-0.115	Minúscula
0.070	Punto
0.066	Plural
0.058	Final

Cuadro 4.7: LSA: Tópico 5

Entendiendo que el visionado de las tablas puede resultar confuso, desmembraremos a continuación cada una de ellas. Aprovechamos para indicar que aspecto lingüístico refleja cada una de ellas, pues, como ya hemos dejado claro con anterioridad, no es una labor que haga el modelo por sí mismo.

Tópico 0: posición de palabras respecto a signos ortográficos

El cuadro 4.2 refleja los *tokens* que hacen referencia a la posición de los signos en ortográficos junto a otras palabras. Se destacan el punto y la coma, de

los que se pregunta su posición tras o después de palabras cuya categoría gramatical sea nombre o verbo. Por lo tanto, este primer tópico tiene un alto componente ortográfico.

Tópico 1: COVID

El segundo (véase cuadro 4.3) se aleja del ámbito lingüístico, aunque es comprensible dada la situación pandémica actual. «COVID» es la palabra que más peso tiene no solo en el tópico, sino entre todos los demás *tokens* que encabezan el resto de tópicos. La idea se soporta con la presencia de términos como «enfermedad» o «virus». Además, queda patente que las dudas vienen en cuanto al género de la palabra «COVID», pues «masculino» y «femenino» forman parte del tópico. Asimismo, se alude a «medios», haciéndonos inferir, no sin prudencia, que se refiere a los medios de comunicación.

Tópico 2: verbos

El tercero de los tópicos (véase cuadro 4.4) alude a los verbos, ya que la palabra verbo tiene un peso muy alto. Se acompaña de palabras clave para el tópico, como «conjugación», «singular» o «pasado».

Tópico 3: tildes

El cuarto tópico (véase cuadro 4.5) alude a la acentuación de palabras y las tildes. Bien es cierto que la asignación de este tópico resulta más compleja y forzada, pero al ser el único de ellos con «tilde» entre sus diez palabras con más peso, y no apareciendo esta en el resto de tópicos a excepción del último, podemos afirmar, que, en relación a otras palabras presentes como «siempre» o «inicial», los tuiteros pueden estar preguntando por la presencia de la tilde en ciertas posiciones de la palabra.

Tópico 4: mayúsculas y minúsculas

Este quinto tópico (véase cuadro 4.6) versa sobre el uso de mayúsculas y minúsculas. Estas dos palabras copan dos de las tres primeras posiciones de la tabla. La presencia de «nombre» o «inicial» nos hace atisbar que los usuarios tienen dudas sobre cuándo escribir con una u otra opción.

Tópico 5: tópico comodín

A diferencia de los cinco tópicos anteriores, ningún tópico en claro puede extraerse en este caso (véase cuadro 4.7). Llegamos a la conclusión de que actúa como un tópico sin un tópico definido. Los términos presentes en este tópico tienen un peso menor total en comparación con los hallados en el resto de tópicos. Este tópico se caracteriza por la pluralidad de palabras. Son *tokens* que encontramos en otros tópicos, lo que nos lleva a pensar que es el tópico en el que se enmarca todo lo que no encaja con otros tópicos.

Para la representación de los resultados, hemos optado por hacerlo de manera visual. De este modo, hemos ilustrado los *tokens* que más aparecen en la figura 4.2: «expresión», «verbo» y «coma» copan los tres primeros puestos, apareciendo cada una de ellas más de cuatrocientas veces en todo el corpus. «Oración», «mayúscula» y «verbo» se encuentran por encima de las doscientas cincuenta apariciones.

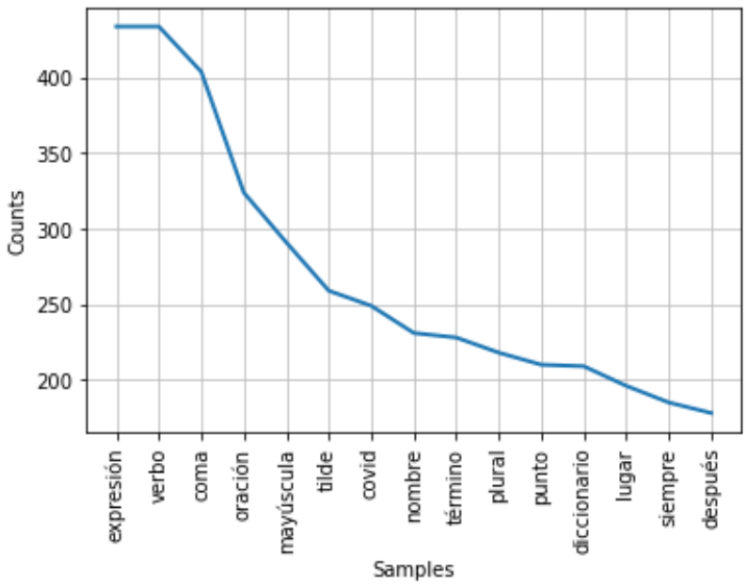

Figura 4.2: Palabras con mayor número de apariciones
Fuente: Código LSA

Con el fin de mostrar qué palabras tienen mayor peso en este corpus, proponemos visualizarlo a través de una nube de palabras (figura 4.3). La nube refleja que «expresión», «verbo», «coma», «oración» y «nombre» son las que tienen un peso mayor para el modelo. En menor tamaño encontramos otras como «mayúscula», «diccionario» o «tilde». Los resultados coinciden con los mostrados por la figura 4.2. Ambas figuras muestran varias de las palabras que aparecen en las tablas del apartado «Resultados: visualización e interpretación» de la sección «LSA»:

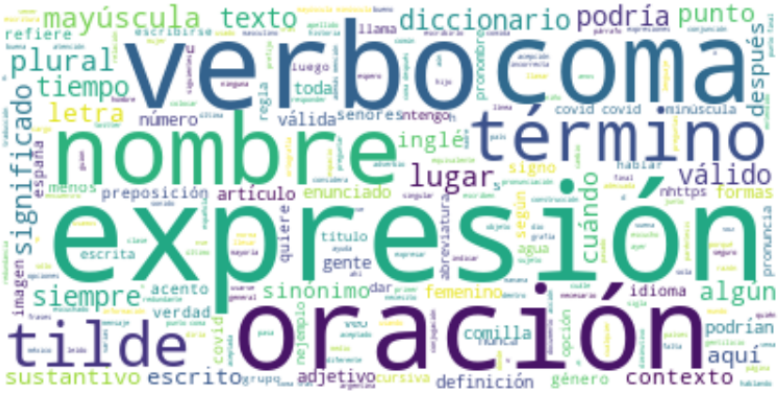

Figura 4.3: Nube que refleja las palabras con más peso para el modelo
Fuente: Código LSA

4.2.2. Predicción de tópicos para nuevos tuits

En este apartado utilizaremos dos ejemplo para observar si los resultados del modelo son coherentes. La finalidad de un estudio de este tipo es aplicar los resultados a un nuevo corpus, y ver si los documentos contenidos en él encajan con los tópicos extraídos en la aplicación del modelo. En nuestro caso, hemos cargado un documento que contiene un

total de treinta y siete tuits. Para ver si la clasificación funciona con nuevos tuits, veremos dos casos en los que adelantamos obtendremos resultados fructíferos.

Ejemplo 1

Una vez se ha generado la relación de palabras que han superado el preprocesado con los valores enteros y a su vez se ha indicado el número de veces que aparece en el tuit, podemos analizar si este tuit encaja en alguno de los tópicos propuestos. Para ello, y al igual que ocurrirá en LDA (véase Ejemplo 1 de LDA), emplearemos el mismo ejemplo:

> #RAEconsultas #dudaRAE En la frase La tortilla, con cebolla está más rica, ¿puede ir esa coma tras el sujeto?, ¿sería una posible excepción de la coma criminal? Porque hay un matiz de diferencia con La tortilla con cebolla está más rica. ¿Cómo lo marco si no es con la coma?

Una vez preprocesado, el resultado es el siguiente:

> En la frase La tortilla con cebolla está más rica puede ir esa coma tras el sujeto sería una posible excepción de la coma criminal Porque hay un matiz de diferencia con La tortilla con cebolla está más rica Cómo lo marco si no es con la coma

El segundo paso consiste en averiguar cuáles de las palabras que componen el tuit preprocesado son consideradas relevantes para LSA. Del tuit preprocesado, LSA solo presta atención a cuatro *tokens*:

```
significado 1
ninguna 2
siempre 3
cantidad 1
```

El número indica cuantas veces aparece ese *tokens* en el tuit preprocesado. Es decir, en la cadena conformada por el tuit, «siempre» aparece exactamente tres veces. Existe la posibilidad de comprobar en qué posición se encuentra cada *token* dentro del diccionario generado a partir del primer corpus de más de nueve mil tuits. Con esto demostramos que no se trata de un elemento alfabéticamente ordenado:

```
(1, 1), (7, 2), (59, 3), (383, 1)
```

Esto indica que «significado» aparece en la posición `1`, «ninguna» en la `7` y «cantidad» en la `383`.

El último paso es ver en qué tópico de los anteriormente propuestos encaja este tuit. El resultado se muestra de la siguiente manera:

```
[(0, -0.20980927229940785),
(1, -0.058112608011616315),
(2, 0.1648455875562258),
(3, -0.18909357407702648),
(4, -0.1693496918921315),
(5, -0.1222575962586621)]
```

Estos resultados muestran en primer lugar los tópicos del 0 al 5, y tras la coma el vector de dispersión que refleja la probabilidad que tiene el primer tuit de pertenecer a un tópico u otro. Así, vemos que con un `-0.209`, es en el primer tópico donde, según el modelo, es más probable encajar el tuit. Recordamos que no importa que el valor sea negativo, pues lo importante en LSA es cuánto se separa de `0` en cualquiera de las dos direcciones. Para comprobar si el modelo está en lo cierto, volvemos a la tabla del tópico 0 (véase cuadro 4.2), del que dijimos que tenía un componente ortográfico que hacía hincapié en la posición de comas y puntos respecto a palabras de distinta categoría gramatical. Al leer el tuit, vemos que su idea general es el preguntar si una coma puede aparecer en una determinada posición. «Coma», «tras» y «nombre» forman parte del tópico 0. Por lo tanto, concluimos que la aplicación del modelo LSA en este caso resulta exitosa.

Ejemplo 2

En el siguiente y último ejemplo veremos como el modelo LSA carece de una precisión óptima. Para ello, aplicaremos el mismo código, siendo el único parámetro cambiado el tuit a analizar. Este ejemplo volverá a ser utilizado en LDA (véase Ejemplo 2):

> #raeconsultas #dudaRAE @RAEinforma Una pregunta, en la frase 'No importa cuanta cantidad de patata echemos, siempre y cuando no sea ninguna', cuál es el significado. Siempre y cuando sea alguna, o siempre y cuando sea ninguna.

Como ya han sido explicados los pasos intermedios del análisis de nuevos tuits, podemos proceder directamente con los resultados:

```
[(0, 0.0036254605539382295),
(1, -0.0016099528011234803),
(2, 0.0014768018627554715),
(3, 0.002647918412362206),
(4, -0.004209751542360152),
(5, 0.010708726262107343)]
```

Observamos que, a excepción del último de los tópicos, todos los resultados comienzan por `0.00`, con lo que averiguamos que este tuit tiene una probabilidad baja de encajar en cualquiera de los tópicos. Es el tópico 5 quien tiene, por una diferencia escasa, una probabilidad mayor de acoger el tuit. Recordamos que este tópico actuaba como tópico comodín (véase cuadro 4.7), por lo que, tras leer el tuit y ver que hace referencia al significado de una frase, podemos comprobar que ninguno de los tópicos versa sobre ello,

por lo que la tímida asignación al último de los tópicos nos parece acertada, aunque no completamente segura y convincente dada la cercanía a otros tópicos.

Tras estos dos ejemplos podemos concluir que la aplicación del modelo LSA es acertada. Infiere de manera lógica el tópico de los nuevos tuits y los encaja en el modelo anteriormente creado. Obtenemos pues, en nuestra opinión, unos resultados altamente satisfactorios.

4.3. LDA

Una vez hemos desglosado LSA y los elementos comunes a los dos algoritmos de *topic modeling*, podemos proceder con la explicación del segundo de los modelos: LDA.

Como bien se puede inferir de secciones anteriores, el código contará con varias etapas bien definidas, lo que nos ayudará a obtener los resultados asegurándonos de no habernos saltado ningún paso. Tras la aplicación del modelo, comprobaremos con los mismos ejemplos que este se ha aplicado de manera correcta y que los resultados por él ofrecidos son adecuados.

4.3.1. Etapas de la programación y definición de funciones

Las etapas necesarias para conseguir los resultados del procesamiento llevado a cabo por el algoritmo LDA se han establecido, como en LSA, en seis fases diferentes:

1. Importación de bibliotecas;
2. Lectura e importación del corpus;
3. Preprocesado: limpieza, *tokenización* y eliminación de `+stopwords+`;
4. Obtención del modelo LDA y resultados;
5. Visualización e interpretación de los resultados obtenidos;
6. Predicción de tópicos para nuevos tuits.

Bibliotecas

La primera de las etapas es la importación de bibliotecas. Si bien ya hemos identificado aquellas que son comunes a LSA, LDA emplea algunas que le facilitan especialemente la obtención de resultados. En el caso de LDA estas bibliotecas son: `NumPy` en colaboración con `Pandas` para el manejo de grandes cantidades de datos, `Gensim.utils` para el preprocesado, `Pprint` para la impresión de datos y `pyLDAvis` para la visualización.

`NumPy` es una biblioteca de Python especialmente diseñada para analizar grandes volúmenes de datos. Su aportación más relevante es la incorporación de objetos llamados *arrays*. Los *arrays* son matrices o colecciones de datos que representan datos de un mismo tipo en diferentes dimensiones favoreciendo la manipulación de estos. Como LDA trabaja con cadenas de palabras, `NumPy` permite al algoritmo manejar estas variables estructuradas que almacenan elementos de manera consecutiva de manera eficiente. En colaboración con `NumPy`, la biblioteca `Pandas` está diseñada para distintas funciones. Principalmente, define

nuevas estructuras de datos a partir de los *arrays* de `NumPy`, pero con nuevas funcionalidades: lee y escribe ficheros en diferentes formatos: series, bases de datos y paneles.

Por su parte, `Gensim.utils` favorece la fase de preprocesado en LDA en general y la biblioteca `Pprint` ofrece la posibilidad de imprimir estructuras de datos arbitrarias de una forma que sea más legible para el programador.

Por último, como se ha mencionado con anterioridad en la sección sobre teoría de LDA, la biblioteca `pyLDAvis` es un paquete desarrollado concretamente para la visualización de los resultados obtenidos por Latent Dirichlet Allocation. Este proporciona un gráfico en el que aparecen representados los tópicos y la relación de los términos que los forman ordenados por su relevancia, como se mostrará en el análisis de resultados realizado en este mismo apartado.

Ahora podemos definir las funciones que agruparán las funcionalidades incluidas en cada paso del código haciendo más eficiente la ejecución completa.

Apertura y lectura de archivos

Las dos primeras funciones son las que se encargan de agrupar la primera etapa: la lectura e importación del fichero que contiene los documentos o tuits. La función de importación de la base de datos tiene como argumento el nombre que se le otorga al fichero dentro del código. Nombre que, posteriormente, devuelve la función encargada de la lectura del fichero, cuyo argumento es la dirección donde se alberga el fichero y cuya funcionalidad es devolver el propio fichero leído.

Preprocesado

Una vez importado y leído el fichero que contiene los documentos, comienza una etapa clave marcará la calidad de los resultados que obtenga el algoritmo: el preprocesado. En este punto, los datos albergados en el conjunto de documentos han de limpiarse y refinarse para que el modelo pueda procesarlos sin problemas adicionales. Esa limpieza afecta a los saltos de línea, a ciertos caracteres innecesarios como comillas y símbolos, a enlaces y a puntuación, así como también a las ya mencionadas *stopwords*. Esta etapa se agrupa en la definición de dos funciones: la encargada de limpieza y *tokenización* y la encargada de eliminar las *stopwords*.

La primera de ellas implica dos transformaciones: por un lado, el fichero se transforma en un conjunto de documentos que ya no cuenta con caracteres innecesarios ni con enlaces, menciones o símbolos de etiqueta (#) que puedan entorpecer el procesamiento;, por otro, debido a la *tokenización*, ese conjunto de documentos tiene un formato de lista de *tokens* (véase ejemplo práctico de preprocesado en la sección 4.1.3 «Pasos comunes: preprocesado, diccionario y *coherence score*»). Esta lista se almacena en un conjunto que permite al programador pasar las `stopwords` predeterminadas que, una vez almacenadas en un conjunto, permite añadir aquellas que se consideren necesarias al mismo, de modo que esta función alberga también esa utilidad para que ambos pasos se ejecuten al llamar a una única función optimizando su ejecución. Respectivamente, los argumentos de estas dos funciones son la base de datos ya limpiada y *tokenizada* y la base de datos junto al conjunto de `stopwords`.

Obtención del modelo LDA

Con el procesamiento hecho, ya pueden ponerse en marcha las funcionalidades de la biblioteca `Gensim` que llevan a la obtención del modelo LDA: la creación del diccionario necesario para el trabajo del algoritmo. Como se ha explicado en apartados anteriores (4.1.3), el diccionario se crea a partir de la lista de palabras que está almacenada en una variable. El proceso de creación de este diccionario termina de limpiar el documento eliminando las palabras con mucha frecuencia, así como las que no son siquiera relevantes por su poca apariencia en el total del conjunto. Con este diccionario creado, puede procederse a la identificación con valores enteros de cada palabra y a la representación de su frecuencia. Estas creaciones, previamente explicadas dentro de las funcionalidades de la biblioteca `Gensim`, se almacenan en una función que da lugar a la obtención del algoritmo LDA que ya en este punto del proceso realiza su entrenamiento, pues ya conoce las palabras y su peso y, por tanto, los resultados ya pueden ser representados.

Ahora bien, la aplicación del modelo LDA requiere en cada caso determinar sus hiperparámetros de la manera más favorable para los resultados que potencialmente se esperan adquirir. Por ello, el entrenamiento en sí del modelo aplicado al documento concreto requiere contemplar los cambios oportunos que hayan de sufrir los hiperparámetros editables respecto a los que el algoritmo determina por defecto.

La función encargada de calcular el modelo es `LdaModel`. Recibe una serie de parámetros en los que que recibe una serie de parámetros. En este caso, se concretan en el algoritmo los siguientes valores que siguen al corpus y al diccionario:

- `num_topics=7` establece el número de tópicos, siendo cada tópico un conjunto de palabras clave que contribuyen con un peso diferente al tópico en cuestión;
- `random_state=2` determina la semilla usada para la generación y distribución aleatoria de los tópicos;
- `update_every=1` indica la actualización del modelo por cada documento;
- `chunksize` se concreta en `10`, indicando el número de documentos que el algoritmo ha de considerar a la vez para el procesamiento;
- `passes=10` indica las veces que el algoritmo debe procesar el corpus entero;
- `alpha`, este está centrado en la distribución de tópicos que, al estar establecido como automático se basa en una distribución asimétrica previa que infiere directamente de los datos que están siendo procesados;
- `per_word_topics` está establecido en `True` permitiendo la extracción de los tópicos más probables dada una palabra. Esto es que cada palabra se asignará a un tópico o se omitirán: según sean o no relevantes en la totalidad del corpus.

```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=7,
                                            random_state=2,
                                            update_every=1,
                                            chunksize=10,
                                            passes=10,
                                            alpha='auto',
                                          per_word_topics=True)
```

Listing 4.2: Código de creación del **Modelo LDA**

Instanciados todos los valores que concretan el funcionamiento del algoritmo, los resultados obtenidos por el modelo ya pueden ser interpretados. Sin la biblioteca `pyLDAvis` que permite una visualización de los resultados más sencilla de interpretar, los resultados en bruto que ofrece LDA representan una serie de palabras acompañadas de su peso dentro del corpus procesado.

Si se inicializa el modelo a partir de una semilla determinada, la identificación de tópicos y los términos que los conforman son más fáciles de interpretar. Este conjunto de semillas lo facilita directamente el algoritmo cuando no se define el parámetro `random-state`. Así, al ejecutar el modelo, ofrece cada conjunto de semillas que se forman de tópicos y términos. Entre ellos, puede seleccionarse cuál de esos conjuntos representa de manera óptima los resultados esperados del procesamiento. Este método de selección de la semilla para LDA está entrenado en diversos trabajos como, por ejemplo: «Automated Seeded Latent Dirichlet Allocation for Social Media Based Event Detection and Mapping», un estudio publicado en 2020 que pretende demostrar cómo la selección de la semilla facilita la interpretación manual de los resultados pudiendo incluso aplicarse cada semilla a un criterio de clasificación concreto. En su caso, cada semilla facilita la identificación de los distintos tuits según a qué desastre natural refieran. Estas palabras clave extraídas se utilizan para guiar a LDA a modelar un solo tópico relacionado con el término más relevante. Los tuits correspondientes a ese tópico en cuestión pueden estimarse de manera más coherente. La elección de la semilla más favorable a este caso se llevó a cabo a partir de la visualización de los resultados generada por `pyLDAvis`. El paquete ofrecía distintos gráficos según la semilla procesada. Con la semilla 2, los siete tópicos quedaban bastante diferenciados y favorables para la interpretación de resultados.

Aunque este procesamiento implique una intervención manual que puede considerarse inapropiada, lo cierto es que favorece al comportamiento del algoritmo cuando se trata de clasificación de tópicos en tuits, como demuestran estudios como el anteriormente mencionado. La implicación manual viene dada por posibilidades propias del algoritmo de modo que el comportamiento no se falsea, sino que se entrena de múltiples maneras hasta dar con la más impecable para la interpretación. Demostrando así que es capaz de obtener lo esperado si se realizan de manera correcta los pasos implicados en la ejecución de LDA, incluyendo los cambios que precisen sus hiperparámetros editables.

Una vez superada la aplicación del modelo LDA y conseguida la visualización más favorable, el código llega a su fase final: el análisis y la interpretación de los resultados obtenidos.

Resultados: visualización e interpretación

Para la correcta interpretación y análisis de los resultados deben atenderse tres fases internas de esta etapa:

- La exposición de los tópicos, los términos que los conforman y su interpretación;
- La lectura del gráfico obtenido con `pyLDAvis` y la denominación de tópicos;
- El estudio de un ejemplo práctico de asignación de un tuit a uno de los tópicos representados en el gráfico.

De acuerdo con el número de tópicos que se concretó al algoritmo, este trabaja la asignación y la distribución para siete. Por ello, antes de ser visualizado por `pyLDAvis`, LDA obtiene los tópicos que se muestran en el cuadro 4.8.

Tópicos	Términos obtenidos del resultado en bruto del algoritmo LDA
0	0.164 mayúscula, 0.149 números, 0.074 minúscula, 0.071 país, 0.051 primer, 0.050 mayúsculas, 0.041 empezar, 0.039 luego, 0.039 pronuncia, 0.033 general
1	0.204 puntos, 0.116 lugar, 0.076 plural, 0.060 singular, 0.054, pronunciación, 0.048 gente, 0.046 preposición, 0.039 anterior, 0.034 siglas, 0.031 regla
2	0.159 tiempo, 0.061 definición, 0.056 grupo, 0.053 historia, 0.051 común, 0.048 pasado, 0.045 mensaje, 0.044 genero, 0.038 fin, 0.033 vídeo
3	0.230 coma, 0.137 punto, 0.114 comas, 0.078 después, 0.070 comillas, 0.043 espacio, 0.042 tras, 0.020 cita, 0.018 medio, 0.017 suena
4	0.125 enunciado, 0.112 covid, 0.085 tilde, 0.053 ortografía, 0.049 vocal, 0.046 aparte, 0.045 nombres, 0.040 navidad, 0.040 adjetivo, 0.039 guion
5	0.053 verbo, 0.032 imagen, 0.027 siempre, 0.022 texto, 0.020 final, 0.018 diccionario, 0.017 todas, 0.016 inglés, 0.016 último, 0.013 siguientes
6	0.147 nombre, 0.069 signo, 0.052 número, 0.049 letra, 0.037 voz, 0.031 lista, 0.029 artículo, 0.028 información, 0.027 cantidad, 0.026 gramatical

Cuadro 4.8: LDA: tópicos generados

Basándonos en estos resultados, puede interpretarse que los tópicos estarían definidos por aquellos términos relacionados con mayúsculas y minúsculas, números, puntuación, plurales y singulares, tiempo, enunciados, COVID y nombres, pues las palabras clave que aluden a estos tópicos son las que presentan mayor relevancia. Especialmente aquellos que se refieren a puntuación: «puntos», «lugar», «coma», «después», entre otros, destacan.

Centrándonos en el último tópico en el cuadro 4.8 y resaltado en el cuadro 4.9, podemos ver que está conformado por una serie de términos acompañados por las cifras que reflejan el peso de cada palabra clave al tópico en cuestión. Atendiendo a estas palabras clave, podemos determinar a qué tópico pertenecen. En este caso, el tópico 7 estaría asociado al tópico que trata sobre «nombres» y que se nombrará más adelante dentro del tópico «Ortografía».

6	0.147 nombre, 0.069 signo, 0.052 número, 0.049 letra, 0.037 voz, 0.031 lista, 0.029 artículo, 0.028 información, 0.027 cantidad, 0.026 gramatical

Cuadro 4.9: LDA: Tópico 6

Con el mismo razonamiento, y como ocurría en LSA, se interpretan las palabras claves del resto de tópicos para concretar a qué hace referencia cada uno de ellos:

Tópico 0: mayúsculas y minúsculas;

Tópico 1: puntuación, plurales y singulares;

Tópico 2: verbos;

Tópico 3: puntuación;

Tópico 4: enunciados y COVID;

Tópico 5: tiempos verbales y verbos;

Tópico 6: nombres.

Una vez realizada la primera interpretación, para la lectura de los resultados obtenidos al ejecutar el algoritmo con el conjunto de la semilla 2 se muestra el gráfico generado por `pyLDAvis`, favoreciendo la interpretación final y la denominación de los tópicos.

Para la correcta comprensión e interpretación de la figura 4.4, cabe destacar que el orden original de los tópicos ofrecido por el algoritmo se ve alterado en la visualización. Por ello, su correspondencia con los tópicos originales (véase lista 4.3.1) se expresará en la interpretación de los resultados visualizados en el gráfico `pyLDAvis`.

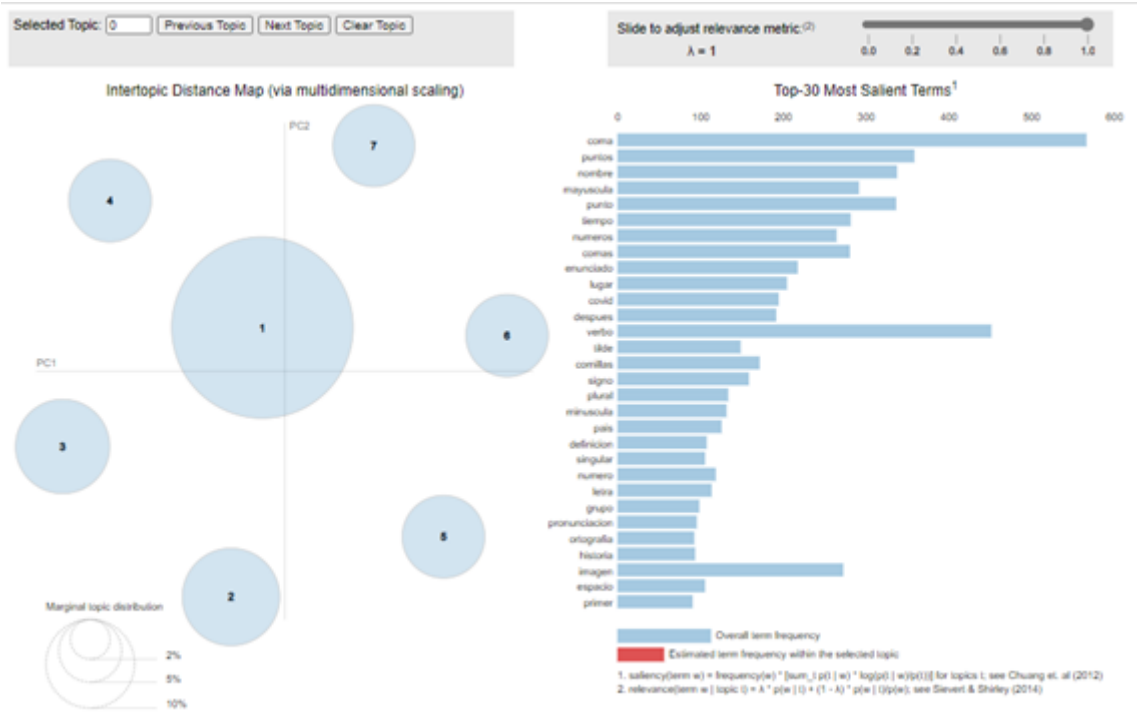

Figura 4.4: Gráficos (`pyLDAvis`) generados para estudio de tópicos LDA

En la parte izquierda se muestra el mapa de los intertópicos, es decir, la escala de los siete tópicos relevantes separados según sus términos dominantes. Como puede observarse, aparecen bastante diferenciados excepto el 1 y el 6 que se presentan con rasgos intermedios.

El gráfico muestra en la parte derecha en color azul la frecuencia general de los términos en el conjunto de tópicos obtenido del corpus procesado por el algoritmo. En este caso los términos de mayor relevancia son los relativos a consultas sobre puntuación, mayúsculas y minúsculas, nombres, tildes, tiempo y verbos, enunciados y COVID, como ya se obtenía en los tópicos ofrecidos directamente por el algoritmo.

Dada la interactividad que presentan los gráficos de `pyLDAvis`, al situarnos sobre uno de los tópicos, se muestra en la parte derecha en color rojo los treinta términos más relevantes en cada tópico, que determinan qué tuits podrían asignarse a cada uno de los siete tópicos según la relevancia interna que presenten.

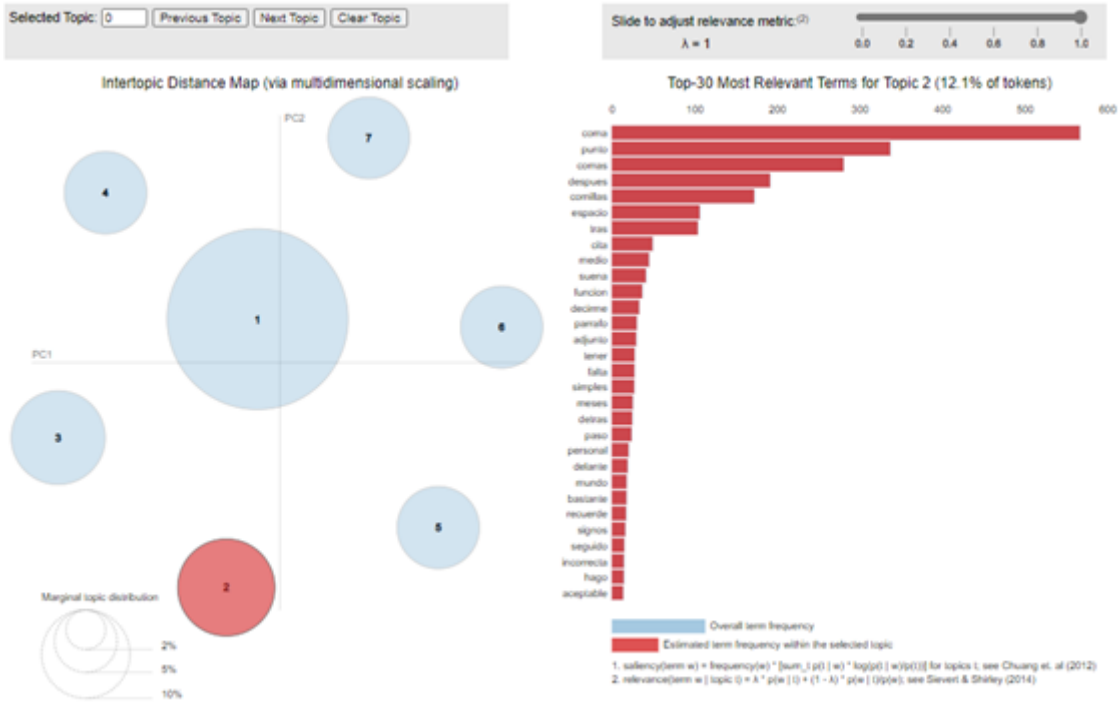

Figura 4.5: Resultados burbuja 2 en gráfico generado por `pyLDAvis`

Según la frecuencia de términos estimados dentro de la burbuja seleccionada «coma» es el término más relevante y «aceptable» el que menos. De modo que aquellos tuits cuyo término de mayor relevancia para el total esté relacionado con puntuación («coma», «punto», «después»), pertenecerá a este tópico.

Así, para determinar de manera más concreta la interpretación de los resultados, acudimos a la visualización de los resultados que ofrece `pyLDAvis` que puede consultarse en la página web diseñada para nuestra investigación en el apartado de código LDA (`https://josemh301.github.io/extrae/index.html`). A partir de esta visualización se concluye que:

El tópico 1, representado por la burbuja más grande y, por tanto, predominante, está formado por un conjunto de palabras clave entre las que destaca de manera notable «verbo», pero que al estar entremezclado sería el tópico que más términos acepte y, por tanto, al que más tuits se asignarían. En el tópico 2 destacan «coma», «punto», «comas» y «después». Respecto al tópico 3, «nombre» es la palabra clave más relevante con bastante

notabilidad. «Mayúscula», «números» y «minúscula» determinan el tópico 4. Por otro lado, «tiempo» es el término más predominante en el tópico 5. En el tópico 6 destacan nuevamente términos relacionados con puntuación: «puntos» y «lugar», aunque también presenta rasgos de mezcla. Y, finalmente, el tópico 7 está conformado por una relación de términos relevantes entre los que destacan de manera mayoritaria «enunciado», «COVID» y «tilde» (véase figura 4.5).

De acuerdo con esto, la correspondencia de los tópicos obtenidos obtenidos por LDA con su posterior visualización en `pyLDAvis` donde son representados en burbujas es la siguiente:

Tópico 0: mayúsculas y minúsculas - burbuja 4;

Tópico 1: puntuación, plurales y singulares - burbuja 6;

Tópico 2: verbos - burbuja 5;

Tópico 3: puntuación - burbuja 2;

Tópico 4: enunciados y COVID - burbuja 7;

Tópico 5: tiempos verbales y verbos - burbuja 1;

Tópico 6: nombres - burbuja 3.

Esta relación de palabras clave y sus frecuencias asociadas a los tópicos procesados, permite denominar como tópicos en el conjunto de documentos trabajado por LDA los siguientes: «Puntuación», «Verbos y tiempos verbales», «Ortografía», «Enunciados» y «COVID». Entre ellos, las burbujas se distribuyen estando claramente definidas la burbuja 2 asignada a «Puntuación», las burbujas 3 y 4 a «Ortografía», la burbuja 5 asignada a «Verbos», y la 7 a «Enunciados», «COVID» y «Ortografía». Siendo la 1 y la 6 las que están más mezcladas, pero que igualmente pueden asociarse con los tópicos «Verbo y tiempos verbales» y «Puntuación», respectivamente. Para visualizar todas las burbujas y su relación de términos correspondiente véase nuevamente el apartado de código dedicado a LDA disponible en exTRAE: `https://josemh301.github.io/extrae/index.html`

4.3.2. Predicción de tópicos para nuevos tuits

Una vez realizada la lectura de los resultados mostrada en el gráfico, la explicación práctica de cómo se realiza la asignación de un tuit nuevo a un tópico es más sencilla. En este apartado veremos cómo se comporta LDA aplicado a los mismos dos ejemplos vistos en LSA para la clasificación de los dos tuits:

Ejemplo 1

Repitiendo el ejemplo ofrecido en el apartado de LSA Ejemplo 1, el primero de los documentos a clasificar es el siguiente:

> #RAEconsultas #dudaRAE En la frase La tortilla, con cebolla está más rica, ¿puede ir esa coma tras el sujeto?, ¿sería una posible excepción de la coma criminal? Porque hay un matiz de diferencia con La tortilla con cebolla está más rica. ¿Cómo lo marco si no es con la coma?

El algoritmo lo procesa y obtiene los datos sobre la probabilidad de que el tuit en cuestión pertenezca a cada uno de los tópicos basándose en las palabras consideradas para el análisis. Es decir, las resultantes después del preprocesado. Así, el modelo da como resultado la siguiente lista de términos acompañada de las probabilidades que determinan la asignación del tuit a los tópicos ya dados:

```
['tortilla', 'cebolla', 'rica', 'coma', 'tras', 'sujeto',
excepción', 'coma', 'criminal', 'matiz', 'tortilla', 'cebolla',
'rica', 'marco', 'coma'],

[(0, 0.04605773), (1, 0.049313694), (2, 0.047454383), (3, 0.4234142),
(4, 0.046283133), (5, 0.32466805), (6, 0.062808834)]
```

De modo que el tópico con mayor probabilidad es el 3 (véase cuadro 4.8) con una relevancia del 42 % aproximadamente. Es decir, el tópico «Puntuación», cuyo término más relevante es «coma». Por consiguiente, este tuit se asignaría al tópico 3 obtenido por LDA, el cual corresponde con la burbuja 2 en el gráfico previamente mostrado (véase figura 4.5), tópico cuyo término más relevante es precisamente «coma» que, a su vez, representa el término más relevante en el corpus total, como se ha expresado en la lectura del gráfico .

Este proceso de asignación puede realizarse con cada nuevo documento/tuit incluido en el conjunto de documentos y puede comprobarse manualmente con los datos aportados en el código y aplicando el razonamiento anterior. Cada tuit presenta una relación de probabilidad con los tópicos según los términos que lo componen y de acuerdo con ese peso se van sumando a la frecuencia de términos total que determina los términos de frecuencia generales comunes a los siete tópicos y, a su vez, la frecuencia de términos en cada uno de los tópicos independientemente.

Ejemplo 2

Una vez hemos comprobado cómo LDA clasifica un tuit partiendo de un modelo generado previamente, podemos aplicarlo a tantos documentos como deseemos. Volvemos a usar el mismo tuit utilizado en el Ejemplo 2 de LSA.

> #raeconsultas #dudaRAE @RAEinforma Una pregunta, en la frase 'No importa cuanta cantidad de patata echemos, siempre y cuando no sea ninguna', cuál es el significado. Siempre y cuando sea alguna, o siempre y cuando sea ninguna.

El algoritmo obtiene el siguiente resultado:

```
(['importa', 'cuanta', 'cantidad', 'patata', 'echemos', 'siempre',
'ninguna', 'significado', 'siempre', 'siempre', 'ninguna'],

[(0, 0.03910909), (1, 0.041873846), (2, 0.040295057), (3, 0.05776168),
(4, 0.11474053), (5, 0.57744795), (6, 0.12877186)]
```

Como puede apreciarse con la primera lectura del tuit, los términos que lo componen no resultan fáciles de clasificar respecto a los tópicos dados. Sin embargo, sabemos que la

consulta es sobre el significado de un enunciado. Los datos de asignación asocian el tuit con el tópico 5 obtenido por LDA que, si nos fijamos en su correspondencia en el gráfico `pyLDAVis` 4.4 (burbuja 1), es el tópico con mayor mezcla y, por tanto, con menor precisión en la relación de sus términos y, por consiguiente, también en el proceso de asociarse con los documentos. Esta asignación al tópico 5-burbuja 1 se hace a partir del término «siempre» clasificado en el tópico 5 de LDA con un peso de `0.027` (véase cuadro 4.8).

Sin embargo, la imprecisión evidente que resulta en la asignación de este tuit en cuestión a uno de los tópicos no significa que LDA no sea capaz de predecir a cuál de los tópicos dados pertenece. El proceso concluye que este ejemplo pertenece al tópico más general demostrando una capacidad de clasificación óptima, pues la composición del tuit ya anunciaba una difícil clasificación.

Estos ejemplos ilustran que este proceso almacenado en una función particular en el código permite que cada nuevo tuit se procese y que se pueda realizar su asignación a un tópico. Particularmente, hay casos en los que los tuits no contienen una palabra clave que permita calcular la probabilidad para su asignación de manera sencilla. No obstante, eso no indica el fracaso del proceso, sino que quizás habría que entrenar el algoritmo indicando un nuevo número de tópicos hasta hallar el procedimiento que mejor funcionase para un determinado tuit. Así, el modelo modificado podría clasificar un nuevo tuit de manera favorable. Para observar otros ejemplos de esta asignación de nuevos tuits como los previamente tratados puede consultarse el código disponible en el repositorio de Google Colab, cuyo enlace se encuentra al comienzo del capítulo 4.

Capítulo 5

Trabajos afines

Llegados a este punto de la memoria, exponemos cinco trabajos relacionados con el que nosotros presentamos en aras de ofrecer una perspectiva más amplia de las aplicaciones que tiene el *topic modeling*.

Estos trabajos ponen a prueba algoritmos con fines distintos al objetivo de la presente investigación. De este modo, contribuyen a nuestra premisa sobre la relevancia de este tipo de análisis del lenguaje natural para mejorar y facilitar ciertas tareas realizadas por el hombre ante la inmensa cantidad de datos digitales con la que contamos en la actualidad.

5.1. *Topic modeling* más allá de la lengua

La aplicación de modelado de tópicos a corpus formados por grandes cantidades de datos de Twitter ha dado lugar a un variado número de estudios de naturaleza o problemática similar. Las publicaciones se dividen entre las que se basan en publicaciones de la red social mencionada para investigaciones concretas y los estudios que emplean directamente métodos de modelado de tópicos a conjuntos de datos formados por tuits.

Algunos de los más destacados se nombran a continuación dados dos motivos principales: por su relevancia en el panorama del modelado de tópicos y por la aportación de una perspectiva, afín pero diferente, a la investigación presentada en este trabajo.

5.1.1. En el análisis de reseñas turísticas

«Analyzing user reviews in tourism with topic models», un estudio que fue publicado por Marco Rosetti, Fabio Stella y Markus Zanker en 2015. Se desarrolló con el apoyo financiero de la Unión Europea (UE), el Fondo Europeo de Desarrollo Regional (FEDER), el Gobierno Federal austriaco y el Estado de Carintia en el programa Interreg IV Italien-Österreich (proyecto O-STAR).

Analizaron un conjunto de documentos formados por contenidos generados por usuarios sobre informaciones relacionadas con la toma de decisiones de los turistas y la gestión en el ámbito del turismo electrónico: reseñas y comentarios, entre otros. Por lo tanto, utilizaron modelado de tópicos con el fin de proporcionar una aplicación en el dominio del turismo.

El modelo se construyó procesando el conjunto de textos para identificar y asignar ciertos temas que proporcionaran apoyo a la toma de decisiones y a las recomendaciones a los turistas en línea, así como para construir una base de información sólida para futuros análisis.

Su contribución se basó en generar modelos que consiguieran unos resultados relevantes y útiles a partir del procesamiento de datos de opiniones publicadas por los usuarios sobre restaurantes y hoteles. En este sentido, estructuraron su investigación en torno a tres posibles escenarios de aplicación: predicción y recomendación según las evaluaciones de los usuarios, análisis e interpretación de temas concretos para personalizar el contenido que se muestra al usuario y sugerencia de calificaciones al usuario según comportamientos anteriores. Todo ello a partir de dos modelos propuestos: *Topic-Criteria* y *Topic-Sentiment* (Rosetti, Stella y Zanker, 2015), dejando la aplicación de LDA como opción para futuros trabajos.

5.1.2. En la recuperación de documentos clínicos

«Synonym, Topic Model and Predicate-Based Query Expansion for Retrieving Clinical Documents», un estudio publicado en 2012 por Qing T. Zeng, Doug Redd, Thomas Rindflesch y Jonathan Nebeker se ocupó de desarrollar y probar tres métodos de expansión de consultas para la recuperación de documentos clínicos. Buscaban solventar las dificultades que se plantean al realizar búsquedas de documentos relevantes en grandes cantidades de datos clínicos.

Para ello, utilizaron tres métodos. En primer lugar, implementaron una estrategia de expansión de sinónimos que usaba vocabularios concretos seleccionados. En segundo lugar, entrenaron un modelo de tópicos en un conjunto de documentos clínicos para posteriormente identificar términos relacionados para le expansión de consultas y, en tercer lugar, obtuvieron los términos relacionados de una base de datos de predicados derivada de los resúmenos de Medline para la expansión de consultas.

Los autores validaron el éxito de los tres métodos probados en un sistema de notas clínicas, siendo el método basado en modelo de tópicos el que obtuvo mejores resultados. Para identificar los términos relacionados mediante el modelado de temas, aplicaron el método a cien mil documentos clínicos, de los que seleccionaron mil documentos de cada uno de los cien tipos de documentos más frecuentes en la iniciativa de mejora del acceso de los investigadores a los datos llamada VINCI (Nebeker, Zeng-Treitlet, Redd y Rindflesch 1052).

Posteriormente y una vez identificados los documentos frecuentes, emplearon el programa MALLET (1052), un *software* de modelado de temas de implementación rápida y escalable, para identificar mil temas.

No todas las palabras identificadas como palabras del tema estaban clínicamente relacionadas, pero la mayoría tenían conexiones semánticas entre sí. Para mejorar el resultado continuaron con una filtración de doscientas palabras principales de los términos relacionados para lo cual se sirvieron de un *tokenizador* que facilitó la posterior conclusión de qué temas contenían términos coincidentes, qué términos existían como vocabulario de cada tema coincidente y cuál era el término con mayor relevancia en cada caso.

5.1.3. En la extracción de temas de salud pública

Comprendido también en el área de la salud, pero sirviéndose directamente de datos recopilados de Twitter el estudio «A Model for Mining Public Health Topics from Twitter», llevado a cabo por Michael J. Paul y Mark Dredze publicado en 2011 se basó en la aplicación del modelo de detección de tópicos *Ailment Topic Aspect Model* (ATAM) para descubrir aspectos de las enfermedades a partir de un conjunto de documentos formado por tuits que mencionaban síntomas, tratamientos y demás cuestiones relacionadas con las enfermedades. Entrenando también el modelo LDA para realizar comparaciones entre los resultados obtenidos.

Tomaron una colección de un millón seiscientos mil tuits recopilados de discusiones en Twitter sobre la salud que el algoritmo procesó, resultando como los tópicos más relevantes infecciones, gripe, obesidad, alergias y otras enfermedades más conocidas y extendidas entre la población.

El comportamiento del modelo utilizado fue óptimo al coincidir los resultados obtenidos con los que proporcionaba la búsqueda realizada con Google Flu Trends, el servicio web de Google que proporcionaba estimaciones sobre la actividad de la gripe a partir de búsquedas de Google entre los usuarios de más de veinticinco países.

Presentaban entonces un nuevo método para extraer información sobre salud pública general a través de datos de Twitter descubriendo síntomas y asociaciones de dichos síntomas con los tratamientos y proporcionando un número de dolencias más coherente que estudios anteriores, con más detalles y con seguimiento de las tasas de enfermedad.

De modo que los resultados obtenidos brindaban informaciones del tipo: los términos alergia, nariz, ojos, alergias, alérgico se asocia con los términos entendidos como síntomas tos, secreción, nasal y con los términos de tratamiento gotas, medicina, Claritin, entre otros (Paul y Dredze 5).

5.1.4. En el comentario de eventos

En el estudio publicado en 2012 por Yuheng Hu, Ajita John, Fei Wang y Subbarao Kambhampati llamado «Joint Topic Modeling for Aligning Events and their Twitter Feedback» consideraron datos recopilados en Twitter para analizar tendencias, comentarios sobre noticias u opiniones en general de acontecimientos como la Super Bowl o los debates políticos, entre otros. Intentaron demostrar que esta red social se ha convertido en el lugar de referencia para compartir este tipo de perspectivas.

Un evento conlleva un gran número de tuits y la cuestión a resolver es la extracción de los temas que surgen alrededor de las publicaciones sobre ese evento en cuestión. Además de ese objetivo, su estudio pretende demostrar que para entender esos temas se debía segmentar el evento con arreglo a esos tuits.

Para llevar a cabo su tarea desarrollaron un modelo basado en LDA para extraer los temas y segmentar el evento de manera conjunta. Como resultado del entrenamiento del modelo elegido, el comportamiento de LDA supuso una mejora significativa en el procesamiento de dos conjuntos de datos de tuits asociados con dos eventos respecto a otros modelos de referencia que habían sido entrenados para causas relacionadas.

5.1.5. En la identificación de género

En el estudio llamado «Yoga-Veganism: Correlation Mining of Twitter Health» publicado por Tunazzima Islam en 2019 se llevó a cabo un análisis de datos de Twitter con el fin de encontrar temas dominantes. Pretendían responder varias cuestiones sobre salud mental, dietas, práctica de yoga y vegetarianismo, entre otros. Para ello, se sirvieron de Twitter como entorno donde millones de usuarios comparten su estilo de vida y, por tanto, precisaban de modelado de tópicos.

El modelo que construyeron se basó en tres algoritmos diferentes: LDA, LSA y *Non-negative Matrix Factorization* (NMF) todos enfocados en inferir los temas del conjunto de tuits. Entrenaron el comportamiento de los tres mencionados algoritmos en la asignación de temas, en la aplicación del modelo en nuevos tuits y en las visualizaciones ofrecidas por cada uno de ellos (Tunazzima 1).

Sus conclusiones llevaron al descubrimiento de una relación entre los temas «Yoga» y «Veganismo» en todos los modelos (7). Para llegar a ellas se siguieron los pasos pertinentes de recopilación de datos, preprocesado, formación del conjunto de datos precisada en cada algoritmo y la obtención del algoritmo útil para su investigación.

Por otro lado, el entrenamiento de tres algoritmos distintos para la misma tarea aportó informaciones sobre el comportamiento de estos muy relevantes para estudios relacionados. En la misma línea, el resto de los estudios aquí expuestos, ya sean de mayor o menor afinidad con la investigación que se lleva a cabo en este trabajo, delimitan complicaciones, ventajas y acercamientos relevantes para el estudio del modelado de tópicos y de sus algoritmos o técnicas implicadas. Aportando, por consiguiente, avances útiles para el contexto en el que se enmarca nuestra investigación.

Capítulo 6

Conclusiones

En esta última sección del trabajo expondremos nuestras conclusiones sobre los distintos recorridos en el proyecto. Prestaremos atención a la necesidad por parte de las instituciones de disponer de herramientas informáticas que le faciliten su labor. Posteriormente, centrándonos en el grueso de nuestro trabajo, compararemos los resultados de los modelos LDA y LSA con la finalidad de ver si sus resultados se asemejan y son satisfactorios, habiendo logrado en tal caso el objetivo primero de este trabajo de fin de máster.

6.1. En el volumen de datos manejados por instituciones

A lo largo de la redacción del «Estado de la cuestión» buscamos poner de manifiesto la realidad actual no solo de la Real Academia Española, sino de las instituciones que deben hacer frente al manejo masivo de datos. Las instituciones, especialmente las que poseen un carácter histórico y están asentadas en la tradición, deben buscar su espacio en el mundo digital. De no ser así, el quedarse atrás podría suponer a medio plazo su disolución, o quizás peor, la pérdida de confianza y respeto por parte del gran público. Con esta motivación, buscan adaptarse a los nuevos tiempos: este trabajo es una muestra más de ello. Atendiendo a la ingente cantidad de tuits recibidos por la Real Academia Española, la posesión de una herramienta de este calibre puede resultar clave a la hora de enfocar su discurso en la red social Twitter, donde goza de un prestigio y repercusión que seguro desea mantener. En los últimos meses, nuevo vocabulario ha llegado a nuestras vidas, y el medio digital se ha convertido en una herramienta clave para el avance social y empresarial y esto se refleja en unas mayores inquietudes lingüísticas de las que la institución que limpia, fija y da esplendor al español debe estar al tanto.

Por consiguiente, el objetivo teórico que nos planteábamos fue alcanzado en aras de satisfacer la necesidad de conocer el contexto referido anteriormente para poder contextualizar de manera óptima nuestro trabajo.

6.2. En la aplicación del modelo LSA

Latent Semantic Allocation es uno de los grandes métodos de *topic modeling*, razón por la que de la misma manera que LDA ha sido empleado. Tal y como desglosamos en

su apartado teórico, no es LSA un método óptimo para el análisis de tuits según varios autores, autores que aconsejan la presencia de un mismo *hashtag* en todos los documentos que conformaran el corpus a analizar. Es quizás esta característica la que nos ha llevado a obtener resultados igualmente satisfactorios en la aplicación de este modelo. La extracción de un total de seis tópicos de los que pudimos comprobar que encajaban bastante bien con los propuestos por LDA, dan buena fe de ello. Aún mejor resulta la aplicación del modelo a los nuevos tuits que no formaban parte del corpus original. LSA no tiene problema en identificar tópicos claros, como los que se refieren a la posición de palabras respecto a signos de puntuación. Cuando el tuit es más ambiguo, careciendo de un tópico presente en modelo, lo asigna de manera prudente dentro del tópico comodín (véase cuadro 4.2). El resultado final es alcanza uno objetivos que consideramos satisfactorios.

6.3. En la aplicación del modelo LDA

El modelo Latent Dirichlet Allocation es otro de los métodos más usados a la hora de detectar tópicos de manera automática. Si bien vimos en su apartado teórico que era más eficaz lidiando con textos de gran tamaño como por ejemplo libros, una vez hemos obtenido los resultados de su aplicación a tuits hemos de indicar que los resultados han sido satisfactorios. El modelo muestra solvencia a la hora de extraer un total de siete tópicos que ilustra de una manera muy visual, dejando claro la separación entre los mismos y los *tokens* que los gobiernan. El resultado final es la clasificación correcta de tuits añadidos posteriormente, tuits que no pertenecían al corpus original con el que se formó el modelo.

De acuerdo con esto, los objetivos prácticos planteados respecto al entrenamiento de ambos modelos se lograron obteniendo resultados óptimos y productivos para el objetivo transversal propuesto.

6.4. En la aplicación de ambos modelos

Creemos haber obtenido dos métodos completamente válidos para la empresa que se nos encargó. Ambos modelos son capaces de identificar el tópico de tuits sobre consultas lingüísticas a la Real Academia Española por su relación de relevancia y frecuencia en los términos que conforman el diccionario que los dos algoritmos procesan. De esta manera hemos podido contestar a las cuestiones que nos planteábamos inicialmente:

- ¿Cuáles son las dudas más frecuentes?
- ¿Es posible agrupar las dudas de alguna manera para ofrecer respuestas automáticas?

En relación a la primera cuestión, como se muestra en los apartados dedicados a cada código y, a continuación, en la comparativa de los resultados, hemos podido resolverla enmarcada en el periodo de tiempo en el que recopilamos los tuits. Y, con un conjunto de documentos mayor o con más tuits recopilados en distintos periodos de tiempo, concluimos que los resultados podrían seguir siendo óptimos permitiendo ampliar la respuesta a la cuestión. Por su parte, respecto a la segunda cuestión, nuestra conclusión es que esto es parte del trabajo futuro que se abre al terminar esta investigación del que creemos que sería posible diseñar de manera que obtuviéramos resultados favorables. En conclusión,

aunque lejos de la perfección, estimamos que nuestra herramienta constituye un punto de partida más que válido a la hora de crear un programa que ayude a la Real Academia Española a clasificar y cuantificar las consultas recibidas a través de su perfil en la red social Twitter. Por lo que estimamos esta investigación como un punto de partida válido para trabajos futuros que estudien aproximaciones comunes a la que aquí se desarrollan.

Ahora, justificamos nuestra conclusión al objetivo transversal con la exposición de la comparación de los resultados obtenidos en ambos algoritmos.

6.5. Comparativa de los resultados de ambas técnicas

6.5.1. Comparación de modelos generados

En la implementación de los modelos LSA y LDA hemos obtenido una serie de resultados derivados del entrenamiento de sendos modelos. En la primera muestra de resultados, correspondiente a la generación de los modelos, obtenemos los temas en los que dividen LSA y LDA el corpus. La primera diferencia reside en el número de temas elegidos por los investigadores en ambos casos, como ya hemos indicado. LSA, como muestra la figura 4.1, propone una división en seis temas. Por el contrario, para LDA, se utilizó un método manual a la hora de seleccionar el número de temas. Aunque también basado en la coherencia, se realizaron diversas pruebas con el algoritmo hasta conseguir el número de temas que resultaba más acertado. Decidimos pues, ajustar tal número a siete (véase tabla 4.8).

Teniendo en cuenta la diferencia en el número de tópicos entre modelos, a la que se suma la diferencia entre las bases matemáticas que utilizan cada uno de los algoritmos, es previsible encontrar diferencias entre los temas y su distribución. Desglosamos las diferencias y similitudes entre LSA y LDA.

1. En LSA, el tópico 0 (véase cuadro 4.2) trata el tema de la puntuación, en concreto, de la posición de signos ortográficos teniendo en cuenta la categoría gramatical de las palabras que los preceden. En LDA existen hasta dos tópicos que cubren este aspecto lingüístico: tópicos 1 y 3 (véase cuadro 4.8). En los resultados ofrecidos por los dos modelos se observan ciertas palabras comunes: «coma», «punto», «puntos», «tras» y «después».

2. El tópico 1 de LSA (véase cuadro 4.3) se aleja del tema normativo de la lengua. Dada la situación actual, era hasta cierto punto comprensible un tema que tratara la cuestión del COVID. Asimismo, LDA cuenta con un tópico de igual temática en el tópico 4 (véase cuadro 4.8). En este caso los modelos comparten «COVID», «virus» y «enfermedad» (véase una relación de todos los *tokens* de LDA en el gráfico `pyLDAvis` 4.4 disponible en la página web del proyecto: `https://josemh301.github.io/extrae/index.html`, concretamente en el apartado de código dedicado a LDA).

3. LSA cuenta con un tema dedicado a los verbos. Se corresponde con el tópico 2 (véase cuadro 4.4). Para esta cuestión lingüística LDA contempla nuevamente dos tópicos: tópicos 2 y 5 (véase cuadro 4.8). Ambas comparten palabras como «verbo», ocupando tanto en el tópico 2 de LSA como en el 5 de LDA el mayor peso del tema. Comparten otros *tokens* de menor relevancia como por ejemplo «pasado».

4. El cuarto de los tópicos (tópico 3) ofrecidos por LSA, aunque ya indicamos que

de manera poco asertiva, es la acentuación y tildes (véase cuadro 4.5). LDA no destina ningún tema en específico a esta cuestión lingüística, pero aparece «tilde» entre los *tokens* del tópico 4 (véase cuadro 4.8). Así, bien se comprueba que ambos modelos no dan relevancia al tema de la acentuación y tildado de palabras, bien esta cuestión no constituye una de las dudas prominentes entre los usuarios de Twitter que preguntan a la Real Academia Española.

5. LSA propone como quinto tópico (tópico 4) las dudas relacionadas con el empleo de mayúsculas y minúsculas (véase cuadro 4.6). Asimismo, LDA reserva el tópico 0 para esta cuestión (véase cuadro 4.8). Entre los dos modelos comparten los siguientes *tokens*: «mayúscula», «minúscula» e «inicial» (véase nuevamente gráfico `pyLDAvis` 4.4).

6. Ante la imposibilidad de dividir de manera certera todo el corpus en tópicos bien diferenciados e inconexos, tanto LSA como LDA reservan uno de sus tópicos como un comodín. De funcionar correctamente, en él enmarcará un tuit que sea incapaz de asignar a cualquiera de los otros tópicos. En el caso de LSA, se trata del tópico 5 (véase cuadro 4.7), mismo índice que tiene el tópico en LDA (véase cuadro 4.8).

Tras haber relacionado los tópicos extraídos por LSA y LDA, encontramos que a pesar de la diferencia en el número de tópicos decidido previamente, existe una patente correspondencia entre los resultados obtenidos. En la siguiente tabla mostramos la relación obtenida:

LSA	**LDA**
Tópico 0	Tópico 1 Tópico 3
Tópico 1	Tópico 4
Tópico 2	Tópico 2 Tópico 5
Tópico 3	Tópico 4 (parcialmente)
Tópico 4	Tópico 0
Tópico 5	Tópico 5

Cuadro 6.1: Relación entre tópicos de LSA y LDA

Quedaría sin correspondencia el tópico 7 de LDA dedicado a nombres, una cuestión que se denomina dentro de la temática de «Ortografía». No obstante, este desajuste entre ambos modelos era de esperar por el número de tópicos determinado para cada uno de ellos y no es una cuestión relevante que signifique un mal funcionamiento de ninguno de los modelos, así como tampoco del trabajo complementario que tiene lugar entre ambos.

6.5.2. Comparación de ejemplos para la comprobación de los modelos

Una vez hemos señalado las correspondencias entre los resultados de la ejecución de los modelos LSA y LDA, resta comprobar si estas similitudes se trasladan a la hora de clasificar nuevos tuits. A continuación veremos, en los mismo tuits, la diferencia o similitud de resultados:

Ejemplo 1

Recordamos el tuit:

> #RAEconsultas #dudaRAE En la frase La tortilla, con cebolla está más rica, ¿puede ir esa coma tras el sujeto?, ¿sería una posible excepción de la coma criminal? Porque hay un matiz de diferencia con La tortilla con cebolla está más rica. ¿Cómo lo marco si no es con la coma?

Como vimos en el apartado Ejemplo 1 de LSA (véase apartado 4.2.2), la clasificación otorgada era `(0, -0.2098027229940785)`, indicando que el tópico en el que encuadrar tal tuit era el tópico 0 (véase cuadro 4.2). Por lo tanto, según LSA, el tuit versa sobre la posición de cierto signo ortográfico respecto a una palabra con una categoría gramatical determinada. En el caso de LDA, en el Ejemplo 1 (véase 4.3.2), este tuit se asigna al tópico 3 (burbuja 2) (véase cuadro 4.8) que, de la misma manera que LSA, es el dedicado a cuestiones relativas a las normas de puntuación. Vemos pues, que existe consenso en la asignación del tuit en cuestión con los tópicos que incluyen cuestiones sobre el uso de las comas.

Ejemplo 2

Para el segundo ejemplo, recordamos cuál era el tuit a clasificar:

> #raeconsultas #dudaRAE @RAEinforma Una pregunta, en la frase 'No importa cuanta cantidad de patata echemos, siempre y cuando no sea ninguna', cuál es el significado. Siempre y cuando sea alguna, o siempre y cuando sea ninguna.

En el caso de LSA (véase apartado 4.2.2), el modelo no posee un tema en específico en el que encuadrar el tuit. Como podemos inferir de apartados anteriores, lo clasifica de manera no muy asertiva dentro del tópico comodín, cuyo índice es el quinto y último (véase cuadro 4.7). De la misma manera, LDA asigna este tuit a su propio tópico comodín (véase apartado 4.3.2), es decir, al que más mezcla de relevancia de términos presenta: el tópico 4 - burbuja 1 (véase cuadro 4.8). Se pone de manifiesto que el tuit carece bien de un tema suficientemente claro como para encajarlo en una de los temas precisos, bien que ninguno de los dos modelos ha considerado generar un tópico exclusivo para el tema que el tuit trata.

6.6. Conclusión final

A lo largo del presente trabajo múltiples han sido los estadios que nos han llevado a obtener unos resultados prometedores. Conocer cuál era el estado de la cuestión en el campo institucional y en específico, de su labor digital, abría las puertas al desarrollo de una herramienta que subsanara las carencias informáticas de tales organismos, concretamente de la Real Academia Española en Twitter. La decisión de qué herramienta del procesamiento del lenguaje natural seleccionar para el logro de unos objetivos que creemos cubiertos, hizo necesario vislumbrar cuál era el estado de la cuestión de esta

rama lingüística, prestando especial atención a la técnica *topic modeling* y sus dos modelos más recurrentes. Latent Semantic Allocation y Latent Dirichlet Allocation cubrían de manera teórica nuestras necesidades. Comprobar si ambas técnicas de modelado de tópicos podían cumplir las expectativas fue el detonante que nos llevó a programar basándonos en los dos modelos. Una vez el código hubo sido completado y posteriormente comprobado, podemos estimar que en general los resultados obtenidos tras los diversos estadios del proyecto, son satisfactorios. Y, además, podemos afirmar, como conclusión final al estudio desarrollado en estas páginas las siguientes cuestiones:

- LSA obtiene de manera más clara los distintos tópicos;
- LDA cuenta con una visualización de los tópicos y su relación de términos más útil y reveladora;
- LSA cuenta con un entrenamiento más automático;
- LDA cuenta con un entrenamiento más manual;
- LSA y LDA son algoritmos útiles para esta clasificación;
- LSA y LDA se complementan para el objetivo de este proyecto;

Índice de figuras

Índice de cuadros

Listings

Bibliografía

Alash, H. M. & Al-Sultany, G. A. (2020), 'Improve topic modeling algorithms based on twitter hashtags', *Journal of Physics: Conference Series* **1660**, 1–9.

Arjones, G. & Hammoe, L. (2018), 'Detección de tópicos utilizando el modelo lda', pp. 1–24.

Asignación de Dirichlet latente: intuición, matemáticas, implementación y visualización (n.d.), https://ichi.pro/es/asignacion-de-dirichlet-latente. Consultado: 2021-06-03.

Ballester, A. (2013), Análisis de la política de comunicación en twitter de las administraciones públicas en la comunidad valenciana, *in* A. Ballester Espinosa, ed., 'Gestión de la escasez participación, territorios y estado del bienestar: experiencias de gobernanza y gestión pública', GOGEP Complutense, Pozueldo de Alarcón, pp. 209–219.

Blanco-Hermida Sanz, E.-J. (2016), 'Algoritmos de clustering y aprendizaje automático aplicado a twitter', pp. 1–56.

Blau, P. (1964), *Exchange and power in social life*, 1 edn, Transaction Publishers.

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', *J. Mach. Learn. Res.* **3**, 993–1022.

Bonnin, J. E. (2014), Pensar en castellano en internet: discursos sobre la norma en los foros de wordreference.com, *in* E. Narvaja de Arnoux & S. Nothstein, eds, 'Temas de glotopolítica', 1 edn, Editorial Biblos, pp. 351–372.

Carrascosa, J. L. (1992), *Comunicacion: una comunicación eficaz para el éxito en los negocios*, 1 edn, CDN.

Castillo Esparcia, A. (2008), 'La comunicación empresarial en internet', *ICONO 14, Revista de comunicación y tecnologías emergentes* **6**(2), 1–18.

Chandía Sepúlveda, B. (2016), 'Aplicación y evaluación lda para asignación de tópicos en twitter', *Pontificia Universidad Católica de Valparaíso.* pp. 1–55.

Comenzando con spaCy para procesamiento de lenguaje natural (2018), `https://medium.com/datos-y-ciencia/comenzando-con-spacy-para-procesamiento-de-lenguaje-natural-e8cf24a18a5a`. Consultado: 2021-07-03.

Daneshvar, S. & Inkpen, D. (2018), Gender Identification in Twitter using N-grams and LSA: Notebook for PAN at CLEF 2018, *in* 'CEUR Workshop Proceedings', Vol. 2125.
URL: *http://ceur-ws.org/Vol-2125/paper_213.pdf*

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.

Dumais, S. T. (2004), 'Latent semantic analysis', *Annual review of information science and technology* **38**(1), 188–230.

ET-LDA: Joint Topic Modeling for Aligning Events and their Twitter Feedback (2012), *CoRR* **abs/1211.3089**. Withdrawn.
URL: *http://arxiv.org/abs/1211.3089*

Ferner, C., Havas, C., Birnbacher, E., Wegenkittl, S. & Resch, B. (2020), 'Automated seeded latent dirichlet allocation for social media based event detection and mapping', *Information* **11**(8).
URL: *https://www.mdpi.com/2078-2489/11/8/376*

Franco Romo, D. (2011), 'Hipermediaciones. elementos para una teoría de la comunicación digitral interactiva', *Mediaciones Sociales* (8), 167–170.

Fukuyama, F. (1995), *Trust: The social virtues and the creation of prosperity*, Vol. 99, Free Press New York.

Garcia, K. & Berton, L. (2021), 'Topic detection and sentiment analysis in twitter content related to covid-19 from brazil and the usa', *Applied Soft Computing Journal* **101**, 1–15.

González Fernández, A. (2017), 'Estudio de neologismos a través de big data en un corpus textual extraído de twitter', *ELUA. Estudios de Lingüística Universidad de Alicante* (31), 171–186.
URL: *http://dx.doi.org/10.14198/ELUA2017.31.09*

Guy, I. (2015), Social recommender systems, *in* F. Ricci, L. Rokach & B. Shapira, eds, 'Recommender Systems Handbook', 1 edn, Springer US, Boston, MA, pp. 511–543.

He, L., Jia, Y., Han, W. & Ding, Z. (2014), 'Mining user interest in microblogs with a user-topic model', *China Communications* **11**(8), 131–144.

Historia de Twitter: de un comienzo brillante a los rumores sobre su futuro incierto (n.d.), `https://marketing4ecommerce.net/historia-de-twitter/`. Consultado: 2021-06-20.

Introducción al topic modeling con Gensim (II): asignación de tópicos (n.d.), `https://elmundodelosdatos.com/topic-modeling-gensim-asignacion-topicos/`. Consultado: 2021-07-04.

Jacobucci, R., Ammerman, B. A. & Tyler Wilcox, K. (2021), 'The use of text-based responses to improve our understanding and prediction of suicide risk', *Suicide & life-threatening behavior* **51**(1), 55–64.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y. & Zhao, L. (2019), 'Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey', *Journal of Latex Class Files* **78**(11), 15169–15211.

Jiménez-Zafra, S. M., Martín-Valdivia, M. T., Martínez-Cámara, E. & Ureña-López, L. A. (2015), 'Studying the scope of negation for spanish sentiment analysis on twitter', *Journal of Latex Class Files* **14**(8), 1–12.

Kalepalli, Y., Tasneem, S., Phani Teja, P. D. & Manne, S. (2020), Effective comparison of lda with lsa for topic modelling, *in* '2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)', pp. 1245–1250.

Kwak, H., Lee, C., Hosung, P. & Sue, M. (2010), 'What is twitter, a social network or news media?', *WWW2010* pp. 591–600.

Landauer, T. K., Foltz, P. W. & Laham, D. (1998), 'An introduction to latent semantic analysis', *Discourse Processes* **25**(2-3), 259–284.

Lauria, D. & López García, M. (2009), 'Instrumentos lingüísticos académicos y norma estándar del español: la nueva política lingüística panhispánica', *Lexis* **33**(1), 49–89.

Lee, J., Kim, H. J. & Ahn, M. J. (2011), 'The willingness of e-government service adoption by business users: The role of offline service quality and trust in technology', *Government Information Quarterly* **28**(2), 222–230.

López Martín, E. (2012), 'Twitter como argumento, herramienta y soporte para la producción artística contemporánea', *Forma. Revista d'Humanitats* **6**, 33–47.

Modelado de temas con LDA: una explicación intuitiva (n.d.), `https://ichi.pro/es/modelado-de-temas-con-lda-una-explicacion-intuitiva-25961424702741`. Consultado: 2021-06-03.

Mottaghinia, Z., Feizi-Derakhshi, M.-R., Farzinvash, L. & Salehpour, P. (2020), 'A review of approaches for topic detection in twitter', *Journal of Experimental & Theoretical Artificial Intelligence* pp. 1–27.

MS Windows NT Kernel Description (n.d.), `https://docs.python.org/es/3/howto/urllib2.html`. Consultado: 2021-06-15.

Narvaja de Arnoux, E. & Nothstein, S., eds (2014), *Temas de glotopolítica: Integración regional sudamericana y panhispanismo*, 1 edn, Editorial Biblos.

Narvaja de Arnoux, Elvira y Del Valle, J. (2010), 'Ideologías lingüísticas y el español en contexto histórico', *CUNY Academic Works* **7**(1), 1–24.

Niklison, L. M. (2020), 'Lo que la rae no nombra no existe: una mirada glotopolítica sobre las respuestas de la rae al lenguaje inclusivo/no sexista.', *Cuadernos de la ALFAL* **1**(12), 13–32.

Nugroho, R., Paris, C., Nepal, S., Yang, J. & Zhao, W. (2020), 'A survey of recent methods on deriving topics from twitter: algorithm to evaluation', *Knowledge and Information Systems* **62**(7), 2485–2519.

Paul, M. & Dredze, M. (2012), 'A model for mining public health topics from twitter', **11**.

Python (2012), `https://www.python.org/`. Consultado: 2021-07-02.

Qian, X., Feng, H., Zhao, G. & Mei, T. (2014), 'Personalized recommendation combining user interest and social circle', *IEEE Transactions on Knowledge and Data Engineering* **26**(7), 1763–1777.

Qiang, J., Qian, Z., Li, Y., Yuan, Y. & Wu, X. (2019), 'Short text topic modeling techniques, applications, and performance: A survey', *Journal of Latex Class Files* **14**(8), 1–19.

Ricci, F., Rokach, L. & Shapira, B., eds (2015), *Recommender Systems Handbook*, 3 edn, Springer US, Boston, MA.

Risch, J. (2016), 'Detecting twitter topics using latent dirichlet allocation', pp. 0–48.

Rizzo, M. F. (2019), 'Discurso normativo de la rae en twitter', *Revista de Investigación Lingüística* **22**, 425–450.
URL: *https://revistas.um.es/ril/article/view/386881/278181*

Rossetti, M., Stella, F., Cao, L. & Zanker, M. (2015), *Analysing User Reviews in Tourism with Topic Models*, Vol. 16, pp. 47–58.

Salazar Puerta, S. & Prieto Dávila, P. R. (2015), 'Gestión y administración de la comunicación institucional en twitter', *Anuario Electrónico de Estudios en Comunicación Social "Disertaciones",* **8**(1), 11–26.

Scolari, C. (2013), *Hipermediaciones: Elementos para una teoría de la comunicación digital interactiva*, Cibercultura, Editorial Gedisa, Barcelona.

Shivam Bansal (2016), 'Beginners guide to topic modeling in python', *Analytics Vidhya* pp. 1–12.

Sievert, C. & Shirley, K. (2014), Ldavis: A method for visualizing and interpreting topics.

Slemp, K., Black, M. & Cortiana, G. (2020), 'Reactions to gender-inclusive language in spanish on twitter and youtube', *Proceedings of the 2020 annual conference of the Canadian Linguistic Association* pp. 1–11.

Steinskog, A. O., Therkelsen, J. F. & Gambäck, B. (2017), 'Twitter topic modeling by tweet aggregation', *Proceedings ofthe 21st Nordic Conference ofComputational Linguistics* pp. 77–86.

Szpektor, I., Maarek, Y. & Pelleg, D. (2013), *When Relevance is not Enough: Prmoting Diversity and Freshness in Personalized Question recomendation*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva.
URL: *http://dl.acm.org/citation.cfm?id=2488388*

Topic modeling: ¿qué, cómo, cuándo? (2016), `http://www.morethanbooks.eu/topic-modeling-introduccion/`. Consultado: 2021-06-13.

Warren, A. M., Sulaiman, A. & Jaafar, N. I. (2014), 'Social media effects on fostering online civic engagement and building citizen trust and trust in institutions', *Government Information Quarterly* **31**(2), 291–301.

Williams, T. & Betak, J. (2018), 'A comparison of lsa and lda for the analysis of railroad accident text', *Procedia Computer Science* **130**, 98–102. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050918303673*

Zeng-Treitler, Q., Redd, D., Rindflesch, T. & Nebeker, J. (2012), 'Synonym, topic model and predicate-based query expansion for retrieving clinical documents', *AMIA ... Annual Symposium proceedings. AMIA Symposium* **2012**, 1050–9.

Zhao, W. X., Jiang, J., Weng, J., He, J. & LIM, E. P. (2011), 'Comparing twitter and traditional media using topic models', pp. 1–14.