Universidad de la República
Facultad de Ingeniería

Identificación automática de tópicos para el Observatorio de Medios del Uruguay

Informe de Proyecto de Grado presentado por

Francisco Carballal, Juan Mauriz

como requisito de graduación de la carrera de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de la República

Supervisores

Guillermo Moncecchi
Juan José Prada

Montevideo, 24 de noviembre de 2022

Agradecimientos

Agradecemos profundamente a la Facultad de Ingeniería y a todo su equipo docente que durante estos años hicieron posible nuestra formación como profesionales. Agradecemos también a la FIC por la posibilidad de formar parte del OMU. Agradecemos a La Diaria por proporcionarnos el corpus de 20000 noticias con el cual realizamos experimentos. Por último queremos agradecer a Juan Jose Prada y Guillermo Moncecchi por ser los supervisores del proyecto y a Luis Chiruzzo y Aiala Rosá y Natalia Uval por conformar el tribunal.

Por otra parte, agradecemos a Sofía Machado, a Gisella Ottonelli, a la familia, por el apoyo constante y a todos nuestros compañeros con los que hemos trabajado hombro con hombro a lo largo de la carrera.

Resumen

Los medios de comunicación tienen un gran impacto sobre la determinación de los temas que la gente debate diariamente y cómo los interpreta. Es pertinente que se realicen investigaciones sistemáticas sobre la cobertura realizada por los medios de comunicación sobre diferentes temas y las prácticas discursivas utilizadas. Para cumplir este fin surge el Observatorio de Medios del Uruguay (OMU), como un proyecto llevado adelante por la Facultad de Información y Comunicación con apoyo de la Facultad de Ingeniería, ambas de la Universidad de la República. El presente trabajo se enmarca en el OMU, con el objetivo de aportar una solución automatizada a alguna de las tareas involucradas en el referido proyecto. En base a reuniones realizadas con su equipo, se decidió que el problema a resolver sea la detección automática de temas.

Dentro del Procesamiento de Lenguaje Natural, la detección automática de temas se denomina Modelado de Tópicos. Es un problema de aprendizaje automático no supervisado, en el que se debe determinar cuáles son los tópicos, en lugar de disponer de categorías predefinidas y limitarse a clasificar noticias. La metodología más utilizada para abordarlo es *Latent Dirichlet Allocation* (LDA). En este trabajo se utilizó una variación reciente, denominada *embedded topic modeling* (ETM), que enriquece LDA con el uso de *word embeddings*. Se implementó en *python* una aplicación web que permite entrenar y utilizar modelos de ETM. Se puede inferir los tópicos presentes en un corpus de noticias y luego clasificar automáticamente otras noticias que se ingresen desde la interfaz web.

Se evaluaron modelos de ETM utilizando un corpus de 20.000 noticias pertenecientes a La Diaria, mediante experimentos cualitativos y cuantitativos. Cualitativamente, los resultados son satisfactorios y se observan similitudes con lo reportado por los autores de la metodología, particularmente en la robustez frente a palabras que no agregan significado o contenido específico (como artículos, preposiciones y algunas palabras comunes). Cuantitativamente, utilizando métricas de desempeño se pudo determinar la cantidad óptima de tópicos para el corpus.

Palabras clave: Procesamiento de Lenguaje Natural, Aprendizaje Automático, modelado de tópicos, *Latent Dirichlet Allocation*, *word embeddings*, análisis de noticias.

Índice general

Capítulo 1

Introducción

Los medios masivos de comunicación tienen un gran impacto tanto en la definición de los temas que se vuelvan objeto de debate y preocupación a nivel de la opinión pública [1], como en la construcción de encuadres sobre acontecimientos, que afectan cómo las personas interpretan los hechos [2]. Considerando lo anterior, es pertinente que se realice investigación sistemática sobre los mecanismos de definición de agenda y las construcciones discursivas de los medios masivos de comunicación. A partir de esta inquietud surge el proyecto de crear un *Observatorio de Medios del Uruguay* (OMU), el cual es llevado adelante por la Facultad de Información y Comunicación (FIC) y cuenta con la colaboración de la Facultad de Ingeniería, ambas de la Universidad de la República (UdelaR).

Concretamente, los objetivos del observatorio incluyen[1]:

- Describir y analizar de forma sistemática y continua la cobertura que realizan los medios de comunicación uruguayos de temas sociales y políticos de actualidad, para contribuir a la comprensión pública de los mecanismos que utilizan para definir agenda y construir realidad.
- Identificar y caracterizar los discursos y representaciones que realizan los medios de comunicación de hechos de actualidad que pueden afectar los derechos humanos.
- Identificar, describir y analizar procesos de desinformación que se producen tanto a nivel de medios de comunicación tradicionales como de las redes sociales.

Para llevar a cabo estos objetivos se utiliza un enfoque multidisciplinario, incluyendo análisis cuantitativo y cualitativo.

Por otra parte, el OMU incluye una componente de difusión de resultados para el público general. Esto se materializa principalmente en un sitio web específico[2] y la presencia en redes sociales.

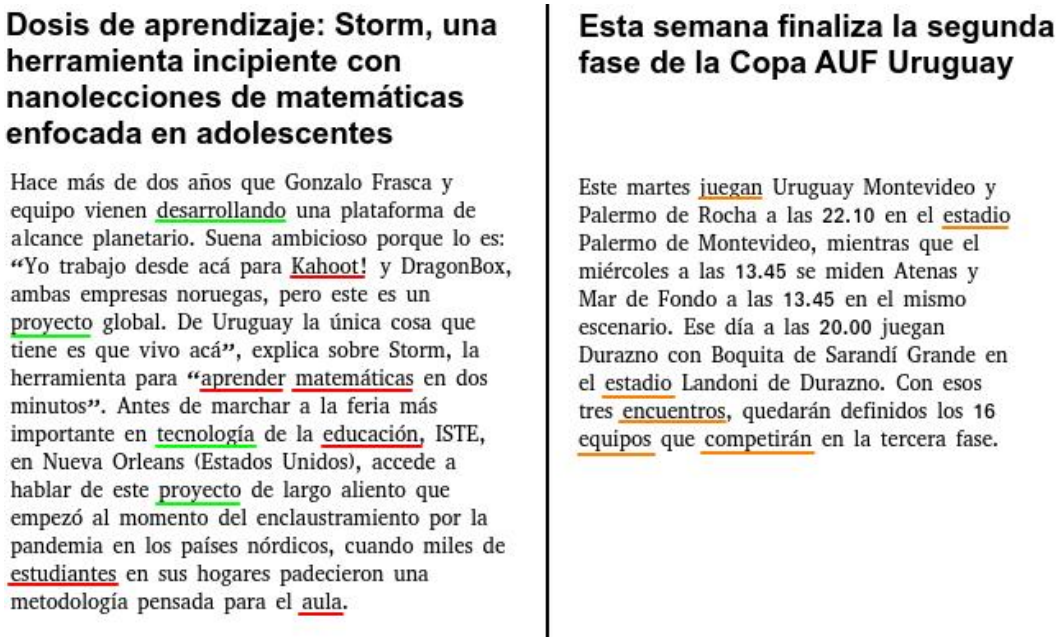
[1] Texto extraído de la postulación del proyecto OMU a la CSIC

[2] `https://omu.fic.edu.uy`

Además de consolidar los estudios de medios en Uruguay, el OMU promueve la investigación en la FIC y potencia sus vínculos académicos, tanto entre áreas internas como con otras facultades de la UdelaR. El presente trabajo se enmarca en el proyecto del OMU.

1.1. Problema y motivación

En este trabajo nos centraremos en la tarea de determinar temas presentes en noticias. Consideremos a modo de ejemplo las mostradas en la figura 1.1. En la noticia de la izquierda pueden observarse palabras subrayadas en verde asociadas con la investigación o la ciencia (desarrollo, proyecto, tecnología) y en rojo palabras relacionadas a la educación (aprender, estudiantes, aula), mientras que en la noticia de la derecha hay subrayadas en naranja palabras vinculadas al deporte (estadio, encuentros, equipos). En base a estas palabras, se podría deducir que la noticia de la izquierda tiene como principales temas la educación y la investigación, mientras que la de la derecha trata de deportes.



Dosis de aprendizaje: Storm, una herramienta incipiente con nanolecciones de matemáticas enfocada en adolescentes

Hace más de dos años que Gonzalo Frasca y equipo vienen desarrollando una plataforma de alcance planetario. Suena ambicioso porque lo es: "Yo trabajo desde acá para Kahoot! y DragonBox, ambas empresas noruegas, pero este es un proyecto global. De Uruguay la única cosa que tiene es que vivo acá", explica sobre Storm, la herramienta para "aprender matemáticas en dos minutos". Antes de marchar a la feria más importante en tecnología de la educación, ISTE, en Nueva Orleans (Estados Unidos), accede a hablar de este proyecto de largo aliento que empezó al momento del enclaustramiento por la pandemia en los países nórdicos, cuando miles de estudiantes en sus hogares padecieron una metodología pensada para el aula.

Esta semana finaliza la segunda fase de la Copa AUF Uruguay

Este martes juegan Uruguay Montevideo y Palermo de Rocha a las 22.10 en el estadio Palermo de Montevideo, mientras que el miércoles a las 13.45 se miden Atenas y Mar de Fondo a las 13.45 en el mismo escenario. Ese día a las 20.00 juegan Durazno con Boquita de Sarandí Grande en el estadio Landoni de Durazno. Con esos tres encuentros, quedarán definidos los 16 equipos que competirán en la tercera fase.

Figura 1.1: Fragmentos de notas periodísticas extraídos de La Diaria.

La identificación de temas en noticias puede ser realizada por personas, las cuales deben leer noticias una por una, detectar palabras asociadas a distintos temas y cuantificar los resultados. Sin embargo, este enfoque tiene desventajas. Si se desea contemplar muchas noticias, se necesita mucho tiempo de trabajo, lo cual puede hacer al proyecto inviable por tardanza o costos. Por otra parte, hay otras tareas no sistemáticas que requieren creatividad (como análisis de resultados, desarrollo de conclusiones y divulgación) que se verían limitadas si los recursos humanos debieran dedicarse a registrar temas de noticias

la mayoría del tiempo. Por lo tanto, surge el deseo de utilizar detección automática de temas. De esta forma:

- Se evita que personas deban dedicar tiempo a esa tarea, permitiendo que concentren sus esfuerzos en análisis más profundos y específicos.
- Es posible procesar mayor cantidad de información en menos tiempo, produciendo resultados más globales y rápidos.

La detección automática de temas es una tarea de Procesamiento de Lenguaje Natural (introducido en la sección 2.1), disciplina que permite automatizar tareas como la mencionada con rendimientos cada vez mejores, al ser un área activa de investigación.

1.2. Objetivos

En base a las reuniones que se realizaron con el equipo del OMU, se definió que este proyecto de grado abarque una aproximación al problema de modelado de tópicos (disciplina del Procesamiento del Lenguaje Natural que puede utilizarse para detectar temas, introducida en la sección 2.4) y el desarrollo de una aplicación web que permita acceder a lo anterior. Esto incluye las siguientes tareas:

- Realizar una investigación del estado del arte en modelado de tópicos.
- Elegir un enfoque de modelado de tópicos, proveniente de la investigación del estado del arte, obtener una implementación funcional, experimentar y analizar los resultados.
- Implementar una aplicación web, accesible para usuarios sin conocimiento técnico, que incluya modelado de tópicos y funcionalidades básicas de manejo de noticias.

1.3. Cronograma

En la figura 1.2 se presenta el cronograma de las actividades realizadas durante el presente proyecto de grado.

1.4. Organización del documento

En el capítulo 2 se presentan brevemente algunos conceptos necesarios para comprender el resto del informe y se exponen los resultados de la investigación del estado del arte en modelado de tópicos. Posteriormente, en el capítulo 3 se explica en detalle el sistema de modelado de tópicos utilizado y en el capítulo 4 se detalla la implementación de la

2021	Mayo	Junio	Julio	Agosto	Setiembre	Octubre	Noviembre	Diciembre
Reuniones con equipo de la FIC								
Investigación del estado del arte								
Experimentación con ETM								
Determinación del alcance								

2022 - 1	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto
Estudio de herramientas para programación web								
Preparación para presentación pública del OMU								
Presentación pública del OMU								
Desarrollo del sistema de modelado de tópicos								
Desarrollo de la aplicación web								
Pruebas con corpus de La Diaria								
Redacción del informe								
Correcciones del informe								

2022 - 2	Setiembre	Octubre
Redacción del informe		
Correcciones del informe		

Figura 1.2: Cronograma de actividades realizadas durante el proyecto.

aplicación web que permite acceder al referido sistema. En el capítulo 5 se analizan los resultados obtenidos en modelado de tópicos y en el capítulo 6 se presentan las conclusiones y reflexiones sobre posible trabajo futuro.

Capítulo 2

Conceptos previos y estado del arte

Dentro del Procesamiento de Lenguaje Natural existe una disciplina denominada modelado de tópicos, la cual incluye como caso particular al problema de identificar tópicos en noticias. Se realizó una investigación del estado del arte en esta disciplina, con el fin de elegir una metodología a aplicar en el presente proyecto.

En las secciones 2.1, 2.2 y 2.3 se presentan conceptos necesarios para comprender el modelado de tópicos y las metodologías que se presentarán. Posteriormente, en la sección 2.4 se define en detalle el modelado de tópicos. En las siguientes secciones se presentan los resultados centrales de la investigación del estado del arte, introduciendo primero la metodología históricamente más relevante (sección 2.6.1) y luego un enfoque reciente que enriquece al anterior (sección 2.6.4).

2.1. Procesamiento de Lenguaje Natural

El Procesamiento de Lenguaje Natural (PLN), según Daniel Jurafsky y James H. Martin [3], es «*un campo interdisciplinario que apunta a realizar automáticamente tareas útiles que involucren lenguaje humano, como comunicación humano-computadora, mejorar comunicación entre humanos o cualquier procesamiento útil de texto o habla*». El término «lenguaje humano» hace referencia a los lenguajes que los humanos utilizamos para comunicarnos día a día, como son el español, el inglés o el guaraní. Es un campo interdisciplinario porque involucra fuertemente la computación, la estadística y la lingüística, entre otras disciplinas.

Para referirnos a los lenguajes humanos utilizaremos el término «lenguaje natural». Los lenguajes naturales, como su nombre lo sugiere, han surgido (y constantemente evolucionan) de forma natural, sin reglas estrictas, por lo que pueden presentar ambigüedades. Estas últimas características (entre otras, como pueden ser la presencia de ironía, humor o doble sentido) complejizan mucho su procesamiento automático.

El PLN es un área activa de investigación, cuyos avances suelen tener aplicaciones

concretas en el corto o mediano plazo. En el Grupo PLN – UdelaR[1], hay líneas de investigación sobre análisis de discurso y extracción de información, entre otras. En muchos problemas del PLN (por ejemplo el modelado de tópicos) los resultados han mejorado significativamente debido al auge del aprendizaje automático, tema que se presenta en la siguiente sección.

2.2. Aprendizaje automático

Tradicionalmente, los problemas de PLN se abordaban mediante conjuntos de reglas escritas manualmente por personas. A modo de ejemplo, para el problema de convertir una palabra del singular al plural, algunas reglas sencillas son que si la palabra termina en vocal se añade el sufijo «s» y si termina en consonante se añade el sufijo «es» (observar que hay excepciones, como la palabra «maní»; para resolver el problema exactamente se deberían agregar más reglas). Para tareas sencillas, como la mencionada, este enfoque daba buenos resultados y se sigue utilizando. Sin embargo, cuando la tarea implica el procesamiento de oraciones o textos completos (como por ejemplo la traducción automática o el modelado de tópicos), la cantidad de reglas necesarias crece tanto que el enfoque se vuelve muy costoso (debido a la cantidad de trabajo necesario para escribirlas) y aún así los resultados pueden ser insatisfactorios.

Actualmente, para muchos problemas de PLN se utiliza aprendizaje automático, que según Tomas M. Mitchell [4] consiste en construir programas que automáticamente mejoran su desempeño con la experiencia. Esto en general se logra utilizando métodos basados en estadística u optimización matemática.

En la mayoría de los casos, esta experiencia proviene de un conjunto de datos. En base a este se infieren patrones, con los que se ajusta un modelo que luego se utiliza para realizar cierta tarea. Cuando el conjunto de datos es un conjunto de textos se lo denomina con la palabra «corpus». A modo de ejemplo, en el modelado de tópicos (que se presenta en detalle en la sección 2.4) se desea detectar los tópicos subyacentes a un corpus formado por documentos, que pueden ser notas de prensa. El modelo resultante luego se puede utilizar para inferir los tópicos presentes en otros documentos.

Para obtener buenos resultados utilizando estas técnicas se necesita una gran cantidad de datos. Por otra parte, para procesarlos se necesita poder de cómputo, cuyo aumento en las últimas décadas ha sido clave para que el aprendizaje automático se vuelva una herramienta aplicable en la práctica. Además de los datos y la capacidad de cómputo, el último componente crucial para los modelos de aprendizaje automático es un algoritmo que permita inferir los patrones presentes en el conjunto de datos y utilizarlos con otros

[1] `https://www.fing.edu.uy/inco/grupos/pln`

datos[2]. A la etapa en que el algoritmo infiere patrones presentes en los datos se le llama entrenamiento. Usualmente, durante el entrenamiento se procesa todo el conjunto de datos reiteradas veces. Cada iteración por el conjunto de datos se denomina *epoch*.

Un tipo de modelo que actualmente proporciona los mejores resultados para muchos problemas es la red neuronal. La metodología de modelado de tópicos que se utilizó contiene internamente una red neuronal, como se verá en el capítulo 3. A modo introductorio, las redes neuronales más elementales son las de tipo *feed–forward*, de las cuales se muestra un ejemplo en la figura 2.1. Este modelo se encuentra conformado por una serie de capas, la primera de las cuales se llama capa de entrada, la última, capa de salida y las demás se denominan capas ocultas. Cada capa está formada por cierta cantidad de neuronas, cada una de las cuales recibe un número como entrada y le aplica una función no lineal para producir una salida. La entrada de las neuronas de la capa de salida o de las capas ocultas es una combinación lineal de las salidas de las neuronas de la capa anterior. Esto se realiza multiplicando el vector de salidas de la capa anterior por una matriz de coeficientes. El funcionamiento de la red neuronal consiste en comenzar con un vector de entrada, multiplicarlo por una matriz para obtener la entrada de la siguiente capa, aplicar la función no lineal coordenada a coordenada y repetirlo hasta llegar a la capa de salida. La cantidad de neuronas por capa y la cantidad de capas ocultas intermedias varían según la red. Con varias capas ocultas, la red en su totalidad representa una función altamente no lineal.

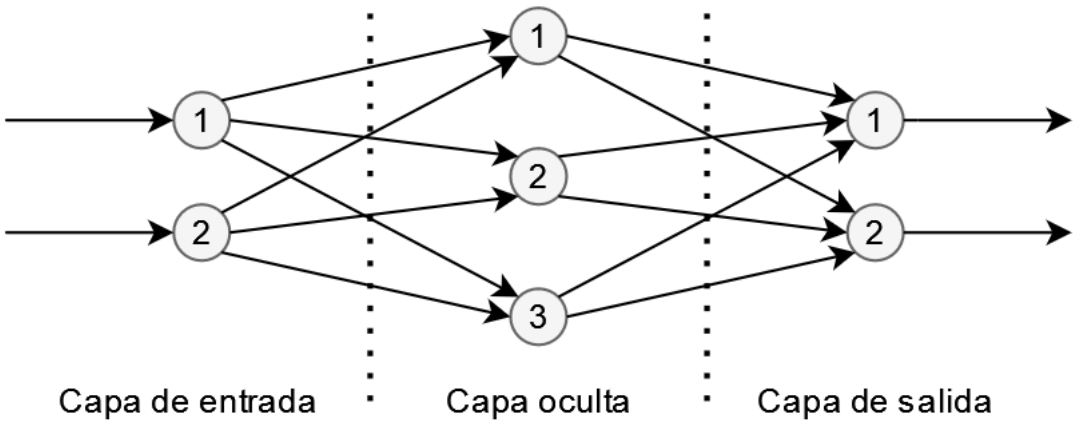


Figura 2.1: Esquema de red neuronal *feed–forward sencilla*.

Las variables de la red neuronal son los coeficientes de las matrices que se multiplican a la salida de una capa para obtener la entrada de la siguiente. Al entrenar la red, el objetivo es encontrar los valores de estas variables que minimicen una función de pérdida dada (la cual estima el error cometido por el modelo). La minimización se suele realizar

[2]En muchos casos los algoritmos fueron ideados por matemáticos o informáticos teóricos anteriormente a la existencia de datos y capacidad de cómputo suficientes para aplicarlos.

mediante descenso por gradiente: se comienza con valores aleatorios en las variables, se calcula el gradiente de la función de pérdida en ese punto y se modifican las variables en la dirección opuesta, iterando el cálculo de gradiente y modificación en dirección opuesta hasta que se cumpla alguna condición de parada. Para calcular el gradiente de la función de pérdida se deben computar sus derivadas parciales respecto a cada variable. Para esto se suele utilizar el algoritmo *backward propagation* [5].

2.2.1. Entrenamiento, validación y verificación

Como ya se mencionó, en la mayoría de las aplicaciones de aprendizaje automático el desempeño se mejora entrenando con datos.

Usualmente los modelos tienen parámetros que afectan los resultados obtenidos. A modo de ejemplo, en el contexto de las redes neuronales, se podrían considerar como parámetros la cantidad de capas y de neuronas por capa. Es deseable utilizar los valores de parámetros que den mejores resultados, pero normalmente no se los puede determinar *a priori*. Para resolver este problema se utiliza la validación: se entrena el modelo con distintos valores posibles de los parámetros y se valida cada uno en un conjunto de datos separado, utilizando alguna medida de desempeño. De este modo, se conserva la combinación de parámetros que produce los mejores resultados.

Una vez que el modelo fue entrenado y validado, si se desea utilizarlo en la práctica es deseable tener alguna medida de su desempeño. Para eso se utiliza la verificación: sin modificar el modelo, se lo prueba y se registra el desempeño obtenido.

Para poder aplicar validación o verificación es necesario disponer de alguna métrica cuantitativa que mida el desempeño del modelo. Dependiendo del problema, puede haberla o no.

Es fundamental que se utilicen distintos datos para el entrenamiento, la validación y la verificación. Usualmente, el conjunto de datos se separa con 70 % para entrenamiento, 10 % para validación y 20 % para verificación, pero dichos porcentajes pueden variar. Esta separación es imprescindible porque si se utilizan los mismos datos para entrenar y evaluar el modelo, este memorizará los patrones observados sin desarrollar capacidad de generalización; los resultados serán perfectos en los datos de entrenamiento pero malos con otros datos distintos, por lo que el modelo será inútil. A este problema se le llama sobreajuste y es uno de los más serios en aprendizaje automático.

2.2.2. Preprocesamiento

El desempeño posterior de un modelo suele estar relacionado con la calidad del conjunto de datos con el que fue entrenado: la existencia de datos mal ingresados, inconsistentes o redundantes puede degenerar el rendimiento del modelo resultante. Por otra parte, en

algunos casos hay que llevar los datos a un formato específico que el modelo necesita para poder entrenar con este. Para mejorar la calidad de los datos y para llevarlos al formato requerido por el modelo, suele haber antes del entrenamiento una etapa denominada preprocesamiento.

El preprocesamiento varía según el problema a resolver y la metodología a utilizar. Algunas tareas que pueden estar incluidas dentro del preprocesamiento son:

- Tratamiento de filas con entradas vacías. Esto se debe realizar cuando los datos son de tipo tabular y hay información faltante.
- Integración de datos. En algunos casos los datos a utilizar tienen diversos formatos y es necesario uniformizarlos.
- *Tokenización*. En PLN cuando se trabaja con textos, la *tokenización* consiste en separarlos en unidades atómicas, denominados *tokens*. Por ejemplo, en la frase «La capital de Argentina es Buenos Aires», los *tokens* son «La», «capital», «de», «Argentina», «es» y «Buenos Aires». Notar que en la mayoría de los casos los *tokens* están dados por una sola palabra, pero hay excepciones, como con el nombre de ciudad «Buenos Aires». Existe la posibilidad de ignorar lo anterior y considerar cada palabra como un *token*, lo cual simplifica la tarea con el costo de perder cierta información (en el ejemplo anterior las palabras «Buenos» y «Aires» se tratarían como independientes).
- Dado un corpus, eliminar las *stop words*: palabras sin significado, como los determinantes o las preposiciones. En la figura 2.2 se pueden observar algunas *stop words* típicas.

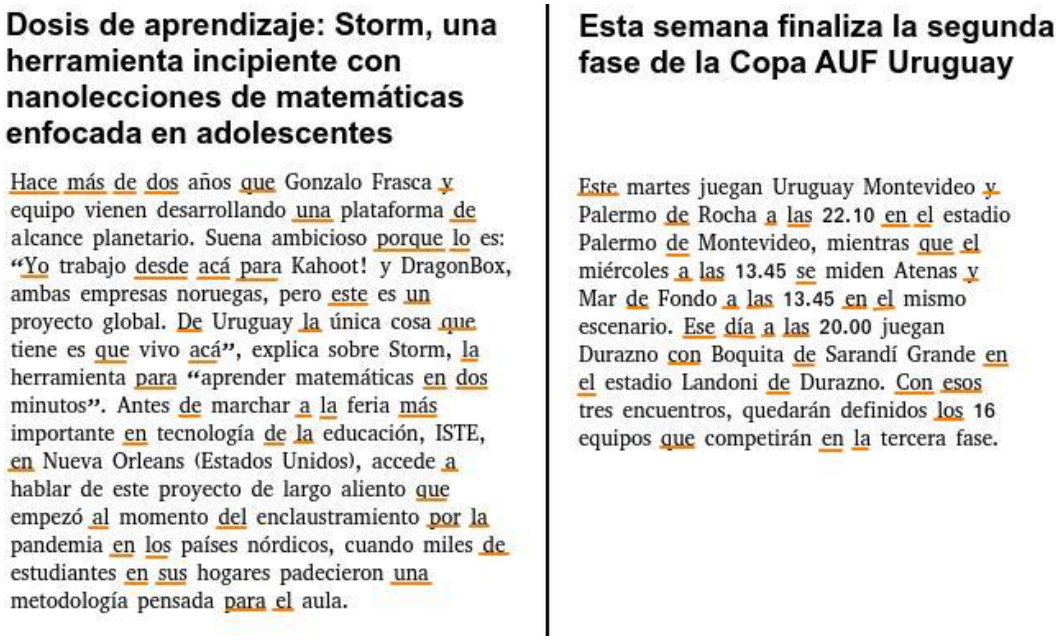
En la sección 3.1.1 se presenta el preprocesamiento utilizado en el presente proyecto.

2.2.3. Aprendizaje supervisado y no supervisado

A grandes rasgos, las técnicas de aprendizaje automático se clasifican en supervisadas y no supervisadas, en función de los datos que necesitan para entrenar.

En el caso supervisado, los datos deben estar acompañados por la respuesta correcta. Intuitivamente, el sistema aprende procesando casos para los que tiene acceso a la solución. Un ejemplo es un sistema que debe aprender a etiquetar una crítica como positiva, neutra o negativa, entrenando con un conjunto de críticas que de antemano están etiquetadas como positivas, neutras o negativas. Este enfoque tiene la dificultad de que además de necesitar datos, estos deben estar etiquetados. El etiquetado de los datos suele ser realizado por humanos, lo cual puede implicar mucho tiempo y costo.

El enfoque no supervisado aprende patrones en datos no etiquetados. El ejemplo más común es el *clustering*, que consiste en tener un conjunto de datos y automáticamente



Dosis de aprendizaje: Storm, una herramienta incipiente con nanolecciones de matemáticas enfocada en adolescentes

Hace más de dos años que Gonzalo Frasca y equipo vienen desarrollando una plataforma de alcance planetario. Suena ambicioso porque lo es: "Yo trabajo desde acá para Kahoot! y DragonBox, ambas empresas noruegas, pero este es un proyecto global. De Uruguay la única cosa que tiene es que vivo acá", explica sobre Storm, la herramienta para "aprender matemáticas en dos minutos". Antes de marchar a la feria más importante en tecnología de la educación, ISTE, en Nueva Orleans (Estados Unidos), accede a hablar de este proyecto de largo aliento que empezó al momento del enclaustramiento por la pandemia en los países nórdicos, cuando miles de estudiantes en sus hogares padecieron una metodología pensada para el aula.

Esta semana finaliza la segunda fase de la Copa AUF Uruguay

Este martes juegan Uruguay Montevideo y Palermo de Rocha a las 22.10 en el estadio Palermo de Montevideo, mientras que el miércoles a las 13.45 se miden Atenas y Mar de Fondo a las 13.45 en el mismo escenario. Ese día a las 20.00 juegan Durazno con Boquita de Sarandí Grande en el estadio Landoni de Durazno. Con esos tres encuentros, quedarán definidos los 16 equipos que competirán en la tercera fase.

Figura 2.2: *Stop words* de las notas periodísticas de ejemplo marcadas con naranja.

detectar patrones y separarlos en cierta cantidad de conjuntos disjuntos. Otro ejemplo, detallado en la sección 2.4, es el modelado de tópicos, en el cual se centra el presente proyecto. Como los datos no deben estar etiquetados, este enfoque suele ser menos costoso.

Existe también un tercer enfoque denominado aprendizaje semisupervisado. Se trata de un punto medio entre los dos anteriores, en el que se utilizan datos etiquetados y no etiquetados conjuntamente. Para profundizar se recomienda [6].

2.3. Embeddings

En PLN suele ser útil tener las palabras representadas de forma vectorial; es decir, que cada palabra tenga asociado un vector $v \in \mathbb{R}^d$, siendo la dimensión d fija.

Una metodología tradicionalmente utilizada es *one hot encoding*. En esta, la dimensión es $d = |V|$, siendo V el vocabulario utilizado y por lo tanto $|V|$ su cantidad de palabras. La representación vectorial de la n-ésima palabra de V, es el vector $v = (v_0, \ldots, v_d)$ con:

$$v_i = \begin{cases} 1 & \text{si } i = n \\ 0 & \text{en otro caso} \end{cases}$$

Equivalentemente, la palabra n-ésima es representada por el vector que tiene 1 en la n-ésima coordenada y 0 en todas las demás. Esta metodología, si bien puede utilizarse en algunos casos, tiene desventajas.

- Si el vocabulario es grande, cada palabra se representa con un vector de dimensión

alta, lo cual implica más gasto de memoria y estructuras de mayor tamaño, lo cual conlleva más cálculos.

- La mayoría de las entradas de los vectores valen 0, por lo que se dice que *one hot encoding* es una representación dispersa.

- La distancia euclídea entre los vectores de dos palabras vale 0 si son la misma palabra y $\sqrt{2}$ si son distintas, por lo que no aporta información respecto a similitud semántica (e.g. la distancia entre los vectores de «perro» y «canino» es la misma que entre los de «perro» y «júpiter»).

Para intentar solucionar estas desventajas surgieron los *embeddings*.

Los *embeddings* conforman una representación vectorial de las palabras que soluciona las 3 desventajas previamente mencionadas: es de dimensión baja (comparada con el tamaño del vocabulario), es densa (los vectores tienden a tener todas las entradas distintas a 0) y baja distancia euclídea se corresponde con semántica similar. Para ejemplificar esto último, es de esperarse que la distancia entre el vector asociado a «rey» y el asociado a «reina» sea menor a la distancia entre el vector asociado a «rey» y el asociado a «pizarra».

Estas representaciones se popularizaron debido a un trabajo por Mikolov et al. en 2013 [7], en el que se obtienen de forma estadística en base a un corpus. La metodología, a grandes rasgos, consiste en:

1. Inicializar los vectores asociados a las palabras de forma aleatoria.

2. Recorrer los documentos y para cada palabra w:

    - Para cada una de las palabras circundantes (en una ventana de tamaño fijo), acercar su vector y el de w (en distancia euclídea).
    - Elegir una palabra aleatoria del vocabulario (que probablemente no tenga nada que ver con w) para alejar su vector y el de w (en distancia euclídea).

De esta manera, los *embeddings* se han vuelto una herramienta común en PLN que se usa para mejorar los resultados en muchos problemas. En particular, se ha aplicado en modelado de tópicos, cómo se verá en la sección 2.6.4.

2.4. Modelado de tópicos

El modelado de tópicos es un problema de PLN que consiste en inferir los tópicos (temas) subyacentes a un conjunto de documentos (textos de cualquier tipo). Este proyecto de grado se centra en el caso en que los documentos son noticias. Como hipótesis, supone que existen tópicos abstractos que cada documento tiene en distintas proporciones. El

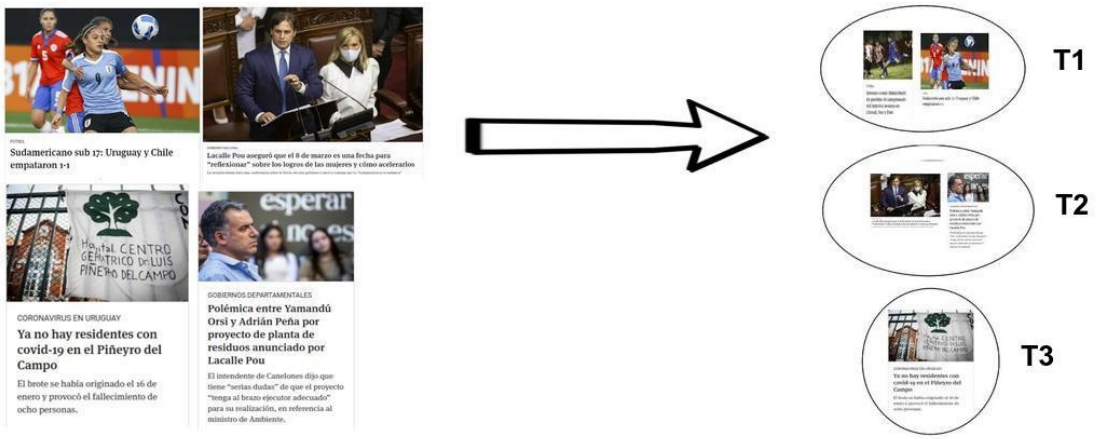


Figura 2.3: Ejemplo simplificado de modelado de tópicos.

objetivo es a la vez determinar cuáles son dichos tópicos, cuáles están en cada documento y en qué proporción. Como los tópicos se infieren por el algoritmo en lugar de estar predefinidos, necesariamente es un problema de aprendizaje no supervisado, que entrena con un conjunto de documentos en el que no se asume ningún orden.

A modo de ejemplo, la figura 2.3 ilustra de forma simplificada el problema a resolver. Se parte de un corpus de noticias (las de la izquierda) y se detectan los tópicos presentes: T1, T2 y T3 en este caso, de modo que cada noticia pertenece a uno de ellos. En otras palabras, se detectan tres grupos de noticias vinculadas entre sí: las de T1, las de T2 y las de T3 (un humano podría observar las noticias pertenecientes a cada tópico y asociarle un tema, por ejemplo que todas las noticias del tópico T1 traten de deporte; sin embargo, eso está por fuera del modelado de tópicos). El ejemplo es una simplificación porque en general las noticias no tienen por qué estar en un único tópico, sino que pueden pertenecer a varios en distintas proporciones (intuitivamente, puede haber una noticia que tenga los temas «deporte» y «salud» en distintas proporciones).

En la siguiente sección se citan aplicaciones de modelado de tópicos para problemas variados y en las posteriores se presentan las metodologías de modelado de tópicos más relevantes que se encontraron en la investigación del estado del arte.

2.5. Aplicaciones de modelado de tópicos

Se han dado muchas aplicaciones de modelado de tópicos, principalmente utilizando LDA (ver seccción 2.6.1) con o sin modificaciones, para diversos problemas dentro y fuera del PLN.

D. Kim y A. Oh presentaron en 2011 un artículo en el que, basándose en modelado de tópicos mediante LDA, determinaron tópicos presentes en archivos de noticias y

estudiaron su evolución temporal [8]. Para esto último utilizaron varias métricas de similitud entre tópicos. Por otra parte, aportaron en la evaluación de calidad de tópicos; encontraron que la mayoría de los tópicos que perduraron en el tiempo tenían significado semántico, mientras que los aislados eran menos coherentes.

X. Liu et al. presentaron en 2021 un artículo en el que utilizan modelado de tópicos para clasificar defectos en *software* para radares [9]. De acuerdo a los autores, es común que sistemas de dichos artefactos contengan centenas de miles de líneas de código, por lo que los defectos suelen ser complejos y diversos. Crearon una variante de LDA que clasifica descripciones de defectos para facilitar la reutilización.

M. P. Peters et al. presentaron en 2022 un artículo en el que utilizan LDA para inferir tipos de bosques [10]. Disponen de datos respecto a presencia de especies de árboles en cierta zona, la cual discretizan mediante una grilla y registran las especies presentes en cada celda. De esta forma, la correlación con conceptos de modelado de tópicos es la siguiente:

- Palabras — especies de árboles
- Documentos — celdas de la grilla
- Tópicos — tipos de bósque

M. Taboada et al. presentaron en 2021 la herramienta *gender gap tracker* para medir desigualdad de género en medios de comunicación [11]. Su principal componente es la determinación de personas citadas y su género, con la cual se puede inferir automáticamente si las noticias están sesgadas hacia cierto género. En una de sus aplicaciones, combina lo anterior con modelado de tópicos mediante LDA , logrando identificar tópicos sesgados hacia mujeres, neutros o hacia hombres[3].

2.6. Metodologías de modelado de tópicos

En esta sección se presentan las metodologías de modelado de tópicos más relevantes que se encontraron en la investigación del estado del arte. Posteriormente, en el capítulo 3 se indica cuál se eligió utilizar, justificando los motivos.

2.6.1. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) es una metodología de modelado de tópicos introducida por Blei en 2003 [12], que a pesar de su antigüedad, es una de las más usadas actualmente.

[3]En `https://gendergaptracker.research.sfu.ca/apps/topicmodel` se puede ver resultados del análisis de sesgo de género por tópicos en tiempo real.

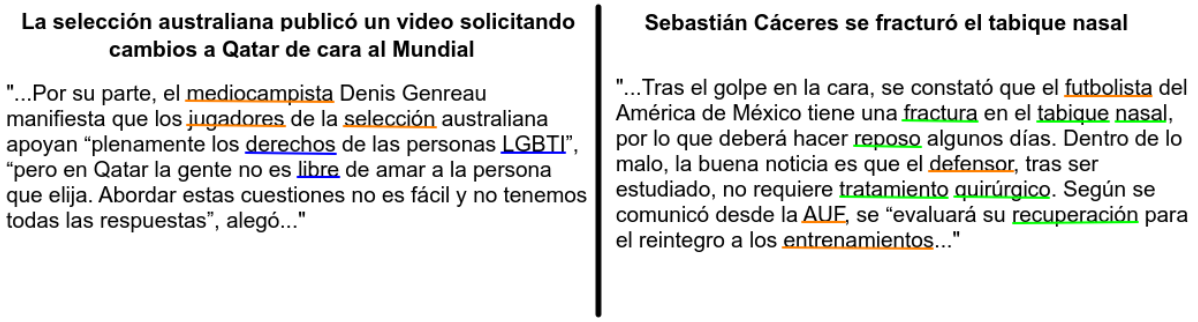


Figura 2.4: Comparación de tópicos presentes en fragmentos de noticias.

Conceptualmente, LDA busca afrontar el modelado de tópicos en función las palabras presentes en cada documento: asume que los tópicos son distribuciones de palabras y que cada documento está conformado por una mezcla de tópicos en distintas proporciones. Cada tópico, en otras palabras, induce tasas de ocurrencia de palabras. A modo de ejemplo, la palabra «Abogado» puede ser relativamente común dentro de tópicos de policiales o políticos, mientras que puede ser muy rara dentro de un tópico de deporte. Esto resulta en palabras muy características de uno o varios tópicos (arbitro, juez, delantero, diputado, delito) y otras que suelen ser comunes a varios tópicos (reunión, trabajo, exterior). Estos conceptos se ejemplifican en la figura 2.4, en donde se muestran dos noticias diferentes, en las cuales se subrayan del mismo color palabras que son características del mismo tópico. Con lo cual a groso modo LDA podría identificar 3 tópicos diferentes, el tópico naranja, común en ambas noticias, el azul, perteneciente solo a la primera y el verde, perteneciente solo a la segunda (observando las palabras pertenecientes a cada tópico, se podría decir que el tópico naranja parece estar relacionado con el deporte, el azul con los derechos sociales y el verde con salud y medicina). LDA expresa la presencia de estos tópicos en un documento en base a proporciones. A modo de ejemplo la distribución la primer noticia de la figura 2.4 podría ser 60 % naranja, 40 % azul y 0 % verde, mientras que en la segunda noticia la distribución podría ser 58 % naranja, 0 % azul y 42 % verde. Si bien con anterioridad se vincularon los tópicos con temas, como por ejemplo deporte o política, cabe aclarar que esto no está incluido dentro del alcance de LDA, sino que es una tarea que se realiza por humanos después de que LDA infirió los tópicos.

Técnicamente, en LDA se asume que los tópicos son variables aleatorias ocultas que caracterizan a los documentos. Cada documento tiene una distribución oculta de tópicos que rige la distribución de palabras que ocurren en el documento, lo que sí podemos ver. El objetivo de LDA es obtener información respecto a los tópicos ocultos basándonos en la información que tenemos: las palabras de los documentos.

Para representar de forma estadística los documentos y los tópicos utiliza las siguientes hipótesis de modelado:

- Cada documento es una distribución probabilística de tópicos. Por ejemplo una noticia sobre la final del mundial podría conformarse por tópicos mayoritariamente asociados al deporte.

- Cada tópico es una distribución probabilística de palabras. Siguiendo el ejemplo de la final del mundial, en un tópico de deporte las palabras más prominentes podrían ser «pelota», «partido» o «cancha».

LDA es un método generativo, lo cual significa que se asume que los documentos del corpus son el resultado de un proceso generativo. El proceso para generar un documento es el siguiente:

1. Elegir la distribución de tópicos que conforman el documento, de acuerdo a la primer hipótesis de modelado.

2. Para cada una de las palabras del documento:

   a) Tomar un tópico según la distribución elegida en el paso 1.

   b) Elegir la palabra de acuerdo a la distribución asociada al tópico elegido.

El algoritmo 1 ilustra el proceso generativo. Se asume que hay M documentos, cada uno con N palabras (la cantidad de palabras puede depender del documento, pero no es relevante). Los demás elementos significan lo siguiente:

- α es un parámetro que define cómo se elige la distribución de tópicos de cada documento. Se utiliza en el ítem 1 del proceso generativo. Es el parámetro que define la proporción de tópicos que habrá en el conjunto de documentos generados. Si los documentos generados tienen en su mayoría el tópico asociado a deporte, es porque el parámetro α lo define de esa forma.

- β es una matriz que contiene la distribución de probabilidades de palabras correspondiente a cada tópico. Se utiliza en el ítem 2b del proceso generativo. Es el parámetro que define los tópicos, de acuerdo a la segunda hipótesis de modelado.

- θ_j es la distribución de tópicos elegida para el documento j. Es el resultado del ítem 1 del proceso generativo.

- z es el tópico elegido para cierta palabra. Es el resultado del ítem 2a del proceso generativo.

- w es la palabra concreta elegida. Es el resultado del ítem 2b del proceso generativo.

Algoritmo 1 Proceso generativo de LDA

```
Entrada: M; α; β
for j = 1, ..., M do
    θ_j = sortearDistribuciónTópicos(α)
            ▷ Dada la distribución de tópicos se genera un documento de N_j palabras
    for i = 1, ..., N_j do
        z = sortearTópico(θ_j)
        w = sortearPalabra(z, β)
        agregarPalabra(w, j)          ▷ Se agrega la palabra w al documento j-esimo
    end for
end for
```

Dados los valores de los parámetros α y β, el modelo generativo de LDA permite calcular la probabilidad de un corpus concreto de documentos (comparando las palabras reales con las que el modelo generativo predice). El entrenamiento de un modelo de LDA con un corpus consiste en hallar los parámetros α y β que maximicen la probabilidad del corpus. Una vez que se tienen los parámetros α y β se puede realizar inferencia: dado un documento concreto, se halla su distribución de tópicos, θ, y el tópico de cada palabra, z, que maximicen la probabilidad del documento. En otras palabras, se determinan los tópicos presentes en el documento y qué palabra pertenece a cada tópico. Tanto la determinación de los parámetros α y β que maximicen la probabilidad del corpus como la tarea de inferencia son computacionalmente intratables, por lo que se utilizan métodos aproximados, como se explica en [12].

Las principales ventajas que presenta LDA son: la posibilidad de asignar varios tópicos a cada documento, la posibilidad de entrenar con corpus grandes y la posibilidad de aplicarlo a documentos que no estén en este; a modo de comparación, pLSI (uno de los principales métodos anteriores [13]) tiene riesgo de sobreajuste al entrenarse con corpus grandes y es difícil de aplicar a documentos fuera de este. Respecto a desventajas de LDA, si bien el método halla automáticamente tópicos, tiene el defecto de que hay que fijar el número de tópicos de antemano. Para esta decisión el método no brinda ninguna ayuda. Otra desventaja es que el desempeño empeora al aumentar el tamaño del vocabulario [14].

2.6.2. Modelado de tópicos mediante algoritmos de redes

La presente metodología fue introducida en [15]. Una de las motivaciones son los siguientes defectos que los autores del artículo identifican en LDA:

- Carencia de fundamentos para las suposiciones bayesianas.
- Discrepancias con propiedades estadísticas de textos reales, como por ejemplo la ley de Zipf [16].

- Dificultad para elegir la cantidad de tópicos (hiperparámetro que uno debe fijar de antemano).

Independientemente del modelado de tópicos, dentro del campo de redes complejas se estudia el problema «detección de comunidades» [17], el cual consiste, a grandes rasgos, en encontrar conjuntos de vértices vinculados en grafos grandes. Al igual que el modelado de tópicos, es un problema de aprendizaje automático no supervisado. Los autores observan similitudes entre dichos problemas y la evolución histórica de las metodologías que se aplican en ambos. En base a esto, proponen una metodología que los unifica. En efecto, representan el modelado de tópicos como un caso particular de detección de comunidades, generando un grafo bipartito de documentos y palabras de la siguiente forma: por un lado están los nodos que representan los documentos y por otro los que representan las palabras, con arista en caso de ocurrencia de una palabra particular en un documento particular. En este contexto, las comunidades halladas se corresponden con los tópicos.

Según los experimentos realizados en el artículo, esta metodología provee mejores tópicos que LDA. Por otra parte, abre la posibilidad de enriquecimiento mutuo entre las dos disciplinas (modelado de tópicos y detección de comunidades).

2.6.3. Modelado de tópicos mediante autoencoders

Se ha introducido una metodología de modelado de tópicos que usa *autoencoders* [18]. Los *autoencoders* constan de dos redes neuronales: un codificador y un decodificador, que se entrenan para que el codificador pueda comprimir la entrada en K números reales (donde K es un hiperparámetro a definir) y luego el decodificador pueda recuperar la entrada lo mejor posible en base a esos K números. Si el tamaño de las entradas es mayor a lo que se puede representar con K números (lo más usual al utilizar *autoencoders*), no es posible recuperar perfectamente la entrada en base a los K números; el objetivo es minimizar el error. Se puede pensar que el *autoencoder* realiza una tarea de compresión de datos con pérdida de información.

La aplicación de *autoencoders* a modelado de tópicos se realiza de la siguiente forma. La entrada es el *bag of words* (BoW) de un documento, es decir un vectir de números que indica cuántas veces ocurre cada palabra del vocabulario. Los K números que se obtienen como salida del codificador representan K tópicos que el documento puede contener en distintas proporciones (en base a esos números, las probabilidades se obtienen utilizando la función *softmax*, explicada en la sección 3.1.2). El decodificador cumple la función de reconstruir el BoW del documento en base a sus tópicos. Como el proceso consiste en comprimir distribuciones de palabras y luego poder recuperarlas, es de esperarse que los datos comprimidos tengan información general, relacionada a conjuntos de palabras que suelen ocurrir en mismos documentos, lo cual hace plausible que representen tópicos.

Los experimentos realizados en [18] indican que esta metodología produce mejores tópicos que las preexistentes.

2.6.4. Embedded topic modeling

Embedded topic modeling (Dieng et al., 2020 [14]), abreviado ETM, es una metodología de modelado de tópicos que enriquece LDA con *embeddings*. La principal diferencia con LDA se da en la forma de representar los tópicos: teniendo disponibles *embeddings* para las palabras, se utiliza un vector del mismo espacio para representar cada tópico. La probabilidad de una palabra en un tópico es proporcional al producto escalar entre sus vectores asociados (el vector del tópico y el de la palabra).

Al entrenar un modelo de ETM se infieren los vectores correspondientes a los tópicos. Por otra parte, se pueden usar *embeddings* preentrenados o crearlos al mismo tiempo que se infieren los tópicos, con el mismo corpus.

En [14] se compara el desempeño de ETM con el de otras metodologías (incluyendo LDA) de forma cuantitativa y cualitativa. Los mejores resultados fueron los obtenidos con ETM. En comparación con LDA, una ventaja que presenta ETM es su robustez ante la presencia de *stop words*: en el estudio empírico observan que el modelo generó tópicos de *stop words* en los que estas quedaron, sin afectar los demás tópicos. Otra ventaja de ETM respecto a LDA es que el desempeño no se deteriora al aumentar el tamaño del vocabulario.

Las métricas cuantitativas utilizadas en [14] son las siguientes:

- Error de predicción. ETM es un modelo generativo (lo hereda de LDA): dados los tópicos de un documento, predice la distribución de probabilidad de palabras presentes. El error de predicción cuantifica la diferencia entre las distribuciones de palabras que genera el modelo y las distribuciones reales en los documentos del corpus (en la sección 3.1.4 se explica en detalle el cálculo del error de predicción).

- Coherencia de tópicos. Cuantifica la interpretabilidad, retornando valores mayores si las palabras del tópico tienden a aparecer en mismos documentos.

- Diversidad de tópicos. Se toman las palabras más relevantes de cada tópico y se calcula el porcentaje de palabras únicas (que están en un único tópico).

ETM es la metodología que se eligió utilizar en el presente proyecto. En el siguiente capítulo se explican los motivos de elegirla, se profundiza en su funcionamiento interno y se muestra cómo es usada para inferir tópicos en noticias de prensa.

Capítulo 3

Identificación automática de tópicos

El eje del presente proyecto radica en construir un sistema de modelado de tópicos, para el cual se debía elegir una metodología. Considerando la investigación del estado del arte realizada, se decidió utilizar ETM. La principal motivación es que ETM está basada en LDA, la metodología más utilizada actualmente, lo cual induce a pensar que se debieran obtener buenos resultados. Además, al enriquecer LDA con una herramienta nueva y exitosa como son los *embeddings*, es de esperar que se obtengan mejoras de desempeño en general.

Por otra parte, en el alcance del proyecto se incluye la implementación de un sistema que permita acceder a la metodología anterior. Este tiene dos componentes: un subsistema de modelado de tópicos y una aplicación web con funcionalidades básicas que permite acceder al primero (ver figura 3.1). El resultado es un sistema web con el cual un usuario sin conocimiento técnico puede utilizar la metodología de modelado de tópicos ETM.

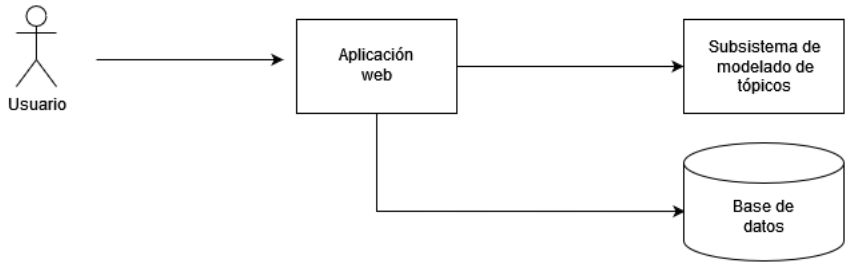


Figura 3.1: Arquitectura del sistema.

En la sección 3.1 se explica en detalle el funcionamiento interno de ETM, pasando por el preprocesamiento, el entrenamiento y la evaluación. Posteriormente, en la sección 3.2 se esbozan algunas consideraciones sobre cómo se utiliza ETM en el subsistema de modelado de tópicos. En la sección 4.1 se presentan algunos detalles de la aplicación web, mientras que en la 4.2 se describe el diseño de la base de datos relacional. Finalmente, en la sección 4.3 se comentan algunos detalles de implementación.

3.1. Lógica de ETM

En esta sección se describe en detalle el funcionamiento interno de la metodología de modelado de tópicos ETM.

3.1.1. Preprocesamiento del corpus

Como es usual en aplicaciones de aprendizaje automático, para utilizar ETM es necesario primero preprocesar el corpus. Las principales tareas que se realizan en el preprocesamiento son las siguientes:

1. Remover las *stop words* (palabras que no aportan información para la tarea a realizar). Se utilizan dos criterios para definir qué palabras son *stop words*:

   a) Frecuencia en documentos: si una palabra está en un porcentaje muy alto de documentos o en un porcentaje muy bajo, es considerada *stop word*.

   b) Lista predefinida de *stop words*: hay palabras que de antemano se sabe que no aportan información para determinar tópicos. Entre estas están los determinantes («el», «la», etc) y las preposiciones («a», «con», etc). La lista utilizada fue obtenida de un compendio de *stop words* en diferentes idiomas[1].

2. Crear un vocabulario, es decir una lista ordenada con las todas las palabras presentes (después de haber removido las *stop words*). Se utilizan *tokens* conformados por únicamente una palabra (si ocurre el nombre de ciudad «Fray Bentos», se lo toma como dos palabras independientes: «Fray» y «Bentos»).

3. Separar los documentos en conjuntos de entrenamiento, validación y verificación. Como se mencionó en la sección 2.2.1, cuando hay métricas de desempeño disponibles, es deseable validar y verificar aparte de entrenar y es fundamental que se usen distintos datos para esas tareas.

4. Se debe llevar los documentos al formato que el modelo de ETM requiere. Cada documento se representa mediante un vector de números enteros de igual tamaño que el vocabulario. En la entrada i-ésima del vector va la cantidad de ocurrencias en el documento de la palabra i-ésima del vocabulario, o 0 si la palabra no ocurre. Representaciones como esta se denominan *bag of words*: se tiene en cuenta la cantidad de veces que ocurre cada palabra del vocabulario, pero no su orden. De esta forma, en lugar de trabajar con cadenas de caracteres (los documentos «crudos») se utilizan listas de números enteros naturales.

[1] `https://github.com/Alir3z4/stop-words`

3.1.2. Estructura del modelo

La estructura interna de un modelo de ETM está determinada por varios hiperparámetros. Los más importantes conceptualmente son los siguientes:

- n: número de tópicos a inferir.
- d: dimensión del espacio de *embeddings*.
- V: tamaño (cantidad de palabras) del vocabulario.

Dados los hiperparámetros, los elementos constituyentes del modelo son los siguientes:

1. Una matriz de *embeddings*, ρ, con dimensión $d \times V$. La columna i-ésima es el vector que representa a la palabra i-ésima del vocabulario.
2. Una matriz de tópicos, α, con dimensión $d \times n$. La columna i-ésima es el vector que representa al tópico i-ésimo.
3. Una red neuronal de predicción de tópicos, θ, con entrada de tamaño V y salida de tamaño n. Se le ingresa como entrada un documento (con el formato descripto en el ítem 4 de preprocesamiento) y retorna la distribución de tópicos que contiene (el porcentaje del documento que corresponde a cada tópico; si son 3 tópicos puede ser 20 % del tópico 1, 30 % del tópico 2 y 50 % del tópico 3, a modo de ejemplo).

La matriz de tópicos, α junto con la matriz de *embeddings*, ρ, inducen una distribución de palabras por tópico (permiten asociar a cada palabra del vocabulario su probabilidad dado un tópico). La probabilidad de cada palabra dado un tópico es proporcional al producto escalar de sus vectores, los cuales están incluidos en las matrices anteriores. A modo de ejemplo, supongamos que un tópico T es representado por el vector $(1, 1, 1)$, mientras que la palabra «deporte» tiene el $(1, 2, 1)$ y la palabra «política» tiene el $(-2, -1, 0)$. Los valores de los productos escalares son los siguientes:

$$\begin{aligned} < T, \text{deporte} > &= 4 \\ < T, \text{política} > &= -3 \end{aligned}$$

Por lo tanto, la palabra «deporte» es más probable que la palabra «política» dentro del tópico T. Como los productos escalares pueden dar cualquier valor real, para obtener una distribución de probabilidades se aplica la función *softmax*[2].

[2]Dado un vector $z = (z_1, \ldots, z_n)$ de números reales, $\text{softmax}(z)$ es un vector con la misma cantidad de coordenadas pero con valores entre 0 y 1, cuya suma de todas las coordenadas da 1. Se calcula como: $\text{softmax}(z)_i = e^{z_i}/(\sum_{j=1}^{n} e^{z_j})$.

3.1.3. Entrenamiento del modelo

Al entrenar el modelo se modifican las entradas de las matrices ρ y α y los parámetros de la red neuronal θ. Esto se hace optimizando respecto a una función de pérdida, que se calcula de la siguiente forma, para cada documento.

1. Utilizando la red neuronal θ, se obtiene la distribución de tópicos del documento.

2. Cada tópico induce una distribución probabilística de palabras. Juntando las distribuciones de palabras de los tópicos con lo obtenido en el paso anterior, se obtiene una distribución esperada de palabras en el documento.

3. Se evalúa la diferencia entre la distribución de palabras calculada en el paso anterior y la verdadera distribución de palabras del documento. Esto se cuantifica con el opuesto del producto escalar entre el vector del documento (en el formato del ítem 4 del preprocesamiento, explicado en la sección 3.1.1) y la distribución de palabras calculada; el valor resultante es mayor cuanto más discrepa la distribución predicha de la real, lo cual es deseable para una función de pérdida. A modo de ejemplo, supongamos que el vocabulario está conformado por dos palabras y en un documento está tres veces la primera y ninguna vez la segunda. El vector del documento es $(3, 0)$. Si la distribución de palabras predicha es $(1, 0)$, significa que la primera palabra tiene 100 % de probabilidad y la segunda 0 %, en este caso la función de pérdida da -3. Por otra parte, si la distribución de palabras predicha es $(0, 1)$, significa que la primera palabra tiene 0 % de probabilidad y la segunda 100 % (lo cual es erróneo), en este caso la función de pérdida da 0, un valor superior.

Intuitivamente, el modelo se entrena para que los tópicos resuman las distribuciones de palabras presentes en los documentos, ya que en el entrenamiento se busca que conociendo los tópicos de un documento se pueda recuperar su distribución de palabras.

3.1.4. Evaluación del modelo

Para la evaluación cuantitativa se utilizan las métricas mencionadas en la sección 2.6.4. Se trata de las siguientes:

- Coherencia de tópicos. Cuantifica la interpretabilidad, retornando valores mayores si las palabras del tópico tienden a aparecer en mismos documentos.

- Diversidad de tópicos. Se toman las palabras más relevantes de cada tópico y se calcula el porcentaje de palabras únicas (que están en un único tópico).

- Error de predicción. Se calcula utilizando una función de pérdida similar a la del entrenamiento. La única diferencia con lo explicado para el entrenamiento es que en

este caso cada documento de prueba se separa en dos mitades: la primera se utiliza para determinar los tópicos del documento y calcular su distribución de palabras probable, la cual se compara con la distribución real de la segunda mitad.

Todas las métricas anteriores se calculan con el conjunto de datos de verificación. La coherencia y la diversidad de tópicos se calculan una única vez cuando el entrenamiento culminó. Por otra parte, el error de predicción (que es más rápido de evaluar) se calcula después de cada *epoch* y al final del entrenamiento se conserva el estado del modelo con el que se obtuvo el menor valor.

Para la evaluación cualitativa se pueden visualizar los tópicos y los *embeddings* del vocabulario. Para visualizar los tópicos se muestran las palabras más probables asociadas a cada uno (recordando que cada tópico induce una distribución de probabilidades para las palabras). Por otra parte, para visualizar los *embeddings* del vocabulario se fijan ciertas palabras y se muestran las que tienen vectores más cercanos. En el capítulo 5 se presentan y analizan resultados de análisis tanto cuantitativo como cualitativo.

3.2. Subsistema de modelado de tópicos

Se diseñó un subsistema que permite entrenar y utilizar modelos de ETM, con una interfaz accesible desde la aplicación web. Esto posibilita generar o enriquecer un corpus con documentos ingresados a la aplicación web y utilizarlo para entrenar un modelo de ETM.

Además de lo anterior, incluye una funcionalidad que dado un documento concreto y un conjunto de tópicos previamente inferidos (es decir, un modelo entrenado) clasifica el documento; específicamente, retorna la distribución de probabilidades de tópicos. Como primera aproximación se diseñó la siguiente heurística: promediar los *embeddings* de las palabras del documento, con lo que se obtiene un vector representativo de este, para luego calcular el producto escalar entre dicho vector y el correspondiente a cada tópico. Ya que el producto escalar entre vectores de palabras y de tópicos representa la probabilidad de que la palabra pertenezca al tópico, es razonable usar esta heurística para estimar la probabilidad de que cada tópico esté en el documento. Los resultados obtenidos fueron cualitativamente buenos, sin embargo, la red neuronal θ del modelo de ETM (descripta en la sección 3.1.2) resuelve el mismo problema con resultados superiores, por lo que se decidió utilizarla.

La última funcionalidad incluida en el subsistema de modelado de tópicos es la visualización de tópicos mediante sus palabras más probables.

3.2.1. Supervisado vs no supervisado

Si bien hasta el momento hemos hablado casi exclusivamente de modelado de tópicos (metodología que se encuadra dentro del aprendizaje no supervisado), esta no es la única forma de alcanzar el objetivo de la identificación de tópicos. En su lugar es posible utilizar un enfoque supervisado, lo cual altera en gran medida el problema, debido a que en vez de hablar de modelado de tópicos, se trata de clasificación de documentos. Dicho en otros términos, a diferencia del enfoque no supervisado en el que se infieren los tópicos, en el supervisado estos están dados a priori y el modelo se limita a clasificar los documentos (determinar qué tópicos están presentes). Para poder utilizar dicho enfoque, es necesario contar de antemano con un conjunto de tópicos y un corpus de documentos etiquetados con dichos tópicos.

Algunas de las ventajas de utilizar un aprendizaje supervisado son:

- Debido a que los tópicos se definen de antemano, se evita que el modelo pueda identificar alguno que no sea deseado. Un tópico podría ser no deseado si, por ejemplo, los investigadores no poseen interés en dicho tópico o si está compuesto por *palabra genéricas*, como se observa que ocurre en el apartado de estudio empírico de la investigación realizada por Dieng et al. [14].

- Los tópicos, al estar definidos de antemano, poseen una etiqueta con significado semántico (e.g. «deporte», «política», etc). Por otra parte, en metodologías no supervisadas, como ETM, los tópicos no poseen una etiqueta que los represente (salvo que un humano la agregue *a posteriori*).

- Son fáciles de evaluar debido a que basta comparar las asignaciones de tópicos del modelo con las provistas por el corpus en un subconjunto de verificación.

A pesar de estas ventajas del enfoque supervisado para la problemática actual, se utilizó un enfoque no supervisado. Esto se debe principalmente a dos motivos: por un lado un enfoque supervisado requiere un gran conjunto de documentos etiquetados por humanos. Por otro lado, la naturaleza de los tópicos es dinámica en el tiempo (e.g. cuatro años atrás el tópico covid no existía, ni era previsto), dificultando la utilización de un modelo supervisado, debido a que se debería agregar continuamente tópicos y etiquetar documentos a mano, lo cual es contrario al objetivo del presente proyecto.

Capítulo 4

Implementación de la solución

En este capítulo se detalla la implementación realizada. Primero se presentan la aplicación web y la base de datos subyacente. Luego se especifican las herramientas utilizadas en todos los componentes del sistema.

4.1. Aplicación web

Se trata de una aplicación con arquitectura cliente-servidor, como es usual en este tipo de sistemas, que permite acceder a funcionalidades básicas de manejo de noticias (agregar, modificar, buscar y borrar) y al modelado de tópicos mediante ETM. Para esto, interactúa con el subsistema de modelado de tópicos y la base de datos.

A modo de ejemplo, en la figura 4.1 se muestra la página principal de la aplicación web, en la que se muestran todas las noticias ingresadas y los tópicos inferidos automáticamente. Además, se pueden aplicar filtros y hay enlaces a las otras funcionalidades.

Algunos casos de uso son los siguientes:

1. Agregar noticia. El usuario puede agregar una noticia al sistema. Una vez que esta fue cargada, automáticamente se utiliza el modelo de ETM para clasificarla y se registran los resultados.

2. Visualizar tópico. El usuario puede elegir un tópico y visualizar las palabras más probables dentro de dicho tópico. Si encuentra que el tópico se corresponde con algún tema en particular, puede agregarle una etiqueta legible, la cual remplaza el nombre por defecto del tópico. La figura 4.2 muestra la página en la que se observan las palabras asociadas a un tópico y la posibilidad de cambiar su etiqueta.

3. Reinferir tópicos. El usuario puede dar la orden de que se entrene un nuevo modelo de ETM con el corpus actual, el cual está dado por un corpus de base y todas las noticias que se hayan agregado al sistema. El usuario puede elegir la cantidad

Figura 4.1: Página principal de la aplicación web.

de tópicos que el modelo debe aprender. Al utilizarse esta funcionalidad el modelo anterior se elimina y el sistema queda inaccesible durante el entrenamiento del nuevo modelo. Cuando este culmina, se lo utiliza para clasificar todas las noticias previamente ingresadas en el sistema con los nuevos tópicos.

4. Agregar palabras a ignorar. El usuario puede agregar palabras para que sean filtradas cada vez que se reinfieren los tópicos.

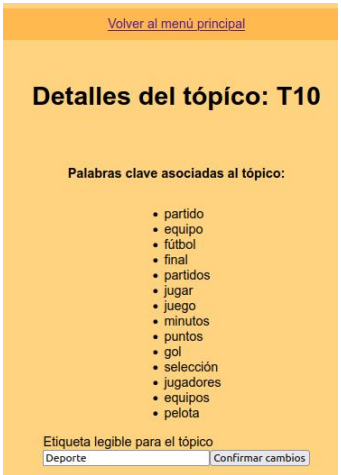


Figura 4.2: Visualización de las palabras asociadas a un tópico.

No está incluido en el alcance un sistema de identificación de usuarios, por lo que todos pueden utilizar todas las funcionalidades; en otras palabras, todos los usuarios

que pueden acceder al sistema tienen rol de administrador. Para que lo anterior sea inicialmente viable, se espera que la aplicación esté disponible solamente en una red local y sea accedida únicamente por el equipo del OMU. No obstante, en caso de que el sistema escale es de esperarse que sea necesario agregar gestión de usuarios.

4.2. Base de datos

El sistema incluye una base de datos relacional sencilla, con las tablas necesarias para cumplir los requisitos del sistema. En la figura 4.3 se presenta el esquema de entidad-relación, en el cual se pueden ver los elementos fundamentales: una entidad para representar los documentos (noticias), otra para los tópicos y una relación para la inclusión de tópicos en documentos.

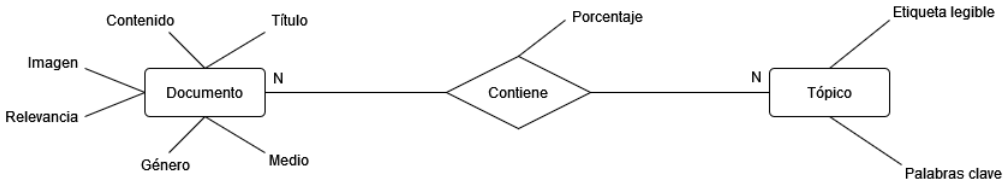


Figura 4.3: Esquema entidad-relación para la base de datos.

La entidad `Documento` representa las notas periodísticas que se ingresan al sistema. Sus atributos (título, contenido, medio, etc) están pensados para ser definidos por el usuario que ingresa la noticia al sistema.

La entidad `Tópico` representa los tópicos que son inferidos por el sistema. El atributo «palabras clave» es calculado por el sistema accediendo al subsistema de tópicos; son las palabras más probables dado el tópico. Por otra parte, el atributo «Etiqueta legible» puede ser ingresado por un usuario.

La relación `Contiene` representa la pertenencia de tópicos a documentos. El atributo «porcentaje» indica la proporción en que el tópico conforma el documento, para calcularlo se accede al subsistema de modelado de tópicos. La relación es total (cada documento está relacionado con cada tópico) y la suma de los porcentajes para cada documento es 100 %. Esto último no se visualiza en los ejemplos de la figura 4.1 (ni en los que se verán en el capítulo siguiente) porque en la aplicación web los porcentajes se redondean y se omiten los tópicos en los que resulta 0 %.

4.3. Herramientas utilizadas

Para todo el sistema se utilizó el lenguaje de programación *Python*. Esta decisión se basó en que los autores de ETM pusieron a disposición una implementación[1] en dicho lenguaje y en que contiene bibliotecas que simplifican el desarrollo de aplicaciones web.

La implementación de ETM utiliza la biblioteca *pytorch* para las tareas centrales de aprendizaje automático. Por otra parte, para implementar la aplicación web en *Python* se utilizó la plataforma Django. La elección de la plataforma se basa en la simplicidad de crear páginas de formularios y de visualización, además de que provee varias características web previamente implementadas.

Respecto a la base de datos relacional, se utilizó el manejador *PostgreSQL*. Además de lo mencionado en la sección 4.2, cabe destacar que se utilizó una tabla de banderas globales para solucionar problemas de concurrencia. Cuando se están reinfiriendo los tópicos, todas las funcionalidades del sistema quedan inhabilitadas para evitar inconsistencias.

[1] `github.com/adjidieng/ETM`

Capítulo 5

Análisis de resultados

En este capítulo se presentan y analizan los resultados obtenidos en modelado de tópicos con ETM. Para todos los resultados presentados se entrenó con un corpus provisto por La Diaria, que consta de las 20.000 noticias más recientes, a partir de junio de 2022 hacia atrás. Primero se presentan resultados cualitativos y luego cuantitativos.

Para el preprocesamiento del corpus de filtró un conjunto prefijado de *stop words*, obtenido del repositorio `https://github.com/Alir3z4/stop-words`. Además, se filtraron las palabras que ocurren en muy pocas noticias y las que ocurren en muchas; de forma más precisa, se estableció como frecuencia mínima 1 % y como frecuencia máxima 90 % (en ambos casos la frecuencia es el porcentaje de documentos en los que ocurre la palabra). Las cantidades de noticias utilizadas para entrenamiento, verificación y validación se muestran en la siguiente table.

	entrenamiento	verificación	validación	total
Noticias	17000	2000	1000	20000
Porcentaje	85 %	10 %	5 %	100 %

5.1. Análisis cualitativo

Para el análisis cualitativo que se presenta en esta sección, se utilizó un modelo de ETM entrenado con 20 tópicos.

En las figuras 5.1 y 5.2 se muestran las palabras más representativas de los tópicos aprendidos automáticamente por el modelo. Las palabras más representativas de cada tópico son las palabras del vocabulario a las que el tópico les asigna mayor probabilidad (como se explica al final de la sección 3.1.4).

Podemos ver que la mayoría de los tópicos que el sistema halló automáticamente se corresponden con temas intuitivos. Por ejemplo, como el tópico 5 tiene entre sus palabras representativas «obra», «película», «música» y «cine», inferimos que es un tópico asociado

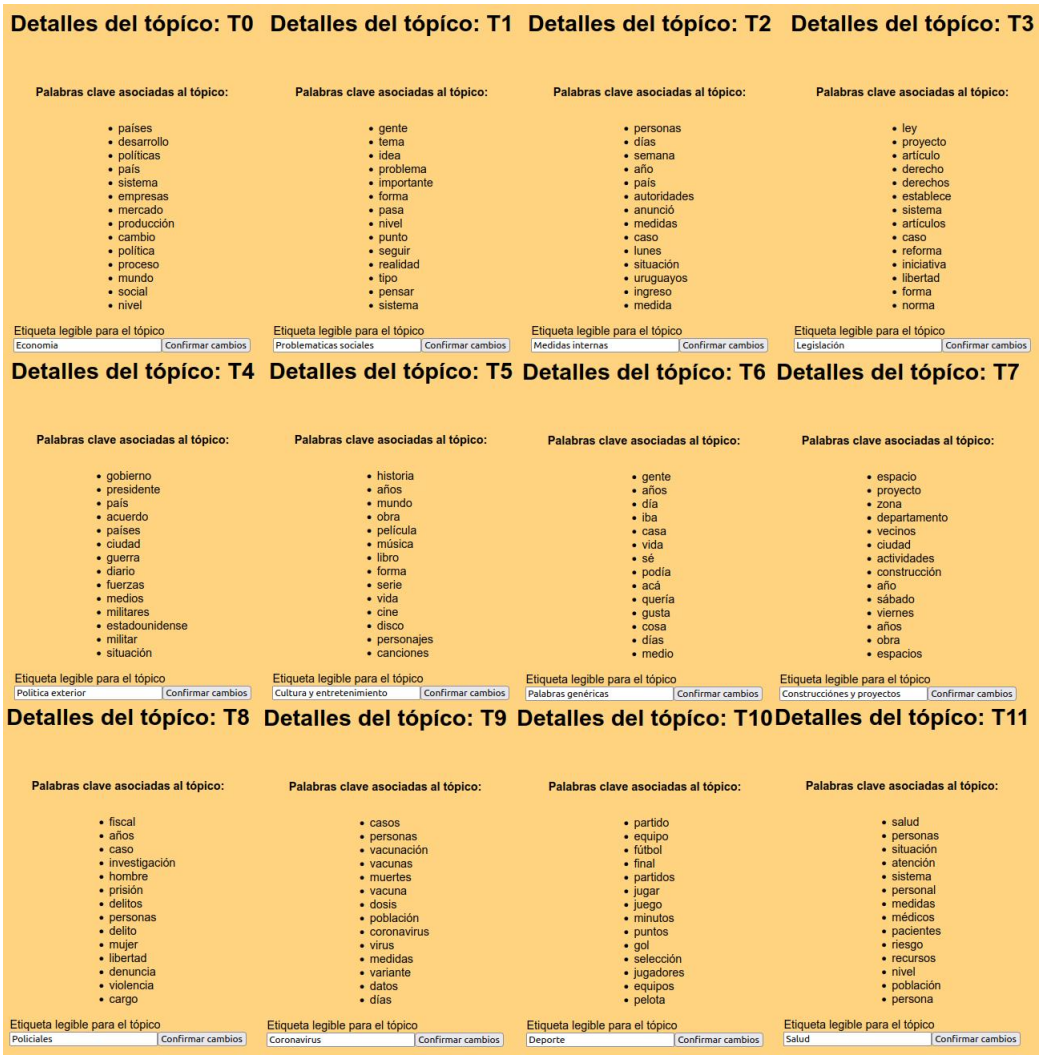


Figura 5.1: Tópicos identificados automáticamente, con etiqueta asignada manualmente en función a sus palabras asociadas – parte 1.

a cultura y entretenimiento. Análogamente, como el tópico 10 tiene las palabras «partido», «final», «gol» y «jugadores», deducimos que está vinculado al deporte.

La única excepción a la observación anterior es el tópico 6, para el cual no se encontró ningún tema intuitivo en base a sus palabras representativas; parece estar conformado únicamente por *stop words* y palabras comunes. Esto puede corresponderse con el comportamiento reportado por Dieng et al.: en caso de haber *stop words*, el sistema les asigna un tópico específico y no interfieren con los tópicos que tienen significado. En este caso se utilizaron metodologías para remover *stop words* durante el preprocesamiento, pero siempre existe la posibilidad de que hayan remanentes o palabras que simplemente son muy genéricas como para pertenecer a algún tópico.

En la figura 5.3 se muestran los mismos documentos de la figura 4.1 del capítulo anterior, tras asignar etiquetas semánticas a los tópicos. Se puede observar que cualitativamente los tópicos asignados automáticamente coinciden en gran medida con lo esperado:

- La noticia titulada «Esta semana finaliza la segunda fase de la Copa AUF Uruguay» tiene en mayor medida el tópico «Deporte».
- La noticia titulada «Uruguay se destaca a nivel internacional por avances en el tratamiento de los accidentes cardiovasculares» tiene en mayor medida el tópico «Salud».
- La noticia titulada «La famosa opera Pagliacci llega este miércoles al teatro Solís» tiene en mayor medida el tópico «Cultura y entretenimiento».

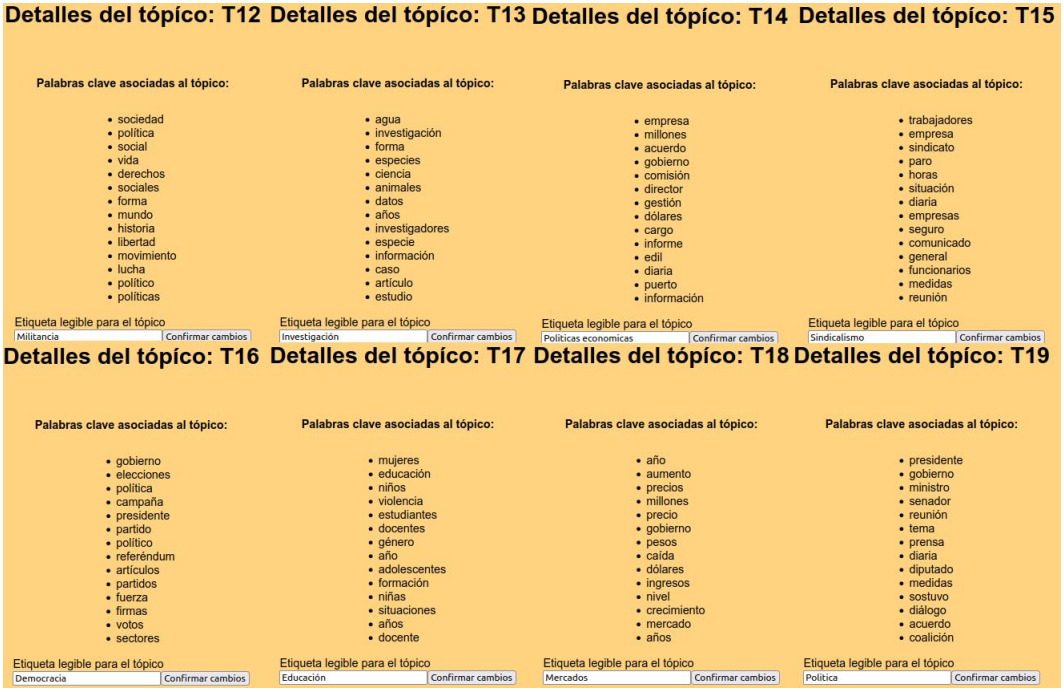


Figura 5.2: Tópicos identificados automáticamente, con etiqueta asignada manualmente en función a sus palabras asociadas – parte 2.

Si bien asignamos al tópico 10 la etiqueta «deporte», debido a las palabras que contiene también podría referirse únicamente al fútbol. Para profundizar sobre dicho tópico se agregó al sistema una noticia sobre un deporte diferente (en este caso el remo), con el fin de evaluar el porcentaje que se asigna al tópico 10 (deporte). Como se puede observar en la figura 5.4, la noticia sobre remo posee un porcentaje relativamente alto en el tópico deporte, sin embargo también presenta un valor mucho más grande en el tópico de palabras genéricas. Podemos ver, entonces, que no a todas las noticias de deporte se

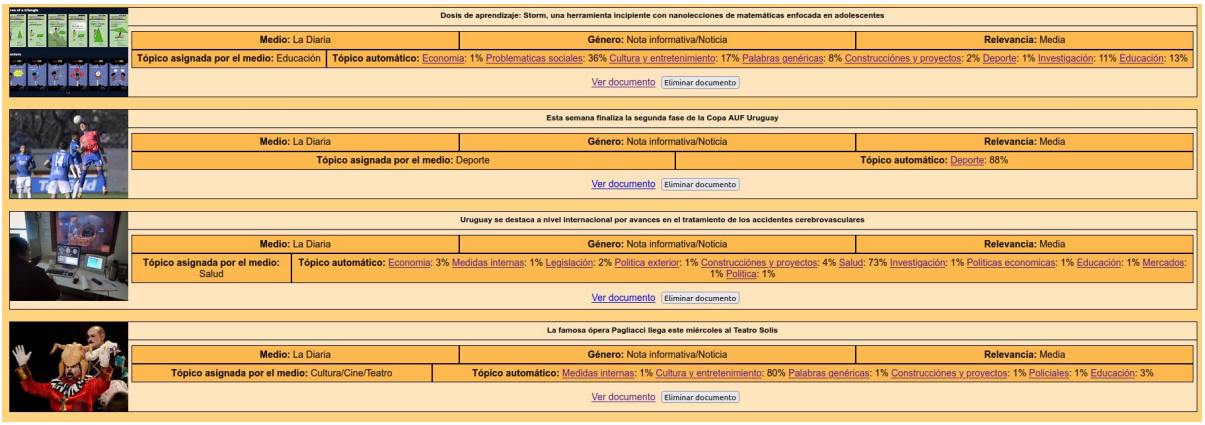


Figura 5.3: Documentos tras asignar etiquetas a los tópicos

les asigna dicho tópico sobre los demás. Esto puede deberse a que el tópico 10 tenga una inclinación hacia el fútbol sobre los demás deportes o que las noticias de remo, al ser muy infrecuentes en el corpus, tengan la mayoría de sus palabras características filtradas durante el preprocesamiento, quedando principalmente *stop words* y palabras comunes.

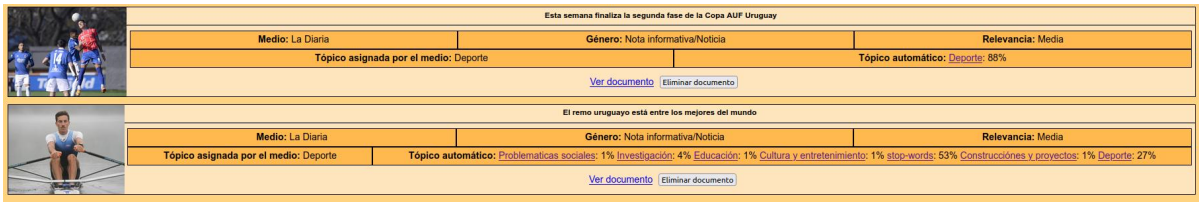


Figura 5.4: Comparación entre las asignaciones de tópicos entre noticias de fútbol y remo

Debido a que tanto el entrenamiento como los experimentos fueron realizados con noticias de La Diaria, se corre el peligro de caer en una suerte de sobreajuste, es decir, que el modelo sea bueno prediciendo noticias de La Diaria pero malo con noticias de otros medios. Para poner a prueba esto se utilizaron 3 noticias muy similares pero de diferentes medios, en este caso el estreno de la ópera Pagliacci. En la figura 5.5 se observa que si bien para la noticia de El País el sistema detecta una gran cantidad de tópicos aparentemente no relacionados, en todos los casos el tópico predominante es el de cultura y entretenimiento.

Culminando el análisis cualitativo, se intentó visualizar la calidad de los *embeddings*. Para esto se eligieron seis palabras de prueba y para cada una de ellas, se buscaron otras 20 del vocabulario con vectores más cercanos, teniendo en cuenta que *embeddings* de buena calidad cumplen que las palabras cercanas tienen una semántica relacionada. Los resultados se muestran en la figura 5.6. Las primeras cinco palabras («deporte», «política», «salud», «educación» y «economía») son temas recurrentes y se puede ver

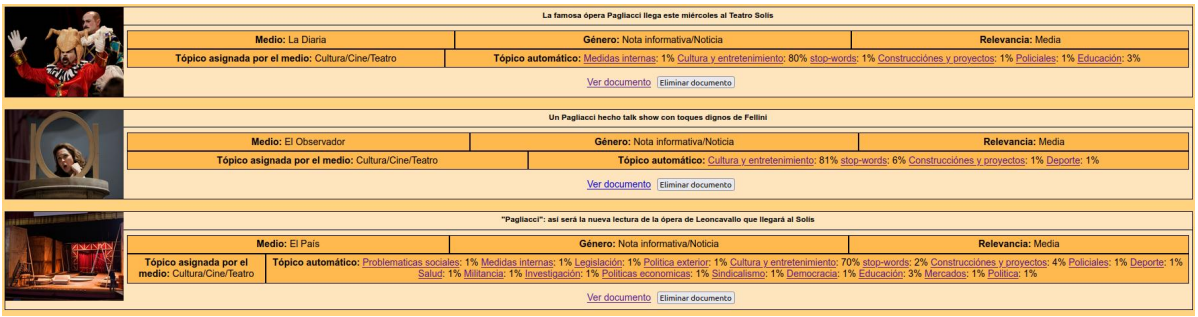


Figura 5.5: El estreno de Pagliacci en diferentes medios

que las palabras cercanas pertenecen a dichos temas. Por otra parte, la sexta palabra («acá») es una *stop word* que no fue filtrada en el preprocesamiento y se observa que las palabras cercanas son otras *stop words*. Esta última observación es interesante, pues indica que la representación en *embeddings* junta las *stop words* que hayan, en lugar de distribuirlas entre palabras con semántica. Esto hace posible que el modelo ETM construya tópicos conformados principalmente por *stop words*, como ocurrió en nuestro caso con el tópico 6 (figura 5.1).

```
Palabra «deporte»: ['deporte', 'plantel', 'fútbol', 'cancha', 'jugadores', 'jugador', 'básquetbol', 'chicas', 'goles', 'femenino', 'hinchas',
'deportistas', 'sub', 'gol', 'jugaron', 'jugaba', 'torneo', 'minuto', 'jugó', 'pelota']

Palabra «política»: ['política', 'gobiernos', 'políticas', 'gobierno', 'debate', 'intereses', 'político', 'interna', 'sectores', 'económico',
'económica', 'públicas', 'políticos', 'social', 'críticas', 'términos', 'acuerdos', 'torno', 'oposición', 'económicos']

Palabra «salud»: ['salud', 'riesgo', 'protección', 'aplicación', 'seguimiento', 'prevención', 'casos', 'médica', 'graves', 'riesgos',
'cobertura', 'privadas', 'aumento', 'reducir', 'evaluar', 'proteger', 'reducción', 'médico', 'aplicar', 'medidas']

Palabra «educación»: ['educación', 'enseñanza', 'educativo', 'transformaciones', 'educativa', 'docentes', 'desigualdades', 'feministas',
'autonomía', 'transformación', 'desigualdad', 'estudiantil', 'equidad', 'escuelas', 'educativos', 'universitario', 'maestros', 'feminista',
'mujeres', 'primaria']

Palabra «economía»: ['economía', 'economista', 'analistas', 'económico', 'impuesto', 'exportaciones', 'impuestos', 'perspectivas',
'reformas', 'económica', 'económicas', 'petróleo', 'ganancias', 'precios', 'financiero', 'gobiernos', 'exportación', 'pobreza', 'creciente',
'mercados']

Palabra «acá»: ['acá', 'sale', 'montón', 'hicimos', 'sé', 'cosa', 'iba', 'empezar', 'íbamos', 'mueve', 'conversar', 'venir', 'gente',
'poquito', 'siente', 'gusta', 'digo', 'diga', 'quiera', 'veo']
```

Figura 5.6: Evaluación cualitativa de los *embeddings*.

5.2. Análisis cuantitativo

Las métricas cuantitativas analizadas son las siguientes:

- Error de predicción. Se calcula utilizando una función de pérdida similar a la del entrenamiento. La única diferencia con lo explicado para el entrenamiento es que en este caso cada documento de prueba se separa en dos mitades: la primera se utiliza para determinar los tópicos del documento y calcular su distribución de palabras probable, la cual se compara con la distribución real de la segunda mitad.

- Coherencia de tópicos. Cuantifica la interpretabilidad, retornando valores mayores si las palabras del tópico tienden a aparecer en mismos documentos.

- Diversidad de tópicos. Se toman las 25 palabras más relevantes de cada tópico y se calcula el porcentaje de palabras únicas (que están en un único tópico).

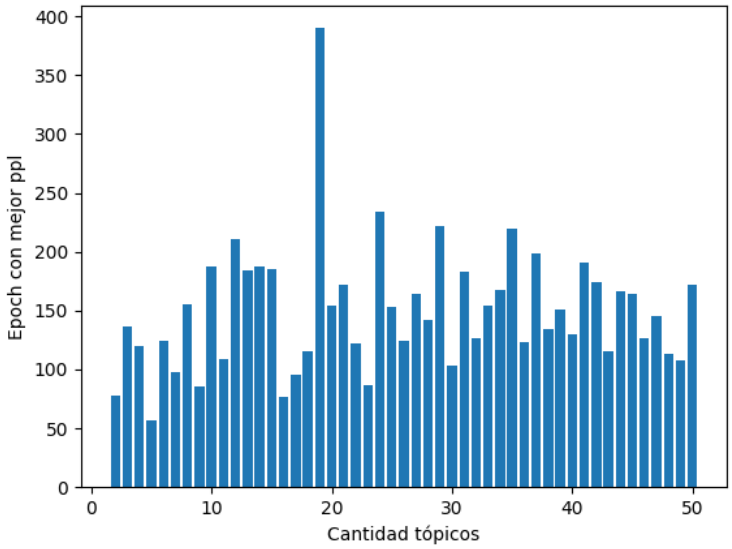


Figura 5.7: *Epoch* en el que se obtuvo el menor error de predicción antes de cumplirse el criterio de parada, para cada cantidad de tópicos.

Se entrenaron modelos de ETM con cantidad de tópicos entre 2 y 50 para evaluar la evolución de las métricas. Al entrenar los modelos es necesario fijar una cantidad de *epochs* o un criterio de parada. Para contemplar la posibilidad de que con distintas cantidades de tópicos se requieran distintas cantidades de *epochs*, se utilizó el criterio de parada del siguiente pseudocódigo:

```
cantSinMejora ← 0
while cantSinMejora < 50 do
    entrenar()                                  ▷ Un epoch de entrenamiento
    evaluar()                                   ▷ Calcular el error de predicción
    if el error de predicción no disminuyó then ▷ Respecto al mínimo previo
        cantSinMejora ← cantSinMejora + 1
    else
        cantSinMejora ← 0
    end if
end while
```

De forma resumida, se entrena hasta que el error de predicción no mejora durante 50 *epochs* consecutivos, lo cual se interpreta como que el modelo está estancado y no seguirá mejorando. En la figura 5.7 se muestran los *epochs* en los que se obtuvo el mejor resultado para cada cantidad de tópicos. Se puede observar que en la mayoría de los casos no se superan los 200 *epochs*. Inicialmente el subsistema realizaba 2000 *epochs* (teniendo en cuenta que los autores de ETM utilizaron 1000 *epochs* por defecto), sin embargo las pruebas muestran que en esta realidad basta con un número mucho más pequeño, reduciendo en un orden de magnitud al tiempo necesario. En concreto un experimento posterior mostró que en promedio cada *epoch* toma 3.72 segundos con una desviación estandar de 0.10 segundos[1].

En las figuras 5.8, 5.9 y 5.10 se muestran los valores obtenidos para las métricas en los modelos entrenados con distintas cantidades de tópicos.

En la figura 5.8 se muestra el error de predicción: el opuesto del producto escalar entre la distribución de palabras predicha por los tópicos (los cuales se obtienen de la primera mitad del documento) y la distribución real (en la segunda mitad del documento). Se puede observar que desde 2 tópicos hasta 20 hay una mejora, pero de ahí en más el error de predicción se estanca.

En la figura 5.9 se muestra la coherencia de tópicos. Se puede observar que, al igual que con el error de predicción, al comenzar a aumentar los tópicos se obtiene una mejora, pero cerca de los 20 tópicos comienza a tender a una asíntota; el error de predicción y la coherencia de tópicos presentan un comportamiento similar. Esto último es de esperarse, ya que alta coherencia de tópicos significa que palabras del mismo tópico tienden a aparecer en los mismos documentos, por lo que obteniendo los tópicos de la primera mitad de un documento, es de esperarse que la distribución de palabras inducida sea similar a la de la segunda mitad del documento (lo cual significa bajo error de predicción).

En la figura 5.10 se muestra la diversidad de tópicos. Se observa que disminuye al aumentar la cantidad de tópicos. Esto es de esperarse, pues significa que al haber más

[1]La plataforma utilizada para esta prueba es un laptop, con un procesador Intel i5 -5200U 2.20Ghz 64 bits, 16 GiB de Ram, el SO es Ubuntu 20.04.4 LTS 64 bit. Posee una GPU Nvidia Geforce 840m.

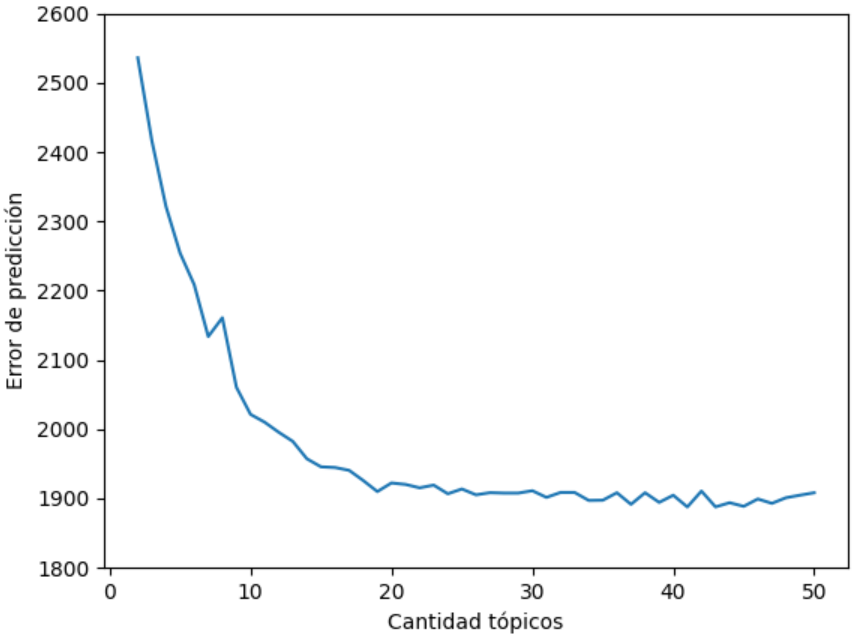


Figura 5.8: Evolución de la función de perdida de la predicción.

tópicos se repiten más palabras entre ellos.

Teniendo en cuenta los resultados observados, se concluyó que la cantidad de tópicos óptima para el presente corpus es 20. En base a esto es que para el análisis cualitativo se usó el modelo entrenado con esa cantidad de tópicos.

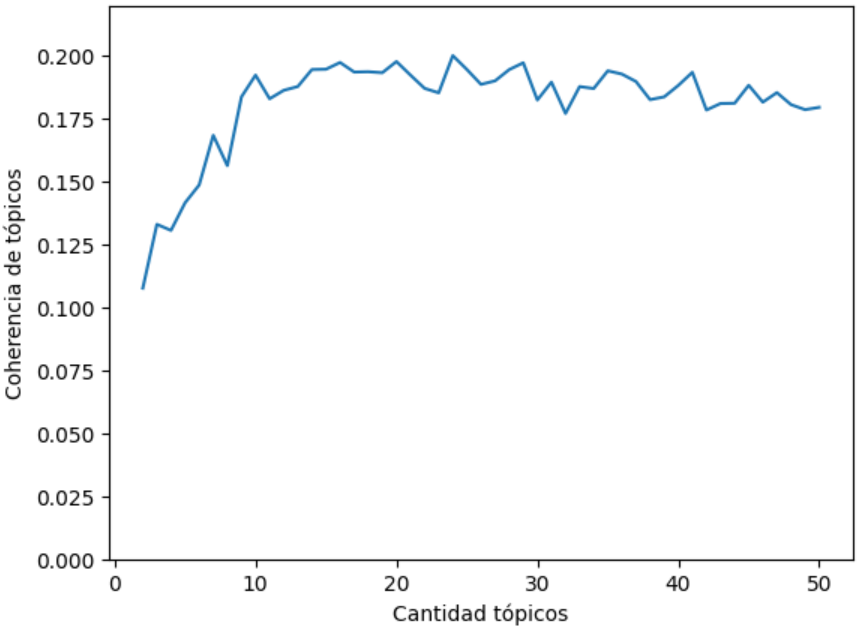


Figura 5.9: Evolución de la coherencia de tópicos.

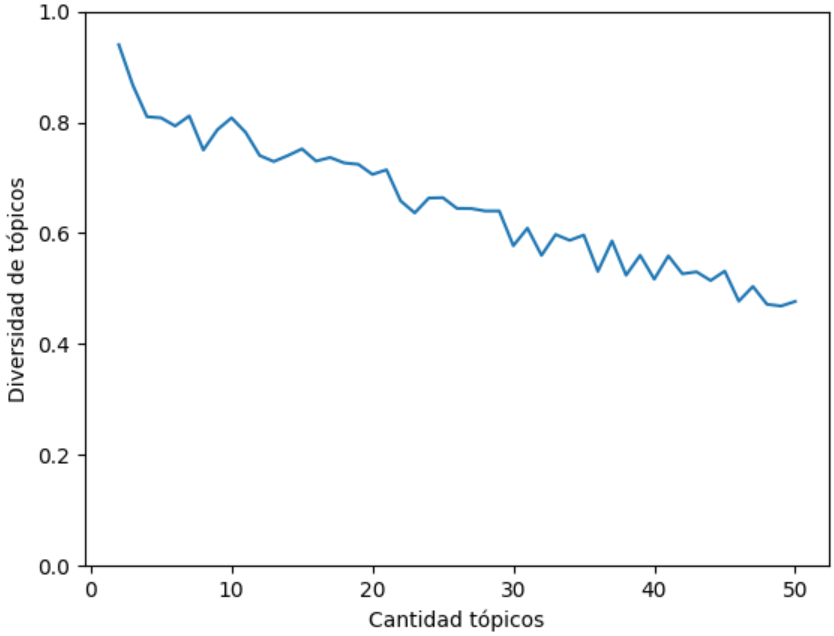


Figura 5.10: Evolución de la diversidad de tópicos.

Capítulo 6

Conclusiones

El presente proyecto de grado se enmarca en el Observatorio de Medios del Uruguay, con el fin de automatizar la detección de temas en corpus de noticias. En este sentido, este trabajo tiene una componente de investigación, al incluir estudio del estado del arte en modelado de tópicos y una componente práctica, al incluir la implementación de una aplicación web desde la que se pueda acceder a un sistema de modelado de tópicos.

Respecto a la investigación del estado del arte en modelado de tópicos, la metodología más relevante históricamente, que además sigue siendo vigente, es LDA [12]. Posteriormente, han surgido muchas variaciones de LDA que aspiran a mejorarlo. Es el caso de la metodología que se eligió utilizar: ETM [14], un enfoque reciente que enriquece LDA con *word embeddings*. Por otra parte, han surgido enfoques de modelado de tópicos que no se basan en LDA, por ejemplo utilizando *autoencoders* [18] o transformando el modelado de tópicos en un problema de detección de comunidades en redes complejas [17].

Se implementó una aplicación web desde la que se puede utilizar un sistema de ETM. Esta aplicación incluye funcionalidades básicas de manejo de noticias (agregar, modificar, visualizar y borrar), entrenamiento de modelos de ETM con corpus de noticias y clasificación automática de noticias según tópicos aprendidos por el modelo. La aplicación web no requiere conocimiento técnico sobre el funcionamiento del modelo para ser utilizada.

Se experimentó con un corpus de veinte mil noticias pertenecientes a La Diaria, llegando a conclusiones tanto cuantitativas como cualitativas.

Cuantitativamente, se analizó la evolución de métricas de desempeño al variar la cantidad de tópicos. Desde 20 tópicos en adelante los resultados no mejoran, en base a lo cual se concluyó que para el corpus dado (y con el preprocesamiento utilizado) la cantidad óptima de tópicos es 20.

Utilizando la cantidad óptima de tópicos proveniente del estudio cuantitativo, se realizó un estudio cualitativo, en el cual se observaron buenos resultados. Los tópicos tienen interpretaciones semánticas claras (deporte, entretenimiento, etc.), salvo uno que está conformado por *stop words* y palabras comunes. Esto es similar a lo reportado por los

autores de ETM [14]: el modelo es robusto a *stop words*, porque las agrupa en un tópico particular, sin que interfieran en los otros. La clasificación automática de noticias es satisfactoria: a noticias que tratan de un tema particular (por ejemplo deporte) les asigna porcentaje alto en el tópico vinculado a ese tema.

A modo de conclusión general, la tarea de identificación de temas en noticias se puede resolver satisfactoriamente mediante técnicas de Procesamiento de Lenguaje Natural. En el contexto del OMU, esto posibilita la realización de análisis globales sin necesidad de que gente realice tareas de etiquetado de muchas noticias. Nos parece claro que con los avances en aprendizaje automático y los datos disponibles, existe la posibilidad real de automatizar muchas tareas trabajosas, como la detección de temas, permitiendo que los recursos humanos centren sus esfuerzos en tareas de más alto nivel, menos repetitivas y que involucren mayor creatividad.

6.1. Trabajo futuro

Una posible extensión al sistema sería la posibilidad de utilizar *embeddings* preentrenados. Creemos que esto conllevaría mejoras en el desempeño por los siguientes motivos:

1. Disminuiría la cantidad de parámetros a ajustar durante el entrenamiento, lo cual implicaría menos tardanza. Por otra parte, con *embeddings* fijos puede que los tópicos se adapten mejor a ellos que si varían.

2. Durante la clasificación el modelo podría utilizar palabras que no estén presentes en el conjunto de entrenamiento. A modo de ejemplo, si en el conjunto de entrenamiento está la palabra «computadora» pero no la palabra «ordenador» y esta segunda aparece en una noticia a clasificar, se la podría utilizar con resultados probablemente buenos, porque los vectores de *embeddings* de ambas palabras probablemente sean cercanos y en base a ellos es que se define la pertenencia a tópicos.

Además de lo anterior, se podría utilizar un enfoque mixto, en el que se parta de *embeddings* preentrenados y se los ajuste con el corpus de entrenamiento (proceso denominado *fine tuning*).

Respecto al preprocesamiento, se ha estudiado que se pueden mejorar los resultados en modelado de tópicos al reducir los textos únicamente a sustantivos [19]. Es decir, usualmente los sustantivos son los mejores indicadores de tópicos. Esto puede ser llevado a cabo mediante el uso de etiquetado gramatical (POS-*tagging* en inglés), para identificar la categoría gramatical (sustantivo, verbo, adjetivo, etc.) de cada palabra y filtrar todas salvo los sustantivos. El etiquetado gramatical es un problema sumamente explorado, para el cual hoy en día existen varias soluciones. Otro posible experimento a realizar vinculado al preprocesamiento sería modificar los límites de frecuencia para el filtrado

de palabras y verificar las diferencias en los modelos resultantes. En particular, sería interesante evaluar si disminuyendo la mínima frecuencia aceptada el modelo aprende tópicos más específicos.

Por otra parte, existe la posibilidad de experimentar con el uso de ETM para detectar subtópicos. Dado que el modelo no dio buenos resultados con grandes cantidades de tópicos (a partir de los 20 no hay mejora en el error de predicción), una posible estrategia sería mantener esta cantidad en 20 pero restringir el corpus a las noticias con cierto tópico en particular. A modo de ejemplo, se podría restringir el corpus a noticias que contienen el tópico «Deporte» y evaluar si el sistema diferencia fútbol de baloncesto o remo. Previamente a poseer el corpus de veinte mil noticias se trabajó con un corpus conformado solamente por noticias vinculadas al covid[1]. El sistema infirió algunos tópicos cualitativamente buenos, correspondientes a educación, deporte y trabajo dentro del contexto del covid. Esto puede indicar que la estrategia mencionada para detectar subtópicos daría buenos resultados.

Considerando la aplicación web, un posible agregado sería un sistema de gestión de usuarios. Por otra parte, se podría agregar soporte para múltiples modelos de ETM, de modo que al reinferir tópicos el modelo anterior no se descarte. Esto implicaría enriquecer la estructura de la base de datos para diferenciar los tópicos de los distintos modelos y agregar a la aplicación web la funcionalidad de elegir uno de varios modelos disponibles. Además cabe la posibilidad de dotar a la aplicación web con una serie de herramientas para facilitar datos estadísticos con respecto a los tópicos de los documentos y sus metadatos (título, medio, relevancia, género y tópico asignado por el medio). Esto puede ser de gran utilidad en investigaciones sobre cuales tópicos están más presentes en cada medio. También podría aplicarse para etiquetar los tópicos en caso de que exista una correlación entre un tópico del subsistema y el asignado por el medio. Para esta última tarea también puede ser beneficioso que al visualizar la información de un tópico, se despliegue a su vez una serie de noticias de gran relevancia en dicho tópico, lo cual facilitaría entender las palabras del tópico en su contexto.

[1]Disponible en `https://github.com/pln-fing-udelar/covid19-qa`.

Bibliografía

[1] Maxwell McCombs and Dixie Evatt. Los temas y los aspectos: explorando una nueva dimensión de la agenda setting. *Communication & society*, 8(1):7–32, 1995.

[2] Manuel Bartolomé-Castro. Teresa sádaba. framing: el encuadre de las noticias. el binomio terrorismo-medios. la crujía, buenos aires, 2008, 252 pp. *Communication & Society*, 22(1), 2009.

[3] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA, 2009.

[4] Tomas M. Mitchell. *Machine Learning, volumen 1.* McGraw-Hill, 1997.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[6] Steven Abney. *Semisupervised learning for computational linguistics*. Chapman and Hall/CRC, 2007.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[8] Dongwoo Kim and Alice Oh. Topic chains for understanding a news corpus. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 163–176. Springer, 2011.

[9] Xi Liu, Yongfeng Yin, Haifeng Li, Jiabin Chen, Chang Liu, Shengli Wang, and Rui Yin. Intelligent radar software defect classification approach based on the latent dirichlet allocation topic model. *EURASIP Journal on Advances in Signal Processing*, 2021(1):1–20, 2021.

[10] Matthew P Peters, Steve N Matthews, Anantha M Prasad, and Louis R Iverson. Defining landscape-level forest types: application of latent dirichlet allocation to species distribution models. *Landscape Ecology*, pages 1–19, 2022.

[11] Fatemeh Torabi Asr, Mohammad Mazraeh, Alexandre Lopes, Vasundhara Gautam, Junette Gonzales, Prashanth Rao, and Maite Taboada. The gender gap tracker: Using natural language processing to measure gender bias in media. *PloS one*, 16(1):e0245533, 2021.

[12] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[13] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[14] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 2020.

[15] Martin Gerlach, Tiago P Peixoto, and Eduardo G Altmann. A network approach to topic models. *Science advances*, 4(7):eaaq1360, 2018.

[16] GK Zipf. The psycho-biology of language, routledge, london. 1936.

[17] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[18] Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. Topic modeling with wasserstein autoencoders. *arXiv preprint arXiv:1907.12374*, 2019.

[19] Fiona Martin and Mark Johnson. More efficient topic modelling through a noun only approach. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 111–115, 2015.

Glosario

- **Bag of words**: representación de textos en la que no se considera el orden de las palabras.
- **Corpus**: Conjunto de textos.
- **Embeddings**: Representación vectorial densa de palabras construida en base a contexto.
- **Epoch**: Un *epoch* es una recorrida por todos los datos de entrenamiento. En algunas aplicaciones de aprendizaje automático (como las que incluyen redes neuronales) es común que se realicen varios *epochs* durante el entrenamiento.
- **ETM**: *Embedded Topic Modeling*. Metodología de modelado de tópicos, basada en LDA y *embeddings*, introducida por Dieng et al. en 2020 [14].
- **Descenso por gradiente**: Herramienta de optimización que consiste en buscar el mínimo de una función diferenciable desplazándose en la dirección opuesta al gradiente. Se suele utilizar en aprendizaje automático para entrenar minimizando una función de error.
- **FIC**: Facultad de Información y Comunicación de la UdelaR.
- **LDA**: *Latent Dirichlet Allocation*. Metodología de modelado de tópicos introducida por Blei en 2003 [12].
- **Lenguaje natural**: Lenguajes que usamos las personas para comunicarnos, como por ejemplo: español, inglés y guaraní.
- **Metadatos**: Datos que describen otros datos. Por ejemplo el título y el nombre del autor de un libro.
- **Modelado de tópicos**: Disciplina del PLN que busca determinar los tópicos subyacentes a un conjunto de documentos.
- **OMU**: Observatorio de Medios del Uruguay.
- **PLN**: Procesamiento de Lenguaje Natural. Campo interdisciplinario que apunta a realizar automáticamente tareas útiles que involucren lenguaje humano.
- **Preprocesamiento**: En el contexto de una aplicación de aprendizaje automático, el preprocesamiento es la tarea de llevar los datos al formato necesario para poder aplicar el algoritmo.
- **Python**: Lenguaje de programación interpretado con bibliotecas disponibles para diversas aplicaciones.

- **pytorch**: Biblioteca de python para aprendizaje automático, especialmente con redes neuronales.
- **Red neuronal**: Modelo que alterna entre transformaciones lineales y funciones no lineales, lo cual le otorga gran expresividad.
- **Stop word**: Palabra que se usa frecuentemente y no suele aportar información para la mayoría de las tareas de PLN. Los determinantes conforman un ejemplo.
- **UdelaR**: Universidad de la República.