UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA

Análisis de noticias sobre seguridad ciudadana en redes sociales

Proyecto de Grado

Ingeniería en computación

Leandro Dominguez

Guillermo Eijo

Sebastian Felix

Agosto 2022

Montevideo Uruguay

Tutores: Aiala Rosá, Guillermo Moncecchi

Resumen

Los medios de comunicación tienen una fuerte injerencia en la opinión de las personas. Hoy en día, estos utilizan cada vez más la red social Twitter como medio de difusión de noticias. Según Latinobarómetro, la seguridad ciudadana es el tema que más preocupa a la sociedad uruguaya desde el 2006.

En función de esto, el presente trabajo busca generar una herramienta que permita a cualquier persona realizar un seguimiento de la temática de Seguridad, a través del análisis de tweets que publican diversos medios de prensa escrita.

Se trabajó en conjunto con investigadores y estudiantes de Facultad de Ciencias Sociales para etiquetar más de dos mil de esos tweets, que luego se utilizaron para entrenar un modelo de aprendizaje automático que identificara los que tratan sobre seguridad. Para representar los tweets se realizaron varias pruebas en base al algoritmo autosupervisado Word2Vec, pruebas con la variante simple y la variante enriquecida con subpalabras, y también con dos implementaciones diferentes: Skipgram y Continuous Bag of Words (CBOW).

Se implementaron herramientas para la detección de tópicos y entidades nombradas. Para la detección de tópicos, se utilizó una mezcla de varias técnicas, empleando un enfoque de aprendizaje no supervisado para agrupar las noticias haciendo uso de su representación vectorial. A esta representación vectorial se le aplica el algoritmo de k-means para detectar agrupaciones semánticas. Dentro de estas se utiliza el algoritmo Latent Dirichlet Allocation (LDA) para detectar tópicos formados por conjuntos de diez palabras.

Para la detección de entidades nombradas, se realizaron pruebas con dos implementaciones: Stanza y Spacy. Ambas son bibliotecas utilizadas en el área para tareas de PLN. Luego se utilizó un conjunto previamente etiquetado para comparar los resultados de cada implementación.

Se desplegó una aplicación que permite visualizar todos los datos y navegar con distintos filtros. Por otro lado, existe un script de Python encargado de descargar los nuevos tweets publicados, procesarlos y actualizar la base de datos con la nueva información.

Además de la tarea de implementación, el presente trabajo requirió contacto estrecho con el beneficiario del producto, teniendo reuniones periódicas donde se pactaron funcionalidades según necesidades y tiempo disponible.

Palabras clave: Aprendizaje Automático, Aprendizaje supervisado, Aprendizaje no supervisado, Procesamiento de lenguaje Natural, Twitter, Red Social, Seguridad, Clasificación.

Índice general

Capítulo 1

Introducción

Los medios de comunicación se presentan hoy en día como una de las partes esenciales para establecer qué temas están en la agenda así como para formar opinión en los ciudadanos. Dependerá de cómo se enmarque la noticia, así como a quiénes se le dé voz (visibilidad), la opinión que se querrá promover.

En consecuencia, surge la necesidad del desarrollo de una herramienta que permita tener en un lugar centralizado toda esta información, facilitando su acceso a los distintos actores que trabajan estudiando temas de agenda y formación de opinión.

A través de las redes sociales se puede acceder a gran variedad de información sobre noticias. En plataformas como Twitter[1], los medios narran sucesos inmediatamente luego de ocurridos o incluso mientras están ocurriendo.

Por último, la seguridad ciudadana es el tema que más preocupa a los uruguayos constantemente desde 2006 según Latinbarómetro 2019.

El presente trabajo se desarrolla en el marco institucional para el egreso en la carrera de Ingeniería en Computación y se enmarca también en un proyecto CSIC denominado *"De la ley de seguridad ciudadana (1995) a la ley de urgente consideración (2020): análisis de las agendas de seguridad durante los últimos 25 años en Uruguay"*. En el mismo participan, por un lado, el Departamento de Sociología de la Facultad de Ciencias Sociales de la Universidad de la República y, por otra parte, Colectivo Catalejo[2]. Este colectivo, se define en su sitio web como una organización *"sin fines de lucro que busca visibilizar temáticas sociales que afectan a la población así como enriquecer el debate ciudadano y promover la participación social y cultural"* [3].

[1] https://twitter.com/
[2] https://colectivocatalejo.org/
[3] https://colectivocatalejo.org/quienes-somos/

1.1. Objetivos

El objetivo general del presente proyecto consiste en generar e instalar una herramienta para el procesamiento y seguimiento de datos periodísticos relacionados a la temática de seguridad. De esta forma, se tendrá un espacio de monitoreo de las discusiones sobre seguridad en redes sociales, conociendo los principales temas y actores protagonistas. Esto se hará en conjunto con la parte involucrada de Ciencias Sociales, estudiando las necesidades de la herramienta a construir.

Como objetivos específicos se identificaron los siguientes:

- Analizar las publicaciones que diferentes medios de prensa uruguayos hacen diariamente en la red social Twitter, con el fin de identificar las que tengan alguna relación con temas de seguridad pública.
- Trabajar con datos reales y generar información de utilidad para un proyecto mayor, con usuarios interesados en trabajar con estos datos.
- Obtener un conjunto de tweets publicados por medios de prensa uruguaya a lo largo de varios años e identificar los relacionados con temas de seguridad ciudadana.
- Desarrollar una herramienta que permita visualizar la información de las publicaciones de medios de prensa de forma ordenada.

1.2. Organización del documento

El presente informe se divide en 7 capítulos, seguidos de la bibliografía, anexos y un glosario. El capítulo dos presenta el marco teórico del trabajo, incluyendo conceptos que fueron necesarios comprender para trabajar con ellos en capítulos posteriores. El capítulo tres presenta una descripción general de la solución propuesta al problema, explicando de forma no exhaustiva las partes del sistema construido. El capítulo cuatro explica la construcción del conjunto de datos que se utilizó a lo largo del trabajo (para entrenar modelos y visualizar información). El capítulo cinco explica detalles de la solución construida, la plataforma auxiliar que fue utilizada para clasificar noticias, así como la selección y pruebas que se hicieron con distintos modelos. El capítulo seis muestra un caso de uso concreto con información relevante que se podría extraer utilizando varias de las funcionalidades para mostrar el potencial de la aplicación. En el séptimo y último capítulo se describen los resultados concretos obtenidos al culminar el trabajo final de grado así como conclusiones finales y algunos puntos que quedaron fuera del alcance por distintas razones y complementarían o mejorarían de alguna forma el proyecto con su implementación futura.

Capítulo 2

Marco Teórico

Como se expresó en el capítulo anterior, un objetivo del proyecto es analizar la publicaciones con temática de seguridad ciudadana de ciertos medios de prensa. Estas tareas se abordaron con técnicas de Aprendizaje Automático (AA) y herramientas de Procesamiento de Lenguaje Natural (PNL). Basándose en el libro *Speech and Language Processing* de Daniel Jurafsky y James H. Martin [1] se presentan estas técnicas, detalles de ellas, y otros conceptos que permitirán un mayor entendimiento del trabajo realizado.

2.1. Aprendizaje automático

El aprendizaje automático consiste en desarrollar diferentes técnicas que permitan que una computadora tenga la capacidad de mejorar su desempeño con la experiencia y la utilización de datos, o sea, aprender. Según Tom Mitchell *"Se dice que un programa de computadora aprende de la experiencia* ***E*** *con respecto a alguna clase de tareas* ***T*** *y medida de desempeño* ***P****, si su desempeño en las tareas* ***T****, medido por* ***P****, mejora con la experiencia* ***E****"* [2]. Las técnicas de AA se pueden clasificar principalmente en 2 grandes grupos según el tipo de experiencia que utilizan: las técnicas supervisadas y las técnicas no supervisadas. También puede considerarse un tercer grupo llamado aprendizaje autosupervisado que incluye características de ambos.

Aprendizaje supervisado

Las técnicas de aprendizaje supervisado parten de un conjunto de datos en los que se tiene un conjunto de entrada X y sus correspondientes salidas (o etiquetas) Y. Esta relación entre los datos de entrada y sus correspondientes salidas es anotada por un "supervisor", que puede ser una o más personas que tienen el conocimiento adquirido para poder determinar, según esas entradas, qué salida corresponde. Posteriormente, con la

información obtenida se entrena un modelo que, dada una entrada, va a determinar una salida a partir de los conocimientos incorporados desde el conjunto de datos etiquetados. Cuando la salida es continua dentro de un rango se le llama *regresión*. Por otro lado, cuando la salida es discreta, se le llama *clasificación*.

Aprendizaje no supervisado

Para estas técnicas, a diferencia de las supervisadas, no se cuenta con un conjunto previamente etiquetado, sino que se parte sólo del conjunto de entrada sin tener un conocimiento de cuál debería ser la salida. La idea de utilizar esta técnica es poder encontrar relaciones entre los datos de entrada y determinar características similares entre ellos. Un ejemplo muy común de aprendizaje no supervisado es el *clustering*, del que también se hablará en más detalle avanzado el capítulo.

Aprendizaje autosupervisado

El aprendizaje autosupervisado combina un poco de las dos técnicas antes mencionadas. Utiliza un conjunto de datos sin etiquetar, pero obtiene información útil que proporciona algún tipo de supervisión. Un caso común es utilizar esta técnica para la predicción de palabras en un texto. Allí, se entrena a partir del conjunto de datos de texto y el modelo aprende cuál o cuáles son las palabras con mayor probabilidad a ser siguientes dado un contexto de palabras.

Para clasificar las publicaciones entre las que hablan sobre Seguridad Ciudadana y las que no, se utilizó la técnica de clasificación llamada *regresión logística*. Es por ello que se describe a continuación.

2.1.1. Regresión logística

Como ya se mencionó anteriormente, los problemas de clasificación son un tipo de aprendizaje supervisado y tienen la característica de que su salida es un valor discreto [1]. Un algoritmo clásico de clasificación es la regresión logística. El objetivo es poder clasificar los datos en una de dos clases. Lo que se mide con la regresión logística es la relación entre una afirmación a predecir (variable dependiente) y el conjunto de características disponibles para el modelo (variables independientes). Para poder realizar esto, se utiliza la función sigmoidea o logística, que es la que determina la probabilidad de la variable dependiente según las variables independientes evaluadas.

La ecuación que define la función sigmoidea es:

$$f(x) = \frac{1}{1 + e^{-x}}$$

y la representación gráfica se muestra en la Figura 2.1.

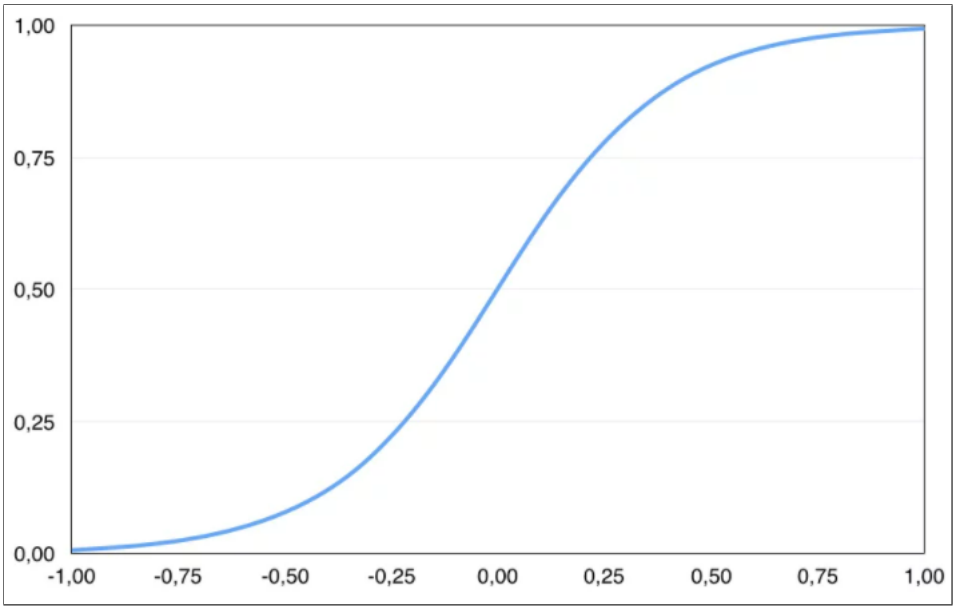

Figura 2.1: Gráfica de la función sigmoidea

Luego de evaluar la función y obtener la probabilidad correspondiente, se utiliza un umbral para poder realizar la clasificación: si la probabilidad es mayor a dicho umbral se clasifica como verdadero (o clase 1) y en caso contrario se clasificaría como falso (o clase 2).

Al momento de trabajar con problemas de clasificación, es necesario tener algún método que permita visualizar, comparar y evaluar el desempeño de los modelos. Es fundamental introducir el concepto de *matriz de confusión* para comprender con facilidad dichas métricas.

Matriz de Confusión

Antes de pasar a definir las diferentes medidas, es necesario definir y entender qué es una Matriz de Confusión. La misma permite clasificar los resultados para poder evaluar posteriormente el modelo. En los casos de clasificación binaria, es una matriz de 2x2 y se construye colocando en las filas los valores de predicción, Verdadero o Falso y en las columnas los valores reales, también Verdadero o Falso, quedando definidos 4 grupos (Figura 2.2).

De esta forma, quedan en el primer cuadrante la cantidad de Verdaderos Positivos

(VP), predicciones verdaderas con resultado esperado verdadero, en el segundo cuadrante la cantidad de Falsos Positivos (FP) que son aquellas predicciones Verdaderas que tenían un resultado falso esperado, en el tercer cuadrante se encuentra la cantidad de predicciones Falsos Negativos (FN) predicciones falsas con resultado esperado verdadero y por último, en el cuarto cuadrante, el número de predicciones Verdaderos Negativos (VN) donde la predicción y el resultado esperado son falso.

Figura 2.2: Matriz de confusión

Una vez que se tiene completa la matriz de confusión, es sencillo definir las métricas utilizadas para evaluar nuestro modelo.

Precision

Es la cantidad de instancias positivas sobre la cantidad de instancias clasificadas como positivas. Se calcula dividiendo la cantidad de VP sobre la suma de VP y FP:

$$Precision = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

Un valor cercano a 1 en la precisión indica que nuestro modelo detecta de una buena forma los casos verdaderos, pero no dice nada de los casos que no se han podido detectar.

Recall

Es la fracción de instancias positivas que fueron clasificadas sobre el total de instancias positivas. Se calcula dividiendo VP sobre la suma de VP y FN

$$Recall = \frac{\text{VP}}{\text{VP} + \text{FN}}$$

Este cociente viene a expresar la proporción de instancias positivas recuperadas, comparado con el total de los documentos que son positivos del total. No indica nada sobre la cantidad de casos clasificados por error.

Medida F

La media armónica o medida F, es una medida que combina el recall y la precisión, permitiendo tener una medida más general de la calidad del modelo, y ayudando en gran medida a mitigar las diferencias entre ambas medidas.

$$F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Accuracy

Es el porcentaje de acierto o la cantidad de predicciones correctas sobre el total de predicciones. Se calcula haciendo la suma de VP y VN, sobre el total de predicciones.

$$Accuracy = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}}$$

2.1.2. Clustering

Cuando se trabaja con conjuntos de textos, un análisis deseable es poder agruparlos según temática. Para esto pueden utilizarse las técnicas de clustering. Como ya se mencionó, es una técnica de aprendizaje no supervisado que agrupa objetos por similitud. Lo hace de manera que los miembros del mismo grupo tengan características similares [1]. A cada conjunto de datos se le llama *cluster*.

Existen distintas formas de aplicar *clustering*, una de ellas (y la que se utiliza en este proyecto) es la técnica de *K-Means*. Para utilizar esta técnica es necesario definir la cantidad de *clusters* en los que se agruparán los datos. Este número depende del objetivo que se tenga así como la cantidad de datos a procesar.

El algoritmo K-means [3] comienza definiendo k centroides, que van a ir modificándose de forma iterativa hasta alcanzar un punto de equilibrio. El primer paso es elegir los centroides de una forma aleatoria, luego comienzan las iteraciones:

1. Se calcula la distancia de cada punto a cada centroide.
2. Se asigna a cada punto el cluster cuyo centroide esté más cercano.
3. Se toman todos los clusters y se recalculan los centroides. Para cada cluster, se toman todos los puntos que pertenecen a él y se calcula el punto medio, que puede o no coincidir con un punto del dato.

4. Con los nuevos centroides se vuelve al punto 1.

Estos pasos se repetirán hasta que se alcance un estado estable, entendiendo esto como el momento en el que no cambie ningún centroide o los puntos no cambien de *cluster*.

2.2. Procesamiento de lenguaje natural

El PLN intenta resolver con computadoras tareas vinculadas al lenguaje humano, logrando una comunicación entre el humano y la computadora utilizando algún lenguaje natural [1]. Al trabajar con PLN es común pasar por varias etapas como ser: Tokenización que consiste en fragmentar el texto en unidades pequeñas llamadas tokens, Lematización o Stemming que llevan las palabras a una versión más reducida o canónica de ellas, Limpieza de datos que consiste en quitar palabras que no aportan información relevante. Una definición más completa de cada etapa se puede encontrar en el glosario. Estas etapas fueron necesarias para las técnicas de PLN utilizadas en la investigación.

Para detectar sobre qué personas se habla en un tweet o en qué lugar sucedió una noticia, se utilizaron técnicas de detección de entidades nombradas que se describen a continuación.

Detección de entidades nombradas

La detección de entidades nombradas o Reconocimiento de Entidades con Nombre [4] (NER por sus siglas en inglés) es el proceso de reconocer y clasificar palabras o frases que refieren a objetos que poseen un nombre propio. Una persona, un lugar o una organización son ejemplos de entidades nombradas. Estas entidades se clasifican según categorías predefinidas: nombres de personas, organizaciones, localidades o lugares, expresiones de tiempo, etc.

Detección de tópicos

La detección de tópicos consiste en encontrar temas o tópicos dentro de un texto. Para esto se buscan similitudes entre las palabras que pertenecen a los textos dentro del corpus. Normalmente los tópicos se definen como una función de densidad $\pi(w)$, que representa la probabilidad de la palabra w en el tópico i. Los modelos que permiten clasificar documentos que no han sido procesados cuando el modelo ya fue entrenado llevan el nombre de *topic modelling generativo*. Particularmente se utilizó Latent Dirichlet Allocation [5] (LDA). LDA es un modelo de tópicos generativos que asume que cada palabra dentro de un documento es generada a partir de un tópico que es tomado de una distribución de tópicos para cada documento y esta distribución es generada a partir

de una distribución de Dirichlet [6], permitiendo que un documento sea parte de varios tópicos, cada uno con peso diferente. [7]

2.2.1. Word embeddings

Para poder comparar y buscar relaciones entre palabras, es necesario llevarlas a una representación que sea sencilla de procesar por una computadora. Para esto en el PLN se suelen utilizar las *word embedding* que consisten en representar las palabras mediante vectores numéricos permitiendo encontrar relaciones entre ellas [1]. Se basa en la hipótesis distribucional de Zellin Harris, que plantea que palabras que se encuentran en contexto similar tienen significados cercanos. Los vectores suelen tener centenas de dimensiones y se construyen según el contexto de la palabra en el corpus utilizado. De esta forma y teniendo en cuenta lo dicho por Harris [8] , dos palabras con similares contextos van a tener vectores cercanos. El método utilizado en este trabajo es *word2vec*.

Word2Vec

Este algoritmo representa cada palabra diferente con un vector. Los vectores son elegidos de tal manera que se pueda evaluar la similitud entre dos palabras de forma sencilla. Una forma de evaluar la similitud entre vectores es utilizando la distancia coseno. Esta distancia representa geométricamente el coseno del ángulo entre dos vectores. Word2Vec aprende relaciones entre palabras partiendo de un gran conjunto de datos y utilizando una red neuronal. Existen principalmente dos arquitecturas de Word2Vec: *Skip-gram* y *Continuous Bag of Words* (CBOW) (Figura 2.3). Skip-gram intenta predecir el contexto a partir de una palabra, mientras que CBOW intenta predecir la palabra central partiendo del contexto [9].

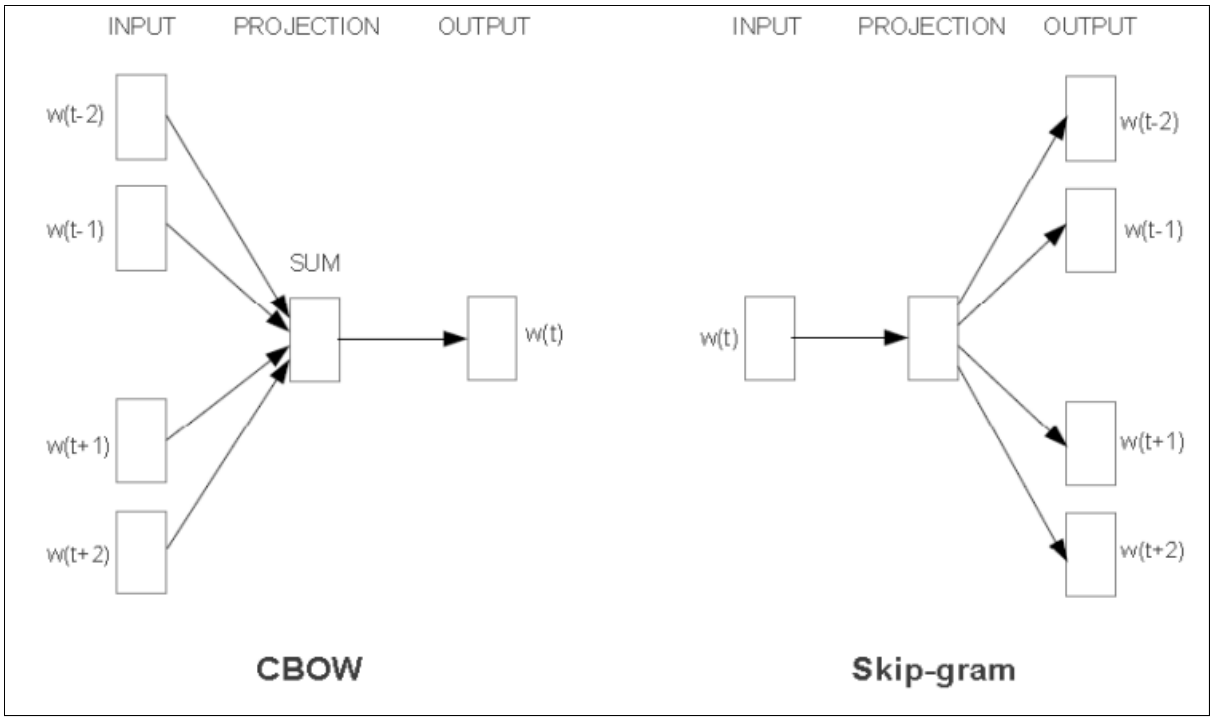

Figura 2.3: Esquema de las arquitecturas de CBOW y Skip-gram

Dado que los word embeddings en los contextos del presente problema suelen tener dimensiones altas, es necesario utilizar técnicas que las reduzcan para visualizar el conjunto en un plano. Existen varias que permiten hacer esto conservando propiedades del espacio original. Para nuestro trabajo utilizamos PCA y t-SNE y se describen en el glosario.

Para cerrar la sección de word embeddings, se define una mejora a la implementación de Word2Vec que también se utiliza en la investigación. La misma se llama *Word2Vec enrriquecido con subpalabras* [10] y consiste en representar cada palabra como un n-grama de caracteres. Estos n-gramas son asociados a una representación vectorial y luego las palabras son representadas como la suma de estos vectores. Esto permite tener en cuenta la morfología de las palabras al momento de buscar parámetros en común o relaciones entre ellas.

Con la información adquirida luego de leer esta sección, el lector será capaz de comprender las técnicas utilizadas en el desarrollo del proyecto. Estas serán mencionadas al explicar en detalle el trabajo realizado en los sucesivos capítulos.

2.3. Trabajo relacionado

Al revisar el estado del arte, no se encontraron trabajos que trataran la seguridad pública en base a información de Twitter, pero sí fueron hallados artículos que sirvieron para tener una idea de cómo encarar el proyecto. Estos trabajos consultados tratan sobre

la investigación de datos en Twitter y la utilización de técnicas de PLN para su análisis. También se consultó una herramienta utilizada actualmente por el equipo de Facultad de Ciencias Sociales cuyas funcionalidades sirvieron de ejemplo a implementar.

En [11] se propone una serie de técnicas relativamente simples para clasificación de textos y análisis de sentimientos, implementaciones eficientes que ahorran tiempo de cómputo y arrojan buenos resultados con cantidades de texto razonables. Utilizan el algoritmo word2vec para representar palabras y el promedio de las palabras que componen a una oración para representarla. Una vez obtenida la representación vectorial de la oración, propone utilizar una regresión logística para implementar un clasificador.

En [12] se proponen distintas estrategias para la Extracción de Entidades Nombradas de datos extraídos de Twitter así como posibles preprocesamientos de texto mostrando y comparando distintos desempeños. Desarrollan un sistema de detección de entidades nombradas que mejora el que utiliza Stanford (Stanford NER System) en el contexto de tweets. Como es de esperar, describe el complejo trabajo de detectar entidades en textos introducidos por usuarios donde, entre otros desafíos, no siempre se respetan mayúsculas. Además tiene en cuenta variaciones léxicas de palabras a la hora de etiquetar las categorías gramaticales. Todo este trabajo está hecho sobre conjuntos de tweets anotados manualmente para entrenar distintas partes del algoritmo frente a tales variaciones léxicas, de forma que la mayoría de las deformaciones propias del idioma inglés escrito sean tenidas en cuenta. Si bien el trabajo es muy interesante y tiene código disponible en un repositorio, es para el idioma inglés.

En [13] propone una implementación similar a Word2Vec para representar tweets como vectores utilizando redes neuronales entrenadas sobre tres millones de tweets en inglés tomados aleatoriamente de la red social. Algo interesante que plantea este trabajo es la utilización de mecanismos de aumento de datos para la generación de ejemplos de texto con *typos* o caracteres no vistos en el ejemplo de entrenamiento. Esto último resulta importante ya que las redes neuronales utilizadas se entrenan a nivel de caracteres en lugar de palabras. Una vez entrenado el modelo que codifica tweets en word embeddings, evalúan el modelo mediante dos tareas de clasificación: clasificación de tweets en sentimientos y vinculación semántica entre tweets.

El paper [14] es un trabajo en el que se detectan tópicos en textos del tipo microblog (como Twitter) y compara distintas implementaciones de LDA en textos con estas características. Según se explica en el articulo, las características de estos textos son complejas para este tipo de algoritmos por ser textos cortos y a raíz de esto compartir pocas palabras entre sí. Los experimentos se hacen en un dataset de más de 100 mil tweets recuperados con la técnica de web crawler.

Por último, el artículo [15] utiliza la técnica de clusterización *bisecting k-means* como mejora de k-means para encontrar en los espacios de vectores de word embeddings acumulaciones semánticas y busca en estas agrupaciones temas en común. Además, propone una distancia alternativa a la similitud coseno que se comporta mejor en este tipo de tareas. Evalúa los resultados utilizando artículos de Wikipedia y sus respectivas categorías.

En el siguiente capítulo se plantea la solución propuesta describiendo las distintas etapas por las que se transcurrió durante el desarrollo del proyecto.

Capítulo 3

Descripción general de la solución

En la presente sección se describe la solución construida así como los pasos seguidos para construir las soluciones a los subproblemas en los que se dividió el trabajo. Como producto final, se obtiene un prototipo de una plataforma que permite analizar noticias extraídas de tweets de las cuentas oficiales de medios de comunicación locales. Para esta primera prueba de concepto fueron seleccionados El País, El Observador, La Diaria, La República, Brecha y Búsqueda.

Para el estudio de las publicaciones se construyeron varias herramientas y luego se sistematizaron. A esto se le agregó un módulo que recolecta diariamente las noticias y derivó en una aplicación web[4].

Las funcionalidades de esta plataforma fueron diseñadas teniendo en cuenta las necesidades de los sociólogos interesados en analizar noticias. Algunas de ellas, que se describirán en esta sección, son: posibilidad de detectar los principales hechos noticiosos a través de detección de tópicos, detección de lugares, detección de personas y detección de organizaciones involucradas, así como la posibilidad de clasificar si una noticia habla o no sobre seguridad ciudadana. Por otro lado, se cuenta con una sección de estadísticas generales sobre las cuentas y sus tweets, que indican la cantidad de publicaciones en un cierto período, así como la proporción de ellas que refieren al tema seguridad ciudadana.

Para la realización de las tareas descriptas, se trabajó sobre un corpus recolectado con la estrategia de Scraping, y para decidir si uno de estos tweets hablaba o no sobre seguridad ciudadana, se debió implementar un clasificador de textos.

El clasificador de textos implicó la implementación de una plataforma de anotación manual, donde se cargaron las publicaciones para que los usuarios pusieran etiquetas de "Seguridad" o "No seguridad" a cada uno de ellos. A partir del conjunto de datos anotados mediante la plataforma se construyó un modelo que intenta predecir si una noticia que no

[4]http://198.211.117.226/

fue etiquetada habla o no sobre seguridad ciudadana.

A continuación se describen con mayor detalle las funcionalidades que se mencionaron, cuya implementación se detalla en la sección 5. Para tener una idea de los flujos de información se presenta el diagrama de la Figura 3.1.

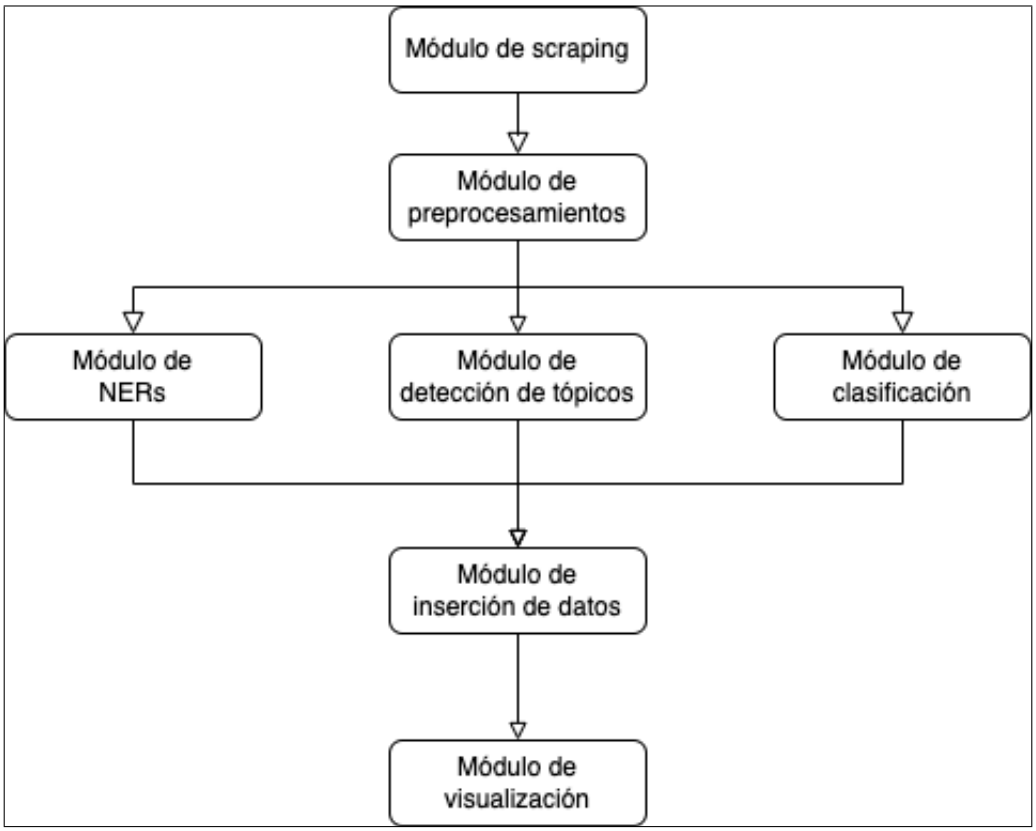

Figura 3.1: Diagrama de flujos de información

Primeramente, se construyó un módulo de scraping que es el encargado de recolectar los tweets de las cuentas de Twitter de los medios previamente detallados.

Luego de ser recolectadas las noticias, se hacen varios preprocesamientos para que puedan ser utilizadas como entrada por los siguientes módulos, de la manera ideal para cada uno (clasificación de seguridad, detección de tópicos, detección de entidades nombradas y estadísticas generales), así como para obtener información estructurada como la cantidad de likes, comentarios, entre otros.

Una vez preprocesado el texto por el módulo anterior, se pasan a detectar nombres propios, lugares y organizaciones. Las entidades serán contadas y se mostrarán las más recurrentes en el período de fechas seleccionado.

Por otro lado se cuenta con un módulo de detección de tópicos. Este se encarga de agrupar noticias según similitud semántica, y detectar dentro de cada grupo los principales tópicos de los que se habla. Como resultado se obtienen varias listas de palabras que conforman los distintos tópicos tratados.

Para que los modelos de aprendizaje supervisado aprendan a clasificar fragmentos de texto que nunca vieron, es necesario contar con ejemplos previamente etiquetados por un humano. Es por esto que se construyó una plataforma que les presenta de forma aleatoria tweets a los usuarios y estos deben indicar si se trata o no de hechos noticiosos que entran dentro del tópico de seguridad ciudadana (añadiendo etiquetas de *seguridad* o *no_seguridad*). Con esta información se entrenó un modelo de representación vectorial de textos que serán la entrada del módulo de clasificación. Se probaron distintos modelos de word embeddings y el más efectivo fue elegido según métricas que se describen en la sección de implementación. Diariamente se reciben noticias y se clasifican con las etiquetas de `seguridad` o `no_seguridad` utilizando el modelo entrenado que permite generalizar propiedades a partir de las noticias previamente clasificadas.

Una vez calculada la etiqueta de seguridad y la información relacionada a entidades nombradas, se ingresa la información en el sistema en las correspondientes bases de datos dependiendo de su finalidad. De esta forma, se logra la persistencia de información independiente de la plataforma Twitter, logrando tener una base de información histórica de fácil consulta.

Una vez calculada y almacenada la información en las correspondientes bases de datos, se construye el módulo de visualización para presentarla de forma amigable y sintetizada para el usuario.

Existen principalmente tres grandes módulos: estadísticas generales, entidades nombradas y clustering y detección de tópicos. Los tres módulos aceptan filtros de fechas, medios, palabras clave y clasificación de seguridad. Además, se muestran los porcentajes totales de noticias de seguridad.

Módulo de estadísticas generales

En este módulo se presenta información sobre cantidad de noticias de seguridad vs no seguridad según los distintos medios, aporte al total de tweets sobre seguridad por cada uno, y evolución de la proporción de tweets de seguridad a lo largo del tiempo en el período seleccionado.

Módulo de entidades nombradas

En este módulo se realizan tareas de procesamiento del lenguaje natural para contar distintas entidades nombradas según sean personas, organizaciones o lugares según un período de fechas seleccionado. A su vez, se provee la posibilidad de visualizar una representación de los vectores en un plano.

Módulo de clustering y detección de tópicos

En este módulo se analiza la semántica de las noticias para agruparlas automáticamente por significado. Dentro de cada grupo pueden detectarse tópicos formados por conjuntos de palabras. También es posible ver las representaciones vectoriales de las noticias coloreadas según el grupo al que pertenecen. A su vez, dentro de cada grupo es posible detectar tópicos así como listar los tweets más populares dentro de cada cluster.

Como los métodos de detección de agrupamientos no son tan precisos (se discutirá más adelante), es posible que los tweets populares no siempre estén relacionados con el tópico. Es por ello que se crea la noción de *"noticias representativas"*. Las noticias representativas son las que se encuentran semánticamente más cerca del agrupamiento y es por ello que tienden a ser las que tratan de temáticas más relacionadas con los tópicos del cluster.

Con este capítulo se introdujo uno de los principales resultados del proyecto de grado. Para la realización de varias de las tareas que se describieron, fue necesaria la construcción un dataset con el que se entrenaron los modelos. Esto es lo que trata el siguiente capítulo.

Capítulo 4

Corpus

La correcta construcción de un corpus es una base fundamental en cualquier proyecto de procesamiento de lenguaje natural. Para resolver el objetivo principal de este trabajo, se necesitó obtener todos los tweets de los seis medios de comunicación definidos previamente. Con ellos se construyó un clasificador que puede indicar si un tweet habla de seguridad y se lo integró a una plataforma web que permite la fácil visualización y estudio de los datos obtenidos.

La gran cantidad de tweets que se crean cada día hace que descargar los datos históricos de los últimos años sea un problema que debe ser sorteado. Para ello se crearon herramientas que permiten de forma automática recorrer cada cuenta de Twitter definida, desde el presente hasta sus inicios. También se debe construir un proceso continuo que día a día descargue los últimos datos para mantener la herramienta actualizada.

En el presente capítulo se describe el proceso de recolección de tweets en el que se manejaron grandes volúmenes de datos. Por otro lado, se describe el proceso de etiquetado manual que se hizo con una fracción de tweets que luego fueron usados como entrada de un modelo clasificador.

4.1. Obtención de tweets

Se recolectó el histórico de tweets creados por las cuentas de medios de información definidos de antemano: El País, El Observador, La Diaria, La República (ahora renombrado como Diario la R), Semanario Brecha y Búsqueda.

Si bien Twitter tiene una API, que puede ser usada para obtener datos de forma estructurada y documentada, sólo sirve para obtener los últimos 3200 tweets. Esto es muy poco para lo que se busca lograr en el proyecto. Por tal motivo, se debió implementar un scraper que accede a Twitter simulando el comportamiento de un usuario humano que

accede a la página y se desplaza indefinidamente hacia el pasado a medida que guarda la información de cada tweet.

De esta manera se descargaron todos los tweets disponibles de cada uno de los medios mencionados desde la actualidad hasta el año 2009.

Medio	Tweets descargados
El País	284,369
El Observador	282,896
La Diaria	80,039
La República	154,153
Búsqueda	27,097
Semanario Brecha	12,409
Total	840,963

Se sistematizó este scraper y se entregó esta herramienta al Departamento de Sociología para su uso. Ofrece la facilidad de construir grandes conjuntos de datos formados por tweets, ya sea descargando todo lo disponible o usando filtros obteneniendo los datos previamente filtrados. Para ver una descripción más técnica sobre la implementación y uso de la herramienta referirse a *Apéndice D: Herramientas para creación de dataset.*

4.2. Creación de conjunto de datos para entrenamiento del clasificador

Una vez obtenido el corpus principal, la siguiente tarea fue etiquetar los distintos tweets en `seguridad` o `no seguridad`. Para ello, se utilizaron los módulos de filtrado y etiquetado explicados a continuación.

4.2.1. Filtrado de tweets para etiquetado

En una primera instancia se intentó pasar directamente a la etapa de etiquetado manual, pero se encontró el inconveniente de que en el conjunto de todos los tweets descargados la proporción de tweets que no refieren a seguridad es mucho mayor a los que sí lo hacen (al rededor del 10 % referían a seguridad). Esto implicó que al recorrer los tweets para etiquetarlos, la mayoría fueran del tipo `no seguridad`. Para que el modelo aprenda de forma correcta a identificar textos de seguridad fue necesario que la proporción de tweets de seguridad fuera mayor.

Para resolver este problema se hizo un filtrado básico buscando obtener un conjunto

formado por tweets que contuvieran palabras relacionadas al tema seguridad.

Como punto de partida se tomó una lista de palabras relacionadas al tema de la seguridad creada por el Departamento de Sociología. Luego se construyó una nueva tabla con una estructura más simple para favorecer su utilización programática. Esta tabla de palabras simplificada se puede ver en *Apéndice A, Palabras Simplificadas.*

Si bien esta nueva lista de 323 palabras fue un buen comienzo para usar como filtro, se dio el problema de que un tweet podía contener una variación o conjugación de una de ellas y no ser detectado. Queriendo mitigar este problema, se probaron dos métodos distintos: stemming y extensión de vocabulario por diccionario.

Usando stemming se convierte cada palabra en una raíz (o *stem*). El problema que trajo este enfoque fue que introdujo mucho ruido con palabras que tenían la misma raíz pero significado diferente. Por ejemplo, al calcular el stem de *"balearon"*, se obtuvo la raíz *"bal"*, que también trajo tweets que contenían la palabra *"balde"*.

La técnica de extensión del vocabulario utiliza un diccionario formado por una lista de pares (*conjugación*, *infinitivo*), con el cual se puede extender la nueva lista simplificada usando sus diferentes flexiones. Primero se construyó un conjunto donde para cada infinitivo del diccionario se conoce una lista de posibles conjugaciones. Luego, si una de las palabras iniciales es igual a uno de los infinitivos entonces se agregan todas sus conjugaciones a la lista inicial. Por último, si una de las palabras iniciales pertenece a alguna lista de conjugaciones, se agrega el infinitivo y el resto de las conjugaciones a la lista inicial. Este proceso equivale a lematizar todo el conjunto de tweets pero de forma mucho menos costosa. De esta manera, se consiguió una lista final de 2874 palabras, en lugar de las 323 iniciales.

Utilizando esta nueva lista se filtró el corpus inicial obteniendo un nuevo conjunto reducido con tweets con mayor probabilidad de estar relacionados al tema seguridad.

Al comparar ambas técnicas utilizando un conjunto de datos reducido como ejemplo, se pudo notar que, usando la extensión de vocabulario a partir de un diccionario, se introducen menos tweets no relacionados al tema buscado, ya que se está usando conjugaciones de las palabras elegidas previamente por las personas del Departamento de Sociología y se evita capturar palabras no relacionadas que comparten la misma raíz. Por este motivo, el método elegido fue el de *extensión de vocabulario*, usando el diccionario para encontrar variaciones de las palabras iniciales.

Se filtró el corpus completo usando la lista extendida de palabras y se tomó una muestra aleatoria de 4000 tweets que fue utilizado como entrada para la etapa de etiquetado.

4.2.2. Etiquetado de instancias

Para poder entrenar el modelo, se necesitó tener datos etiquetados como `seguridad` o `no seguridad`. Si bien una posibilidad fue utilizar las mismas planillas donde se encontraban los tweets agregando una nueva columna con el valor, esto hubiera sido una tarea muy lenta con gran margen de confusión y dando lugar a errores de etiquetado. Fue requerida una solución más simple de usar que permitiera trabajar en equipo de forma ordenada y al final del proceso descargar todos los tweets etiquetados de forma unificada y estructurada.

Proceso de etiquetado

Se desarrolló una web[5] donde los integrantes del equipo, tutores y personal del Departamento de Sociología pudieran ingresar y clasificar de manera manual los diferentes tweets.

Figura 4.1: Vista de un tweet para ser etiquetado en la web

En un primer paso se cargó un conjunto de tweets previamente preprocesados y filtrados (resultado de la sección anterior) de aproximadamente cuatro mil tweets seleccionados de forma aleatoria para que los usuarios etiquetadores pudieran analizar y etiquetar de manera manual.

Luego de que los datos fueran cargados y los diferentes usuarios creados por administradores, se avanzó con el proceso de etiquetado.

Cada usuario, luego de iniciar sesión, se encuentra con una vista donde se muestra el

[5]https://tweet-tagger.herokuapp.com/

texto de un tweet en particular (elegido de forma aleatoria entre los posibles), un link al perfil del medio que lo publicó, un link al tweet, y por último, tres botones con las opciones: "Yes" (Sí), "No" y "Skip" (Saltear).

Al momento de elegir "Yes" o "No", se crea una instancia de un objeto *"clasification"* en la base de datos que vincula al tweet, usuario y la clasificación elegida. Se visualiza un ejemplo en la Figura 4.2. De esta manera, luego de obtener una cantidad satisfactoria de tweets etiquetados por distintos usuarios, se descarga mediante el portal de administración los tweets con su clasificación asignada. Una vez que un usuario etiqueta un tweet, este ya no será mostrado a él para una nueva clasificación.

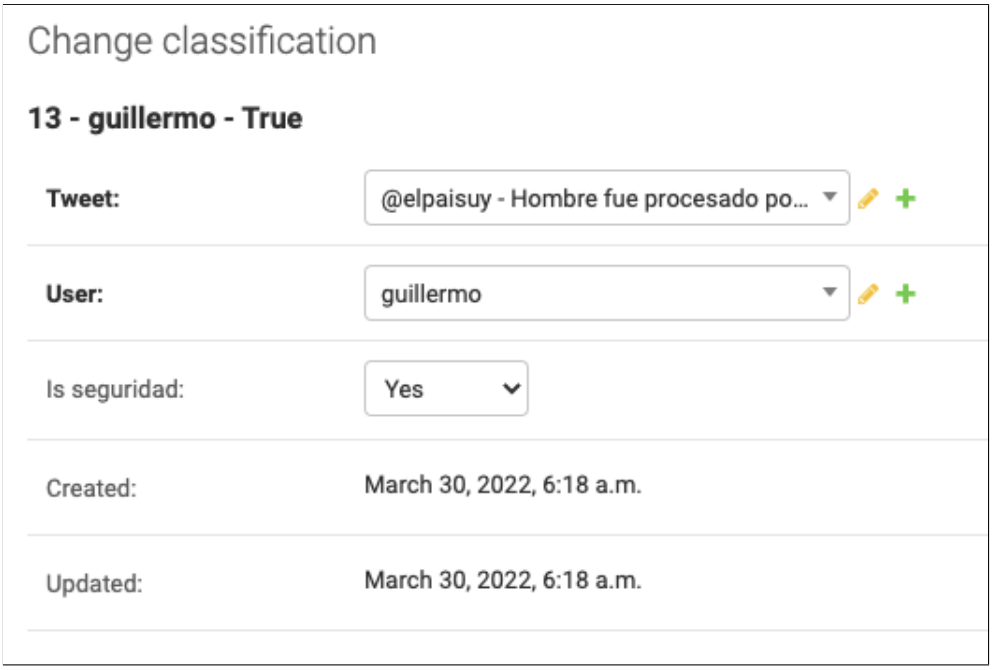

Figura 4.2: Ejemplo de una instancia de objeto "classification"

Si se elige "Skip", el tweet es salteado, sin crear una clasificación y el usuario puede seguir etiquetando nuevos tweets de seguridad. Esta opción es necesaria ya que muchas veces un tweet en particular puede ser confuso o ambiguo. A diferencia de las otras opciones, luego de ser salteado un tweet puede ser mostrado nuevamente al usuario, ya que puede ser que en otro momento se sienta más seguro sobre la etiqueta a elegir.

Se tomó la decisión de permitir que distintas personas pudieran etiquetar el mismo tweet. Esto permitió identificar los casos en los cuales un tweet recibió una etiqueta de seguridad por un lado, y una etiqueta de no seguridad por otro. Una vez que un tweet fue etiquetado tres veces, se dejó de mostrar para todos los usuarios ya que se consideró que se tenía la información suficiente.

Reglas de etiquetado y conflictos

Una vez empezado el proceso de etiquetar los tweets cargados, se hizo evidente que era una tarea más difícil de lo esperado. Fue muy común encontrarse con un tweet ambiguo o que generaba dudas sobre su posible etiqueta de forma frecuente. Se muestra un ejemplo en la Figura 4.3.

Figura 4.3: Ejemplo de tweet ambiguo

Este tweet menciona violencia de género pero es sobre una canción. Por otro lado, se tiene el ejemplo de la Figura 4.4 que también habla de seguridad pero en un contexto de campaña política.

Estos casos donde distintas personas pueden clasificar el mismo tweet con distintas etiquetas añadiría ruido al entrenamiento del modelo. Con el fin de unificar criterios, los expertos crearon la siguiente guía para usar en ciertos casos que pueden ser confusos, indicando si deben o no ser marcados como de la temática seguridad:

- Tweets relacionados al Sindicato Policial (que no involucren directamente hecho delictivo): **Sí**
- Tweets relacionados al Ministro del Interior (que no involucren directamente hecho delictivo): **Sí**
- Tweets relacionados a la Baja de Edad de Imputabilidad: **Si**
- Tweets que hablan sobre atentados terroristas: **Si**
- Tweets que hablan sobre incautación de drogas: **Si**
- Tweets que hablan sobre consumo de drogas: **Si**

- Tweets que hablan sobre hechos de la última Dictadura Militar: **Si**
- Tweets que hablan sobre presupuesto para la seguridad: **Sí**
- Tweets sobre accidentes/muertes de tránsito: **No**
- Tweets sobre leyes de seguridad: **Sí**
- LUC: **No, sólo si habla particularmente sobre seguridad**
- COVID y represión de aglomeraciones: **Sí**
- Abuso sexual (abuso sexual a mujeres, infantil, etc): **Sí**

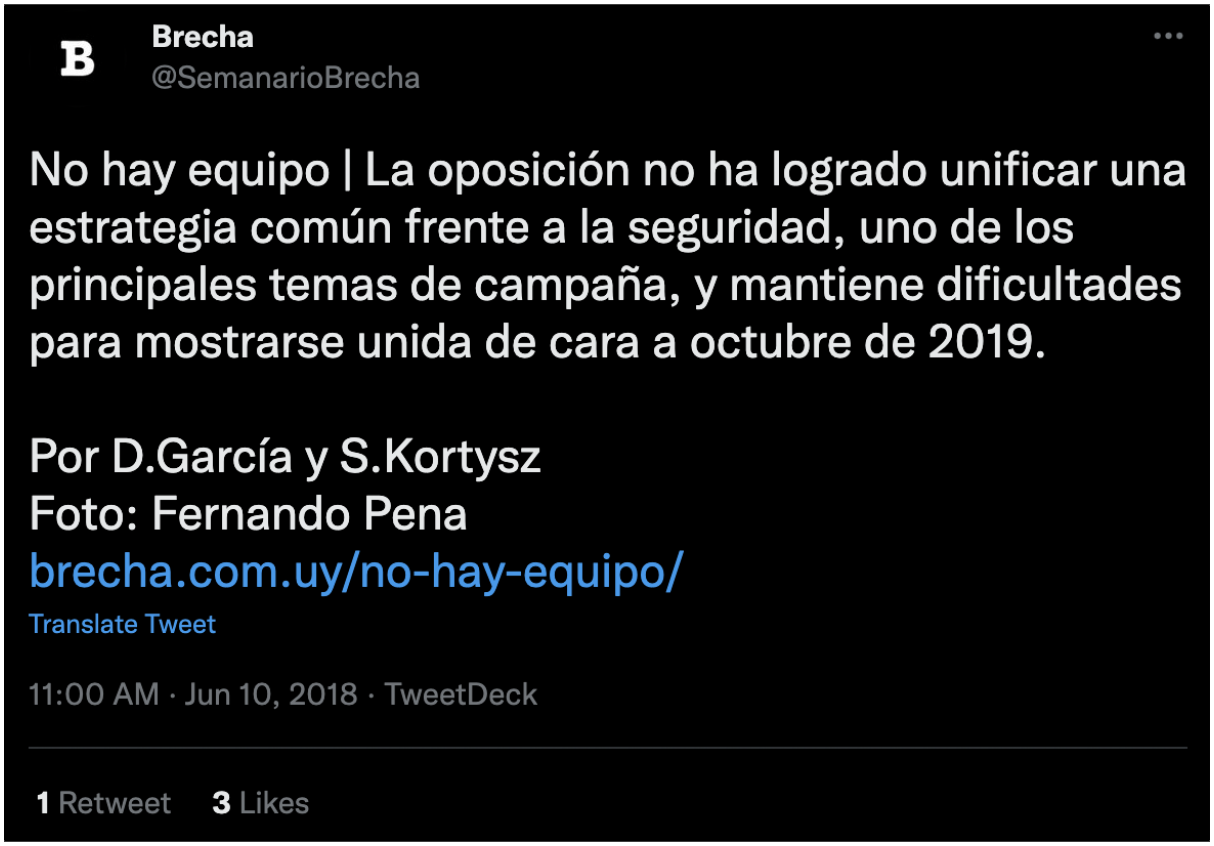

Figura 4.4: Ejemplo de tweet ambiguo (2)

Si bien igual hubo casos donde un mismo tweet fue etiquetado de manera contradictoria por distintas personas, no fue común obtener tweets con empates. Es decir, en caso de tener un tweet con dos clasificaciones `seguridad` y una `no seguridad`, se toma como válida la etiqueta `seguridad` por contar con más votos.

En caso de empate se consideró usar la etiqueta elegida por uno de los expertos, pero al final se decidió descartarlos ya que no era una cantidad significativa de tweets (60 de 2691) y no siempre uno de los etiquetadores era un experto.

Resultado del proceso de etiquetado

Como resultado del uso de esta herramienta se obtuvieron 2691 tweets etiquetados de forma manual como `seguridad` o `no seguridad`, de los cuales 60 de ellos estaban empatados y por lo tanto contaban con una etiqueta indefinida. Esto nos deja con un total final de 2631 tweets únicos y con una etiqueta definida, los cuales fueron usados para el entrenamiento del clasificador explicado en el siguiente capítulo. Este proceso fue realizado por siete anotadores de los cuales dos de ellos eran expertos.

4.3. Resultados y actualización de los datos

Al final del proceso de recolección y etiquetado se cuenta con un conjunto total de 840,963 tweets y un subconjunto de 2631 de ellos anotados manualmente como `seguridad` o `no seguridad`.

A partir del subconjunto etiquetado se entrena un modelo clasificador que será usado para clasificar el resto de los tweets del corpus total para luego poder ser visualizados y estudiados en la plataforma web. Este proceso se ve explicado en el siguiente capítulo.

Si bien a efectos del estudio y redacción del informe se utilizó el conjunto inicial recolectado de 840,963 tweets, los datos fueron actualizados con la herramienta de scrapeo ya mencionada y con un script que todos los días descarga los tweets del último día y los agrega a la base de datos. Al día 29 de julio del 2022 se cuenta con 1,064,378 tweets guardados en la base de datos disponibles para analizar utilizando la plataforma web distribuidos de la siguiente manera:

	Tweets descargados tomados en cuenta para el análisis y redacción del informe	Tweets al 18 de mayo
El País	284,369	388,322
El Observador	282,896	323,883
La Diaria	80,039	119,652
La República	154,153	187,290
Búsqueda	27,097	30,726
Semanario Brecha	12,409	14,505
Total	840,963	1,064,378

En el siguiente capítulo se describe la implementación de las principales funcionalidades de la aplicación construida.

Capítulo 5

Plataforma para análisis de noticias de seguridad

Como resultado tangible, se construyó una aplicación que permite analizar noticias a lo largo de los años y extraer de ellas información relevante de forma automatizada. En la presente sección se describen en detalle las principales funcionalidades de la aplicación implementada, incluidos los modelos de aprendizaje automático entrenados para la representación de word embeddings, el clasificador de tweets, pruebas de detectores de entidades nombradas implementadas por distintas bibliotecas, detecciones de tópicos y visualizaciones.

Tal como se describió en la sección anterior, se construyó un conjunto de datos de aproximadamente 840.000 tweets. Este corpus será el utilizado para el entrenamiento de modelos de word embeddings.

Cronológicamente, se intentó primero entrenar un modelo de word embeddings con la implementación básica del algoritmo *word2vec*. Una vez obtenidos, se pretendía usarlos como entrada de un clasificador, pero los resultados de las primeras experimentaciones no fueron nada alentadores y se tuvo que buscar una alternativa. Es así que se encuentra el paper *Bag of Tricks for Efficient Text Classification* [11], que propone una serie de técnicas eficientes para construir clasificadores de texto que iguala el estado del arte pero no implica contar con gran poder de cómputo ni excesivo corpus. La principal diferencia con las pruebas anteriormente realizadas radican en la arquitectura del modelo utilizado: el paper propone la utilización de información a nivel de subpalabras. Dicho trabajo propone los siguientes pasos para construir un clasificador de textos:

1. Representar las palabras de un texto como word embeddings y de esta forma poder usar los vectores numéricos como entradas de algoritmos.
2. Representar oraciones como el promedio de los vectores que las componen.

3. Con los ejemplos etiquetados, entrenar una regresión logística que permita clasificar una oración dentro de las clases definidas previamente.

5.1. Representaciones vectoriales de palabras

En la presente sección se describen los detalles de la implementación del modelo no supervisado que asigna una representación vectorial en un espacio de dimensión alta (300 en este caso) a cada una de las palabras del corpus inicialmente compuesto por 840.000 tweets. Esta técnica es conocida como *word embeddings*.

5.1.1. Entrenamiento de modelos

Para seleccionar el modelo óptimo se entrenaron cuatro modelos de word embeddings: dos que utilizan el algoritmo *word2vec* y otros dos que utilizan una variante de *word2vec enriquecida con subpalabras* [15]. Las dos implementaciones utilizadas y sus respectivas variantes se basan en redes neuronales.

Una vez disponibles los cuatro modelos, se compararon las respectivas representaciones vectoriales obtenidas y se eligió la más adecuada (este procedimiento se describe en siguientes subsecciones).

Modelos word2vec

El entrenamiento de estos modelos se hizo utilizando la biblioteca de código abierto *Gensim* [16] y haciendo uso del poder de cómputo del Cluster de Facultad de Ingeniería[6] sobre el conjunto total de tweets. Esta biblioteca provee la implementación del algoritmo *word2vec* con arquitectura *CBOW* y *skip-grams*. En ambos casos los resultados obtenidos fueron muy malos, dando a entender que la red neuronal utilizada no fue capaz de captar correctamente el contexto de las palabras, y por lo tanto, se obtuvieron representaciones vectoriales muy imprecisas. Para entrenar dichos modelos, se requirió cuatro días de cómputo para cada uno de los dos.

Modelos word2vec enriquecidos con subpalabras

En este caso, se utilizó la biblioteca *FastText* [11] (también de código abierto) desarrollada por Facebook. Aquí no fue necesario utilizar el poder de cómputo del Cluster, siendo que la implementación del algoritmo utilizado de *word2vec enriquecido con subpalabras* cuenta con una versión optimizada que tomó alrededor de media hora para entrenar cada una de sus dos variantes. En este caso, las representaciones vectoriales

[6] https://cluster.uy/

de las palabras del corpus comienzan a tener más sentido bajo la premisa de que palabras con representaciones vectoriales cercanas deberían tener significados similares. Ejemplos de esto se verán en la próxima subsección.

5.1.2. Elección del modelo

Se obtuvieron cuatro modelos que proveen representaciones vectoriales para las palabras que aparecen en el corpus. Todas las técnicas utilizadas para construir los modelos entrenados se basan en modelos probabilísticos que aprenden distribuciones estadísticas de las palabras a lo largo del corpus resultando en una herramienta para determinar probabilidades de ciertas palabras condicionadas a la aparición de otras. Usando este mismo principio se hizo una primera distinción del desempeño entre dos de las familias de modelos probadas: se eligieron ciertas palabras del contexto noticioso y se dejó que el algoritmo predijera las diez palabras más probables (y por lo tanto distancias menores a sus respectivas representaciones vectoriales del espacio de dimensión alta) esperando una cierta coherencia entre las elegidas. Las palabras elegidas fueron las siguientes: **“Ancap”**, **“cancha”**, **“Uruguay”**, **“mató”** y **“Bonomi”**.

En la Figura 5.1 se presenta la lista de las diez palabras más cercanas semánticamente según cada modelo: en la primera columna se encuentra la palabra elegida y en las siguientes las diez más cercanas según la similitud coseno. De la tabla se puede apreciar una enorme diferencia entre los modelos entrenados con la arquitectura *word2vec* y la *word2vec enriquecida con información de subpalabras*. Los primeros parecen sugerir palabras aleatorias, mientras que los segundos presentan una mayor coherencia y dan una idea de que aprendieron de forma más correcta los contextos de las palabras. Las palabras que sugieren son siempre sintáctica y/o semánticamente similares. Es por esta razón que se decidió utilizar los segundos modelos para continuar con el trabajo y sólo se entrenaron los clasificadores con ellos.

	w2v_cbow	w2v_skipgram	w2v_cbow_enr	w2v_skipgram_enr
Ancap	'torra', 'autoridades', 'primeravoz', 'biología', 'interceptada', 'antunes', 'desahogo', 'supervisores', 'antagónicos', 'religiosas'	'besarlo', 'latitud', 'toxicológicos' 'gooooooollll', 'brillar', 'rbc', 'sanchiz', 'bandana', 'estreptococo' 'whatsappen'	'fancap', 'ancapuruguay', 'anc', 'anca', 'aap', 'anp', 'pluna', 'snap', 'alur', 'anpl'	'fancap', 'ente', 'alur', 'ancapuruguay', 'pórtland', 'combustibles', 'coya', 'sobrecosto', 'combustible', 'regasificadora'
cancha	'razonar', 'irae', 'larsen', 'colabora', 'bloqueos', 'expulse', 'ménendez', 'exreal', 'respondían', 'polvo'	'abordarla', 'unificó', 'helene', 'sovereign', 'coronara', 'ubiquen', 'chupando', 'roboprok', 'adriano', 'respetuosa'	'canchas', 'jugar', 'céspedes', 'partidazo', 'albiceleste', 'futbolito', 'zalayeta', 'partidazos', 'elche', 'albicelestes'	'canchita', 'canchas', 'cagancha', 'chancha', 'ancha', 'canchero', 'pelota', 'plancha', 'revancha', 'jugara'
Uruguay	'tinte', 'perí', 'bosques', 'iptv', 'gratuita', 'omisa', 'equivocaron', 'merienda', 'grabois', 'pasóhoy'	'hontou', 'estilista', 'goma', 'pudras', 'undíasinnosotras', 'asaditos', 'hyeres', 'banking', 'minimalmambo', 'desarticulen'	'uruguaybrasil', 'uruguayperú', 'uruguay365', 'uruguayparaguay', 'uruguayxxi', 'uruguaí', 'uruguayeeuu', 'chileuruguay', 'uruguayoargentina' 'uruguaychile'	'uruguayparaguay', 'uruguayeeuu', 'uruguayxxi', 'uruguayxmas', 'uruguaí', 'uruguaybrasil', 'uruguayperú', 'urugua', 'chinaamérica', 'uruguaylibre'
mató	'volodimir', 'vibrante', 'espiando', 'agremiados', 'impresionantes', 'estallaron', 'manya', 'influye', 'pólizas', 'corporativos'	'argentina', 'apostaremos', 'fracturó', 'portia', 'manipuladores', 'sublimes', 'exsuplente', 'insult', 'delimitar', 'decepcionado'	'asesinó', 'mataba', 'mata', 'ató', 'mataron', 'apuñaló', 'matara', 'hirió', 'matón', 'matarás'	'asesinó', 'mataron', 'hirió', 'matara', 'huía', 'asesina', 'apuñaló', 'mataran', 'huyó', 'asesine'
Bonomi	'banchero', 'cotidianas', 'prerreferéndum', 'delgrossi', 'trasladados', 'seguí', 'raqa', 'energizar', 'refugiadas', 'indagada',	'culminación', 'finalmente', 'cachanosky', 'antidoping', 'asia', 'solapados', 'guanabara', 'maratea', 'chanchos', 'pailós'	'larrañaga', 'poularrañaga', 'larrañanga', 'bordaberry', 'lacallelarrañaga', 'interior', 'jorgegandini', 'gandini', 'pedrobordaberry', 'pomi'	'interior', 'larrañaga', 'layera', 'interpele', 'poularrañaga', 'interpelar', 'interpelado', 'larrañanga', 'paternain', 'interpelará'

Figura 5.1: Tabla de las palabras que comparten mayor nivel de semántica según cada modelo

Vale destacar cómo dentro de los resultados de la palabra *"cancha"*, en los modelos de FastText, aparecen palabras que comparten caracteres similares pero no así mucho significado como *"chancha"* o *"cagancha"*. Por otro lado, en los resultados de la palabra "Uruguay" se ven varios tokens que parecieran haber sido hashtags antes de la fase del preprocesamiento.

5.2. Clasificación de tweets

El clasificador es el encargado de decidir si una noticia pertenece o no al tema *"seguridad ciudadana"* generalizando los datos vistos en *el conjunto de entrenamiento* previamente etiquetado. Para esta tarea se entrenó un modelo de regresión logística. Dicha regresión toma como entrada un vector **x** que representa cada tweet, y como salida **y** los valores 1 o 0 si el tweet es clasificado como `seguridad` o `no seguridad`.

5.2.1. Entrada del modelo

Como se comentó, el entrenamiento del clasificador requiere que las entradas sean vectores de largo fijo. A su vez, los modelos de embeddings utilizados proveen representaciones vectoriales para palabras y no para conjuntos de palabras (como lo es un tweet). Es por esto que se representó cada tweet como el promedio de los vectores de las palabras que lo conforman, tomando dicho centroide como valor **x** de entrada del modelo clasificador. Por otro lado, se tomó la etiqueta de seguridad codificada (0 o 1) como salida **y** (o valor a predecir). Si bien se sabe que bajo estas hipótesis se pierde gran parte de la información que brindan los vectores de las palabras por separado, los resultados demostraron que esta simple técnica permite captar información suficiente para construir un clasificador con métricas de *acierto*, *precisión* y *recuperación* bastante buenas.

5.2.2. Elección del modelo de clasificación

Se entrenaron dos regresiones logísticas, utilizando en cada caso uno de los dos modelos de word embeddings obtenidos (*w2v_cbow_enr* y *w2v_skipgram_enr*) en un total de 2104 ejemplos etiquetados como *conjunto de entrenamiento*. Los resultados fueron evaluados en un *conjunto de test* con 527 ejemplos, de los cuales 151 estaban etiquetados como *seguridad* y los restantes 376 como *no_seguridad*. En ninguno de los casos se tuvo que experimentar con ajuste de hiperparámetros ya que la biblioteca utilizada (FastText) cuenta con un módulo de autotune que lo hace por el programador utilizando técnicas de cross validation con distintos valores y seleccionando el mejor.

Sobre el modelo CBOW enriquecido con subpalabras

Con la regresión logística ajustada utilizando las representaciones del modelo *w2v_cbow_enr* se obtiene un modelo con una accuracy (acierto) de 0.85. Además, se obtienen 0.74 de precisión y 0.72 de recall sobre la clase de seguridad, lo que fue considerado como un buen número. Para entender cómo se distribuyen las clasificaciones y poder comprender cuáles son los defectos y virtudes del modelo se generó la matriz de confusión de la Figura 5.2

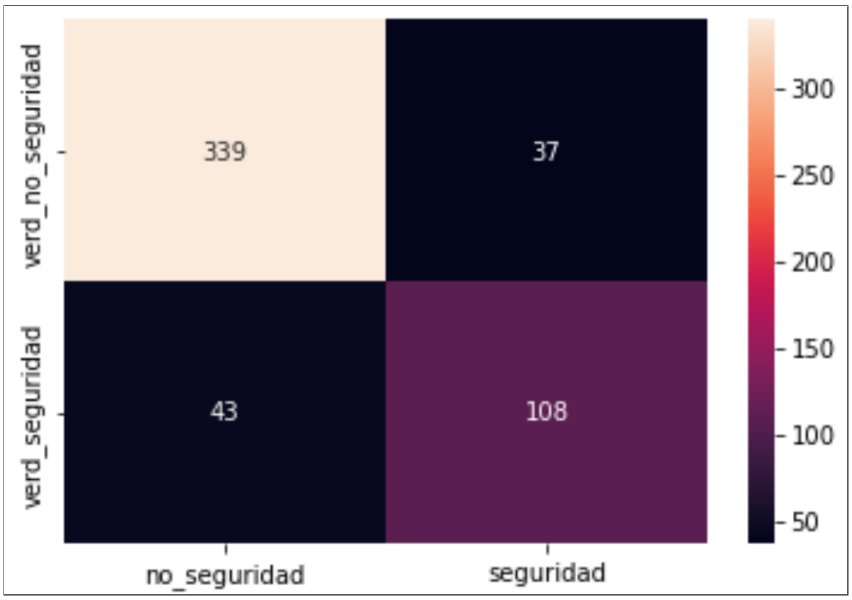

Figura 5.2: Matriz de confusión modelo CBOW

Etiqueta	precisión	recall	f1-score	support
no seguridad	0.89	0.90	0.89	376
seguridad	0.74	0.72	0.73	151

Figura 5.3: Métricas por clase modelo CBOW

De los datos expuestos en la Tabla de la Figura 5.3 se puede concluir que el modelo detecta correctamente 90 % de los tweets que hablan sobre otros temas que no están relacionados con la seguridad ciudadana y 72 % de los tweets que hablan de seguridad.

Sobre el modelo skipgram enriquecido con subpalabras

Con la regresión logística ajustada utilizando las representaciones del modelo *w2v_skipgram_enr*, se obtuvo un modelo con accuracy (acierto) del 0.87. Además, se obtienen una precisión de 0.83 y recall de 0.70 sobre la clase de seguridad, lo que también fue considerado como un buen número. Para entender cómo se distribuyen las clasificaciones y poder comprender cuáles son los defectos y virtudes del modelo se generó la matriz de confusión que se visualiza en la Figura 5.4.

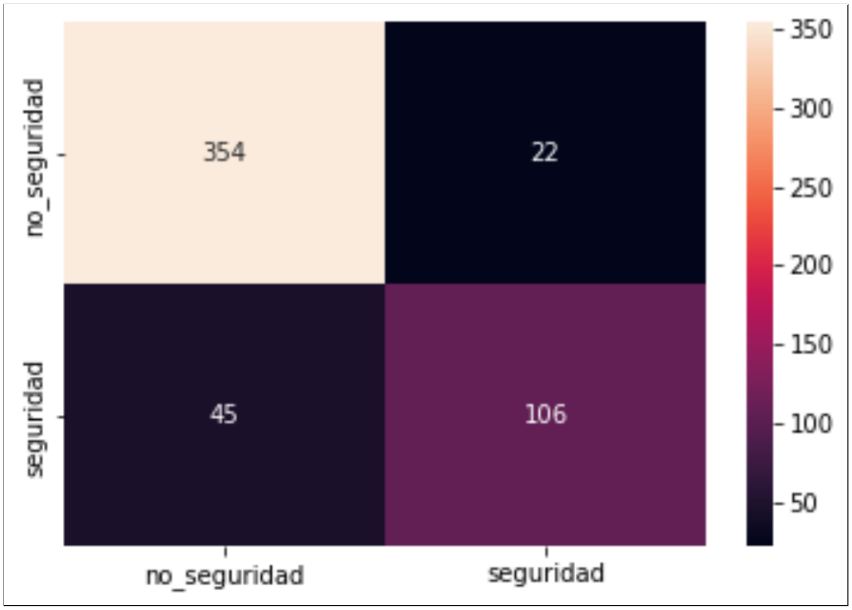

Figura 5.4: Matriz de confusión modelo Skipgram

etiqueta	precisión	recall	f1-score	support
no seguridad	0.89	0.94	0.91	376
seguridad	0.83	0.70	0.76	151

Figura 5.5: Métricas por clase modelo Skipgram

Se puede concluir, a partir de los datos de la tabla de la Figura 5.5, que el modelo detectó 94 % de los tweets que hablan sobre otros temas que no están relacionados con la seguridad ciudadana y 72 % de los tweets que hablan de seguridad.

Comparación de los modelos

En ambos casos, la diferencia entre el acierto y la precisión de las clases, se explica porque están desbalanceadas: hay 376 noticias de no seguridad frente a 151 noticias de seguridad.

Si bien el modelo *w2v_skipgram_enr* tiene un acierto superior al *w2v_cbow_enr*, este último tiene un mayor recall sobre la clase de seguridad, lo que significa que identifica con mayor exactitud a los pertenecientes a esta clase. Ya que interesa particularmente detectar las noticias de seguridad, se eligió el modelo que detecta dicha clase con mayor recall: *w2v_cbow_enr*.

5.3. Detección de entidades nombradas

Para la implementación del módulo detector de entidades nombradas, se probaron dos alternativas implementadas por: Stanza[7] y Spacy[8]. Para calcular cuál de las implementaciones mostraba mayor rendimiento en el contexto de nuestros datos, se construyó un *ground truth* para tener contra qué comparar.

En todos los casos, antes de pasar la noticia por el extractor de NERs se hizo un preprocesamiento relativamente sencillo: se eliminan saltos de línea y se eliminan URLs. Con esta decisión se quiere que se detecten la mayor cantidad de entidades y se incluyan tanto hashtags como menciones a cuentas.

En la tabla de la Figura 5.6 se muestran los textos preprocesados junto con las organizaciones, personas o lugares detectados.

[7]https://stanfordnlp.github.io/stanza/
[8]https://spacy.io/

Texto	Lugares	Personas	Organizaciones
#Tribuna | Peñarol: Pellistri al Lyon y esta noche enfrenta al Colo Colo Peñarol | Diario La República Desde las 19:15 horas juega un partido decisivo por la Libertadores.	[]	['Pellistri']	['Colo Colo', 'Lyon', 'Peñarol', 'Diario La República']
Es falso que el FA haya solicitado depósitos bancarios para financiar viajes de votantes para el balotaje". Nuevo chequeo de @verificadouy	[]	[]	['FA', '@verificadouy']
Manini sobre el Frente Amplio: "Sienten que se van y apelan a todos los recursos, tenemos que estar alerta" El líder de Cabildo Abierto dijo que en el FA "sienten que se van".	[]	['Manini']	['Frente Amplio', 'Cabildo Abierto', 'FA']
El fiscal Pacheco sostuvo que Sendic "incumplió flagrantemente" el reglamento de las tarjetas de Ancap y negó que el delito de abuso de funciones sea un "cajón de sastre" Escribe @vicfer88	[]	['Pacheco', 'Sendic', '@vicfer88']	['Ancap']
DISIDENTE CUBANA - Hilda Molina se reúne con su familia en Buenos Aires	['Buenos Aires']	['Hilda Molina']	[]
Exdirector de Casinos a Arbeleche sobre las 200 máquinas tragamonedas: "Se sabía de su existencia", "Está claro que no sabe de qué habla", dijo Chá.	[]	['Exdirector de Casinos', 'Arbeleche', 'Chá']	[]
Argentina | Conmoción de los famosos por la renuncia de #Messi @tvshowuru	['Argentina']	['Messi']	['@tvshowuru']
Encuesta de Factum marca tendencia de subida del FA y caída colorada El 40% de los encuestados piensa votar al Frente Amplio, el 28% al Partido Nacional, el 13% al Partido Colorado y el 11% a Cabildo Abierto	[]	[]	['Factum', 'FA', 'Frente Amplio', 'Partido Nacional', 'Partido Colorado', 'Cabildo Abierto']

Figura 5.6: Textos preprocesados junto con sus organizaciones, personas o lugares

Se listan en la tabla de la Figura 5.7 los resultados que describen la performance de los algoritmos.

Lugares			Personas			Organizaciones		
Total	Detectadas Spacy	Detectadas Stanza	Total	Detectadas Spacy	Detectadas Stanza	Total	Detectadas Spacy	Detectadas Stanza
2	1	1	10	3	8	17	5	14
	Mal detectadas Spacy	Mal detectadas Stanza		Mal detectadas Spacy	Mal detectadas Stanza		Mal detectadas Spacy	Mal detectadas Stanza
	12	1		4	1		0	2

Figura 5.7: Resultados de performance de los algoritmos

En general, el modelo de Stanza tiene una mejor performance comparado con el de Spacy sobre nuestro conjunto de datos. De las 29 entidades, el primero identifica 23 correctamente, frente a 9 que identifica el segundo. Por otro lado, Stanza detecta 4 entidades incorrectas, frente a las 16 incorrectas que detecta Spacy.

Es por esto que se procede a integrar el detector de entidades nombradas de Stanza en la aplicación.

5.4. Detección de tópicos

Si bien este problema es complejo en sí, se intentó hacer un primer acercamiento para agregar la funcionalidad al trabajo final. En este módulo se combinan varias técnicas y herramientas que permiten agrupar noticias para luego detectar tópicos en ellas. Se decidió utilizar un enfoque no supervisado para agrupar las noticias haciendo uso de su representación vectorial previamente descripta (formada por el promedio de vectores de las palabras que la componen). Una vez que se cuenta con dicha representación vectorial se aplica el algoritmo de k-means para detectar agrupaciones semánticas, esperando que noticias que comparten tópicos o conjuntos de tópicos formen parte del mismo cluster. Para aplicar k-means al conjunto de publicaciones seleccionado, el usuario debe ingresar un valor de k que crea conveniente. A su vez, dentro de cada cluster, puede correrse el algoritmo LDA (Latent Dirichlet Allocation) para detectar tópicos formados por conjuntos de 10 palabras.

Como forma de mostrar el resultado, se presenta a continuación un ejemplo del sistema tomando la primera semana de marzo del año 2020.

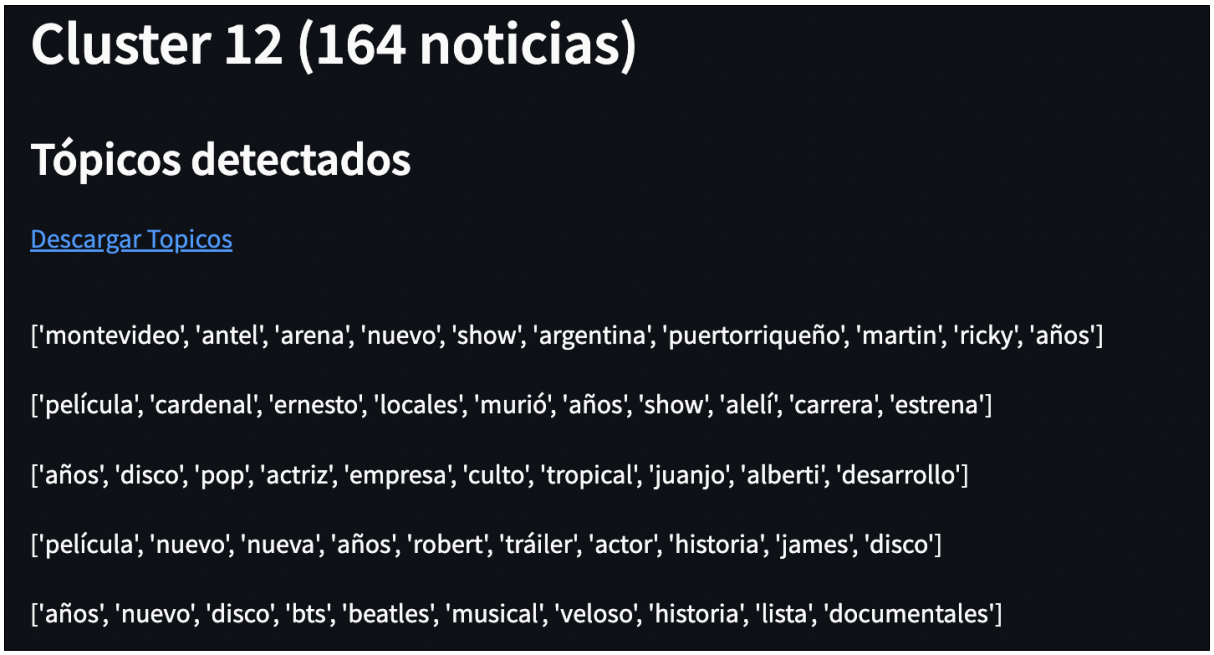

Figura 5.8: Ejemplo de cluster y tópicos detectados

De la Figura 5.8 puede verse cómo en el cluster numerado como 12, por el algoritmo de k-means, se agrupan noticias relacionadas con eventos y entretenimiento en general, habiendo varios tópicos con sentido que se pueden identificar.

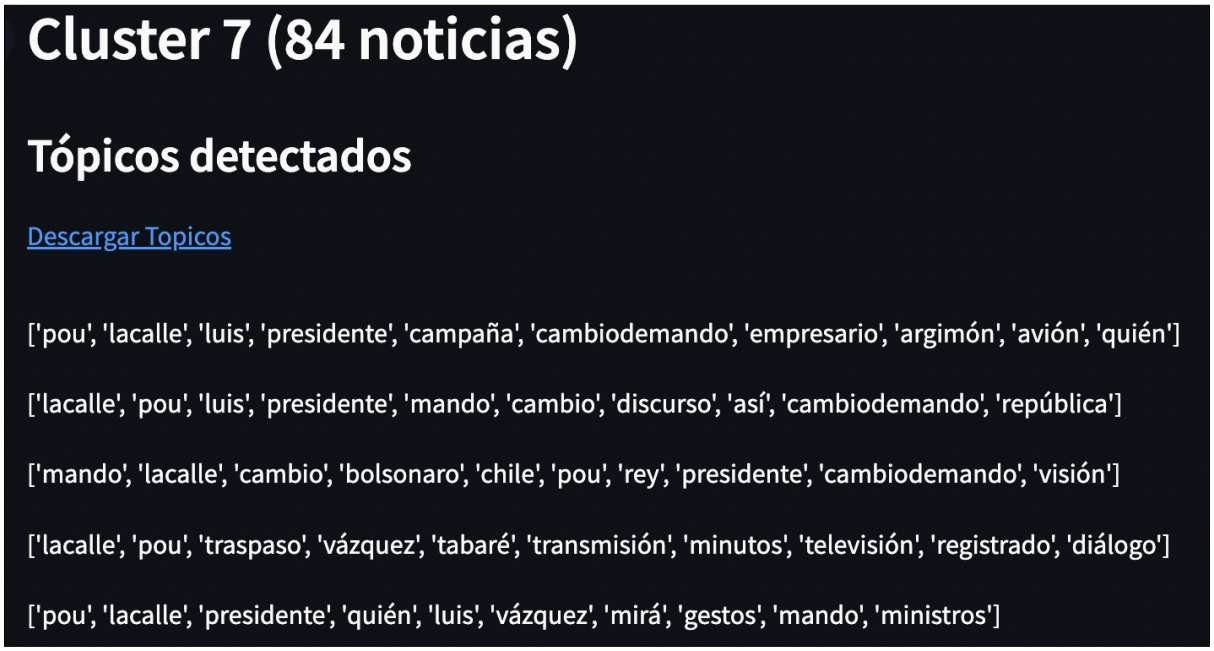

Figura 5.9: Ejemplo de cluster y tópicos detectados

Por otro lado, puede verse cómo en el cluster de la Figura 5.9 con menor cantidad de noticias, se detectan tópicos relacionados al cambio de mando ocurrido el primero de marzo del año analizado.

Dentro de cada cluster es posible detectar tanto *noticias representativas* del cluster como *noticias populares*. Las *noticias populares* son las que han recibido más retweets, mientras que las *representativas* son las que están más cerca (utilizando la distancia coseno) del centroide del respectivo cluster. Las noticias que están más lejos del centroide de cada cluster tienden a ser las que menos están relacionadas con los temas del cluster. Estas últimas afirmaciones asumen que el centroide es el que "concentra" información de la semántica, por lo que, si muchas noticias del cluster utilizan determinadas palabras, el centroide estará cerca de ellas.

En las Figuras 5.10 y 5.11 se visualizan las noticias populares y representativas de un cluster concreto en el ejemplo anteriormente descripto (todas las noticias de la primera semana de marzo del 2020). En este caso, tanto las noticias representativas como las populares hablan de temas similares (principalmente espectáculos), pero esto no siempre sucede en la práctica.

Noticias populares

Descargar Noticias

	Handle	Tweet	Attachment	PostDate	RetweetCount
1296	@elpaisuy	El ex One Direction Niall Horan viene a Montevideo con su show solista: los detalles	El ex One Direction Niall Horan viene a Montevideo con su show solista: los detalles El cantante pop, que ya estuvo en Uruguay en 2014 con su vieja boyband, llegará en noviembre al Teatro de Verano con su "Nice To Meet Ya Tour" tvshow.com.uy	2020-03-04T16:01:18+00:00	25.0000
1219	@ObservadorUY	Ricky Martin volvió a Montevideo con su Movimiento Tour y dejó un tendal en el Antel Arena tras su impresionante show. Por @felipellambias .	Ricky Martin revivió la fantasía sexual adolescente en las uruguayas y quiere más El cantante puertorriqueño volvió a Montevideo con su Movimiento Tour y dejó un tendal en el Antel Arena tras su impresionante show elobservador.com.uy	2020-03-04T11:39:10+00:00	7.0000
604	@elpaisuy	Asesinaron a Luis Alfonso Mendoza, actor de doblaje que le puso voz en español latino a personajes como Joey Tribbiani en “Friends”; Sheldon Cooper en “The Big Bang Theory”; y Gohan en “Dragon Ball Z”	Asesinaron a Luis Alfonso Mendoza, actor de doblaje de "Friends" y "The Big Bang Theory" El intérprete mexicano le puso voz a personajes como Joey Tribbiani en “Friends”; Sheldon Cooper en “The Big Bang Theory”; y Gohan en “Dragon Ball Z” tvshow.com.uy	2020-03-02T17:26:15+00:00	7.0000

Figura 5.10: Noticias populares del cluster 12

	Handle	Tweet	Attachment	PostDate
0	@elpaisuy	Un Gaucho Influencer que agota entradas en el UnderMovie con humor para toda la familia	Un Gaucho Influencer que agota entradas en el UnderMovie con humor para toda la familia Eduardo Fernández y su personaje el Gaucho Influencer, un fenómeno que salió de las redes sociales y agota funciones con humor para la familia tvshow.com.uy	2020-03-07
1	@elpaisuy	Backstreet Boys llega a Montevideo con un viaje a la adolescencia que promete diversión en el Antel Arena	Backstreet Boys llega a Montevideo con un viaje a la adolescencia que promete diversión La boy band estadounidense se presentará domingo y lunes en el Antel Arena. Quedan entradas para el segundo show, las del primero están agotadas tvshow.com.uy	2020-03-06

Figura 5.11: Noticias representativas del cluster 12 (2)

5.5. Visualizaciones

Gran parte del valor agregado de la plataforma reside en la visualización de los datos: tanto de estadísticas generales (que se mostraron en la sección de descripción de la solución) como en la visualización de los word embeddings que aparecen transversalmente durante todo el presente trabajo.

Una vez elegida la representación final de vectores que se utilizó, se probaron dos técnicas de reducción de dimensiones para visualizar los vectores de dimensión 300 en un plano de dos dimensiones: t-SNE y PCA. Para decidir cuál de los dos algoritmos utilizar, se hicieron proyecciones y visualizaciones de distintos conjuntos de vectores, esperando que nuevamente se cumpliera la hipótesis de que representaciones vectoriales de palabras o expresiones con semántica similar tuvieran representaciones vectoriales cercanas. Vale aclarar que ambas técnicas tuvieron resultados distintos dependiendo si se utilizaban para palabras (word embeddings) o conjuntos de palabras (promedios de word embeddings que componen las oraciones).

Pruebas en palabras

Los siguientes pares de palabras fueron seleccionados con la intención de que la representación vectorial de su proyección en dos dimensiones fuera cercana:

- (vacuna, covid)
- (policía, bonomi)
- (playas, turismo)

Dados los resultados de ambos algoritmos de reducción de dimensiones visualizados en las Figuras 5.12 y 5.13, PCA genera distancias menores en palabras con semánticas y/o contextos similares. Es por esto que se decidió utilizarlo para representar vectores de palabras (esto excluye vectores de oraciones).

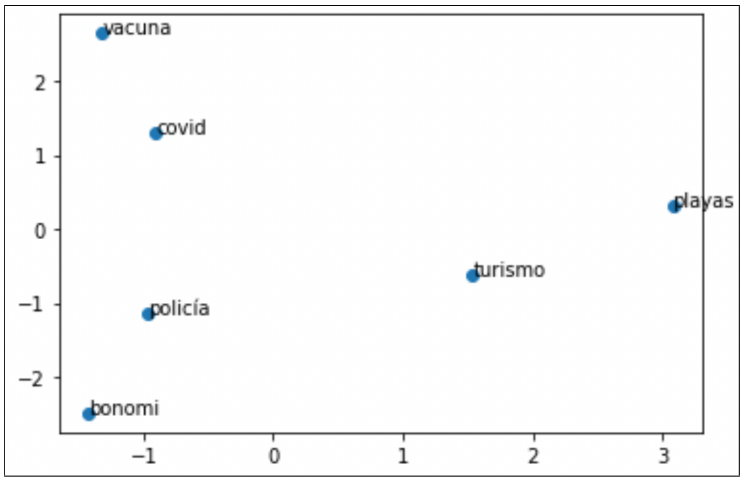

Figura 5.12: Reducción de dimensiones utilizando PCA

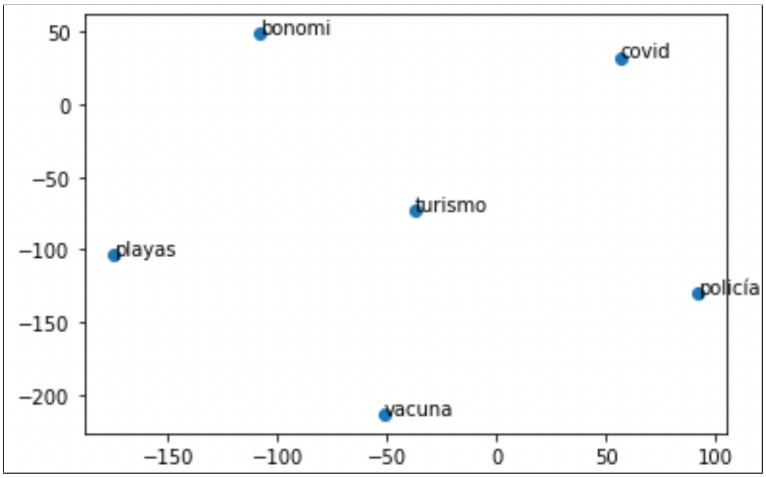

Figura 5.13: Reducción de dimensiones utilizando t-SNE

Pruebas en oraciones

Las noticias o tweets generalmente no están formados por una sola palabra sino un conjunto de ellas (frase u oración). Recordando que nuestro modelo representa a dichos conjuntos como el vector promedio de los vectores de las palabras que lo componen, se comparó el desempeño de los algoritmos PCA y t-SNE en el conjunto de todos los tweets. En este caso, por facilidad, se intentó visualizar agrupamientos por semántica del conjunto de todas las noticias. Para poder visualizar la calidad de las proyecciones y el hecho de que se conserven distancias, se corrió un algoritmo de k-means para generar 80 clusters que serán coloreados y estudiados esperando que tweets que pertenezcan a las mismas agrupaciones tengan sus representaciones cercanas. El número 80 fue tomando por recomendación de los involucrados y se sabe que puede ser un poco arbitrario pero sirve con el fin de identificar de qué forma se agrupan los vectores en el espacio proyectado.

Si se comparan las representaciones del conjunto de todas las noticias proyectadas con ambas técnicas (PCA en la Figura 5.14 y t-SNE en la Figura 5.15), puede verse cómo las distancias de los clusters se conservan en t-SNE viéndose agrupaciones de colores mientras que en PCA las distribuciones parecen aleatorias, estando los colores de los clusters esparcidos por todo el plano.

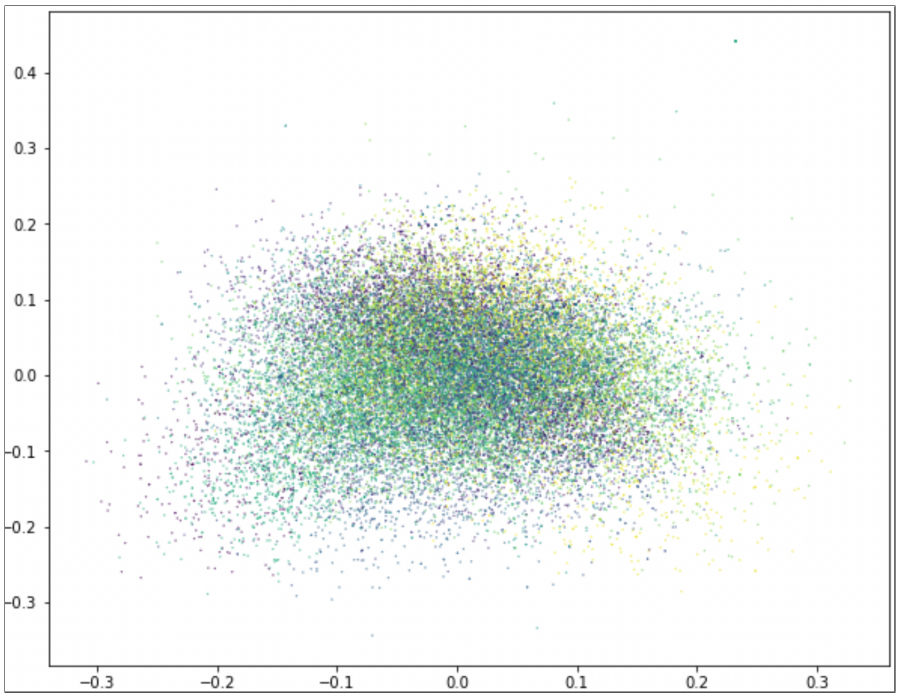

Figura 5.14: Reducción de dimensiones de tweets utilizando PCA

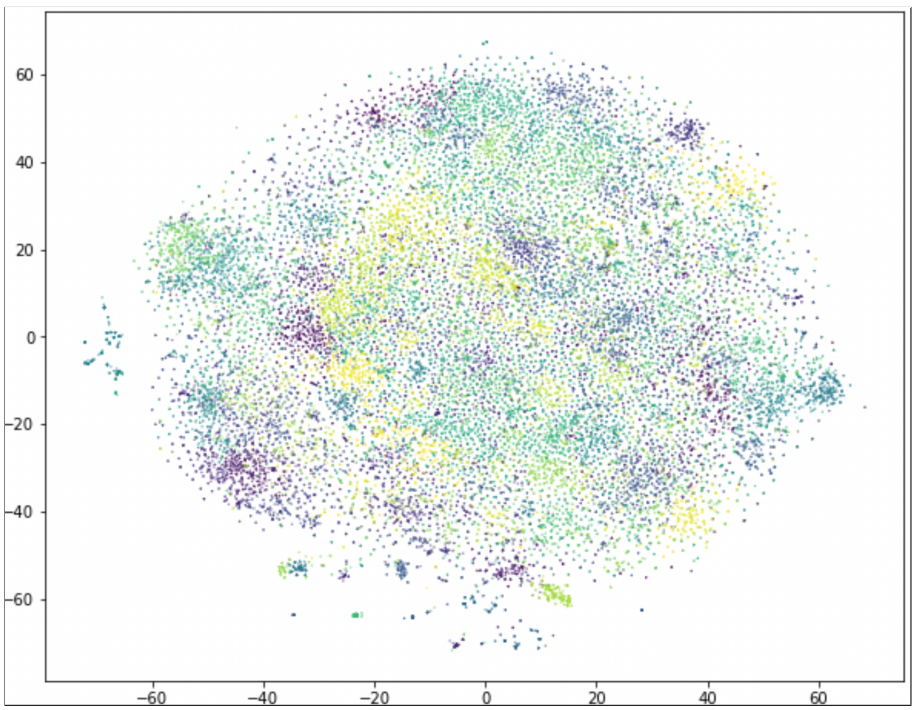

Figura 5.15: Reducción de dimensiones de tweets utilizando t-SNE

En el presente capítulo se describieron las implementaciones de las principales funcionalidades de la aplicación que se construyó con el fin de analizar datos de publicaciones ciudadanas. Todas ellas pueden usarse para estudiar distintas aristas de la temática de seguridad ciudadana. Para la construcción de las visualizaciones se utilizó la biblioteca Streamlit en conjunto con bibliotecas de gráficas en Python como lo es Matplotlib. Ejemplos y más detalles de esto pueden encontrarse en el *Apéndice B: Despliegue de aplicación*.

A modo de ejemplo del potencial que tiene la herramienta, el siguiente capítulo muestra un caso de uso completo.

Capítulo 6

Estudio concreto utilizando la herramienta construida

La herramienta construida permite realizar tareas de análisis cualitativos y cuantitativos sobre las noticias publicadas en cierto período. En el presente capítulo se muestra un flujo completo de la aplicación a modo de ejemplo para mostrar su potencial.

En la figura 6.1 se muestra cómo luce la aplicación en su página de inicio. Por defecto se cargan las noticias de los últimos tres meses, y se muestran estadísticas de las publicaciones incluyendo qué porcentaje están relacionadas con la temática de seguridad.

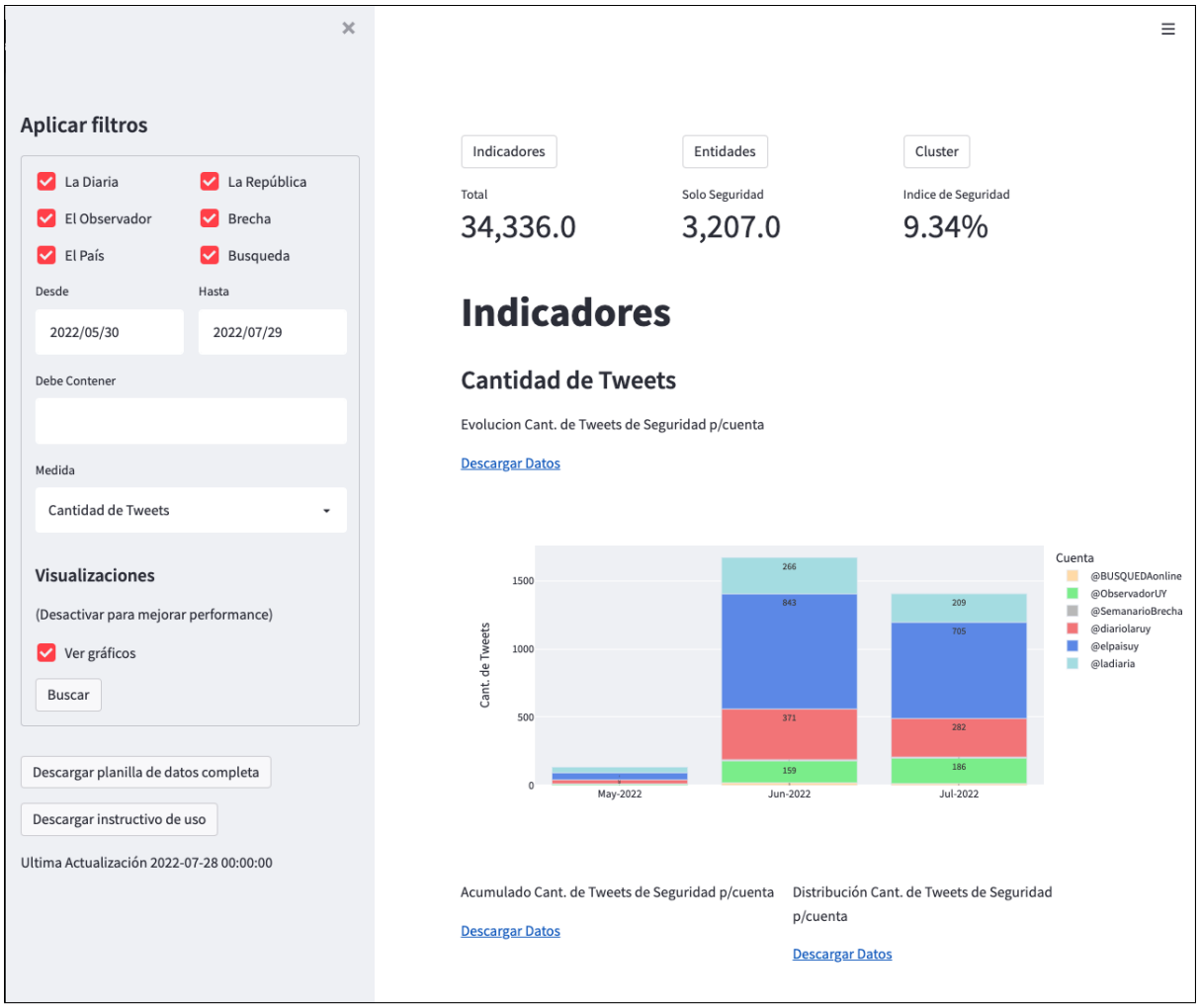

Figura 6.1: Pantalla inicial de la aplicación

El análisis a realizar consiste en buscar las publicaciones que contuviesen la palabra *"LUC"* publicadas desde enero del 2012 hasta mayo del 2021 (fecha de la última actualización del conjunto de datos desde que se escribe el presente informe). Las fuentes fueron los seis medios con los que se trabajó. La barra de filtros luce como se muestra en la Figura 6.2.

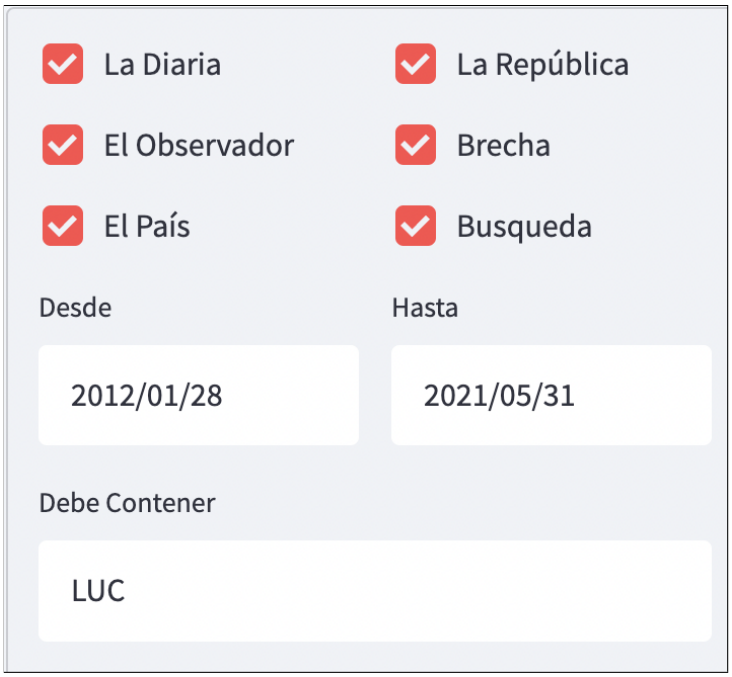

Figura 6.2: Barra de filtros

6.1. Estadísticas generales

En la Figura 6.3 se puede apreciar la evolución de noticias que contienen la palabra *"LUC"*. Allí puede verse cómo la utilización del término empieza a cobrar relevancia en enero del 2020, siendo el 100 % de las ocurrencias anteriores referentes a nombres (líder izquierdista francés Jean-*Luc* Mélenchon, Jean-*Luc* Lagarce actor, escritor y director de teatro francés), alcanzando un pico sobre mayo del 2020. El total de noticias fue de 1438.

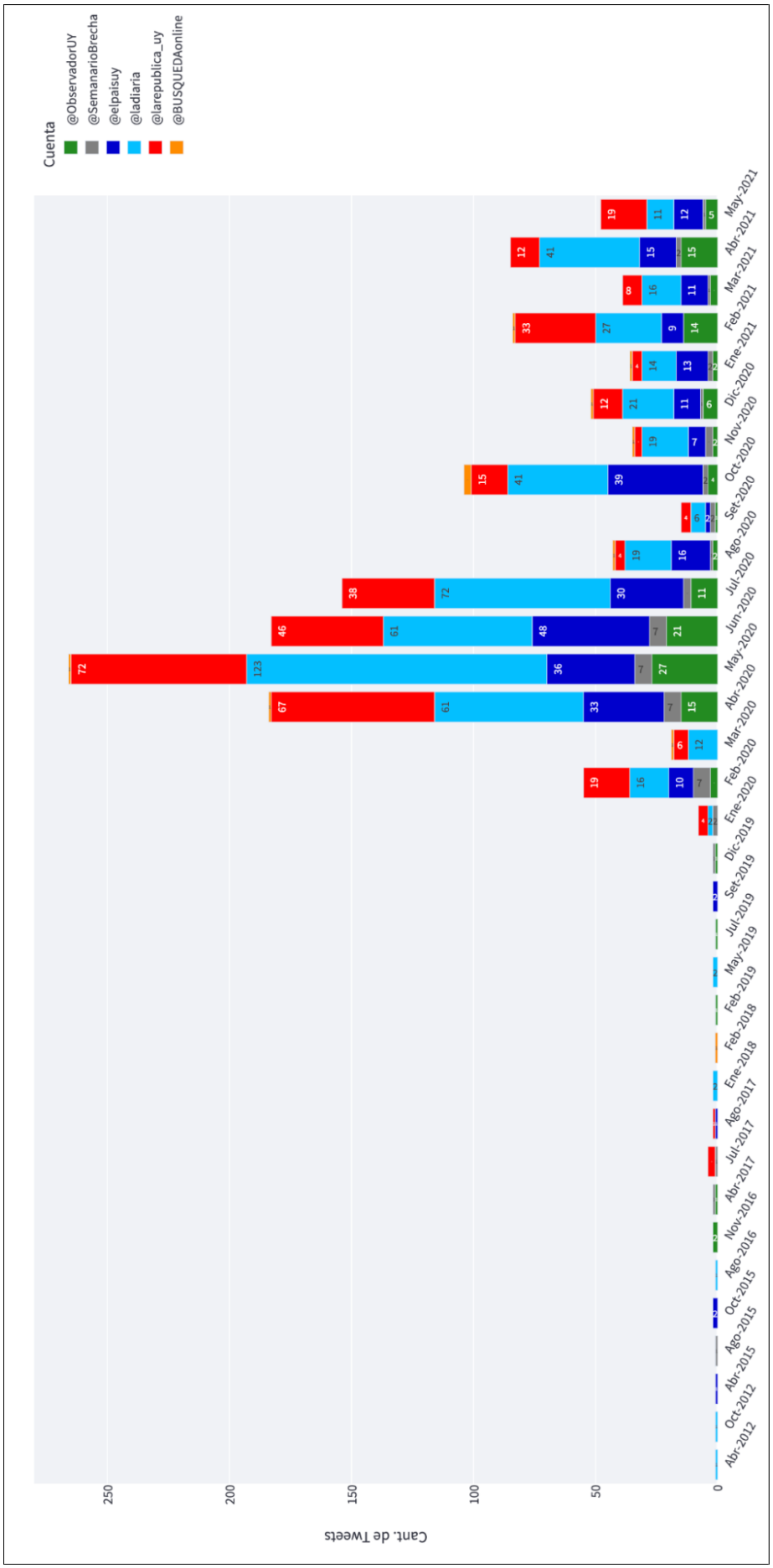

Figura 6.3: Evolución cantidad de tweets sobre LUC por mes

Por otro lado, se puede afirmar que de los 1438 tweets, 95 (6,61 %) están clasificados como seguridad (Figura 6.3) usando los métodos descriptos en la sección anterior.

Total	Solo Seguridad	Indice de Seguridad
1,438.0	95.0	6.61%

Figura 6.4: Estadísticas de seguridad sobre tweets que hablan de la LUC

En la Figura 6.5 pueden verse las distribuciones de publicaciones según fuentes. La Diaria fue quien más publicó sobre este tema, seguido de La República.

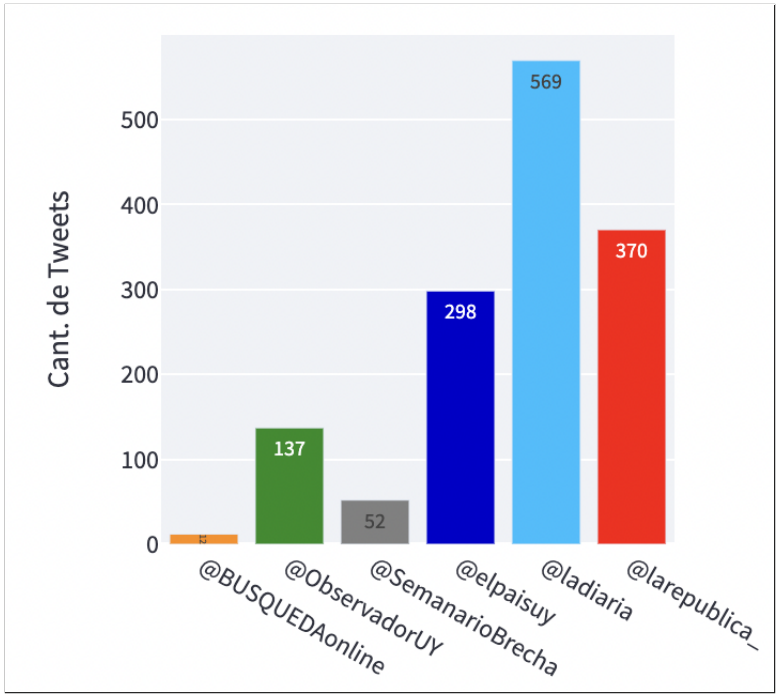

Figura 6.5: Distribución por cuentas

De la Figura 6.6 se desprende que, a pesar de no ser el medio que más publicaciones hace sobre el tema, El País es el que logra captar más la atención de los usuarios siendo el que más comentarios recibe en sus publicaciones.

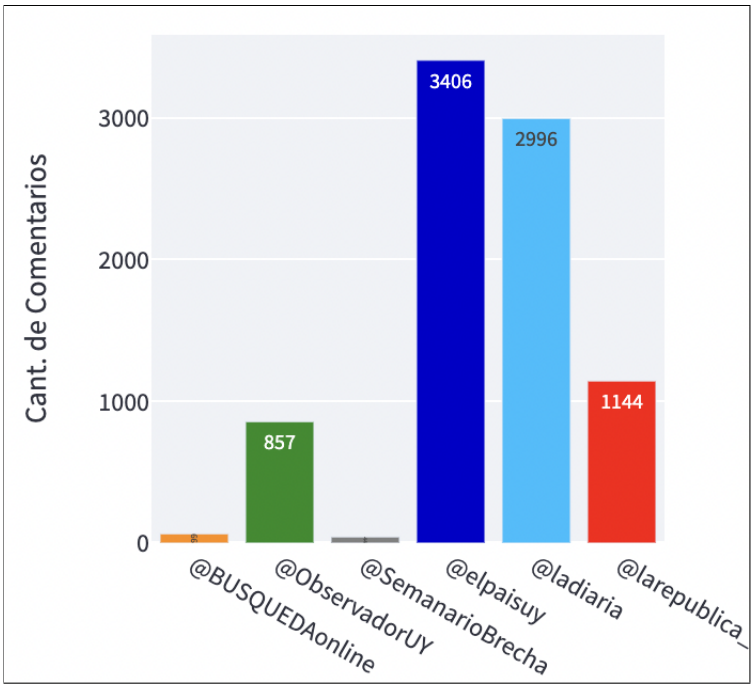

Figura 6.6: Cantidad de comentarios por publicación

6.2. Entidades nombradas

En la sección de entidades nombradas se encuentran varias cosas a comentar.

Personas

Las primeras doce personas más nombradas pueden verse en el cuadro de la Figura 6.7. Se ve cómo el algoritmo utilizado detecta a las entidades *"Lacalle Pou"*, *"Luis Lacalle Pou"* y *"Lacalle"* como si fueran distintas. Si bien las últimas dos podrían aludir a distintas personas (Lacalle padre y Lacalle hijo), se entiende que las primeras dos refieren a una misma persona. Por otro lado, en cuanto a proporciones de género, de las diez personas distintas detectadas, sólo dos son mujeres (*Argimón* y *Cosse*) frente a los restantes ocho varones. No hay representación alguna de otro género o disidencias.

Entidad	Ocurrencias
Lacalle Pou	49
Fernando Pereira	37
Argimón	21
Andrade	18
Gandini	17
Lacalle	16
Luis Lacalle Pou	16
Michelini	15
Cosse	14
Manini Rios	13
Miranda	11
Marcelo Abdala	10

Figura 6.7: 12 personas más nombradas

Organizaciones

En cuanto a las organizaciones, nuevamente se detectaron algunas irregularidades que no impidieron hacer un estudio de los datos: “FA” y “Frente Amplio”, son detectadas como distintas (esto se explica, naturalmente, ya que el algoritmo no cuenta con ningún tipo de resolución de correferencias) así como ocurre con “PIT-CNT” y “Pit-Cnt”. Más allá de los mencionados detalles puede decirse que las organizaciones más relacionadas con la LUC han sido el partido Frente Amplio junto con la organización sindical PIT-CNT. (detalle Figura 6.8)

Entidad	Ocurrencias
FA	194
Frente Amplio	96
PIT-CNT	93
Parlamento	86
Senado	56
Pit-Cnt	36
Diputados	32
Cabildo Abierto	28
Intersocial	28
Partido Colorado	25

Figura 6.8: 10 principales organizaciones

Lugares

Por último, dentro de la sección de entidades nombradas se encuentran los lugares. Las principales 10 son las que muestra la Figura 6.9.

Entidad	Ocurrencias
Uruguay	24
Montevideo	15
Maldonado	8
Colonia	6
Palacio Legislativo	6
Hospital de Clínicas	4
Paysandú	4
La LUC	3
Cerro Largo	3
Argentina	3

Figura 6.9: 10 principales lugares

De esta lista podemos ver que *"Palacio Legislativo"* y *"Hospital de Clínicas"* son detectados como lugares aunque en otras circunstancias podrían verse como organizaciones. Es importante tener en cuenta que este tipo de dualismos podrían afectar las clasificaciones.

6.3. Detección de clusters y tópicos

Los clusters son agrupaciones de vectores según cercanía en el espacio que, como se describió anteriormente, comparten semántica por la forma en que se construyen los word embeddings. Una vez que se detectan los clusters, el sistema los enumera para distinguirlos y ser referenciados si se necesitara. En este caso se seleccionan 100 clusters. Pueden verse varios clusters interesantes:

Cluster 4 (160 noticias)

La mayoría hablan de posturas de distintos analistas sobre la LUC. Las noticias representativas de este cluster se muestran en la Figura 6.10.

Noticias representativas

Descargar Noticias

	Handle	Tweet	Attachment	PostDate
0	@ladiaria	Trabajo y vivienda en la LUC: desregulación, concentración y represión. La #postura de @DanielOlesker https://ladiaria.com.uy/articulo/2020/5/trabajo-y-vivienda-en-la-luc-desregulacion-concentracion-y-represion/…		2020-05-27
1	@ladiaria	Acción y reacción: propiedad privada y legítima defensa en la LUC. La #postura de Marcos Hernández https://ladiaria.com.uy/articulo/2020/5/accion-y-reaccion-propiedad-privada-y-legitima-defensa-en-la-luc/…		2020-05-29
2	@ladiaria	La inconstitucionalidad de la LUC en materia de adolescentes infractores; la #postura de Pablo Rodríguez Almada https://ladiaria.com.uy/articulo/2020/5/la-inconstitucionalidad-de-la-luc-en-materia-de-adolescentes-infractores/…		2020-05-23
3	@ladiaria	El orden público laboral y la LUC; la #postura de Fabrizio Bacigalupo https://ladiaria.com.uy/articulo/2020/6/el-orden-publico-laboral-y-la-luc/…		2020-06-17
4	@ladiaria	Enfrentar la LUC, ¿y después? La #postura de Héctor Altamirano https://ladiaria.com.uy/opinion/articulo/2020/12/enfrentar-la-luc-y-despues/…		2020-12-28

Figura 6.10: Noticias representativas del cluster 4

Cluster 8 (34 noticias)

El cluster 8 agrupa muchas noticias que tienen que ver con la recolección de firmas y el referéndum (Figura 6.10).

	Handle	Tweet	Attachment	PostDate
0	@ladiaria	Intersocial Feminista apoyará recolección de firmas para referéndum contra la LUC	Intersocial Feminista apoyará recolección de firmas para referéndum contra la LUC El colectivo se suma a lo propuesto por el PIT-CNT y planteará derogar algunos artículos de la norma. ladiaria.com.uy	2020-10-12
1	@ladiaria	Comisión Prorreferéndum contra la LUC: no postergar plazo para reunir las firmas “conspira contra la participación ciudadana”	Comisión Prorreferéndum contra la LUC: no postergar plazo para reunir las firmas “conspira contra... La comisión llamó a la militancia a redoblar esfuerzos para cumplir con el objetivo “garantizando los cuidados sanitarios” ladiaria.com.uy	2021-04-23
2	@ladiaria	El próximo lunes comenzará a trabajar la Comisión Nacional Pro Referéndum contra la LUC	El próximo lunes comenzará a trabajar la Comisión Nacional Pro Referéndum contra la LUC Marcelo Abdala aseguró que hay confianza en que se van a alcanzar las firmas. ladiaria.com.uy	2020-12-08
3	@ladiaria	FA resolvió avanzar por el “camino largo” para recolectar firmas contra la LUC, que implica impugnar los artículos "más perjudiciales" #LaDiariaFinDeSemana	FA resolvió avanzar por el “camino largo” para recolectar firmas contra la LUC El lunes el Secretariado analizará qué artículos se impugnarán. ladiaria.com.uy	2020-10-24

Figura 6.11: Noticias representativas del cluster 8

Cluster 86 (32 noticias)

El cluster 86 concentra varias noticias que tienen que ver con el Frente Amplio o actores del partido (Figura 6.11).

@elpaisuy	Miranda sobre la LUC: "La inconstitucionalidad más importante es la utilización de un mecanismo constitucional con un fin que no está previsto en la Constitución"	Impulsar un referéndum contra la LUC "está dentro del menú de lo posible", dijo Miranda "Hay un problema de oportunidad, de contenido y constitucional", subrayó el presidente del Frente Amplio, Javier Miranda. elpais.com.uy	2020-04-24T20:57:13+00:00
@elpaisuy	La senadora frenteamplista puntualizó sobre el artículo 44 del nuevo proyecto y en el 268 de la LUC y opinó: "Esto es para desguazar a Antel".	Cosse dijo que nuevo proyecto de Ley de Medios pretende "desguazar" a Antel La senadora frenteamplista puntualizó sobre el artículo 44 del nuevo proyecto y en el 268 de la LUC y opinó: "Esto es para desguazar a Antel". elpais.com.uy	2020-04-24T16:17:43+00:00
@larepublica_uy	Frente Amplio: urgente es enfrentar la situación sanitaria de nuestra población El FA reitera que considera un “profundo error” la entrada formal del proyecto de LUC, considera que “atenta contra el esfuerzo necesario para enfrentar la crisis sanitaria.	Frente Amplio: urgente es enfrentar la situación sanitaria de nuestra población | Diario La... El Frente Amplio (FA) emitió un comunicado en el que rechaza la decisión del presidente de la República, Luis Lacalle Pou, de enviar el proyecto de ley de urgente consideración a los legisladores de... republica.com.uy	2020-04-11T14:35:04+00:00
@elpaisuy	"Mala señal", "inentendible y preocupante", "lo único urgente es la salud de nuestros compatriotas", "no es momento", "el presidente elige separar y no unir", "exhibición de fuerza", "no ayuda" y más reacciones del FA por envío de la #LUC al Parlamento	Así reaccionó la oposición al inminente envío de la Ley de Urgencia al Parlamento El presidente Luis Lacalle Pou anunció este jueves que enviará a todos los legisladores la ley de urgente consideración antes de presentarlo formalmente en el Parlamento. elpais.com.uy	2020-04-09T21:45:22+00:00

Figura 6.12: Noticias representativas del cluster 86

En el presente capítulo se mostró un flujo completo de utilización de la aplicación. Para ver un análisis completo de los datos y un estudio de cada medio por separado ver *Apéndice C: Análisis del conjunto de datos.*

Capítulo 7

Conclusiones y trabajo futuro

Para cumplir con el objetivo principal del proyecto se tuvo que realizar cuatro grandes pasos: construir el conjunto de datos, entrenar un modelo de word embeddings, entrenar el clasificador de tweets de seguridad, y por último, implementar una aplicación web para visualizar la información.

Para el primero de estos pasos, la idea principal fue utilizar la API de Twitter, pero se encontró el obstáculo de que su versión gratuita solo permite descargar hasta 3200 tweets por cuenta. Por tal motivo, se desarrolló un scraper de Twitter. A su vez, la implementación de este software permitió enriquecer el producto entregable y ayudar con una nueva herramienta a los sociólogos, a través de un sistema que les permite descargar tweets de cualquier rango de fechas, cuenta y temática, facilitando su trabajo y ahorrando mucho tiempo.

Luego de tener el scraper pronto, se pudo obtener la totalidad de los tweets de los medios seleccionados, y comenzar el preprocesamiento para el entrenamiento del modelo. Para definir los distintos preprocesamientos, nos inspiramos en los trabajos similares encontrados. Por otro lado, se vio una gran diferencia en los resultados entre los modelos con *Word2Vec* puro y los entrenados con *Word2Vec enriquecido con subpalabras*. De los modelos entrenados con *Word2Vec enriquecido* se tiene el entrenado con *CBOW* que presenta un mayor valor de recall, y el entrenado con *Skip-gram* que presenta un mejor valor de precisión. Esto permitió interiorizarse más en las medidas de evaluación existentes y cuál era la más importante para el objetivo planteado. En este proyecto lo que interesa más es poder detectar la mayor cantidad de tweets de seguridad, por lo que el recall resulta más importante que la precisión. Por esto último, se eligió el modelo entrenado con *Word2Vec enriquecido* con arquitectura *CBOW*.

Por último, se tuvo que aprender no sólo las distintas formas de representación de los datos obtenidos, sino también cómo mantener la base de datos actualizada día a día.

También se investigaron varias alternativas que fueran atractivas y útiles a la hora de presentar los datos utilizando bibliotecas que permitieron desarrollar la interfaz gráfica de una forma fácil y amigable. De esta parte surgen posibles mejoras detalladas en la sección correspondiente.

Se logró construir una herramienta que los interesados entienden será de mucha utilidad para futuros trabajos e informes sobre la temática de seguridad. Al contar con diversos filtros, tanto de fecha como medios, "seguridad" o "no seguridad" (entre otros), permite adaptar los datos a las necesidades que tenga quien esté utilizando la aplicación.

Con estos pasos se logró cumplir con el objetivo de presentar una herramienta que permita el análisis y seguimiento de la temática de seguridad en Twitter. Se entregó un scraper que permitirá obtener información para futuras investigaciones tanto de la misma temática como de otras, y se pudo entrenar un modelo con muy buenos resultados para discernir si un tweet habla o no de seguridad.

7.1. Resultados

A continuación se listan resultados concretos obtenidos al finalizar el presente trabajo. Se obtuvo:

- Una plataforma[9] funcional desplegada en la nube para el análisis de noticias locales que permite a sociólogos identificar temas recurrentes en ellas, identificar los actores de los que se habla así como lugares donde ocurren los hechos. Por otro lado, se puede analizar el interés de determinados temas a lo largo del tiempo utilizando filtrado de noticias por palabras clave.
- Un scraper adaptado a Twitter para que los sociólogos puedan filtrar por otros medios o palabras clave, y luego descargar a su computadora los tweets de las cuentas que seleccionen en los rangos de fechas que deseen.
- Un conjunto de datos de noticias formado por las publicaciones del conjunto de medios seleccionado para el trabajo de grado, junto con información relevante (como NERs, cantidad de likes y comentarios) desde la creación de las cuentas hasta la fecha. Este conjunto se publicó de forma abierta en la plataforma Kaggle[10] para facilitar su accesibilidad a cualquiera que desee utilizar los datos para futuros trabajos.
- Un modelo no supervisado de word embeddings entrenado con el conjunto de datos construido. Este conjunto mismo también fue publicado pero en la plataforma HuggingFace[11] para facilitar su acceso y reutilización por la comunidad, ya que es

[9]Plataforma web: http://198.211.117.226/
[10]https://www.kaggle.com/datasets/leadominguez/uruguayan-media-historical-tweets
[11]https://huggingface.co/leandrodzp/cbow_uruguayan_news

sabido que no es fácil encontrar recursos de este tipo para lenguajes no angloparlantes y mucho menos en el contexto local o regional.

7.2. Trabajo futuro

En esta sección se puntualizan propuestas que permitirán mejorar el trabajo realizado hasta el momento, o agregar nuevas funcionalidades complementarias a las existentes.

Mejoras al conjunto de datos

Como fue mencionado anteriormente el conjunto de datos fue construido a partir de los tweets publicados por cierto conjunto de medios. Una posible mejora para el conjunto de datos sería adaptar la solución propuesta para obtener también los tweets de otros medios.

Por otro lado, también sería interesante poder acceder a los comentarios de cada tweet ya que eso permitiría hacer un estudio de las reacciones de los usuarios a cada uno. Esto es una de las pautas que quedaron fuera del alcance y que permitiría detectar para cada tweet el porcentaje de reacciones positivas y negativas que se tienen. Este último punto, formaba parte de la idea principal pero tuvo que ser postergado por priorización de otras tareas.

Otra mejora que se plantea para futuro es poder ampliar el origen de los datos y poder scrapear las noticias completas, ya que algunos tweets son simples titulares con muy poca información relevante. Poder acceder a la noticia completa permitiría mejorar el aprendizaje del modelo y la precisión al momento de etiquetar los tweets así como obtener información más completa de la noticia como una lista más extensa de entidades nombradas.

Un problema que poseen los datos hoy en día es que la cantidad de likes, retweets y comentarios muestra el estado en el momento en el cual el tweet fue descargado sin tener en cuenta cómo varían con el paso del tiempo. Por esto, sería bueno buscar un mecanismo que pueda actualizar los valores de likes, comentarios y retweets sin la necesidad de volver a descargar el tweet por completo.

Mejoras en la interfaz de usuario

No se dedicó casi tiempo al estudio de la interfaz de usuario. En este punto hay gran margen de mejora que podría hacer más entendibles las distintas funcionalidades. Una parte que genera confusión en los usuarios que utilizan la plataforma por primera vez

resulta ser el manejo de parámetros y específicamente la combinación de "cantidad de tópicos" y "cantidad de clusters" en la sección de detección de tópicos.

Personalización por usuarios

Para mejorar la experiencia de usuario, una posibilidad a futuro es que la aplicación tenga manejo de usuarios y que puedan loguearse y tener nuevas herramientas para trabajar en la web. Un ejemplo de esto sería la posibilidad de guardar marcadores con filtros prefijados. De esta forma, al momento de volver a realizar una búsqueda, no sería necesario configurar todos los filtros nuevamente sino simplemente seleccionar un marcador. También podría permitirse tener un historial de las últimas búsquedas realizadas, facilitando así el uso recurrente de la herramienta.

Mejoras de performance

La aplicación web desarrollada podría ser optimizada para que la performance y los tiempos de espera sean mejorados. Una posible mejora podría ser en relación a la carga de los datos. Allí se podría buscar la forma de resolver algunas consultas directamente en la base de datos en vez de resolverlo en el momento de la ejecución. También se debería buscar alguna tecnología que permita comprimir los datos para que ocupen menos espacio, y por lo tanto, sea más rápido el procesamiento, la carga y su descarga.

Mejoras en el procesamiento y preprocesamiento

El procesamiento de los datos es una de las tareas más importantes realizadas en este trabajo y tiene aún mucho para poder mejorar. Una mejora al preprocesamiento sería utilizar algún método de tokenización más sofisticado que reconozca multipalabras. De esta forma, se atacaría el problema que existe con palabras que en sí se comportan como una sola, como por ejemplo "Luis Lacalle Pou" o "Estados Unidos". También referido a las entidades nombradas, queda pendiente realizar un linkeo de las mismas para poder asociar aquellas entidades diferentes que refieren a la misma persona u organización. Ejemplos son "José Mujica" y "Pepe Mujica" que aparecen como si fueran distintas personas cuando claramente no lo son.

Mejoras en las representaciones vectoriales

Los word embeddings son una parte fundamental del presente trabajo ya que son las entradas de los modelos predictivos. En este sentido es que sería de gran utilidad probar

otras implementaciones y comparar resultados con la actual. Una opción sería utilizar un modelo pre-entrenado con un corpus de mayor dimensión como lo es BERT.

Por otro lado, en lugar de usar promedios de vectores para representar oraciones, podrían utilizarse Sentence Embeddings.

Otras mejoras

Otra mejora que se le podría realizar al trabajo sería profundizar en las características que tiene una red social aprovechando la estructura de grafo que se genera entre seguidores y seguidos. Se podría por ejemplo, utilizar esa estructura para trabajar con las similitudes y diferencias que presentan los seguidores de una cuenta, o quienes reaccionaron o comentaron una noticia. También se podrían analizar las conductas e intereses de quienes siguen a cada una de las cuentas o si existe alguna relación entre quienes comentan positiva o negativamente a alguna noticia.

Bibliografía

[1] Dan Jurafsky y James H Martin. *Speech and Language Processing. 3rd.* 2022.

[2] Tom Mitchell y Machine Learning McGraw-Hill. *Machine Learning.* 1997.

[3] Lester Marrero y col. "Uso de algoritmo K-means para clasificar perfiles de clientes con datos de medidores inteligentes de consumo eléctrico: Un caso de estudio". En: *Ingeniare. Revista chilena de ingeniería* 29 (dic. de 2021).

[4] Susan Li. *Reconocimiento de Entidades con Nombre.* [Online; Último acceso Julio-2022]. 2018.

[5] *Latent Dirichlet Allocation.* [Online; Último acceso Julio-2022]. 2018.

[6] G. Casella, R.L. Berger y Brooks/Cole Publishing Company. *Statistical Inference.* Duxbury advanced series in statistics and decision sciences. Thomson Learning, 2002. ISBN: 9780534243128.

[7] Luciano Hammoe. "Detección de tópicos: utilizando el modelo LDA". En: (2018).

[8] Zellig S. Harris. "Distributional Structure". En: *WORD* 10.2-3 (1954), págs. 146-162.

[9] Radim Řehřek, Petr Sojka y col. "Xplore Word Embedding Using CBOW Model and Skip-Gram Model". En: *Retrieved from genism. org* (2011).

[10] Piotr Bojanowski y col. *Enriching Word Vectors with Subword Information.* 2017.

[11] Armand Joulin y col. "Bag of tricks for efficient text classification". En: *arXiv preprint arXiv:1607.01759* (2016).

[12] Alan Ritter, Sam Clark, Oren Etzioni y col. "Named entity recognition in tweets: an experimental study". En: *Proceedings of the 2011 conference on empirical methods in natural language processing.* 2011.

[13] Soroush Vosoughi, Prashanth Vijayaraghavan y Deb Roy. "Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder". En: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval.* 2016.

[14] Bo Huang y col. "Microblog topic detection based on LDA model and single-pass clustering". En: *International Conference on Rough Sets and Current Trends in Computing.* Springer. 2012.

[15] Christian Wartena y Rogier Brussee. "Topic detection by clustering keywords". En: *2008 19th international workshop on database and expert systems applications*. IEEE. 2008.

[16] Radim Řehřek, Petr Sojka y col. "Gensim—statistical semantics in python". En: *Retrieved from genism. org* (2011).

[17] Joaquın Amat Rodrigo. *Análisis de componentes principales (principal component analysis, pca) y t-sne*. 2017.

Apéndice A

Palabras Simplificadas

Lista de palabras simplificadas utilizadas para filtrar el corpus antes de etiquetar como `seguridad` o `no_seguridad`.

Palabras				
usurpar	operativo	procesó	acusados	violencia doméstica
amenaza	ocupantes	cuerpo	atracador	insultos
murió	dispersar	condenaron	juzgado	niño
denuncia	disturbios	organización delictiva	agredidos	allanamientos
narcotrafico	pericias	homicidios	denuncias	incidentes
puñetazos	formalizan	asesinato	bala	arrestado
copamiento	puñaladas	imputó	agredida	detenida
represivas	heridas	desalojada	asesinan	asesinatos
agresiones	rapiñar	asesinados	muerta	víctima
golpeado	aumentar penas	delinquir	atacan	siniestro
incomunicado	muerte brutal	operativos policiales	ladrón	plaza seregni
allanmiento	rapiñas	imputados	operación	cuerpo en descomposición
combate	violadores	policía	represiva	delincuentes
detuvieron	estafa	encerrar	matado	femicidio
atropellado	asalto	abusadores	violento	formalización
fuerzas armadas	víctima	arrojaron de un puente	incidente	policiales
manifestación	baleados	matanzas	cárcel	manifestantes

policiamiento	disidentes	delincuente	disparó	operación murmullo
rotura	odio	condenaron	narcotráfico	caída del gobierno
Denuncia	Ministerio del interior	homicidio	rehén	enemigo
operación océano	usurpación	robar	agredió	abigeato
atacados	paliza	testigos protegidos	joyería	asenidados
imputaron	golpes	condenadas	recluso	abatidos
reppresión	investiga	robo	policia	posesión
femicida	golpiza	atacante	cómplice	penitenciaria
asesinada	intervención policial	asesinó	hematomas	capturado
personas ausentes	dispersó	contrabando	secuestró	mataron
denunció	procesan	operación océno	protestas	terrorista
muertos	ataque	caos	asesinaron	exceso policial
disparos	cifras	tropas	detenidos	mató
justicia	desacato	rapiñero	condenan	manifestaciones
rapiñaron	control	privado de libertad	evidencia	disparar
operativos	pena	violencia sexual	cautelar	MI
baleado	formalizaron	rehenes	ajuste de cuentas	pelea
copó	abuso	destrozos	espirometría	prender fuego
explotación sexual	trabajo policial	hurtos	Bonomi	medidas cautelares
operación gallego	amenazar	narco	condena	detención
roba	violadres	comcar	muerte	rapiña
atacada	operación el fogón	herido de bala	chocó	represión
disparo	identificador facial	golpeado	condenó	tiroteo
intervenciones policiales	encubrimiento	requerido	detenidas	inseguridad

indicentes	policías	procedimiento	pandillas	policial
incendian	detuvo	robó	incautación	hurto
agresión	condenado	testigo	abusadores sexuales	explotación de menores
racista	agredir	amenzó	agrede	crimen
delicuentes	secuestro	seguridad	sospechas	secuestrador
represión policial	condenar	tentativa	militares	investigan
atropello	culpables	condenados	atropellos	excesos policíales
imputan	herir	balas	detenido	amenazó
muertes	peligrosa	heridos	homicida	delito
consecuencias	prisión	imputado	emplazado	Lola Chomnalez
desalojo	estallido social	enjuiciamiento	balearon	enfrentamiento
preocupación	protestan	indagado	autopsia	detenciones
revuelta	herido	matar	quemarropa	tensión
accionar	pena máxima	falleció	procesado	refugiados
abuso policial	punta de pistola	incautó	usurpaciones	allanamiento
detienen	robos	operativos represivos	violencia de género	multado
acoso	persecusión	arma	boca de drogas	víctimas
reprimir	manifestaron	abuso sexual	violencia	conflicto
guerra	asesinado	altercado	atacó	genocidio
operación hack	muerto	dispararon	confuso incidente	amenazaba
droga	baleó	baja delitos	restos encontrados	hirió
género	investigado	narcotráfico	fosa común	robar
ataques	atentado	delitos	calcinado	violento desalojo

Apéndice B

Despliegue de aplicación

Para que nuestra aplicación de visualización de datos sea accesible públicamente se configuró un servidor donde se corren los siguientes servicios:

- App de Streamlit[12] (frontend de la app y dashboard interactivo).
- Servicio de MongoDB como principal base de datos donde se persisten los tweets junto con información enriquecida y calculada al momento de procesamiento (NERs, cantidad de likes, comentarios, etc).
- Servicio de Elasticsearch donde se guardan principalmente representaciones vectoriales de los tweets para su consulta y recuperación de forma eficiente utilizando la distancia coseno.
- Servidor Nginx como proxy inverso.

Se utilizó Docker[13] para configurar y coordinar todos los servicios mencionados de manera sencilla y reproducible. De esta manera, se definió un contenedor para cada uno de los puntos mencionados anteriormente y la forma de comunicación entre ellos usando docker compose.

Los servicios se encuentran en un servidor de Digital Ocean[14] (droplet) con las siguientes especificaciones:

- 4 vCPUs
- 8GB RAM
- 80GB Almacenamiento

Además se configuró el crontab del servidor para que todos los días se corra un script encargado de descargar los últimos tweets de los medios de comunicación, clasificarlos y guardarlos tanto en la base de datos MongoDB como en ElasticSearch.

[12]https://streamlit.io/
[13]https://www.docker.com/
[14]https://www.digitalocean.com/

Apéndice C

Análisis del conjunto de datos

En este anexo se hace un análisis cuantitativo del conjunto de datos haciendo énfasis en estadísticas generales sobre cantidad de información, y qué tanta es sobre seguridad, viendo las relaciones entre las distintas cuentas. Los resultados que se presentan fueron construidos y calculados con nuestra propia herramienta.

Descripción general

El conjunto de datos fue construido a partir del total de tweets publicados por las seis cuentas seleccionadas, contemplando desde la primera publicación de cada una de ellas, hasta mediados de mayo de 2021. La excepción fue El Observador, donde se tomaron los tweets a partir de abril de 2016.

Se deben tener presentes varias consideraciones, la primera tiene fuerte relación con el pasar del tiempo y el aumento en popularidad de la red Twitter, ya que a medida que pasan los años, los medios de comunicación han tomado a esta red social como medio de primicias y actualización constante, provocando que cada día se realicen más publicaciones.

En los primeros 2 años que cubre el conjunto de datos (entre abril de 2009 y abril de 2011) se publicaron 16.900 tweets (Figura H.1), mientras que en los últimos años este número ha aumentado considerablemente. Por ejemplo, entre abril de 2019 y abril de 2021 la cantidad de publicaciones fue de 259.013 tweets (Figura H.2) lo que representa un aumento de más de 1500

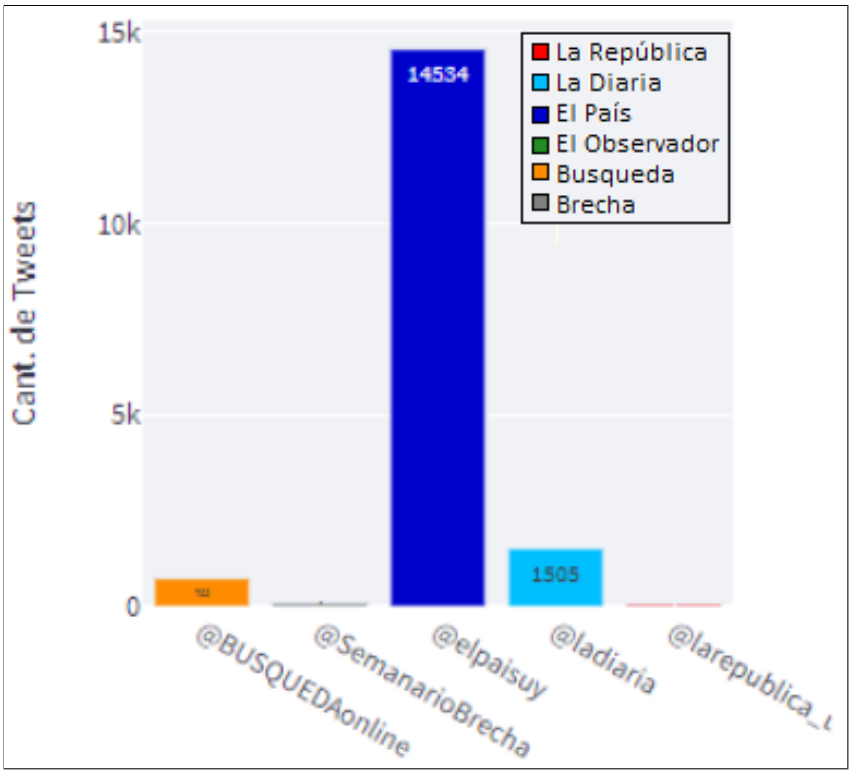

Figura C.1: Cantidad de tweets por cuenta entre abril 2009 y abril 2011

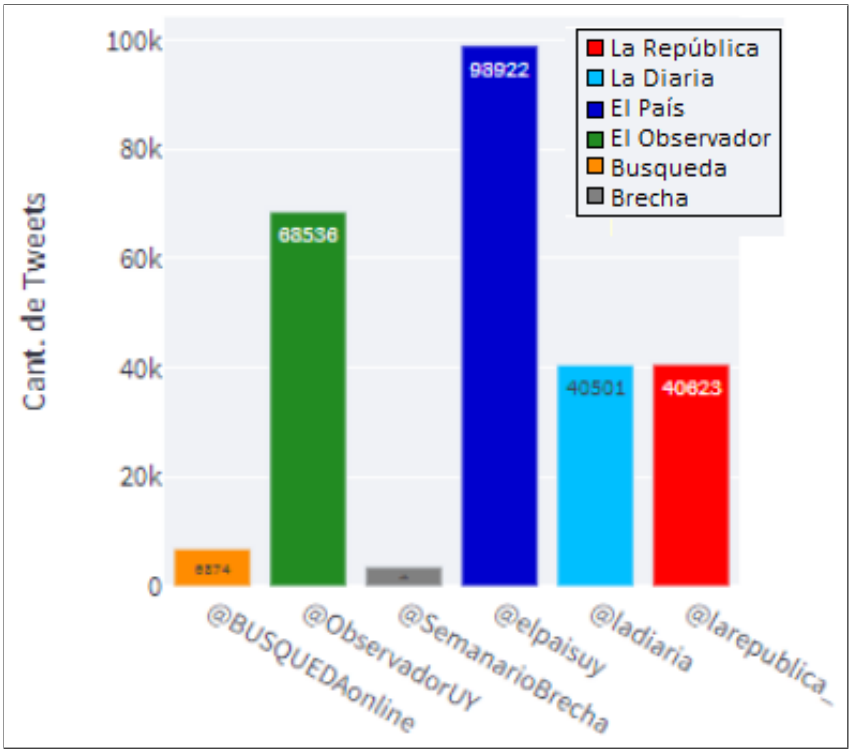

Figura C.2: Cantidad de tweets por cuenta entre abril 2019 y abril 2021

Otra consideración a destacar, es la cantidad de publicaciones que realiza cada cuenta. Si se ve el acumulado hasta el 30 de abril de 2021 que tiene cada cuenta, se puede apreciar

que El País ha publicado casi el 40

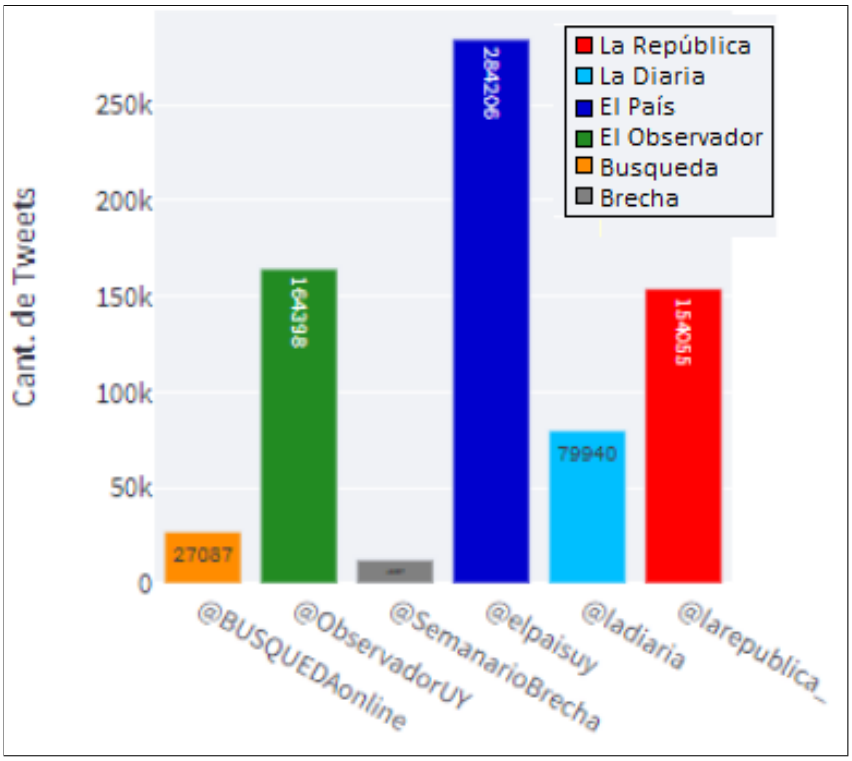

Figura C.3: Tweets publicados hasta abril 2021

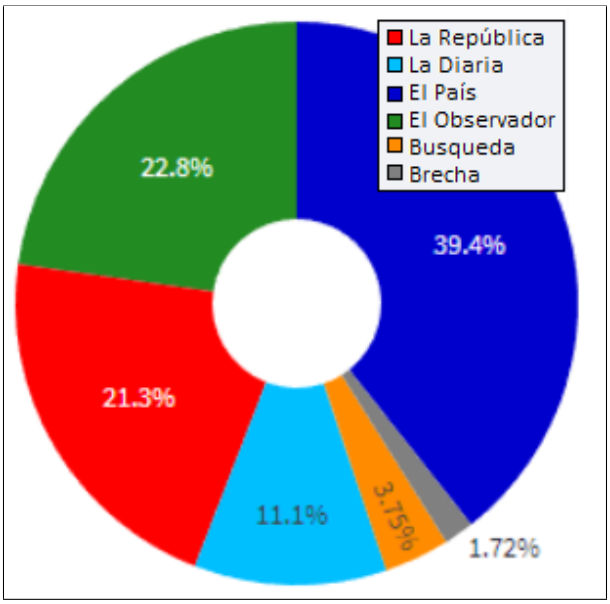

Figura C.4: Distribución por cuenta de los tweets publicados hasta abril 2021

Este comportamiento también se puede ver reflejado en los tweets etiquetados como de seguridad, donde casi la mitad han sido publicados por el diario El País (Figura H.5), y de los restantes tweets, más del 90 % fue publicado entre La República, El Observador y La Diaria (Figura H.6)

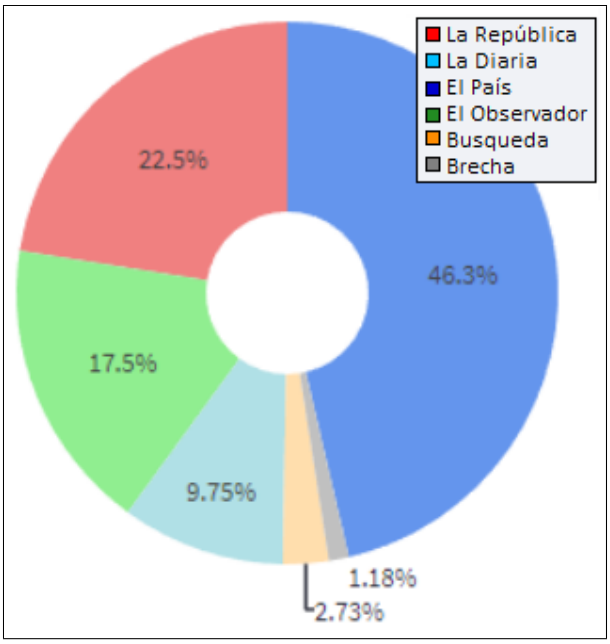

Figura C.5: Distribución por cuenta de los tweets de seguridad publicados hasta abril 2021

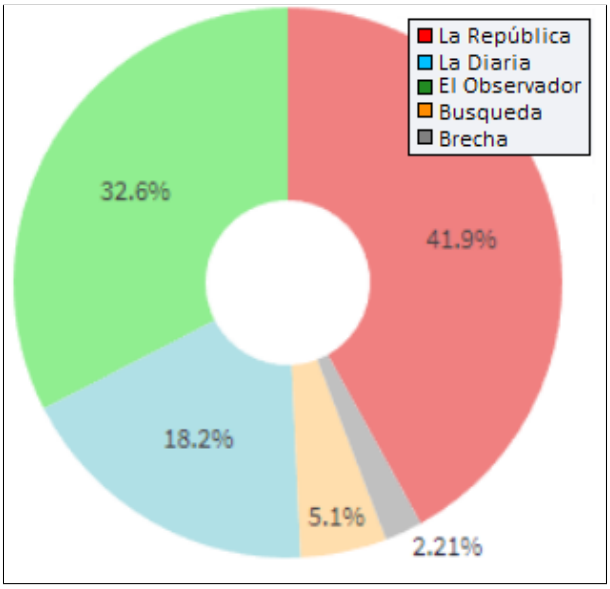

Figura C.6: Distribución por cuenta de los tweets de seguridad publicados hasta abril 2021 sin El país

Respecto a los porcentajes de tweets sobre seguridad, si se mira la tendencia se ve que al inicio, los porcentajes eran muy variables. Debido a que no se twitteaba mucho, cuando se publicaba un tweet sobre seguridad, este influía bastante en el porcentaje. Con el pasar del tiempo y el aumento de las publicaciones, se puede ver que los mismos tienden a estabilizarse, en un margen de entre 6 % y 10 % mensual (Figura H.7).

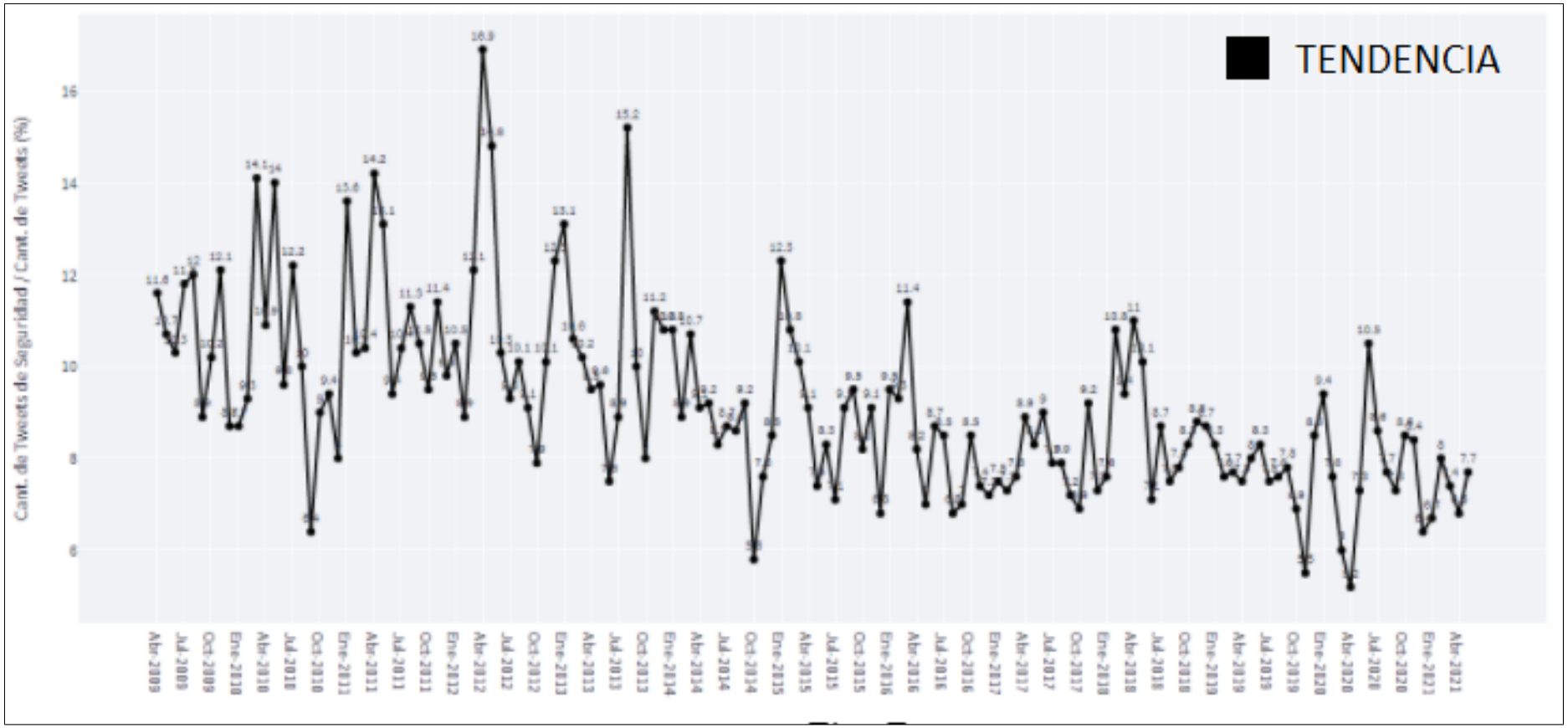

Figura C.7: Tendencia del índice de seguridad de los tweets publicados hasta abril 2021

Descripción por cuenta

El País

Como fue mencionado anteriormente, la mayor cantidad de tweets pertenecen a la cuenta del diario El País. En esta cuenta hay 284.206 tweets, de los cuales 27.890 fueron etiquetados como *seguridad*, lo que representa un 9.81 % del total. El primer tweet fue publicado el 15 de abril de 2009. El mes en el que más tweets publicó fue en noviembre de 2020 con 5.820 tweets (Figura H.8). En este mismo mes se dio el récord sobre tweets de seguridad, publicando un total de 515 (Figura H.9). El índice de seguridad más grande lo tuvo en abril de 2012, donde de un total de 1.031 tweets, el 21,7 % fueron de seguridad, mientras que el valor más bajo se dio en abril de 2020 con solo un 4,8 % (Figura H.10).

Semanario Búsqueda

Hasta el 31 de diciembre de 2010 no se cuenta con tweets de otros medios. Recién el primero de enero de 2011 aparece el primer tweet del Semanario Búsqueda. Este es uno de los medios que menos utiliza la red social. Tan solo ha publicado 27.087 tweets en algo más de 10 años. Por otro lado, el índice de seguridad acumulado es de 6,09 % ya que solo 1.649 tweets de los publicados han sido etiquetados como de seguridad.

A diferencia de los demás medios, el récord de tweets publicados por Búsqueda se dio al comienzo de sus publicaciones, en julio de 2012 donde publicó 497 tweets (Figura H.11) y en mayo de ese mismo año publicó 61 tweets de seguridad (Figura H.12), siendo esta su mayor marca y por un gran margen, ya que ni antes ni después ha vuelto a superar los 45 tweets de seguridad. En ese mismo mes, se dio también el máximo del índice de seguridad siendo 14.5 %. Luego el índice ha sido muy variado, teniendo algunos meses donde no se publicó ninguno sobre seguridad y por lo tanto el índice fue 0 % (Figura H.13).

La Diaria

Temporalmente, el tercer medio que aparece en el conjunto de datos es La Diaria, que desde el 3 de enero de 2011 hasta mediados de mayo 2021, ha publicado 79.940 tweets con 5.876 etiquetados como de seguridad. Esto equivale a un 7.35 %. La Diaria ha crecido mucho en el uso de Twitter, convirtiéndose en la actualidad en uno de los medios que más publicaciones realiza, muestra de esto es que en el último año y medio, publicaron más de un tercio de los tweets publicados en total. En julio de 2020 el medio publicó 357 tweets de seguridad alcanzando su máximo (Figura H.14), ese mismo mes también publicó un total de 2.796 tweets, siendo su segunda mayor marca, superado sólo por el mes de marzo

de 2021, donde las publicaciones alcanzaron los 2.864 tweets (Figura H.15).

Respecto al índice de seguridad, el comportamiento de La Diaria ha sido bastante estable, salvo en sus primeros años de utilización de la red, donde tuvo meses con 0 tweets de seguridad, y un mes, donde de solo 11 publicaciones, 4 fueron de seguridad representando el 36.4 % (Figura H.16). Este porcentaje de tweets de seguridad es el mayor registro entre todas las cuentas.

La República

La República publicó su primer tweet el 17 de marzo de 2011 y es el segundo medio con más tweets de seguridad en el conjunto de datos, y el tercero, en cantidad de tweets totales. De los 154.055 tweets publicados por este medio, un 8.77 % son de seguridad, lo que supone 13.513 publicaciones.

El comportamiento de publicación de esta cuenta, es un poco diferente al resto; no es tan pronunciado el aumento de publicaciones con el pasar de los meses. Entre fines de 2014 y todo el 2015 tuvo un aumento considerable en la cantidad de publicaciones que luego cayó abruptamente. En el período antes mencionado fue cuando se dieron los máximos de publicaciones tanto en total de tweets como en tweets de seguridad. Estos números fueron 344 en septiembre de 2015 (Figura H.17) y 3.527 en octubre de 2014 (Figura H.18) respectivamente.

El comportamiento en cuanto al índice de seguridad, tuvo su pico máximo el segundo mes de publicaciones con 23,5 %, luego se ha mantenido relativamente estable (Figura H.19).

Semanario Brecha

El semanario Brecha, al igual que Búsqueda, realiza muy pocas publicaciones, en este caso se cuentan con 12.397 tweets desde el primero de abril de 2011. De todos esos tweets, solo 713 fueron etiquetados como de seguridad, lo que representa un 5,75 %. El comportamiento de Brecha en la red social, en la actualidad es bastante estable. Tuvo un crecimiento constante desde su llegada hasta fines de 2017, pero luego se mantuvo relativamente estable entre los 100 y los 200 tweets mensuales. La máxima cantidad de tweets publicados fue en mayo de 2018 con 275 (Figura H.20), mientras que la mayor cantidad de tweets de seguridad, se dio en agosto de 2013 (Figura H.21). Esto último representó el máximo índice de seguridad que volvió a repetirse posteriormente un año después, pero en este caso con 8 tweets de seguridad en un total de 44 (Figura H.22).

El Observador

Pese a contar con los tweets de El Observador desde el 10 de abril de 2016, y no desde su primera publicación como el resto de los medios, es el segundo con mayor cantidad de tweets y el tercero con mayor cantidad de tweets de seguridad, teniendo un acumulado de 6,43 % en el índice de seguridad. Si se toman en cuenta las publicaciones de todos los medios desde la misma fecha, los 164.398 tweets de El Observador representan el 30.1 % del total, y los 10.564 tweets de seguridad, el 24.5 %. Se observa que tiene un ritmo de publicación muy cercano al diario El País y bastante superior al resto de los medios. En noviembre de 2018 publicó la mayor cantidad de tweets con 3.574 (Figura H.23), pero la mayor cantidad de publicaciones de seguridad, la realizó en abril de ese mismo año con 313 de 2.875 (Figura H.24), representando un índice de seguridad récord de 10.9 % (Figura H.25).

Apéndice D

Herramientas para creación de dataset

En este apéndice se describe en más detalle la implementación e instrucciones de uso del scraper creado y la herramienta de etiquetado utilizada.

Scraper

Para llevar a cabo el scraping, la herramienta utilizada fue Selenium[15], más específicamente la biblioteca disponible para Python[16]. Utilizando dicha herramienta, se creó un script que recorre los tweets buscados y los guarda en un archivo CSV. Para ello, primero se tiene que definir las credenciales del usuario de Twitter y los parámetros a buscar. A continuación se describen las variables que deben ser definidas en un archivo de configuración:

Credenciales:

- USER: Nombre de usuario o email usado para iniciar sesión en Twitter
- PASSWORD: Contraseña correspondiente para el usuario ingresado

Configuraciones de búsqueda:

- DESDE: Fecha de inicio de tweets buscados.
- HASTA: Fecha final de tweets buscados.
- CUENTAS: Identificadores de las cuentas (entre comillas y separadas por coma).
- INCLUIR_PALABRAS: Lista de palabras a buscar en el tweet.
- EXCLUIR_PALABRAS: Lista de palabras a ignorar.
- HASHTAGS: Lista de hashtags que debe contener.

Ejemplo de configuración:

- DESDE = “2021-05-01”

[15]https://www.selenium.dev/
[16]https://selenium-python.readthedocs.io/

- HASTA = “2021-12-22”
- CUENTAS = [“observadoruy”, “elpaisuy”, “diariolaruy”, “ladiaria”, “semanariobrecha”, “busquedaonline”]
- INCLUIR_PALABRAS = [“policia”, “seguridad”, “robo”]
- EXCLUIR_PALABRAS = [“futbol”, “partido”]
- HASHTAGS = [“uruguay”]

Estas búsquedas tan específicas son posibles gracias a la funcionalidad de búsqueda avanzada disponible en el sitio web de Twitter. Utilizando los datos ingresados en el archivo de configuración se crea una URL con todos los parámetros necesarios que representa la búsqueda avanzada y, luego de haber iniciado sesión, se dirige a la página con todos los resultados esperados. Con los datos de ejemplo anteriores se puede crear la siguiente URL:

https://twitter.com/search?q=(policia%20OR%20seguridad%20OR%20robo)%20-futbol%20-partido%20(from:observadoruy%20OR%20from:elpaisuy%20OR%20from:diariolaruy%20OR%20from:ladiaria%20OR%20from:semanariobrecha%20OR%20from:busquedaonline)%20since:2021-05-01%20until:2021-12-22&src=typed_query&f=live

Que se traduce como la siguiente búsqueda avanzada: `(policia OR seguridad OR robo) -futbol -partido (from:observadoruy OR from:elpaisuy OR from:diariolaruy OR from:ladiaria OR from:semanariobrecha OR from:busquedaonline) since:2021-05-01 until:2021-12-22` (Figura 4.1)

Figura D.1: Ejemplo de búsqueda avanzada en Twitter

Utilizando Selenium se obtiene y guarda la información que se muestra en pantalla, se simula scroll de usuario, y cuando se llega al final de la página, se vuelve a hacer un nuevo pedido. Este proceso se repite hasta llegar a la fecha elegida en la configuración o hasta que se cierre el programa manualmente.

Cada tweet se guarda de forma estructurada en archivos CSV obteniendo los siguientes campos:

- TweetID: Identificador del tweet
- User: Nombre del usuario (medio de comunicación)
- Handle: Nombre identificador de la cuenta
- PostDate: Fecha de publicación
- Tweet: Contenido del tweet
- Attachment: Adjuntos
- ReplyCount: Cantidad de respuestas
- RetweetCount: Cantidad de retweets
- LikeCount: Cantidad de "likes"
- TweetLink: Link al tweet original

Etiquetador

Se desarrolló una web[17] donde los integrantes del equipo, tutores y personal del Departamento de Sociología pudieran ingresar y clasificar de manera manual los diferentes tweets.

La web de etiquetado es una aplicación creada usando el framework Django[18] y una base PostgreSQL[19] hosteada en Heroku[20].

[17]https://tweet-tagger.herokuapp.com/
[18]https://www.djangoproject.com/
[19]https://www.postgresql.org/
[20]https://www.heroku.com/

Glosario

CBOW Arquitectura de Word2Vec que utiliza una red neuronal con una única capa oculta y el objetivo es entrenar la red con el corpus de tal forma que, para un contexto (conjunto de palabras), se obtenga la palabra con mayor probabilidad de pertenecer a ese contexto. Para esto el entrenamiento es el siguiente: se define una ventana de tamaño n, (que sería la cantidad de palabras vecinas a tener en cuenta) y se recorren todos los datos, construyendo pares entre la palabra central de la ventana y un vector con las $\frac{n}{2}$ palabras a la izquierda y a la derecha de la palabra central. Una vez que se tienen todos los pares, dado un conjunto de palabras de entrada al modelo, se puede obtener un vector con tamaño igual a la cantidad de palabras existentes en el vocabulario que contiene la probabilidad de que cada palabra pertenezca a ese contexto.

Elasticsearch Elasticsearch[21] es un motor de análisis distribuido, gratuito y abierto para todo tipo de datos. Los mismos son recibidos sin procesar por Elastic desde varias fuentes para previamente ser indexados. Los datos son parseados y normalizados. Elastic también utiliza la estructura de índice invertido haciendo una lista de cada palabra única que aparece en algún momento e identifica a todos los documentos que tienen esa palabra. De esta forma permite búsquedas rápidas de textos completos.

Lematización Al momento de analizar un texto, se encuentran varias palabras diferentes que se derivan de una misma. Ejemplos son:

[“estás”, “están”, “estarían”] [“gatos”, “gata”, “gatito”]

En el primer caso, son todas palabras que derivan del verbo *“estar”*, mientras que en el segundo, son palabras derivadas de *“gato”*. La lematización consiste en llevar todas las palabras de un texto a su lema, o sea, relacionar cada palabra con su forma más básica o reducida, esto permite simplificar y dar un grado de generalización al texto para su procesamiento.

[21] https://www.elastic.co/what-is/elasticsearch

Limpieza de datos El preprocesamiento de datos o limpieza de datos, consiste en preparar el conjunto de datos, ya sea normalizando, reorganizando o limpiándolo para que cumpla con las condiciones necesarias para ser bien procesado, por ejemplo, por un modelo de aprendizaje automático. La limpieza de datos es una parte importante del preprocesamiento y consiste en quitar del conjunto de datos todo aquello que no aporta información relevante o genera mucho ruido para el modelo a entrenar. Qué palabras son las que se van a quitar depende mucho de la temática y de lo que se quiere investigar ya que palabras o frases que pueden ser muy importantes para un contexto, puede ser que no aporten nada en otro. Lo más común en una limpieza de datos es quitar las *stopwords* (aquellas palabras que no tienen contenido léxico como artículos, pronombres, preposiciones, etc). Estas palabras suelen repetirse mucho en un texto generando mucho ruido y por eso lo más común es quitarlas. También es normal quitar los signos de puntuación al momento de limpiar el conjunto de datos. Particularmente en el caso de que el conjunto de datos provenga de redes sociales (como el presente), el proceso de limpieza también suele retirar o remplazar con símbolos especiales emojis, links o hashtags.

MongoDB MongoDB[22] es una Base de datos NoSQL distribuida que almacena los datos como documentos y a su vez los agrupa en colecciones. Los documentos que utiliza MongoDB para organizar y almacenar los datos son similares a documentos JSON: un conjunto de pares clave-valor, donde la clave es un identificador único por el cual se van a poder relacionar datos. MongoDB permite indexar cualquier campo, lo que facilita las búsquedas y mejora performances. A su vez, en una colección permite definir un índice de texto, el cual puede incluir uno o más campos con formato texto. Esto permite realizar búsquedas rápidas de cadenas de texto en dichos campos utilizando índices invertidos.

PCA Principal Component Analysis (PCA) es un método que permite disminuir la complejidad de un espacio muestral de muchas dimensiones y a su vez conservar su información. Dado un espacio de dimensión p y vectores de la forma $(x_1, x_2, ..., x_p)$, PCA permite encontrar un número z con $z < p$, que permite explicar aproximadamente lo mismo que las p variables originales. Básicamente existen dos formas de aplicar PCA

1. Basado en la matriz de correlación
2. Basado en la matriz de covarianzas

.

[22] https://www.mongodb.com/

Scraping Scraping es el proceso de extraer información de la web generalmente de manera automatizada. Se utiliza esta técnica para poder obtener información desde cualquier página web y poder procesarla o analizarla. Usualmente se utiliza un código que simula la navegación en la web de un humano, y de esa forma va recopilando la información de forma estructurada.

Skip-gram Arquitecturas de Word2Vec que utiliza una red neuronal con una única capa oculta. El objetivo es entrenar la red con el corpus, de forma que para una palabra, se obtenga la probabilidad que tienen cada una de las demás palabras del vocabulario de aparecer en una frase junto a la palabra de entrada. Para esto el entrenamiento es el siguiente: se define una ventana de tamaño n, que sería la cantidad de palabras vecinas a tener en cuenta y se recorren todos los datos, construyendo pares entre la palabra central de la ventana y las palabras vecinas dentro de la ventana. Una vez que se tienen todos los pares, se puede obtener un vector con tamaño igual a la cantidad de palabras existentes en el vocabulario que contiene la probabilidad de que cada palabra sea vecina de la palabra de entrada.

Stemming Es una versión más simple de la lematización. Es un proceso para reducir las derivaciones de palabras a su raíz (o *stem*). Generalmente se eliminan los prefijos y sufijos de las palabras para obtener una raíz común. Por ejemplo:

[“allanamiento”, “amenazar”, “investigado”, “investigan”]

derivan en

[“allan”, “amenaz”, “investig”, “investig”]

donde se puede observar que tanto “investigado” como “investigan” tienen a “investig” como raíz. De esta forma, cuando cualquiera de las dos versiones de la palabra aparezcan, serán tratadas como la misma.

t-SNE t-Distributed Stochastic Neighbor Embedding[23] (t-SNE) es una técnica para la reducción de la dimensionalidad muy utilizada en visualizaciones de conjuntos con datos de muchas dimensiones. t-SNE se ejecuta en dos pasos:

1. Se construye una distribución de probabilidades entre parejas del espacio original, de tal forma que la probabilidad será alta si las muestras son similares, y una baja probabilidad de ser seleccionadas si las muestras son muy diferentes. La similaridad entre las muestras x_i y x_j es la probabilidad condicional de que x_i escogiese a x_j como su vecino si los vecinos fuesen escogidos proporcionalmente a su densidad de probabilidad bajo una curva gaussiana centrada en x_i.

[23]https://lvdmaaten.github.io/tsne/

2. t-SNE intenta reproducir en el espacio dimensional nuevo, la distribución existente en el espacio original, para esto, de forma aleatoria va pasando cada punto del espacio de dimensión alta, al espacio de dimensión baja, formando una distribución de probabilidad similar a la original y buscando minimizar la divergencia de Kullback-Leibler (la divergencia de Kullback-Leibler mide la similitud o diferencia entre dos funciones de distribución de probabilidad). [17]

.

Tokenización La tokenización es una tarea fundamental para el trabajo en PLN. Consiste en fragmentar el texto en unidades más pequeñas que se llaman tokens. Estos tokens pueden ser palabras, subpalabras o caracteres. Por ejemplo, en la lengua española, algo muy común es utilizar los espacios como separadores de tokens, por lo que de la frase *"La policía atrapó al ladrón."* se obtienen los siguientes tokens:

[“La”, “policía”, “atrapó”, “al”, “ladrón.”]

.