Descubrimiento de tópicos y análisis de sentimientos usando hastags relacionados al Covid-19

1st Alvaro Machuca Breña
Universidad ESAN
Código: 15101364
15101364@ue.edu.pe
Lima, Perú

2th Jhonathan Camasca Huamán
Universidad ESAN
Código: 15100036
15100036@ue.edu.pe
Lima, Perú

***Abstract*—En el presente documento se detallarán 3 papers relacionados al curso de analítica de la web con el objetivo de poder observar diferentes trabajos relacionados como análisis de sentimiento, LDA y el procesamiento de texto en base a los tweets que nos permitirán conocer diversas técnicas que se han desarrollado para mejorar la toma de decisiones y encontrar patrones de conducta en data que se encuentra mayormente en las redes sociales. Otro objetivo, será seleccionar la metodología más adecuada para utilizar en el trabajo final de curso. Se justificó cada una de estas propuestas con artículos científicos que han realizado temas relacionados. Asimismo, para cada una de las propuestas se aplicarán técnicas y metodologías de inteligencia artificial, así como de Deep Learning.**

***Index Terms*—Covid-19, LDA, Sentiment Analysis, Text Mining, Coronavirus**

I. Introduction

En estos tiempos, alrededor del mundo se están viviendo momentos difíciles por el surgimiento del nuevo coronavirus conocido como Covid-19. Este virus letal surgido en china en diciembre del 2019 tiene a las personas viviendo con un profundo temor hacia esta nueva enfermedad dado que cada día se incrementan el número de contagiados y fallecidos por esta enfermedad. Como medida de prevensión, muchos gobiernos han tomado medidas drásticas con el objetivo de prevenir la propagación de esta enfermedad. Una de las medidas optadas por los países ha sido una cuarententa total con el objetivo de poder disminuir el transito de personas y así evitar la propagación del virus.Dado a lo último mensionado, las personas al estar mucho más tiempo en las casas, han usado las redes sociales para expresar sus emociones con respecto a todas las declaraciones o nuevas medidas optadas por el propio gobierno y con respecto al coronavirus en general.

Dada la cantidad gran cantindad de información que se genera por las personas en estos tiempos de coronavirus, es dificil dar seguimiento a cada uno de los comentarios que se van publicando en las distintas redes sociales como Twitter o Facebook con el objetivo de determinar si las personas estan a favor o en contra con respecto a algún tipo de situación relacionada al coronavirus o de qué tópico suelen estar hablando mucho más en sus comentarios. No obstante, con el surgimiento de nuevas técnicas para el análisis de información y para la detección de tópicos, es posible la realización de un modelo que permita dado un comentario de entrada, poder detectar a qué tópico pertenece y qué tipo de sentimiento tiene (positivo o negativo).

En el presente proyecto se utiliza la técnica de LDA para la detección de tópicos de tweets recolectados de twitter usando hashtags relacionados al coronavirus en el país de Perú. Además, se utiliza una técnica de análisis de sentimientos para la clasificación de los tweets en positivos o negativos. Para la elección de la técnica de análisis de sentimiento a usar, se hace una comparación de tres modelos: 1 modelo de machine learning, 1 modelo de deep learning y un modelo que combine los dos anteriores. Se evaluarán los 3 modelos utilizando el Acuracy, Precision, Recall y F1-score con el objetivo de determinar qué modelo es el más adecuado para realizar el análisis de sentimiento.

II. Estado del arte

En esta sección, se procederán a explicar tres artículos relacionados con las técnicas que se aplicarán en este documento, la metodología que siguieron los autores y los resultados obtenidos que servirán de base para el desarrollo del presente proyecto de investigación.

A. Latent Dirichlet allocation (LDA) for topic modeling of the CFPB consumer complaints

Este primer trabajo consiste en la utilización de la técnica Latent Dirichlet Allocation conocida como LDA con el objetivo de detectar los tópicos presentes en las quejas que recibe la Ofinica para la protección Finaciera del Consumidor (CFPB por su siglas en inglés) y así agilizar el proceso de análisis de quejas dado que estas últimas han ido en aumento en los últimos años y el hacer el análisis de queja por queja por parte del personal de la CFPB se vuelve una tarea muy laboriosa y poco efectiva [1]. Es importante mensionar que el objetivo principal de la CFPB es recibir las quejas de todas las personas con respecto a los servicios financieros que ellos adquieren y asegurarse que los bancos, prestamistas y otras compañias

brinden sus servicios de manera correcta y transparente [1]. La base de datos que se utilizó fueron al inicio de 615,273 quejas. Además de las quejas, se tenian los valores de otros campos importantes como la fecha de presentación de la queja, el código del consumidor, en nombre del banco infractor, entre otros. Dentro de la base de datos, se encontraron algunas filas que poseían el campo de la queja vacio y otras en donde se repetian la misma queja por lo que procedieron a eliminar estos tipos particulares de registros [1]. Luego de tener la data limpia, se obtuvo una nueva fuente de datos de 86,803 registros con los cuales se va a trabajar. La metodología que se utilizó consta de 5 pasos para el tratamiento de los datos antes utilizar LDA para detectar los tópicos presentes en el texto, la cual se encuentra presente en la figura 1.

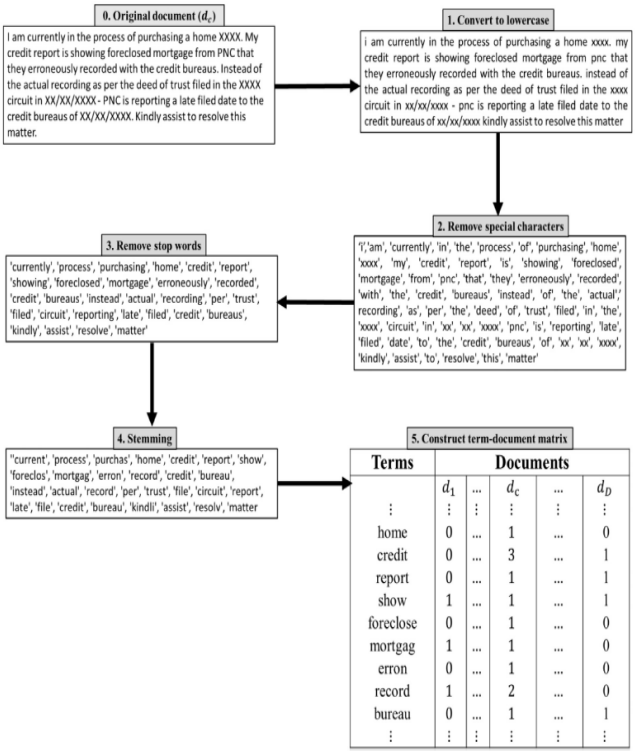

Fig. 1. Metodología empleada para el tratamiento de los datos

El paso cero consiste en la recopilación de la información. Esta data se obtuvo del portal web de la CFPB cuyos datos de las quejas de los clientes son públicos. Además, la eliminación de quejas duplicadas o en blanco. El paso 1 consiste en la conversión de todo el texto a minúsculas. El paso 2 consiste en remover todos los caractéres especiales como signos de exclamación, interrogación, numerales, entre otros y realizar tokenización a los datos. El paso 3 consiste en remover los Stopwords (palabras sin significado propio) como artículos, pronombres, etc. presentes en el documento. El cuarto paso consiste en realizar steamming a las palabras. Esta técnica consiste en obtener una palabra y quedarse solamente con la palabra raíz de la misma. Finalmente, el quinto y último paso consiste en la conversión de las palabras obtenidas a una matriz de términos en donde por cada palabra se ubica su frecuencia en cada documento.

Para la realización del algoritmo LDA cuya representación se encuentra en la figura 2, es necesario definir 3 valores. De estos 3 valores, 2 de ellos son los hiperparámetros alpha y etha y el último viene a ser K que es la cantidad de tópicos que uno desea extraer del texto [1]. Para el caso de los hiperparámetros, se utilizaron los valores de 0.1 tanto en alpha como en etha y un valor de K = 40. Por otro lado, los valores de N y D son la cantidad de palabras únicas en el documento y la cantidad total de documentos respectivamente. Además, W(d,n) representa cualquier palabra n dentro del documento d y Z(d,n) representa la asignación del tema o tópico por cada palabra en el texto. Finalmente, βk y θd representan las distribuciones sobre las palabras de cada tópico y la distribución por tema presente en el documento.

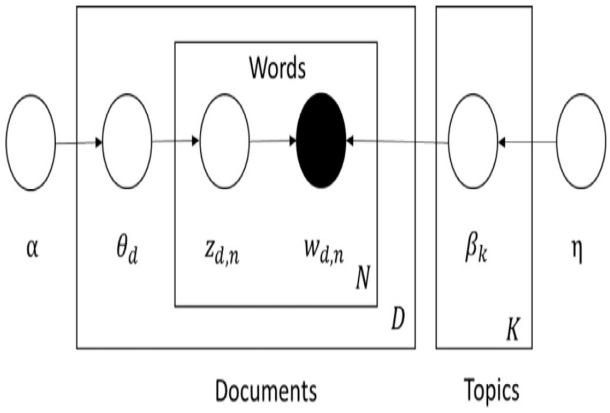

Fig. 2. Latent Dirichlet allocation (LDA) para detección de tópicos

Al implementar el algoritmo de LDA, cada uno de los 40 tópicos diferente era una colección de 10 palabras. A cada tópico se decidió darle una etiqueta en particular de manera manual para que sea más entendible por el usuario. Los resultados que se obtuvieron fueron presentados utilizando la herramienta de BI Tableau en donde cada usuario podía ingresar a la aplicación y observar por ejemplo por cada banco cual era el o los tópicos de quejas que más se repetian y así la persona podría decidir si desearia comprar un producto o servicio de un banco determinado. Por otro lado, otro reporte fue el agrupamiento de bancos o compañias en función al tipo de tópico, de tal forma que la CFPB podría darle prioridad a regular los servicios ofrecidos o productos ofrecidos por determinados bancos en función al tópico al cual pertenecen.

B. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks

En este trabajo se busco mejorar el performance de la clasificación de texto corto, siendo esta una tarea importante en muchas áreas del procesamiento de lenguaje natural, siendo en este caso el de análisis de sentimientos.En este trabajo de investigación se evalúo muchos enfoques como el uso de maquinas de soporte vectorial (SVM), la combinación de SVM y Naive Bayes y construcción de árboles de

dependencia. También examinaron los resultados de varios estudios recientes donde se utilizo redes neuronales artificiales (ANN), redes neuronales convolucionales (CNN) y redes neuronales recurrentes (RNN) que mostraron resultados prometedores.Posterior a ello, se evaluó que las ANN realiza la clasificación de forma aislada, es decir no se tiene en cuenta los textos cortos anteriores. Para ello en este trabajo de investigación se presento un modelo basado en redes neuronales recurrentes (RNN) y CNN para la clasificación secuencial de texto corto, y lo evaluaron en la clasificación de diálogos, siendo un dialogo una combinación de criterios pragmáticos, semánticos y sintácticos.Al final se logro resultados de vanguardia en los conjuntos de datos usados.Las bases de datos que se utilizaron fueron 3: DSTC 4: Dialog State Tracking Challenge. Esta base de datos pertenece a un desafio que busca crear un ”rastreador” que pueda predecir el estado del diálogo para nuevos diálogos.La base de datos contiene data de dialogos etiquetada con la información del estado del dialogo, MRDA: ICSI, Registro de Reuniones, Diálogo, Acta, Ley.Esta base de datos contiene diálogos anotado a mano de las reuniones y las actas del ICSI, SwDA: Switchboard Dialog Act Corpus: Esta base de datos esta conformada por diálogos por teléfono etiquetadas con los resúmenes de información sintáctica, semántica y pragmática sobre el giro asociado.La primera base de datos (DSTC 4) contiene 89 clases y un vocabulario del tamaño de 6k, la segunda base de datos (MRDA) contiene 5 clases y un vocabulario del tamaño de 12k y la tercera base de datos (SwDA) contiene 43 clases y un vocabulario del tamaño de 20k.La metodología presente se divide en dos partes, la primera parte consta de la generación de un vector por cada texto mediante el uso de la CNN o la RNN y la segunda parte realiza la clasificación basandose en la vector del texto actual y el texto anterior.

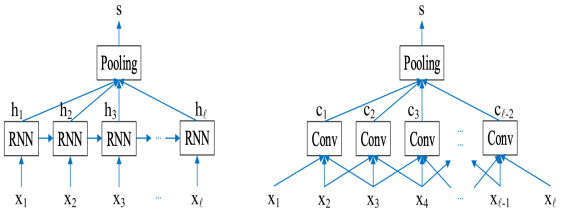

Fig. 3. Arquitecturas CNN y RNN

En la Figura 3 podemos ver las arquitecturas de redes neuronales convlucionales (CNN) y recurrentes (RNN) que se usaron para convertir un texto que esta conformado por X1 a Xl en vectores. Cabe destacar que en las CNN el valor de h1 a hl significa la dimensión del kernel (filtro) que realiza la convolución.

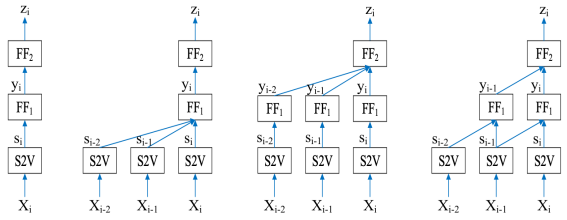

Fig. 4. Clasificador de los textos cortos

En la Figura 4, podemos ver 4 ejemplos de redes neuronales que constan de dos capas (FF1 Y FF2) para predecir la distribucion de la probabilidad con respecto a la clase Zi para el elemento i del texto corto Xi, siendo SV2 la representacion del vector caracteristico del texto generado por las arquitecturas de deep learning anteriormente mencionadas.Las arquitecturas de deep learning(RNN y CNN) trabajan con un texto corto de tamano ’l’ que es una secuencia de ’m’ vectores de palabras. Se uso un tipo de RNN llamado Long Short Term Memory (LSTM) y una CNN que se baso en una arquitectura compuesta por convoluciones con kernels de tamano h, una capa de maxpooling y funciones de activacion ReLU. En la etapa de clasificacion, los vectores generados por las arquitecturas de deep learning son la entrada de las redes neuronales de dos capas en forma de secuencia para determinar la distribucionn de probabilidad sobre las k clases. Para el entrenamiendo de las arquitecturas previamente mencionadas, se realizo mediante la gradiente descendiente estocastica con la finalidad de actualizar los pesos y los vectores de palabras. Ademas se implemento una capa de dropout despues de la capa de pooling y un early stopping que tiene como maximo el tope de 10 epocas en el set de validacion.

Hyperparameter	**Choice**	**Experiment Range**
LSTM output dim. (n)	100	50 – 1000
LSTM pooling	max	max, mean, last
LSTM direction	unidir.	unidir., bidir.
CNN num. of filters (n)	500	50 – 1000
CNN filter height (h)	3	1 – 10
Dropout rate	0.5	0 – 1
Word vector dim. (m)	200, 300	25 – 300

Fig. 5. Parametros usados en la CNN y RNN

En la Figura 5, podemos ver los hiperparametros que se usaron en las arquitecturas de la LSTM(RNN) y la CNN. Como se puede visualizar se uso un max pooling en la LSTM, siendo esta de caracter unidireccional y contando con salidas de dimension 100, mientras que la CNN contaba con 500 kernels (filtros) de dimension 3x3, una capa de dropout del 0.5 y un vector de salida de dimensiones entre 200 y 300.Con respecto a los resultados obtenidos

Model	DSTC 4	MRDA	SwDA
CNN	65.5	**84.6**	**73.1**
LSTM	**66.2**	84.3	69.6
Majority class	25.8	59.1	33.7
SVM	57.0	–	–
Graphical model	–	81.3	–
Naive Bayes	–	82.0	–
HMM	–	–	71.0
Memory-based Learning	–	–	72.3
Interlabeler agreement	–	–	84.0

Fig. 6. Resultados de los modelos propuestos vs modelos convencionales

En la Figura 6, se puede visualizar el accuracy de los modelos propuestos en comparación de los modelos de la literatura o antecedentes del trabajo de investigación. Para los modelos CNN y LSTM, los resultados presentados son los resultados con mayor accuracy en el conjunto de validación. Se puede visualizar que los modelos de deep learning muestran mejor performance que los modelos de los antecedentes. No obstante, esto depende del tipo del pre-procesamiento del texto,el porcentaje del split del train, test y val y la asignacion de aleatoriedad de cada clasificador.Tambien se puede concluir que los resultados son de vanguardia para los tres datasets.

C. Detecting opinion spams and fake news using text classification

Actualmente, las noticias falsas y las criticas falsas conocidas como spam afectan a las personas tanto como a las organizaciones.Estos dos fenomenos se caracterizan en informacion textual falsa que es difundida y que afecta a consumidores, tiendas y personas al no poder determinar si estas son reales o falsas.Estos fenomenos han incrementado de manera notoria en los ultimos anos y se han vuelto tema de investigacion en crecimiento debido a la abundancia de contenido generado por los usuarios. Ahora es fácil para cualquiera escribir críticas falsas o escribir noticias falsas en la web. Uno de los mayores desafios es distinguir de manera eficiente de una revisión real y una falsa como una noticia real de la falsa. En este trabajo de investigación se propuso un modelo de detección que combina el análisis de texto usando características de n-gramas y términos métricas de frecuencia y machine learning. Para ello se comparo 2 técnicas de extracción de características diferentes y 6 técnicas de clasificación de aprendizaje automático y se realizo la evaluación experimental utilizando los conjuntos de datos públicos existentes y un conjunto de datos de noticias falsas. Las bases de datos que se usaron para el trabajo de investigacion fueron 3 que tenian contenido real y falso. El primero de Ott et al. que consiste de 400 opiniones reales y 400 falsas sacadas de TripAdvisor y Amazon Mechanical Turk (AMT), la segunda base de datos de Horne BD contiene 4110 datos reales, 110 datos falsos y 305 datos satiricos. El ultimo dataset fue recolectado para este trabajo de investigacion y continen 12 600 noticias reales y 12600 falsas. La metodologia del presente trabajo de investigacion empezo con el pre-procesamiento del conjunto de datos eliminando caracteres y palabras innecesarios de los datos.Posteriormente, se extrajeron las características de n-gramas y se formo una matriz de características que representa los documentos involucrados. Finalmente,se entreno al clasificador con los siguientes algoritmos de machine learning :descenso de gradiente estocástico (SGD), SVM, máquinas de vectores de soporte lineal (LSVM), k vecinos mas cercanos (KNN), LR y árboles de decisión (DT).La metodologia se puede visualizar de forma mas detallada en la Figura 7.

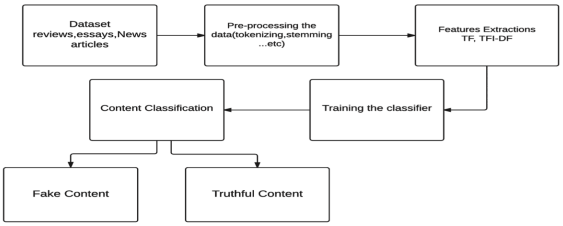

Fig. 7. Proceso de clasificacion de deteccion de contenido reales

El modelo de n-grams fue basado en palabras para representar el contexto del documento y generar características para clasificar el documento y diferenciar entre contenido falso y honesto. Basicamente, se trato de generar varios conjuntos de perfiles de frecuencia de n-gramas a partir de los datos de entrenamiento para representar contenido falso y verdadero y se examino el efecto de la longitud de n-gramas en la precisión de diferentes algoritmos de clasificación. El pre-procesamiento usado fue: eliminación de palabras de detención, tokenización, conversion a minúsculas, segmentación de oraciones y eliminación de signos de puntuación para reducir el tamaño de los datos reales al eliminar la información irrelevante que existe en los datos.Las palabras de detencion o stop words que se eliminaron, por ejemplo, fueron: a, sobre, un, son, como, en, es, por, para, de, de forma, dentro, sobre, o, que, el, ese, este, estos, fue, cuándo, dónde, quién, y así sucesivamente.Estas palabras fueron eliminadas de cada documento, y los documentos procesados se almacenaron y pasaron al siguiente paso. Posteriormente, se realizo el proceso de steamming con la finalidad de reducir el tipo de palabras y convertir todas las palabras a su forma base. Para ello se uso PorterStemmer debido a su precision a la hora de realizar el steamming. Los métodos de extracción de características que se usaron en el trabajo de investigacion fueron:frecuencia de término (TF) y frecuencia de documento de frecuencia invertida de término (TF-IDF). Posteriormente a la extraccion de caracteristicas se separo el corpus de documentos o conjunto de datos en conjuntos de entrenamiento y prueba (0.20 prueba y 0.8 entrenamiento con validacion cruzada 5). Para el entrenamiento del clasificador se uso 6 clasificadores diferentes para predecir la clase de los documentos, incluidos SGD, SVM, LSVM, LR, KNN y DT.

En el presente trabajo se realizo 2 experimentos: uno para evaluar la capacidad del modelo propuesto para detectar revisiones falsas y el otro para evaluar su capacidad para detectar noticias falsas. El procedimiento fue el mismo para los dos experimentos. Como anteriormente se menciono se busco medir el impacto del tamaño de n-grams en el rendimiento. Por ello se comenzo,con unigram (n = 1), luego bigram (n

= 2), y luego se aumento constantemente en 1 hasta llegar a n = 4. Cabe destacar que se uso los dos tipos de tecnicas de extracción de caracteristicas (TF y TF-IDF) para cada n-gram. Ademas se uso una validacion cruzada de 5 dividiendose la data en 0.2 test y 0.8 train. Finalmente se uso los 6 modelos de clasificacion.

	TF-IDF				TF			
n-gram size	1000	5000	10 000	50 000	1000	5000	10 000	50 000
Unigram	83.0	82.0	83.0	82.0	83.0	82.0	79.0	82.0
Bigram	80.0	83.0	82.0	80.0	80.0	83.0	82.0	80.0
Trigram	73.0	78.0	79.0	75.0	75.0	78.0	77.0	75.0
Fourgram	76.0	69.0	68.0	47.0	70.0	69.0	68.0	42.0

Fig. 8. Accuracy del SVM para el dataset1 por tamano de características y tamaño de n-gram para la detección de reviews falsos

Como se visualiza en la Figura 8, los resultados del primer experimento se visualizo que mientras mas se incrementa el tamaño de las características y el tamaño del n-gram el accuracy disminuye. Esto se visualiza en los dos tipos de técnicas de extracción de características.

	TF-IDF				TF			
n-gram size	1000	5000	10 000	50 000	1000	5000	10 000	50 000
Unigram	84.0	86.0	84.0	84.0	85.0	72.0	69.0	69.4
Bigram	78.0	73.0	67.0	54.0	68.0	51.0	47.0	47.0
Trigram	71.0	59.0	53.0	48.0	53.0	47.0	53.0	47.0
Fourgram	55.0	37.0	37.0	45.0	47.0	48.0	40.0	47.0

Fig. 9. Accuracy del SVM para el dataset1 por tamano de caracteristicas y tamano de n-gram para la deteccion de noticias falsos

Como se visualiza en la Figura 9, los resultados del segundo experimento se visualizo que mientras mas se incrementa el tamano de las caracteristicas y el tamano del n-gram el accuracy disminuye. Esto se visualiza en los dos tipos de tecnicas de extraccion de caracteristicas.

Dataset	Classifier	Features	Performance metrics	Score	Reference
Reviews Amazon website	Logistic regression	Review and reviewer features	AUC	78%	8
Dataset 1 (Ott et al. reviews dataset)	SVM	LIWC + Bigrams	Accuracy	89%	16
Dataset 1 (Ott et al. reviews dataset)	SVM	Stylometric features	F-measure	84%	16
Dataset 1 (Ott et al. reviews dataset)	LSVM	Bigram	Accuracy	90%	Our Results
Dataset 2 (Our news dataset)	LSVM	Unigram	Accuracy	92%	Our Results
Buzzfeed news and random new articles (Horne and Adali's news dataset)	SVM	Text-based features	Accuracy	71%	17
Buzzfeed news and random new articles (Horne and Adali's news dataset)	LSVM	Unigram	Accuracy	87%	Our results

Fig. 10. Comparación de trabajos anteriores y nuestro trabajo para el spam de opinión y la detección de noticias falsas

Como se visualiza en la Figura 10, los resultados de la metodologia propuesta superan a las metodologias convencionales teniendo un 0.92 de accuracy y superando por mucho a las demas propuestas

III. Metodología

En esta sección del presente documento, se detallará la metodología a utilizar para poder detectar el tópico al cual pertenece cada tweet y el sentimiento que posee.

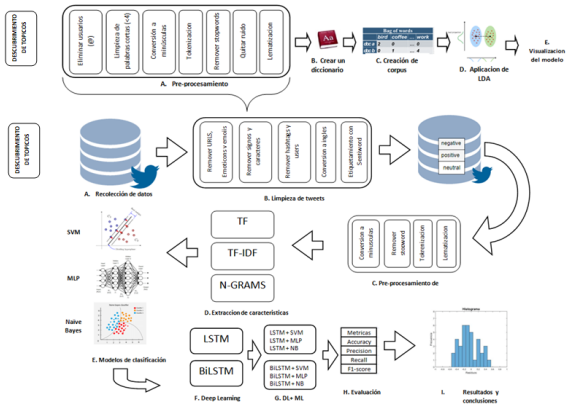

Fig. 11. Metodología del presente trabajo de investigación

A. Recolección de datos

Para la extracción de la data se necesitó contar con una cuenta de desarrollador para usar la API de twitter. Una vez creada la cuenta, se procedió a crear una nueva Twitter Application. Al momento de la creación de nuestra nueva app, se nos otorga principalmente valores para 4 principales campos que son el consumer key, cosumer secret key, access token y el access token secret. Estos 4 campos con sus respectivos valores, servirán de autentificación para poder conectarnos a twitter y poder extraer los tweets más adelante. Posteriormente, se procedió a instalar R junto con R studio versión 1.2.1335 para poder realizar la extracción. Las librerías que se utilizaron fueron ”twitterR” para la extracción de los datos , ”rtweet” para activar la extracción por zonas específicas y ”readr” para la exportación e importacion de archivos. Se procedió a importar las 3 librerías y se proceció a crear 4 variables llamadas api_key,api_secret_key,access_token y access_token_secret en donde cada una almacenaba el valor proporcionado por la aplicación de twitter cuando se creó. Luego, para la autentificación, se utilizó la función setup_twitter_oauth() proporcinada por la librería ”twitterR” y dentro de esta función se agregaron las 4 variables mensionadas. Para la extracción de los tweets, se creó una variable llamada ”tweets” que almacenaria todos los tweets que se iban extrayendo. La función searchTwitter, permitió hacer la conexión por twitter para realizar la extracción. Además, hubieron otros argumentos que intervinieron en la función de extracción. El primer argumento para la extracción es la propia palabra o hashtag que se desea extraer, como segundo argumento esta la función lookup_coords en donde dentro de esta función proporcionada por la librería ”rtweet” se introducía el pais del cual se deseaba hacer la extracción que en este caso es solamente de perú. El siguiente argumento es el n que viene a ser la cantidad de tweets que se desea extraer. Seguidamente, se encuentra

el argumento ”lang” que es el idioma con el cual se desea que tengan los tweets. En este caso esta variable lleva el valor de ”es” porque los tweets serán extraidos en el idioma español. Finalmente están los parámetros de ”since” y ”until” en donde se introduce desde qué fecha a qué fecha se desea hacer la extracción. Es preciso mensionar que la API de twitter permite hacer la extracción solamente de los 7 últimos días, por lo que se hizo la extracción de manera díaria de 20000 en 20000 tweets utilizando hashtags relacionados al coronavirus como ”covid-19”, ”vacuna”, ”bono”, ”QuedateEnCasa” entre otros. Una vez extraidos los tweets, se procedió a convertir los resultados en dataframe con la función twListToDF y almacenado en una nueva variable. Finalmente, con la función write_csv proporcionada por la librería ”readr”, se exportó el dataframe a un archivo csv cuyo nombre del archivo era el nombre del hashtag con el cual se extrajo para un mejor manero de los archivos. En total, se recopilaron 137874 tweets por un periodo de 6 semanas usando diversos hashtags relacionado a coronavirus en el Perú.

Para la etiquetación de los datos, se removieron URL, emoticones y emojis; signos de puntuación y stop words. Por otro lado, se convirtieron los tweets al idioma inglés usando la aplicación de Google Translate y se ha hecho una etiquetación usando sentiword el cual es un clasificador que en este caso ha sido entrenado usando reviews de películas. Finalmente, se obtuvo una data limpia en donde cada tweet esta clasificado como ”positivo”, ”negativo” o ”neutro”.

REFERENCES

[1] K. Bastani, H. Namavari and J. Shaffer, Latent Dirichlet allocation (LDA) for topic modeling of the CFPB consumer complaints, Expert Systems With Applications, 2019, vol. 127, 2019, pp. 256–271.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].