Universidad Nacional Autónoma de México

Facultad de Ingeniería

Análisis y Procesamiento Inteligente de Textos

Clasificador Sentimental de tweets de COVID-19 basado en Procesamiento Inteligente de Textos mediante Redes Neuronales RRN-GRU

Profesor:

- Sánchez Velázquez, Octavio Augusto

Alumnos:

- Murrieta Villegas, Alfonso. alfonsomvmx@comunidad.unam.mx
- Pérez Valdez, Edgar michaelvaldez@comunidad.unam.mx
- Ceballos Equihua, Conan Nathaniel. ceballos.equihua@comunidad.unam.mx

Table of Contents

Objetivos

- Aplicar conceptos relacionados con Procesamiento del Lenguaje Natural, Inteligencia Artificial, Estadística y Probabilidad en un clasificador sentimental de tweets de 3 etiquetas (neutral, positivo y negativo).

- Con base a los conceptos aprendidos en Análisis y Procesamiento Inteligente de Datos, ajustar nuestro modelo de Inteligencia Artificial para hacerlo lo más preciso posible.

Introducción

Una realidad que ha acompañado al ser humano a lo largo de las últimas 3 décadas es el convivir de una u otra forma con el internet y con las nuevas formas de interactuar con la sociedad. A pesar de que el objetivo explicito de este proyecto no es abordar cómo ha perjudicado sentimentalmente o mínimo afectado la cuarentena originada por el COVID-19 en las personas, lo que si aborda es que con ayuda de Inteligencia Artificial y concretamente con ayuda del Procesamiento de Textos se puede crear herramientas que concreten tareas particulares como el interpretar los mensajes que las personas hablan acerca de lo qué es nuestra realidad este 2020 y 2021.

Por ello lo primero que debemos considerar es que hay dos principales conceptos en este trabajo lo que se denomina *análisis de sentimientos* y lo que podría llamarse *minería de opinión*.

La minería de opinión es una nueva tendencia que se caracteriza explícitamente en lo que previamente se ha denominado como *minería de textos,* la diferencia es que concretamente esta rama de los textos se centra en textos que tienen una característica principal que es la ***subjetividad*** de los mismos, sobre todo, este concepto es algo que debemos considerar pues si bien el presente proyecto y herramientas analíticas tienen un sustento comprobable y matemático, por otro lado, los datos de entrada como de salida pueden ser interpretados con base a la subjetividad o criterio con que cada persona pueda asumirlo.

Marco Teórico

Procesamiento Natural del Lenguaje e Inteligencia Artificial

El objetivo principal del Procesamiento del Lenguaje Natural (NLP) es hacer que las máquinas comprendan los textos no estructurados y extraigan la información relevante de estos. Ante la gran cantidad de información en texto que generamos actualmente, surge la posibilidad de analizarla y aprovecharla. Las técnicas de NLP permiten extraer datos relevantes automáticamente de la información disponible en cualquier sector.

Redes Neuronales

Las redes neuronales, son modelos simples del funcionamiento del sistema nervioso. Las unidades básicas son las neuronas, que generalmente se organizan en capas, como se muestra en la siguiente imagen:

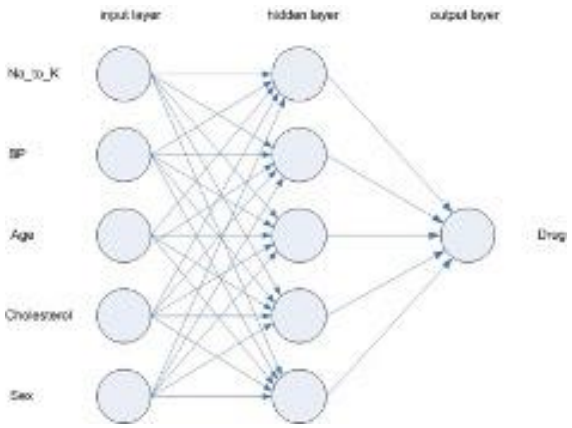


Imagen 1. Estructura de una Red Neuronal

Las unidades de procesamiento se organizan en capas. Hay tres partes normalmente en una red neuronal:

1. Una capa de entrada, con unidades que representan los campos de entrada
2. Una o varias capas ocultas
3. Una capa de salida, con una unidad o unidades que representa el campo o los campos de destino.

La red aprende examinando los registros individuales, generando una predicción para cada registro y realizando ajustes a las ponderaciones cuando realiza una predicción incorrecta. Este proceso se repite muchas veces y la red sigue mejorando sus predicciones hasta haber alcanzado uno o varios criterios de parada.

Si bien al principio todas las ponderaciones son aleatorias y las respuestas que resultan de la red también lo son, a través del entrenamiento la red se va haciendo cada vez más precisa en la replicación de resultados conocidos. Una vez entrenada, la red se puede aplicar a casos futuros en los que se desconoce el resultado.

Stopwords

Las *stopwords* son palabras que en los idioma no agregan mucho significado a una oración, es tal el caso que pueden ignorarse con seguridad sin sacrificar el significado, es el caso de conectores o artículos.

Específicamente, en computación, las stopwords o “palabras vacías” son palabras que se filtran antes o después de que se procesen los datos del lenguaje natural (texto).

TF-IDF

TF-IDF son las siglas de "Frecuencia del término - Frecuencia de documento inversa". Ésta es una técnica para cuantificar una palabra en documentos, generalmente se calcula un peso para cada palabra que representa la importancia de la palabra en el documento y en el corpus.

Se usa para determinar qué palabras en un corpus de documentos podrían ser más favorables para usar en una consulta. TF-IDF calcula los valores de cada palabra en un documento al porcentaje de documentos en los que aparece la palabra. La idea detrás del enfoque TF-IDF es que las palabras que aparecen menos en todos los documentos y más en un documento individual contribuyen más a la clasificación. TF-IDF es el producto de TF e IDF, los cuales se calculan de la siguiente forma:

$$TF = (Frecuencia\ de\ la\ palabra\ en\ el\ documento)/(Total\ de\ palabras\ en\ el\ documento)$$

$$IDF = Log((Número\ total\ de\ documentos)/(Número\ de\ documentos\ que\ contienen\ la\ palabra))$$

Evidentemente, estos términos se calculan para cada palabra, y por cada documento en el caso de TF. Esto genera, entonces, una matriz, donde por un lado se tiene a los documentos, y por el otro a todas las palabras que componen el vocabulario del corpus.

Esta matriz no contiene mucha información directamente útil para el observador humano; sin embargo, es valiosa para un modelo de inteligencia artificial, pues le permite saber cuál es la relevancia de las palabras en el corpus y, en este caso, identificar tendencias en la relación de las palabras con el sentimiento del tweet.

Desarrollo

1. Ambiente de desarrollo y lenguaje de programación

Para el desarrollo del presente proyecto una de las primeras decisiones como equipo realizamos fue establecer un ambiente de programación que nos brindara una facilidad al interactuar con el código de forma conjuntar, es por ello que para el desarrollo en código se escogió a Python como lenguaje de programación específicamente por 2 principales razones la primera es que es un lenguaje amigable y con una gran comunidad detrás de el, sin embargo, como segunda y principal razón escogimos Python sobre todo por la familiaridad que ya teníamos con el lenguaje además de todas las bibliotecas y recursos que este tiene para el desarrollo de Inteligencia Artificial.

Con base a lo anterior es que escogimos como IDE uno de los más conocidos dentro del desarrollo de AI que es Google Colaboratory que destaca principalmente por disponer de GPU's de uso gratuito lo cual más adelante será detallado, además de que da la posibilidad de poder segmentar el código en pequeños módulos los cuales hacen bastante organizado y limpio el código principalmente para presentarlo.

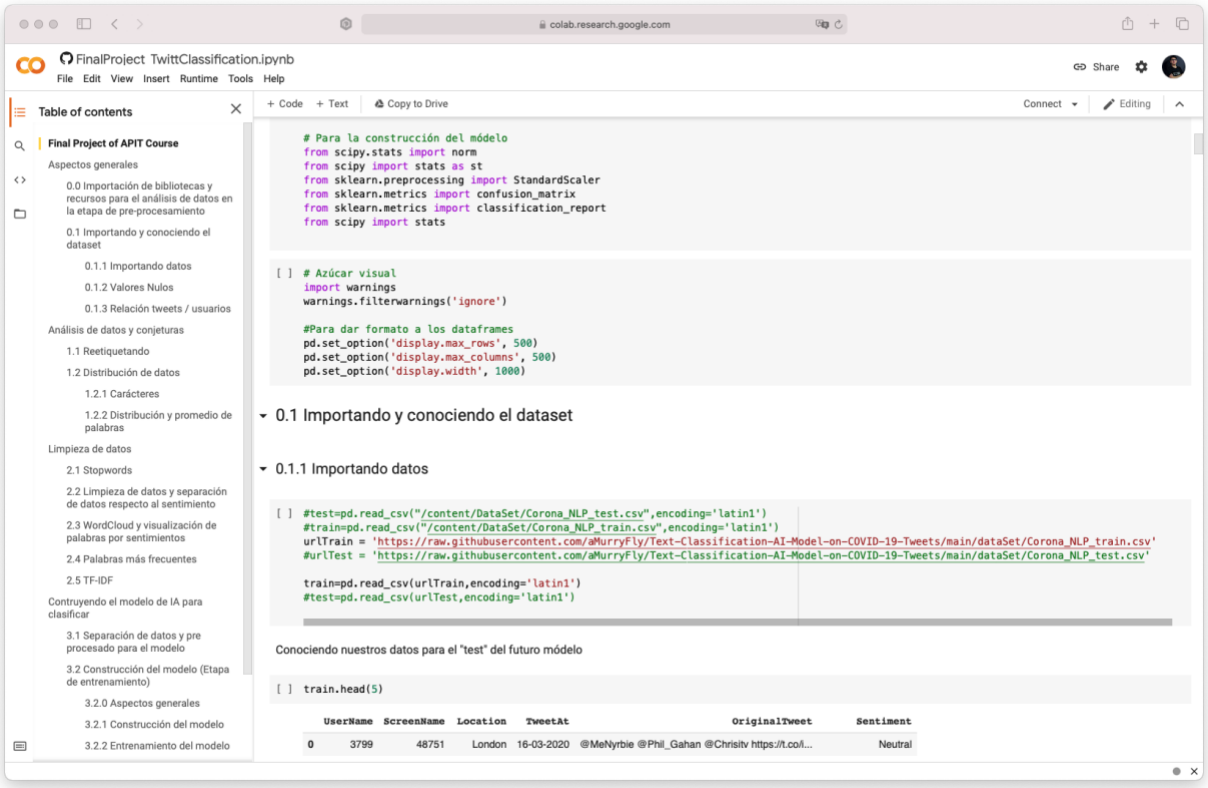


Imagen 1: Captura de pantalla donde se muestra la estructura del código en Google Colaboratory

2. Bibliotecas y recursos

Python es uno de los lenguajes más utilizados en el mundo del desarrollo y probablemente es el más utilizado en Ciencia de Datos, y esto se debe principalmente a la amplia variedad de bibliotecas, API's y un sinfín de recursos que la comunidad ha proporcionado y además sigue dando tanto soporte como actualizaciones.

Para este proyecto de manera general se emplearon las siguientes bibliotecas:

- ***Numpy, Pandas*** y ***Matplotlib*** principalmente para manejo de datos y de manera general para graficación y muestreo de analítica
- ***NLTK*** para el tratamiento de corpus o datos amplios y ricos en texto (cadenas)
- ***Tensorflow y Keras*** para la construcción del modelo de inteligencia artificial
- ***Sklearn y Scipy*** para el muestreo y ejemplificación de resultados de nuestro modelo una vez obtenido

Cabe mencionar que en los siguientes capítulos se abordará a detalle el uso de cada una de estas bibliotecas.

3. Conjunto de datos (Dataset) y Análisis exploratorio de datos

Uno de los sitios más grandes del mundo de Ciencias de Datos es Kaggle, esta es una comunidad subsidiada por Google que está orientada ampliamente a compartir de forma publica 2 principales recursos, datos y código *Open Source*.[1]

Para el presente proyecto utilizamos un conjunto de datos o dataset recopilado de Kaggle, el cual consiste en Tweets con temática de COVID-19 que previamente ya han sido clasificados y etiquetados (En 5 categorías; Neutral, Positivo, Negativo, Extremadamente Negativo y Extremadamente positivo), para ello a continuación se muestra la estructura general de nuestro dataset:

```
[ ] train.head(5)
```

	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	Neutral
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to excha...	Positive
2	3801	48753	Vagabonds	16-03-2020	Coronavirus Australia: Woolworths to give elde...	Positive
3	3802	48754	NaN	16-03-2020	My food stock is not the only one which is emp...	Positive
4	3803	48755	NaN	16-03-2020	Me, ready to go at supermarket during the #COV...	Extremely Negative

Tabla 1: Estructura general del dataset

Para empezar con el proyecto lo primero que realizamos fue determinar de que manera cargaríamos nuestro dataset, por lo que se propuso tenerlo directamente en nuestro repositorio de Github y llamarlo mediante un URL de tipo raw:

Imagen 2: Captura de pantalla donde se muestra la importanción del dataset

Una vez cargada nuestro dataset lo primero que se hace es un análisis exploratorio breve en donde se ve el tamaño de la muestra o dataset además e comprobar si existen datos tanto nulos como repetidos:

Imagen 3: Captura de pantalla donde se muestra el tamaño del dataset además de la cantidad de datos nulos

Cabe destacar que tras estos pasos notamos la carencia de datos respecto a la ubicación o procedencia de los tweets, sin embargo, para el análisis que será realizado no se necesita esa información.

4. Analítica de datos y primeras conjeturas

Como previamente se mencionó, el dataset o conjunto de datos original tiene contemplado una categorización sentimental de 5 etiquetas, sin embargo, para este proyecto se decidió realizar una nueva re-categorización cambiando las siguientes etiquetas:

- Extremadamente positivo => Positivo
- Extremadamente negativo => Negativo

Las razones para justificar este cambio son principalmente 2:

1. Al tener una menor cantidad de etiquetas podemos emplear menos cantidad de recursos de computo para procesar nuestra información
2. Si bien ya el hablar de qué es negativo, positivo o incluso neutro resulta en parte subjetivo, el medir o concretar algo como “extremadamente positivo” o “extremadamente negativo”, puede resultar todavía más complejo.

A continuación, se muestra un diagrama de la distribución de la cantidad de tweets antes de ser re-categorizados:

```
[ ]  print(concData.Sentiment.value_counts())
```

```
Positive              11422
Negative               9917
Neutral                7713
Extremely Positive     6624
Extremely Negative     5481
Name: Sentiment, dtype: int64
```

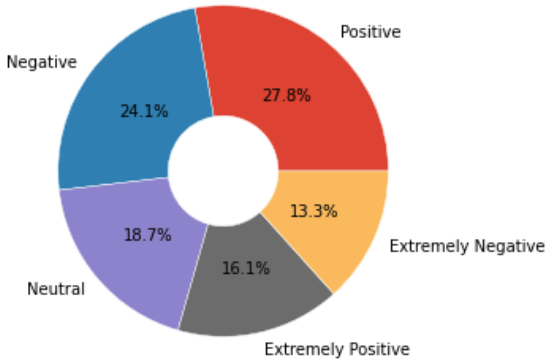


Imagen 4: Gráfica de distribución de datos previo a ser recategorizada

Tras redefinir las etiquetas solo en Negativas, positivas y neutrales, el resultado tras re-categorizar nuestros datos es el siguiente:

```
Distribución previa:
  Positive             11422
Negative               9917
Neutral                7713
Extremely Positive     6624
Extremely Negative     5481
Name: Sentiment, dtype: int64


Distribución actual:
 positive    18046
negative    15398
neutral      7713
Name: sentiment, dtype: int64
```

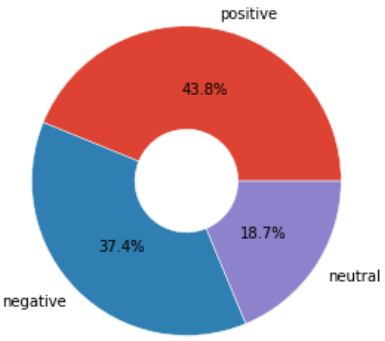


Imagen 5: Gráfica de distribución de datos tras haber sido re-categorizada

Con base a la nueva distribución y observamos que tenemos muestras parcialmente del mismo tamaño, es como continuamos con algunas primeras conjeturas o premisas para validar la forma en que fueron categorizados los datos

Cantidad de Caracteres

A continuación, se muestra la cantidad de letras o caracteres que tienen cada uno de los tweets dependiendo del tipo de sentimiento que reflejan:

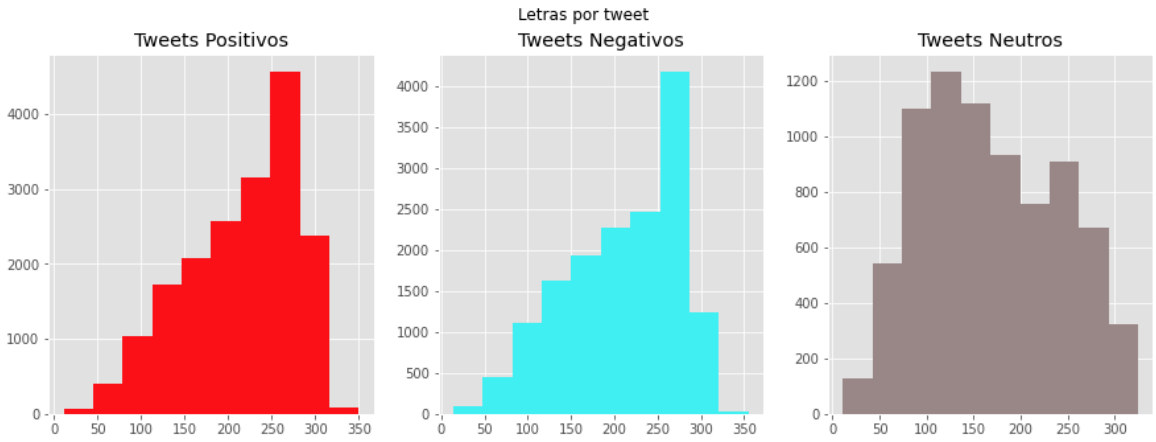


Imagen 6: Gráficas de cantidad de caracteres por tweet

Podemos observar a priori que notamos ciertos patrones de asimetría en las respectivas distribuciones, sin embargo, también observamos que el promedio de caracteres en los tweets neutros suele ser mucho menor respecto a la cantidad usada en los tweets positivos y negativos

Distribución y promedio de palabras

A continuación, se muestra la cantidad de letras por palabras que tienen cada uno de los tweets dependiendo del tipo de sentimiento que reflejan:

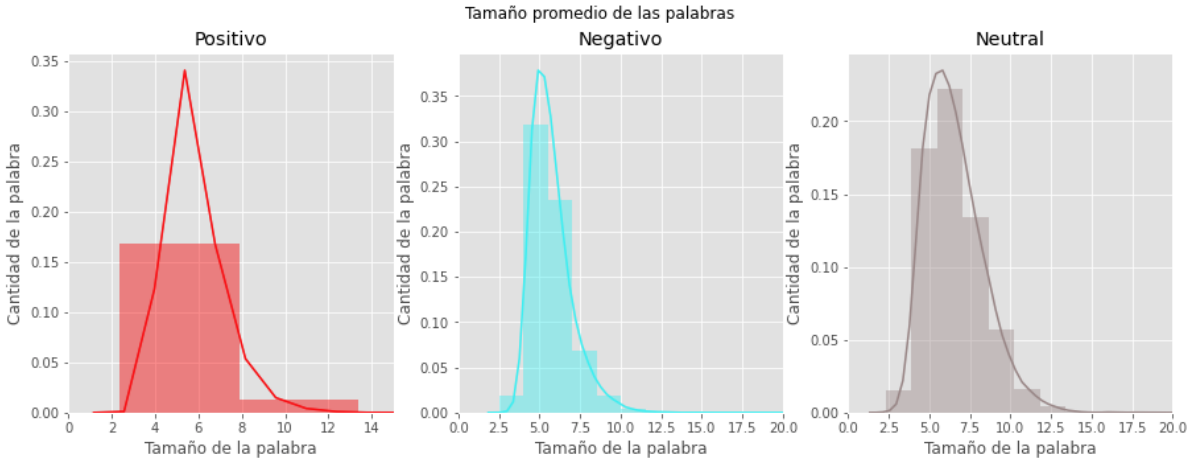


Imagen 6: Gráficas de cantidad de palabras por tweet

A priori podemos observar patrones como el que los mensajes neutros de manera general manejan palabras mucho más extensas en cantidad de letras, mientras que los mensajes positivos suelen tener una menor cantidad de letras por palabras,es decir, los mensajes motivadores suelen ser pequeños (*Para reafirmar esta hipótesis haremos uso de wordclouds como forma de visualización de las palabras*)

Al igual que en el apartado anterior podemos observar un comportamiento relativamente similar en los tweets neutrales, donde la media de la distribución de las palabras tiende a ser menor respectos a los tweets positivos y negativos.

Por otro lado, podemos llegar a observar nuevamente umna desviación de datos hacia la izquierda, es decir una simetría negativa que probablemente se deba a factores como la cantidad de palabras/carácteres que se puede usar en Twitter

TF-IDF

En primera instancia, por el tamaño del vocabulario y la cantidad de documentos en el corpus, la gran mayoría de los datos se muestran como ceros (Ver en tabla 2); esto se debe a que hay muchas palabras en el vocabulario que aparecen en pocos documentos, y, como sólo se muestra una parte

de todos los documentos, en muchos casos no aparece la columna correspondiente al documento donde una palabra dada aparece.

Sin embargo, en el caso de palabras muy frecuentes (como “stores”, que aparece en la salida actual del programa), sí se pueden observar datos correspondientes a su relevancia en ciertos documentos. Entre mayor sea el número, mayor es la relevancia de la palabra en el documento. Esta relevancia la una palabra es proporcional a la cantidad de información que brinda sobre su contexto (una oración, un documento o un conjunto de datos completo). Es decir, las palabras más relevantes son aquellas que nos ayudarían, como seres humanos, a comprender mejor un documento completo sin leerlo todo (Ver en imágenes de wordclouds).

	0	1	2	3	4	5	6	7	8	9	10	11
theskipper	0	0	0	0	0	0	0	0	0	0	0	0
wuhanflu	0	0	0	0	0	0	0	0	0	0	0	0
hooghly	0	0	0	0	0	0	0	0	0	0	0	0
wk	0	0	0	0	0	0	0	0	0	0	0	0
hifi	0	0	0	0	0	0	0	0	0	0	0	0
mistakes	0	0	0	0	0	0	0	0	0	0	0	0
supportlivemu	0	0	0	0	0	0	0	0	0	0	0	0
___	0	0	0	0	0	0	0	0	0	0	0	0
everchanging	0	0	0	0	0	0	0	0	0	0	0	0
procedures	0	0	0	0	0	0	0	0	0	0	0	0
sanitizepeople	0	0	0	0	0	0	0	0	0	0	0	0
dprinting	0	0	0	0	0	0	0	0	0	0	0	0
jacknicholson	0	0	0	0	0	0	0	0	0	0	0	0
communitybased	0	0	0	0	0	0	0	0	0	0	0	0
tossing	0	0	0	0	0	0	0	0	0	0	0	0
wiring	0	0	0	0	0	0	0	0	0	0	0	0
updated	0	0	0	0	0	0	0	0	0	0	0	0
rigging	0	0	0	0	0	0	0	0	0	0	0	0
thisâ	0	0	0	0	0	0	0	0	0	0	0	0
pocketsized	0	0	0	0	0	0	0	0	0	0	0	0
grocerystorewarehouse	0	0	0	0	0	0	0	0	0	0	0	0
hav	0	0	0	0	0	0	0	0	0	0	0	0
ogunrinde	0	0	0	0	0	0	0	0	0	0	0	0
shrinking	0	0	0	0	0	0	0	0	0	0	0	0
fecker	0	0	0	0	0	0	0	0	0	0	0	0
meansofproduction	0	0	0	0	0	0	0	0	0	0	0	0
increases	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 2: TF-IDF obtenido

Distribución de palabras mediante Wordclouds

Para obtener la distribución de las palabras más habituales en nuestros tweets previamente se realizó una limpieza en nuestros datos mediante expresiones regulares para omitir datos que no nos proporcionaban información relevante, es el caso de:

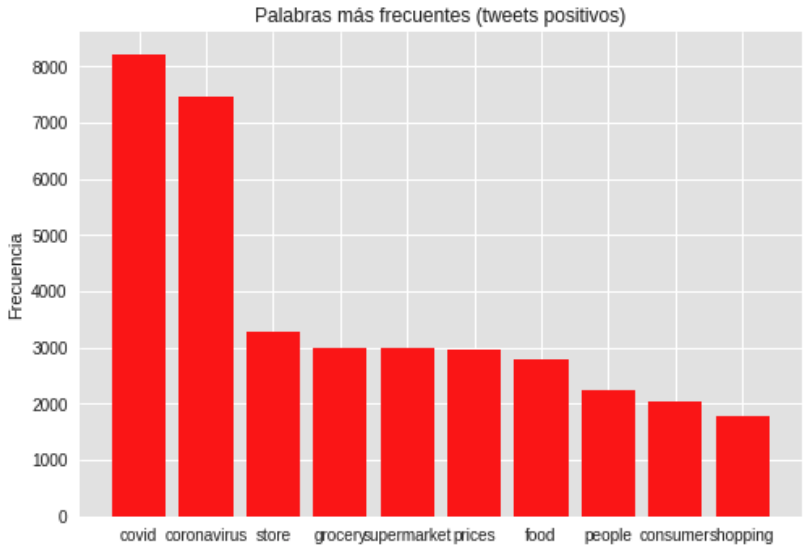
- Etiquetas a otras personas
- Links o referencias externas en los mismos tweets
- Hashtags

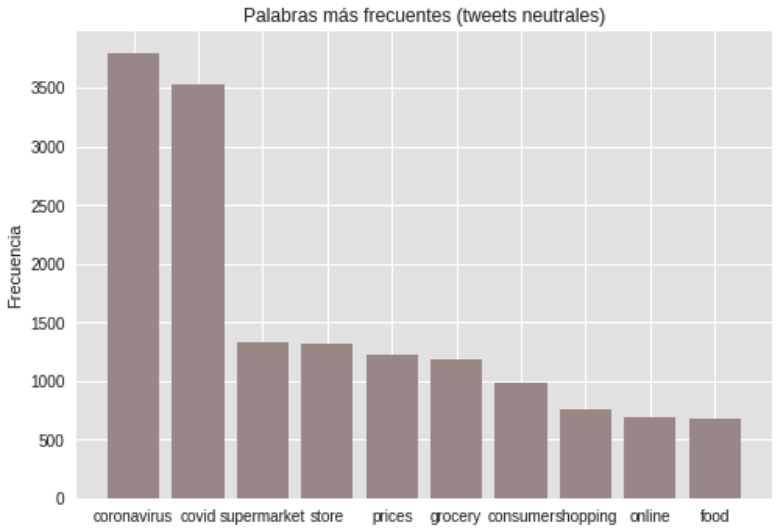
Cabe destacar que, si bien para nuestro análisis esta información no fue tomada en cuenta, podría tomarse de forma paralela e incluso validar si existe una correlación entre el sentimiento del tweet respecto a que tipo de etiquetas o hashtags se usan.

Además de limpiar esta información de nuestros tweets, también se omitieron stopwords del inglés empleando como principal recurso la biblioteca NLTK. A continuación, se muestran los resultados obtenidos tras la limpieza de tweets:

	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment	sentiment	Clean_Tweet
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	Neutral	neutral	
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to excha...	Positive	positive	advice talk neighbours family exchange phone n...

Tras la limpieza y homogenización de nuestros datos, se prosiguió a realizar debidamente un análisis del comportamiento de las palabras dentro de cada uno de nuestras etiquetas. A continuación, se muestras las palabras más utilizadas en los tweets dependiendo del tipo de sentimiento:





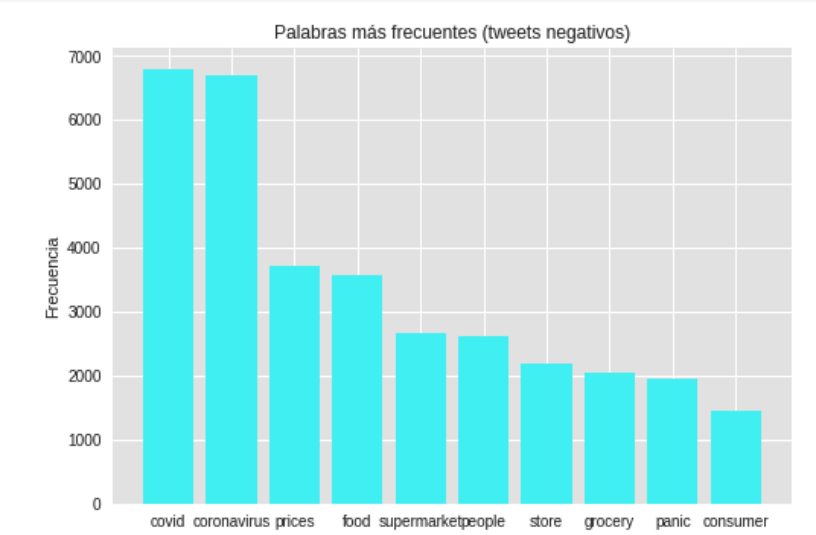


Imagen 7,8,9: Gráficas de palabras más frecuentes respecto al tipo de tweet

Por otro lado y como otra alternativa visual y comúnmente utilizada para la distribución de palabras, se emplearon wordclouds para cada tipo de tweet:

Imagen 10: Wordcloud de palabras en tweets positivos

Imagen 11: Wordcloud de palabras en tweets neutros

Imagen 12: Wordcloud de palabras en tweets negativos

Evidentemente, las palabras más comunes en todos los casos fueron “COVID” y “coronavirus”, lo único destacable al respecto es que, en los tweets neutros, “coronavirus” tuvo una mayor prevalencia que “COVID”. El resto de las palabras más frecuentes se mantuvo similar también, siendo “prices”, “supermarket”, “food”, “store” y “grocery” otras de las palabras más comunes. El orden de su frecuencia varía ligeramente entre los diferentes sentimientos.

En el caso de los tweets positivos, las palabras más utilizadas se dieron en el orden de “store”, “grocery”, “supermarket”, “prices” y “food”, obviando las mencionadas “COVID” y “coronavirus”. Es posible que muchos de los tweets positivos hayan sido sobre consejos acerca de abastecerse de los suministros necesarios al inicio de la crisis de la pandemia. Una palabra que aparece aquí, pero no en los casos del ranking de palabras de los tweets negativos y neutros, es “people”.

En el caso de los tweets negativos, las palabras más utilizadas se dieron en el orden de “prices”, “food”, “supermarket”, “people” y “store”. Algo destacable es cómo “prices” y “food” toman los primeros puestos, lo que puede deberse a la gente quejándose de la subida de precios en la comida y en general a causa de la pandemia. Otro dato importante es que aquí aparece la palabra “panic”, la cual de por sí tiene una denotación negativa.

Finalmente, en el caso de los tweets neutros, las palabras más utilizadas se dieron en el orden de “supermarket”, “store”, “prices”, “grocery”, “consumer” y “shopping”. En este caso, la palabra “food” también aparece, pero ya es la última de las 10 palabras más repetidas, siendo que “consumer” se vuelve más frecuente que en los otros tweets. En este caso, la palabra que aparece aquí, pero no en los tweets positivos y negativos, es “online”; esto puede reflejar que los tweets neutros se enfocan en este aspecto que trajo la pandemia, dándole más importancia a las actividades en línea, como las compras y la escuela.

5. Construcción del Modelo de Inteligencia Artificial

Pre-procesado de datos

Como parte esencial previa a cualquier uso de modelo de inteligencia artificial, es el discretizar nuestros datos, por ejemplo, en el caso de las etiquetas de nuestros tweets se empleó una simple asimilación a valores enteros:

```
[ ]  re_classes = {'negative': 0,
                  'neutral': 1,
                  'positive':2
                 }

     labels = ['negative', 'neutral', 'positive']
```

Imagen 13: Discretización de etiquetas

Por otro lado, y respecto al corpus o datos de cada tweet, lo que se realizó fue una tokenización, es decir asignar un valor numérico a cada una de nuestras palabras para poder crear un bagword el cual pueda ser asimilado por un modelo de inteligencia artificial:

```
Texto original:  coronavirus australia woolworths give elderly disabled dedicated shopping hours amid covid outbreak


Texto después del tokenizing:
 [ 2552  1808 13428  3746  1259  5472  7472   189  9329   797  7473  3295
     4     3  1386   229    15 10936   235  1567   207    72 10937   127
  1049   571  3178  3087     1     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0]
```

Imagen 14: Tokenazing de contenido de cada tweet

Modelo de IA: Red Neuronal RNN

Para poder emplear un modelo de inteligencia artificial lo primero que debe plantearse es qué tipo de modelo será empleado, en este caso particular y debido al tipo de problema al que nos estamos enfrentado que es el categorizar y clasificar nuestros datos, es como se escogió una Red Neuronal, que previamente se ha detallado como es que funcionan.

Sin embargo, el primer intento y red neuronal que se escogió fue de tipo ***LSTM*** o *Long Short-Term Memory* que resultan bastante favorables sobre todo para la clasificación de información que tiene secuencias o comportamientos como el que podemos encontrar en el análisis de textos. Sin embargo, tras realizar varios intentos, al final observamos que una red neuronal de tipo ***GRU*** nos ofrecía un mejor resultado (Esta comparación de resultados se muestra en el apartado de resultados).

A continuación, se muestra la estructura general de nuestra Red Neuronal:

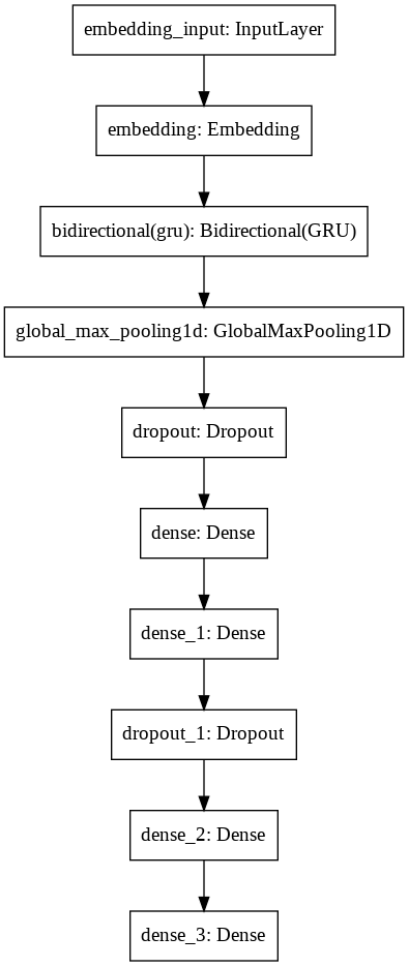


Imagen 15: Diagrama estructural de nuestra red neuronal GRU

Podemos observar que nuestra red neuronal se compone de una capa de entrada seguida de una capa bidireccional de tipo GRU que además esta seguida de otras dos capas de densidad que están apoyadas por 2 funciones de activación (Una sigmoide y una softmax)

Recordemos que para resolver el problema del *gradiente de desaparición* de un RNN estándar, las redes de tipo GRU utilizan los llamados “puerta de actualización” y “puerta de reinicio”.

Básicamente, estas puertas son dos vectores que deciden qué información se debe pasar a la salida. Lo especial de estas es que pueden ser entrenadas para guardar información de hace mucho tiempo, sin considerar el tiempo o eliminar información que es irrelevante para la predicción.

Es decir, a diferencia de las redes neuronales de tipo LSTM que si bien se parecen mucho, las redes neuronales de tipo GRU mediante estos vectores contemplan una forma algo distinta de asimilar la información de salida (Ver imagen 16):

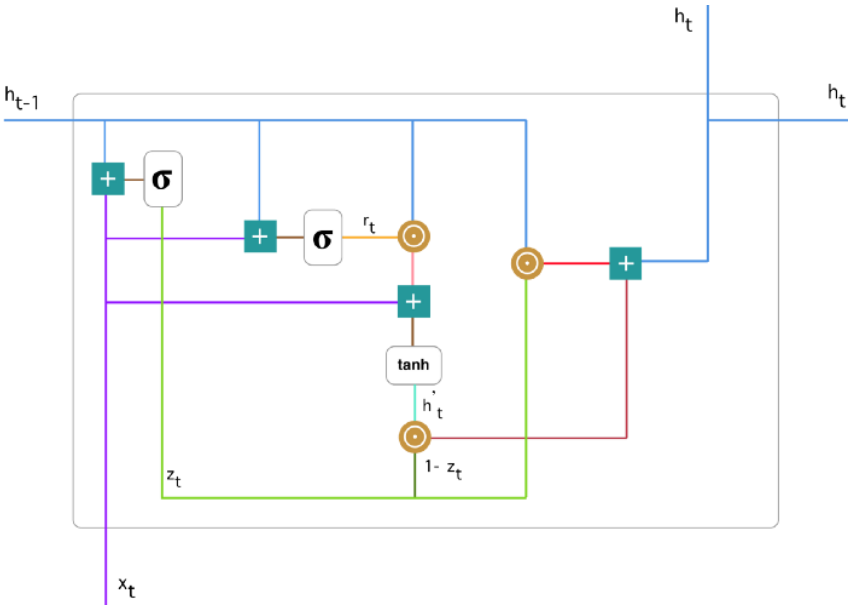


Imagen 16: Estructura de una Red neuronal de tipo GRU

Entrenamiento de la Red Neuronal

Como es bien sabido, de nuestro dataset es común emplear para el “train” o entrenamiento de nuestro modelo alrededor de 70 a 80%, mientras que para el test se suele emplear la cantidad restante, en nuestro caso para el entrenamiento se empleó un 80% del total de los datos obteniendo los siguientes resultados:

```
[ ] history = model.fit(X, y_train, epochs=EPOCHS, validation_split=0.12, batch_size=BATCH_SIZE) # cambiamos validation

    Epoch 1/4
    906/906 [==============================] - 35s 34ms/step - loss: 0.8713 - accuracy: 0.6081 - val_loss: 0.6574 - val_accuracy: 0.7818
    Epoch 2/4
    906/906 [==============================] - 32s 35ms/step - loss: 0.5352 - accuracy: 0.8270 - val_loss: 0.5322 - val_accuracy: 0.8206
    Epoch 3/4
    906/906 [==============================] - 30s 33ms/step - loss: 0.3685 - accuracy: 0.8929 - val_loss: 0.5227 - val_accuracy: 0.8274
    Epoch 4/4
    906/906 [==============================] - 32s 35ms/step - loss: 0.2758 - accuracy: 0.9242 - val_loss: 0.5307 - val_accuracy: 0.8211
```

Imagen 17: Resultados tras fit y entrenamiento del modelo de AI

Posteriormente, se empleó el restante de nuestros datos para el test del modelo, obteniendo los siguientes resultados que a priori muestran un buen comportamiento de nuestro modelo:

```
[ ] loss, acc = model.evaluate(X_test,y_test)
    print('Test loss (Perdida): ', loss)
    print("Test Accuracy (Precisión):", acc)

    258/258 [==============================] - 3s 12ms/step - loss: 0.5355 - accuracy: 0.8201
    Test loss (Perdida):  0.5354605317115784
    Test Accuracy (Precisión): 0.8200923204421997
```

Imagen 18: Resultados tras evaluación del modelo con datos de prueba o test

Resultados

Análisis estadístico

Con base a nuestro modelo una vez entrenado, se prosiguió a realizar las primeras validaciones estadísticas. A continuación, se muestran los datos obtenidos del entrenamiento y validación de nuestro modelo:

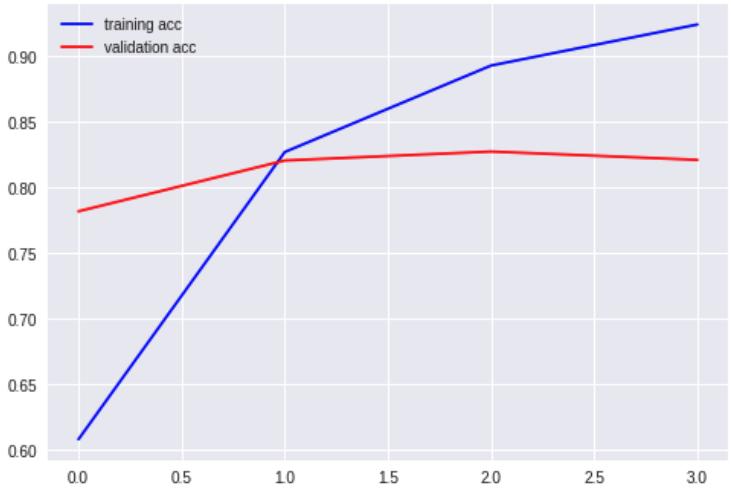


Imagen 19: Gráfica de validación y entrenamiento del modelo en cada epoch del modelo

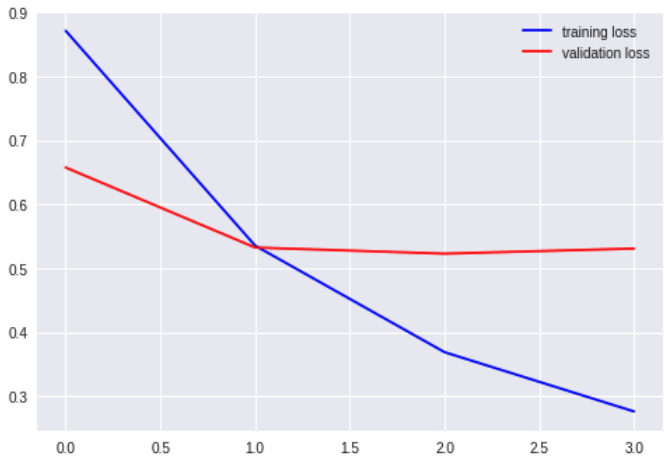


Imagen 20: Gráfica de validación y entrenamiento de la perdida de datos del modelo en cada epoch del modelo

Como bien se mencionó en el apartado de la construcción del modelo de inteligencia artificial al inicio de nuestro proyecto el tipo de red neuronal no era de tipo GRU sino de tipo LSTM, la razón de por qué se decidió realizar un cambio respecto al tipo de red y de funciones de activación se puede apreciar en las matrices de confusión obtenidas.

A continuación, se muestra la matriz de confusión de nuestra red de tipo LSTM:

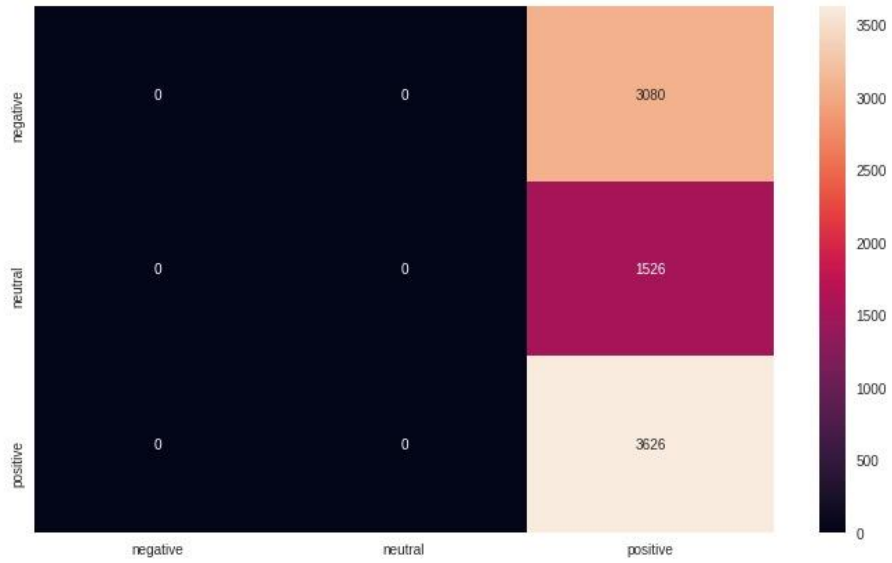


Imagen 21: Matriz de confusión en red tipo LSTM

Observamos que hay un evidente sesgo de información en los tweets negativos y neutrales, donde prácticamente se han clasificado de forma errónea al ser categorizados como positivos.

Por otro lado, y contemplado a una red neuronal de tipo GRU, se obtuvo los siguientes resultados:

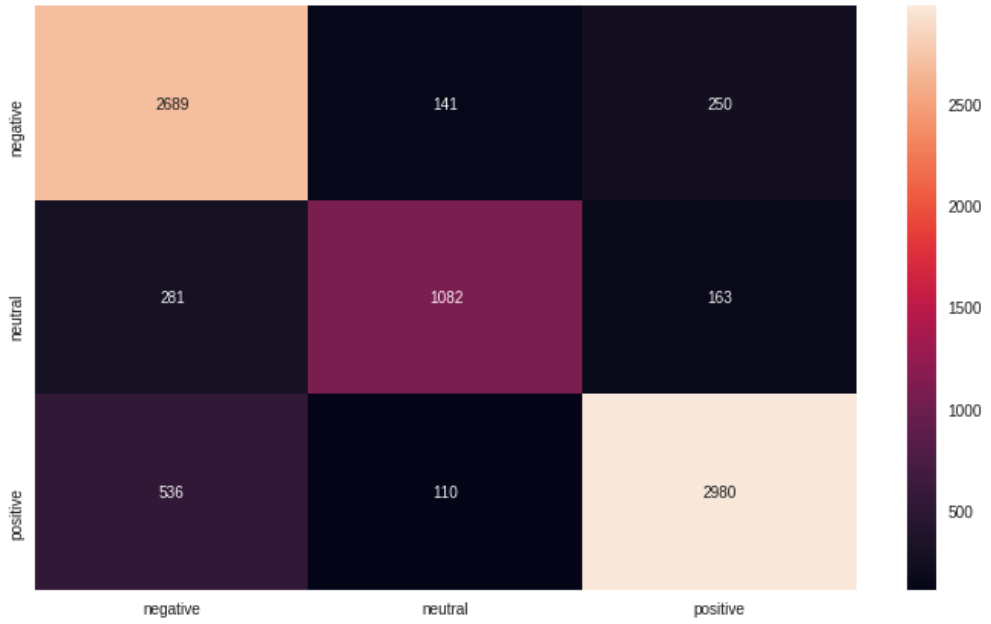


Imagen 21: Matriz de confusión en red tipo GRU

A priori observamos que la diagonal de la matriz que es donde se dan nuestros casos positivos-positivos son los que mayor abundancia tienen, lo cual es un buen indicio y además corrobora ese 82% de precisión que tiene nuestro modelo.

Análisis en casos particulares

Si bien el contemplar casos particulares para una muestra grande no supone realmente un error o afirmación determinante, particularmente decidimos realizar este apartado sobre todo por lo visual que resulta.

A continuación, se muestra el caso particular del tweet 41154 el cual es positivo:

```
[ ] dataFrame.tail(10)
```

	sentiment	Clean_Tweet
41147	negative	yâall really shitting much home covid coronavi...
41148	positive	uv light sterilizer sanitizer mask mobile phon...
41149	negative	still shocked number toronto supermarket emplo...
41150	positive	never weâd situation world going supermarket p...
41151	positive	definitely man feel like fall need honor heroe...
41152	neutral	airline pilots offering stock supermarket shel...
41153	negative	response complaint provided citing covid relat...
41154	positive	know itâs getting tough rationing toilet paper...
41155	neutral	wrong smell hand sanitizer starting turn coron...
41156	negative	well newused rift going amazon rn although nor...

Tabla 3: Tweets para muestreo particular

Tras escoger el tweet y meterlo a nuestro modelo a continuación observamos el resultado particular:

```
[ ] string = dataFrame.iloc[41151]
    #string = "BREAKING: The UK is sending India 9 airline container loads of medical supplies this week to help fight against th
    string

    sentiment                                              positive
    Clean_Tweet    definitely man feel like fall need honor heroe...
    Name: 41151, dtype: object
```

NOTA FINAL:

- x == 0:
  - Tweet Negativo
- x == 1:
  - Tweet Neutro
- x == 2:
  - Tweet Positivo

```
[ ] testOne = tokenizer.texts_to_sequences(string)
    testOne = pad_sequences(testOne, padding='post')
    testTweet = model.predict_classes(testOne)
    testTweet

    WARNING:tensorflow:Model was constructed with shape (None, 261) for input KerasTensor(type_spec=TensorSpec(shape=(None, 261),
    array([2, 2])
```

Imagen 22: resultado arrojado con nuestro modelo ya entrenado

Conclusiones

El Análisis y Procesamiento de Textos Inteligentes son una de las ramas más modernas de la inteligencia artificial que sin duda tendrá una gran importancia en la construcción y análisis de fenómenos sociales.

Por otro lado, los resultados de nuestro modelo de inteligencia artificial nos parecen bastante agradables de observar, pues a pesar de no tener una precisión tan alta como es un 90% de accuracy, la realidad es que el obtener arriba de 80% en un tema tan moderno además de complejo y que en muchos casos puede llegar a ser bastante subjetivo es bastante satisfactorio.

A su vez y como parte critica de nuestro trabajo sabemos que pueden existe varias formas en las que podemos mejorar el resultado obtenido, a priori lo primero podría ser encontrar otros ejes a contemplar dentro de nuestro modelo como es el caso de considerar y validar si hay correlación con otros elementos como los hashtags o referencias en los mismos tweets, o incluso y si es el caso cambiar el modelo a alguno más moderno y de mayor versatilidad como son Transformers.

Trabajo a futuro:

- Emplear Transformers como BERT o ELMo que son modelos mucho más versátiles y con capacidades para encontrar relaciones entre datos de una forma mucho más acertiva

- Contemplar nuevos ejes para la clasificación y validación de cada etiqueta en nuestros datos

Referencias

[1] Kaggle. Start with more than a blinking cursor. Recuperado el 2 de agosto de 2021, de https://www.kaggle.com

[2] Towards data science. How Recurrent Neural Networks work. Recuperado el 2 de agosto de 2021, de https://towardsdatascience.com/learn-how-recurrent-neural-networks-work-84e975feaaf7

[3] Alice Zheng. Evaluating Machine Learning Models. Recuperado el 2 de agosto de www.stratoconf.com