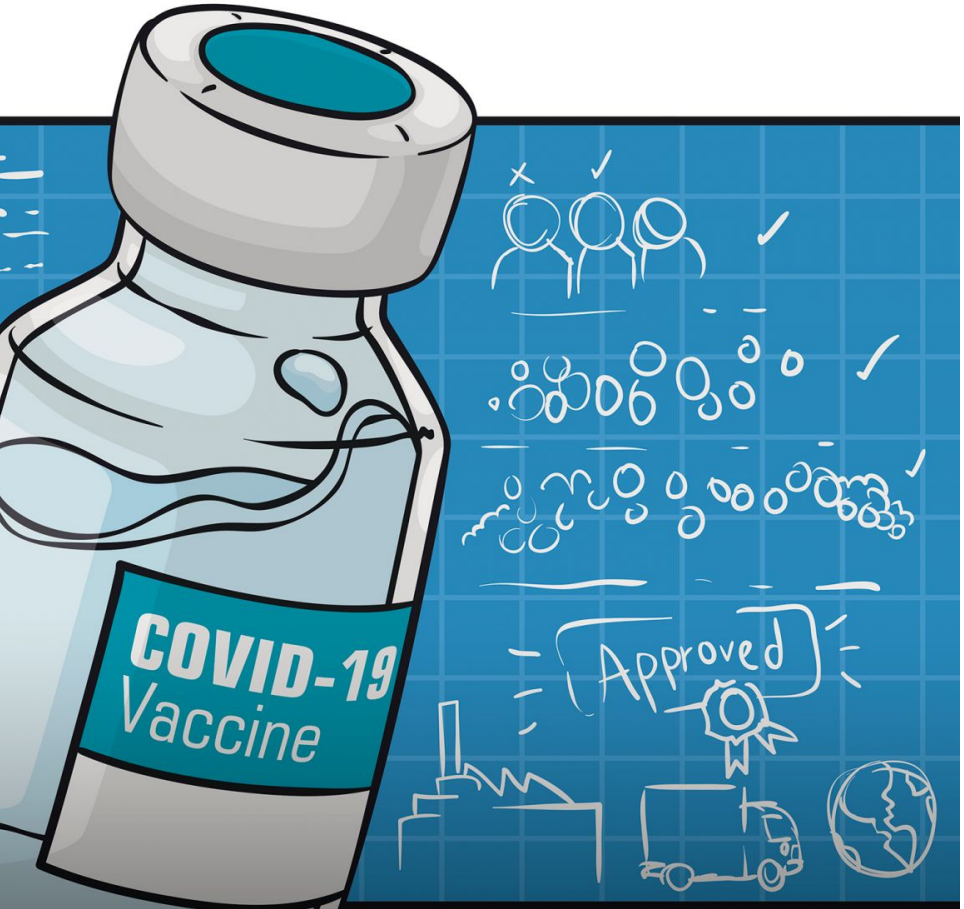
El procesamiento del lenguaje natural (PLN) es una disciplina que se encuentra en la intersección de la informática, la lingüística, la inteligencia artificial (IA) y la ciencia cognitiva.

Por: GPT-2

Twitter Sentiment Analysis and Tweet Generator Using NLP

- Fredy Alejandro Mendoza López.
- Christian Ruiz Lagos.

¿Cuales son los problema?

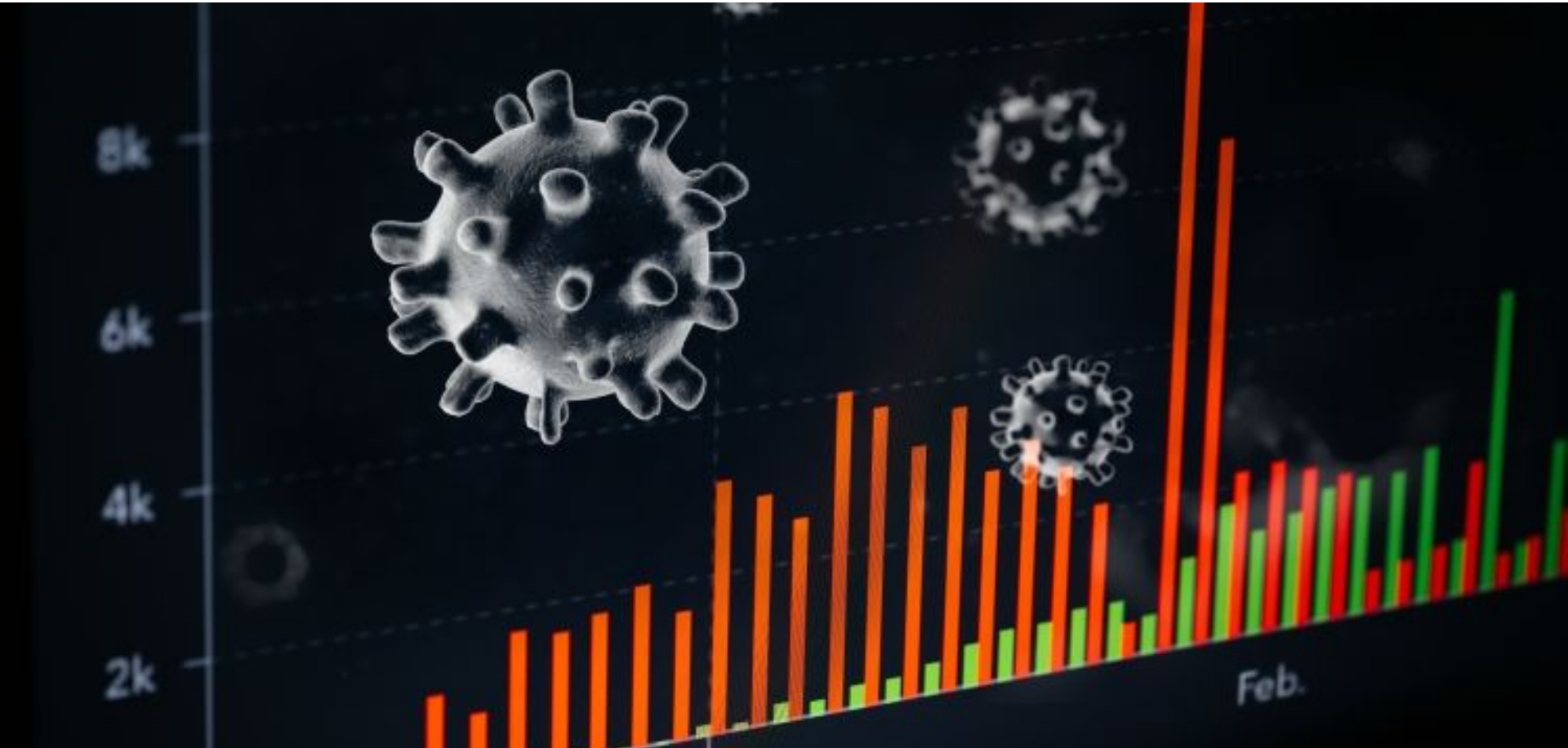


- ¿Cual es la posición de Colombia ante el tema de las vacunas?
- La desinformación, contribuye a un posicionamiento negativo en las personas.

Objetivos

- Realizar un proceso de Web Scrapping para la recolección de datos en Twitter.
- Desarrollar un modelo que permita la clasificación de tweets, con el fin de determinar una aproximación de la posición de los colombianos frente a la aplicación de la vacuna.
- Implementar un generador de tweets, con el fin de proveer información que incentive la vacunación

¿De donde obtenemos los datos?



Web Scrapping

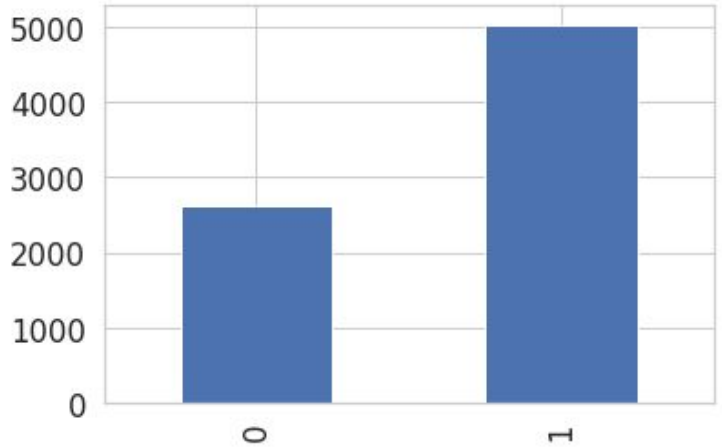
Clasificador

Construcción del dataset

#YoNoMeVacuno	0
#YoMeVacuno	1

- Username
- Text
- Tweet date

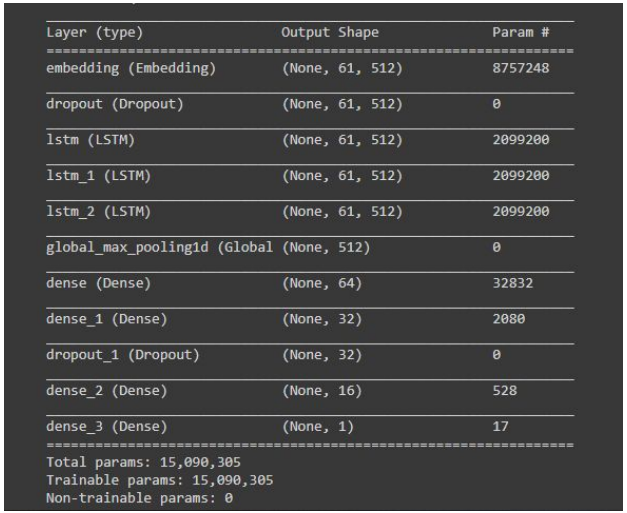
Datos



	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Nombre de usuario	Usuario	Fecha	Texto	Sentimiento
0	0	0	0	Aurora Fuentes	@paysandusiempre	2021-03-06	no dejen de mirar numero de casos positivos de...	1
1	1	1	1	majo	@majoattacks	2021-03-06	y todos los adultos mayores de mi familia tamb...	1
2	2	2	2	#YoApruebo	@xapahernandez	2021-03-06	hasta el dalai lama se vacuno y todavia alguno...	1
3	3	3	3	Ashishito\n#MascarillaBienPuesta	@jorgeapolaya	2021-03-06	contra el terruqueo contra la desinformacion d...	1
4	4	4	4	Jose Ragas	@joseragas	2021-03-06	lo que hace willax no es libertad de expresion...	1
...	...	...	...	...	...	...	...	...
7659	7659	7659	2622	lid	@ldutari	2020-11-04	ni loca aunque sea obligatoria? y ni digan que...	0
7660	7660	7660	2623	Juan Manuel	@majud03	2020-11-04	ni mi familia metanse la vacuna bien en el	0
7661	7661	7661	2624	Florencia Balcarce	@beappatt	2020-11-04	la vacuna del swine flu en 1976 ocasiono mas m...	0
7662	7662	7662	2625	No, Korruptos	@carlita_River19	2020-11-04	me causa gracia xq todo es grieta ahora la vac...	0
7663	7663	7663	2626	Josefina	@JoseDominino_	2020-11-04	Se meten cada cosa en el cuerpo y ahora salen ...	0

7664 rows × 8 columns

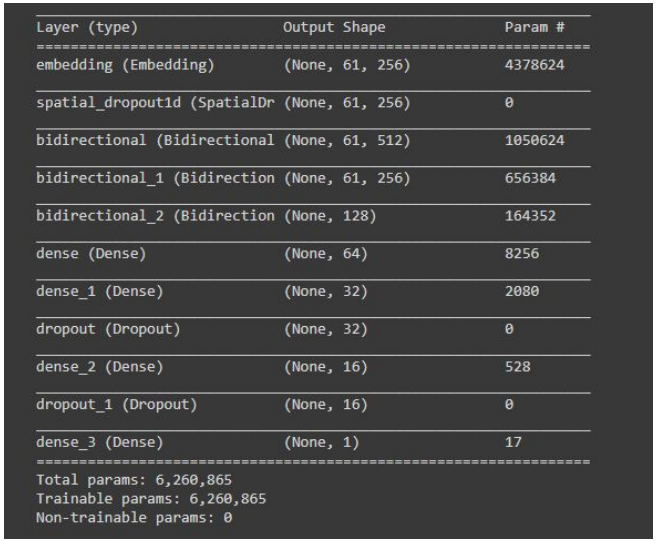
Tres modelos, dos datasets



Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 61, 512)	8757248
dropout (Dropout)	(None, 61, 512)	0
lstm (LSTM)	(None, 61, 512)	2099200
lstm_1 (LSTM)	(None, 61, 512)	2099200
lstm_2 (LSTM)	(None, 61, 512)	2099200
global_max_pooling1d (Global	(None, 512)	0
dense (Dense)	(None, 64)	32832
dense_1 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dense_3 (Dense)	(None, 1)	17

Total params: 15,090,305
Trainable params: 15,090,305
Non-trainable params: 0

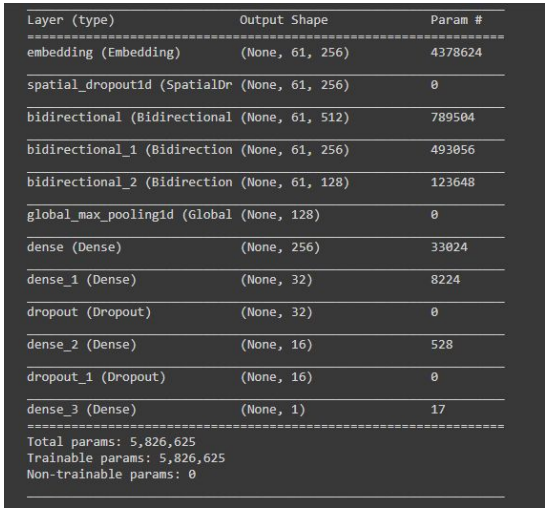
LSTM



Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 61, 256)	4378624
spatial_dropout1d (SpatialDr	(None, 61, 256)	0
bidirectional (Bidirectional	(None, 61, 512)	1050624
bidirectional_1 (Bidirection	(None, 61, 256)	656384
bidirectional_2 (Bidirection	(None, 128)	164352
dense (Dense)	(None, 64)	8256
dense_1 (Dense)	(None, 32)	2080
dropout (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_1 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 1)	17

Total params: 6,260,865
Trainable params: 6,260,865
Non-trainable params: 0

LSTM Bidirectional

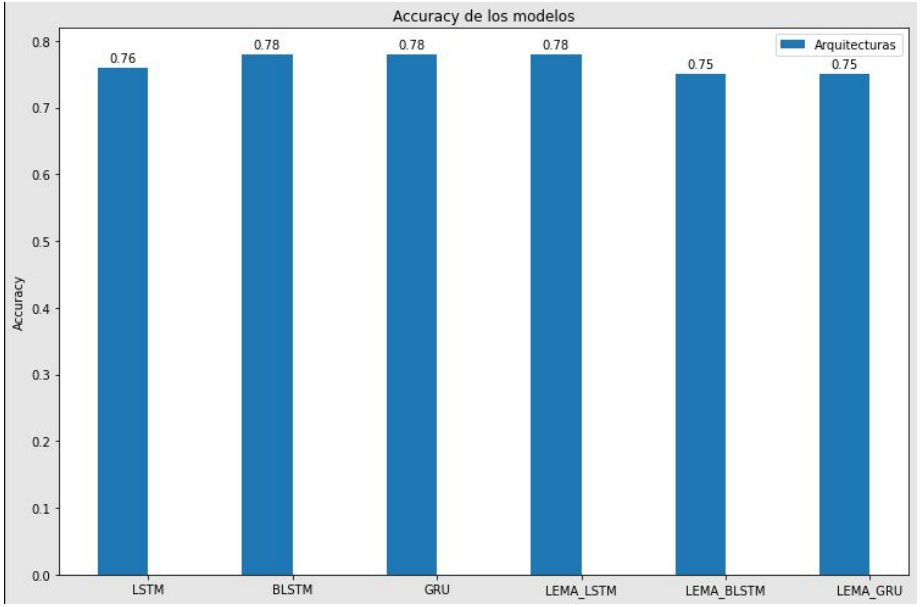


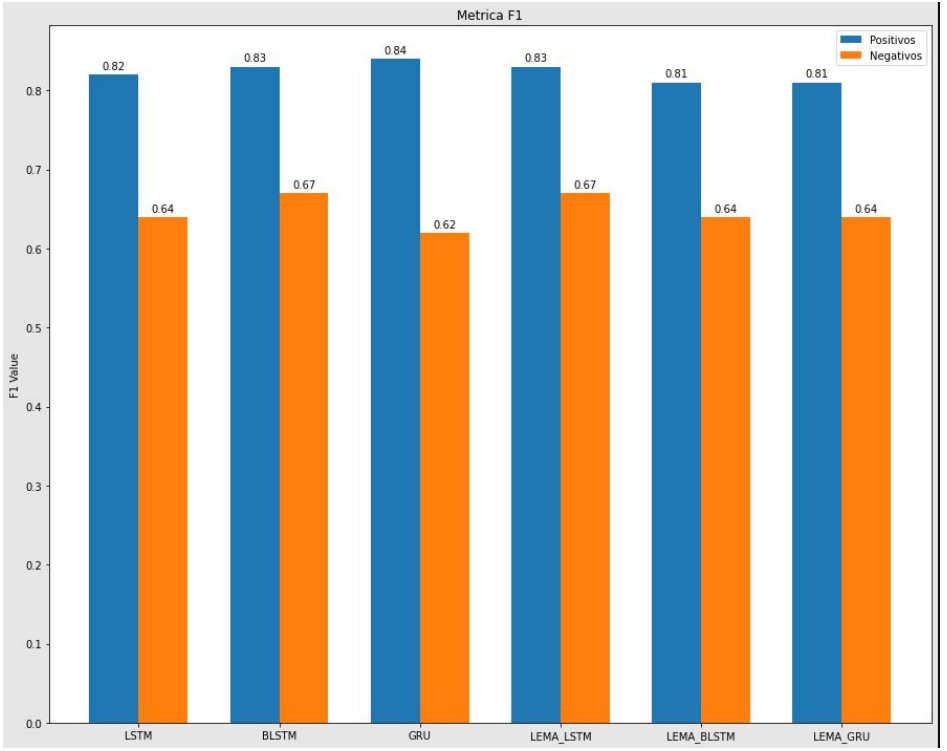
Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 61, 256)	4378624
spatial_dropout1d (SpatialDr	(None, 61, 256)	0
bidirectional (Bidirectional	(None, 61, 512)	789504
bidirectional_1 (Bidirection	(None, 61, 256)	493056
bidirectional_2 (Bidirection	(None, 61, 128)	123648
global_max_pooling1d (Global	(None, 128)	0
dense (Dense)	(None, 256)	33024
dense_1 (Dense)	(None, 32)	8224
dropout (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_1 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 1)	17

Total params: 5,826,625
Trainable params: 5,826,625
Non-trainable params: 0

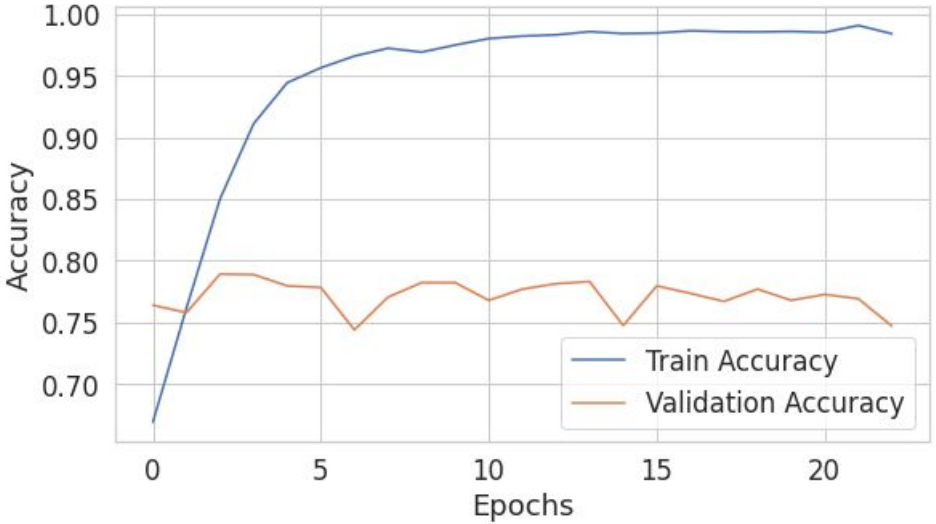
GRU

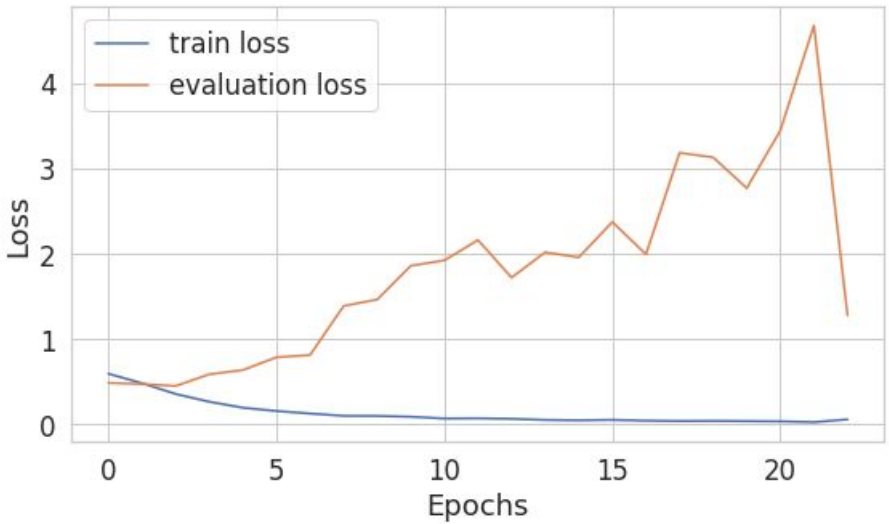
Resultados


Accuracy de los modelos
Arquitecturas
Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.76
0.78
0.78
0.78
0.75
0.75
LSTM
BLSTM
GRU
LEMA_LSTM
LEMA_BLSTM
LEMA_GRU

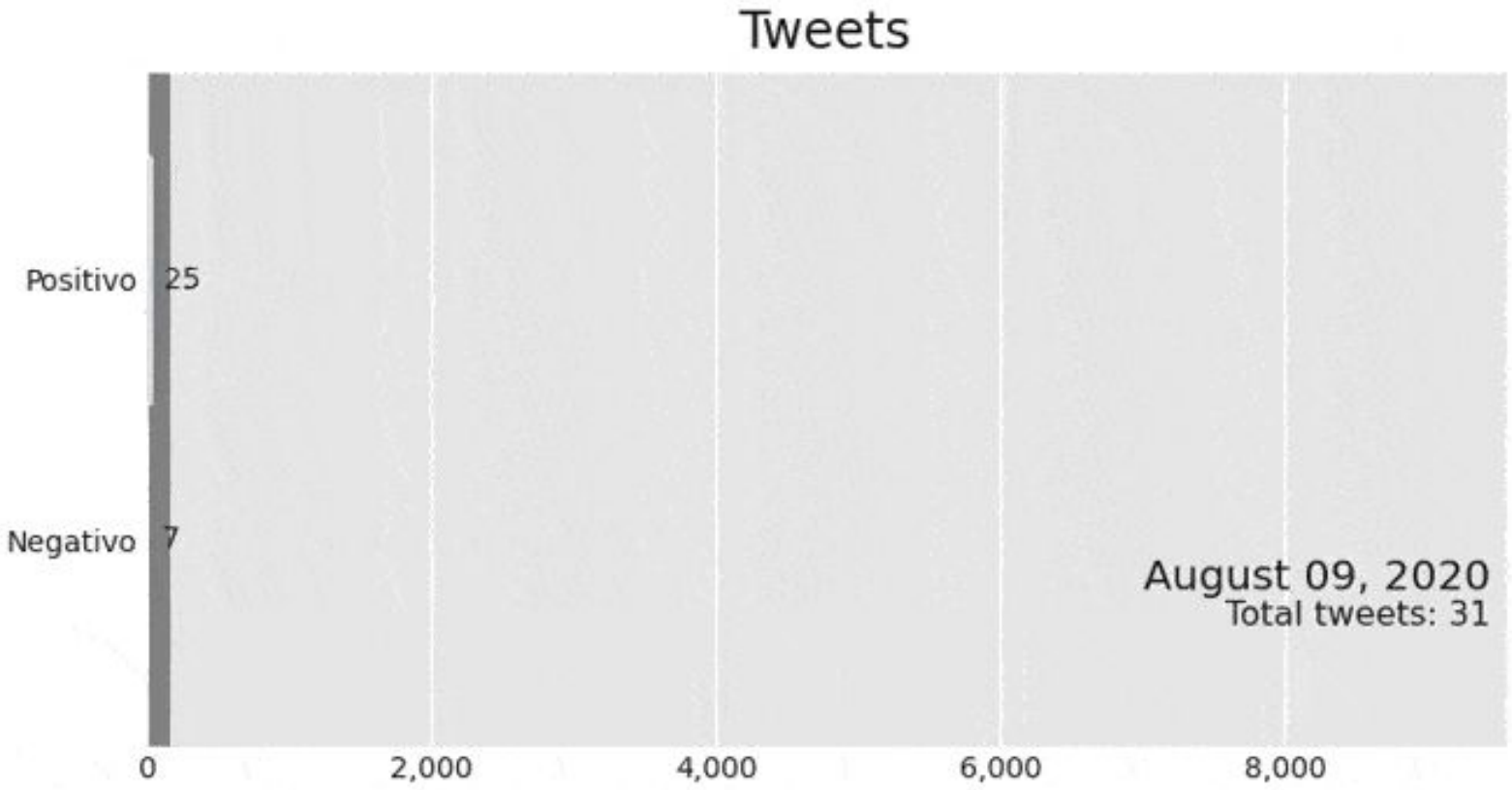

Metrica F1
Positivos
Negativos
F1 Value
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.82
0.64
0.83
0.67
0.84
0.62
0.83
0.67
0.81
0.64
0.81
0.64
LSTM
BLSTM
GRU
LEMA_LSTM
LEMA_BLSTM
LEMA_GRU

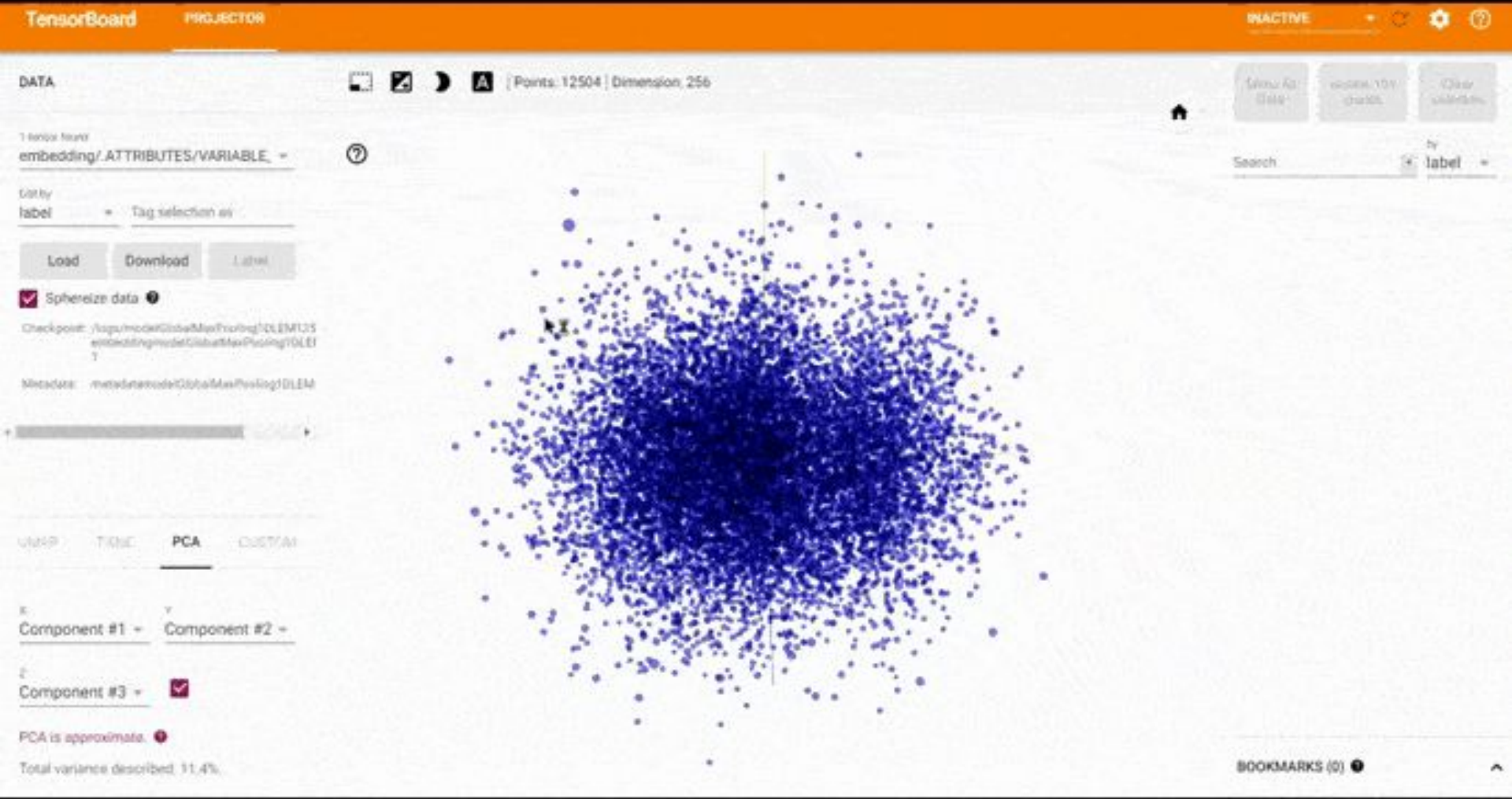
Resultados


1.00
0.95
0.90
0.85
0.80
0.75
0.70
Accuracy
0
5
10
15
20
Epochs
Train Accuracy
Validation Accuracy


train loss
evaluation loss
4
3
2
1
0
Loss
0
5
10
15
20
Epochs

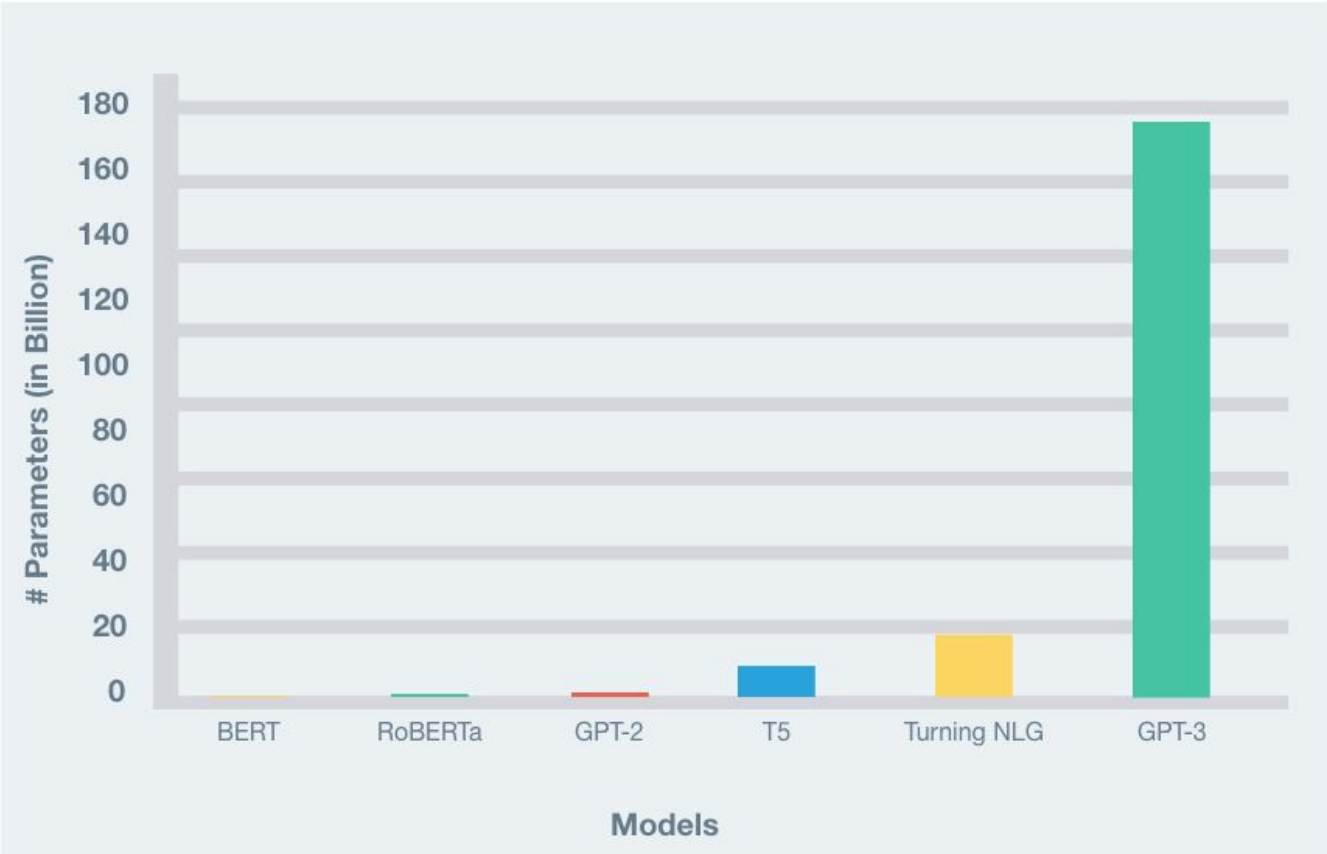
Predicción de tweets




TensorBoard
PROJECTOR
INACTIVE
DATA
Points: 12504 | Dimension: 256
embedding/.ATTRIBUTES/VARIABLE_
label
Load
Download
Sphereize data
PCA
Component #1
Component #2
Component #3
PCA is approximate.
Total variance described: 11.4%.
Search
label
BOOKMARKS (0)

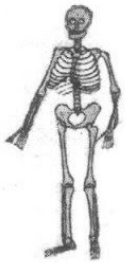
Generador de texto

State-of-the-Art
(modelos de generación de texto)



DATO CURIOSO: ENTRENAR GPT-3 CON SÓLO LA GPU MÁS POTENTE DEL MERCADO, TARDARÍA ALREDEDOR DE **355 AÑOS.**

*GPT-2 ES **1/116** VECES EL TAMAÑO DE GPT-3*



GPT-2
1.5B Parameters

GPT-3
175B Parameters

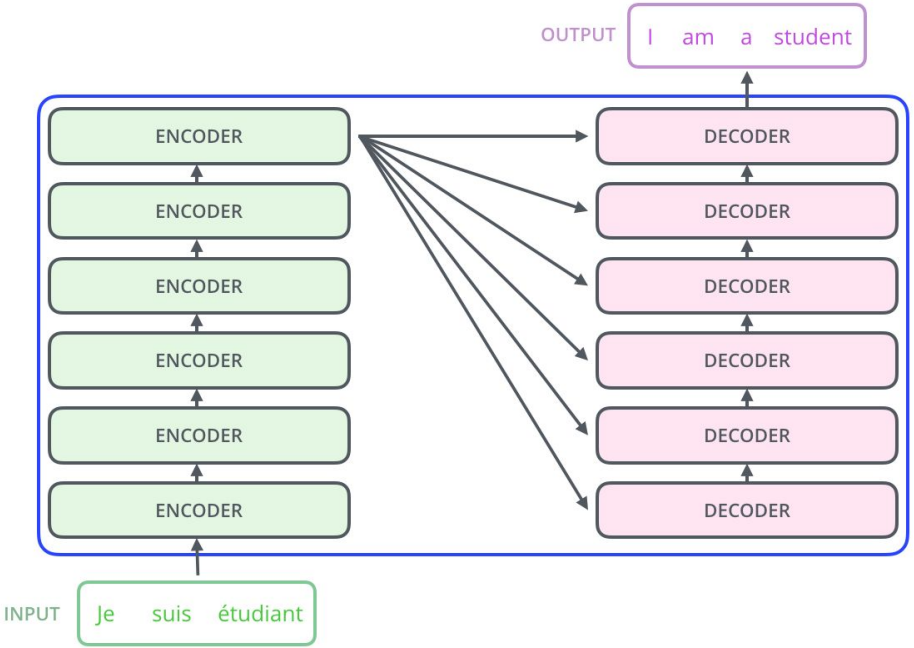
GPT-2
(Tamaños)

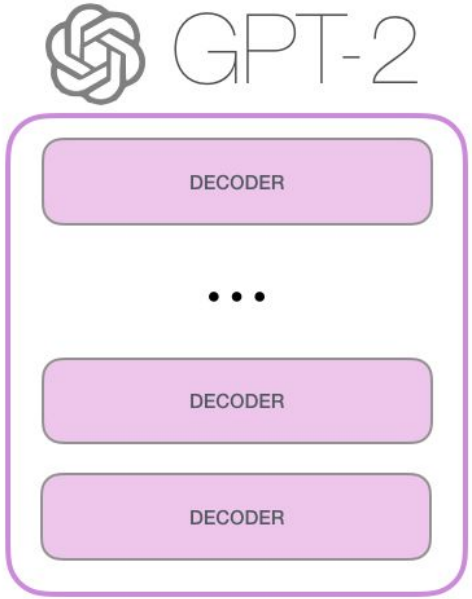
GPT-2 FUE ENTRENADA CON ALREDEDOR DE **40GB** DE TEXTO.

FINE-TUNE: Tomar los pesos de una red neuronal ya entrenada y usarla como inicialización para un nuevo modelo con datos del mismo dominio. Para el caso de estudio, se tomó los pesos aprendidos de un corpus en Inglés y se re utilizó vocabulario y embeddings comunes entre el Inglés y el Español.

Esta estrategia es factible porque las reglas gramaticales entre el Inglés y Español, pese a ser diferentes, guardan algo de similitud.

Arquitectura GPT-2
(Basada en Transformers)

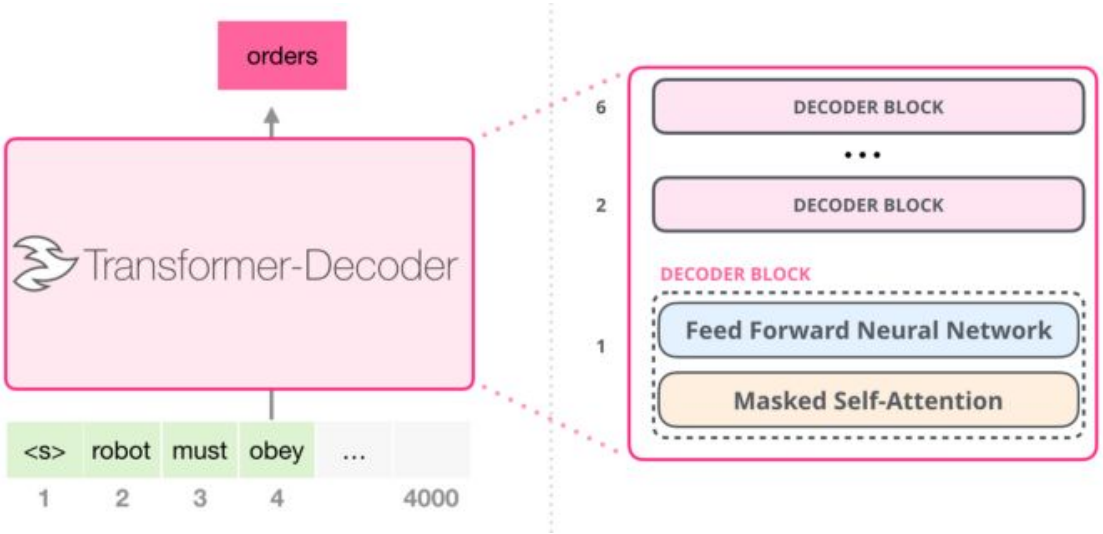


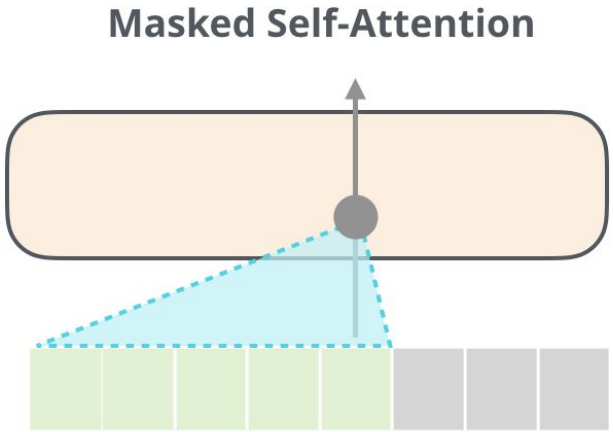


Decoder-Only-Block

El modelo GPT-2 es construido usando **bloques de decoders**. Al igual que los modelos de lenguaje tradicionales, genera **un token a la vez**.

Decoder-Only-Block





Dada una sentence como input, el propósito de la capa **‘Masked Self-Attention’** es asegurarse de que los estados no consideren los tokens que están "en el futuro" sino solo a los del "pasado" y el token actual.

Librería Huggingface
(Modelos del estado del arte fáciles de implementar)

Transformers

Descripción breve del modelo a utilizar:

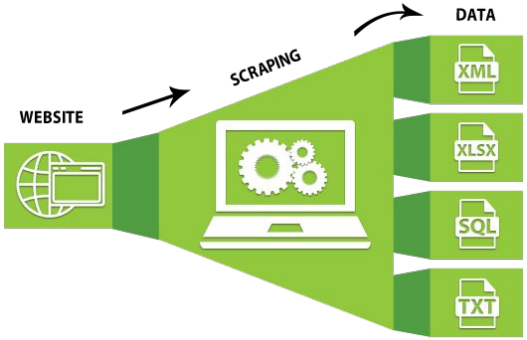
English pre-trained GPT-2 small

- 12 capas, 768-ocultas
- 124M de parámetros
- Tiempo de descarga: aproximadamente 10 minutos

English pre-trained Byte-level BPE tokenizer

- Byte-level BPE
- vocabulario de 50.257 tokens

Implementación



Pasos para llevar a cabo la implementacón:

- Cargar el dataset (datos recopilados por nosotros a través de **web scrapping**)
- Preparar el dataset y construir un **TextDataset**.
- Inicializar el **Trainer** con sus **TrainingArguments** y el modelo **GPT-2**.
- Entrenar y guardar el modelo.
- Testear el modelo.

17117		SecretaríaSaludCesar	@CesarSecSalud
17118		SecretaríaSaludCesar	@CesarSecSalud
17119	Secretaría de Desarrollo de la Salud - Córdoba.		@CordobaSalud

17120 rows × 5 columns

```python
import re
import json
from sklearn.model_selection import train_test_split


# with open('recipes.json') as f:
#     data = json.load(f)

def build_text_files(df, dest_path):
    f = open(dest_path, 'w')
    data = ''
    summaries = df['Texto'].tolist()
    for texts in summaries:
        summary = str(texts).strip()
        summary = re.sub(r"\s", " ", summary)
        data += summary + " "
    f.write(data)

train, test = train_test_split(tweets, test_size=0.15)


build_text_files(train,'train_dataset.txt')
build_text_files(test,'test_dataset.txt')

print("Train dataset length: "+str(len(train)))
print("Test dataset length: "+ str(len(test)))
```

```
Train dataset length: 14552
Test dataset length: 2568
```

```python
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("datificate/gpt2-small-spanish")

train_path = 'train_dataset.txt'
test_path = 'test_dataset.txt'
```

```
Downloading: 100% 817/817 [00:03<00:00, 264B/s]
Downloading: 100% 850k/850k [00:02<00:00, 323kB/s]
Downloading: 100% 508k/508k [00:01<00:00, 438kB/s]
Downloading: 100% 387/387 [00:00<00:00, 689B/s]
Downloading: 100% 620/620 [00:00<00:00, 6.22kB/s]
```

Los **tokenizers** se obtuvieron del modelo pre-entrenado de GPT-2 Small-Spanish, alojado en **HugginFace.**

```python
from transformers import TextDataset,DataCollatorForLanguageModeling

def load_dataset(train_path,test_path,tokenizer):
    train_dataset = TextDataset(
          tokenizer=tokenizer,
          file_path=train_path,
          block_size=128)

    test_dataset = TextDataset(
          tokenizer=tokenizer,
          file_path=test_path,
          block_size=128)

    data_collator = DataCollatorForLanguageModeling(
        tokenizer=tokenizer, mlm=False,
    )
    return train_dataset,test_dataset,data_collator
```

```python
from transformers import Trainer, TrainingArguments, AutoModelWithLMHead

model = AutoModelWithLMHead.from_pretrained("datificate/gpt2-small-spanish")


training_args = TrainingArguments(
    output_dir="./gpt2-tuits+gobierno_50", #The output directory
    overwrite_output_dir=True, #overwrite the content of the output directory
    num_train_epochs=50, # number of training epochs
    per_device_train_batch_size=32, # batch size for training
    per_device_eval_batch_size=64,  # batch size for evaluation
    eval_steps = 400, # Number of update steps between two evaluations.
    save_steps=4500, # after # steps model is saved
    warmup_steps=500,# number of warmup steps for learning rate scheduler
    prediction_loss_only=True,
    )


trainer = Trainer(
    model=model,
    args=training_args,
    data_collator=data_collator,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
)
```

Se entrenó con **50 épocas**, un batch_size de entrenamiento de 32, un batch_size de evaluation de 64. El optimizador fue **Adam** y se utilizó un **learning-rate** de 5e-5 (0.00005).

```
trainer.train()
```

[6000/6000 2:02:09, Epoch 50/50]

Step	Training Loss
500	4.504900
1000	3.710300
1500	3.347000
2000	3.087100
2500	2.880500
3000	2.709600
3500	2.566900
4000	2.452000
4500	2.364200
5000	2.293400
5500	2.246000
6000	2.218200

Para realizar **50 épocas,** tardó **2:02:09 horas** en completarse. Ahora, guardemos y módelo y véamos los resultados obtenidos.

```
trainer.save_model()
```

```
from transformers import pipeline

generador_tweets = pipeline('text-generation',model='./gpt2-gobierno', tokenizer='datificate/gpt2-small-spanish',config={'max_length':800})
```

```
generador_tweets('El covid')[0]['generated_text']
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
'El covid-19   #COVID19   #vacunassalvadoras Colombia sin vacuna!! El país en la peor crisis del mundo, donde las muertes y la falta de insumos hacen de los líderes más esenciales.
```

Twitter Bot para la generación de Tweets a favor de la vacuna


```
[ ] #@title **Función para publicar tweets**
    import time
    semillas = ['Me parece', 'En el día de hoy, ', 'El covid', 'La vacuna', 'En Colombia']
    # Create tweets
    def publicarTweets(num_tweets, int_t):
      for i in range(num_tweets):
        api.update_status(generador_tweets(semillas[i])[0]['generated_text'])
        time.sleep(int_t)
```

```
[ ] numTweets = 5 # Número de tweets a publicar.
    t = 60 # Número de segundos a esperar hasta publicar otro tweet.
    publicarTweets(numTweets, t)

    Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
    Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
    Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
    Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
    Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
```

Fue posible implementar un bot que publique Tweets por medio de la **API de Twitter**

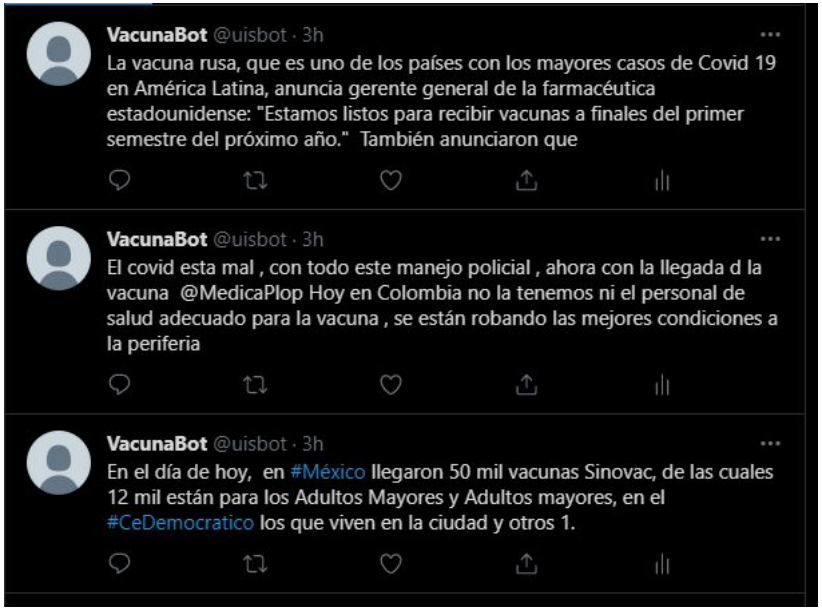
VacunaBot @uisbot · 3h
El covid-19 #COVID19 #vacunassalvadoras Colombia sin vacuna!! El país en la peor crisis del mundo, donde las muertes y la falta de insumos hacen de los líderes más esenciales. La solución

VacunaBot @uisbot · 3h
En el día de hoy, más de 1 millón de vacunados ! #YoMeVacuno A Colombia llegó el 20 de febrero la vacuna del covid - 19. No se puede hacer show con esto, basta mostrar la

VacunaBot @uisbot · 3h
Me parece que a Colombia le interesa las vacunas contra el #COVID—19. Yo me vacuno y lo haré pensando si puedo o no volver a contagiar, me pone encima de la mesa una vacuna que me protege de esta enfermedad

VacunaBot @uisbot · 3h
La vacuna llega a Colombia en el segundo semestre del 2021, porque según @DANE_Colombia , Colombia fue uno de los últimos países en aprobar la vacuna. Con la pandemia ya casi erradicada, y la vacuna aún en proceso de confirmación

VacunaBot @uisbot · 8 mar.
La vacuna de Rusia no ha sido aprobada para ser aplicada en Colombia. ¿Se acuerdan de que el virus chino que mató a millones de Colombianos en Colombia, o el uribismo que creó el actual virus?



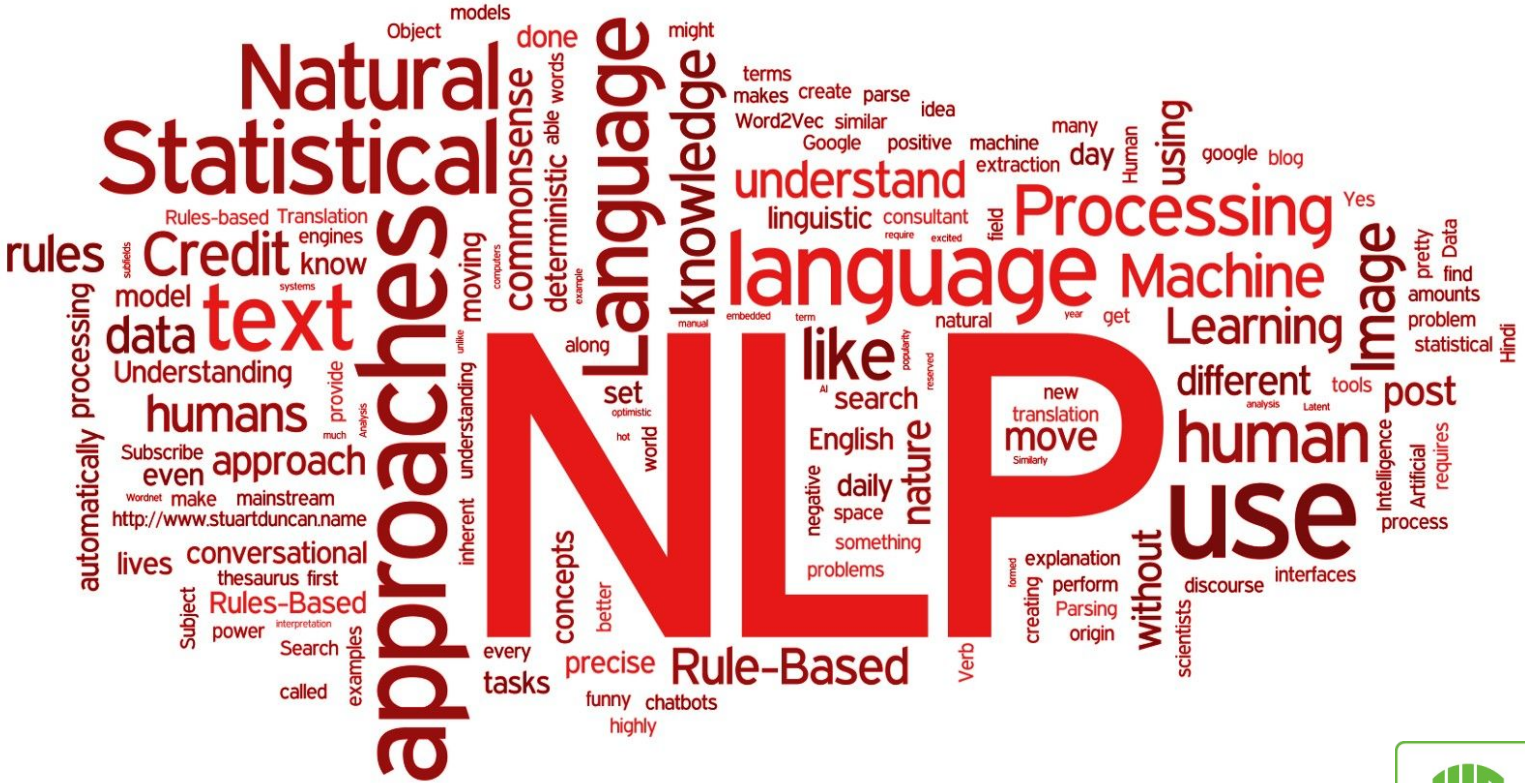
VacunaBot @uisbot · 3h
La vacuna rusa, que es uno de los países con los mayores casos de Covid 19 en América Latina, anuncia gerente general de la farmacéutica estadounidense: "Estamos listos para recibir vacunas a finales del primer semestre del próximo año." También anunciaron que

VacunaBot @uisbot · 3h
El covid esta mal , con todo este manejo policial , ahora con la llegada d la vacuna @MedicaPlop Hoy en Colombia no la tenemos ni el personal de salud adecuado para la vacuna , se están robando las mejores condiciones a la periferia

VacunaBot @uisbot · 3h
En el día de hoy, en #México llegaron 50 mil vacunas Sinovac, de las cuales 12 mil están para los Adultos Mayores y Adultos mayores, en el #CeDemocratico los que viven en la ciudad y otros 1.

Conclusiones

Para más información, escanea nuestro código QR



Universidad Industrial de Santander