

Índice de contenidos

- 1. Procesamiento de lenguaje natural
- 2. Potencial de las técnicas existentes de procesamiento de lenguaje. Limitaciones
 - 1. Potencial de procesamiento del lenguaje natural
 - 1. Reconocimiento del habla (ASR, Automatic Speech Recognition)
 - 2. Síntesis de texto o voz
 - 3. Detección de entidades nombradas (NER, Named Entity Recognition)
 - 4. Traducción automática
 - 5. Similitud de textos
 - 6. Análisis del sentimiento
 - 2. Modelos y técnicas existentes
 - 1. BERT (Bidirectional Encoder Representations from Transformers)
 - 2. Otros modelos
 - 3. Limitaciones. La ambigüedad
- 3. Formación del investigador en PLN.
- 4. Elaboración de un sistema de procesamiento de lenguaje orientado a una tarea específica





El PLN puede emplear también el etiquetado característico del aprendizaje automático supervisado para entrenar un modelo, de tal manera que sea capaz de captar en un texto:

- Positividad
- Negatividad

? Artículo sobre las emociones en la era de la IA

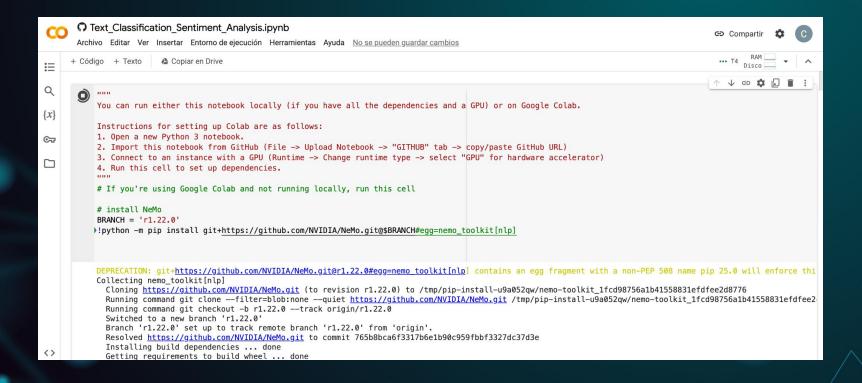
Utilidad: sondeos "invisibles" de la opinión de grandes grupos muestrales alrededor de un tema.

Ej:

redes sociales como Twitter → valoración de un político o de un evento.

Chatbots → en una conversación larga puede averiguar el estado de ánimo del usuario y actuar en consecuencia.





Las distintas versiones de NeMo requieren distintas dependencias, y uuede ser que sea necesario añadir mediante ejecución de la CLI a través del magic command !pip un determinado paquete.

!pip install folium==0.8.3 !pip install imgaug==0.2.5

Nota: se recomienda verificar el nº de etapas de entrenamiento config.trainer.max_epochs = 3

Las críticas positivas son etiquetadas como 1 y las negativas 0 processed_results = postprocessing(preds,{"0": "negative", "1": "positive"})

GLUE Tasks

Name	Download	More Info	Metric
The Corpus of Linguistic Acceptability	<u>*</u>		Matthew's Corr
The Stanford Sentiment Treebank	±		Accuracy
Microsoft Research Paraphrase Corpus	±		F1 / Accuracy
Semantic Textual Similarity Benchmark	<u></u>		Pearson-Spearman Corr
Quora Question Pairs	<u></u>		F1 / Accuracy
MultiNLI Matched	<u></u>		Accuracy
MultiNLI Mismatched	<u></u>		Accuracy
Question NLI	<u></u>		Accuracy
Recognizing Textual Entailment	<u></u>		Accuracy
Winograd NLI	<u>*</u>		Accuracy
Diagnostics Main	<u>*</u>	<u>Z</u>	Matthew's Corr

DOWNLOAD DATA



Para una correcta explicación de los modelos empleados en PLN (en especial BERT) es necesario entender:

- El proceso de **Atención** Redes Neuronales LSTM (Long Short-Term Memory) *
- El modelo de **Transformadores**

Una red neuronal modelada con un proceso de atención que actúa de una manera similar a como lo hace la atención del ser humano → atiende en función de la **relevancia** e **interés** que le suscita.

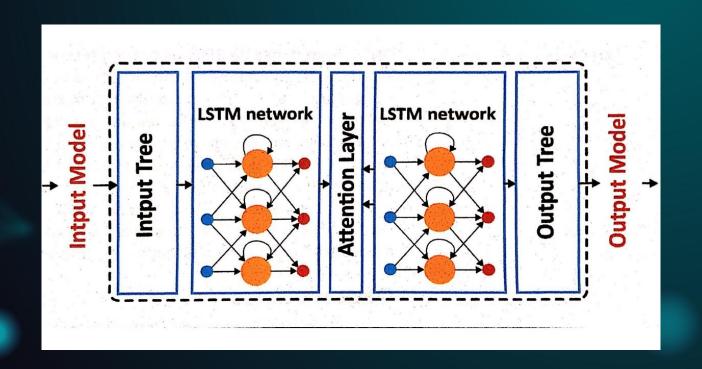
Una red neuronal que lleva a cabo un proceso de atención es capaz de "hacer caso" a algunos valores de sus variables de estado en tiempos pasados, mientras que ignora otras.

Ej: en la traducción de los idiomas el orden de los sintagmas de las frases no siempre sigue la misma colocación, como del alemán al español.

Traductor — Almacena no sólo la palabra previa a la que está actualmente procesando, sino toda la frase.

Una red neuronal que emplea estos mecanismos utilizará:

- Los pesos de la última iteración en el proceso de propagación inversa
- Los pesos de iteraciones anteriores para algunas variables.



Los transformadores usan un proceso:

- 1º. Codificación
- 2°. Decodificación
- 3º. Etapas de atención y autoatención para de entrenar la red



BERT (Bidirectional Encoder Representations from Transformers)

Técnica que incluye transformadores y hacen uso de proceso de atención.

Desarrollado en *2018* por *Google (Jacob Devlin)* → evolución de las RRNN recurrentes (con o sin memoria)

ETAPAS:

- 1°. Aprendizaje no supervisado
- 2ª. Aprendizaje supervisado → afinar el modelo

Una vez que el modelo es preentrenado puede ser compartido \rightarrow reduce sets de entrenamiento

Origen en los modelos de aprendizaje de secuencia semisupervisado:

- ELMO (Microsoft)
- ULMFit

BERT (Bidirectional Encoder Representations from Transformers)

Características:

- Bidireccional → se analizan las palabras ubicadas a la izquierda y a la derecha de cada uno de los términos actuales
- Sin supervisión → no requiere un corpus anotado, que busca mediante los autotransofmadores recrear una salida a partir de una entrada, auto-validándose.
- Tiene en cuenta el *contexto* para cada aparición de una palabra concreta, ayudando a la desambiguación.
 - Ej: "Se ha caído la bolsa de New York" y "La bolsa de la compra está llena"

Roberta (A Robustly Optimized BERT Pretraining Approach) → introducida por Facebook.

- Difiere en la manera de llevar el enmascaramiento en el proceso de atención.
- Cambios en el proceso de afinado de los hiperparámetros, tamaño del lote, valor de la tasa de aprendizaje.
- Entrenamiento más lento.
- Set de datos 10 veces mayor.
- Mejora entre el 2-20%.



ALBERT (A little BERT) → del equipo de Google que incorpora

- Una parametrización factorizada embebida
- Una capa cruzada de compartición de parámetros que supuso una mejora del 70% en coste computacional



ALBERT en Huggingface

DistilBERT (Distilled BERT)

- Lleva a cambio modificaciones en el algoritmo para que a cambio de sacrificar precisión se mejoren los tiempos y disminuyan los requisitos de costo computacional
- Reducción de hasta 4 veces menos que el tiempo de BERT a cambio de perder un 3% de precisión.

XLNET

- Incrementa el tiempo de entrenamiento (x5)
- Obtiene una mejora de entre el 2 y el 15% en la precisión

MEGATRON → desarrollado por nVIDIA

- Modelo con 8.3 millardos de parámetros entrenado en 512 GPU nVidia TESLA V100
- Alcanza el tamaño de 24 veces BERT y 5.6 veces GPT-2



GPT-3 → evolución del GPT original de OpenAl

- No es un modelo Open Source como el de Google, propietario y uso restringido.
- Tamaño colosal frente a BERT
- No es bidireccional, lo que hace que el afinado sea muy costoso.
- Autoregresivo
- Su gran tamaño tanto en el modelo como en los sets de entrenamiento hacen que supere a BERT en muchos campos sin requerir afinamiento.

