Poner el título adecuado

A1
Universidad
Lima, Perú
correo

A2
Universidad
Lima, Perú
correo

***Resumen*—El modelado del lenguaje se ha abordado recientemente utilizando métodos de entrenamiento no supervisados como ELMo y BERT. Sin embargo, sigue siendo un desafío implementar adecuadamente las redes neuronales con dependencias de largo plazo. Además de ello, estos métodos trabajan con cadenas de longitud fija.**

En el presente trabajo se implementaron métodos actuales basados en la arquitectura de los Transformers que intentan resolver estos problemas. Estos modelos fueron evaluados sobre un conjunto reducido de datos.

Índice de Términos— RNN, modelos seq2seq, mecanismos de atención, transformers.

I. Introducción

Una tendencia dada en el ICRL [1] y en el NAACL- HLT [2] celebrados en el 2019, muestra que las RNN presentan un declive en el número de publicaciones realizadas. Dicha tendencia era de esperarse, aunque las RNN son intuitivas para los datos secuenciales tienen como principal incoveniente la parelización en su procesamiento, por lo tanto, no pueden aprovechar el factor más importante que ha impulsado el progreso en la investigación desde 2012: la potencia de cálculo. Los RNN nunca han sido populares en visión computacional y aprendizaje por refuerzo y para NLP, están siendo reemplazados por arquitecturas basadas en la atención.

Los Transformers, presentados en 2017, introdujeron un nuevo enfoque: módulos de atención. Este modelo ha conllevado a una enorme cantidad de variantes que mejoran el desempeño de tareas del NLP, como el Transformer Universal, BERT (google) o el Transformer-XL; la mayoría de modelos de vanguardia que requieren gran cantidad de datos de entrenamiento y días de entrenamiento en hardware haciéndolo costoso. Sin embargo, con el lanzamiento de librerías especializadas en NLP y arquitecturas Transformer, ahora pueden ser utilizados en datos reducidos.

De esta manera, el presente trabajo busca explorar la aplicación de estos modelos sobre el conjunto de datos reducido AG News y Multi30k. En primer lugar se realizará la definición del modelo Transformer. A continuación, se describirá el estado del arte de modelos basados en el Transformer, para diferentes tareas del NLP. Posteriomente, se explica la metodología a utilizar en los experimentos con diferentes arquitecturas. Las secciones siguientes tratarán los resultados experimentales, las conclusiones y discusiones, y los trabajos futuros que se pueden realizar con esta nueva familia de técnicas del NLP.

[1]International Conference on Learning Representations

[2]Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

II. Transformers

El modelo Transformer, propuesto en el artículo Attention Is All You Need [1], se basa en la auto-atención sin el uso de RNN. Como resultado, es altamente paralelo y requiere menos tiempo para entrenamiento, al tiempo que establece resultados de vanguardia en modelamiento de lenguaje y la traducción automática.

El Transformer se basa en una estructura encoder-decoder. La diferencia entre este y cualquier otro modelo es que el Transformer se basa completamente en mecanismos de atención.

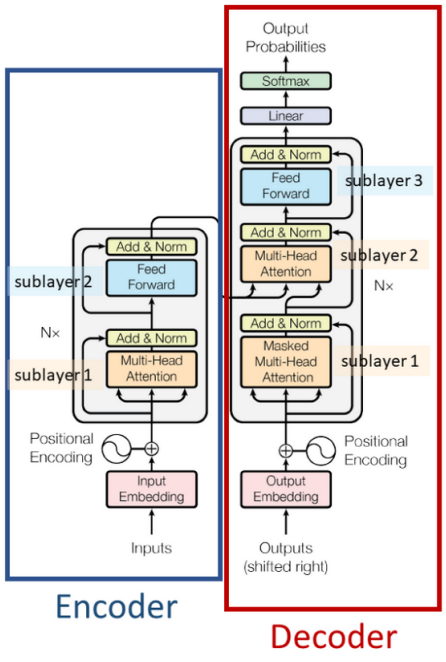

Figura 1. Encoder-Decoder del Transformer. Fuente The Transformer: Attention Is All You Need. [1]

El encoder está formado por una pila de $N = 6$ capas, cada una de las cuales está compuesta por dos subcapas: un mecanismo de atención de múltiples cabeceras y una red de alimentación directa completamente conectada más conexiones residuales [2] en ambas etapas, seguidas de un procedimiento de normalización de capas [3]. El decoder se define de manera similar, solo que cada capa está compuesta de 3 subcapas: atención de múltiples cabeceras, capas completamente conectadas y atención de múltiples cabeceras enmascarada.

Este modelo fue entrenado durante 300000 pasos, aproximadamente 3,5 días, utilizando `8 GPU NVIDIA P100`. De esta manera, el modelo Transformer alcanzó un puntaje BLUE [4] de 26,4 cuando se aplicó sobre el conjunto de datos `newstest2013` como un conjunto de prueba, que estableció un nuevo resultado de vanguardia.

El Transformer es una mejora con respecto a los modelos seq2seq basados en RNN, pero tiene algunas limitaciones.

- La atención solo puede ocuparse de cadenas de texto de longitud fija, lo cual implica que el texto debe dividirse en un cierto número de segmentos antes de ser alimentado al sistema como entrada.
- Esta segmentación del texto provoca la fragmentación del contexto [3]. Por ejemplo, si una oración se divide en dos, se pierde una cantidad significativa de contexto. En otras palabras, el texto se divide sin considerar la oración o cualquier otro límite semántico.

III. Estado del Arte

Con la aparición del Transformer, han surgido diversas variantes y mejoras del modelo para mejorar las limitaciones y el desempeño en diversas tareas del NLP. En esta sección describiremos algunas relacionadas a este trabajo.

III-A. Transformer Universal

El Transformer Universal, propuesto en el artículo del mismo nombre Universal Transformers [5], es una variante del modelo Transformer que tiene como objetivo lograr un buen rendimiento tanto en traducción de lenguaje como en tareas algorítmicas con un solo modelo. Los autores del Transformer Universal señalan que es un modelo completo de Turing.

De acuerdo al artículo, las principales diferencias entre el Transformer y el Transformer Universal consistn en que el Transformer Universal aplica el encoder para un número variable de pasos para cada token de entrada/salida (T pasos), mientras que el Transformador aplica exactamente 6 capas de encoder/decoder, respectivamente. Así mismo, el Transformer Universal utiliza una representación de entrada ligeramente diferente: incluye un embedding de paso de tiempo además de la codificación posicional.

Los números variables de pasos presentes en el Transformer Universal se logran mediante el uso de Adaptive Computation Time, un mecanismo propuesto por Alex Graves [6] que permite la aplicación del encoder y decoder un número variable de veces.

III-B. Transformer-XL

El transformer-XL es uno de los primeros modelos exitosos en abordar el problema de poseer una longitud fija en la secuencia de entrada. En el artículo Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context [7] se propone este método novedoso para el modelado de lenguaje, que permite que una arquitectura transformer aprenda dependencia a largo plazo, a través de un mecanismo de recurrencia, más allá de una longitud fija sin alterar la coherencia temporal.

Se introduce un mecanismo recurrente a nivel de segmento que permite que el modelo reutilice estados ocultos anteriores en el momento del entrenamiento, abordando tanto los problemas del contexto de longitud fija como la fragmentación del contexto. En otras palabras, la información histórica se puede reutilizar y se puede extender tanto como lo permita la memoria de la GPU.

Para reutilizar adecuadamente los estados ocultos, los autores proponen un mecanismo llamado codificaciones posicionales relativas que ayuda a evitar la confusión temporal. Los modelos actuales no pueden distinguir la diferencia posicional entre entradas en diferentes segmentos en diferentes capas. La codificación de posición relativa soluciona este problema al codificar el sesgo de información posicional en los estados ocultos, que difiere de otros enfoques que realizan esto como el nivel de entrada.

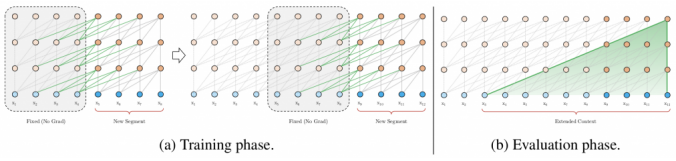

Figura 2. Transformer XL: Entrenamiento y Evaluación. Fuente: Transformer XL [7]

El transformer-XL reduce el puntaje de perplejidad SoTA anterior en varios conjuntos de datos como text8, enwiki8, One Billion Word y WikiText-103. Los autores afirman que el método es más flexible, más rápido durante la evaluación (aceleración de 1874 veces), se generaliza bien en conjuntos de datos pequeños y es eficaz para modelar secuencias cortas y largas.

[3] Context fragmentation

Así mismo, los autores proponen una nueva métrica llamada Relative Effective Context Length que proporciona una manera justa de comparar modelos que se prueban con mayores longitudes de contexto.

III-C. XLNet

XLNet es un modelo presentado en el artículo XLNet: Generalized Autoregressive Pretraining for Language Understanding [8] que es un lenguaje autoregresivo que genera la probabilidad conjunta de una secuencia de tokens basada en la arquitectura del Transformer con recurrencia. Este modelo introduce una variante de modelado de lenguaje llamada modelado de lenguaje de permutación. Los modelos de lenguaje de permutación están entrenados para predecir un token dado el contexto anterior como el modelo de lenguaje tradicional, pero en lugar de predecir los tokens en orden secuencial, predice los tokens en un orden aleatorio.

Además de usar el modelado de lenguaje de permutación, XLNet mejora BERT al usar el Transformer XL como su arquitectura base. XLNet utiliza las dos ideas clave de Transformer XL: embeddings posicionales relativas y el mecanismo de recurrencia. Los estados ocultos del segmento anterior se almacenan en caché y se congelan mientras se realiza el modelado del lenguaje de permutación para el segmento actual. Como todas las palabras del segmento anterior se usan como entrada, no es necesario conocer el orden de permutación del segmento anterior.

III-D. BERT

Este modelo [9],publicado el 2018, esta conformado por una pila de bloques Transformers, la cual esta preentrenada en un corpus de dominio general con 800 millones de palabras. Este modelo obtiene un mejor rendimiento que modelos anteriores a su publicación debido a su naturaleza bidireccional donde la atención se centra en toda la secuencia.

III-E. RoBERTa

Es un modelo introducido en Facebook. El enfoque BERT robustamente optimizado RoBERTa [10], es un reentrenamiento de BERT con una metodología de entrenamiento mejorada, un 1000 % más de datos y potencia de cálculo. Para mejorar el procedimiento de entrenamiento, RoBERTa elimina tareas como Next Sentence Prediction (NSP) del entrenamiento previo de BERT e introduce el enmascaramiento dinámico para que el token enmascarado cambie durante las épocas de entrenamiento.

RoBERTa supera a BERT y XLNet en los resultados de referencia de GLUE.

III-F. DistilBERT

Este modelo aprende una versión destilada (aproximada) de BERT, que retiene el 95 % de rendimiento pero utiliza solo la mitad del número de parámetros. Espec'ificamente, no tiene embeddings de tipo token, pooler y retiene solo la mitad de las capas del BERT de Google.

DistilBERT [11], utiliza una técnica llamada destilation, que se aproxima a BERT de Google. La idea es que una vez que se ha entrenado una gran red neuronal, sus distribuciones de salidas completas se pueden aproximar usando una red más pequeña. Una de las funciones clave de optimización utilizadas para la aproximación posterior en las estadísticas bayesianas es la divergencia de Kulback-Leiber.

IV. Metodología

En esta sección se describen las herramientas y la métodología que se usarán en el presente trabajo

IV-A. Pytorch-Transformers

PyTorch-Transformers [13] es una biblioteca de modelos pre-entrenados de última generación para el procesamiento del lenguaje natural (NLP), que ahora se llama Transformers y es desarrollado por HuggingFace.

De acuerdo al framework [12], Transformers se ha diseñado en torno a una interfaz unificada para todos los modelos: parámetros y configuraciones, tokenización e inferencia de modelos, que son esenciales para construir un pipeline de NLP: definir la arquitectura del modelo, procesar los datos de texto y, finalmente, entrenar al modelo y realizar inferencia.

Esta biblioteca contiene implementaciones de PyTorch, pesos de modelos previamente entrenados, scripts de uso y utilidades de conversión para los siguientes modelos: BERT, GPT, GPT-2 (de OpenAI), Transformer-XL, XLNet, XLM.

IV-B. SimpleTransformers

SimpleTransformers [14] es una libreía construida en base a la biblioteca Pytorch-Trasformers [10], que también contiene modelos de arquitectura Transformer preentrenados pero cuyo objetivo principal es simplificar la codificación y evaluación de los modelos.

IV-C. Apex

n Apex es una extensión de Pytorch con utilidades mantenidas por NVIDIA para optimizar la precisión mixta y el entrenamiento distribuido. Podemos usar con esta librería, precisión de 16 bits para ciertas cosas, pero mantener los pesos en 32 bits y optimizar la memoria a la mitad.

IV-D. Métricas

En el presente trabajo se analizará la exactitud, F1 y el coeficiente de correlación de Matthews como métricas para la tarea de clasificación y la perplejidad como métrica de evaluación para la tarea de traducción automática

IV-D1. Exactitud: La exactitud se define como la razón del total de predicciones correctas sobre el total de predicciones realizadas.

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$$

donde VP=Verdaderos positivos,VN=Verdaderos negativos,FP=Falsos positivos y FN=Falsos negativos.

IV-D2. F1: Esta métrica es la media armónica de la precisión y el recall. Cuando se construye un clasificador siempre hacemos un balance entre el recall y la precisóin y es un poco difícil comparar un modelo con un alto recall y una baja precisión versus un modelo con alta precisión pero bajo recall. F1 es una medida que podemos usar para comparar ambos modelos.

$$F1 = 2 * (Recall * Precisión)/(Recall + Precisión)$$

donde:

$$Precisión = \frac{VP}{VP + FP}$$

y

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

IV-D3. Coeficiente de correlación de Matthews (MCC): MCC es un coeficiente de correlación entre el objetivo y las predicciones y se define como:

$$\begin{aligned} n &= VP + VN + FP + FN \\ \overline{S} &= \frac{FN + VP}{n} \\ \overline{P} &= \frac{FP + VP}{n} \\ MCC &= \frac{VP/n - \overline{SP}}{\sqrt{\overline{SP}}(1 - \overline{S})(1 - \overline{P})} \end{aligned}$$

Generalmente varía entre -1 y $+1$. -1 cuando hay un desacuerdo perfecto entre datos reales y predicción, 1 cuando hay un acuerdo perfecto entre datos reales y predicciones. 0 cuando la predicción puede ser aleatoria con respecto a los datos reales.

IV-D4. Perplejidad: La perplejidad se define como el inverso multiplicativo de la probabilidad asignada al conjunto de prueba por un modelo de lenguaje, normalizada por el número de palabras en el conjunto de prueba.

$$PP(S) = P(w_1, \ldots, w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, \ldots, w_N)}} = \sqrt[N]{\prod_i^N \frac{1}{P(w_i | w_1, \ldots, w_{i-1})}}$$

donde S es una sentencia consistente de N palabras equiprobables.

Si un modelo de lenguaje puede predecir palabras no vistas del conjunto de prueba, es decir, `P(una oración de un conjunto de prueba)` es la más alta; entonces ese modelo de lenguaje es más preciso. Como resultado, mejores modelos de lenguaje tendrán valores de perplejidad más bajos o valores de probabilidad más altos para un conjunto de pruebas.

IV-E. Metodología de trabajo

El método a seguir para implementación básica de arquitecturas relacionados al Transformer, en datos reducidos es el siguiente:

1. Utilización de datos especializados en las tareas a experimentar.
2. Adaptación del modelo TransformerXL para la tarea de traducción y comparación con modelos seq2seq y sus variantes y el transformer.
3. Uso de las bibliotecas SimpleTransformers para la implementación y evaluación de modelos preentrenados en tareas de clasificación.
4. Evaluación y comparación de modelos seq2seq, transformer y transformer-XL, usando la perplejidad para tareas de traducción automática.

V. Experimentación y resultados

La experimentación se ejecutó en Google Colab en un entorno con GPU habilitado y el uso de Apex en tareas de clasificación.

V-A. Conjuntos de Datos

Se emplearon dos datasets reducidos para la evaluación de los modelos:

- AG News [15]: Este dataset fue usado para la tarea de clasificación. Contiene 496,385 artículos de noticias organizados en 4 clases: mundo, deportes, negocios y científicos. El tamaño de la muestra de entrenamiento para cada clase es de 30 000 y de 9 000 para pruebas.
- Multi30k [16]: Este dataset fue usado para la tarea de traducción automática. Extiende el dataset Flickr30K añadiéndole descripciones de las imágenes en alemán. Contiene 31 014 imágenes con descripciones, dichas descripciones se dividen en 145 000 para la muestra de entrenamiento, 5 070 para validación y 5 000 para pruebas.

V-B. Tarea: Clasificación

Usando la librería SimpleTransformers, se evaluaron las arquitecturas de clasificación pre-entrenadas mostradas en el cuadro I . El dataset usado fue AG News.

V-B1. Resultados: En el cuadro II se muestran los resultados comparativos de la tarea de clasificación en base a la medida de exactitud, F1 y el coeficiente de correlación de Matthews.

Modelo	Arquitectura	Parámetros
BERT	12-layer, 768-hidden, 12-heads	110M
XLNet	12-layer, 768-hidden, 12-heads	110M
RoBERTa	12-layer, 768-hidden, 12-head	125M
DistilBERT	6-layer, 768-hidden, 12-heads	66M

Cuadro I
ARQUITECTURAS DE MODELOS DE CLASIFICACIÓN

Modelo	Exactitud	F1	MCC
BERT	93.60 %	93.60 %	91.47 %
XLNet	94.00 %	94.00 %	92.01 %
RoBERTa	94.21 %	94.21 %	92.29 %
DistilBERT	94.36 %	94.37 %	92.49 %

Cuadro II
MÉTRICAS EN MODELOS DE CLASIFICACIÓN

V-C. Tarea: Traducción

Se adaptó, en base al código en [17], el modelo Transformer-XL cuya tarea original es el modelamiento de lenguaje para evaluar su comportamiento en la tarea de traducción automática.

La modificación se realizó para convertir el modelo Transformer-XL original sequence-to-one a un modelo sequence-to-sequence, propio de la tarea de traducción, para que pueda recibir como entrada frases en alemán, y como salida frases en inglés.

Para esta tarea se realizó la visualización de la función de pérdida durante el entrenamiento

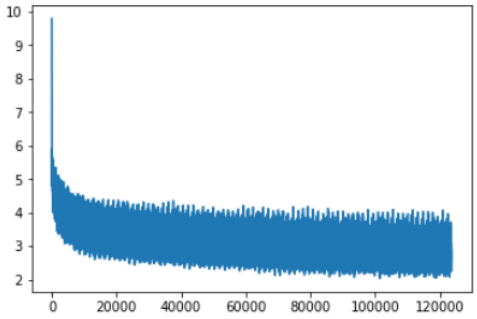

Figura 3. Visualización de la función de pérdida en el entrenamiento

y además el la visualización de la función de pérdida durante la validación

Además se experimentó con 3 modelos adicionales basados en redes neuronales recursivas y el transformer. El detalle de las arquitecturas para esta tarea se pueden ver en el cuadro III. El dataset usado fue Multi30k.

V-C1. Resultados: En el cuadro IV se muestran los resultados comparativos de la tarea de traducción en base a la medida de perplejidad.

La respuesta de los modelos a consultas de traducción se muestran a continuación:

⟨*sos*⟩ mehrere männer mit schutzhelmen bedienen ein antriebsradsystem .⟨*eos*⟩

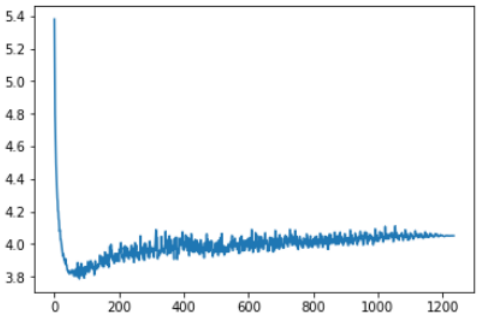

Figura 4. Visualización de la función de pérdida en validación

Modelo	Arquitectura	Parámetros
RNN+LSTM	encoder-decoder	13M
RNN+GRU	encoder-decoder	14M
RNN+GRU+Atención	encoder-decoder(atención)	20M
Transformer	encoder(atención)-decoder(atención)	55M
Transformer-XL	encoder(atención)-decoder(atención)	

Cuadro III
ARQUITECTURAS DE MODELOS DE TRADUCCIÓN

Modelo	Pérdida prueba	Perplejidad
RNN+LSTM	3.889	48.87
RNN+GRU	3.508	33.37
RNN+GRU+Atención	3.175	23.91
Transformer	2.265	9.63
TransformerXL	2.97	18.10

Cuadro IV
PERPLEJIDAD EN MODELOS DE TRADUCCIÓN

⟨*sos*⟩ several men in hard hats are operating a giant pulley system . ⟨*sos*⟩

⟨*sos*⟩zwei junge weiße männer sind im freien in der nähe vieler büsche . ⟨*sos*⟩
⟨*sos*⟩two young , white males are outside near many bushes . ⟨*sos*⟩

VI. DISCUSIÓN DE RESULTADOS

- En cuanto al Transfomer-XL modificado, se observó que en ciertos casos llegó a obtener sentido en cuanto a palabras al realizar la traducción, sin embargo también existen casos en los que no se tiene sentido alguno en la sentencia. Esto pudo deberse a la forma de cargar los datos, pues las sentencias en alemán e inglés no poseen la misma longitud. Esto podría arreglarse implementando una capa de encoder previa a los embeddings iniciales para evitar los problemas en la coherencia de la cantidad de términos. Esto incluiría mejorar la forma en la carga de datos.

VII. CONCLUSIONES

- Se ha realizado clasificación multiclase utilizando la librería simpletransformers y modelos pre-entrenados, basados en BERT, obteniendo buenos resultados en métricas de clasificación, especialmente del modelo distilBERT, que tiene un menor número de paramétros que BERT y que utiliza el aprendizaje `teacher-students`, donde

se entrena a una red de `students` para imitar la distribución de salida completa de la red del `teacher` (su conocimiento) y retiene el rendimiento de BERT.

- En la tarea de traducción, se realizaron comparaciones entre distintos modelos seq2seq y los modelos transformers, donde estos últimos obtuvieron mejores resultados. Sin embargo el modelo Transformer XL modificado no logró superar al Transformer original.

VIII. Trabajos Futuros

- Los modelos Transformers se han vuelto un referente del procesamiento del lenguaje natural, por que lo que se necesita explorar el desempeño de estos modelos preentrenados en distintas tareas, donde los modelos seq2seq han tenido excelentes resultados, como es el caso de respuestas a preguntas o clasificación de tokens.

Referencias

[1] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In

[2] Sergey Ioffe, Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift (2015). Disponible en https://arxiv.org/abs/1502.03167.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E.Hinton. Layer Normalization (2016). Disponible en https://arxiv.org/abs/1607.06450.

[4] Kishore Papineni , Salim Roukos , Todd Ward , Wei-jing Zhu (2002). BLEU: a Method for Automatic Evaluation of Machine Translation, pp. 311-318. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).

[5] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, Łukasz Kaiser. Universal Transformers (2018). ICLR 2019. Disponible en https://arxiv.org/abs/1807.03819

[6] Alex Graves. Adaptive Computation Time for Recurrent Neural Networks (2017). Disponible en https://arxiv.org/abs/1603.08983.

[7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (2019). Disponible en https://arxiv.org/abs/1901.02860.

[8] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding (2019). Disponible en https://arxiv.org/abs/1906.08237.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018). Disponible en https://arxiv.org/abs/1810.04805.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov- RoBERTa: A Robustly Optimized BERT Pretraining Approach (2019). Disponible en https://arxiv.org/abs/1907.11692.

[11] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter (2019). Disponible en https://arxiv.org/abs/1910.01108.

[12] Pytorch-Transform:Implementación de transformer para NLP. https://pytorch.org/hub/huggingface_pytorch-transformers/.

[13] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Jamie Brew. Transformers: State-of-the-art Natural Language Processing (2019), HuggingFace Inc. Disponible en https://arxiv.org/abs/1910.03771.

[14] SimpleTransformers https://pypi.org/project/simpletransformers/.

[15] Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level Convolutional Networks for Text Classification(2015). Disponible en https://arxiv.org/abs/1509.01626.

[16] Desmond Elliott, Stella Frank, Khalil Sima'an, Lucia Specia. Multi30K: Multilingual English-German Image Descriptions(2016). Disponible en https://arxiv.org/abs/1605.00459.

[17] TransformerXL from Scratch notebook.