e-ISSN: 1390-5902
CEDAMAZ Revista del Centro de Estudio y Desarrollo de la Amazonia , Vol. 11, No. 1, pp. 79–86, enero–junio 2021

Revisión sistemática de literatura: características y funcionamiento respecto a los modelos BERT y SQuAD

Systematic literature review: characteristics and functioning of the BERT and SQuAD models.

José Carrión[1,*] y Victor Serrano[1]

[1]*Carrera de Ingeniería en Sistemas/Computación, Universidad Nacional de Loja, Loja, Ecuador*
jose.a.carrion.o@unl.edu.ec, victor.serrano@unl.edu.ec

[*]*Autor para correspondencia: jose.a.carrion.o@unl.edu.ec*

Fecha de recepción del manuscrito: 27/06/2021 Fecha de aceptación del manuscrito: 07/07/2021 Fecha de publicación: 15/07/2021

Resumen—En la actualidad con la pandemia que se padece, se han producido colapsos en el sistema de salud lo que ha ocasionado pérdidas humanas y económicas en mayor parte, ha provocado el resguardo de la población y a limitado el acceso a centros de salud. Lo que ha provocado decesos en la población por no poder tener acceso a atención medica básica, como pueden ser consultas sobre los principales síntomas. La presente Revisión Sistemática de Literatura (RSL) asumió el propósito de identificar qué características y funcionamiento óptimo son necesarios para empleo de BERT y SQuAD con el fin de desarrollar posteriormente un agente virtual centrado en dar Respuesta a Preguntas sobre temas comunes del Covid-19. Ya que al no dar abasto los centros de salud el agente ofrecería una mayor cobertura en temas de asistencia sobre el covid a la población. La presente RSL se basó en las fases de la metodología de Bárbara Kitchenham, la revisión se planteó en base a tres preguntas de investigación y definió el transcurso de la revisión; obteniendo a PyTorch y TensorFlow como frameworks para el desarrollo de software, como lenguaje programación a Python por su vinculación en aprendizaje automático, el modelo BERT BASE empleado para hardware de pocos recursos y SQuAD 2.0 por ser más completo respecto a pares de preguntas y respuestas razonables.

Palabras clave—BERT, SQuAD, Covid, Respuestas a preguntas, Agentes conversacionales.

Abstract—Currently, with the current pandemic, there have been collapses in the health system, which has caused human and economic losses in most cases, has caused the protection of the population and has limited access to health centers. This has caused deaths in the population due to lack of access to basic medical care, such as consultations on the main symptoms. This Systematic Literature Review (SLR) was undertaken to identify what features and optimal performance are necessary for the use of BERT and SQuAD in order to further develop a virtual agent focused on answering questions on common Covid-19 topics. The agent would provide greater coverage of Covid assistance issues to the population, since the health centers are not able to meet the needs of the population. The present RSL was based on the phases of Barbara Kitchenham's methodology, the review was based on three research questions and defined the course of the review; obtaining PyTorch and TensorFlow as frameworks for software development, Python as programming language for its linkage in machine learning, the BERT BASE model used for low-resource hardware and SQuAD 2.0 for being more complete with respect to pairs of questions and reasonable answers.

Keywords—BERT, SQuAD, Covid, Answers to Questions, Conversational agents.

INTRODUCCIÓN

Actualmente la población por temas relacionados al Covid19, no puede acceder a la sanidad pública ni privada, lo que genera miedo e inestabilidad en las familias al no poder consultar a un profesional de la salud cuando se encuentran con síntomas leves. Las condiciones de los centros de salud de solo atender a pacientes graves no suponen un alivio para las personas, si a esto se le suma la desinformación que circula sobre diferentes temas de la pandemia la situación se complica aún más (Ayoub et al., 2021). Surgen formas de ayudar a la gente en los momentos de duda respecto al Covid-19 con la ayuda de la tecnología y en este caso mediante agentes conversacionales. En la presente RSL, mediante la metodología de

Bárbara Kitchenham y empleando sus fases se ha podido conseguir artículos sobre estudios y trabajos relacionados que permitieron identificar la información correspondiente a las características y funcionamiento de los modelos BERT y SQuAD.

El presente estudio se realizó en secciones como: la sección de Metodología donde se definieron las fases a seguir que propone Barbara Kitchenham para realizar revisiones de literatura, posteriormente, se realiza el proceso para cada una de las fases propuestas detallando el proceso y las salidas en la sección de Resultados, en la sección de Discusión para analizar, explicar e interpretar los resultados y finalmente, se plantean las Conclusiones obtenidas durante la realización de la presente RSL.

MATERIALES Y MÉTODOS

La realización de la revisión sistemática de literatura se basa en el proceso de la metodología de Bárbara Kitchenham(Kitchenham Charters, 2007), dicho proceso se resume en tres fases principales:

- Planificación de la revisión
- Realizar de la revisión
- Revisión de informes

De acuerdo a Kitchenham (Kitchenham Charters, 2007), algunas tareas de estas fases no son obligatorias, como por ejemplo:

- Puesta en marcha de una revisión, depende de la revisión sistemática que se está haciendo sobre una base comercial.
- Evaluar el protocolo de revisión y la evaluación del informe, son opcionales y dependen de los procedimientos de garantía de calidad decididas por el encargado de la revisión sistemática

Para la presente RSL no se ejecutaron todos los pasos propuestos por Kitchenham, justamente porque su metodología es flexible en cuanto a la extensión que el investigador necesite dar a su revisión y depende del investigador el alcance que requiera dar a la presentación y publicación de sus resultados. De acuerdo a esto, en la TABLA 1, se presentan las fases y tareas que fueron consideradas:

Tabla 1: Proceso de revisión sistemática de literatura propuesto

Fases	Tareas
Planificación de la revisión	Identificación de la necesidad de una revisión.
	Especificación de las preguntas de investigación.
	El desarrollo de un protocolo de revisión.
Realizar la revisión	Identificación de la investigación. Selección de los estudios primarios. Síntesis de los datos
Revisión de informes	Dar formato al informe principal.

RESULTADOS

Planificación de la revisión

Identificación de la necesidad de una revisión

Mediante la ejecución de una revisión sistemática de literatura, se puede encontrar el conocimiento ya existente acerca de un tema en particular, en este caso es necesaria para optar por las configuraciones y parámetros necesarios para el desarrollo de un agente virtual; así mismo conocer las diferentes versiones existentes de los modelos BERT y SQuAD que se emplearan en su construcción. Por consecuente, para abarcar este tema se propusieron tres preguntas de investigación para obtener información concisa y relevante.

Preguntas de investigación

En la TABLA 2, se presentan las preguntas de investigación que guiaron el desarrollo de la revisión sistemática de literatura, estas preguntas son:

Tabla 2: Preguntas de investigación

	Preguntas de investigación
P1	¿Cuáles son las características principales sobre el funcionamiento del modelo BERT y SQuAD?
P2	¿Cuáles son los parámetros que se emplean en los modelos de BERT con SQuAD?
P3	¿Qué versión de los modelos BERT y SQuAD es factible para el desarrollo del agente virtual?

El desarrollo de un protocolo de revisión

Diseño del protocolo de búsqueda

Estrategias de búsqueda

Petticrew y Roberts (Petticrew Roberts, 2008), recomiendan criterios para plantear la investigación, mediante el uso de la nemotécnica PICOC (Población, Intervención, Comparación, Resultado y Contexto), se estructuran los cinco componentes para definir la cadena de búsqueda; así como también, para una mejor organización y selección de todos los resultados, se utilizó la herramienta en línea Parsifal1, la cual ayuda en el contexto de la Ingeniería del Software para realizar revisiones sistemáticas de literatura, gracias a esta herramienta que permite dar seguimiento a la RSL, la cual está diseñada para abarcar todos los campos necesarios y obtener resultados óptimos.

Fuentes bibliográficas

Como fuentes bibliográficas, se ha seleccionado algunas bibliotecas virtuales, como son:

- ACM Digital Library (https://dl.acm.org/)
- IEEE Digital Library (https://ieeexplore.ieee.org/)
- Scopus (http://www.scopus.com)

Además, se realizó la investigación con ayuda del buscador:

- Google Scholar (https://scholar.google.com/)

Definir palabras claves para el problema de estudio

Con la definición de los criterios PICOC se obtuvo un conjunto de palabras claves, las mismas que permitieron construir las cadenas de búsquedas, estas son: **BERT, Bidirectional Encoder, Representations from Transformer, Configuration, Model trained, Parameters training, SQuAD, Stanford Question Answering Dataset, Transformers parameters**.

Cadenas de búsqueda

En la TABLA 3, se presentan las cadenas de búsqueda aplicadas que se realizaron de acuerdo a cada biblioteca virtual: ACM Digital Library, IEEE Digital Library, Google Scholar, y Scopus.

Criterios de inclusión

Durante la búsqueda, se consideraron los siguientes criterios de inclusión:

- Solo documentos que sean artículos científicos.
- Se mencione BERT.
- Contenga características de BERT y SQuAD
- Idioma: Inglés-Español
- Publicaciones superiores a los últimos 4 años

Se consideraron los siguientes criterios de exclusión

- Documentos que no sean artículos científicos.
- Idiomas que no sean inglés o español.
- Publicaciones mayores a los últimos 4 años.
- Que no mencionen los modelos BERT o SQuAD

Planificación de la revisión

Identificación de la investigación

El objetivo de la presente revisión sistemática de literatura es dar respuesta a las preguntas de investigación planteadas, mediante la búsqueda de estudios primarios que aporten información verídica con respecto al tema. Selección de los estudios primarios.

Para realizar la selección de los estudios se ha seguido el proceso que a continuación se muestra en la FIGURA 1.

En la TABLA 4, se presenta un resumen de los trabajos o estudios relacionados que fueron encontrados junto con el número de estudios seleccionados según la fuente bibliográfica.

Tabla 3: Scripts de Búsqueda

Bibliotecas virtuales	Cadenas de búsqueda
ACM Digital Library	[[Publication Title: bert] OR [Publication Title: squad]] AND [All: performance of machine learning models] AND [Keywords: bert] AND [Abstract: answers to questions]
IEEE Digital Library	"BERT.ᴬND "SQuAD.ªnd çonfiguration.ᴬND "training.ᴬND "parameters.ᴬND "Bidirectional Encoder Representations from Transformers.ᴬND "Stanford Question Answering Dataset.ᴬND performance of machine learning models
Google Scholar	("Título de la publicación": BERT) Y (Resumen": SQuAD) Y (Resumen": configuración) O (Resumen": formación) Y (Resumen": parámetros) Y (Resumen": Representaciones de codificador bidireccional de Transformers) O (Resumen": conjunto de datos de respuesta a preguntas de Stanford) Y (-esumen": rendimiento de los modelos de aprendizaje automático)
Scopus	(TITLE-ABS-KEY (bert AND bidirectional AND encoder AND representations AND from AND transformer) OR TITLE-ABS-KEY (squad) AND TITLE-ABS-KEY (transformer AND models) OR TITLE-ABS-KEY (question AND answering AND systems) OR TITLE-ABSKEY (performance AND of AND the AND bert AND models) AND TITLE-ABS-KEY (performance AND of AND machine AND learning AND models)) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018)) AND (LIMIT-TO (DOCTYPE , .ªr")) AND (LIMIT-TO (LANGUAGE , .Englishˢ"))

Tabla 4: Resumen de estudios seleccionados

Fuente	Encontrados	Seleccionados
ACM Digital Library	4	2
IEEE Digital Library	5	3
Google Scholar	11	9
Scopus	27	18
Total	47	32

Se obtuvieron 32 estudios relacionados, los mismos que luego de pasar por la evaluación de calidad, resultaron en 21

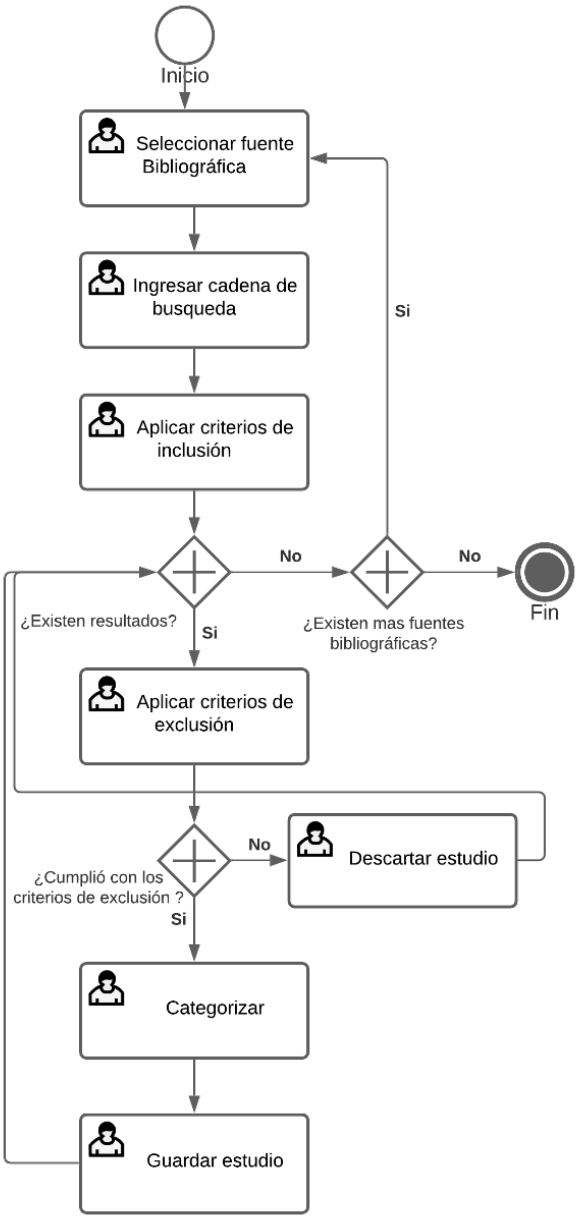

Fig. 1: Proceso de selección de artículos

artículos, los cuales poseían la información necesaria para la realización de la RSL, estos se detallan en la TABLA 5.

Síntesis de los datos

Cada artículo fue analizado para identificar su aporte más relevante y así finalmente obtener información relacionada a los modelos BERT y SQuAD requeridos que contestan a las preguntas de investigación inicialmente planteadas (ver TABLA 2).

¿Cuáles son las características principales sobre el funcionamiento del modelo BERT y SQuAD?

De acuerdo a los estudios (Liu et al., 2019) (Zhou et al., 2020) (Yang et al., 2020) (L. Su et al., 2019b) (Gao et al., 2019) (El-Geish, 2020) (Chang et al., 2020) (Zeng et al., 2020) (Vinod et al., 2020) las características principales sobre el funcionamiento del modelo BERT y SQuAD son:

Para BERT:

Se puede desarrollar en los Frameworks de Tensorflow y PyTorch.

Dependiendo del framework usado se puede programar en los lenguajes Python (PyTorch y Tensorflow) o C++ (Tensorflow).

Consta de un entrenamiento previo y ajuste fino, donde, se siguen dos pasos para aplicar BERT que incluyen preentrenamiento de un modelo BERT utilizando grandes corpus sin etiquetar y afinando el modelo pre entrenado utilizando corpus anotados específicos de la tarea.

Se recomienda ajustar la versión más pequeña, BERT Base, en una tarjeta gráfica con 12 GB de RAM. Donde el entrenamiento toma aproximadamente 2 días y el modelo base reporta un f1 de 65.28 % en el conjunto de datos SQuAD 2.0

El entrenamiento de BERT Large en 16 Cloud TPU toma 4 días.

Para el refinamiento de BERT Large empleando una GPU RTX 2080 Ti es de 1 día aproximadamente.

BERT Large, en el ajuste fino es inestable en conjuntos de datos pequeños. Pero BERT Large logra un aprovechamiento de mayor precisión de tres veces y media más precisión más parámetros.

En base a los artículos obtenidos se ha podido constatar que la mayoría de proyectos emplean como lenguaje Python junto a PyTorch (Bruke Mammo, Praveer Narwelkar, 2018) por su vinculación y puntos fuertes en el desarrollo de este tipo de proyectos (Zeng et al., 2020). Además, emplean BERT Base por los recursos limitados de la GPU y el tiempo que se toma en entrenar al modelo es más corto y la precisión no varía demasiado en comparación con BERT Large (Liu et al., 2019).

¿Cuáles son los parámetros que se emplean en los modelos de BERT con SQuAD?

En base a los artículos seleccionados y estudiados, se constató que los siguientes parámetros son los más empleados como se mencionan en (Liu et al., 2019) (Zhou et al., 2020) (Devlin et al., 2019a) (Maghraoui et al., 2021) (Zadeh et al., 2020)(Bruke Mammo, Praveer Narwelkar, 2018) (Yang et al., 2020) (Özçift et al., 2021) (Chintalapudi et al., 2021) (Balagopalan et al., 2021) (Hulburd, 2020) para trabajos en desarrollo.

BERT Base emplea los siguientes parámetros previamente entrenados que son los siguientes: son L = 12, H = 768, A = 12, Parámetros totales = 110M, donde L es el número de capas (es decir, bloques de Transformers), H es el tamaño oculto y A es el número de capas de atención propia. El tamaño del filtro de alimentación hacia adelante se establece en 4H, es decir, 3072 para H = 768.

BERT toma una secuencia de tokens con una longitud máxima de 512 y produce una representación de la secuencia en un vector de 768 dimensiones, según la documentación base de BERT. Donde se prefiere un tamaño de lote más grande para obtener una estimación suficientemente estable de cuál sería el gradiente del conjunto de datos completo. Donde, el tamaño del lote siempre se establece en potencia

Tabla 5: Estudios seleccionados

N°.	Estudios seleccionados	Ref.	Fuente
ES01	Combat COVID-19 infodemic using explainable naturallanguage processing models	(Ayoub et al., 2021)	Scopus
ES02	Transfer Learning from BERT to Support Insertion of NewConcepts into SNOMED CT	(Liu et al., 2019)	Scopus
ES03	Deep Learning Based Fusion Approach for Hate Speech Detection	(Zhou et al., 2020)	Scopus
ES04	A Two-Stage Transformer-BasedApproach for Variable-Length Abstractive Summarization	(M. H. Su et al., 2020)	IEEE Digital Library
ES05	BERT: Pre-training of deep bidirectional transformers for language understanding	(Devlin et al., 2019b)	Google Scholar
ES06	Performance Analysis of Deep Learning Workloads on a Composable System	(Maghraoui et al., 2021)	Google Scholar
ES07	GOBO: Quantizing attention-basednlp models for low latency and energy efficient inference	(Zadeh et al., 2020)	Google Scholar
ES08	Towards Evaluating the Complexity of Sexual Assault Cases with Machine Learning	(Bruke Mammo, Praveer Narwelkar, 2018)	Google Scholar
ES09	Extracting family history ofpatients from clinical narratives: Exploring an end-to-endsolution with deep learning models	(Yang et al., 2020)	Scopus
ES10	Advancing natural languageprocessing (NLP) applications of morphologically rich languages with bidirectional encoder representations rom t ransformers (BERT):an empirical case study for Turkish	(Özçift et al., 2021)	Scopus
ES11	Sentimental analysisof COVID-19 tweets using deep learning models	(Chintalapudi et al., 2021)	Scopus
ES12	Comparing Pre-trained and Feature-Based Models for Prediction of Alzheimer's Disease Based on Speech	(Balagopalan et al., 2021)	Scopus
ES13	Exploring BERT Parameter Efficiency on the Stanford Question Answering Dataset v2.0	(Hulburd, 2020)	Google Scholar
ES14	Target-dependent sentiment classification with BERT	(Gao et al., 2019)	Scopus
ES15	Gestalt: a Stacking Ensemble for SQuAD2.0	(El-Geish, 2020)	Google Scholar
ES16	Controlling Risk of Web Question Answering	(L. Su et al., 2019a)	Google Scholar
ES17	Know What You Don't Know: Unanswerable Questions for SQuAD	(Rajpurkar et al., 2018)	Google Scholar
ES18	SQuAD: 100,000+ Questions for Machine Comprehension of Text	(Rajpurkar et al., 2016)	Google Scholar
ES19	Generating contextual embeddings for emergency department chief complaints	(Chang et al., 2020)	Scopus
ES20	An ensemble learning strategy for eligibility criteria text classificationfor clinical trial recruitment: Algorithm development and validation	(Zeng et al., 2020)	Scopus
ES21	Fine-tuning the BERTSUMEXT model for Clinical Report Summarization	(Vinod et al., 2020)	IEEE Digital Library

de dos, por ejemplo, 512, 1024, 2048.

Teniendo en cuenta la memoria de la GPU, si se elige BERT Base con el modelo uncase para un ajuste fino se puede evitar el desbordamiento de memoria de la GPU. De acuerdo con la sugerencia oficial de BERT, establece una la longitud máxima de la oración de 64, mini-lote de 32, una la tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3,0 para el entrenamiento del modelo.

En BERT, los datos se pasaron a tres capas de incrustación que incluyen una incrustación de token, segmento y posición capas. En el primer paso del procesamiento, las oraciones se tokenizaron y después de eso, cada token de entrada se pasó a través de una capa de incrustación de tokens para transformarlo en una representación vectorial de dimensión fija (es decir, vector de 768 dimensiones). Además, se agregaron tokens de clasificación adicional [CLS] y separador [SEP] al inicio y al final de la oración tokenizada para servir como una representación de entrada y un separador de oración para la tarea de clasificación. La capa de incrustación de segmentos ayuda a clasificar un texto dado un par de textos de entrada.

Cabe afirmar que cada capa cumple una función de filtrado para lograr encontrar patrones en los datos que contienen datos invisibles a medida que pasa por cada capa, la cual se basa en los patrones de la capa anterior, todo esto empleando los token creados en cada palabra al inicio de la primera capa en la creación de los tokens.

Emplean la biblioteca PyTorch-Pretrained-BERT para construir el modelo BERT. Luego, ajustan su capa lineal (mediante la adición de nuevas palabras junto a la creación de sus respectivos tokens, mediante codificación de los parámetros mencionados en el anterior punto) y la activación sigmoidea (no importa cuál sea la entrada, la salida obtenida está entre valores de 0 y 1) para obtener las predicciones con el conjunto de datos etiquetado COVID-19. Durante el proceso de ajuste fino, se usaron el optimizador Adam con una tasa de aprendizaje de 3x10-6 y un tamaño de lote de 12. Y se ajustó el modelo en el conjunto de datos COVID-19 recopilado durante tres épocas, basado en BERT Base mencionado en el primer punto.

Dichos parámetros se consideraron como factibles para emplearlos en el entrenamiento del modelo BERT. De los estudios que contribuyeron a dar luz a la pregunta de investigación, fue el de Yanling Zhou (Zhou et al., 2020), ya que evalúa el rendimiento de los diferentes modelos como el de ElMo y BERT entre otros, donde el que sigue destacando por encima de los demás es BERT con una precisión del 70% y en el parámetro F1 del 62%, ofrece un margen considerable para poder mejorar con entrenamientos adicionales en modelos pre entrenados y enfocados a un tema en particular, (Ayoub et al., 2021) en el conjunto de (Bruke Mammo, Praveer Narwelkar, 2018) datos SQuAD 2.0. Cabe destacar que para el ajuste de cada modelo se basa en diferentes parámetros según el propósito que se tenga y donde se debe hacer diferentes ajustes para llegar a una configuración optima mediante los entrenamientos.

¿Qué versión de los modelos BERT y SQuAD es factible para el desarrollo del agente virtual?

De acuerdo a los estudios (Liu et al., 2019) (Zhou et al., 2020)(M. H. Su et al., 2020) (Devlin et al., 2019a) (Maghraoui et al., 2021) (Zadeh et al., 2020) (Bruke Mammo, Praveer Narwelkar, 2018)(Yang et al., 2020) (Özçift et al., 2021) (Chintalapudi et al., 2021) (Balagopalan et al., 2021) las principales versiones existentes de BERT son:

BERT Base: Se compone de 12 bloques transformers, 12 capas de atención y 110 millones de parámetros, el cual es más empleado con recursos limitados de hardware.

BERT Large: Se compone de 24 capas, 16 capas de atención y 340 millones de parámetros, es más empleado en grandes equipos de hardware con alta capacidad computacional. Lo que dificulta su aplicación en tiempo real sin aceleración. hardware, como GPU y TPU.

BERT Largue emplea una cantidad de almacenamiento de 1,12 GB de registros, frente a BERT Base que sacrificando precisión en los resultados alcanza los 326 MB destinados en registros. [9]Emplea parámetros de punto flotante 32b.

En cambio, para SQuAD según (Devlin et al., 2019a) (L. Su et al., 2019b) (Rajpurkar et al., 2018) (Rajpurkar et al., 2016) menciona que:

SQuAD 1.0 se trata de un conjunto de datos que contiene respuestas a preguntas que consta de más de 100.000 pares de preguntas, junto al párrafo relevante correspondiente que contiene la respuesta. Pero no es factible ya que para preguntas sin respuesta no tiene definido una respuesta lo que da lugar a malas interpretaciones sobre las respuestas.

SQuAD 2.0: Permite entrenar modelos para que se abstengan de responder preguntas sin respuesta, la tarea de SQuAD 2.0 amplía la definición del problema de SQuAD 1.1 al permitir la posibilidad de que no exista una respuesta corta en el párrafo proporcionado, lo que hace que el problema sea más realista. Por lo que es más factible emplear SQuAD 2.0, ya que, como se menciona es más realista por su funcionabilidad antes preguntas que no tienen respuesta.

Dichas versiones tales como BERT Base y BERT Large (Ayoub et al., 2021) tienen diferentes contrastes dependiendo básicamente del hardware con el que se cuente para el desarrollo de los modelos. Cabe mencionar que, en los artículos encontrados la mayoría da por hecho el uso de modelos pre entrenados, de los cuales solo se va mejorando a través de nuevos entrenamientos y parámetros. Por lo tanto, la versión de BERT base es más factible, ya que no demanda un hardware con altas prestaciones.

SQuAD posee varias versiones, pero las más usadas son SQuAD 1.0 (Rajpurkar et al., 2016) y SQuAD 2.0 (El-Geish, 2020), las que se diferencian básicamente en las preguntas sin respuesta que ofrece la última versión SQuAD 2.0 y se asemeja a la realidad por su función. Por lo que para que sea

más fiel a la realidad, es más factible SQuAD 2.0, ya que combina el conjunto de datos de SQuAD 1.0 con preguntas sin respuesta.

Discusión

Esta RSL tuvo como propósito investigar acerca de los principales parámetros y configuraciones de los modelos BERT y SQuAD para el desarrollo de agentes conversacionales, así como de las diferentes versiones existentes de estos modelos.

De los resultados obtenidos en la mayoría de artículos se mencionan el empleo del lenguaje Python junto a PyTorch (Bruke Mammo, Praveer Narwelkar, 2018) (Zeng et al., 2020). por su vinculación y puntos fuertes en el desarrollo de este tipo de proyectos. Además, emplean BERT Base por los recursos limitados de la GPU y el tiempo que se toma en entrenar al modelo (Liu et al., 2019).

En relación a los parámetros y configuraciones relacionados con el modelo BERT teniendo en cuenta los escasos recursos del hardware los parámetros más factibles serian el uso de tres capas de incrustación junto al modelo BERT base uncase, ya que debe procesar menos datos y establece una la longitud máxima de la oración de 64, mini-lote de 32, una tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3,0 para el entrenamiento del modelo (Zhou et al., 2020).

Por lo cual sería la configuración idónea para empezar a entrenar un modelo junto a BERT y SQuAD 2.0 para un equipo de hardware no tan robusto.

Se han identificado algunas versiones de BERT, tales como BERT Base y BERT Large (Ayoub et al., 2021) las cuales tienen diferentes contrastes dependiendo del hardware con el que se cuente para el desarrollo de los modelos. En la mayoría de artículos encontrados se da por hecho el uso de modelos pre entrenados, de los cuales solo se va mejorando a través del nuevos entrenamientos y parámetros. La versión que se he encontrado como más factible ya que no demanda un hardware con altas prestaciones es BERT Base.

Por parte de SQuAD se han encontrado varias versiones, pero las más usadas son SQuAD 1.0 y SQuAD 2.0 (Rajpurkar et al., 2016) (El-Geish, 2020), las que se diferencian básicamente en las preguntas sin respuesta que ofrece la última versión y se asemeja a la realidad por su función.

Conclusiones

Como características principales sobre el funcionamiento del modelo BERT y SQuAD, se concluyó que se emplea del lenguaje Python junto a framework PyTorch para desarrollar en la nube o con la ayuda de una GPU, empleando además el modelo BERT Base, para tener un mejor desempeño si se posee hardware con recursos limitados, destacando que, se debe realizar un entrenamiento previo y ajuste fino posterior del modelo para su optimización.

Los parámetros más destacados para proyectos relacionados con BERT y que no supongan un procesamiento extremadamente grande teniendo en cuenta los escasos recursos del hardware son el uso de tres capas de incrustación junto al modelo BERT base uncase, ya que debe procesar menos datos y establece una la longitud máxima de la oración de 64, minilote de 32, una la tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3,0 para el entrenamiento del modelo.

Para la elaboración del agente conversacional basado en BERT y SQuAD se concluyó que se usará el modelo de BERT base junto a SQuAD 2.0, con a una GPU Nvidea para entrenar de manera física o en la nube empleando Google Colab, ya que también cuenta con tarjetas gráficas Nvidea. Y los parámetros de acuerdo con la sugerencia oficial de los creadores de BERT, que establecen una la longitud máxima de la oración de 64, un mini-lote de 32, una la tasa de aprendizaje de 2e-5, con un número de épocas de entrenamiento de 3 para el entrenamiento del modelo.

Agradecimientos

A la Universidad Nacional de Loja y todos sus docentes que nos formaron tanto intelectualmente y profesionalmente.

Contribuciones de los Autores

Conceptualización: JC y VS; metodología: JC y VS; análisis formal: JC y VS.; investigación: JC y VS; recursos: JC y VS; curación de datos: JC y VS; redacción — preparación del borrador original: JC y VS; redacción — revisión y edición: JC y VS; visualización: JC y VS; supervisión: JC y VS; administración de proyecto: JC y VS; adquisición de financiamiento para la investigación: JC y VS. Todos los autores han leído y aceptado la versión publicada del manuscrito. José Carrión: JC. Victor Serrano: VS.

Financiamiento

El presente estudio tuvo un financiamiento de procedencia propia por parte de los integrantes.

Referencias

Ayoub, J., Yang, X. J., Zhou, F. (2021). Combat COVID-19 infodemic using explainable natural language processing models. Information Processing and Management, 58(4). https://doi.org/10.1016/j.ipm.2021.102569

Balagopalan, A., Eyre, B., Robin, J., Rudzicz, F., Novikova, J. (2021). Comparing Pre-trained and Feature-Based Models for Prediction of Alzheimer's Disease Based on Speech. Frontiers in Aging Neuroscience, 13. https://doi.org/10.3389/fnagi.2021.635945

Bruke Mammo, Praveer Narwelkar, R. G. (2018). Towards Evaluating the Complexity of Sexual Assault Cases with Machine Learning. 1–25.

Chang, D., Hong, W. S., Taylor, R. A. (2020). Generating contextual embeddings for emergency department chief complaints. JAMIA Open, 3(2), 160–166.

https://doi.org/10.1093/jamiaopen/ooaa022

Chintalapudi, N., Battineni, G., Amenta, F. (2021). Sentimental analysis of COVID-19 tweets using deep learning models. Infectious Disease Reports, 13(2). https://doi.org/10.3390/IDR13020032

Devlin, J., Chang, M. W., Lee, K., Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1, 4171–4186. https://github.com/tensorflow/tensor2tensor

El-Geish, M. (2020). Gestalt: a Stacking Ensemble for SQuAD2.0. http://arxiv.org/abs/2004.07067

Gao, Z., Feng, A., Song, X., Wu, X. (2019). Target-dependent sentiment classification with BERT. IEEE Access, 7, 154290–154299. https://doi.org/10.1109/ACCESS.2019.2946594

Hulburd, E. (2020). Exploring BERT Parameter Efficiency on the Stanford Question Answering Dataset v2.0. http://arxiv.org/abs/2002.10670

Kitchenham, B., Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering.

Liu, H., Perl, Y., Geller, J. (2019). Transfer Learning from BERT to Support Insertion of New Concepts into SNOMED CT. AMIA ... Annual Symposium Proceedings. AMIA Symposium, 2019, 1129–1138.

Maghraoui, K. El, Herger, L. M., Choudary, C., Tran, K., Deshane, T., Hanson, D. (2021). Performance Analysis of Deep Learning Workloads on a Composable System. 1, 1–10. http://arxiv.org/abs/2103.10911

Özçift, A., Akarsu, K., Yumuk, F., Söylemez, C. (2021). Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish. Automatika. https://doi.org/10.1080/00051144.2021.1922150

Petticrew, M., Roberts, H. (2008). Systematic Reviews in the Social Sciences: A Practical Guide. In Systematic Reviews in the Social Sciences: A Practical Guide. Blackwell Publishing Ltd. https://doi.org/10.1002/9780470754887

Rajpurkar, P., Jia, R., Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. ArXiv Preprint ArXiv:1806.03822.

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. (2016). SQuad: 100,000+ questions for machine comprehension of text. EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, 2383–2392. https://doi.org/10.18653/v1/d16-1264

Su, L., Guo, J., Fan, Y., Lan, Y., Cheng, X. Controlling Risk of Web Question Answering. SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 115–124. https://doi.org/10.1145/3331184.3331261

Su, M. H., Wu, C. H., Cheng, H. T. (2020). A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization. IEEE/ACM Transactions on Audio Speech and Language Processing, 28, 2061–2072. https://doi.org/10.1109/TASLP.2020.3006731

Vinod, P., Safar, S., Mathew, D., Venugopal, P., Joly, L. M., George, J. (2020, June 1). Fine-tuning the BERTSUMEXT model for clinical report summarization. 2020 International Conference for Emerging Technology, INCET 2020. https://doi.org/10.1109/INCET49848.2020.9154087

Yang, X., Zhang, H., He, X., Bian, J., Wu, Y. (2020). Extracting family history of patients from clinical narratives: Exploring an end-to-end solution with deep learning models. JMIR Medical Informatics, 8(12). https://doi.org/10.2196/22982

Zadeh, A. H., Edo, I., Awad, O. M., Moshovos, A. (2020). GOBO: Quantizing attention-based nlp models for low latency and energy efficient inference. Proceedings of the Annual International Symposium on Microarchitecture, MICRO, 2020-Octob, 811–824. https://doi.org/10.1109/MICRO50266.2020.00071

Zeng, K., Pan, Z., Xu, Y., Qu, Y. (2020). An ensemble learning strategy for eligibility criteria text classification for clinical trial recruitment: Algorithm development and validation. JMIR Medical Informatics, 8(7). https://doi.org/10.2196/17832

Zhou, Y., Yang, Y., Liu, H., Liu, X., Savage, N. (2020). Deep Learning Based Fusion Approach for Hate Speech Detection. IEEE Access, 8, 128923–128929. https://doi.org/10.1109/ACCESS.2020.3009244