UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE CIENCIAS

SEMINARIO DE TESIS 1:

SISTEMA INTELIGENTE DE PREGUNTAS Y RESPUESTAS UTILIZANDO TECNICAS DE PROCESAMIENTO DE LENGUAJE NATURAL

ELABORADO POR:
JOSE LUIS NAVÍO TORRES

ASESOR:
Mg. HERMINIO PAUCAR CURASMA

LIMA- PERÚ
(2021)

Agradecimientos

Mi mas sincero agradecimiento a toda mi familia, por darme todo el apoyo necesario para mis estudios y la realización de este trabajo.

Ademas, agradecerle también al profesor Herminio Paucar por el apoyo brindado a lo largo de la tesis y sus bien dirigidas recomendaciones.

Por ´ultimo también agradecer a mis compañeros de la facultad, y al grupo de investigación ACECOM- UNI por brindarme importantes pautas para el desarrollo de la tesis.

Abstract

Los avances recientes en el campo del procesamiento del lenguaje natural(PLN) y la intelgencia artificial han mejorado el desempeño de muchas tareas del lenguaje natural, entre ellas la tarea de preguntas y respuestas. Sin embargo, toda esta investigación y desarrollo ha sido realizada para el idioma inglés. El idioma español ocupa el cuarto lugar y lo hablan unos 534 millones de personas en todo el mundo. Sin embargo, a diferencia del inglés, no hay un conjunto de datos de referencia de preguntas y respuestas a gran escala recopilado para el español. En este trabajo usamos modelos de arquitectura de tipo transformers [1] y evaluamos su performance para un dataset traducido al español Squad 1.0-es [2]. Experimentamos con los modelos más recientes y encontramos el modelo que nos brinde la mejor performace en base a métricas EM y F1. Finalmente, desplegamos el modelo en un servidor con una interfaz web para que se puedan realizar consultas de manera remota.

Contents

1 Justificación Científica

La tarea de Preguntas y Respuestas (QA, por sus siglas en inglés) consiste en la creación de sistemas que pueden responder automáticamente a las preguntas planteadas por los humanos en un lenguaje natural. Las tareas de QA basadas en texto se pueden formular como problemas de recuperación de información en los que queremos encontrar los documentos que responden a una determinada pregunta, extraer las posibles respuestas de los documentos y clasificarlas, o como problemas de comprensión de lectura donde la tarea es encontrar la respuesta de un pasaje del texto. Otros tipos de tareas de QA pueden ser: tareas visuales donde el contexto es una imagen, de dominio abierto donde el contexto es un conjunto de documentos o artículos, multimodal donde el contexto puede ser una mezcla de imagen, vídeo, audio y texto. Esta investigación se enfoca en la comprensión de lectura basada en texto. Los modelos de comprensión lectora tienen un gran valor práctico, especialmente para fines de la industria, ya que un modelo de comprensión lectora debidamente capacitado puede funcionar como un chatbot para responder preguntas frecuentes. El motor detrás del progreso de la investigación de sistemas de Preguntas y Respuestas ha sido la creación de múltiples datasets de comprensión de lectura de QA de gran tamaño y calidad, y la publicación de modelos que funcionan bien en estos conjuntos de datos. Sin embargo, para el español, no hay conjuntos de datos de la misma calidad. Para resolver este problema, podemos entrenar modelos desde cero o usar la técnica del aprendizaje por transferencia. Debido a la falta de un gran conjunto de datos específico de QA en los modelos de entrenamiento español desde cero, no es factible, por lo que se considera el uso de aprendizaje por transferencia. El aprendizaje por transferencia se refiere a la adaptación (o finetuning) de un modelo previamente entrenado para un dominio , y especializarlo en otro dominio. En el contexto del lenguaje no supervisado de PLN existen modelos como BERT[3], ELMO [4], ULMFIT [5] entrenados en grandes corpus lingüísticos que se adaptan con éxito a tareas posteriores como clasificación, reconocimiento de entidades nombradas, preguntas y respuestas, traducción automática. Otros experimentos mostraron que en modelos previamente entrenados en corpus multilinguaje como el BERT multilinguaje, el aprendizaje de tipo zero-shot es factible para varias tareas de procesamiento de idiomas que incluyen a QA. El aprendizaje de transferencia zero-shot se refiere al uso de un modelo previamente entrenado en una nueva tarea en un lenguaje (inglés) sin usar ningún ejemplo etiquetado para el lenguaje objetivo (español). La idea es resolver una tarea sin recibir ninguna muestra de esa tarea respectiva durante el entrenamiento. La razón por la que este proceso es eficiente es porque el modelo tiene la capacidad de realizar respuestas extractivas a preguntas, pero no necesariamente en el idioma deseado. La investigación existente sobre el aprendizaje de transferencia entre idiomas en conjuntos de datos de comprensión lectora mostró resultados exitosos utilizando BERT multilinguaje. Se encontró que BERT multilenguaje ajustado en el entrenamiento establecido en el idioma de origen y evaluado en un idioma de destino mostró un rendimiento comparable al de los modelos entrenados desde cero en un solo idioma. Los aportes del trabajo son los siguientes:

- Identificar el modelo que obtenga mejores métricas para la tarea específica de preguntas y respuestas en el idioma español, experimentando con los modelos BERT, BERT para español (BETO)[6], mBERT. También se utiliza otras variantes del modelo BERT como RoBERTa [7] y DistilBERT [8] para comparar las métricas.
- Contribuir con un método de identificar pasajes dentro de un párrafo y responder preguntas específicas en relación con el párrafo dado. Como tal, este modelo busca ahorrar tiempo y beneficiar, entre otras cosas, a niños, a personas con trastornos por déficit de atención e hiperactividad, o incluso a adultos que buscan respuestas específicas de cualquier tipo de literatura que de otra manera podría llevar mucho tiempo para revisar a fondo.

2 Objetivos

2.1 Objetivo Principal

- Investigar el estado del arte para resolver la tarea de Procesamiento de Lenguaje Natural de Preguntas y Respuestas.
- Establecer cuales son los modelos más adecuados para la tarea de Preguntas y Respuestas para el idioma español.

2.2 Objetivos Secundarios

- Aprender cuales son las últimas arquitecturas en los modelos de inteligencia artificial y entender su funcionamiento.
- Aprender cuales son las últimas técnicas que se usan en inteligencia artificial para solucionar los problemas de Procesamiento de Lenguaje Natural.
- Realizar un prototipo de despliegue de los modelos estudiados.

3 Antecedentes

En esta sección, se discute el progreso en las tareas de comprensión lectora en inglés e intentos de otros idiomas para replicar su progreso, destacando específicamente el conjuntos de datos de referencia SQUAD [9] y cómo se han utilizado los conjuntos de datos traducidos en otros idiomas para entrenar modelos de Preguntas y Respuestas.

3.1 Preguntas y Respuestas en Inglés

Desde 2015 han habido múltiples intentos de crear conjuntos de datos de respuestas a preguntas a gran escala para el idioma inglés como son: CNN /Daily Mail Corpus, SQuAD 1.0, RACE, SQuAD 2.0, QuAC, CoQA y Natural Question

Corpus. De estos datasets, SQuAD 1.1 y SQuAD 2.0 son ampliamente usados como puntos de referencia para la tarea de QA. El corpus, las preguntas y las respuestas de los datasets también pueden tener diferentes subtipos. La unidad del corpus puede variar como pasajes únicos o múltiples, documentos, URL, párrafos con imágenes, solo imágenes, videos, etc. Las preguntas pueden ser texto natural o sintética. Las respuestas pueden ser un intervalo extraído del pasaje de contexto, de forma libre (la respuesta puede ser o no del pasaje de contexto) o respuestas de opción múltiple incluyendo sí / no. Los conjuntos de datos también varían en su complejidad, tamaño, fuente y métricas de evaluación. Las métricas de uso común son: Precisión, Recuperación, Exactitud, F1, Exact Match, ROUGE, BLEU, METEOR y HEQ. En esta investigación se trabaja con F1 y Exact Match.

3.2 Preguntas y Respuestas en Español

Existen trabajos en la traducción de datasets de QA del inglés al árabe, coreano, hindi y español donde tradujeron dataset del inglés SQuAD 1.0 a sus respectivos idiomas y se desarrollaron modelos de tipo transformers mediante la técnica de aprendizaje por transferencia para la formación de sus propios sistemas de QA. Recientemente aparecieron otros datasets multilenguaje como XQuAD [10] y MLQA [11].

4 Modelos

En una secuencia típica de procesamiento de lenguaje natural, un texto inicialmente se tokeniza, se codifica en vectores numéricos y luego se alimenta como entrada a los modelos. Para codificar las palabras como vectores numéricos muchas representaciones diferentes se han creado como Bolsa de palabras (BOW por sus siglas en inglés), tf-idf,Word2Vec, GLOVE, FastText, cada una con sus propias limitaciones. En representación de BOW cada palabra se asigna a un vector codificado en unos y ceros, sin embargo, cuando el vocabulario crece este enfoque muestra sus problemas. Las representaciones de tf-idf se han utilizado en los primeros años de diseño de sistemas de recuperación de información , sin embargo, se basa en gran medida en superposiciones de términos entre consultas y el documento, por lo que no permite aprender representaciones más complejas.Word2Vec, GLOVE y FastText se consideran representaciones distribuidas donde la relación semántica entre diferentes palabras se aprende, de estos FastText es capaz de manejar palabras fuera del vocabulario ya que se basa en aprender el nivel del carácter con representaciones de n-gramas, sin embargo FastText no puede manejar la polisemia (una palabra que puede tener diferentes significados posibles según su contexto). Para una tarea compleja como QA contextualizar representaciones de palabras (word embeddings) es muy importante ya que nos permite resolver el problema de polisemia.

Mientras que modelos como BiDAF, DocQA, DrQA muestran un buen desempeño en la clasificación de SquAD 1.1, todos los los modelos actuales de

alto rendimiento en SquAD 2.0 utilizan variaciones de los modelos basados en BERT, que pueden producir representaciones contextualizadas de palabras.

Dado que el aprendizaje por transferencia requiere significativamente menos datos y capacitación en comparación al entrenamiento desde cero, debido a la falta de recursos y datos computacionales se prefirió usar modelos de aprendizaje por transferencia previamente entrenados en un gran corpus multilingüe como Wikipedia de otro idioma así como un modelo entrenado en español. De esta forma se eligió a BETO(BERT entrenado en Corpus de idioma español), BERT, DistilBERT una versión comprimida del BERT original y RoBERTA como modelos para experimentar y crear un prototipo de un sistema de preguntas y respuestas. Los componentes de los modelos y los hiperparámetros son descritos en las siguientes secciones.

4.1 Redes Neuronales Recurrentes (RNN)

Las entradas textuales de las tareas de Procesamiento de Lenguaje Natural son secuencias que deben procesarse a lo largo del tiempo. Las redes neuronales feedforward no son adecuadas para secuencias de texto porque los vectores de entrada y sus correspondientes pesos son de tamaño fijo, que tienden a capturar todos los aspectos relevantes de una muestra a la vez. Las secuencias de proceso de las redes neuronales recurrentes (RNN) capturan una entrada a la vez y la capa oculta del paso de tiempo anterior sirve como un mecanismo de memoria para el contexto. Específicamente, la capa oculta anterior incluye información del contexto desde el principio de la secuencia. Para lograr esto, los RNN tienen un conjunto adicional de pesos que conectan la capa oculta del paso de tiempo anterior, h(t-1), a la capa oculta actual h(t) . Por lo tanto, estas ponderaciones proporcionan al modelo información del contexto pasado al calcular una salida actual. Los RNN también se pueden apilar o entrenar en una secuencia de entrada en orden inverso. El resultado de combinar las redes hacia adelante y hacia atrás es un RNN bidireccional, donde las entradas procesadas de ambas direcciones se concatenan para formar una única representación que captura los contextos izquierdo y derecho de una entrada en cada momento.

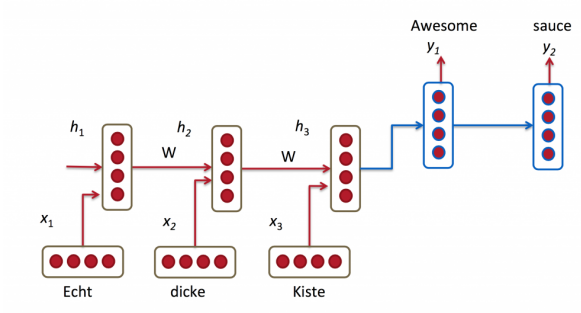

Figure 1: Diagrama de RNN

4.2 RNNs con compuertas y celdas de Memoria(LSTM)

En muchas aplicaciones de lenguaje, la información distante dentro de una secuencia puede ser crítica. Los problemas de desvanecimiento de gradientes ocurren durante la propagación hacia atrás, ya que las capas ocultas son necesarias para calcular multiplicaciones repetidas y donde los gradientes finalmente se llegan a cero. Las redes de memoria de largo alcance de corto plazo (LSTM) [12] se utilizan para mantener el contexto relevante a lo largo del tiempo al entrenar a la red para que olvide (elimine) información que ya no se necesite y recuerde (agregue) información que probablemente se necesitará más adelante. Para lograr esto, se agregan una capa de contexto adicional y tres puertas a la arquitectura. Las puertas son unidades neuronales que operan en la entrada, la capa oculta anterior y las capas de contexto anteriores.

4.3 Modelos Encoder-Decoder

Los RNN se pueden aplicar a tareas de transducción, donde las secuencias de entrada se transforman en secuencias de salida novedosas utilizando modelos de secuencia a secuencia, también conocidos como modelos de codificador-decodificador. Por ejemplo, la traducción automática y el control de calidad son tareas que utilizan modelos de codificador-decodificador. La arquitectura de codificador-decodificador, típicamente implementada con RNN consiste en una red de codificador que pasa el estado oculto final, hn, de un RNN a la red de decodificador. A continuación, el decodificador genera automáticamente una secuencia de salida específica de la tarea, como la respuesta a una pregunta.

4.4 Transformers

Desde la introducción del Transformer, sus variantes se han aplicado a muchas, si no a todas, las tareas del Procesamiento de Lenguaje Natural, logrando el estado del arte. Antes de que se introdujera el Transformer, los modelos RNN eran la estructura más avanzada para el modelado de secuencias y los problemas de transducción. Sin embargo, dada la naturaleza secuencial del cálculo RNN, la paralelización durante el entrenamiento es principalmente limitada, lo que resulta en un entrenamiento menos eficiente. Los autores propusieron una estructura codificador-decodificador, apoyándose completamente en el mecanismo de atención, sin el uso de estructuras recurrentes. El codificador del Transformer consta de 6 capas, cada una compuesta por un mecanismo de autoatención de Multi-Head, seguido de una red de alimentación hacia adelante. Antes y después de cada subcapa, se agregan una conexión residual y una capa add & norm. El decodificador tiene el mismo número de 6 capas. Sin embargo, se inserta una capa de atención de codificador-decodificador entre la capa de atención propia y la de retroalimentación. La máscara de atención se aplica para ocultar tokens de posiciones posteriores. El mecanismo de atención utilizado se denomina Atención Escalada de Producto-Punto, que normaliza los logits para evitar la convergencia lenta debido a un pequeño gradiente de softmax.

Además, el paper original propuso la Atención Multi-Head, de manera que el modelo pueda atender información de diferentes subespacios de representación. Los estados ocultos de todos los subespacios se concatenan y proyectan, antes de la capa de clasificación final.

En las capas de auto atención del codificador, todas las claves, valores y consultas provienen de la capa anterior, lo que permite que el codificador atienda todas las posiciones de la capa anterior. La auto-atención del decodificador, sin embargo, sólo atiende a las posiciones principales debido a la máscara de atención. Por otro lado, para la atención del codificador-decodificador, las consultas se extraen de la capa del decodificador anterior, mientras que las claves y los valores son de los estados ocultos del codificador. Esto permite que el decodificador preste atención a todas las posiciones en la secuencia codificada, lo cual es crucial en las tareas de secuencia a secuencia. Para inyectar información posicional, ya sea relativa o absoluta, se agrega una codificación posicional a las incrustaciones de entrada. En teoría, puede aprenderse o arreglarse; sin embargo, se adopta la codificación posicional sinusoidal para facilitar el aprendizaje y la extrapolación a secuencias más largas.

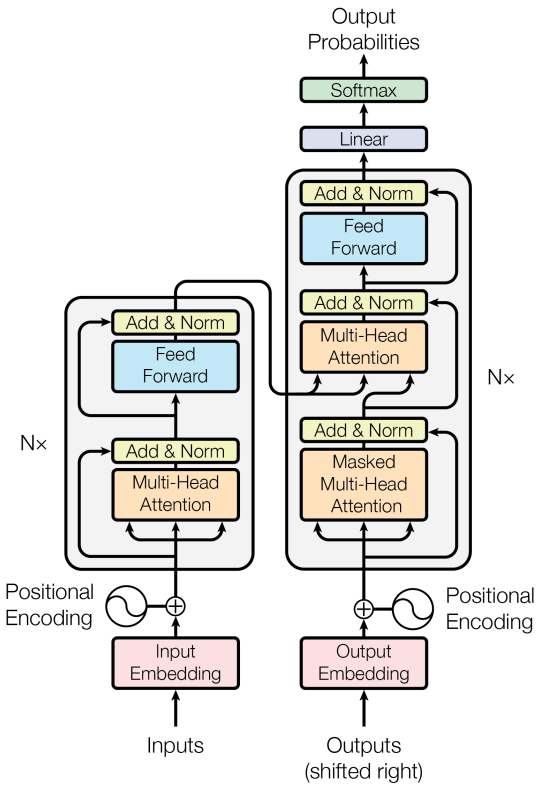

Figure 2: Arquitectura de Transformers

4.5 BERT

BERT se refiere a una técnica de preentrenamiento de redes neuronales (Bidirectional Encoder Representation of Transformers). Por su excelente desempeño en varias tareas de procesamiento de lenguaje natural en GLUE, incluidas SQuAD 1.0 y SQuAD 2.0, ha habido muchas otras arquitecturas que son variantes de BERT. Recientemente se ha incluido a BERT en la tarea de búsqueda que usa Google. BERT se basa en transformers , donde las palabras se analizan en relación con todas las demás palabras en una oración en lugar de mirarlas una por una. Esto permite que BERT observe contextos antes y después de una palabra en particular y le ayuda a captar características de un idioma.

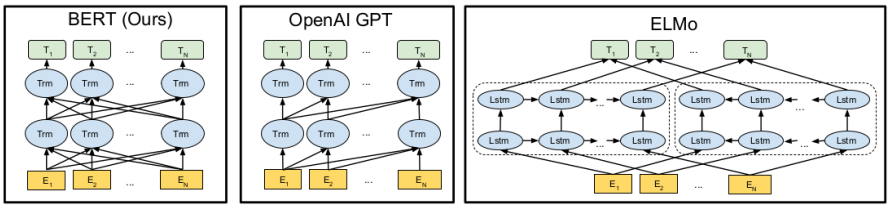

Figure 3: Modelos basados en transformers (BERT and GPT) vs modelo basado en LSTM (ELMo)

4.5.1 Arquitectura de BERT

BERT hace uso de la arquitectura del transformers que utiliza el auto-Mecanismo de atención para aprender las relaciones contextuales entre palabras y subpalabras. En general, la arquitectura del transformer utiliza el método codificador-decodificador. El codificador lee la secuencia de entradas y el decodificador crea predicciones según la tarea. Modelos de secuencia a secuencia, basados en arquitecturas RNN o LSTM leen una secuencia de izquierda a derecha o de derecha a izquierda, pero el codificador del transformador lee toda la secuencia de entrada a la vez. Sin embargo, BERT solo usa la mitad del transformer, el codificador. La arquitectura BERT-base tiene 12 bloques de transformers, 768 unidades ocultas y 12 cabezas de atención,total alrededor de 110 millones de parámetros. La arquitectura BERT-large tiene 24 bloques de transformers, 1024 unidades ocultas y 16 bloques de atención con una cantidad de 340 millones de parámetros. La entrada para BERT es una secuencia de tokens que se asignan a vectores de incrustación (word embeddings) y luego pasan a ser procesados en los bloques de transformers.

4.5.2 Representación de las entradas

La representación de entrada para BERT puede manejar tanto una sola oración como un par de oraciones, como una pregunta y una respuesta, en una única secuencia de tokens. La tokenización de palabras utiliza un vocabulario de 30000

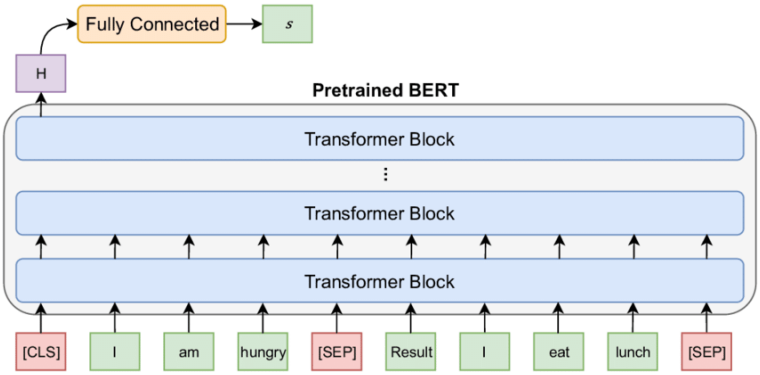

Figure 4: Arquitectura interna de BERT

palabras (del idioma inglés). El primer token de cada secuencia es siempre un token especial llamado token [CLS]. Para tareas de clasificación, el estado oculto asignado al token CLS se utiliza como entrada. Si se dan un par de oraciones como entrada, entonces las oraciones se separan mediante un símbolo separador [SEP]. También se utilizan incrustaciones de segmentos para indicar si una oración pertenece a la oración 1 o 2. La representación de entrada final que ingresa al modelo es la suma de las incrustaciones de tokens, segmentos y posiciones.

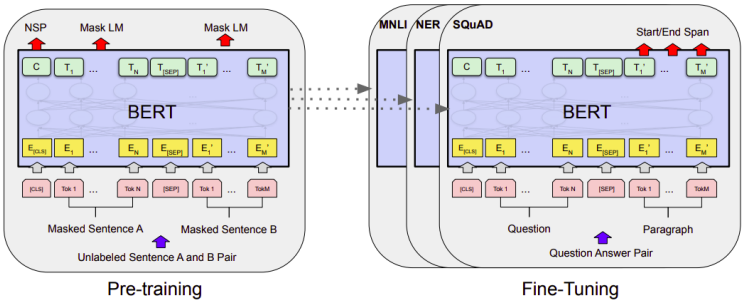

Figure 5: Proceso de Finetuning sobre modelos BERT

4.5.3 Pre entrenamiento de BERT

BERT está preentrenado usando dos tareas no supervisadas llamadas modelado de lenguaje enmascarado y predicción de la siguiente oración.

- Modelado de lenguaje enmascarado: para su entrenamiento se enmascara al azar un porcentaje de los tokens de entrada y luego se predice esos tokens enmascarados. Los vectores de salida final del codificador se alimentan en

una capa softmax sobre el vocabulario para obtener las predicciones de palabras.

- Muchas tareas de QA son basadas en la comprensión de la relación de dos oraciones que no son directamente capturado por modelado de lenguaje enmascarado. Para capturar las relaciones a nivel de oración, se agrega la tarea no supervisada de la predicción de la siguiente oración. Para generar ejemplos de esta tarea en cada ejemplo el 50% de las veces la siguiente oración es la siguiente oración real y la otra mitad del tiempo es una oración aleatoria del corpus.

4.5.4 BERT para preguntas y respuestas

Con la utilización de modelos BERT previamente entrenados, se puede aprovechar el conocimiento aprendido sobre la estructura de las oraciones, la sintaxis, gramática y el lenguaje relacionado con el corpus de millones de oraciones del lenguaje utilizado. El modelo pre-entrenado entonces podría ajustarse con poca información con otras tareas de procesamiento de lenguaje natural como la tarea de preguntas y respuestas, lo que genera mejoras notables en el rendimiento en comparación con otros modelos entrenados desde cero para una tarea específica. El modelo BERT está ligeramente modificado para la tarea de preguntas y respuestas. Dada la pregunta y el contexto de la secuencia total se considera que es una única secuencia empaquetada. A la pregunta se le asigna el segmento 'A' y al contexto se le da el segmento 'B'. Se agregan dos nuevos vectores de la misma dimensionalidad que los vectores de salida,el vector inicial y el vector final. La probabilidad de que una palabra sea el comienzo de la respuesta, el intervalo se calcula como el producto escalar entre la representación de salida de la palabra y el vector de inicio seguido de un softmax sobre el conjunto de palabras del contexto.

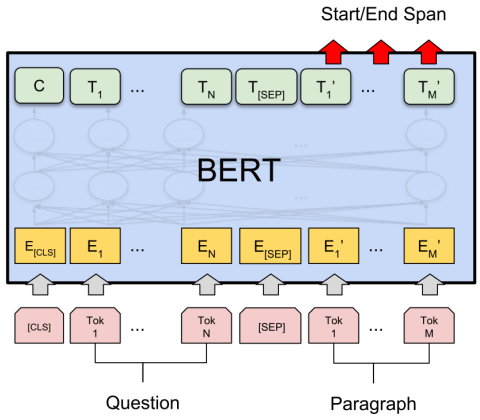

Figure 6: BERT para tarea de Preguntas y Respuestas

4.6 DistilBERT o BERT destilado

DistilBERT es una versión comprimida de BERT donde la técnica de destilación de conocimiento se utilizó para comprimir la arquitectura BERT, reduciendo el tamaño del modelo BERT en un 40% mientras se conserva el 97% de sus capacidades de comprensión del lenguaje y siendo un 60% más rápido. Para el entrenamiento, utiliza una pérdida triple (triplet loss) que combina el modelado del lenguaje, destilación y pérdida de incrustación de coseno. Modelar técnicas de compresión como la destilación de conocimiento está motivada por el deseo de desplegar arquitecturas masivas como los transformers a entornos de bajos recursos informáticos, así como reducir el tiempo de entrenamiento y costo computacional.

4.6.1 Destilación de conocimiento

Destilación del conocimiento [13] es una técnica de compresión donde un modelo compacto más pequeño está entrenado para reproducir el comportamiento de un modelo más grande. El modelo más grande a menudo se llama el maestro y el modelo más pequeño se llama estudiante. En la destilación del conocimiento, la red de estudiantes minimiza una función de pérdida donde el objetivo es la distribución de probabilidades de clases predicho por el maestro. Esta distribución de probabilidad generalmente tiene la clase con una alta probabilidad, mientras que las otras clases tienen una probabilidad cercana a cero. La técnica de destilación de conocimiento se puede pensar en la red de profesores que enseña al estudiante cómo producir resultados como él mismo. Ambas redes reciben la misma entrada. Mientras el objetivo de la red de profesores son las etiquetas reales, la red de estudiantes es recompensada imitando el comportamiento de la red de profesores.

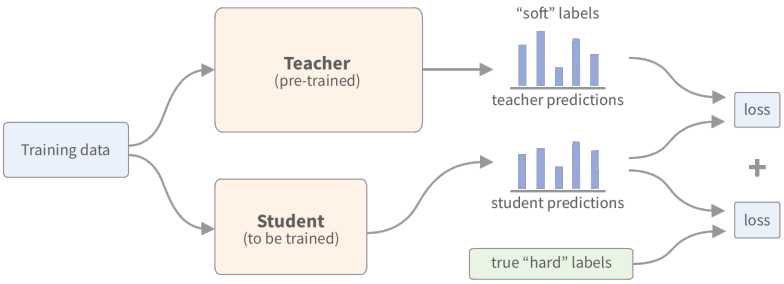

Figure 7: Destilación de Conocimiento

4.6.2 Arquitectura de BERT destilado

Se considera una red de estudiantes (DistilBERT) como una arquitectura de transformer reducida que se entrena con la supervisión de una arquitectura de

transformer maestra más grande (BERT-base). El número de capas se reduce en un factor de 2 y se hacen otras menores modificaciones como eliminar la incrustación de tipo token. DistilBERT también se entrena en el mismo corpus que el modelo BERT original, una concatenación de Wikipedia en inglés con una gran colección de libros de lengua inglesa.

4.7 RoBERTa

Introducido por Facebook, RoBERTa (Robustly Optimized BERT Pretraining Approach), es un reentrenamiento de BERT con varios ajustes para mejorar su rendimiento. Estos incluyen: expandir el tamaño del lote, entrenar el modelo durante más tiempo y sobre más datos. El objetivo de predicción de la siguiente oración (NSP) de BERT es eliminado. RoBERTa también introduce el enmascaramiento dinámico para que los tokens enmascarados se cambiaran durante las épocas de entrenamiento y se utilizaron 160 GB de texto para el pre-entrenamiento.

5 Colección de Conjuntos de Datos

Se usó el dataset SQuAD 1.1 (Stanford Question Answering Dataset), un conjunto de datos de comprensión lectora, compuesto por preguntas anotadas por una selección de artículos de Wikipedia. Para el conjunto de datos de entrenamiento, se tradujo este dataset al español. La respuesta a cada pregunta es una sección del texto (un íntervalo) de la lectura correspondiente.

El conjunto de datos de respuesta a preguntas de Stanford (SquAD) es de comprensión lectora recopilado en inglés con anotaciones de humanos. El conjunto de datos contiene alrededor de 100 mil pares de QA de 442 temas. Para cada tema hay un conjunto de pasajes y para cada pasaje los pares de QA se anotan marcando el tramo o la parte de texto que responde a la pregunta. El dataset SQuAD 2.0 plantea un desafío adicional agregando 50 mil preguntas adversas adicionales al conjunto de datos donde las preguntas no tienen respuesta. Las preguntas contradictorias se introducen para que los sistemas de QA construidos con los conjuntos de datos sean capaces de comprender cuándo no se puede responder una pregunta, así como responder a las posibles preguntas correctamente.

Después de la traducción, se encontró por observación que el contexto, las preguntas y las respuestas fueron traducidas con alta calidad en general y los significados de las oraciones fueron preservados. Sin embargo, la traducción automática tiende a ser consciente del contexto. Palabras idénticas y frases en los pasajes de contexto y las preguntas se pueden traducir diferente basadas en el contexto de las palabras, lo que viola la restricción de que la respuesta de una pregunta debe estar en un intervalo del pasaje para la tarea de comprensión de lectura. También se encontró en muchos casos que los datos estaban corruptos debido a errores de traducción tales como errores de ortografía menores e inserción de caracteres especiales.

6 Métricas de Evaluación

Se usaron las métricas para comparar con la tabla de clasificación de SQuAD en inglés, puntuación de coincidencia exacta(EM) y puntuación F1.

6.1 Coincidencia Exacta (Exact Match)

Esta es una métrica binaria (verdadero / falso) que mide el porcentaje de predicciones que coinciden exactamente con cualquiera de las respuestas. Para una pregunta, si la respuesta predicha y las respuestas verdaderas son exactamente iguales, entonces la puntuación es 1, de lo contrario 0.

$$EM = \sum_{i=1}^{N} F(x_i)/N$$

Donde $F(x_i) = 1$ si la respuesta predicha es igual a la respuesta correcta y $F(x_i) = 0$ en caso contrario

6.2 Puntuación F1

La puntuación F1 es una métrica menos estricta que la coincidencia exacta calculada como la media armónica de precisión y recuperación.

$$F1 = 2 \cdot \frac{Precision \cdot Recuperacion}{Precision + Recuperacion}$$

Si una pregunta tiene múltiples respuestas, la respuesta que da la máxima puntuación F1 es tomado como respuesta verdadera. La puntuación F1 también se promedia sobre todo el conjunto de datos para obtener el puntaje total.

7 Experimentos y Resultados

7.1 Ajuste Fino o Fine Tuning

Para el ajuste fino, se entrenó a BETO(BERT entrenado en corpus en español), BERT multilenguaje, DistilBERT multilenguaje y RoBERTa con el set de entrenamiento sintético traducido al español para 2 épocas con los siguientes hiperparámetros:

- tasa de aprendizaje : 5e-5.La tasa de aprendizaje inicial de Adam.
- max seq length : 384. La longitud total máxima de la secuencia de entrada después de tokenización WordPiece. Las secuencias más largas que esto se truncarán y a las secuencias más cortas se les hará un padding.
- docs stride : 128. Para el overlapping entre el contexto y la pregunta.
- train batch size : 12. Tamaño de lote por GPU / CPU para entrenamiento.

Como los modelos ya adquirieron conocimiento de la estructura y semántica específicas del lenguaje de preentrenamiento, adaptarlos a la tarea de preguntas y respuestas en español requiere un pequeño conjunto de datos para obtener buenos resultados. Hemos mostrado los resultados que obtuvimos de BETO, mBERT, mDistilBERT y RoBERTa en la siguiente tabla:

Modelo	Exact Match(EM)	F1	Peso del Modelo
BETO(BERT español)	57.9978	73.6394	420MB
BERT multilenguaje	57.4358	73.0959	678MB
DistilBERT multilenguaje	51.8764	67.7269	514MB
RoBERTa base	51.0861	67.4771	476MB

Table 1: Resultado experimental de métricas de distintos modelos sobre la tarea de Preguntas y Respuestas aplicando FineTuning.

7.2 Costo computacional

Según las implementaciones oficiales de BERT de Google Research, la construcción de BERT desde cero es computacionalmente costosa, su entrenamiento puede demorar aproximadamente 2 semanas a un costo de aproximadamente $ 500 USD. El entrenamiento DistilBERT requiere 4 veces menos tiempo que BERT, y RoBERTa que es computacionalmente más extenso ya que es entrenado con datos de 160 GB en comparación con BERT(16 GB). En el experimento se utilizo colab, que brindó un GPU NVIDIA Tesla T4 de 16 GB de RAM para afinar los modelos. Los experimentos de ajuste para mBERT y mDistilBERT tomaron más de 3h y para RoBERTa tomó alrededor de 4 horas.

8 Resultados y Discusión

- De la tabla 1 se observa que el modelo de BETO(BERT en español) obtuvo las mejores métricas y con el menor peso del modelo.
- De la tabla 1 también se observa que el modelo BERT multilenguaje quedo bastante cercano, pero con un peso de modelo mayor, lo cual podría dificultar su despliegue en proyectos reales.
- De la tabla 1 también se puede observar, como se esperaba, que el modelo DistilBERT fuera más ligero que su maestro BERT, y que a su vez obtuviera métricas más bajas.
- De la tabla 1 también es interesante observar que el modelo RoBERTa obtuvo métricas aceptables para ser un modelo entrenado con un corpus de inglés puro, además que el peso del modelo no resulta tan elevado.

- Finalmente considerando al modelo que obtuvo mejor performace se realizó un despliegue en el framework de Streamlit, el cual facilita la puesta en marcha de proyectos de inteligencia artificial. En la Figura 8 muestran los resultados del despligue.

Figure 8: Sistema de Preguntas y Respuestas - Despliegue en Streamlit

9 Trabajo Futuro

- El procesamiento de información por máquinas desde la comprensión lectora es una tarea de gran potencial. Debido a que el procesamiento de la información por máquinas con comprensión lectora es de gran potencial para la industria, se ha comenzado estudiando y experimentando con algunos modelos aplicables para el idioma español.
- A futuro se planea experimentar con conjuntos de datos de preguntas y respuestas de opciones múltiples, conjuntos de datos de preguntas y respuestas visuales y con modelos de generación de texto conversacional de tipo chatbot.
- Por otro lado, se planea experimentar con otros modelos más avanzados y nuevos como ALBERT, XLM-ROBERTA, con los cuales se pueda transformar el prototipo desplegado a una interfaz web lista para producción.
- También se planea hacer que el modelo de compresión de lectura se expanda a un sistema de preguntas y respuestas de dominio abierto. Los sistemas de

preguntas y respuestas de dominio abierto recuperan información de una gran colección de documentos no estructurados donde primero se tienen que recuperar los datos relevantes de pasajes o documentos para poder encontrar una respuesta.

10 Conclusiones

- La tarea de Preguntas y Respuestas, formulada como un problema de recuperación de información, es una sección importante del procesamiento de lenguaje natural con el potencial de ser utilizado en numerosas industrias.
- El reciente aumento de modelos de lenguaje potentes como BERT y sus variantes ha hecho posible que todo tipo de tareas de procesamiento del lenguaje natural hayan despertado el interes de los investigadores y esto haya generado un progreso en el área en los últimos años.
- Demostramos que para el lenguaje español se pueden utilizar conjuntos de datos sinteticos traducidos desde conjuntos de datos de alta calidad de lengua inglesa, y obtener un buen rendimiento para el despliegue y producción.

11 Referencias

References

[1] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[2] Casimiro Pio Carrino, Marta R. Costa-jussà, and José A. R. Fonollosa. *Automatic Spanish Translation of the SQuAD Dataset for Multilingual Question Answering*. 2019. arXiv: 1912.05200 [cs.CL].

[3] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[4] Matthew E. Peters et al. *Deep contextualized word representations*. 2018. arXiv: 1802.05365 [cs.CL].

[5] Jeremy Howard and Sebastian Ruder. *Universal Language Model Fine-tuning for Text Classification*. 2018. arXiv: 1801.06146 [cs.CL].

[6] José Cañete et al. "Spanish Pre-Trained BERT Model and Evaluation Data". In: *PML4DC at ICLR 2020*. 2020.

[7] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[8] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *CoRR* abs/1910.01108 (2019). arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

[9] Pranav Rajpurkar et al. *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. 2016. arXiv: 1606.05250 [cs.CL].

[10] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. *On the Cross-lingual Transferability of Monolingual Representations*. 2020. arXiv: 1910.11856 [cs.CL].

[11] Patrick Lewis et al. *MLQA: Evaluating Cross-lingual Extractive Question Answering*. 2020. arXiv: 1910.07475 [cs.CL].

[12] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML].