ALBERTA: Un ensamble para Question Answering

Benjamín Farías, Juan C. Hernández, Benjamín Lepe

Pontificia Universidad Católica de Chile
Email: {bffarias, jchernandez2, balepe}@uc.cl

Uno de los campos principales dentro del procesamiento del lenguaje natural (NLP) es el de Question Answering. En este trabajo revisamos modelos del estado del arte de esta área del aprendizaje de máquina para posteriormente proponer un método de ensamble que logre entregar un buen rendimiento sobre el dataset *SQuAD v2.0*. Este ensamble es creado a partir de dos de los modelos con mayor rendimiento en el *SQuAD Competition* ranking: ALBERT y RoBERTa. En nuestra propuesta, utilizamos un clasificador que, dada una pregunta en particular, es capaz de elegir cuál modelo es el adecuado para responder satisfactoriamente, basándose en la estructura misma de la pregunta. Esta idea surge luego de observar diferencias en las respuestas de los modelos ALBERT y RoBERTa sobre preguntas idénticas. Para el entrenamiento del modelo se usó el dataset *SQuAD v2.0*, al igual que para testear su rendimiento, donde nuestro modelo finalmente obtuvo un 1% más de rendimiento que el modelo ALBERT, pero un 1% menos que el modelo RoBERTa original. Finalmente, encontramos que nuestro ensamble logra un buen rendimiento en estas tareas, a pesar de que no es mejor que RoBERTa para este dataset específico.

Palabras Clave: Deep Learning; NLP; Question Answering; SQuAD; BERT; Ensemble

1. INTRODUCCIÓN

Hasta hace poco, los humanos eran los únicos capaces de leer un texto, comprenderlo y luego responder preguntas relacionadas a este correctamente. Sin embargo, esto ha cambiado considerablemente gracias a los avances en materias de procesamiento del lenguaje natural. Con técnicas de aprendizaje profundo se han logrado entrenar máquinas capaces de recibir un texto y luego responder preguntas sobre el contenido de dicho texto. Esto ha sido posible gracias a la creación de redes neuronales que son entrenadas con cientos de miles de textos, los que traen consigo preguntas y respuestas que pueden o no tener relación con el contexto, como es el caso del Dataset presentado a continuación.

1.1. SQuAD Dataset [1]

Para poder apoyar en el desarrollo de este campo de la inteligencia artificial, la universidad de Stanford creó el dataset **SQuAD (Stanford Question Answering Dataset).** Este dataset contiene una gran cantidad de textos que tratan distintos temas, donde cada uno, además, trae un **set de preguntas junto a sus respectivas respuestas.** Muchos modelos fueron entrenados utilizando este dataset, pero luego se dieron cuenta que hacía falta considerar un tipo de preguntas que estaban ignorando hasta ese entonces: aquellas que son imposibles de responder dado el contexto.

1.2. SQuAD v2.0 Dataset [2]

Debido a lo anterior, se creó el dataset **SQuAD v2.0**, en el cual se incluyen preguntas que no tienen respuesta bajo el contexto dado, y por lo tanto, actúan como distractores. La idea de este dataset es que los modelos logren responder correctamente a preguntas dado un contexto previo (en forma de texto), y que además **se abstenga de responder** cuando la pregunta no tiene respuesta. A partir de esto, los investigadores de Stanford han creado un ranking en el cuál se exponen las puntuaciones obtenidas por distintos modelos, así como la puntuación promedio que consigue un humano sobre el dataset **SQuAD v2.0**. Esto nos permite conocer algunas de las estrategias que han conseguido los mejores rendimientos sobre el dataset hasta la fecha, determinar el estado del arte y estudiar las arquitecturas de estos nuevos modelos.

1.3. Estado del Arte

Al revisar el ranking de **SQuAD competition**, nos damos cuenta que hoy en día existe una gran cantidad de métodos que permiten obtener predicciones exactas con una **precisión mayor al 87%**, lo que supera al rendimiento humano (**EM**= 86.831% [1]). Una gran cantidad de estos métodos son mejoras o ensambles de **BERT (Bidirectional Encoder Representations from Transformers) [3]**, una

técnica desarrollada por **Google AI** para el pre-entrenamiento del procesamiento del lenguaje natural, la cual es capaz de comprender textos como oraciones y no sólo palabra por palabra, lo que permite entender las palabras como parte de un contexto y no como piezas aisladas en un texto en particular.

Entre las mejoras de **BERT** con mayor rendimiento se encuentran los siguientes modelos:

- **ALBERT (A Lite BERT)**[4]**:** Al igual que el algoritmo original, este fue desarrollado por el equipo de **Google AI**. Consiste en una variación *lightweight* de BERT. Esta no sólo ha demostrado contar con un número menor de parámetros que el algoritmo original, sino que además es capaz de obtener mejores resultados en **SQuAD v2.0**.
- **RoBERTa**[5]**:** Fue implementado por Facebook AI. Consiste en una versión de **BERT** entrenada por un mayor tiempo sobre un número más grande de ejemplos y con diferentes hiperparámetros. Específicamente, se incrementó el *learning rate* y se usaron *mini-batches* más grandes. Además, se eliminó el objetivo de la **siguiente oración** dentro del pre-entrenamiento de **BERT**.

1.4. Propuesta

Al probar el rendimiento de **RoBERTa** y **ALBERT** en el dataset de testing de **SQuAD v2.0**, pudimos notar que ambos logran una exactitud similar, pero para ciertas preguntas uno predomina por sobre el otro. De esta forma, nosotros propondremos el ensamble **ALBERTA**, que contiene a ambos modelos y se encarga de decidir cuál aplicar según la pregunta que recibe de entrada.

2. ANTECEDENTES

2.1. Aspectos Teóricos

BERT es una técnica utilizada para la modelación del lenguaje natural y es reconocido en el mundo del Deep Learning por su aporte en las áreas del **Question Answering (SQuAD), General Language Understanding Evaluation (GLUE) y Situations With Adversarial Generations (SWAG).** BERT se basa en un mecanismo de atención llamado Transformer, pero aplicado al procesamiento del lenguaje, el que se encarga de codificar el texto de entrada de forma bidireccional o, más bien dicho, sin dirección, puesto que recibe el texto completo de una vez y no palabra por palabra en una dirección específica. Esto le permite al modelo aprender sobre el contexto de cada palabra, a diferencia de otros embeddings tales como **Word2Vec**[6] y **GloVe**[7].

Existen modelos mejorados a partir de **BERT**, tales como **ALBERT** y **RoBERTa**, los que son *open source* y pueden ser utilizados para realizar fine-tuning o ensambles, entre otros. Por un lado, **ALBERT** realiza una reducción de parámetros, la que permite estabilizar el entrenamiento y ayudar con la generalización. Mientras que **RoBERTa** implementa un embedding que puede recibir textos más largos y también procesar batches de mayor tamaño, lo que da mejores resultados que el modelo original de **BERT**.

2.2. Trabajos Relacionados

Se han escrito varios papers que nos han servido de inspiración para construir a ALBERTA, entre ellos se encuentran los siguientes trabajos realizados a partir de ensambles de modelos derivados de BERT:

Retrospective Reader on ALBERT[8]**:** Luego de que fueran añadidas las preguntas imposibles de responder como objetivo para los modelos de **machine reading comprehension** (**MRC**), fue necesario modificar las arquitecturas del estado del arte agregando un *verifier* al modelo, el cual se encarga de detectar si la pregunta es o no posible de responder. En *Retrospective Reader for Machine Reading Comprehension*, se estudian nuevas formas de realizar estos *verifiers* y se propone un innovador método para la lectura de un texto, el cual se basa en cómo los humanos lo hacemos: primero realiza una lectura rápida con el objetivo de captar las principales interacciones entre la pregunta y respuesta para formar un primer juicio, y luego, relee el texto con el objetivo de verificar la respuesta. La mejor puntuación sobre el dataset SQuAD v2.0 asociada al modelo **Retro-Reader** fue obtenida con el uso de un ensamble que utiliza al modelo pre-entrado **ALBERT** como *encoder* y a **Retro-Reader** como verificador de preguntas imposibles de responder. Con el uso del ensamble anterior, se consiguieron los puntajes EM= 88.1 y F1= 91.4.

Ensemble BERT for classifying medication-mentioning Tweets[9]**:** Este trabajo fue creado con el propósito de detectar mensajes de Twitter que tuvieran menciones sobre medicamentos o información relacionada con la salud. Para lograr esto se realizó un embedding sobre los modelos *Bio+Clinical BERT* y *BERT-Large Uncased*. Los resultados obtenidos arrojaron que la mejor forma de ensamblar estos modelos era calcular el promedio entre los pesos de los clasificadores de ambos modelos, de esta forma se logró una precisión casi un 10% mayor que la de los modelos evaluados individualmente.

3. PROPUESTA Y ESTRATEGIAS

En esta sección serán descritas las decisiones de diseño de nuestra propuesta, conocida como el ensamble **ALBERTA**.

3.1. Arquitectura del Modelo

Este ensamble se basa en los modelos RoBERTa y ALBERT que ya están **pre-entrenados con el dataset SQuAD v2.0**, y se encarga de decidir cuál de los dos modelos es el adecuado para enfrentar a cada pregunta en específico. Esta idea nace al observar cómo en algunos ejemplos de testeo los modelos RoBERTa y ALBERT entregaban distintas respuestas, existiendo casos en los cuales uno de ellos respondía correctamente mientras que el otro fallaba al predecir. Para poder informar al ensamble sobre las características de la pregunta entrante, se utiliza la versión codificada de dicha pregunta utilizando solamente el tokenizer del modelo ALBERT, dado que tanto el input de RoBERTa como el de ALBERT representan una misma pregunta y un mismo contexto. De esta forma, dada una secuencia de tokens, el modelo pueda aprender el tipo de pregunta intrínseco al que corresponde, y así elegir al mejor modelo para abarcarla. Esta decisión es realizada por un clasificador lineal, que corresponde a la parte entrenable de nuestra red, finalmente eligiendo al modelo por medio de una capa de activación **Softmax**[10] que nos entrega una respuesta probabilística sobre cuál es el mejor modelo a utilizar.

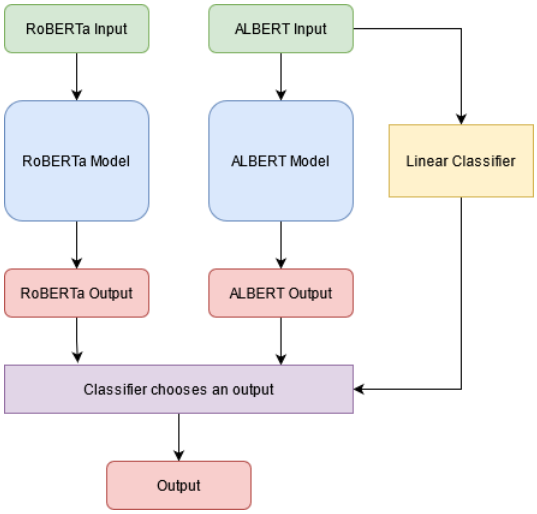

FIGURE 1. Diagrama del ensamble ALBERTA

3.2. Estrategia de Entrenamiento

Dado que nuestro ensamble utiliza modelos pre-entrenados, el entrenamiento consistió en entrenar exclusivamente al clasificador lineal que se encarga de elegir el modelo a utilizar para la predicción. Para lograr esto, realizamos un **entrenamiento de nuestra capa lineal** usando el set de training de SQuAD v2.0, y calculando la *loss* asociada como:

$$\mathrm{L}(x) = \begin{cases} 0 & \text{si } M : \mathrm{L}(x) = \min_x\{\mathrm{L}_A, \mathrm{L}_R\} \\ |\mathrm{L}_A(x) - \mathrm{L}_R(x)| & \text{e.o.c} \end{cases}$$

donde L, L_A, L_R representan las funciones de pérdida del modelo elegido M, ALBERT y RoBERTa respectivamente. La idea es que el ensamble aprenda a seleccionar al modelo adecuado para responder cada pregunta, por lo cual penalizamos cuando elegimos el modelo que tiene asociada la mayor pérdida para una pregunta x en específico, mientras que al elegir el modelo correcto la pérdida es nula.

3.3. Estrategia de Evaluación

Para realizar la evaluación del modelo, se utilizó un dataset de 11873 ejemplos (el dev set de SQuAD v2.0), donde los hiperparámetros utilizados para probar el modelo fueron:

- **n_best:** consiste en el número de predicciones que se deben considerar al revisar la confiabilidad de las respuestas, ordenadas de la mejor predicción a la peor. El valor de este parámetro fue fijado en 10.
- **null_threshold**: umbral usado para decidir si la predicción nos informa o no de una respuesta válida para dicho contexto. El valor del umbral fue fijado en -2.9 y fue elegido testeando varios valores en el intervalo $[-4, 4]$, siendo éste el valor que maximizaba el *score* del modelo sobre el set de testing.

Finalmente, se calcularon las métricas de rendimiento *F1*, que indica qué tan bien se aproximan las respuestas del modelo a las reales, y *EM (Exact Match)*, que indica la proporción de ejemplos que el modelo logró responder perfectamente.

4. RESULTADOS

ALBERTA fue entrenado por 2 épocas en el set de training, obteniendo la evolución de la pérdida acumulada mostrada a continuación:

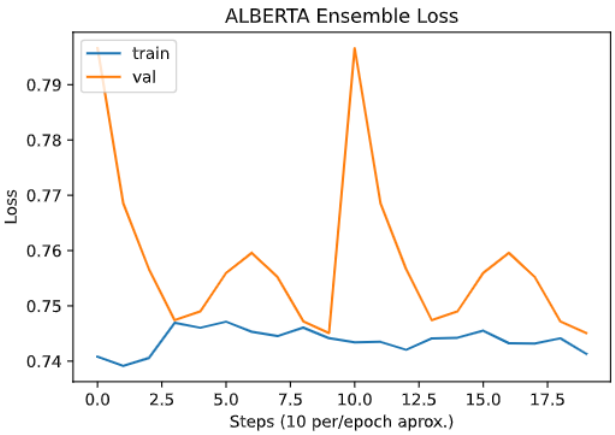

FIGURE 2. Evolución de la pérdida acumulada a través de cada época

En este gráfico se va mostrando la evolución de la pérdida acumulada a través de cada época, es decir, a medida que avanza se va estabilizando el valor, y al llegar a la mitad comienza la segunda época de entrenamiento. Esto explica que en el set de entrenamiento no cambie mucho en la mitad, logrando aprender un poco sobre la selección del modelo correcto en sus ejemplos, mientras que en el set de validación da un salto ya que sus ejemplos no fueron tomados en cuenta para el aprendizaje. Además, se puede argumentar el aumento en la pérdida al comenzar cada época de validación, debido a que esos ejemplos iniciales probablemente eran más complejos para el modelo, al igual que los que aparecen más adelante y causan otro "cerro" en el gráfico.

Con el ensamble ya entrenado, se presenta a continuación una comparación entre el rendimiento de RoBERTa, ALBERT y ALBERTA sobre el set de testing:

Modelo	**EM**	**F1**
Human (Rajpurkar et al. 2018)	86.8	89.45
ALBERT	77.51	80.91
RoBERTa	79.87	82.34
Our Implementation		
ALBERTA	78.67	81.64

TABLE 1: Resultados (%) obtenidos sobre el dataset de testing de SQuAD v2.0.

Los valores mostrados en la *Tabla 1* representan el score obtenido por ambos modelos utilizando los hiperparámetros mencionados en la sección anterior, es por esta razón que estos no coinciden con los números mostrados en el ranking de la página de SQuAD 2.0. Se puede observar que RoBERTa arroja un mejor score que ALBERT, incluyendo ambas categorías (EM y F1). ALBERTA logra vencer a ALBERT, pero no a RoBERTa, de lo que podemos entender que la superioridad previa de RoBERTa en este dataset fue mayor que la mejora lograda con nuestro ensamble de ambos modelos.

5. CONCLUSIONES

El ensamble logró mejorar en comparación con el modelo ALBERT, sin embargo, respecto a RoBERTa el score empeoró, lo que significa que nuestro modelo únicamente obtuvo un promedio aproximado entre ambos modelos y no logró una mejora significativa. Este resultado creemos que se debe a que los modelos son relativamente parecidos, pues ambos son derivados de BERT, lo que conlleva a que no existen muchas preguntas en donde un modelo responda mejor que el otro, al contrario que nuestra hipótesis inicial. Una forma de mejorar esto sería escoger modelos que tengan una mayor diferencia en sus estrategias y diseño, ya que de esta forma es más probable encontrarse con casos en donde un modelo es preferible al otro, y por lo tanto, aumentaría la eficiencia al momento de ensamblarlos con una arquitectura similar a ALBERTA. Por último, como trabajo a futuro sería ideal probar con modelos que rindan mejor y sean más actuales, tales como **ELECTRA**[11] o **Retro Reader**, en donde se podrían obtener mejores resultados. Finalmente, una última posibilidad es la de entrenar el ensamble con más datos y añadiendole más épocas, para lo que será necesario un mejor hardware de procesamiento tensorial.

6. REFERENCIAS

[1] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text.

[2] Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad.

[3] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding.

[4] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations.

[5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach.

[6] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space.

[7] Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

[8] Zhang, Z., Yang, J., and Zhao, H. Retrospective reader for machine reading comprehension.

[9] Dang, H. N., Lee, K., Henry, S., and Uzune, . Ensemble bert for classifying medication-mentioning tweets.

[10] Bridle, J. S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Soulié, F. F. and Hérault, J. (eds.), *Neurocomputing*, Berlin, Heidelberg, pp. 227–236. Springer Berlin Heidelberg.

[11] Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators.