Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Análisis de modelos basados en Transformers para Sistemas de Búsqueda de Respuestas en un dominio académico

Trabajo Fin de Máster

Presentado por: Corrales Solera, Iván

Director/a: Verdú Pérez, Elena

Ciudad: Madrid
Fecha: 02/02/2022

Resumen

Palabras Clave: Sistemas de búsqueda de respuesta, Procesamiento Lenguaje Natural, Transformadores

El uso de los modelos lingüísticos basados en el mecanismo de atención, conocidos como Transformers, y más concretamente las variantes pre-entrenadas a partir de BERT (Bidirectional Encoder Representations from Transformers), se han convertido en el estado del arte para los sistemas de búsqueda de respuesta. El aprendizaje por transferencia permite partir de modelos previamente pre-entrenados, de carácter generalistas, y entrenarlos para su especialización. La aparición de nuevos conjuntos de datos en español, han favorecido la proliferación de nuevos modelos que, apoyándose en modelos ya pre-entrenados, proporcionan un gran rendimiento para la realización de tareas de Procesamiento de Lenguaje Natural en español. Se establece que el entrenamiento de un modelo a partir de un dataset especializado, creado con el temario de la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento, mejoraría los resultados obtenidos.

Abstract

Keywords: Question Answering systems, Natural Language Processing, Transformers

The linguistic models based on the attention mechanism, known as Transformers, and more concretely the pre-trained variants from BERT (Bidirectional Encoder Representations from Transformers) have become the state of the art for Question and Answering Systems. Transfer Learning is used to address specific tasks starting from generalist pre-trained models. The emergence of new Spanish datasets has favored the growth of new models, that relying on pre-trained models, provide great performance for carrying out Natural Language Processing tasks in Spanish. It is established that training a model with a specific dataset, created with questions and answers extracted from the syllabus of the Artificial Intelligence and Knowledge Engineering subject would improve the results obtained by the models in this study.

Índice de contenidos

Índice de tablas

Índice de figuras

1. Introducción

A continuación, se presentan tanto la motivación que me ha llevado a realizar este trabajo de investigación, así como los objetivos que se pretenden alcanzar durante el desarrollo de este. A parte de esto, en este capítulo se muestra el planteamiento del trabajo, así como la organización de la propia memoria.

1.1. Motivación

Desde la producción en cadena de ordenadores, en la década de 1970, el uso principal de estos, tanto para uso personal como comercial, ha sido el de obtener respuestas a nuestras preguntas. Esto se engloba dentro de lo que conocemos como sistemas de información. Los sistemas de información sufren una gran revolución en la primera década del siglo XXI, cuando el uso de Internet en los hogares crece vertiginosamente hasta superar los 4700 millones de usuarios a principios de abril de 2021, según el informe publicado por Digital 2021[1]. Este número de usuarios supone más del 60% de la población mundial.

Los Sistemas de Búsqueda de Respuestas (SBR) – del inglés Question Answering systems (QA), también conocidos como sistemas pregunta-respuesta (PR), adquieren el conocimiento de conjuntos de datos disponibles para posteriormente ser capaces responder a preguntas realizadas. El desarrollo de los SBR aporta gran valor a los sistemas de información debido a que cubren una tarea de alta complejidad y con un elevado coste para ser realizada manualmente por personas.

Es importante destacar que la mayor ventaja de estos sistemas de consulta de la información radica en que dan respuestas en lenguaje natural a preguntas realizadas también en lenguaje natural. Esto es una gran ventaja, frente a otros sistemas que requieren del uso de habilidades de computación para ser capaces de obtener la información. Esto ocurre por ejemplo con el uso de SQL[2] (lenguaje de consultas estructuradas - del inglés Structured Query Language (SQL) que requiere conocimientos de un lenguaje de computación para ser capaces de obtener información de sistemas de base de datos relacionales.

Los SBR son de gran utilidad dentro del campo de la Inteligencia Artificial (IA) puesto que la mayoría de los problemas relacionados con el aprendizaje profundo, del inglés Deep Learning (DL), se pueden modelar como un problema de respuesta a preguntas. Como consecuencia de esto, este campo es uno de los más investigados en informática en la actualidad. En los últimos años se han producido avances y mejoras considerables en el estado del arte, gran parte de los cuales se pueden atribuir a la

[1] Informe Digital 2021, https://www.slideshare.net/DataReportal/digital-2021-july-global-statshot-report-v02
[2] SQL, Structured Query Language, https://es.wikipedia.org/wiki/SQL

llegada del DL y más concretamente en las técnicas de Procesamiento del Lenguaje Natural (PLN) - del inglés Natural Language Processing.

El objetivo de este trabajo surge de la necesidad de cubrir un requisito planteado en un proyecto de investigación de la Universidad Internacional en la Rioja (UNIR[3]). Este requisito consiste en ser capaz de extraer automáticamente, de los temarios de las asignaturas, la respuesta a una pregunta. Por lo tanto, trabajaremos con un dominio cerrado, que viene determinado por el temario en castellano de asignaturas de UNIR, como es la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento del Grado en Ingeniería Informática.

En la actualidad son numerosas las alternativas que nos permiten construir un SBR que resuelva la problemática presentada en el párrafo anterior. Como consecuencia de esto, la motivación de este trabajo es identificar la estrategia de IA que nos ayude a construir la solución óptima a partir de un estudio de investigación de diferentes técnicas existentes.

1.2. Planteamiento del trabajo

Tal y como mencionamos en el apartado anterior, el objetivo de este trabajo es realizar un estudio comparativo entre diferentes técnicas de DL y PLN para la construcción de SBR en un dominio cerrado.

Además, es importante mencionar que el buen funcionamiento de un sistema de QA demanda de grandes cantidades de texto para ser entrenados. Es probable que no contemos con las cantidades de datos necesarias para realizar un estudio por lo que para este estudio se contempla el uso de modelos ya pre-entrenados.

El problema de hacer un SBR completamente funcional ha sido bastante popular entre los investigadores del área de la IA. A pesar de que los nuevos algoritmos basados en la utilización de DL han logrado grandes avances en este campo, existe mucho margen de mejora posible.

De acuerdo con Zhang et al (2020) en su publicación “A novel bidirectional LSTM and attention mechanism based neural network for answer selection in community question answering” se ha demostrado que los modelos de aprendizaje profundo tienen grandes ventajas en las tareas de selección de respuestas. Concretamente los modelos que emplean redes neuronales recurrentes, del inglés Recurrent Neural Networks (RNN[4]) codificador-decodificador, son los más eficaces para

[3] UNIR, Universidad Internacional de la Rioja, https://www.unir.net/
[4] P. J. Angeline, G. M. Saunders and J. B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," in *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 54-65, Jan. 1994, doi: 10.1109/72.265960.

resolver este tipo de problemas. No obstante, las RNN tienen algunas limitaciones como son la representación de datos de alta dimensión en el PLN y las ponderaciones sesgadas para siguientes palabras. Estas limitaciones son apreciables en el trabajo en modelos tradicionales de series de tiempo. Para solventar estos problemas tenemos que hacer uso las redes de memoria de largo y de corto plazo, conocidas en inglés como Long Short-Term Memory (LSTM[5]) y a las arquitecturas basadas en mecanismos de atención.

1.3. Estructura de la memoria

El trabajo que hemos presentado en los apartados anteriores será desarrollado a lo largo de los siguientes capítulos.

En el capítulo 2 se describe el contexto y el estado del arte. Este capítulo nos dará una perspectiva de cuáles son las diferentes soluciones proporcionadas por la IA que utilizaremos para desarrollar nuestra comparativa.

A continuación, en el capítulo 3, se describirán tanto los objetivos como las metodologías de trabajo a utilizar. La metodología que sigamos será fundamental para asegurar que se abordan todos los puntos requeridos.

En el capítulo 4 nos centramos en el planteamiento de la propia comparativa. En este capítulo no solo cubriremos el trabajo previo realizado para identificar el problema concreto a tratar, sino también las posibles soluciones alternativas que se van a evaluar.

En el capítulo 5 desarrollaremos todo detalle la comparativa realizada, con todos los resultados y mediciones obtenidos. Acompañaremos todos estos datos con gráficas, tablas y otros instrumentos donde se plasmarán los datos obtenidos.

En el capítulo 6 se abordará la discusión sobre el significado de los resultados obtenidos en el capítulo 5, así como el análisis de las ventajas y desventajas de las distintas soluciones evaluadas.

En el capítulo 7 se muestran las conclusiones obtenidas tras el trabajo de investigación, desarrollo y evaluación realizados a lo largo del TFM. Además, este capítulo servirá para describir los trabajos futuros a realizar así como la posibilidad de nuevas líneas de trabajo a desarrollar.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

Finalmente y no por ello menos importante, en el capítulo 8 se recogerá la bibliografía que ha servido para el desarrollo de este TFM.

La sección de anexos contiene todo el material complementario que se ha generado para lograr el éxito de este TFM. Adicionalmente se incluye el artículo de investigación correspondiente al trabajo realizado.

2. Contexto y estado del arte

2.1. Sistemas de Búsqueda de Respuestas

2.1.2. Historia de los Sistemas de Búsqueda de Respuestas

Es a comienzos la década de 1960 cuando se comienzan a desarrollar los primeros SBR. La publicación del trabajo que habla sobre el programa Baseball (Green, Wolf, et al, 1961) supone el punto de partida en el desarrollo de futuros sistemas. Baseball trabaja con un dominio cerrado (preguntas relacionadas con el deporte de este mismo nombre), y es capaz de responder a preguntas en inglés sobre datos almacenados. El sistema lee las preguntas de tarjetas perforadas e imprime las respuestas. Las preguntas que es capaz de responder son del estilo "¿Dónde jugó cada equipo el 7 de julio?".

En la década de 1970 se publica el trabajo sobre LUNAR[6] (Woods et al., 1972). LUNAR es un sistema capaz de responder hasta un 90% de las preguntas planteadas en una convención científica lunar en el año 1971. Durante los siguientes años se siguió trabajando en el desarrollo de SBR de dominio cerrado. Estos sistemas adquieren el conocimiento a partir de una base de datos central o un sistema de conocimiento escrito a mano por expertos. En el año 1975 se presenta el proyecto PHLIQA 1 en la publicación "PHLIQA 1: Multilevel Semantics in Question Answering". A diferencia de otros sistemas existentes, que trasladan directamente una estructura sintáctica profunda en un programa a ejecutar, PHLIQA 1 conduce una pregunta a través de varias etapas intermedias de análisis semántico. En cada etapa, la pregunta se representa como una expresión de un lenguaje formal.

En la década de 1980, el mundo de la computación tiene un gran avance debido a que se comienza a trabajar en teorías lingüísticas que permiten el desarrollo de proyectos más ambiciosos en el área de la comprensión de textos y SBR. Los ejemplos más remarcables en este contexto son Unix Consultant (UC) y LILOG. UC fue desarrollado por Robert Wilensky[7] en U.C. Berkeley a finales de esta década. Este sistema era capaz de responder a preguntas relacionadas con el sistema operativo Unix. Como ocurría con los sistemas desarrollados anteriormente, obtiene el conocimiento a partir de una base de conocimiento con un dominio cerrado. LILOG es un sistema capaz de responder preguntas sobre información turística de ciudades alemanas.

[6] https://www.gabormelli.com/RKB/LUNAR_System
[7] https://en.wikipedia.org/wiki/Robert_Wilensky

En la década de 1990 es necesario mencionar la publicación de START[8] (Katz, 1997; Katz 1990). START es un sistema de acceso público de acceso a la información que ha estado disponible para su uso en Internet desde 1993. START responde preguntas de lenguaje natural presentando componentes de texto e información multimedia extraída de un conjunto de recursos de información que se alojan localmente o se accede a ellos de forma remota a través de Internet. A diferencia de otros sistemas anteriores, START no requiere que la información se encuentre estructurada, sino que también es capaz de trabajar con información semiestructurada y no estructurada. La capacidad de START para responder a las preguntas se deriva de su uso de anotaciones en lenguaje natural como un mecanismo por el cual las preguntas se relacionan con las respuestas de los candidatos. En el año 1998 Thompson & Mooney presentan el Sistema WOLFIE (Word Learning From Interpreted Examples) en su publicación "Automatic Construction of Semantic Lexicons for Learning Natural Language Interfaces". WOLFIE es parte de un sistema integrado que aprende a analizar oraciones en representaciones semánticas, como consultas de bases de datos lógicas. Es en esta época cuando comienzan a tener lugar unos talleres conocidos como TREC[9] (Text REtrieval Conference) El objetivo de estos talleres, de carácter anual, es avanzar en el estado del arte en la evaluación de metodologías de recuperación de texto. En el año 1999 tiene lugar TREC8 (Octavo taller de TREC) en el que se la primera evaluación a gran escala de sistemas de pregunta-respuesta de dominio

En la primera década del siglo XXI ocurre un crecimiento mundial en el uso de Internet y como consecuencia en la búsqueda de información online. Tal y como nos indican Olvera-Lobo y Gutiérrez-Artacho (2011) en la publicación de su estudio "Evaluación del Rendimiento de los sistemas de búsqueda" (2011), en demasiadas ocasiones, al plantear una determinada consulta en las herramientas de búsqueda de información web (buscadores, directorios o meta buscadores) el número de páginas web recuperadas resulta excesivo y no todas ellas son relevantes ni útiles para los objetivos del usuario. Es en esta época cuando los SBR se convierten realmente en una alternativa a los tradicionales sistemas de recuperación de información tratando de ofrecer respuestas precisas y comprensibles a preguntas factuales. En este estudio se evalúa la eficacia de cuatro sistemas QA disponibles en la Web —QuALiM, SEMOTE, START, y TrueKnowledge. Se observa que START y TrueKnowledge presentan un nivel aceptable de respuestas correctas, precisas y en una secuencia bien ordenada. Los resultados obtenidos revelan el potencial de esta clase de herramientas en el ámbito del acceso y la recuperación de información de dominio general.

En la década de 2010 ocurren grandes avances en el desarrollo de las tecnologías de aprendizaje profundo, convirtiéndose las RNN en la arquitectura de facto para la construcción de SBR. Además,

[8] http://start.csail.mit.edu/start-system.php
[9] https://trec.nist.gov/

las investigaciones en el uso de modelos de LSTM y arquitecturas basadas en mecanismos de atención han contribuido a la mejora de los resultados obtenidos en la construcción de estos sistemas.

2.1.2. Clasificación de Sistemas de Búsqueda de Respuestas

A lo largo de la historia han sido numerosas las clasificaciones de SBR realizadas, presentando estos sistemas desde diferentes puntos de vista y clasificándolos en función de determinados criterios. Por ejemplo, Jurafsky[10] et al. (2015) clasificaron los SBR según su fuente de datos, Pundge et al. (2016) clasifican los SBR en función del dominio y las técnicas utilizadas para su construcción. Mishra et. al. también en el año 2015 realizan una clasificación más exhaustiva e identifican los siguientes ocho criterios para la categorización de los SBR: dominio, tipo de preguntas, análisis de consultas, técnicas para recuperar la información, características de las bases de datos, tipos de funciones de coincidencia, bases de datos y forma de las respuestas generadas. Por lo tanto y fijándonos en que no existe un acuerdo común a la hora de clasificar los SBR se proponen los siguientes:

2.1.2.1. Clasificación basada en dominio

Debemos entender el domino de un SBR como el área o áreas de conocimiento sobre el que el propio sistema es capaz de aprender. Siguiendo esta clasificación podemos identificar los siguientes dos tipos de dominio: dominio cerrado y dominio abierto.

Dominio abierto

Pertenecen a esta categoría aquellos SBR que son capaces de responder a preguntas de cualquier área. Algunos de los sistemas más destacables son STAR o Google Knowledge Graph entre otros. La calidad de las respuestas obtenidas por estos sistemas no es muy elevada y sea poyan en el uso de textos estructurados para adquirir el conocimiento.

Dominio cerrado

En esta categoría encontramos aquellos sistemas que son capaces de responder a preguntas únicamente en algunos temas en concreto. Ejemplos claros de esta categoría serían los SBR Baseball y LUNAR que respondían a preguntas relacionadas con la liga de béisbol y con muestras de roca respectivamente. En general, la precisión de estos sistemas es elevada debido a que se limita el

[10] https://web.stanford.edu/~jurafsky/

campo de las consultas y como consecuencia el propio tamaño de las bases de datos. Los SBR de dominio cerrado en ocasiones se combinan para crear sistemas de dominio abierto.

2.1.2.2. Clasificación basada en tipo de preguntas

La generación de las respuestas por parte de los SBR está completamente relacionada con el tipo de pregunta que deben interpretar (Moldovan, Pasca y Harabagiu 2003). Moldovan (2003) considera que en más de un tercio de los errores producidos por los SBR se deben a una mala identificación del tipo de pregunta recibida. Chanda y K. Madhavi (2017) identifican seis tipos de preguntas que se pueden utilizar para clasificar los SBR. La ilustración 1 muestra los diferentes tipos de preguntas en un SBR.

Ilustración 1 - Tipos de preguntas en un Sistema de Búsqueda de Respuestas

Preguntas de tipo factoide

Se trata de preguntas que responden a preguntas del tipo ¿Qué …?, ¿Cuál o cuáles ...?, ¿Cuándo…?, ¿Quién…? O ¿Cómo? Se trata de preguntas sencillas que pueden ser respondidas con una única frase. Normalmente adquieren el conocimiento de grandes repositorios de preguntas dónde las respuestas son denominadas entidades y ofrecen una gran precisión a la hora de responder. Ejemplos de este tipo de preguntas podrían ser, ¿Cuál es la capital de Italia?, ¿Quién fue el inventor de la bombilla?

Preguntas de tipo lista

La respuesta a este tipo de preguntas son una lista de hecho o entidades. Los SBR no requieren de grandes técnicas de PLN para obtener gran precisión en sus respuestas. Por otro lado, uno de los mayores problemas planteados para este tipo de preguntas es identificar el número umbral de entidades dentro de nuestra lista de respuestas. Algunas de las cuestiones que podríamos considerar

dentro de esta categoría serían “Enumerar los reyes godos en España” o “Listado de la fauna y flora autóctona de la Sierra de Segura”.

Preguntas de confirmación

Este tipo de preguntas tienen respuestas de sí o no. En ocasiones, la respuesta para las preguntas dentro de esta categoría requiere de gran investigación por parte de expertos en la materia. Aparte de este tipo de preguntas, dentro de esta categoría también puede haber preguntas de opinión que requieran información subjetiva sobre un evento o entidad. Los SBR utilizan técnicas de minería de opiniones y la web social para obtener respuestas a preguntas de tipo opinión. La mayor ventaja de este tipo de preguntas es la gran cantidad de opiniones públicas que podemos encontrar en Internet, aunque por otro lado la mayor desventaja es la cantidad de respuestas fallases o spam y la detección de estas. Ejemplo de preguntas que podríamos englobar en esta categoría sería: ¿Fue Rodrigo Díaz de Vivar un personaje real?, ¿Es justo el sistema de financiación de la Iglesia Cristiana en España?

Preguntas de tipo causal

Categorizamos en este grupo aquellas preguntas del estilo de. ¿Por qué…? O ¿Cómo…? Para responder estas preguntas se requiere de una descripción sobre una entidad. Normalmente la respuesta consiste en párrafos o incluso documentos completos. Ejemplos de estas preguntas podrían ser: ¿Por qué ha sido galardonado el artista X con el premio Goya honorífico a su trayectoria cinematográfica?, ¿Cómo se preparada una tortilla de patata?

Preguntas hipotéticas

Las preguntas hipotéticas esperan una respuesta de situaciones las cuales no han ocurrido y por lo tanto la precisión de esta respuesta dependerá del usuario que realiza la pregunta y del contexto de este. Por este mismo motivo, la precisión de preguntas hipotéticas es normalmente baja. Suele tratarse de preguntas del estilo ¿Qué sucedería si…?

Preguntas complejas

La respuesta a estas preguntas es compleja, puesto que como normal general requiere sintetizar e inferir la información de diferentes y diversas fuentes de información. Por ejemplo, este tipo de preguntas pueden ser del estilo de ¿Cuáles son los motivos del agujero de la capa de ozono?

2.1.2.3. Clasificación basada en fuente de información

Dependiendo de la fuente de información utilizada para que el SBR adquiera el conocimiento podemos encontrar las siguientes categorías descritas a continuación.

Sistemas basados en texto

Se considera dentro de este grupo aquellos SBR que adquieren el conocimiento a partir de fuentes en modo textual. Estas fuentes de datos suelen tener los datos estructurados, aunque no es una condición necesaria. En este punto, es importante que comprendamos los diferentes modos en los que se presenta la información textual:

- Estructurados: se refiere a textos que siguen un esquema bien definido, como es el caso de las tablas en las bases de datos.

- Semiestructurados: Aunque los textos no sigan un esquema rígido mantienen cierta estructura. Esto se puede observar en recetas de cocina o en prospectos médicos.

- No estructurados: Los datos de texto no siguen ningún tipo de estructura o esquema.

Ilustración 2 Tipos de fuentes en sistemas de búsqueda de respuestas basados en texto

Sistemas multimedia

Los SBR dentro de esta categoría obtienen el conocimiento de fuentes de datos cuyo contenido no se encuentra en modo texto, sino formatos de audio, imagen o video.

Sistemas híbridos

Existen sistemas que obtienen el conocimiento tanto de fuentes en modo texto como de sistemas multimedias.

2.1.2.4. Clasificación basada en tipo de respuesta generada

En función de cómo se genera la respuesta podemos clasificar los SBR en dos tipos.

Respuestas extraídas

En este grupo encontramos los SBR cuya respuesta se encuentra del mismo modo en el que aparece en las fuentes de información. La respuesta son oraciones o párrafos extraídos de documentos que tienen la información o contenido multimedia sin procesar.

Respuestas generadas

Encontramos en esta categoría SBR cuyas respuestas son confirmativas, que ofrecen opiniones o que se muestran en modo conversación con el usuario.

2.1.2.5. Clasificación basada en estrategia seguida

En función de la estrategia o el mecanismo utilizado por los SBR para encontrar una respuesta a las preguntas de los usuarios podemos clasificarlos en los siguientes tipos.

Sistemas basados en métodos lingüísticos

Se trata de sistemas que utilizar reglas, plantillas redes semánticas u ontologías que funcionan muy bien para un dominio en concreto. Por el contrario, si modificamos el dominio para el que se utilizan estos sistemas necesitaremos volver a cambiar estar reglas. Otro inconveniente de este tipo de SBR es que construir una base de conocimiento es un trabajo pesado y lento.

Sistemas basados en métodos estadísticos

El auge de estos sistemas se debe a que obtienen la base de conocimiento de los datos existentes en Internet. Para obtener respuestas con precisión requieren de muchos datos, aunque pueden ser utilizados para responder preguntas de otros dominios e incluso lenguas.

2.1.3. Aplicaciones de Sistemas de Búsqueda de Respuestas

Los usos de los SBR son muy amplios y según se avanza en los campos de DL y PLN cada vez se descubren nuevas oportunidades para ellos. Entre los campos donde más se está apostando por este tipo de sistemas cabe destacar la robótica, domótica, sistemas financieros, sectores de turismo y atención al cliente, salud o sistemas de comunicación con empleados entre otros.

Sin embargo y dado que nos encontramos ante el desarrollo de un estudio comparativo de diferentes estrategias dentro de un dominio de carácter académico nos centraremos en los usos de los SBR dentro de la educación.

2.1.3.1. Aplicaciones en educación

El uso de los SBR dentro del campo de la educación es muy interesante y cada vez está siendo más investigado por parte de los equipos especializados en IA y más concretamente aquellos que hacen uso de las tecnologías de PLN.

Dentro de la enseñanza hay muchos subcampos como los que veremos a continuación donde el uso de SBR está aportando gran valor.

Tutores virtuales

El uso de SBR como tutores virtuales se realiza a través del uso de sistemas conversacionales. Estos sistemas conversaciones se comportan de manera similar al que tienen los sistemas de atención al cliente en portales de venta de productos o a los sistemas de comunicación con empleados en el campo de los recursos humanos. La gran diferencia del uso de SBR como tutores virtuales es el dominio para el que han sido desarrollados, que es de carácter administrativo y académico.

El éxito de estos tutores virtuales se debe a que en la actualidad existe mucha ambigüedad entre los tipos de centros y las necesidades propias de cada alumno. Por lo tanto, el uso de sistemas conversacionales, también conocidos como chatbots[11], para la comunicación con el estudiante permite ofrecerles un sistema personalizado.

Aprendizaje de idiomas

Otro de los campos donde se está aplicando el uso de SBR es en el aprendizaje de idiomas. Estos sistemas se adaptan al nivel académico del estudiante y son de gran utilidad para innovar y reforzar los métodos de enseñanza. Al igual que en el caso de los tutores virtuales, estos sistemas se suelen presentar en modo de sistemas conversacionales. Un claro ejemplo de un SBR utilizado para la enseñanza de idiomas es el de Gengobot[12], un chatbot multilingüe usado para enseñar conceptos básicos de japonés.

Aprendizaje individual

La utilización de SBR para el aprendizaje individual permite adapta los sistemas a la evolución de un único estudiante que desea aprender de manera autónoma. Es el estudiante el que determina el progreso en el que los conocimientos se van adquiriendo a partir de las respuestas obtenidas por los sistemas.

Aprendizaje colectivo

[11] Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. In I. Maglogiannis, L. Iliadis, & E. Pimenidis (Eds.), Artificial Intelligence Applications and Innovations (pp. 373–383). Springer International Publishing.
[12] https://gengobot.com/

El aprendizaje colectivo permite que dos o más personas puedan adquirir conocimiento sobre un mismo tema de manera independiente.

Este tipo de sistemas se adapta al nivel pedagógico de la fuente de información utilizada, ya que el nivel de las respuestas que se dé debe ser acorde al del estudiante que lo está utilizando. Por ese mismo motivo, se utilizan sistemas de dominio cerrado debido a que los sistemas de dominio abierto no son capaces de abordar este requisito.

En otras palabras, estos sistemas deben ser adaptar sus respuestas en función de la persona que realiza la pregunta. Un claro éxito de este tipo de aprendizaje es el de EduQA[13].

2.1.4. Enfoques de Sistemas de Búsqueda de Respuestas

A continuación, se muestran la lista de los diferentes enfoques que han sido utilizados para el desarrollo de los SBR.

Enfoque lingüístico

Este enfoque contiene métodos basados en IA que integran técnicas de PLN y base de conocimiento. El conocimiento es organizado utilizando plantillas, reglas, ontologías y redes semánticas. Se consideran técnicas lingüísticas: Parsing, Tokenization y POS tagging[14]. Estas técnicas se aplican sobre la pregunta del usuario para extraer la respuesta adecuada de la base de datos estructurada.

Enfoque estadístico

Este tipo de enfoque ha ido incrementando su uso debido que han aparecido nuevos repositorios de texto en línea. El éxito de estos enfoques es la posibilidad de formular las consultas en lenguaje natural en lugar de utilizar consultas en lenguajes técnicos como es SQL. Sin embargo, la desventaja del enfoque estadístico es que trata cada término de forma independiente y no identifica las características lingüísticas de una combinación de palabras o frases.

Enfoque basado en coincidencia de patrones

Este enfoque se basa en el uso de patrones de texto. A diferencia de otros simplifica el procesamiento sofisticado de otras técnicas puesto que aprenden estructuras de texto automáticamente a partir de pasajes en lugar de utilizar conocimientos lingüísticos complejos para recuperar respuestas (Dwivedi

[13] A. Agarwal et al., "EDUQA: Educational Domain Question Answering System Using Conceptual Network Mapping," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 8137-8141, doi: 10.1109/ICASSP.2019.8683538.
[14] Straka, M., & Straková, J. (2017, August). Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (pp. 88-99).

et al., 2013). La simplicidad de tales sistemas los hace bastante favorables para pequeñas implementaciones y usos.

Los SBR que utilizan patrones pueden hacerlo de dos modos diferentes: Utilizando los patrones de superficie recomendado para los SBR de dominio abierto (Ravichandran et al., 2002) y basados en el uso de plantillas que se utilizan para SBR de dominio cerrado (Guda, Sanampudi and Manikyamba 2011).

Enfoque generativo

Este enfoque se presenta como el futuro a seguir dentro de los SBR. La esencia de este enfoque se basa en que podemos formular preguntas al sistema y esperamos que este sea capaz de producir respuestas que obtengan la información de múltiples fuentes y la presente en forma de respuesta resumida.

El uso de un enfoque generativo nos permite que el SBR sea capaz de explicar toda la pregunta y no únicamente de responderla como ocurre con otros enfoques, los cuales sufren el problema de que se sobre ajustan a sesgos superficiales en conjuntos de datos.

2.1.5. Arquitecturas de Sistemas de Búsqueda de Respuestas

Entendiendo la arquitectura de un sistema de Software como la identificación de los diferentes módulos que componen un sistema completo, observamos que las arquitecturas de los SBR han sufrido pocas variaciones a lo largo de los 50 años de historia que tienen. Al fin y al cabo, un SBR consta de dos entidades principales que son la pregunta y la respuesta y las relaciones entre ambas.

Martínez-barco et al. (2007) consideran que la arquitectura típica de un SBR tiene como núcleo más básico un sistema de recuperación. Sin embargo, la especialización de estos sistemas requiere entender la información más allá de la obtención de esta. Por lo tanto, es común que la mayoría de los SBR estén compuestos por al menos los 3 componentes mostrados en la ilustración 3.

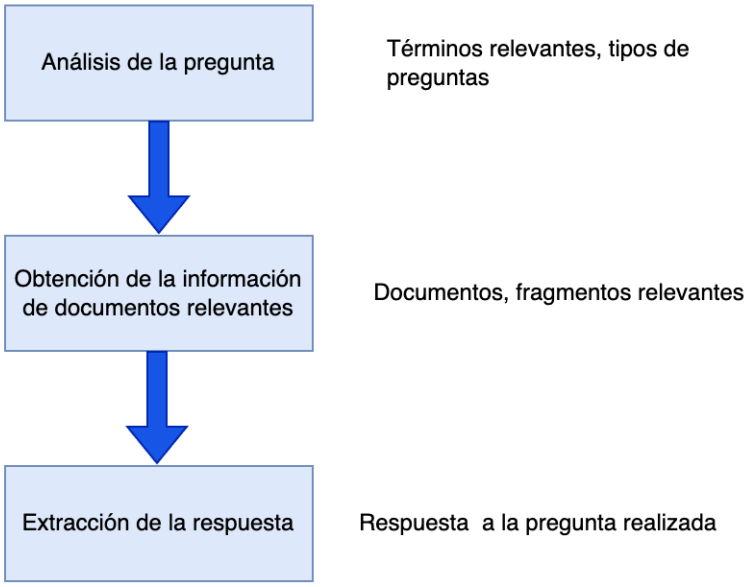


Ilustración 3 Arquitectura básica de 3 capas de un sistema de búsqueda de respuestas

Es habitual encontrar diseños separados en 4 módulos, como propone M. Ajitkumar et al. (2016). Las arquitecturas de 4 módulos dividen el módulo central en: procesamiento de documentos y extracción de párrafos

Partiendo del diagrama identificamos las funcionalidades que deben ser proporcionadas por cada uno de los módulos:

Análisis de pregunta

En este módulo nos centramos en identificar el tipo de pregunta realizada. Además, partiendo de la extracción de los términos más importantes, el SBR debe identificar la clase de respuesta que ha de devolver al usuario. Para lograr esto existen una gran variedad de técnicas tales como Bag of Words[15] , Stemming[16], Lemmatization[17], Eliminación de stop words[18] entre otras.

Obtención de información

[15] Hanna M. Wallach. 2006. Topic modeling: beyond bag-of-words. In Proceedings of the 23rd international conference on Machine learning (ICML '06). Association for Computing Machinery, New York, NY, USA, 977–984. DOI:https://doi.org/10.1145/1143844.1143967
[16] Watzlawik, M., & Valsiner, J. (2012). The Making of Magic: Cultural Constructions of the Mundane Supernatural. The Oxford Handbook of Culture and Psychology, 2(6), 1930–1938. https://doi.org/10.1093/oxfordhb/9780195396430.013.0038
[17] Díaz, Abdel & Perez-Suarez, Airel. (2022). La lematización en el preprocesamiento de textos para RI. Evaluación de distintos algoritmos de lematización.
[18] Tin Kam Ho, "Bootstrapping text recognition from stop words," Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170), 1998, pp. 605-609 vol.1, doi: 10.1109/ICPR.1998.711216.

El objetivo de este módulo, cómo su propio nombre indica, es obtener la información necesaria para dar una respuesta a la pregunta realizada por el usuario. La forma de acceder a la información variará no sólo dependiendo de la pregunta realizada sino en función de la estructuración de los datos.

En el caso más sencillo la información se encuentra estructurada y a través de mecanismos como es el SQL podemos obtener la información que buscamos de modo sencillo. Sin embargo, la información no se encuentra estructurada normalmente sino semiestructurada o no estructurada.

Extracción de la respuesta

Es en este módulo dónde los SBR concentran su mayor esfuerzo para ser capaces de identificar cual es la respuesta más adecuada para la pregunta realizada por el usuario.

2.2. Técnicas de Aprendizaje profundo en PLN

Al igual que ha ocurrido con otras áreas de la IA, el PLN está viviendo un gran momento debido a los avances realizados en las técnicas de aprendizaje profundo. Esto se debe principalmente a dos factores: grandes avances a nivel de computación y el interés de grandes empresas que dedican equipos de investigación en el trabajo de nuevos paradigmas y arquitecturas.

A pesar de que en los últimos años parece que el uso de Transformers se ha convertido en el estándar de facto para el trabajo con textos, es importante que conozcamos cuales han sido las técnicas de DL que nos han traído aquí.

2.2.1 Redes Neuronales Recurrentes tipo codificador-decodificador

En la segunda década del siglo XXI, el uso de las arquitecturas codificador-decodificador en RNN se convirtió en el enfoque más utilizado tanto para la traducción automática como para la predicción de secuencia a secuencia (seq2seq[19]) en general. Las principales ventajas que trajeron estos sistemas fue la posibilidad de entrenar un solo modelo directamente en oraciones de origen y de destino y el hecho de trabajar con secuencias de entrada y de salida de longitud variable.

Sutskever et al. (2014) publican “Sequence to Sequence Learning with Neural Networks” convirtiéndose en uno de los primeros documentos en introducir el modelo basado en codificador-decodificador. La idea es usar un LSTM[20] para leer la secuencia de entrada, un paso de tiempo a la vez, para obtener una gran representación vectorial de dimensión fija, y luego usar otro LSTM para extraer la secuencia de salida de ese vector. La idea fundamental de las LSTM se basa en el uso de

[19] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, 4(January), 3104–3112.
[20] Kumar, J., Goomer, R., & Singh, A. K. (2018). Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model for Cloud Datacenters. Procedia Computer Science, 125, 676–682. https://doi.org/10.1016/j.procs.2017.12.087

celdas de memoria, creadas para propagar información de las primeras capas de la red, tal y como se muestra en la ilustración 4.

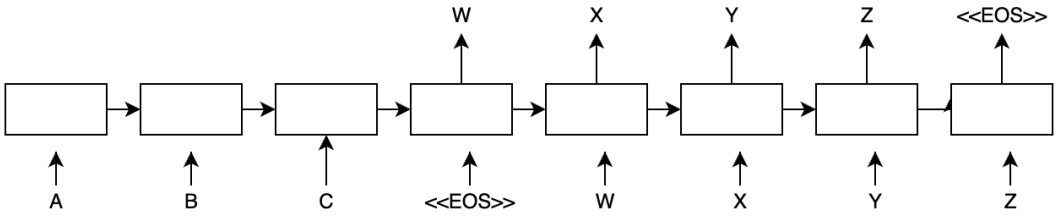


Ilustración 4 Representación del modelo de codificador-decodificador de Sutskever para la traducción de textos

Otro artículo fundamental para comprender la importancia de esta arquitectura es “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation.”, de Cho, et al. (2014). Al igual que Sutskever desarrolla un RNN codificador-decodificador que consta de dos RNN Una RNN codifica una secuencia de símbolos en una representación vectorial de longitud fija y el otro decodifica la representación en otra secuencia de símbolos. Sin embargo, en su propuesta no utiliza unidades LSTM, sino que desarrolla una unidad RNN más simple llamada unidad recurrente cerrada, por sus siglas en inglés GRU de Gated recurrent unit. La ilustración 5 representa la arquitectura propuesta por Cho.

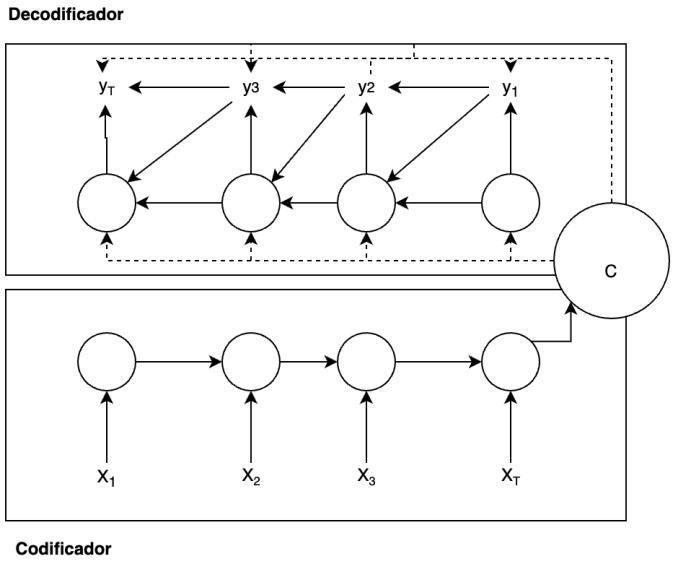


Ilustración 5 Representación de arquitectura codificador-decodificador propuesta por Cho

El gran inconveniente que presentan las arquitecturas basadas en RNN es que son muy lentas de entrenar ya que es complicado paralelizar el trabajo, debido a que reciben las entradas de forma secuencial.

2.2.2. Mecanismos de atención

El mecanismo de atención, introducido por Bahdanau et al. (2014), surge para mejorar el rendimiento del modelo codificador-decodificador para la traducción automática. La idea fundamental era permitir que el decodificador utilizase las partes más relevantes de la secuencia de entrada de una manera flexible, mediante una combinación ponderada de todos los vectores de entrada codificados, asignando a los vectores más relevantes lo pesos más altos.

Es importante mencionar que el desarrollo de los mecanismos de atención está inspirado en la neurociencia y el comportamiento humano. Este mecanismo permite al cerebro priorizar la percepción de una fuente de información frente a otras y dedicar menos atención al resto. La aparición del mecanismo de atención ha revolucionado el modo en el que se estaba utilizando el DL para enfrentarse a nuevos retos.

La arquitectura propuesta por Bahdanau consiste en el uso de un mecanismo de atención en el medio entre una RNN bidireccional[21] usada como codificador y una RNN como decodificador.

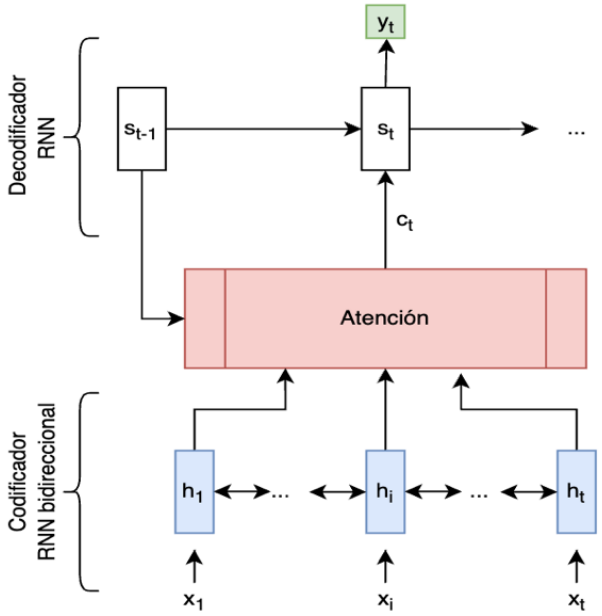


Ilustración 6 Representación de arquitectura de Bahdanau

[21] Chen, X., Ragni, A., Liu, X., & Gales, M.J. (2017). Investigating Bidirectional Recurrent Neural Network Language Models for Speech Recognition. INTERSPEECH.

El mecanismo de atención permite que el modelo del decodificador acceda a todos los estados ocultos en lugar de a un solo vector como ocurría con las LSTM. De este modo pretende acabar con el cuello de botella de información del estado intermedio

Aunque el uso del mecanismo de atención mejora el rendimiento que los sistemas basados en RNN proporcionan, presentan otros inconvenientes. El principal inconveniente de esta arquitectura sigue siendo que el entrenamiento de los modelos es muy lento, porque ha de ser secuencial y no permite fácilmente la transferencia de los pesos del modelo.

2.2.3. Transformers

Los Transformers se basan en la idea de la auto atención introducida por primera vez por un grupo de investigadores de Google[22] en el artículo “Attention is All you Need” (Vaswan, Shazeer Parmar et al. 2017). Estos son modelos de aprendizaje automático semi-supervisados que se utilizan principalmente con datos de texto y han reemplazado a las redes neuronales recurrentes en tareas de procesamiento de lenguaje natural.

Los Transfomers utilizan una arquitectura de tipo codificador-decodificador como la propuesta por Bahdebau (2014). Sin embargo, estos proponen una composición interna completamente diferente a las soluciones actuales en ese momento, ya que se prescinde del uso de las RNN y utilizan unos módulos llamados auto atención, del inglés self-attention. Esta nueva arquitectura mejora el rendimiento del entrenamiento de los modelos al paralelizar el aprendizaje.

Los Transformers están diseñados para trabajar con datos de secuencia y tomarán una secuencia de entrada y la usarán para generar una secuencia de salida un elemento a la vez. Un transformador está formado por dos componentes principales. El primero es un codificador que se centra en la secuencia de entrada y el segundo es un decodificador que lo hace en la secuencia de salida y predice el siguiente elemento de la secuencia.

De una manera simplista podríamos decir que le objetivo del mecanismo de auto atención es ayudarnos a crear conexiones similares dentro de una secuencia de texto. Esto se logra gracias a que para obtener una representación de una secuencia se relaciona diferentes posiciones de esta.

El objetivo del mecanismo de auto atención es conocer con que otra palabra de la secuencia está relacionada la palabra que se procesa en un instante de tiempo. Para ello, se obtiene un vector de salida a partir de los siguientes 3 vectores de entrada:

[22] https://research.google/

- Vector de consulta (Q): La consulta que representa el vector de una palabra.
- Vector de claves (K): Las claves que se corresponden con todas las palabras de la propia secuencia.
- Vector de valores (V): El valor vectorial de la propia palabra que se está procesando en un instante de tiempo concreto.

La capa de auto atención produce una salida secuencial de la misma longitud para cada elemento de la entrada. Los elementos de salida de esta capa se pueden calcular en paralelo, consiguiendo una implementación más eficiente que la proporcionada por los sistemas basados en RNN.

Otro punto de ventaja que presentan los Transformer frente a las RNN es el uso del contexto, o tamaño de ventana de memoria que pueden utilizar las arquitecturas basadas en mecanismos de auto atención. Las RNN son capaces de referenciar a palabras que han aparecido anteriormente en la secuencia de entrada. Sin embargo, cuando trabajamos con secuencias muy largas, no pueden acceder a palabras muy antiguas. A pesar de que con las GRU y las LSTM se consigue ampliar este tamaño de ventana en las RNN, la capacidad sigue siendo limitada. La gran ventaja del mecanismo de auto atención es que posee una ventana infinita y que únicamente queda limitada por la potencia computacional de nuestros sistemas.

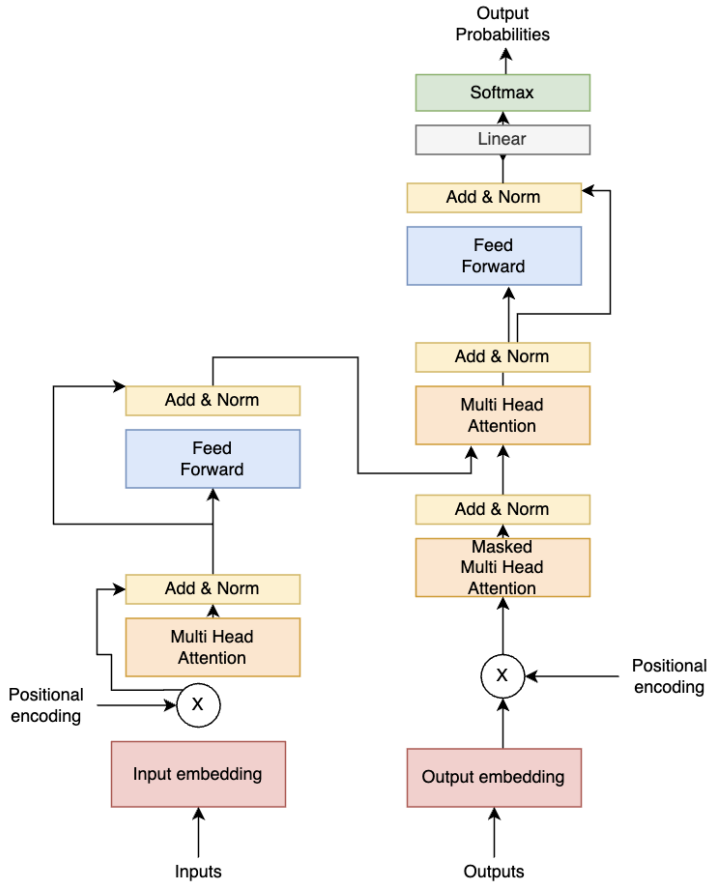


Ilustración 7 Representación de arquitectura de Transformers,

Codificación

A partir de la secuencia de texto que se recibe en la entrada se generan dos secuencias:

- Secuencia de vectores de palabras. Cada una de estas palabras se representa como un vector.
- Secuencia de codificaciones posicionales. Consisten en una representación vectorial de la posición de la palabra en la oración original.

A continuación, se unen las dos secuencias creadas a partir de la entrada y se aplica el mecanismo de atención de múltiples cabezas, Multi-Head Attention[23] en inglés. Este mecanismo que se invocará varias veces en paralelo es capaz de atender partes de la secuencia de entrada de manera diferente (por ejemplo comparación de dependencias a corto plazo frente a largo. Finalmente, la salida de este se envía una red neuronal de retro propagación, conocidas en inglés como Feed Forward[24].

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017.
Thomas
[24] Sazli, M.H. (2006). A brief review of feed-forward neural networks.

Decodificación

El proceso de la decodificación comienza obteniendo el vector de representación de las palabras del resultado del punto temporal justamente anterior.

Al igual que en el proceso de codificación se une la secuencia de codificación posicional y se envía al mecanismo de atención de múltiple cabezas que nos devuelve un vector de salida. Este vector, se envía junto con el resultado de la codificación para el instante actual a un nuevo mecanismo de atención.

La salida de este último mecanismo de atención se envía una red de retro propagación. La salida de este será un vector de probabilidades de la siguiente palabra. Por ello al aplicar la función Softmax[25] que se observa en la ilustración 8 obtendremos la palabra con mayor probabilidad.

2.3. Modelos basados en Transformer

Gracias a la arquitectura de Transformer, que hemos descrito en el apartado anterior, han aparecido diferentes tecnologías que pretenden convertirse en el estado del arte del PLN. Destacan especialmente GPT-3 y BERT, en las cuales profundizaremos a continuación. Antes de profundizar en estas tecnologías es importante destacar que BERT es una herramienta de código abierto y permite que los desarrolladores puedan realizar ajustes según sus necesidades y resuelvan varias tareas posteriores. GPT-3, por otro lado, no es de código abierto y tiene acceso limitado vía API.

2.3.1. BERT

BERT es desarrollado por Google y fue presentado por Devlin et al. (2018) y su nombre se corresponde con las siglas en inglés de “Bidirectional Encoder Trepresentations from Transformers”. El hecho de que sea bidireccional se refiere a que BERT analiza las frases de búsqueda en ambas direcciones, considerando las palabras situadas a la izquierda y la derecha de cada palabra clave. Esto se logra gracias a la técnica conocida como Masked LM[26] (MLM) que permite el entrenamiento bidireccional en modelos en los que antes era imposible.

Este modelo se ha entrenado haciendo uso de datos no etiquetados, como artículos de Wikipedia, grandes corpus de noticias o libros, consiguiendo de este modo entrenar el modelo sobre una cantidad de datos enorme. Gracias a ello, BERT tiene una representación general del lenguaje natural que posteriormente podremos utilizar para solventar tareas más concretas. BERT presenta resultados de última generación en una amplia variedad de tareas de PLN, incluidos los sistemas de búsqueda de

[25] https://www.semanticscholar.org/topic/Softmax-function/966784
[26] Kushilevitz, G., Markovitch, S., & Goldberg, Y. (2020). A Two-Stage Masked LM Method for Term Set Expansion. 6829–6835. https://doi.org/10.18653/v1/2020.acl-main.610

respuesta (SQuAD v1.1), la inferencia de lenguaje natural o clasificación de textos entre otras. Por lo tanto, el uso de BERT para realizar una tarea específica es trivial, simplemente necesitamos entrenar el modelo ya pre-entrenado con los datos del problema que queremos cubrir. Por ejemplo, en nuestro caso, que estamos construyendo un SBR debemos introducir la pregunta, el separador [SEP] y la respuesta.

El hecho de que BERT sea un sistema de código abierto ha permitido que otros puedan utilizarlo como base para el desarrollo de nuevas tecnologías para el procesamiento de lenguaje natural. Destacamos los casos de RoBERTA[27] , DistilBERT[28] o XLNet[29] entre otros.

La principal ventaja proporcionada por BERT es el hecho de que el modelo sea bidireccional, en lugar de un modelo unidireccional como es el caso de GPT. El uso del modelo bidireccional hace que el contexto aprenda en función de las palabras que lo rodean en lugar de solo considerar la palabra anterior o posterior.

2.3.2. GPT's

GPT son las siglas de Generative Pre-Trained Transformer y se trata de una serie de herramientas que han sido desarrolladas por la organización OpenAI[30]. En concreto GPT-3, su última entrega hasta el momento, recibe el nombre por ser la tercera generación de estas tecnologías.

La primera entrega de GPT, GPT-1, fue lanzada en el año 2018. El modelo se entrenó con un conjunto enorme de corpus de libros y generando un modelo con 117 millones de parámetros. Su arquitectura aplicaba un decodificador de 12 capas con un mecanismo de auto atención para el entrenamiento. Uno de sus mayores logros fue la capacidad que demostró para desempeñar con rendimiento zero-shot[31] en varias tareas. GPT-1 demostró que el modelado del lenguaje generativo se puede explotar con un concepto de pre-entrenamiento eficaz para generalizar el modelo, gracias al uso del aprendizaje por transferencias y con muy pocos ajustes.

Un año más tarde, en 2019, se liberó la segunda versión, GPT-2. Consistió en un modelo que fue entrenado con un conjunto de datos mucho más grande que su primera versión y generando un modelo 10 veces más grande, con 1500 millones de parámetros. Consistió en una arquitectura mucho

[27] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. 1. http://arxiv.org/abs/1907.11692
[28] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2–6. http://arxiv.org/abs/1910.01108
[29] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. Advances in Neural Information Processing Systems, 32(NeurIPS), 1–18.
[30] https://openai.com/blog/tags/scholars/
[31] Xian, Y., Lampert, C.H., Schiele, B., & Akata, Z. (2019). Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41, 2251-2265.

más grande. Esta segunda entrega de OpenAI superó su primera entrega al mejorar los resultados de la realización de diversas tareas como son la traducción, el resumen y la clasificación de textos.

GPT-3, tal y como se ha mencionado anteriormente, fue lanzada en el año 2020. Actualmente está considerado como el modelo más poderoso de IA para su uso en procesamiento de lenguaje natural debido a que es capaz de generar teto indistinguible de si lo hubiera hecho un humano. Según investigadores y desarrolladores, GPT-3 es capaz de mantener "conversaciones extrañamente naturales",

La razón por la que ha conseguido superar los resultados de sus versiones anteriores, aún usando los mismos principios, es que ha sido entrenado con un conjunto de datos mucho más extenso. Estos datos fueron extraídos de fuentes cómo CommonCrawl[32] o Wikipedia[33]. El modelo generado cuenta con 175 mil millones de parámetros. Sin embargo, su alta complejidad, y enorme tamaño, hace que sea mucho más costoso e inconveniente realizar inferencias. La cantidad de parámetros tan elevada lo hace pesado en recursos y un desafío para la aplicabilidad práctica en tareas en su forma actual.

La tabla 1 muestra las diferencias de los modelos de cada una de las versiones de GPT.

	GPT-1	GPT-2	GPT-3
Parámetros	117 millones	1500 millones	175 mil millones
Capas de decodificadores	12	48	96
Tamaño de contexto (tokens)	512	1024	2048
Capas ocultas	768	1600	12288
Tamaño de batch	64	512	3.2 millones

Tabla 1 Características de modelos de la serie GPT

GPT-3 tiene una clara ventaja sobre BERT, ya que requiere muy pocos ejemplos de datos para entrenar el modelo. En contraposición, el hecho del enorme tamaño del modelo hace que sea extremadamente costoso de ejecutar, y por lo tanto el usuario promedio no cuente con sistemas adecuados.

[32] https://commoncrawl.org/
[33] https://en.wikipedia.org/

2.4. Aprendizaje por transferencia

El aprendizaje por transferencia, del inglés Transfer Learning, se refiere a un conjunto de métodos que se benefician de modelos de ML previamente entrenados para generar nuevos modelos especializados en un problema concreto a resolver.

El aprendizaje por transferencia se inspira en las capacidades de los seres humanos para transferir conocimientos a través de tareas, el aprendizaje de transferencia tiene como objetivo aprovechar el conocimiento de un dominio de origen para mejorar el rendimiento del aprendizaje o minimizar el número de ejemplos etiquetados necesarios en un dominio de destino (Wei, Zhang et al, 2018).

Es habitual encontrar que necesitamos resolver una tarea de clasificación en un dominio de interés, pero solo tenemos suficientes datos de entrenamiento en otro dominio. En tales casos, la transferencia de conocimientos, si se realiza con éxito, mejoraría enormemente el rendimiento del aprendizaje al evitar esfuerzos muy costosos de etiquetado de datos.

En los últimos años, el aprendizaje por transferencia ha surgido como un nuevo marco de aprendizaje para abordar numerosos problemas dentro del área de PLN (Peters et al., 2018; Howard y Ruder, 2018; Radford et al., 2018; Devlin et al., 2018). La ilustración 8 representa como podemos partir del conocimiento adquirido por un modelo para ser capaces de especializarnos y solucionar otros problemas.

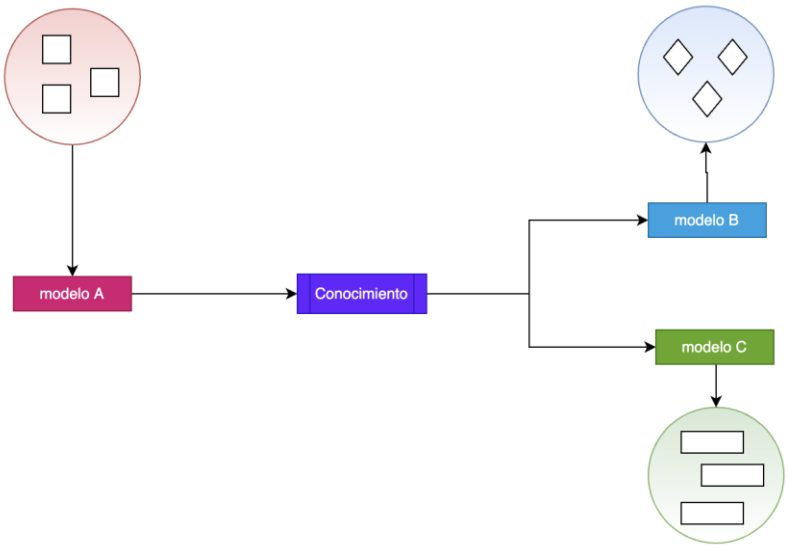


Ilustración 8 Representación de sistemas de Aprendizaje por transferencia

Por lo tanto, partiendo de que el aprendizaje por transferencia consiste en aplicar el conocimiento adquirido de un modelo para resolver problemas más pequeños podemos identificar las siguientes ventajas:

- Los requisitos son mucho más sencillos que cuando pretendemos resolver un problema desde cero.
- A diferencia de los modelos tradicionales de aprendizaje profundo no se requiere la presencia de científicos de datos para resolver problemas.
- La ventaja principal, y que ha sido determinante para este trabajo, es que se reduce la necesidad de tener grandes cantidades de datos para entrenar nuestros modelos.
- La duración de los entrenamientos se reduce abismalmente, llegando a pasar de un entrenamiento que podría durar días a simplemente unos pocos segundos.
- Se tienen requisitos de memoria menores. Esto se debe a que como consecuencia de utilizar un modelo ya pre-entrenado se reduce el número de operaciones matemáticas que se han de realizar.

A pesar de los beneficios proporcionados al utilizar técnicas de aprendizajes por transferencia, la clave radica en saber escoger cual de los diferentes modelos ya pre-entrenados se adapta mejor a nuestras necesidades.

2.3.1. Aproximaciones de uso de aprendizaje por transferencia

Existen dos aproximaciones a la hora de utilizar el aprendizaje por transferencia cuando queremos enfrentamos a problemas de modelado predictivo: extracción de características y ajuste fino (del inglés fine tunning). Estas técnicas no son excluyentes y la combinación de ambas puede ofrecernos unos grandes resultados en un plazo de tiempo muy reducido.

Extracción de características

Esta estrategia consiste en “congelar” las capas del modelo previamente entrenado para preservar el aprendizaje existente y a partir de aquí se agregan nuevas capas al modelo de aprendizaje profundo para aprender información adicional.

Esta aproximación es nuestra mejor opción cuando se desea transferir el conocimiento de un modelo de aprendizaje automático a otro, pero no queremos volver a entrenar el segundo modelo de datos.

Ajuste fino

Se descongela completamente el modelo pre-entrenado anteriormente y se entrena con un factor de aprendizaje, del inglés learning rate (lr)[34], mucho menor para lograr que el modelo se adapte a los nuevos retos a los que se enfrenta.

El uso de esta estrategia tiene sentido cuando ya estamos entrenando nuestros propios modelos de DL o deseamos ajustar la salida de un modelo ya existente para un conjunto de datos. Al partir de un modelo más grande para entrenar uno mucho más pequeño nos beneficiamos de cualquier trabajo ya realizado sin tener que entrenarlo de nuevo. Esta estrategia es más rápida y eficiente que la extracción de características por sí sola

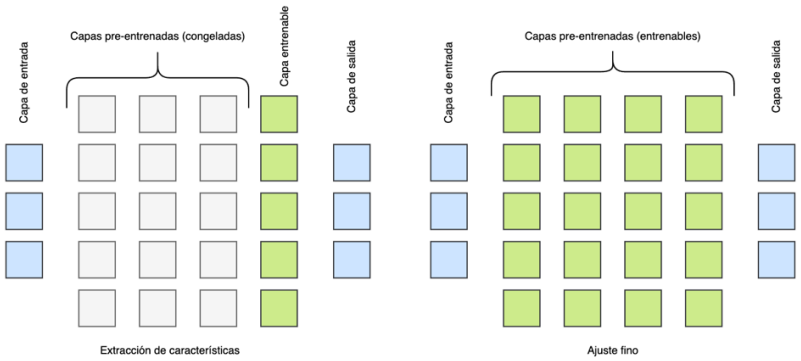


Ilustración 9 Representación de extracción de características frente a ajuste fino

2.3.2. Pasos en aprendizaje por transferencia

Se identifican las siguientes etapas o pasos necesarios para pasar de un modelo pre-entrenado a un nuevo modelo haciendo uso del aprendizaje por transferencia.

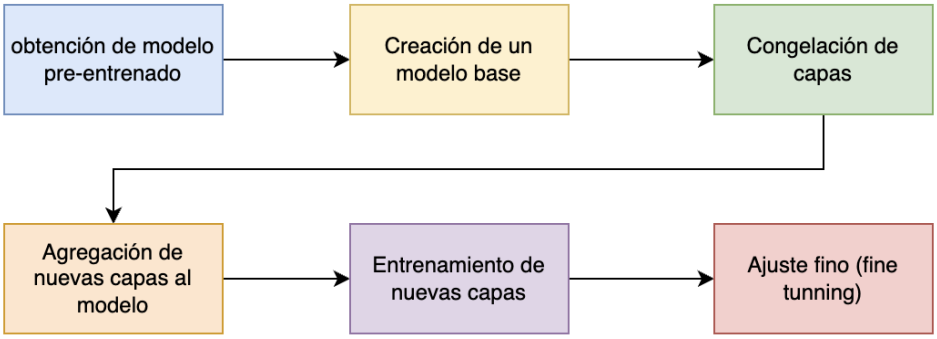


Ilustración 10 Pasos en aprendizaje por transferencia

Obtención de modelo pre-entrenado

[34] Y. Wu et al., "Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 1971-1980, doi: 10.1109/BigData47090.2019.9006104.

El primer paso consiste en identificar el modelo pre-entrenado que vamos a utilizar como base de nuestro entrenamiento. Para ello deberemos entender la tarea que vamos a realizar. Y el modelo que seleccionemos debe ser compatible con la misma.

Creación de un modelo base

Una vez que hemos seleccionado el modelo pre-entrenado que utilizaremos podemos descargar los pesos de la red, de este modo ahorraremos tiempo de entrenamiento adicional a la hora de entrenar el modelo. En ocasiones puede que el modelo base tenga más neuronas en la capa de salida de las que nosotros necesitamos para resolver nuestro problema. En este caso necesitamos eliminar la capa de salida final y cambiarla en consecuencia.

Congelación de capas

Para evitar el trabajo de tener que aprender las características básicas debemos congelar las capas iniciales del modelo que hemos seleccionado. De no congelar las capas iniciales, perderemos todo el aprendizaje que ya ha tenido lugar, es decir, empezaríamos con los pesos establecidos en esas capas y se irían modificando.

Adicción de nuevas capas de entrenamiento

El único conocimiento que se reutiliza del modelo base es la capa de extracción de características. Por lo tanto, se necesita agregar nuevas capas para ser entrenadas. Estas capas suelen corresponderse con las capas de salida finales.

Entrenamiento de nuevas capas

Necesitamos entrenar el modelo haciendo uso de las nuevas capas que hemos añadido. Esto normalmente debemos hacerlo porque el modelo previamente entrenado no tendrá el mismo número de neuronas de salida que el que nosotros tenemos que construir.

Ajuste fino (fine tunning)

Este último paso se utiliza para mejorar el rendimiento de nuestro modelo. Tal y como se mencionó anteriormente, consiste en descongelar una parte del modelo base y entrenar todo el modelo nuevamente haciendo uso de una tasa de aprendizaje muy baja. Al hacer uso de una tasa de aprendizaje muy baja evitamos el conocido problema del sobreajuste, del inglés overfitting[35].

[35] https://pubs.acs.org/doi/full/10.1021/ci0342472

2.3.3. Áreas de utilización de aprendizaje por transferencia

El aprendizaje por transferencia tiene diferentes áreas de aplicación, siendo las más extendidas el área de visión por computador, el procesamiento de audio y el procesamiento de lenguaje natural.

Visión por computador

Es conocido el uso de redes de aprendizaje profundas para resolver cualquier tipo de tarea relacionada con el procesamiento de imágenes, debido a que estas son capaces de identificar características complejas en imágenes. El tipo de redes neuronales utilizadas cuenta con capas densas[36], en las que las capas superiores no suelen afectar a la lógica básica. Las capas primeras capas de estas redes neuronales se centran en la detección de objetos, eliminación de ruido, etc. por lo que es adecuado el aprendizaje por transferencia puesto que todas las tareas de visión por computados requieren de conocimientos básicos y detecciones de patrones de imágenes familiares.

Procesamiento de audio

El uso de aprendizaje por transferencias está muy presente en sistemas de domótica[37] como son Siri u Ok Google entre otros. Es muy habitual que modelos entrenados para el reconocimiento de voz en inglés se utilicen para reconocer otros idiomas como el francés.

Procesamiento de lenguaje natural

Este es uno de los campos que resulta más atractivo para las técnicas de aprendizaje por transferencia, especialmente en los últimos años donde la aparición de modelos de aprendizaje como BERT[38], XLNet[39], Albert[40], etc. han proporcionado muy bueno resultados para tareas como son la predicción de la siguiente palabra, la respuesta a preguntas y la traducción automática.

En concreto esta área resulta de especial interés para este trabajo debido a que se trata del área de la inteligencia artificial que se centra en resolver el problema que es cubierto en este trabajo: los sistemas de búsqueda de respuestas.

---

[36] Pelt, D. M., & Sethian, J. A. (2017). A mixed-scale dense convolutional neural network for image analysis. Proceedings of the National Academy of Sciences of the United States of America, 115(2), 254–259. https://doi.org/10.1073/pnas.1715832114

[37] https://hal.laas.fr/hal-01916886

[38] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). 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(Mlm), 4171–4186.

[39] Song, X., Wang, G., Huang, Y., Wu, Z., Su, D., & Meng, H. (2020). Speech-XLNet: Unsupervised acoustic model pretraining for self-attention networks. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2020-October, 3765–3769. https://doi.org/10.21437/Interspeech.2020-1511

[40] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. 1–17. http://arxiv.org/abs/1909.11942

3. Objetivos y metodología de trabajo

En este tercer capítulo, se exponen los objetivos y la metodología de trabajo que se ha llevado a cabo para la elaboración de este proyecto de fin de Máster.

3.1. Objetivo general

El objetivo general de este proyecto es realizar un estudio comparativo de las diferentes técnicas para la construcción de sistemas de búsquedas de respuestas con el fin de encontrar cuál de ellas proporciona mejores resultados para un dominio cerrado, que viene determinado por el temario en castellano de asignaturas del Grado de Informática de la Universidad Internacional de la Rioja.

3.2. Objetivos específicos

Los objetivos específicos del presente estudio son los siguiente:

1. Revisión sistemática de las investigaciones y avances realizados para la construcción de Sistemas de Búsqueda de Respuestas haciendo uso de tecnologías de Inteligencia Artificial.
2. Explorar diferentes técnicas y arquitecturas de inteligencia Artificial utilizadas para la construcción de Sistemas de Búsqueda de Respuestas.
3. Diseñar e implementar pruebas de conceptos siguiendo las diferentes técnicas de Inteligencia Artificial exploradas para la construcción de SBR.
4. Identificar las asignaturas y los datos de las mismas que serán utilizados para el desarrollo de el estudio presente.
5. Determinar las métricas utilizadas para realizar una comparación objetiva entre las diferentes técnicas de Inteligencia Artificial exploradas.
6. Sintetizar los resultados obtenidos y enumerar la lista de puntos a favor y en contra de cada una de las técnicas utilizadas en el estudio comparativo.

3.3. Metodología del trabajo

El proceso de investigación seguido para la elaboración de este trabajo de fin de Máster se ha dividido en 4 fases: fase exploratoria, fase de inicio y planificación, fase de desarrollo y fase de análisis.

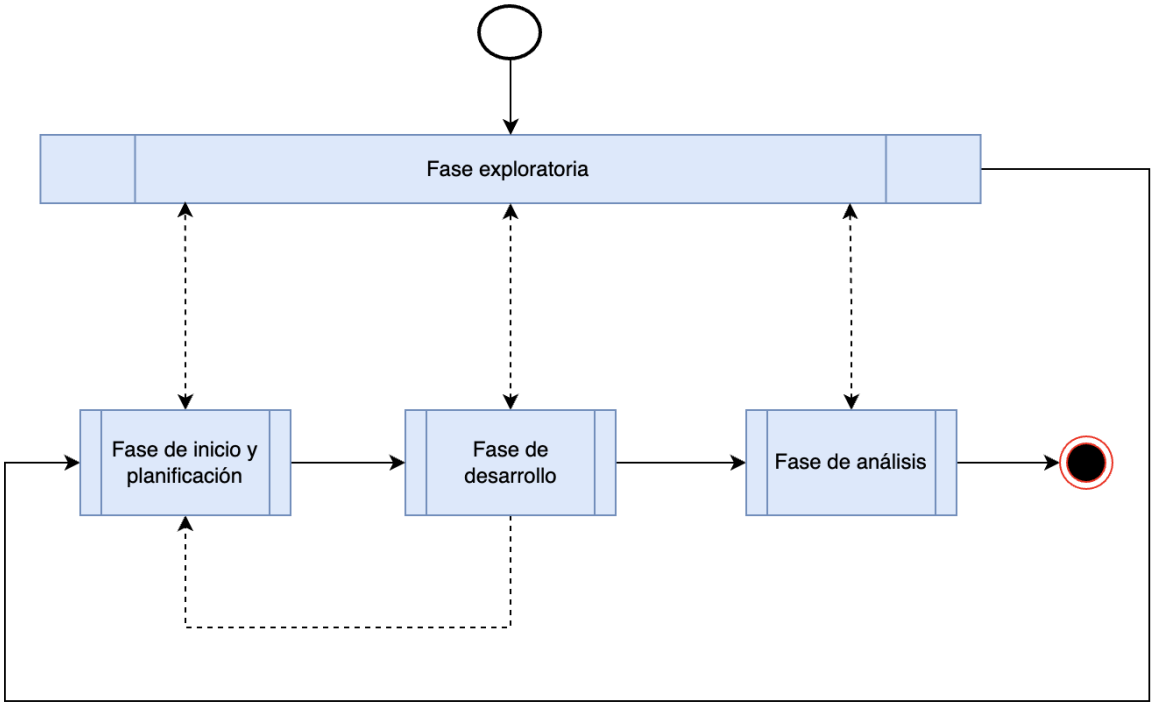


Ilustración 11 Metodología de trabajo

La fase exploratoria se lleva a cabo para la identificación de las técnicas de Inteligencia Artificial que serán consideradas para el desarrollo del estudio comparativo. Tal y como se observa en el diagrama anterior, esta fase se prolonga a lo largo del ciclo de vida completo del estudio puesto que la adquisición y actualización de los conocimientos teóricos es fundamental para el éxito del trabajo.

- Estudio de las historia y evolución de los Sistemas de Búsqueda de Respuesta para una comprensión de los pasos que se han de seguir para la elaboración de este trabajo.

- Estudio de diferentes sistemas de clasificación de SBR que permite orientar la identificación de las técnicas que deben ser comparadas para nuestro caso de estudio.

- Elaboración de una revisión continua de la literatura relacionada con las técnicas de IA involucradas en el desarrollo de SBR.

La fase de inicio y planificación permite la elaboración de los pasos a seguir para los desarrollos realizados en este estudio.

- Identificar qué técnicas de IA y modelos pre-entrenados se utilizarán en las fases de desarrollo y análisis de este trabajo.
- Selección del temario de las asignaturas del Grado de Informática ofrecido por la UNIR que formaran parte del estudio de investigación.
- Definición de banco de pruebas que será utilizado para la comparación de las diferentes técnicas de IA que utilizadas.
- Identificación de las métricas de evaluación para una medición objetiva de los resultados obtenidos en cada técnica utilizada.
- Planificación de las fases de desarrollo para cada técnica de IA.

La fase de desarrollo nos permite describir el funcionamiento y comportamiento de las diferentes técnicas y modelos de IA utilizadas para el desarrollo de nuestro sistema. Es importante que la detección de cualquier contratiempo o problema detectado durante la fase de desarrollo podría implicar cambios en la fase de planificación.

- Implementación de SBR y desarrollo de pruebas de concepto requeridas por cada una de las técnicas identificadas en la fase de inicio y planificación.
- Parametrización y personalización de las implementaciones realizadas en el punto anterior.
- Obtención de los resultados de entrenamiento y de evaluación de cada una de las técnicas utilizadas.
- Valoración de cada uno de los modelos estudiados de manera independiente.

La fase de análisis nos permite una comparación de los desarrollos realizados en este estudio, así como la elaboración de conclusiones y recomendaciones aprendidas durante el mismo.

- Valoración global considerando los resultados obtenidos en la fase de desarrollo para cada una de las técnicas y corroboración de estos resultados con la información obtenida durante la fase exploratoria.
- Desarrollo de conclusiones obtenidas tras haber realizado el análisis de los resultados de las diferentes técnicas utilizadas.

- Identificación y recomendación de investigaciones futuras a partir del trabajo realizado durante el desarrollo del presente.

- Elaboración de recomendaciones a seguir para el desarrollo real del caso de estudio de este trabajo.

4. Planteamiento de la comparativa

4.1. Trabajo previo

Es importante destacar que sin el trabajo de investigación realizado hasta este momento no sería posible abordar este punto ni los siguientes. Gracias a la lectura y profundización de los diferentes trabajos, referenciados a lo largo de este documento, he adquirido los conocimientos teóricos necesarios no sólo para elegir las tecnologías que participarán en este estudio comparativo, sino también para comprender la necesidad de un sistema como el que se pretende construir.

4.1.1. Descripción de sistema

El estudio del estado del arte nos ha permitido comprender la evolución que han tenido el desarrollo de los SBR a lo largo de la historia. Por otro lado, este proceso de investigación me ha proporcionado comprender cuales son las arquitecturas y tecnologías más adecuadas que deben formar parte del estudio presente. Concretamente las arquitecturas de IA basadas en los mecanismos de auto atención, en concreto los Transformers, han revolucionado el mundo del procesamiento del lenguaje natural y por tanto serán las elegidas para formar parte de este estudio comparativo.

Sin embargo, partiendo de las limitaciones existentes, tanto temporales como de recursos, este estudio comparativo se apoyará en el uso de modelos ya pre-entrenados a partir de corpus enormes y haciendo uso de grandes sistemas con los que no cuento para el desarrollo de este trabajo. Gracias al uso del aprendizaje por transferencia, el cual ya ha sido explicado en este documento, nos beneficiaremos de estos modelos para poder usarlos para resolver nuestras propias tareas.

En el estudio únicamente usaremos modelos que han sido pre entrenados con corpus en castellano o multiidioma, ya que estos nos darán mejores resultados para la necesidad que queremos cubrir: Un SBR que se utilizará para validar respuestas a preguntas del temario de la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento, ofrecida en el grado de informática de la UNIR.

El SBR que se requiere para este estudio entenderá única y exclusivamente preguntas en castellano, del mismo modo que las respuestas serán en el mismo idioma. La fuente de datos utilizada para la evaluación de los modelos será el propio temario de la asignatura, descartándose cualquier contenido multimedia y seleccionando únicamente el contenido en modo texto. Este temario se encuentra en castellano completamente.

El tipo de respuesta que evaluaremos serán de tipo factoide. Tal y como vimos en el estado del arte estas preguntas se responden con una o varias frases y encuentras su contenido en la propia fuente de información

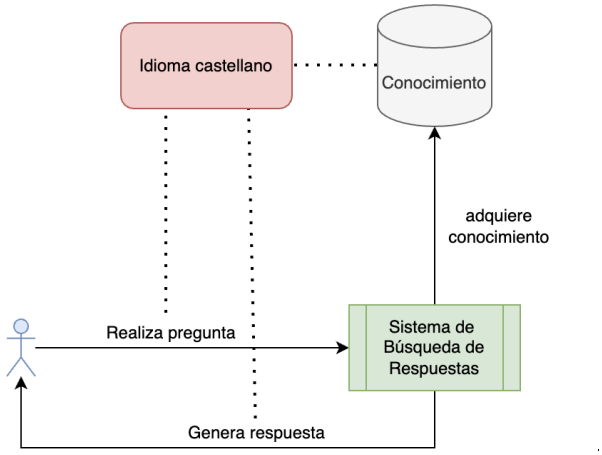


Ilustración 12 Representación de flujo de datos en el sistema de Búsqueda de Respuestas

En referencia al tipo de dominio del lenguaje de los modelos pre entrenados, es importante mencionar que se trata de dominios abiertos ya que las fuentes de información con la que han sido entrenados no se limitan a un área de conocimiento en concreto. Es importante mencionar que el uso de un modelo de dominio cerrado proporcionaría mejores resultados, tal y como hemos visto en el documento, sin embargo, las limitaciones existentes hacen que esto sea inabordable en el ámbito académico de este trabajo.

4.1.2. Datos de entrenamiento

Tal y como se ha mencionado previamente, el estudio comparativo será realizado para asignaturas de UNIR, más concretamente para la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento del Grado en Ingeniería Informática. Por lo tanto, utilizaremos el contenido de los temas tanto para evaluar los modelos existentes como para el entrenamiento de aquellos modelos ya pre-entrenados a través del uso de técnicas de ajuste fino.

Los datos serán extraídos del temario de las asignaturas y adaptados a un fichero en texto en el mismo formato seguido por el bien conocido SQuAD2.0[41], que le debe su nombre a “The Standford Question Answering Dataset”. Este formato nos da gran flexibilidad a la hora de definir nuestros propios conjuntos de datos para el entrenamiento de modelos especializados en el desarrollo de SBR,

[41] https://rajpurkar.github.io/SQuAD-explorer/

a la vez que nos permiten beneficiarnos de dudas y problemas ya resueltos, como es el procesamiento de este tipo de formatos.

Ilustración 13 Extracto del dataset utilizado para ela evaluación de modelos

4.2. Entorno de trabajo

Para el desarrollo de esta comparativa se ha utilizado la herramienta de desarrollo online Colab[42] en su modalidad de pago por subscripción para contar de este modo con instancias sin límite de uso y con la posibilidad de hacer uso de mejores GPU's y memorias RAM de mayor tamaño. El código fuente utilizado para la implementación de los diferentes sistemas es de libre acceso y se encuentra publicado en Github[43] , concretamente en la siguiente URL https://github.com/ivancorrales/master-ia-tfm. El código fuente ha sido desarrollado en su totalidad haciendo uso del lenguaje de programación Python[44]. Para el trabajo con los Transformer he utilizado el framework de aprendizaje automático, de código abierto, PyTorch[45]

En este repositorio de carácter público, aparte del código fuente utilizado para el estudio, podremos encontrar los ficheros utilizados para generar los datasets así como el conjunto de datos utilizado para evaluar los diferentes modelos.

[42] https://colab.research.google.com/
[43] Plataforma de alojamiento que permite el versionado de proyectos con Git, https://github.com/
[44] Python es un lenguaje de programación interpretado, https://www.python.org/
[45] https://pytorch.org/

4.3. Soluciones

Son muchas las soluciones existentes basadas en Transformers que nos permiten el tratamiento de lenguaje natural y en concreto para la construcción de sistemas de búsqueda de respuesta. Para acotar el número de soluciones contempladas en el estudio presente me centraré en el uso de modelos basados en la arquitectura BERT, como son RoBERTa[46] y DistilBERT[47] .

Gracias a las técnicas de aprendizaje por transferencia, un modelo no tiene que partir de cero, sino que se puede aprovecharse de modelos ya entrenados y entrenarlos con otro conjunto de datos. De hecho, existen modelos publicados que han partido del mismo modelo como base y. han utilizado el mismo conjunto de datos para entrenarlos. Esto ocurre puesto que los modelos son públicos y los conjuntos de datos utilizados también. Las predicciones de estos modelos no tienen porque ser idénticas debido a la aleatoriedad de los sistemas de Deep Learning y al hecho de que los híper parámetros de estos modelos no tienen porque haber sido inicializados con los mismos valores.

4.3.1. RoBERTa

RoBERTa es desarrollado por el equipo de IA de Facebook[48] y es anunciado en el artículo de Liu, Ott et al. (2019) en la publicación de "RoBERTa: A Robustly Optimized BERT Pretraining Approach".

RoBERTa se basa en la estrategia del enmascaramiento del lenguaje seguida por BERT. Sin embargo, realiza cambios como es la modificación de los hiperparámetros clave en BERT, aparte de realizar el entrenamiento haciendo uso de tasas de aprendizaje mucho mas grandes. A parte de estos cambios arquitectónicos RoBERTa ha sido entrenado con un orden de magnitud de más datos que BERT. Durante un periodo de tiempo más largo. Para el entrenamiento de esta arquitectura se utilizó conjunto de datos sin anotar a parte de un conjunto de datos novedoso extraído de artículos de noticias públicos.

Los resultados obtenidos por RoBERTa en la prueba comparativa GLUE demuestran que se encuentra en el nivel más alto de la clasificación y que se pueden mejorar los resultados obtenidos por BERT en el desempeño de diferentes tareas de PLN.

Los modelos basados en RoBERTa que formarán parte del estudio vienen descritos a continuación:

PlanTL-GOB-ES/roberta-base-bne-sqac[49]

[46] https://ai.facebook.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/
[47] https://huggingface.co/docs/transformers/model_doc/distilbert
[48] https://ai.facebook.com/
[49] Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., ... & Villegas, M. (2021). Spanish language models. arXiv preprint arXiv:2107.07253.

Se trata de un modelo de Transformer basado en RoBERTa para el idioma español. Este modelo fue originalmente pre-entrenado con el corpus en idioma español más grande hasta el momento de su publicación, con un tamaño de 570 GB de datos, después de haber sido limpiados y eliminando las duplicaciones.

El sufijo del nombre, bne, son las siglas de Biblioteca Nacional de España. Esto se debe a que fue la propia Biblioteca Nacional de España la que utilizando técnicas de crawling[50], creó el corpus a partir de datos publicados en Internet entre los años 2009 y 2019. La tabla 2 recoge los detalles del corpus BNE.

Corpora	Número de documentos	Número de Tokens	Tamaño
BNE	201,080,084	135,733,450,668	570GB

Tabla 2 Detalles de corpus BNE

En este estudio utilizaremos una versión del modelo que ha sido entrenado con el dataset SQAC[51] (Spanish Questions-Answering Corpus). Este dataset permite entrenar a los modelos para resolver tareas de SBR. Está formado por un conjunto de 18817 preguntas con 6247 contextos y teniendo entre 1 y 5 posibles respuestas cada pregunta.

Este trabajo ha sido parcialmente financiado por la Secretaría de Estado de Digitalización e Inteligencia Artificial (SEDIA) de España en el marco del Plan-TL, y el Centro de Cómputo del Futuro, una iniciativa del Centro de Supercomputación de Barcelona e IBM (2020)

PlanTL-GOB-ES/roberta-large-bne-sqac[52]

La diferencia de este modelo con el modelo anteriormente descrito, PlanTL-GOB-ES/roberta-base-bne-sqac, es la arquitectura del modelo de RoBERTa del cual se ha partido. En lugar de utilizar la versión reducida de RoBERTa, se utiliza la versión large que tiene 355 millones de parámetros en lugar de los 125 millones.
El proceso de pre-entrenamiento de este modelo ha sido exactamente igual, realizando un pre entrenamiento con el corpus bne y posteriormente con SQAC.

jamarju/roberta-base-bne-squad-2.0-es

Este modelo, a pesar de que no tiene una organización que lo sustente como era el caso de los modelos pertenecientes a Plan TL-GOB-ES, me ha resultado interesante de tener en cuenta para el

[50] https://link.springer.com/chapter/10.1007/978-3-662-10874-1_7
[51] https://huggingface.co/PlanTL-GOB-ES/roberta-large-bne-sqac
[52] Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., ... & Villegas, M. (2021). Spanish language models. arXiv preprint arXiv:2107.07253.

estudio comparativo porque ha sido entrenado inicialmente con el corpus de bne y posteriormente con el corpus squad-2.0-es[53]. El hecho de considerar para le estudio un modelo que utilice el dataset squad-2.0-es me parece relevante puesto que es una traducción automática al idioma español del bien conocido Standord Question Answering Dataset (SQUAD v2). El modelo inicial del que parte es el modelo simplificado de RoBERTa.

jamarju/roberta-large-bne-squad-2.0-es

Lo único que diferencia a este modelo de la versión jamarju/roberta-base-bne-squad-2.0-es es que en lugar de haber partido con la versión reducida de RoBERTa ha sido basado en la versión large.

4.3.2. DistilBERT

DistilBert es anunciado por Sanh, Debut et al. (2019) en la publicación de su artículo "DistilBERT, a destilled versión of BERT: smarller, faster, cheaper and lighter". En este artículo destacan que DistilBERT es un modelo más pequeño, rápido y liviano, y por lo tanto es más barato de preentrenar.

Tal y como se indica en el apartado anterior, la esencia de DistilBERT radica en proporcionar un modelo más genérico pero que sea más fácil de ajustar para gran variedad de tareas. Además, a diferencia con otros trabajos DistilBERT destaca por reducir el tamaño de BERT en un 40% al mimo tiempo que conserva el 97% de su comprensión de idioma. A nivel de rendimiento los estudios existentes han demostrado que es un 60% más rápido.

mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

Este modelo ha sido entrenado partiendo del modelo base bert-base-multilingual-cased., también conocido como DistilmBERT, y usando el conjunto de datos de SQuAD-es-v2.0 para su entrenamiento.

4.3.3. Ixambert

Se trata de un modelo que ha sido preentrenado para los idiomas inglés , español y euskera. El corpus utilizado está compuesto por las Wikipedias en inglés, espaol y euskera, a parte de noticias en euskera de periódicos online.

Es importante destacar que este modelo se ha utilizdo para transferir conocimientos del inglés al euskera en un sistema de control de calidad conversacional, tal y como indican Otegi, Agirre et al.

[53] Carrino, C. P., Costa-jussà, M. R., & Fonollosa, J. A. R. (2020, Mei). Automatic Spanish Translation of SQuAD Dataset for Multi-lingual Question Answering. Proceedings of the 12th Language Resources and Evaluation Conference, 5515–5523. Opgehaal van https://aclanthology.org/2020.lrec-1.677

(2020) en su publicación “Conversational Question Answering in Low Resource Scenarios: A Dataset and Case Study for Basque”.

MarcBrun/ixambert-finetuned-squad

Este modelo se trata de una implementación básica del modelo multilingüe "ixambert-base-cased", entrenado con el dataset SQuAD v1.1, que es capaz de responder preguntas básicas sobre hechos en inglés, español y euskera.

4.3.4. Tuneado de un modelo

Adicionalmente al uso de los modelos ya entrenados, como parte de este estudio se incluirá la evaluación de un modelo entrenado, es decir, que partir de un modelo ya existente se entre con un conjunto de preguntas y respuestas proporcionadas.

4.4. Criterios de evaluación

4.4.1. Datos de evaluación

Las preguntas que se utilizarán para evaluar los diferentes modelos que participan en nuestro estudio serán extraídas del temario de la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento. Se utilizarán las mismas preguntas y el mismo contexto para la inferencia por parte de los modelos.

4.4.2. Criterios de éxito

La elaboración de las preguntas será realizada de manera manual. Tal y como nos proporciona el dataset SQuAD.

Cuantificar si una respuesta ha sido correcta ante una pregunta es una tarea complicada. Esto se debe a que para una misma pregunta y un contexto de entrada podemos obtener diferentes respuestas válidas.

Tal y como indicamos anteriormente el conjunto de datos que elaboraremos estará inspirado en SQuAD, más concretamente en la versión 2.0, SQuAD2.0. Este a diferencia de SQuAD1.1 incluye preguntas adicionales que no pueden ser respondidas con el contexto proporcionado. Nuestro conjunto de preguntas que utilizaremos para el desarrollo contendrá 112 preguntas. De estas 112 preguntas 66 de ellas no tendrán respuesta valida. Las preguntas con respuesta valida contendrán hasta un máximo de 6 posibles respuestas correctas. Todas las respuestas deben corresponderse con

extractos directos del propio contexto, pero estos pueden ser encontrados de diferentes maneras. Un ejemplo se muestra en la ilustración 14:

```
ID: d8527d97-80e9-4b0b-9e02-6efcd5767372
----------------------------------------
Q: ¿Con qué se corresponden los nodos interiores de un algoritmo de búsqueda?

Context:
('Los algoritmos de búsqueda exploran el espacio de estados generando un árbol '
 'de búsqueda cuya raíz es el estado inicial. Los nodos al final de las ramas '
 'son los nodos hoja, que se corresponden con la lista abierta de estados no '
 'expandidos de un algoritmo. Los nodos interiores se corresponden con la '
 'lista cerrada de estados ya expandidos. La forma en la que el árbol se '
 'genera eligiendo expandir unos estados u otros determina el comportamiento '
 'de algoritmo de búsqueda. Hay que tener en cuenta que, aunque el espacio de '
 'búsqueda se represente como un árbol, pueden existir múltiples caminos desde '
 'el estado inicial a un estado cualquiera. De esta manera, puede darse el '
 'caso de que el mismo estado aparezca en dos nodos del árbol diferentes. Para '
 'evitar desperdiciar trabajo y memoria es pues conveniente realizar detección '
 'de duplicados. La detección de duplicados se suele implementar ayudándose de '
 'una tabla hash. Así pues, si se detecta que un estado ya existe existen dos '
 'posibilidades: si el nuevo camino es de mayor o igual coste, el estado se '
 'descarta; si el nuevo camino es de menor coste, se actualiza el coste, la '
 'acción que lo genera y el puntero al estado padre, es decir, se mueve de una '
 'rama del árbol a otra. En este último caso puede ser necesario reexpandir el '
 'estado, es decir, regenerar sus sucesores y todo el subárbol del cual es '
 'raíz, aunque esto depende del algoritmo en cuestión. Los algoritmos de '
 'búsqueda tienen cuatro características fundamentales')

True Answers:
 - Los nodos interiores se corresponden con la lista cerrada de estados ya expandidos
 - se corresponden con la lista cerrada de estados ya expandidos
 - con la lista cerrada de estados ya expandidos
```

Ilustración 14 Ejemplo de pregunta con más de una posible respuesta válida

La otra mitad del conjunto de datos estará compuesta por preguntas que no tienen una respuesta válida. Un ejemplo se muestra en la ilustración 15.

```
ID: 671e5932-353d-4625-9946-128b9fb2e9ad
----------------------------------------
Q: ¿Con que se corresponden los nodos exteriores de un algoritmo de búsqueda?

Context:
('Los algoritmos de búsqueda exploran el espacio de estados generando un árbol '
 'de búsqueda cuya raíz es el estado inicial. Los nodos al final de las ramas '
 'son los nodos hoja, que se corresponden con la lista abierta de estados no '
 'expandidos de un algoritmo. Los nodos interiores se corresponden con la '
 'lista cerrada de estados ya expandidos. La forma en la que el árbol se '
 'genera eligiendo expandir unos estados u otros determina el comportamiento '
 'de algoritmo de búsqueda. Hay que tener en cuenta que, aunque el espacio de '
 'búsqueda se represente como un árbol, pueden existir múltiples caminos desde '
 'el estado inicial a un estado cualquiera. De esta manera, puede darse el '
 'caso de que el mismo estado aparezca en dos nodos del árbol diferentes. Para '
 'evitar desperdiciar trabajo y memoria es pues conveniente realizar detección '
 'de duplicados. La detección de duplicados se suele implementar ayudándose de '
 'una tabla hash. Así pues, si se detecta que un estado ya existe existen dos '
 'posibilidades: si el nuevo camino es de mayor o igual coste, el estado se '
 'descarta; si el nuevo camino es de menor coste, se actualiza el coste, la '
 'acción que lo genera y el puntero al estado padre, es decir, se mueve de una '
 'rama del árbol a otra. En este último caso puede ser necesario reexpandir el '
 'estado, es decir, regenerar sus sucesores y todo el subárbol del cual es '
 'raíz, aunque esto depende del algoritmo en cuestión. Los algoritmos de '
 'búsqueda tienen cuatro características fundamentales')

True Answers:
```

Ilustración 15 Ejemplo de pregunta sin respuestas válidas

4.4.3. Mecanismo de predicción de respuestas

Para entender mejor como funciona este mecanismo de predicción de respuestas, vamos a descomponer este proceso en 3 pasos:

Tokenización: Partimos de una pregunta y del contexto que utilizaremos para encontrar la respuesta.

- Pregunta: ¿Quién nación en 1867?
- Contexto: Maria Salomea, conocida como Marie Curie, nació en 1867.

```
question = "¿Quién nació en 1867?"
context  = "Maria Salomea, conocida como Marie Curie, nació en 1867"
inputs = tokenizer(question,context,return_tensors="pt")
tokens = [item for sublist in inputs['input_ids'] for item in sublist]
print(f'[Tokens]\n{tokens}')
tokenized_text = tokenizer.convert_tokens_to_string(
    tokenizer.convert_ids_to_tokens(tokens)
)
print(f'[Texto tokenizado]\n{tokenized_text}')
sequence_ids = inputs.sequence_ids(0)
print(f'[Secuencia de Tokens]\n{sequence_ids}')
```

```
Tokens]
[tensor(0), tensor(729), tensor(8554), tensor(7858), tensor(334), tensor(1805), tensor(12046),
tensor(85), tensor(2), tensor(2), tensor(29868), tensor(37306), tensor(780), tensor(66),
tensor(6658), tensor(466), tensor(29094), tensor(5902), tensor(649), tensor(66), tensor(7858),
tensor(334), tensor(1805), tensor(12046), tensor(2)]
[Texto tokenizado]
<s>¿Quién nació en 1867?</s></s>Maria Salomea, conocida como Marie Curie, nació en 1867</s>
[Secuencia de Tokens]
[None, 0, 0, 0, 0, 0, 0, 0, None, None, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, None]
```

Ilustración 16 Proceso de tokenización de pregunta y contexto

Como se observa en la ilustración 16, a partir de la cadena de texto correspondiente a la pregunta y su contexto se crea una lista con valores numéricos que se corresponden con cada uno de los identificadores asociados a cada token que compone la entrada. De hecho, al convertir a modo texto los tokens, podemos observar que se han añadido unos símbolos especiales. Estos símbolos se utilizan para concatenar diferentes cadenas de texto.

Además, la secuencia de tokens nos indica con un valor de 1 que tokens se corresponden con el contexto (se representan con 1), cuales con la pregunta (se representan con 0) y cuales con los caracteres especiales (se representan como None) . Esto nos permitirá saber en que parte tenemos que buscar la respuesta.

Calculo de probabilidades: Se aplica un modelo, ya entrenado, sobre el conjunto de tokens que calculamos en el paso anterior. Esto nos devolverá dos listas de valores:

- start_logits: Lista de puntajes de que el token en esta posición sea el primer token de la respuesta.
- end_logits. Lista de puntajes de que el token en esta posición sea el último token de la respuesta.

```
outputs = model(**inputs)
starts = [item for sublist in outputs[0].tolist() for item in sublist]
ends   = [item for sublist in outputs[1].tolist() for item in sublist]
print(f'[start_logits]\n{starts}')
print(f'[end_logits]\n{ends}')
```

```
start_logits
[-6.810694694519043, -7.736298561096191, -7.660343170166016, -7.828719615936279, -
7.755526065826416, -7.351312637329102, -7.8634352684021, -7.791051387786865, -
7.376559257507324, -5.133049964904785, 8.920348167419434, -4.54658317565918, -
5.9402923583984375, -6.51347541809082, -7.024957656860352, -6.998234748840332, -
1.4827914237976074, -4.917675971984863, -4.268630504608154, -7.83840274810791, -
7.3958916664123535, -6.9027886390686035, -5.124028205871582, -6.8381500244140625, -
7.353182315826416]
end_logits
[-6.175658702850342, -7.8384928703308105, -7.7628865242004395, -7.781766891479492, -
7.950565338134766, -7.955604076385498, -7.582818031311035, -7.6477952003479, -
6.890715599060059, -6.618699550628662, -2.387667179107666, -5.650929927825928,
2.451488494873047, -5.962212085723877, -7.296243667602539, -7.4021830558776855, -
6.196744441986084, -5.408938884735107, 7.080603122711182, -4.936789512634277, -
6.027084827423096, -7.276189804077148, -6.457286357879639, -5.91390323638916, -
6.583681583404541]
```

Ilustración 17 Puntaje de tokens de entrada y salida

Búsqueda de la/s mejore/s respuestas: Se obtienen las posibles predicciones encontrando los elementos en start_digits y end_digits que maximizan el puntaje. Se debe tener en cuenta varias consideraciones para realizar este proceso correctamente.

- La posición del token final debe ser igual o mayor que la del token del principio.
- Únicamente tendremos en cuenta los tokens que pertenecen al contexto. Es decir, ni las preguntas ni los tokens especiales pueden ser parte de la respuesta. Para ello nos apoyaremos en la secuencia de tokens que obtuvimos en el primer paso.

```
max_score = 0
n_predictions = 4
answers = []
for start, start_value in enumerate(starts):
  if sequence_ids[start]==1:
    sub_ends = ends[start:]
    for end_idx, end_value in enumerate(sub_ends):
      end = start + end_idx
      if sequence_ids[end]==1:
        score = start_value + end_value
        if score > 0:
          answers.append({
            'score': score,
            'tokens': tokens[start:end+1],
          })
answers = list(reversed(sorted(
      answers,
      key=operator.itemgetter("score"),
)))[:n_predictions]
for answer in answers:
  print(f"({round(answer['score'].item(),3)})
{tokenizer.convert_tokens_to_string(tokenizer.convert_ids_to_tokens(answer['tokens']))}")
```

```
(16.001) Maria Salomea, conocida como Marie Curie
(11.372) Maria Salomea
(6.533) Maria
(5.598)  Marie Curie
```

Ilustración 18 Obtención de mejores predicciones

Como podemos observar en el ejemplo anterior predicciones con menos puntaje que la primera pueden ser consideradas igual o más validas que esta.

Como hemos observado en este apartado, siempre se pueden obtener una respuesta, una cadena de texto, que tenga un puntaje mayor que el resto.

Por otro lado, en nuestro conjunto de datos, existen preguntas que no se pueden responder con el contexto proporcionado. Para abordar este problema se hace uso de un threshold, o umbral de respuesta nula. Si no se obtiene ninguna respuesta con un puntaje mayor que el umbral la respuesta predicha será la cadena vacía. En la ilustración 18 se utiliza un umbral de respuesta nula de 0, como puede observarse en la línea “if score > 0”.

4.4.4. Métricas de evaluación basadas en predicción

Para la evaluación de los modelos que forman parte del estudio cubriremos aquellas métricas que nos permiten cuantificar la calidad, además daremos importancia a las preguntas que no tienen respuesta. Esto nos ayudará a mejorar el rendimiento y obtener resultados de más realistas.

En general, se emplean dos métricas para evaluar los SBR: Coincidencia exacta, del inglés exact match (EM) y F1. Cuando son posibles varias respuestas correctas para una pregunta determinada, se calcula la puntuación máxima sobre todas las posibles respuestas correctas.

Antes de comparar las predicciones con las respuestas verdaderas se realiza un proceso de normalización de texto. Esto consiste en convertir el texto en minúsculas, eliminar espacios innecesarios, eliminar signos de puntuación y eliminar artículos

Exact match

El cálculo de esta métrica es muy sencillo. Por cada par de pregunta y respuesta si los caracteres de la predicción se corresponden con una de las posibles respuestas válidas el resultado será EM=1, en caso contrario el valor será EM=0. Esta medida es estricta, por lo que con que haya un carácter invalido el resultado será EM=0. Por otro lado, al evaluar preguntas sin respuesta correcta, si el modelo predice cualquier texto, la respuesta será EM=0, mientras que si no recibe nada será EM=1.

```
ID: d8527d97-80e9-4b0b-9e02-6efcd5767372
----------------------------------------
Q: ¿Con qué se corresponden los nodos interiores de un algoritmo de búsqueda?

True Answers:
 - Los nodos interiores se corresponden con la lista cerrada de estados ya expandidos
 - se corresponden con la lista cerrada de estados ya expandidos
 - con la lista cerrada de estados ya expandidos

Prediction:  con la lista cerrada de estados ya expandidos

EM=1


ID: 104f97a2-9efc-482a-92b7-3665cfd400b9
----------------------------------------
Q: ¿Cuantos caminos hay desde un estado inicial a otro estado?

True Answers:
 - pueden existir múltiples caminos desde el estado inicial a un estado cualquiera
 - pueden existir múltiples caminos
 - múltiples caminos

Prediction:  múltiples

EM=0
```

Ilustración 19 Cálculo de EM para una respuesta válida y una invalida

En el primer ejemplo de la ilustración 19 el valor de EM es 1 porque la predicción se corresponde con una de las posibles respuestas verdaderas para la pregunta. Sin embargo, para el segundo ejemplo la

respuesta predicha no se corresponde con ninguna de las posibles respuestas, por eso el valor de EM es 0.

F1

Esta métrica se usa comúnmente para medir el rendimiento de los sistemas de clasificación de texto Esta métrica es apropiada cuando damos importancia a la precisión, pero también al recall[54], conocido en castellano como exhaustividad. En este caso, se calcula sobre las palabras individuales de la predicción frente a las de la respuesta verdadera. La cantidad de palabras compartidas entre la predicción y la verdad es la base de la puntuación F1: la precisión es la relación entre la cantidad de palabras compartidas y la cantidad total de palabras en la predicción, y el recuerdo es la relación entre la cantidad de palabras compartidas. al número total de palabras en la verdad fundamental. Concretamente esta métrica la podemos calcular con la siguiente fórmula:

$$\mathrm{F}_1 = 2 * \frac{\text{precisión} * \text{recall}}{\text{precisión} + \text{recall}}$$

dónde el valor de precisión se calcula dividiendo el número total de tokens comunes entre la respuesta esperada y la predicha entre el número de tokens de la respuesta predicha. El valor de recall se calcula dividiendo el número total de tokens comunes entre la respuesta esperada y la predicha entre el número entre el número de tokens de la respuesta esperada.

La ilustración 20 muestra un ejemplo de cálculo de la métrica F1 para un caso real.

[54] https://ieeexplore.ieee.org/abstract/document/791887/

```
ID: 104f97a2-9efc-482a-92b7-3665cfd400b9
----------------------------------------
Q: ¿Cuantos caminos hay desde un estado inicial a otro estado?

True Answers:
 - pueden existir múltiples caminos desde el estado inicial a un estado cualquiera
 - pueden existir múltiples caminos
 - múltiples caminos

Respuesta esperada: pueden existir múltiples caminos desde el estado inicial a un estado cualquiera
Respuesta predicha:  múltiples
Nº tokens coincidentes: 1
Nº tokens en respuesta predicha: 1
Nº tokens en respuesta esperada: 11
precisión: 1.0, recall: 0.09090909090909091
F1: 0.16666666666666669

Respuesta esperada: pueden existir múltiples caminos
Respuesta predicha:  múltiples
Nº tokens coincidentes: 1
Nº tokens en respuesta predicha: 1
Nº tokens en respuesta esperada: 4
precisión: 1.0, recall: 0.25
F1: 0.4

Respuesta esperada: múltiples caminos
Respuesta predicha:  múltiples
Nº tokens coincidentes: 1
Nº tokens en respuesta predicha: 1
Nº tokens en respuesta esperada: 2
precisión: 1.0, recall: 0.5
F1: 0.6666666666666666

F1=0.6666666666666666
```

Ilustración 20 Cálculo de métrica F1

En la evaluación de los resultados obtenidos para cada modelo se presentarán las métricas para el conjunto de respuestas totales, y también agrupadas en preguntas con posibles respuestas válidas y preguntas sin respuesta posible.

En los resultados mostrados en el siguiente punto observaremos que se calculan las métricas para diferentes valores de umbral de respuesta nula. El calculo de las métricas para diferentes valores del umbral, nos dará una visión más amplia de como debemos utilizar el modelo seleccionado para obtener mejores resultados.

5. Desarrollo de la comparativa

En este capítulo se mostrarán los resultados obtenidos por cada una de las soluciones planteadas, así como una comparativa entre las diferentes soluciones.

5.1. Entorno de ejecución

La ejecución del código implementado para la elaboración del estudio comparativo será ejecuta en una instancia de Google Colab. La ilustración 21 muestra las características de la GPU asignada para la ejecución del Notebook.

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.46       Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   36C    P0    27W / 250W |      2MiB / 16280MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Ilustración 21 Detalles de GPU

La ilustración 22 muestra la cantidad disponible de memoria RAM para la instancia asignada para la ejecución del Notebook.

```
from psutil import virtual_memory
ram_gb = virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))

if ram_gb < 20:
  print('To enable a high-RAM runtime, select the Runtime > "Change runtime type"')
  print('menu, and then select High-RAM in the Runtime shape dropdown. Then, ')
  print('re-execute this cell.')
else:
  print('You are using a high-RAM runtime!')
```

```
Your runtime has 27.3 gigabytes of available RAM

You are using a high-RAM runtime!
```

Ilustración 22 Código y ejecución de celda que muestra la memoria RAM disponible

5.2. Dataset de evaluación

Tal y como se indicaba anteriormente, para la evaluación de los modelos, un conjunto de datos creado manualmente que tiene el mismo número de preguntas con respuestas que de preguntas sin ninguna respuesta posible. En concreto este conjunto de datos está compuesto por 112 preguntas. A continuación, se muestran las estadísticas del dataset para ayudar a una mejor comprensión de los resultados obtenidos.

- Según el número de posibles respuestas:

Nº de respuestas	Total de preguntas
0	56
1	22
2	16
3	11
4	3
5	2
6	1

Tabla 3 Total de preguntas por número de respuestas

- Longitud de texto número de tokens

	Nº total	Nº medio de caracteres	Nº medio de Tokens
Preguntas	112	55.76	11.47
Respuestas	120	50.69	9.78
Contexto	15	871.07	170.94

Tabla 4 Estadísticas de preguntas, respuestas y contexto

Para calcular el número de tokens se ha utilizado BETO[55], se trata de un de un modelo pre-entrenado con texto en Español

[55] Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., & Perez, J. (2020). Spanish Pre-Trained BERT Model. Workshop Paper at PML4DC, ICLR, 1–10.

5.3. Resultados

5.3.1. PlanTL-GOB-ES/roberta-base-bne-sqac

EM

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.375	0.589286	0.160714
-2	0.428571	0.589286	0.267857
0	0.428571	0.589286	0.267857
2	0.446429	0.589286	0.303571
4	0.455357	0.589286	0.321429
6	0.473214	0.589286	0.357143
8	0.508929	0.589286	0.428571
10	0.508929	0.571429	0.446429
12	0.553571	0.571429	0.535714
14	0.580357	0.482143	0.678571
16	0.625	0.375	0.875
18	0.517857	0.035714	1

Tabla 5 Valores de EM para PlanTL-GOB-ES/roberta-base-bne-sqac

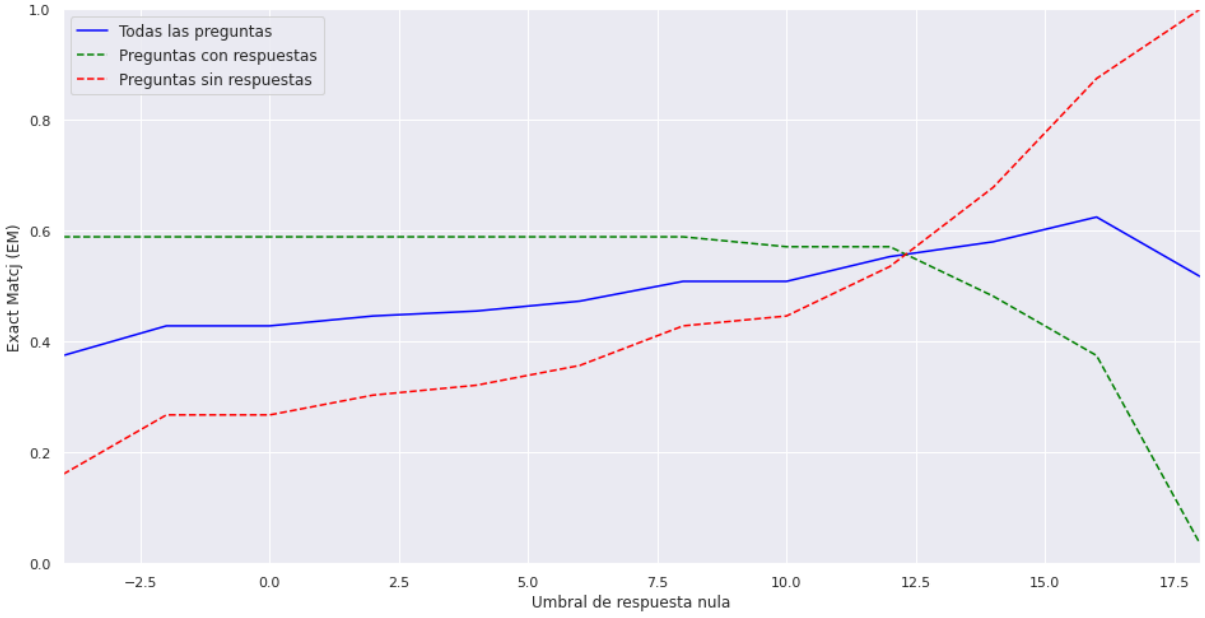


Ilustración 23 Variación de EM para PlanTL-GOB-ES/roberta-base-bne-sqac

F1

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.45402	0.747326	0.160714
-2	0.507592	0.747326	0.267857
0	0.501318	0.734778	0.267857
2	0.519175	0.734778	0.303571
4	0.528103	0.734778	0.321429
6	0.545961	0.734778	0.357143
8	0.577211	0.72585	0.428571
10	0.570365	0.694302	0.446429
12	0.615008	0.694302	0.535714
14	0.635318	0.592065	0.678571
16	0.66423	0.453459	0.875
18	0.525183	0.050366	1

Tabla 6 Valores de F1 para PlanTL-GOB-ES/roberta-base-bne-sqac

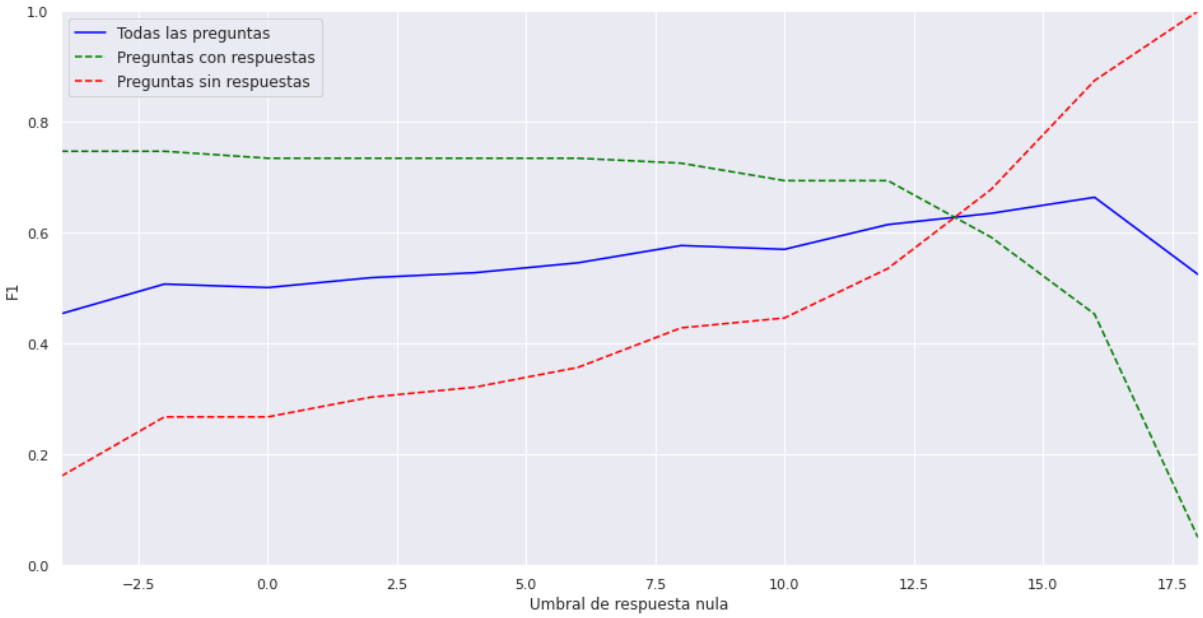


Ilustración 24 Variación de F1 para PlanTL-GOB-ES/roberta-base-bne-sqac

5.3.2. PlanTL-GOB-ES/roberta-large-bne-sqac

EM

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.580357	0.660714	0.5
-2	0.580357	0.625	0.535714
0	0.607143	0.625	0.589286
2	0.616071	0.625	0.607143
4	0.607143	0.553571	0.660714
6	0.625	0.553571	0.696429
8	0.633929	0.553571	0.714286
10	0.651786	0.553571	0.75
12	0.642857	0.535714	0.75
14	0.678571	0.535714	0.821429
16	0.696429	0.535714	0.857143
18	0.723214	0.535714	0.910714

Tabla 7 Valores de EM para PlanTL-GOB-ES/roberta-large-bne-sqac

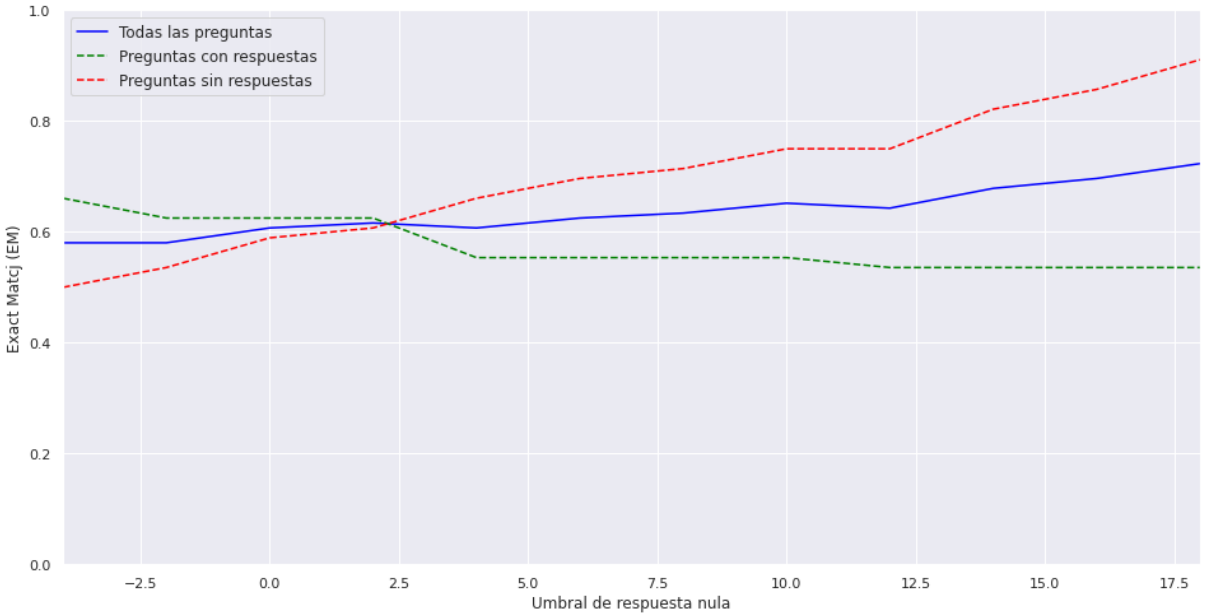


Ilustración 25 Variación de EM para PlanTL-GOB-ES/roberta-large-bne-sqac

F1

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.662466	0.824932	0.5
-2	0.663657	0.791599	0.535714
0	0.690442	0.791599	0.589286
2	0.693419	0.779694	0.607143
4	0.681757	0.702799	0.660714
6	0.697063	0.697697	0.696429
8	0.705991	0.697697	0.714286
10	0.723848	0.697697	0.75
12	0.71492	0.67984	0.75
14	0.750634	0.67984	0.821429
16	0.757777	0.658411	0.857143
18	0.766706	0.622697	0.910714

Tabla 8 Valores de F1 para PlanTL-GOB-ES/roberta-large-bne-sqac

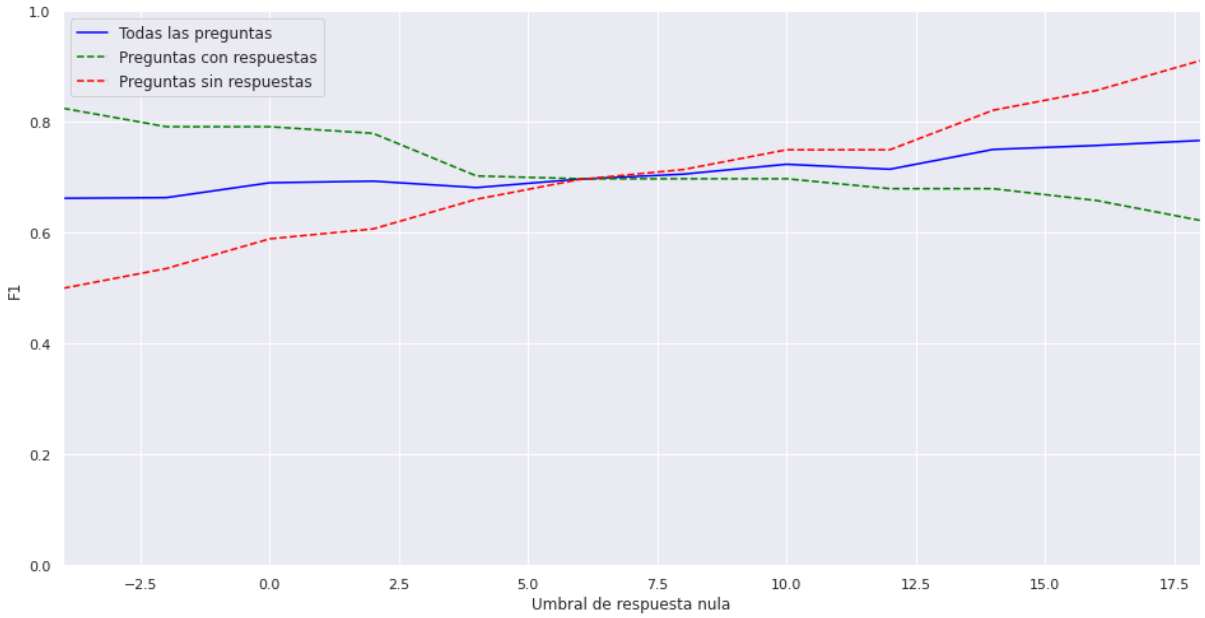


Ilustración 26 Variación de F1 para PlanTL-GOB-ES/roberta-large-bne-sqac

5.3.3. jamarju/roberta-base-bne-squad-2.0-es

EM

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.589286	0.589286	0.589286
-2	0.625	0.589286	0.660714
0	0.678571	0.589286	0.767857
2	0.705357	0.589286	0.821429
4	0.723214	0.589286	0.857143
6	0.714286	0.553571	0.875
8	0.705357	0.517857	0.892857
10	0.714286	0.482143	0.946429
12	0.660714	0.339286	0.982143
14	0.571429	0.142857	1
16	0.517857	0.035714	1
18	0.508929	0.017857	1

Tabla 9 Valores de EM para jamarju/roberta-base-bne-squad-2.0-es

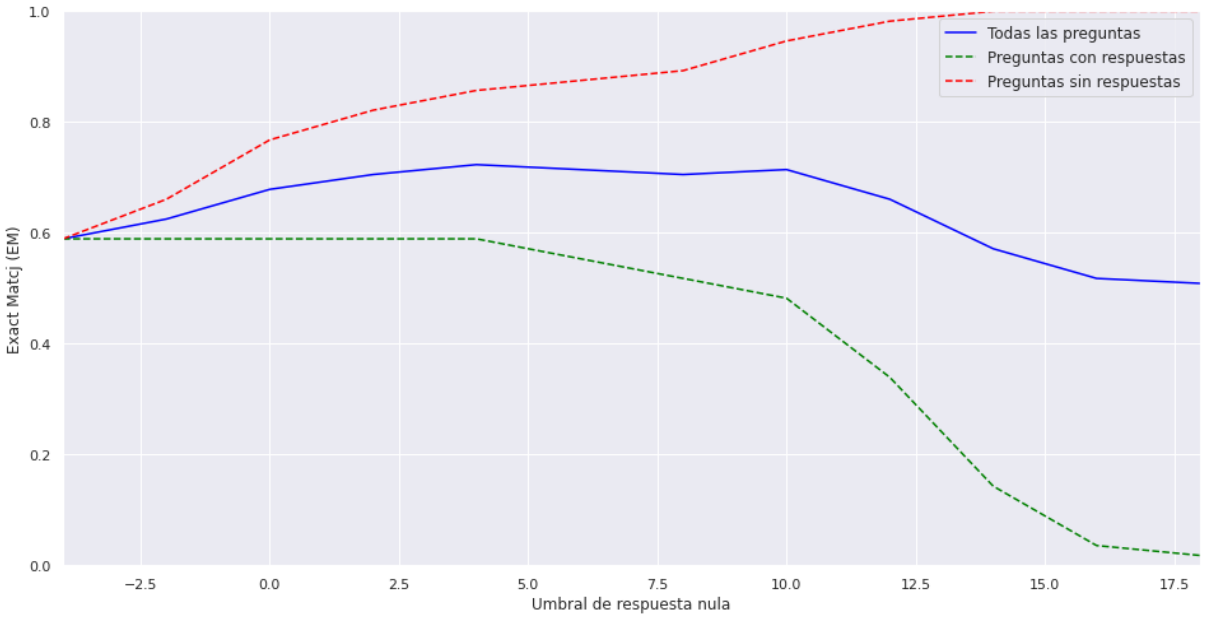


Ilustración 27 Variación de EM para jamarju/roberta-base-bne-squad-2.0-es

F1

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.589286	0.589286	0.589286
-2	0.625	0.589286	0.660714
0	0.678571	0.589286	0.767857
2	0.705357	0.589286	0.821429
4	0.723214	0.589286	0.857143
6	0.714286	0.553571	0.875
8	0.705357	0.517857	0.892857
10	0.714286	0.482143	0.946429
12	0.660714	0.339286	0.982143
14	0.571429	0.142857	1
16	0.517857	0.035714	1
18	0.508929	0.017857	1

Tabla 10 Valores de F1 para jamarju/roberta-base-bne-squad-2.0-es

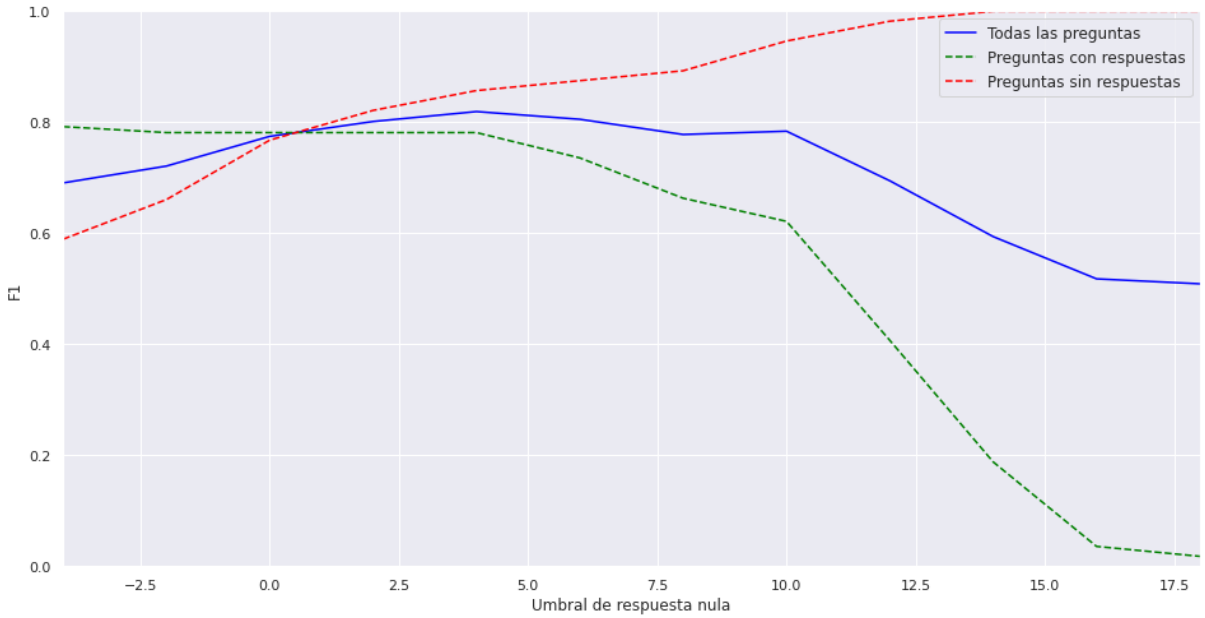


Ilustración 28 Variación de F1 para jamarju/roberta-base-bne-squad-2.0-es

5.3.4. jamarju/roberta-large-bne-squad-2.0-es

EM

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.392857	0.517857	0.267857
-2	0.508929	0.517857	0.5
0	0.571429	0.5	0.642857
2	0.651786	0.5	0.803571
4	0.660714	0.428571	0.892857
6	0.660714	0.428571	0.892857
8	0.660714	0.392857	0.928571
10	0.660714	0.375	0.946429
12	0.589286	0.232143	0.946429
14	0.553571	0.107143	1
16	0.508929	0.017857	1
18	0.5	0	1

Tabla 11 Valores de EM para jamarju/roberta-large-bne-squad-2.0-es

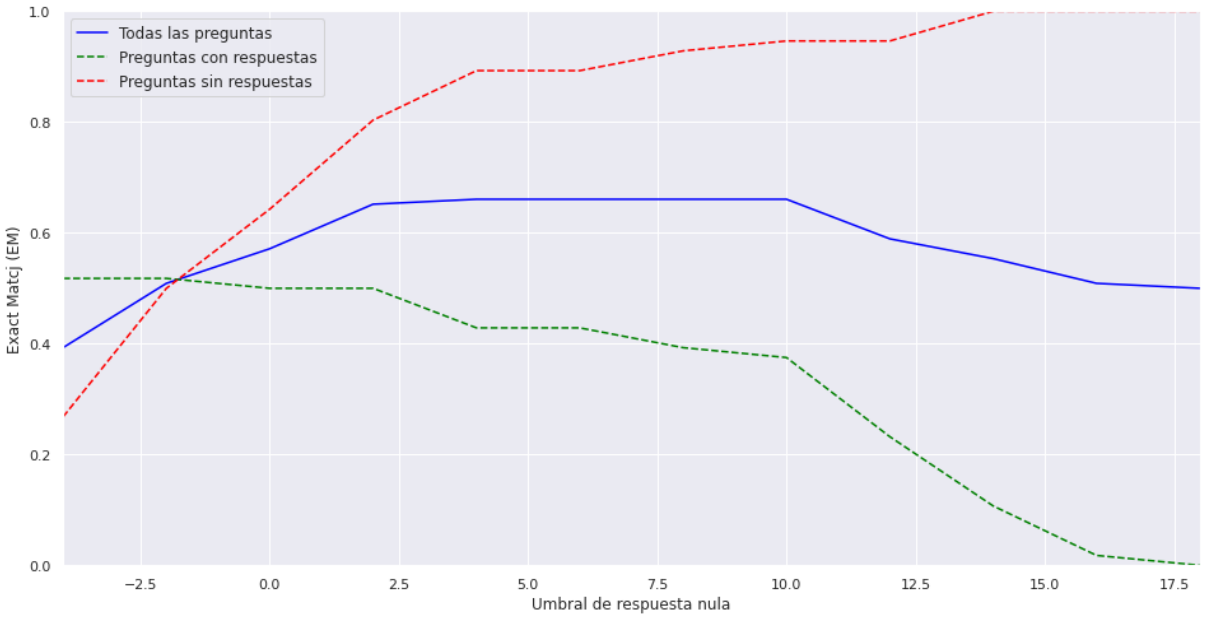


Ilustración 29 Variación de EM para jamarju/roberta-large-bne-squad-2.0-es

F1

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.533644	0.79943	0.267857
-2	0.649715	0.79943	0.5
0	0.704451	0.766045	0.642857
2	0.781291	0.759011	0.803571
4	0.79022	0.687582	0.892857
6	0.79022	0.687582	0.892857
8	0.766159	0.603748	0.928571
10	0.755163	0.563898	0.946429
12	0.641094	0.335759	0.946429
14	0.570473	0.140946	1
16	0.51717	0.034341	1
18	0.5	0	1

Tabla 12 Valores de F1 para jamarju/roberta-large-bne-squad-2.0-es

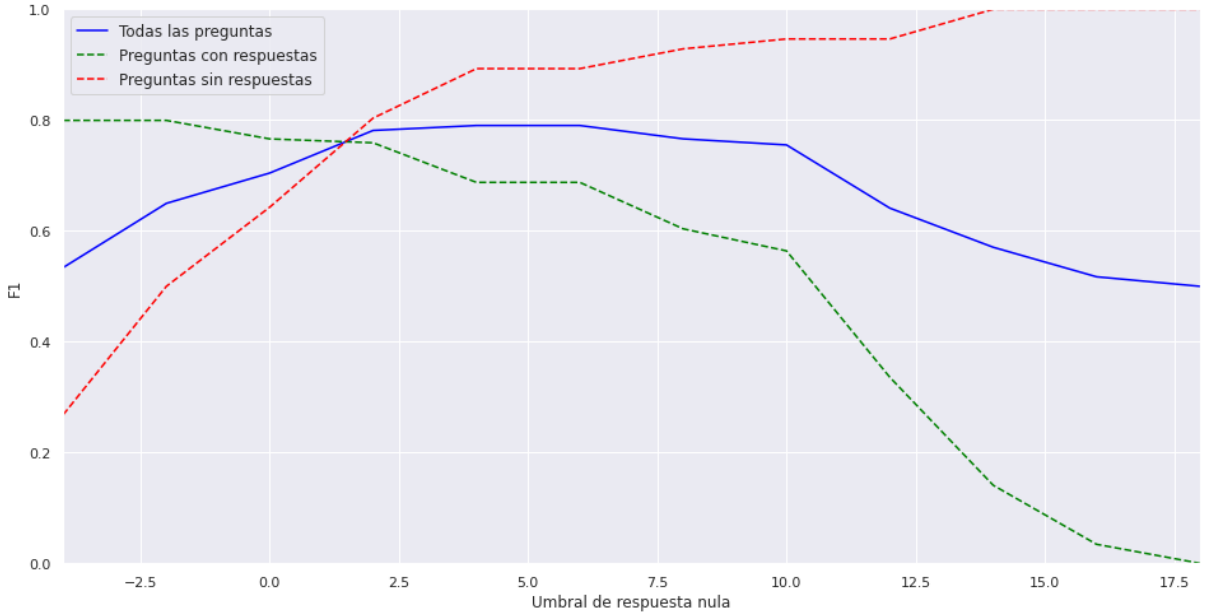


Ilustración 30 Variación de F1 para jamarju/roberta-large-bne-squad-2.0-es

5.3.5. mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

EM

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.080357	0.160714	0
-2	0.080357	0.160714	0
0	0.142857	0.160714	0.125
2	0.321429	0.160714	0.482143
4	0.428571	0.142857	0.714286
6	0.517857	0.142857	0.892857
8	0.544643	0.142857	0.946429
10	0.544643	0.089286	1
12	0.5	0	1
14	0.5	0	1
16	0.5	0	1
18	0.5	0	1

Tabla 13 Valores de EM para mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

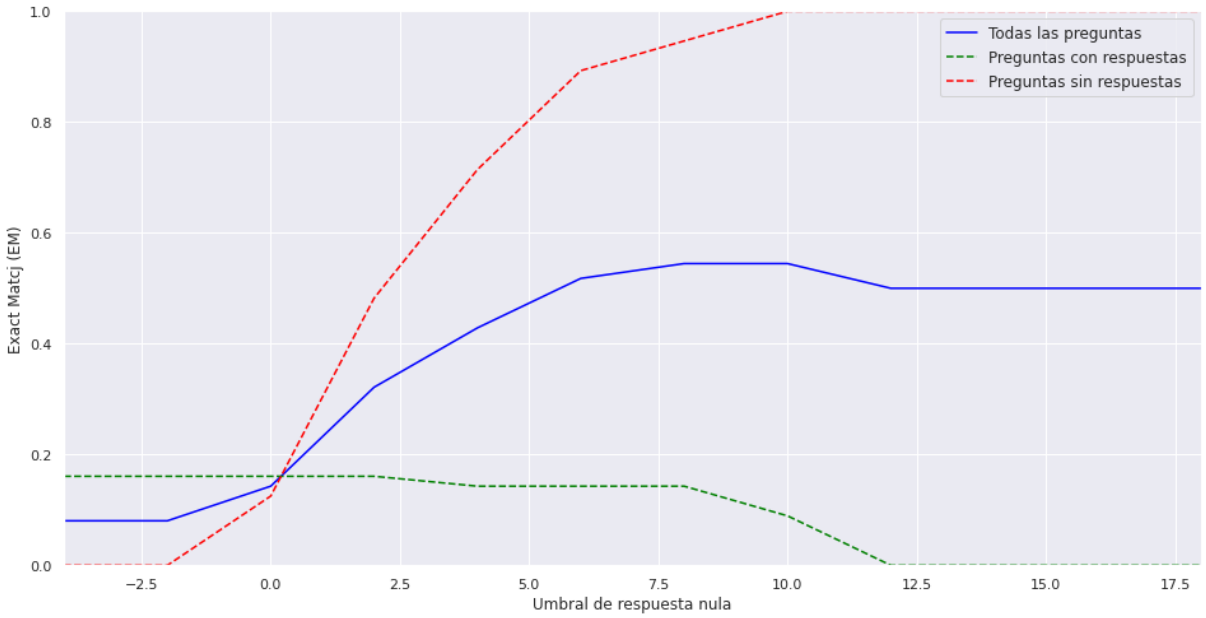


Ilustración 31 Variación de EM para mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

F1

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.198847	0.397693	0
-2	0.198847	0.397693	0
0	0.261347	0.397693	0.125
2	0.416489	0.350835	0.482143
4	0.51199	0.309694	0.714286
6	0.585006	0.277155	0.892857
8	0.596202	0.245975	0.946429
10	0.571188	0.142376	1
12	0.5	0	1
14	0.5	0	1
16	0.5	0	1
18	0.5	0	1

Tabla 14 Valores de F1 para mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

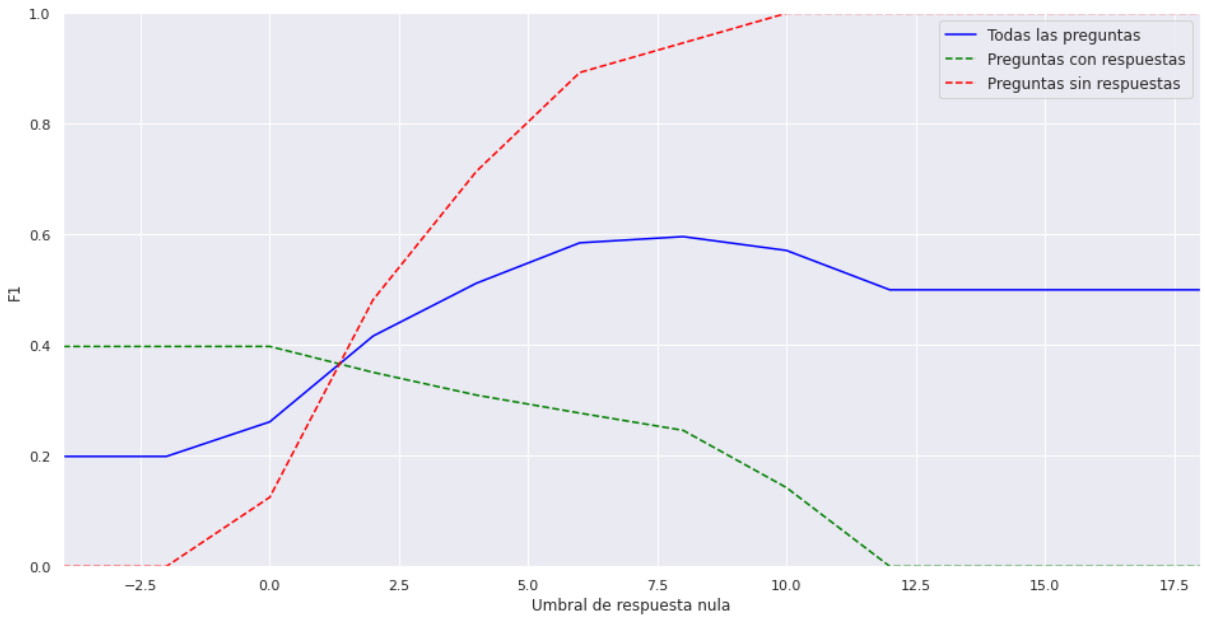


Ilustración 32 Variación de F1 para mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

5.3.6. MarcBrun/ixambert-finetuned-squad

EM

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.473214	0.482143	0.464286
-2	0.5625	0.464286	0.660714
0	0.607143	0.464286	0.75
2	0.598214	0.446429	0.75
4	0.607143	0.392857	0.821429
6	0.616071	0.321429	0.910714
8	0.607143	0.285714	0.928571
10	0.598214	0.25	0.946429
12	0.607143	0.214286	1
14	0.589286	0.178571	1
16	0.553571	0.107143	1
18	0.535714	0.071429	1

Tabla 15 Valores de EM para MarcBrun/ixambert-finetuned-squad

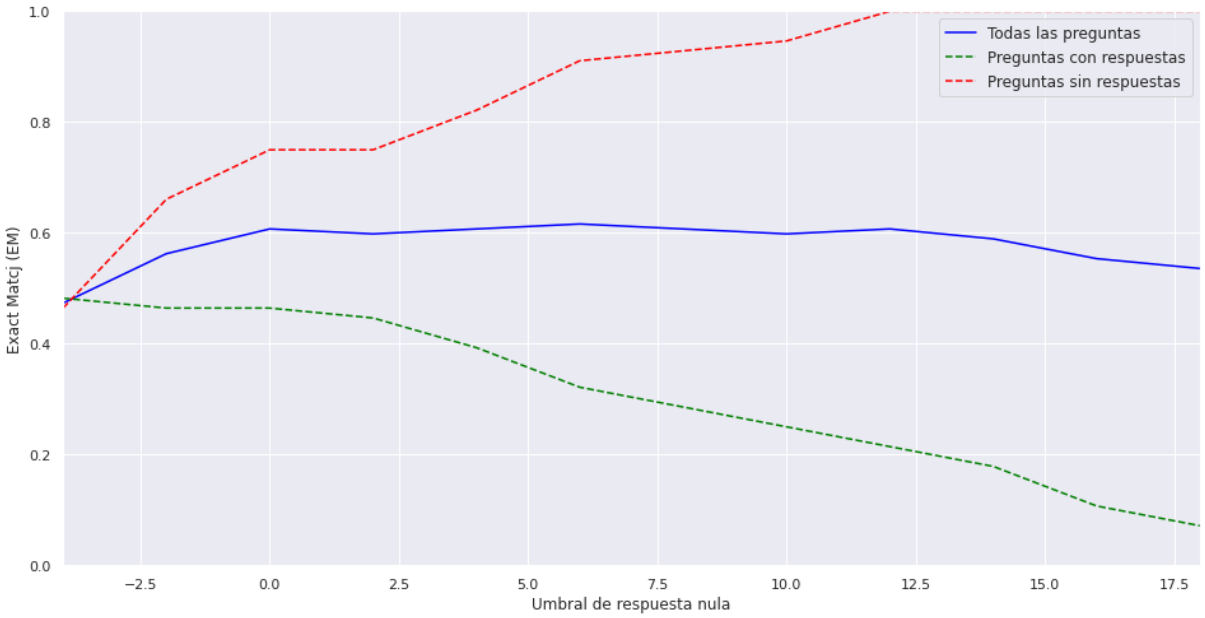


Ilustración 33 Variación de EM para MarcBrun/ixambert-finetuned-squad

F1

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.567432	0.670579	0.464286
-2	0.638592	0.616471	0.660714
0	0.683235	0.616471	0.75
2	0.669307	0.588615	0.75
4	0.665034	0.50864	0.821429
6	0.672177	0.43364	0.910714
8	0.663249	0.397926	0.928571
10	0.655213	0.363997	0.946429
12	0.664142	0.328283	1
14	0.63054	0.26108	1
16	0.580157	0.160315	1
18	0.558242	0.116484	1

Tabla 16 Valores de F1 para MarcBrun/ixambert-finetuned-squad

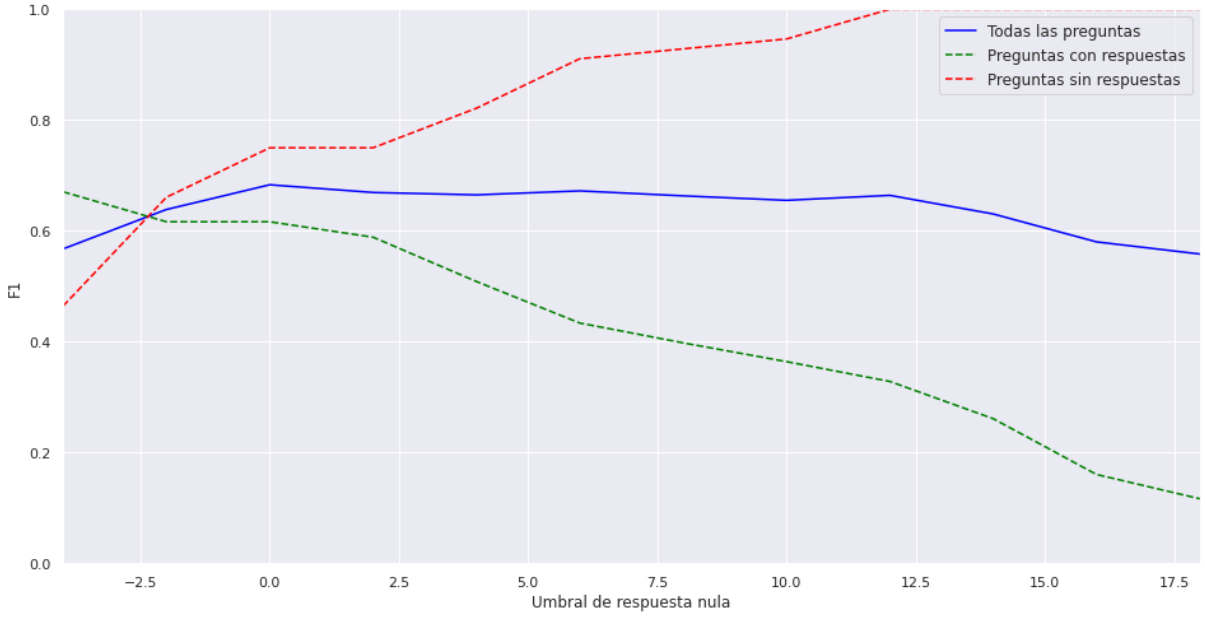


Ilustración 34 Variación de F1 para MarcBrun/ixambert-finetuned-squad

5.3.7. Modelo entrenado

Este modelo ha sido entrenado usando PlanTL-GOB-ES/roberta-large-bne-sqac como modelo base.

EM

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.580357	0.660714	0.5
-2	0.580357	0.625	0.535714
0	0.607143	0.625	0.589286
2	0.616071	0.625	0.607143
4	0.607143	0.553571	0.660714
6	0.625	0.553571	0.696429
8	0.633929	0.553571	0.714286
10	0.651786	0.553571	0.75
12	0.642857	0.535714	0.75
14	0.678571	0.535714	0.821429
16	0.696429	0.535714	0.857143
18	0.723214	0.535714	0.910714

Tabla 17 Valores de EM para modelo entrenado

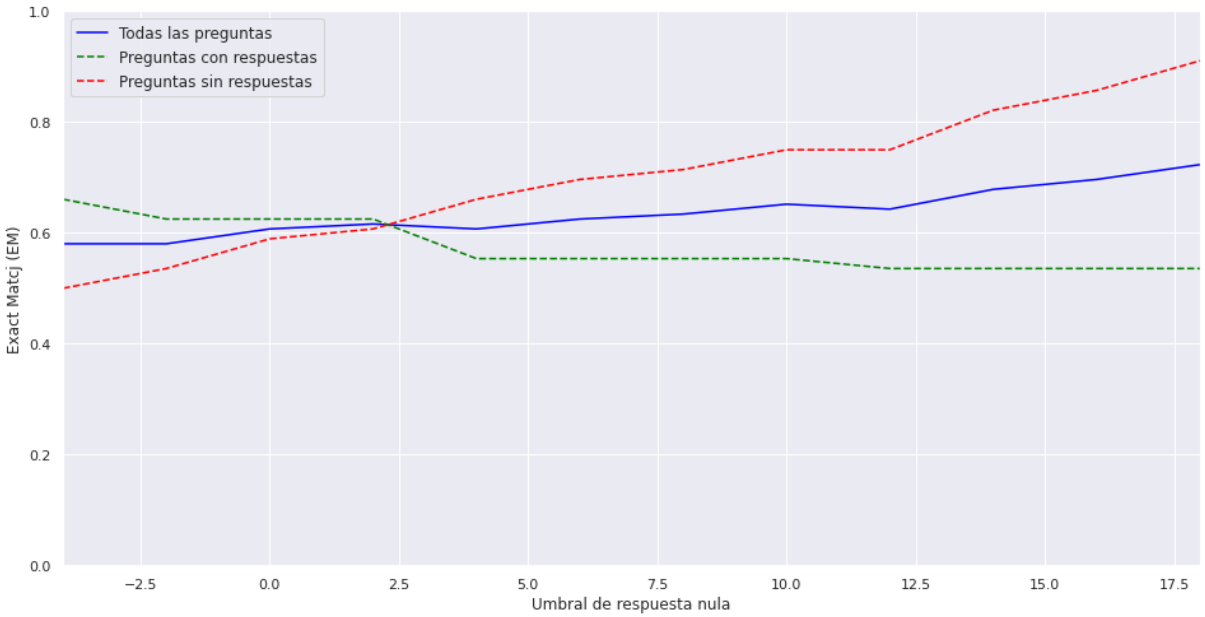


Ilustración 35 Variación de EM para modelo entrenado

F1

Threshold	Todas las preguntas	Preguntas con respuestas	Preguntas sin respuestas
-4	0.662466	0.824932	0.5
-2	0.663657	0.791599	0.535714
0	0.690442	0.791599	0.589286
2	0.693419	0.779694	0.607143
4	0.681757	0.702799	0.660714
6	0.697063	0.697697	0.696429
8	0.705991	0.697697	0.714286
10	0.723848	0.697697	0.75
12	0.71492	0.67984	0.75
14	0.750634	0.67984	0.821429
16	0.757777	0.658411	0.857143
18	0.766706	0.622697	0.910714

Tabla 18 Valores de F1 para modelo entrenado

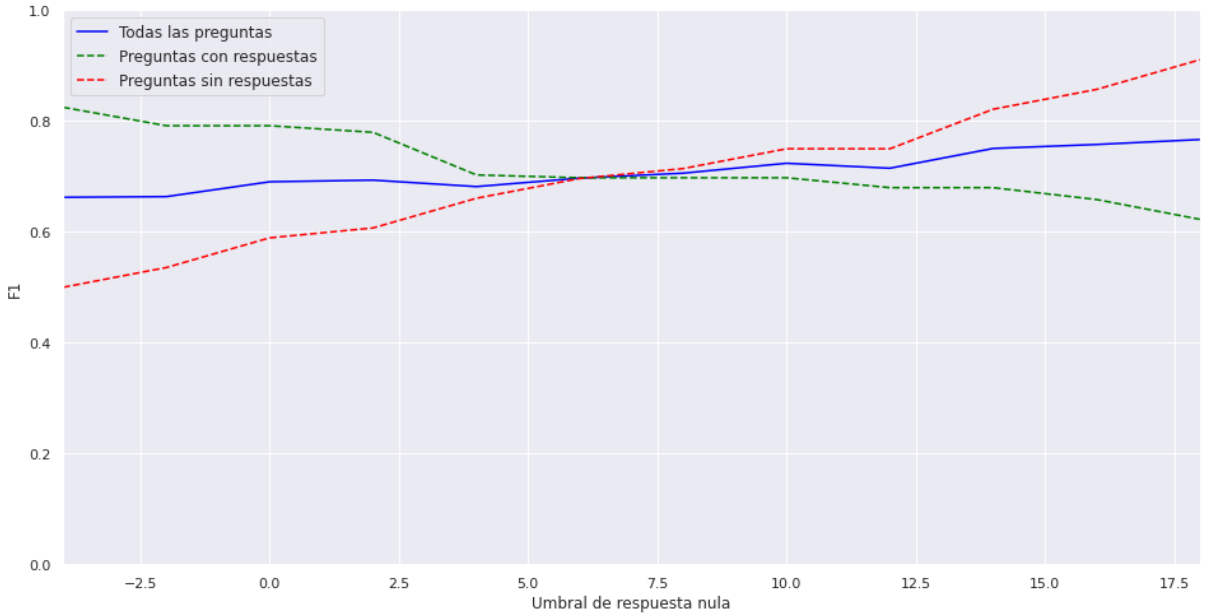


Ilustración 36 Variación de F1 para modelo entrenado

Para el entrenamiento de este modelo se ha utilizado un conjunto de datos formado por 70 preguntas. Se han entrenado durante 8 épocas utilizando lotes de tamaño 4.

6. Discusión y análisis de resultados

Antes de adentrarnos en la discusión y comparación de los diferentes modelos que se

El motivo por el que he decidido seleccionar esos modelos como parte del estudio es por su significado. Hemos comparado modelos basados en diferentes arquitecturas, todas ellas basadas en BERT. Además, en el caso de los modelos basados en RoBERTa se puede observar que hemos

6.1. Umbrales de respuesta nula

Tal y como se ha explicado en apartados anteriores, el uso del umbral de respuesta nula nos permite identificar cuando nuestro modelo debe devolver la cadena vacía en lugar de una posible respuesta.

Para el caso de nuestro estudio, un sistema que nos permite evaluar las respuestas de los alumnos a preguntas de una asignatura, podríamos pensar que encontrar el umbral de respuesta vacío se trata de una tarea prescindible, puesto que todas las preguntas han de tener una respuesta. Sin embargo, esto no es así, puesto que si únicamente nos fijamos en que el sistema nos devuelva respuestas podríamos construir un sistema que puede devolver respuestas sin sentido.

Si nos fijamos en cualquiera de las ilustraciones 23-34, observamos como a medida que aumentamos el umbral las líneas de los grupos “Preguntas con respuestas” y “Preguntas sin respuestas” comienzan a divergir. La línea que se corresponde con “Preguntas sin respuestas” tiende a alcanzar el valor máximo (1) tanto para la métrica EM como para la métrica F1, mientras que la línea que se corresponde con el grupo de Preguntas con respuestas tiende a 0. Esto ocurre por el hecho de que somos más exigentes con el mínimo puntaje requerido para muestras predicciones.

En cualquiera de las ilustraciones anteriormente citadas podemos observar un punto de corte, en el se cruzan las 3 líneas. Este punto ocurre en un valor diferente del umbral dependiendo del modelo que estamos mostrando, e incluso dependiendo de la métrica en concreto. No debemos olvidar que para el cálculo de las métricas mostradas se ha utilizado el mismo conjunto de datos, y por lo tanto lo que estamos observando es que para cada modelo y nuestro conjunto de datos tendremos un valor de umbral óptimo diferente. Por lo tanto, las conclusiones obtenidas en este estudio no se pueden extrapolar a otros conjuntos de datos.

Es importante recordar que nuestro dataset está compuesto por el mismo número de preguntas con respuestas quede preguntas sin respuestas. Esto tiene implicaciones que pueden hacer que las métricas sean engañosas. Por ejemplo, si usamos un valor de umbral muy elevado, el modelo considerará que no hay ninguna respuesta lo suficientemente válida para ninguna pregunta y aún así

el valor de F1 y de EM será del 0.5 (como poco) porque al menos habrá predicho con exactitud los casos donde no es posible encontrar una respuesta válida.

Por ejemplo, si nos fijamos en la ilustración 34 dónde se muestra los valores de la métrica F1 para el modelo MarcBrun/ixambert-finetuned-squad, podríamos pensar que el valor óptimo del umbral es cualquier valor comprendido entre 0 y 12 puesto que el valor de la métrica se mantiene constante en torno a F1=0.65 para todos estos valores. Ese valor se mantiene constante única y exclusivamente porque el modelo está prediciendo con éxito la amplia mayoría de los casos negativos (preguntas sin respuestas). Sin embargo, si usamos umbrales próximos a 12 el modelo no sería capaz de predecir correctamente preguntas que si tienen respuestas.

Devlin, Chang et al. (2019) dicen que se predice una respuesta no nula cuando sˆi,j > s_null + τ , donde el umbral τ se selecciona en el conjunto de desarrollo para maximizar F1. El método squad_evaluate del paquete data.metrics.squad_metrics de la librería de Transformers[56] nos proporciona las métricas para e conjunto de datos proporcionado, pero además cual sería el valor de umbral óptimo para maximizar los valores de EM y F1.

Mi pensamiento al respecto, a cerca de cual sería el valor umbral óptimo que deberíamos utilizar para cada modelo, es que no se puede basar únicamente en el valor que nos proporciona el mejor resultado. De hecho, considero que el valor optimo que se debería utilizar está cercano al punto de corte de las 3 líneas representadas en las ilustraciones anteriores. Además, independientemente del punto de corte, considero que se deben considerar otros factores, como es el caso de la pendiente de la línea que se corresponde con el conjunto de datos formado por las preguntas que si tienen respuestas. Por ejemplo, en la ilustración 28, que muestra los valores de F1 para el modelo jamarju/roberta-base-bne-squad-2.0-es, considero que el valor óptimo del umbral sería próximo a 4, porque para ese valor se ha mantenido el valor de la métrica que se obtuvo para el uso de umbral de 2, a la par que se ha aumentado el valor de la métrica para el conjunto total de todas las preguntas.

6.2. Evaluación de métricas EM y F1

Teniendo en cuenta los valores obtenidos para las métricas F1 y EM en los modelos estudiados se puede confirmar que ambas tienen un comportamiento similar. De hecho, si nos fijamos en las ilustraciones 23-34 y prestamos atención a los valores utilizados como umbral de respuesta vacía, que están representados en el eje X, se observa como la gráfica que representa la métrica F1 podría considerarse como una representación desplazada hacia la derecha de la gráfica que muestra los valores de la métrica EM para un mismo modelo.

[56] https://github.com/huggingface/transformers/blob/master/src/transformers/data/metrics/squad_metrics.py

Lo citado en el párrafo anterior tiene sentido, puesto que la métrica EM es más estricta que la métrica F1. Como consecuencia de esto, los valores de la métrica EM empiezan a tener una tendencia descendente antes que para los valores de la métrica F1. Esto provoca que el punto de corte ocurra para valores de umbrales menores en el caso de la métrica EM.

Para el caso del conjunto de datos formado por preguntas negativas, es decir, sin respuestas, los valores de EM y F1 serán exactamente iguales, puesto que no hay posibilidad de estar correcta en un porcentaje de la respuesta.

Para un entendimiento de estas métricas es importante tener en cuenta el tipo de preguntas que componen nuestro conjunto de datos. En nuestro caso, para entender los resultados obtenidos, es fundamental que observemos la tabla 3, que muestra cuantas preguntas hay para un número de respuestas en concreto y la tabla 4 que nos muestra las longitudes y número de tokens medio para las respuestas.

En caso de tener preguntas con muchas posibles respuestas, los valores de la métrica F1, y probablemente la métrica EM, serán más elevados, ya que el valor de la métrica se obtiene calculando el valor máximo de la predicción para las posibles respuestas validas. Por lo tanto, tener más posibilidades de acierto implica tener más posibilidades de tener un valor elevado de las métricas.

Por otro lado, el tipo de pregunta que forman nuestro sistema, o más bien, el tipo de respuesta, condicionará no sólo el valor de las métricas sino también que métrica resulta más importante para nuestro caso. En nuestro caso, dónde la amplia mayoría de las preguntas cuenta con una o varias frases, el uso de la métrica F1 aportará más valor a la hora de decidir que modelo es el más adecuado. Sin embargo, si quisiéramos construir un sistema que respondiera a preguntas factoides del tipo ¿En que año se invento la maquina de Turing?, ¿Cuál es el nombre completo del jugador de ajedrez vencido por Deep Blue? ó ¿En que mes tiene lugar el evento del TRec? Deberíamos fijarnos en los valores obtenidos por la métrica EM.

6.3. Comparación de modelos

No es una tarea sencilla decantarse por el modelo más adecuado para nuestro sistema teniendo en cuenta el tamaño reducido del conjunto de datos utilizado Sin embargo, teniendo en cuenta estas limitaciones, me atrevería a decir que el modelo más adecuado para nuestro sistema, sería aquel dónde, fijándonos en las ilustraciones que muestran los valores de la métrica EM, la intersección de las líneas se produzca en un valor del eje Y (valor de la métrica) más elevado. Del mismo modo, deberíamos fijarnos en el eje X (valor de umbral utilizado) para utilizarlo como valor.

Siguiendo los criterios establecidos en el apartado anterior, los modelos que mejores resultados nos darían para la construcción de nuestro sistema sería cualquiera de los siguientes 2 modelos: jamarju/roberta-base-bne-squad-2.0-es jamarju/roberta-large-bne-squad-2.0-es. Curiosamente el modelo entrenado a partir de la arquitectura reducida de RoBERTa obtiene unas métricas ligeramente superiores a las obtenidas que el modelo entrenado a partir de la versión con más parámetros de RoBERTa

Con respecto al modelo entenado por mi, como parte del estudio, no se aprecia ninguna mejora sobre el modelo base, como era de esperar.. Esto es lógico, ya que se ha entrenado con un número muy reducido de preguntas. De hecho, aunque se hayan utilizado un subconjunto de las preguntas que se utilizan para evaluar los modelos, los resultados obtenidos no han variado absolutamente nada. Nunca se debe hacer eso, es decir, los datos utilizados para evaluar un modelo no deberían ser parte del entrenamiento, pero en este caso, debido al tamaño del dataset y la presuposición de los datos obtenidos, no he considerado que fuera a afectar en los resultados del estudio.

7. Conclusiones y trabajo futuro

7.1. Conclusiones

El estudio comparativo realizado en este trabajo me ha permitido no sólo profundizar en el estado del arte de los sistemas de búsqueda de respuestas sino establecer un contacto directo con las tecnologías que estas ocupando la primera línea de la IA, es decir, las arquitecturas basadas en mecanismos de atención, conocidas como Transformers.

En cuanto al problema en estudio, para el que se quería encontrar la solución más adecuada he de reconocer que con las limitaciones temporales, obvias de un proyecto de fin de máster, y las limitaciones de recursos, no se puede descartar la opción de entrenar un modelo propio, en lugar de utilizar directamente uno ya entrenado. Poder entrenar un modelo propio requiere de sistemas de computo mejores que los utilizados para el desarrollo de este trabajo. Al fin y al acabo lo que he utilizado han sido instancias de pago por cuota mensual ofrecidas por Google, pero para un trabajo de investigación con mayor alcance se deberían considerar otras alternativas

Por otro lado, se puede confirmar que el camino a seguir para la construcción de un sistema de búsqueda de respuestas para el ámbito académico, y más concretamente para la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento, debe apoyarse en el uso de los Transformers. En concreto, los resultados obtenidos por los modelos basados en RoBERTa han sido los mejores para el conjunto de datos que ha formado parte del estudio.

Más allá del estado de este trabajo, y de los resultados obtenidos, me gustaría destacar que gracias a los conocimientos adquiridos me han permitido profundizar en el uso de las tecnologías basadas en Transformers para otros usos. Como prueba de ello me gustaría compartir como parte de este trabajo la publicación de Transformers for Natural Language Processing (Corrales 2021)[57], con un formato no académico, en el cual hablo tanto de los beneficios de una arquitectura basada en Transformers como del uso de la misma para la clasificación de textos según su contenido, es decir, otro área del PLN dónde los Transformer se han convertido en el estado del arte.

[57] Corrales, I., 2022. Transformers for Natural Language Processing (NLP). [Blog] *Mediium*, Available at: <https://ivan-corrales-solera.medium.com/b5c47aee4d65> [Accessed 1 February 2022].

7.2. Líneas de trabajo futuro

Tras haber completado este trabajo se me ocurren las siguientes ideas que abren nuevas líneas de trabajo futuros. He de reconocer, que estas ideas son muy diferentes de las que tenía antes de empezar a trabajar en este estudio.

Elaboración de dataset académico en español

Aportaría gran valor la elaboración de un dataset con un formato basado en SQUAD v2 . La versión española de SQUADv2 se ha creado a partir del uso de traducciones automáticas. Esto no digo que sea una mala aproximación para tener un gran corpus con el que poder entrenar nuestros modelos, pero creo que sería muy interesante poder desarrollar un corpus académico en idioma español. Este dataset podría especializarse en preguntas relacionadas con el área de la Ingeniería Informáticas y más concretamente con el área de Inteligencia Artificial.

Métricas de detección de umbrales de respuesta óptima nulos

Tras haber realizado la evaluación de los modelos y haber intentado resolver la pregunta inicial, cual de los modelos es el que mejor se adecúa a nuestras necesidades, creo que no es una respuesta sencilla, y que los cálculos utilizados en la actualidad para obtener los valores óptimos de umbrales no son lo suficientemente potentes. Se podría trabajar en el desarrollo de técnicas que permitan encontrar el uso óptimo de de los valores de umbrales a utilizar para un modelo.

Nuevo formato para datasets

Considero que el uso del formato JSON para el almacenamiento de datasets hace que estos ficheros tengan un tamaño muy elevado de lo necesario. A pesar de que JSON reduce el tamaño de los ficheros, si lo comparamos con los documentos de lenguaje de marcado, XML[58], creo que la solución más adecuada debería apostar por el diseño de un propio DSL, del inglés Domain Specific Language, y en español conocido como Lenguaje específico de dominio.

[58] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (1997). Extensible markup language (XML). World Wide Web Journal, 2(4), 27-66.

8. Bibliografía

Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology BT - Artificial Intelligence Applications and Innovations (I. Maglogiannis, L. Iliadis, & E. Pimenidis (eds.); pp. 373–383). Springer International Publishing.

Agarwal, A., Sachdeva, N., Yadav, R. K., Udandarao, V., Mittal, V., Gupta, A., & Mathur, A. (2019). EDUQA: Educational Domain Question Answering System Using Conceptual Network Mapping. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2019-May, 8137–8141. https://doi.org/10.1109/ICASSP.2019.8683538

Calijorne Soares, M. A., & Parreiras, F. S. (2020). A literature review on question answering techniques, paradigms and systems. Journal of King Saud University - Computer and Information Sciences, 32(6), 635–646. https://doi.org/10.1016/j.jksuci.2018.08.005

Cardoso, A. C., Bini, A., & Pérez Abelleira, M. A. (2015). Diseño de un sistema de búsqueda de respuestas para diversos tipos de preguntas. Simposio Argentino de Inteligencia Artificial, 44° Jornadas Argentinas de Informática (JAIIO), 26. http://sedici.uPLN.edu.ar/bitstream/handle/10915/52018/Documento_completo.pdf-PDFA.pdf?sequence=1

Eduardo Ruiz de Pascual Núñez. (2016). Sistema de búsqueda de respuestas sobre DBpedia. Trabajo Fin de Grado. http://www.fnb.upc.edu/content/treballs-fi-de-grau-i-màster

ElKafrawy, P. M., Sauber, A. M., & Sabry, N. A. (2018). Semantic question answering system using dbpedia. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10868 LNAI (Issue January). Springer International Publishing. https://doi.org/10.1007/978-3-319-92058-0_79

Ferr, S. (2008). Un sistema de búsqueda de respuestas basado en ontologías, implicación textual y entornos reales. Procesamiento de Lenguaje Natural, 41(41), 47–54.

Green, B. F., Wolf, A. K., Chomsky, C., & Laughery, K. (1961). Baseball: An automatic question-answerer. Proceedings of the Western Joint Computer Conference: Extending Man’s Intellect, IRE-AIEE-ACM 1961, 219–224. https://doi.org/10.1145/1460690.1460714

Kaisser, M. (2008). The QuALiM question answering demo: supplementing answers with paragraphs drawn from Wikipedia. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies Demo Session - HLT ’08, June, 32–35. https://doi.org/10.3115/1564144.1564153

Kumar, P., Kumar Goel, R., & Sagar Sharma, P. (2014). A New Architecture of Automatic Question Answering System using Ontology. International Journal of Computer Applications, 97(20), 1–4. https://doi.org/10.5120/17120-7671

Li, X., Grandvalet, Y., Davoine, F., Cheng, J., Cui, Y., Zhang, H., Belongie, S., Tsai, Y. H., & Yang, M. H. (2020). Transfer learning in computer vision tasks: Remember where you come from. Image and Vision Computing, 93. https://doi.org/10.1016/j.imavis.2019.103853

Li, Y. (2018). Two layers LSTM with attention for multi-choice question answering in exams. IOP Conference Series: Materials Science and Engineering, 323(1). https://doi.org/10.1088/1757-899X/323/1/012023

Liddle, S., Schewe, K., & Zhou, X. (2012). Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface. In Lecture Notes in Computer Science (Vol. 7446). https://doi.org/10.1007/978-3-642-32600-4

Lopez, V., Fernández, M., Motta, E., & Stieler, N. (2012). PowerAqua: Supporting users in querying and exploring the Semantic Web. Semantic Web, 3(3), 249–265. https://doi.org/10.3233/SW-2011-0030

M., A., S.A., K., & Namrata, C. (2016). Question Answering System, Approaches and Techniques: A Review. International Journal of Computer Applications, 141(3), 34–39. https://doi.org/10.5120/ijca2016909587

Martínez-barco, P., Vicedo, J. L., Saquete, E., & Tomás, D. (2007). Sistemas de Pregunta-Respuesta. 1–19.

Moreno, O., Directora, S., & Díaz Fernández, J. (2016). Universidad Politécnica De Madrid Trabajo Fin De Máster.

Ojokoh, B., & Adebisi, E. (2019). A review of question answering systems. Journal of Web Engineering, 17(8), 717–758. https://doi.org/10.13052/jwe1540-9589.1785

Olvera-Lobo, M. D., & Gutiérrez-Artacho, J. (2013). Evaluación del rendimiento de los sistemas de búsqueda de respuestas de dominio general. Revista Espanola de Documentacion Cientifica, 36(2). https://doi.org/10.3989/redc.2013.2.921

P.M, A., M, S., & P.C, R. (2013). Architecture of an Ontology-Based Domain-Specific Natural Language Question Answering System. International Journal of Web & Semantic Technology, 4(4), 31–39. https://doi.org/10.5121/ijwest.2013.4403

Poonguzhali, R., & Lakshmi, K. (2020). Evaluating the performance of recurrent neural network based question answering system with easy and complex bAbI QA tasks. International Journal of Advanced Science and Technology, 29(5 Special Issue), 1389–1402.

Reddy, A. C. O., & Madhavi, K. (2017). A Survey on Types of Question Answering System. 19(6), 19–23. https://doi.org/10.9790/0661-1906041923

Reddy, A. C. O., & Madhavi, K. (2020). Convolutional recurrent neural network with template based representation for complex question answering. International Journal of Electrical and Computer Engineering, 10(3), 2710–2718. https://doi.org/10.11591/ijece.v10i3.pp2710-2718

Ruder, S., Peters, M., Swayamdipta, S., & Wolf, T. (2019). Transfer Learning in NLP. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, 2010, 15–18. https://aclanthology.org/N19-5004/

Sasikumar, U., & L, S. (2014). A Survey of Natural Language Question Answering System. International Journal of Computer Applications, 108(15), 42–46. https://doi.org/10.5120/18991-0444

Sharma, Y., & Gupta, S. (2018). Deep Learning Approaches for Question Answering System. Procedia Computer Science, 132, 785–794. https://doi.org/10.1016/j.procs.2018.05.090

Vicedo, J. L. (2004). La Búsqueda de Respuestas: Estado Actual y Perspectivas de Futuro. Inteligencia Artificial, 8(22), 37–56. https://doi.org/10.4114/ia.v8i22.805

Vicedo, J.-L., Rodríguez Hontoria, H., Peñas Padilla, A., & Massot Bayés, M. (2003). Los sistemas de búsqueda de respuestas desde una perspectiva actual. Procesamiento Del Lenguaje Natural, 31(June 2014), 351–367.

Voorhees, E. M. (2001). The TREC question answering track*. Natural Language Engineering, 7(4), 361–378. https://doi.org/10.1017/S1351324901002789

Wahyudi, Khodra, M. L., Prihatmanto, A. S., & Machbub, C. (2018). A Question Answering System Using Graph-Pattern Association Rules (QAGPAR) on YAGO Knowledge Base. 2018 International Conference on Information Technology Systems and Innovation, ICITSI 2018 - Proceedings, 536–541. https://doi.org/10.1109/ICITSI.2018.8696046

Wan, B. W., & Qin, S. Y. (1992). Recent development in large-scale systems research in China. In IEEE International Symposium on Industrial Electronics (pp. 22–28). https://doi.org/10.1109/ISIE.1992.279627

Yin, J., Xin, J., Lu, Z., Shang, L., Li, H., & Li, X. (2016). Neural generative question answering. IJCAI International Joint Conference on Artificial Intelligence, 2016-January, 2972–2978. https://doi.org/10.18653/v1/w16-0106

Yuan, X., Côté, M. A., Fu, J., Lin, Z., Pal, C., Bengio, Y., & Trischler, A. (2020). Interactive language learning by question answering. EMPLN-IJCPLN 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, 2796–2813. https://doi.org/10.18653/v1/d19-1280

Zhang, B., Wang, H., Jiang, L., Yuan, S., & Li, M. (2020). A novel bidirectional LSTM and attention mechanism based neural network for answer selection in community question answering. Computers, Materials and Continua, 62(3), 1273–1288. https://doi.org/10.32604/cmc.2020.07269

Anexos

Anexo 1. Cálculo de métricas EM y F1 con Python

```python
import string
import re

re_articles = re.compile(r"\b(el|la|los|las|unos|unas|un|una)\b", re.UNICODE)

def normalize_text(text):
  # convert the input text to lowercase format
  text = text.lower()
  # remove the punction symbols from the text
  exclude = set(string.punctuation)
  text = "".join(ch for ch in text if ch not in exclude)
  # remove the articles
  text = re.sub(re_articles, " ", text)
  # remove the whitespaces from the text
  return " ".join(text.split())
```

```python
def calculate_exact_match(prediction, truth):
    return int(normalize_text(prediction) == normalize_text(truth))
```

```python
def calculate_f1(prediction, truth, verbose=False):
    pred_tokens = normalize_text(prediction).split()
    truth_tokens = normalize_text(truth).split()

    # if either the prediction or the truth is no-answer
    # then f1 = 1 if they agree, 0 otherwise
    if len(pred_tokens) == 0 or len(truth_tokens) == 0:
        return int(pred_tokens == truth_tokens)

    common_tokens = set(pred_tokens) & set(truth_tokens)

    # if there are no common tokens then f1 = 0
    if len(common_tokens) == 0:
        return 0
    prec = len(common_tokens) / len(pred_tokens)
    rec = len(common_tokens) / len(truth_tokens)
    if verbose:
      print(f'\nRespuesta esperada: {truth}')
      print(f'Respuesta predicha: {prediction}')
      print(f'Nº tokens coincidentes: {len(common_tokens)}')
      print(f'Nº tokens en respuesta predicha: {len(pred_tokens)}')
      print(f'Nº tokens en respuesta esperada: {len(truth_tokens)}')
      print(f'precisión: {prec}, recall: {rec}')
      print(f'F1: {2 * (prec * rec) / (prec + rec)}')
    return 2 * (prec * rec) / (prec + rec)
```

Anexo 2. Código Python para predicción de múltiples respuestas

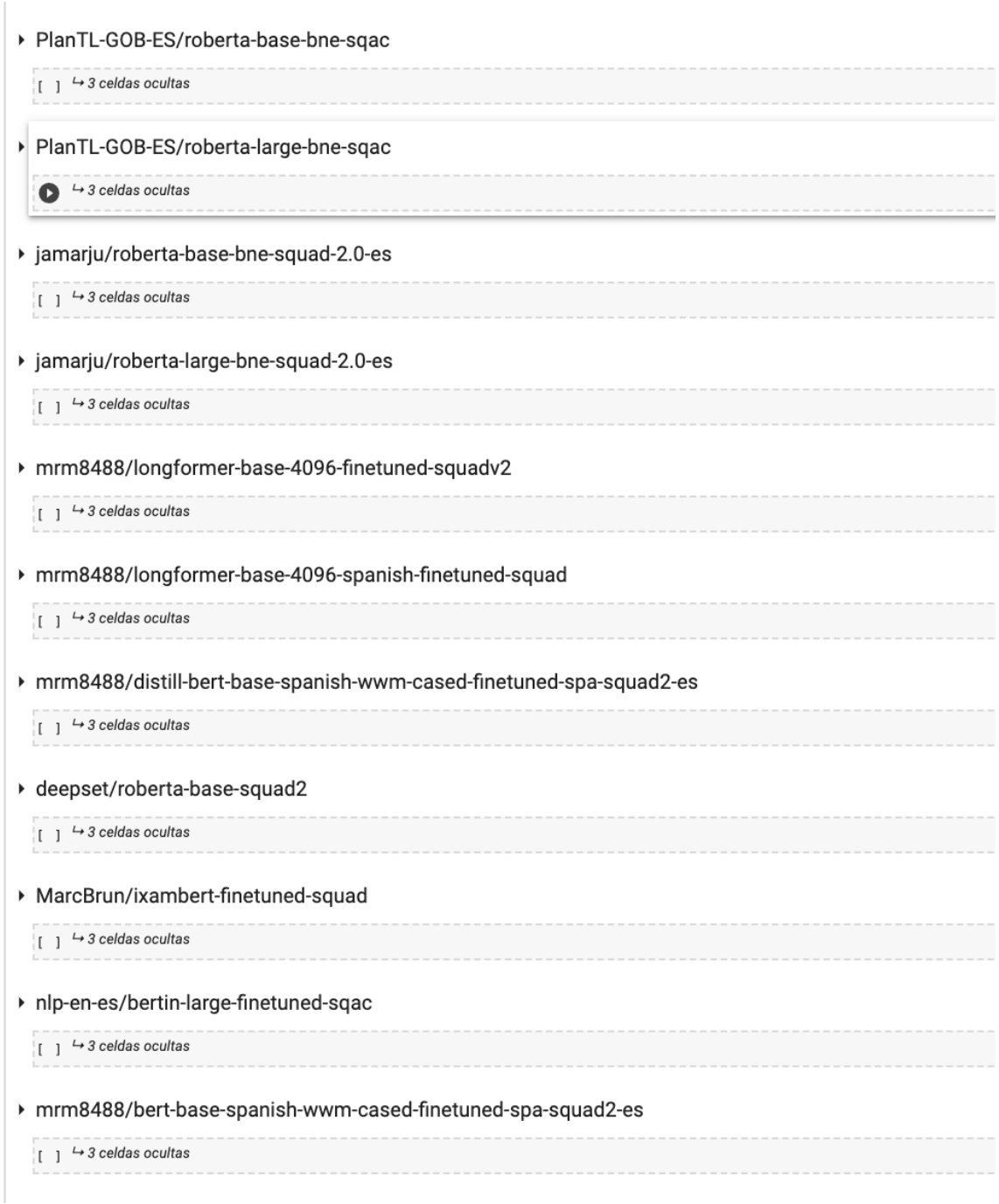
```python
import operator

def best_predictions(example, tokenizer, model, n_predictions=5, threshold=1.0,
                     max_length=512, stride=128):
  predictions = []
  inputs = tokenizer(
      example.question_text,
      example.context_text,
      max_length                = max_length,
      stride                    = stride,
      add_special_tokens        = True,
      return_overflowing_tokens = True,
      return_offsets_mapping    = True,
      truncation                = 'only_second',
      padding                   = 'max_length',
      return_tensors            = "pt"
  )
  overflow_to_sample_mapping = inputs.pop("overflow_to_sample_mapping")
  offset_mapping = inputs.pop("offset_mapping").tolist()
  tokenized_context = [item for sublist in inputs['input_ids'].tolist() for item in sublist]
  sequence_ids = inputs.sequence_ids(0) * len(offset_mapping)
  outputs = model(**inputs)
  starts = [item for sublist in outputs[0] for item in sublist]
  ends   = [item for sublist in outputs[1] for item in sublist]
  score_null = starts[0] + ends[0]
  for start, start_value in enumerate(starts):
    if sequence_ids[start]==1:
      sub_ends = ends[start:]
      for end_idx, end_value in enumerate(sub_ends):
        end = start + end_idx
        if sequence_ids[end]==1:
          score = start_value + end_value
          if score > threshold:
            predictions.append({
              'score': score.item(),
              'tokens': tokenized_context[start:end+1],
            })
  predictions = list(reversed(sorted(
        predictions,
        key=operator.itemgetter("score"),
  )))[:n_predictions]
  for prediction in predictions:
    prediction['text'] = tokenizer.convert_tokens_to_string(
        tokenizer.convert_ids_to_tokens(prediction['tokens']))
  return predictions
```

Anexo 3. Código Python utilizado para generar dataset en formato SQUADv2 y ejemplo de fichero de entrada

```python
import uuid
import json
from os import walk
import os
from transformers import BertTokenizer
tokenizer = BertTokenizer.from_pretrained(
    "dccuchile/bert-base-spanish-wwm-uncased")

def generate_dataset(input):
  items = []
  for item in input:
    paragraphs=[]
    for paragraph in item['paragraphs']:
      qas=[]
      for question in paragraph['questions']:
        answers = []
        if 'answers' in question:
          for answer in question['answers']:
            answers.append({
                "text": answer,
                "answer_start": paragraph['context'].find(f'{answer}'),
            })
        qas.append({
            "id": f"{uuid.uuid4()}",
            "question": question['question'],
            "is_impossible": False,
            "answers": answers,
        })
      paragraphs.append({
          "context": paragraph['context'],
           "qas": qas
      })
    items.append({
      "title": item['title'],
      "paragraphs":paragraphs,
    })
  return { "version": "v2.0", "data": items,}
questions_path = '/Users/icorrales/My Drive/TFM/questions'
filenames = next(walk(questions_path), (None, None, []))[2]
items = []
for filename in filenames:
  with open(os.path.join(questions_path, filename), 'rb') as file:
    print(f'processing {filename} file...')
    samples = json.load(file)
    items.append(samples)

out = generate_dataset(items)
app_json = json.dumps(out, sort_keys=False, ensure_ascii=False).encode('utf8')
print(app_json.decode())
```

```
{
    "title": "Conceptos comunes de búsqueda",
    "paragraphs": [
        {
            "context": "Los algoritmos de búsqueda exploran el espacio de estados generando un árbol de búsqueda cuy
            "questions": [
                {
                    "question": "¿Cómo exploran los algoritmos de búsqueda el espacio de estados?",
                    "answers": [
                        "Los algoritmos de búsqueda exploran el espacio de estados generando un árbol de búsqueda cu
                        "exploran el espacio de estados generando un árbol de búsqueda cuya raíz es el estado inicia
                    ]
                },

                {
                    "question": "¿Con que se corresponden los nodos exteriores de un algoritmo de búsqueda?"
                },
                {
                    "question": "¿Cuantos caminos hay desde un estado inicial a otro estado?",
                    "answers": [
                        "pueden existir múltiples caminos desde el estado inicial a un estado cualquiera",
                        "pueden existir múltiples caminos",
                        "múltiples caminos"
                    ]
                },
            ]
        },
        {
            "context": "Los algoritmos de búsqueda exploran el espacio de estados generando un árbol de búsqueda cuy
            "questions": [
                {
                    "question": "¿Qué exploran los algoritmos de búsqueda?",
                    "answers": [
                        "Los algoritmos de búsqueda exploran el espacio de estados generando un árbol de búsqueda cu
                        "el espacio de estados generando un árbol de búsqueda cuya raíz es el estado inicial",
                        "el espacio de estados",
                        "Los algoritmos de búsqueda exploran el espacio de estados",
                        "exploran el espacio de estados generando un árbol de búsqueda cuya raíz es el estado inicia
                    ]
                },
                {
                    "question": "¿Cuales son los nodos hojas?",
                    "answers": [
                        "Los nodos al final de las ramas son los nodos hoja",
                        "Los nodos al final de las ramas"
                    ]
                },
                {
                    "question": "¿En que parte del árbol se encuentran los frutos?"
                },
                {
                    "question": "¿Qué influye en el modo en el que actúa un algoritmo de búsqueda?",
                    "answers":[
                        "La forma en la que el árbol se genera",
                        "La forma en la que el árbol se genera eligiendo expandir unos estados u otros"
                    ]
                },
                {
                    "question": "¿De que depende el comportamiento de las personas?"
                }
            ]
        }
    ]
}
```

Anexo 4. Listado de modelos BERT evaluados (incluidos los que no forman parte del estudio comparativo)



Anexo. Artículo de investigación

Análisis de modelos basados en Transformers

Iván Corrales Solera

Universidad Internacional de la Rioja, Logroño (España)

02/02/2022

RESUMEN

El uso de los modelos lingüísticos basados en el mecanismo de atención, conocidos como Transformers, y más concretamente las variantes pre-entrenadas a partir de BERT (Bidirectional Encoder Representations from Transformers), se han convertido en el estado del arte para los sistemas de búsqueda de respuesta. El aprendizaje por transferencia permite partir de modelos previamente pre-entrenados, de carácter generalistas, y entrenarlos para su especialización. La aparición de nuevos conjuntos de datos en español, han favorecido la proliferación de nuevos modelos que, apoyándose en modelos ya pre-entrenados, proporcionan un gran rendimiento para la realización de tareas de Procesamiento de Lenguaje Natural en español. Se establece que el entrenamiento de un modelo a partir de un dataset especializado, creado con el temario de la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento, mejoraría los resultados obtenidos.

PALABRAS CLAVE

PLN Sistema de búsqueda de respuestas, Transformers

I. INTRODUCCIÓN

Los Sistemas de Búsqueda de Respuestas (SBR) – del inglés Question Answering systems (QA), también conocidos como sistemas pregunta-respuesta (PR), adquieren el conocimiento de conjuntos de datos disponibles para posteriormente ser capaces responder a preguntas realizadas. El desarrollo de los SBR aporta gran valor a los sistemas de información debido a que cubren una tarea de alta complejidad y con un elevado coste para ser realizada manualmente por personas.

Es importante destacar que la mayor ventaja de estos sistemas radica en que dan respuestas en lenguaje natural a preguntas realizadas también en lenguaje natural. Los SBR son de gran utilidad dentro del campo de la Inteligencia Artificial (IA) puesto que la mayoría de los problemas relacionados con el aprendizaje profundo, del inglés Deep Learning (DL), se pueden modelar como un problema de respuesta a preguntas.

El objetivo de este trabajo surge de la necesidad de cubrir un requisito planteado en un proyecto de investigación de la Universidad Internacional en la Rioja (UNIR). Este requisito consiste en ser capaz de extraer automáticamente, de los temarios de las asignaturas, la respuesta a una pregunta. Por lo tanto, trabajaremos con un dominio cerrado, que viene determinado por el temario en castellano de asignaturas de UNIR, como es la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento del Grado en Ingeniería Informática.

Tras haber realizado un proceso de investigación y estudio del estado del arte de los SBR, se confirma el hecho de que las arquitecturas basadas en los mecanismos de atención, conocidas como Transformers, son las que mejores resultados ofrecen en la actualidad. Por lo tanto, los modelos que participaran en el estudio están basados en esta arquitectura, y más concretamente en BERT.

II. ESTADO DEL ARTE

Los Transformers se basan en la idea de la auto atención introducida por primera vez por un grupo de investigadores de Google[59] en el artículo “Attention is All you Need” [1] (Vaswan, Shazeer Parmar et al. 2017). Estos son modelos de aprendizaje automático semi-supervisados que se utilizan principalmente con datos de texto y han reemplazado a las redes neuronales recurrentes en tareas de procesamiento de lenguaje natural.

Los Transfomers utilizan una arquitectura de tipo codificador-decodificador. Sin embargo, estos proponen una composición interna completamente diferente a las soluciones actuales en ese momento, ya que se prescinde del uso de las Redes Neuronales Recurrentes [2], conocidas en inglés como Recurrent Neural Networks (RNN), y utilizan unos módulos llamados auto atención, del inglés self-attention. Esta nueva arquitectura mejora el rendimiento del entrenamiento de los modelos al paralelizar el aprendizaje.

Los Transformers están diseñados para trabajar con datos de secuencia y tomarán una secuencia de entrada y la usarán para

generar una secuencia de salida un elemento a la vez. Un transformador está formado por dos componentes principales. El primero es un codificador que se centra en la secuencia de entrada y el segundo es un decodificador que lo hace en la secuencia de salida y predice el siguiente elemento de la secuencia.

El objetivo del mecanismo de auto atención es conocer con que otra palabra de la secuencia está relacionada la palabra que se procesa en un instante de tiempo.

Otro punto de ventaja que presentan los Transformer frente a las RNN es el uso del contexto, o tamaño de ventana de memoria que pueden utilizar las arquitecturas basadas en mecanismos de auto atención. Las RNN son capaces de referenciar a palabras que han aparecido anteriormente en la secuencia de entrada. Sin embargo, cuando trabajamos con secuencias muy largas, no pueden acceder a palabras muy antiguas. A pesar de que con las GRU [3], Gated Recurrent Unit, y las LSTM [4], del inglés Long short-term memory, se consigue ampliar este tamaño de ventana en las RNN, la capacidad sigue siendo limitada. La principal ventaja del mecanismo de auto atención es que posee una ventana infinita y que únicamente queda limitada por la potencia computacional de nuestros sistemas.

Gracias a la arquitectura de Transformer, han aparecido diferentes tecnologías que pretenden convertirse en el estado del arte del Procesamiento de Lenguaje Natural (PLN). Destacan especialmente GPT-3 [5] y BERT [6]. Para el desarrollo de este artículo nos centraremos en las arquitecturas basadas en BERT.

BERT es desarrollado por Google y fue presentado por Devlin et al. (2018) y su nombre se corresponde con las siglas en inglés de “Bidirectional Encoder Trepresentations from Transformers”. El hecho de que sea bidireccional se refiere a que BERT analiza las frases de búsqueda en ambas direcciones, considerando las palabras situadas a la izquierda y la derecha de cada palabra clave. Esto se logra gracias a la técnica conocida como Masked LM [7] (MLM) que permite el entrenamiento bidireccional en modelos en los que antes era imposible.

Este modelo se ha entrenado haciendo uso de datos no etiquetados, como artículos de Wikipedia, grandes corpus de noticias o libros, consiguiendo de este modo entrenar el modelo sobre una cantidad de datos enorme. Gracias a ello, BERT tiene una representación general del lenguaje natural que posteriormente podremos utilizar para solventar tareas más concretas. BERT presenta resultados de última generación en una amplia variedad de tareas de PLN, incluidos los sistemas de búsqueda de respuesta (SQuAD v1.1), la inferencia de lenguaje natural o clasificación de textos entre otras. Por lo tanto, el uso de BERT para realizar una tarea específica es trivial, simplemente necesitamos entrenar el modelo ya pre-entrenado con los datos del problema que queremos cubrir. Por ejemplo, en nuestro caso, que estamos construyendo un SBR debemos introducir la pregunta, el separador [SEP] y la respuesta.

El hecho de que BERT sea un sistema de código abierto ha permitido que otros puedan utilizarlo como base para el desarrollo de nuevas tecnologías para el procesamiento de lenguaje natural. Destacamos los casos de RoBERTA [8], DistilBERT [9] o XLNet [10] entre otros.

La principal ventaja proporcionada por BERT es el hecho de que el modelo sea bidireccional, en lugar de un modelo unidireccional como es el caso de GPT. El uso del modelo bidireccional hace que el contexto aprenda en función de las palabras que lo rodean en lugar de solo considerar la palabra anterior o posterior.

Además, gracias a la técnica de Aprendizaje por Transferencia, conocida en inglés como Transfer Learning, han emergido numerosos modelos que, en lugar de partir de cero, para su creación, se benefician de otras implementaciones existentes.

El aprendizaje por transferencia se inspira en las capacidades de los seres humanos para transferir conocimientos a través de tareas, el aprendizaje de transferencia tiene como objetivo aprovechar el conocimiento de un dominio de origen para mejorar el rendimiento del aprendizaje o minimizar el número de ejemplos etiquetados necesarios en un dominio de destino (Wei, Zhang et al, 2018 [11]).

Es habitual encontrar que necesitamos resolver una tarea de clasificación en un dominio de interés, pero solo tenemos suficientes datos de entrenamiento en otro dominio. En tales casos, la transferencia de conocimientos, si se realiza con éxito, mejoraría enormemente el rendimiento del aprendizaje al evitar esfuerzos muy costosos de etiquetado de datos.

En los últimos años, el aprendizaje por transferencia ha surgido como un nuevo marco de aprendizaje para abordar numerosos problemas dentro del área de PLN (Peters et al., 2018 [12]; Howard y Ruder, 2018 [13]; Radford et al., 2018; Devlin et al., 2018 [14]).

III. Objetivos y Metodología

El objetivo general de este proyecto es realizar un estudio comparativo de las diferentes técnicas para la construcción de sistemas de búsquedas de respuestas con el fin de encontrar cuál de ellas proporciona mejores resultados para un dominio cerrado, que viene determinado por el temario en castellano de asignaturas del Grado de Informática de la Universidad Internacional de la Rioja.

Los objetivos específicos del presente estudio son los siguiente:

1. Revisión sistemática de las investigaciones y avances realizados para la construcción de Sistemas de Búsqueda de Respuestas haciendo uso de tecnologías de Inteligencia Artificial.
2. Explorar diferentes técnicas y arquitecturas de inteligencia Artificial utilizadas para la construcción de Sistemas de Búsqueda de Respuestas.
3. Diseñar e implementar pruebas de conceptos siguiendo las diferentes técnicas de Inteligencia Artificial exploradas para la construcción de SBR.
4. Identificar las asignaturas y los datos de las mismas que serán utilizados para el desarrollo de el estudio presente.
5. Determinar las métricas utilizadas para realizar una comparación objetiva entre las diferentes técnicas de Inteligencia Artificial exploradas.
6. Sintetizar los resultados obtenidos y enumerar la lista de puntos a favor y en contra de cada una de las técnicas utilizadas en el estudio comparativo.

El proceso de investigación seguido para la elaboración de este trabajo de fin de Máster se ha dividido en 4 fases: fase exploratoria, fase de inicio y planificación, fase de desarrollo y fase de análisis.

IV. Contribución

Son muchas las soluciones existentes basadas en Transformers que nos permiten el tratamiento de lenguaje natural y en concreto para la construcción de sistemas de búsqueda de respuesta. Para acotar el número de soluciones contempladas en el estudio presente me centraré en el uso de modelos basados en la arquitectura BERT.

Gracias a las técnicas de aprendizaje por transferencia, un modelo no tiene que partir de cero, sino que se puede

aprovecharse de modelos ya entrenados y entrenarlos con otro conjunto de datos. De hecho, existen modelos publicados que han partido del mismo modelo como base y. han utilizado el mismo conjunto de datos para entrenarlos. Esto ocurre puesto que los modelos son públicos y los conjuntos de datos utilizados también. Las predicciones de estos modelos no tienen porque ser idénticas debido a la aleatoriedad de los sistemas de aprendizaje profundo y al hecho de que los híper parámetros de estos modelos no tienen porque haber sido inicializados con los mismos valores. A continuación, se enumeran las arquitecturas que participaran en este estudio.

RoBERTa

RoBERTa es desarrollado por el equipo de IA de Facebook[60] y es anunciado en el artículo de Liu, Ott et al. (2019) en la publicación de "RoBERTa: A Robustly Optimized BERT Pretraining Approach".

RoBERTa se basa en la estrategia del enmascaramiento del lenguaje seguida por BERT. Sin embargo, realiza cambios como es la modificación de los hiperparámetros clave en BERT, aparte de realizar el entrenamiento haciendo uso de tasas de aprendizaje mucho mas grandes. A parte de estos cambios arquitectónicos RoBERTa ha sido entrenado con un orden de magnitud de más datos que BERT. Durante un periodo de tiempo más largo. Para el entrenamiento de esta arquitectura se utilizó conjunto de datos sin anotar a parte de un conjunto de datos novedoso extraído de artículos de noticias públicos.

Los resultados obtenidos por RoBERTa en la prueba comparativa GLUE [15] demuestran que se encuentra en el nivel más alto de la clasificación y que se pueden mejorar los resultados obtenidos por BERT en el desempeño de diferentes tareas de PLN.

DistilBERT

DistilBert es anunciado por Sanh, Debut et al. (2019) en la publicación de su artículo "DistilBERT, a destilled versión of BERT: smarller, faster, cheaper and lighter". En este artículo destacan que DistilBERT es un modelo más pequeño, rápido y liviano, y por lo tanto es más barato de preentrenar.

Tal y como se indica en el apartado anterior, la esencia de DistilBERT radica en proporcionar un modelo más genérico pero que sea más fácil de ajustar para gran variedad de tareas. Además, a diferencia con otros trabajos DistilBERT destaca por reducir el tamaño de BERT en un 40% al mimo tiempo que conserva el 97% de su comprensión de idioma. A nivel de rendimiento los estudios existentes han demostrado que es un 60% más rápido.

Ixambert [16]

Se trata de un modelo que ha sido preentrenado para los idiomas inglés , español y euskera. El corpus utilizado está compuesto por las Wikipedias en inglés, espaol y euskera, a parte de noticias en euskera de periódicos online.

Es importante destacar que este modelo se ha utilizdo para transferir conocimientos del inglés al euskera en un sistema de control de calidad conversacional, tal y como indican Otegi, Agirre et al. (2020) en su publicación "Conversational Question Answering in Low Resource Scenarios: A Dataset and Case Study for Basque".

V. RESULTADOS

Las siguientes tablas muestran las estadísticas de las preguntas que componen el dataset utilizado para evaluar los modelos

Nº de respuestas	Total de preguntas
0	56
1	22
2	16
3	11
4	3
5	2
6	1

Table 1 Total de preguntas por número de respuestas

	Nº total	Nº medio de caracteres	Nº medio de Tokens
Preguntas	112	55.76	11.47
Respuestas	120	50.69	9.78
Contexto	15	871.07	170.94

Tabla 2Estadísticas de preguntas, respuestas y contexto

A continuación se muestran los valores de las métricas EM (Exact Match) y F1para cada uno de los modelos que han participado en el estudio.

PlanTL-GOB-ES/Roberta-base-bne-sqac

Se trata de un modelo de Transformer basado en RoBERTa para el idioma español. Este modelo fue originalmente pre-entrenado con el corpus en idioma español más grande hasta el momento de su publicación, con un tamaño de 570 GB de datos, después de haber sido limpiados y eliminando las duplicaciones. En este estudio utilizaremos una versión del modelo que ha sido entrenado con el dataset SQAC[61] (Spanish Questions-Answering Corpus). Este trabajo ha sido parcialmente financiado por la Secretaría de Estado de Digitalización e Inteligencia Artificial (SEDIA) de España en el marco del Plan-TL, y el Centro de Cómputo del Futuro, una iniciativa del Centro de Supercomputación de Barcelona e IBM (2020)

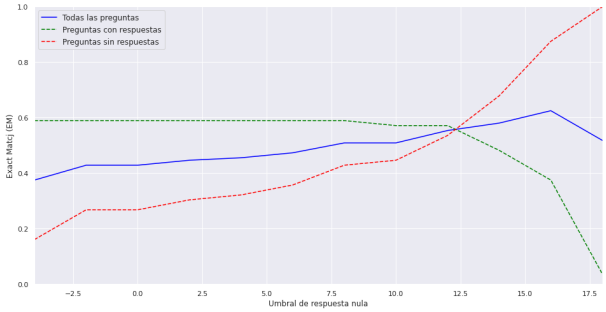


Figura 1 Valor de EM para PlanTL-GOB-ES/roberta-base-bne-sqac

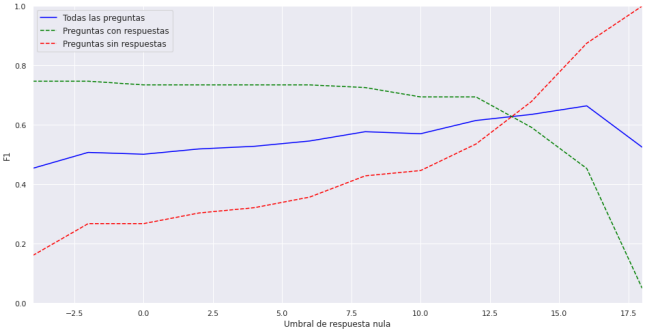


Figura 2 Valor de F1 para PlanTL-GOB-ES/roberta-base-bne-sqac

La figura 1 y 2 muestran los valores para las métricas EM y F1 respectivamente para el modelo PlanTL-GOB-ES/roberta-base-bne-sqac..

jamarju/Roberta-base-bne-squad-2.0-es

Este modelo, a pesar de que no tiene una organización que lo sustente como era el caso de los modelos pertenecientes a Plan TL-GOB-ES, me ha resultado interesante de tener en cuenta para el estudio comparativo porque ha sido entrenado inicialmente con el corpus de bne y posteriormente con el corpus squad-2.0-es[62]. El hecho de considerar para le estudio un modelo que utilice el dataset squad-2.0-es me parece relevante puesto que es una traducción automática al idioma español del bien conocido Standord Question Answering Dataset (SQUAD v2). El modelo inicial del que parte es el modelo simplificado de RoBERTa.

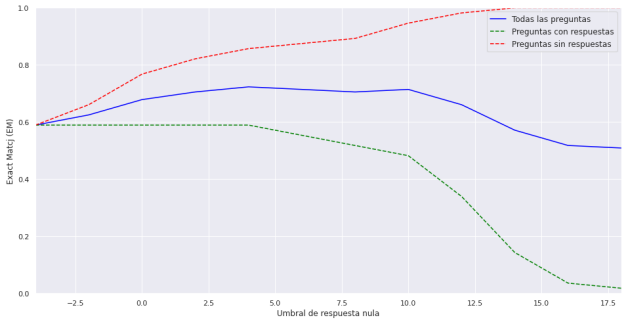
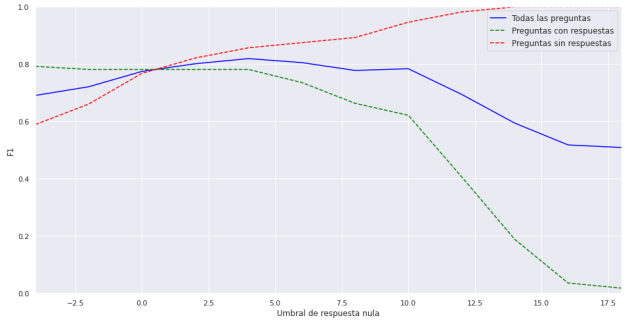


Figura 3 Valores de EM para jamarju/roberta-base-bne-squad-2.0-es



Las figuras 3 y 4 muestran los valores para las métricas EM y F1 respectivamente para el modelo jamarju/roberta-base-bne-squad-2.0-es.

mrm8488/distill-bert-vase-spanish-wwm-cased-finetuned-spasquead2-es

Este modelo ha sido entrenado partiendo del modelo base bert-base-multilingual-cased., también conocido como DistilmBERT, y usando el conjunto de datos de SQuAD-es-v2.0 para su entrenamiento.

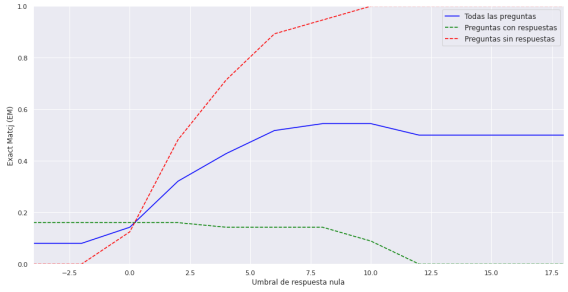


Figura 5 Valores de EM para mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

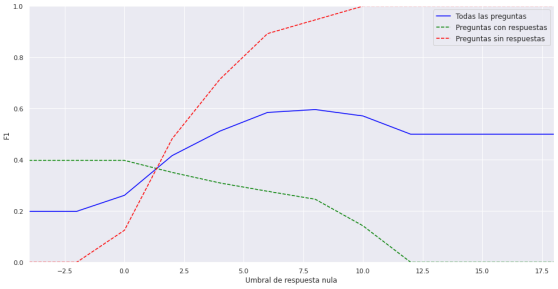


Figura 6 Valores de F1 para mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

Las figuras 5 y 6 muestran los valores para las métricas EM y F1 respectivamente para el modelo mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es.

MarcBrun/ixambert-finetuned-squad

Este modelo se trata de una implementación básica del modelo multilingüe "ixambert-base-cased", entrenado con el dataset SQuAD v1.1, que es capaz de responder preguntas básicas sobre hechos en inglés, español y euskera.

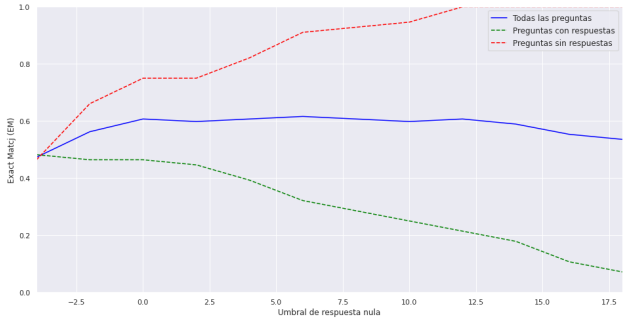


Figura 7 Valores de EM para MarcBrun/ixambert-finetuned-squad

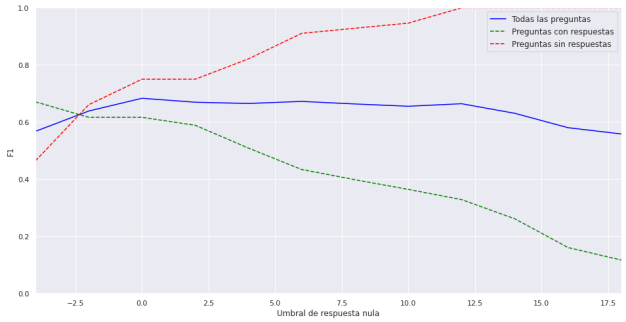


Figura 8 Valores de F1 para MarcBrun/ixambert-finetuned-squad

Las figuras 7 y 8 muestran los valores para las métricas EM y F1 respectivamente para el modelo MarcBrun/ixambert-finetuned-squad.

VI. Discusión

Los resultados obtenidos en el estudio serán discutidos en los siguientes bloques

Umbral de respuesta nula

El uso del umbral de respuesta nula nos permite identificar cuando nuestro modelo debe devolver la cadena vacía en lugar de una posible respuesta.

Para el caso de nuestro estudio, un sistema que nos permite evaluar las respuestas de los alumnos a preguntas de una asignatura, podríamos pensar que encontrar el umbral de respuesta vacío se trata de una tarea prescindible, puesto que todas las preguntas han de tener una respuesta. Sin embargo, esto no es así, puesto que si únicamente nos fijamos en que el sistema nos devuelva respuestas podríamos construir un sistema que puede devolver respuestas sin sentido.

Si nos fijamos en cualquiera de las figuras anteriores, observamos como a medida que aumentamos el umbral las líneas de los grupos “Preguntas con respuestas” y “Preguntas sin respuestas” comienzan a divergir. La línea que se corresponde con “Preguntas sin respuestas” tiende a alcanzar el valor máximo (1) tanto para la métrica EM como para la métrica F1, mientras que la línea que se corresponde con el grupo de Preguntas con respuestas tiende a 0. Esto ocurre por el hecho de que somos más exigentes con el mínimo puntaje requerido para muestras predicciones.

En cualquiera de las figuras anteriormente citadas podemos observar un punto de corte, en el se cruzan las 3 líneas. Este punto ocurre en un valor diferente del umbral dependiendo del modelo que estamos mostrando, e incluso dependiendo de la métrica en concreto. No debemos olvidar que para el cálculo de las métricas mostradas se ha utilizado el mismo conjunto de datos, y por lo tanto lo que estamos observando es que para cada modelo y nuestro conjunto de datos tendremos un valor de umbral óptimo diferente. Por lo tanto, las conclusiones obtenidas en este estudio no se pueden extrapolar a otros conjuntos de datos.

Es importante recordar que nuestro dataset está compuesto por el mismo número de preguntas con respuestas quede preguntas sin respuestas. Esto tiene implicaciones que pueden hacer que las métricas sean engañosas. Por ejemplo, si usamos un valor de umbral muy elevado, el modelo considerará que no hay ninguna respuesta lo suficientemente válida para ninguna pregunta y aún así el valor de F1 y de EM será del 0.5 (como poco) porque al menos habrá predicho con exactitud los casos donde no es posible encontrar una respuesta válida.

Por ejemplo, si nos fijamos en la figura 8 dónde se muestra los valores de la métrica F1 para el modelo MarcBrun/ixambert-finetuned-squad, podríamos pensar que el valor óptimo del umbral es cualquier valor comprendido entre 0 y 12 puesto que el valor de la métrica se mantiene constante en torno a F1=0.65 para todos estos valores. Ese valor se mantiene constante única y exclusivamente porque el modelo está prediciendo con éxito la amplia mayoría de los casos negativos (preguntas sin respuestas). Sin embargo, si usamos umbrales próximos a 12 el modelo no sería capaz de predecir correctamente preguntas que si tienen respuestas.

Devlin, Chang et al. (2019) dicen que se predice una respuesta no nula cuando $\hat{s}_{i,j} > s_{null} + \tau$, donde el umbral τ se selecciona en el conjunto de desarrollo para maximizar F1. El método squad_evaluate del paquete data.metrics.squad_metrics de la librería de Transformers[63] nos proporciona las métricas para e conjunto de datos proporcionado, pero además cual sería el valor de umbral óptimo para maximizar los valores de EM y F1.

Mi pensamiento al respecto, a cerca de cual sería el valor umbral óptimo que deberíamos utilizar para cada modelo, es que no se puede basar únicamente en el valor que nos proporciona el mejor resultado. De hecho, considero que el valor optimo que se debería utilizar está cercano al punto de corte de las 3 líneas representadas en las figuras anteriores. Además, independientemente del punto de corte, considero que se deben considerar otros factores, como es el caso de la pendiente de la línea que se corresponde con el conjunto de datos formado por las preguntas que si tienen respuestas. Por ejemplo, en la figura 3, que muestra los valores de F1 para el modelo jamarju/roberta-base-bne-squad-2.0-es, considero que el valor óptimo del umbral sería próximo a 4, porque para ese valor se ha mantenido el valor de la métrica que se obtuvo para el uso de umbral de 2, a la par que se ha aumentado el valor de la métrica para el conjunto total de todas las preguntas.

Evaluación de métricas EM y F1

Teniendo en cuenta los valores obtenidos para las métricas F1 y EM en los modelos estudiados se puede confirmar que ambas tienen un comportamiento similar. De hecho, si nos fijamos en las figuras 1-8 y prestamos atención a los valores utilizados como umbral de respuesta vacía, que están representados en el eje X, se observa como la gráfica que representa la métrica F1 podría considerarse como una representación desplazada hacia la derecha de la gráfica que muestra los valores de la métrica EM para un mismo modelo.

Lo citado en el párrafo anterior tiene sentido, puesto que la métrica EM es más estricta que la métrica F1. Como consecuencia de esto, los valores de la métrica EM empiezan a tener una tendencia descendente antes que para los valores de la métrica F1. Esto provoca que el punto de corte ocurra para valores de umbrales menores en el caso de la métrica EM.

Para el caso del conjunto de datos formado por preguntas negativas, es decir, sin respuestas, los valores de EM y F1 serán

exactamente iguales, puesto que no hay posibilidad de estar correcta en un porcentaje de la respuesta.

Para un entendimiento de estas métricas es importante tener en cuenta el tipo de preguntas que componen nuestro conjunto de datos. En nuestro caso, para entender los resultados obtenidos, es fundamental que observemos la tabla 1, que muestra cuantas preguntas hay para un número de respuestas en concreto y la tabla 2 que nos muestra las longitudes y número de tokens medio para las respuestas.

En caso de tener preguntas con muchas posibles respuestas, los valores de la métrica F1, y probablemente la métrica EM, serán más elevados, ya que el valor de la métrica se obtiene calculando el valor máximo de la predicción para las posibles respuestas validas. Por lo tanto, tener más posibilidades de acierto implica tener más posibilidades de tener un valor elevado de las métricas.

Por otro lado, el tipo de pregunta que forman nuestro sistema, o más bien, el tipo de respuesta condicionará no sólo el valor de las métricas sino también que métrica resulta más importante para nuestro caso. En nuestro caso, dónde la amplia mayoría de las preguntas cuenta con una o varias frases, el uso de la métrica F1 aportará más valor a la hora de decidir que modelo es el más adecuado. Sin embargo, si quisiéramos construir un sistema que respondiera a preguntas factoides del tipo ¿En que año se invento la maquina de Turing?, ¿Cuál es el nombre completo del jugador de ajedrez vencido por Deep Blue? ó ¿En que mes tiene lugar el evento del TRec? Deberíamos fijarnos en los valores obtenidos por la métrica EM.

Comparación de modelos

No es una tarea sencilla decantarse por el modelo más adecuado para nuestro sistema teniendo en cuenta el tamaño reducido del conjunto de datos utilizado Sin embargo, teniendo en cuenta estas limitaciones, me atrevería a decir que el modelo más adecuado para nuestro sistema, sería aquel dónde, fijándonos en las figuras que muestran los valores de la métrica EM, la intersección de las líneas se produzca en un valor del eje Y (valor de la métrica) más elevado. Del mismo modo, deberíamos fijarnos en el eje X (valor de umbral utilizado) para utilizarlo como valor.

Los modelos que mejores resultados nos darían para la construcción de nuestro sistema sería jamarju/roberta-base-bne-squad-2.0-es.

VII. Conclusiones

El estudio comparativo realizado en este trabajo me ha permitido no sólo profundizar en el estado del arte de los sistemas de búsqueda de respuestas sino establecer un contacto directo con las tecnologías que estas ocupando la primera línea de la IA, es decir, las arquitecturas basadas en mecanismos de atención, conocidas como Transformers.

En cuanto al problema en estudio, para el que se quería encontrar la solución más adecuada he de reconocer que con las limitaciones temporales, obvias de un proyecto de fin de máster, y las limitaciones de recursos, no se puede descartar la opción de entrenar un modelo propio, en lugar de utilizar directamente uno ya entrenado. Poder entrenar un modelo propio requiere de sistemas de computo mejores que los utilizados para el desarrollo de este trabajo. Al fin y al acabo lo que he utilizado han sido instancias de pago por cuota mensual ofrecidas por Google, pero para un trabajo de investigación con mayor alcance se deberían considerar otras alternativas

Por otro lado, se puede confirmar que el camino a seguir para la construcción de un sistema de búsqueda de respuestas para el ámbito académico, y más concretamente para la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento, debe apoyarse en el uso de los Transformers. En concreto, los resultados obtenidos por los modelos basados en RoBERTa han sido los mejores para el conjunto de datos que ha formado parte del estudio.

Este trabajo abre la puerta a otros trabajos de investigación y estudios, siendo los siguientes los más relevantes

Elaboración de dataset académico en español

Aportaría gran valor la elaboración de un dataset con un formato basado en SQUAD v2 . La versión española de SQUADv2 se ha creado a partir del uso de traducciones automáticas. Esto no digo que sea una mala aproximación para tener un gran corpus con el que poder entrenar nuestros modelos, pero creo que sería muy interesante poder desarrollar un corpus académico en idioma español. Este dataset podría especializarse en preguntas relacionadas con el área de la Ingeniería Informáticas y más concretamente con el área de Inteligencia Artificial.

Métricas de detección de umbrales de respuesta óptima nulos

Tras haber realizado la evaluación de los modelos y haber intentado resolver la pregunta inicial, cual de los modelos es el que mejor se adecúa a nuestras necesidades, creo que no es una respuesta sencilla, y que los cálculos utilizados en la actualidad para obtener los valores óptimos de umbrales no son lo suficientemente potentes. Se podría trabajar en el desarrollo de técnicas que permitan encontrar el uso óptimo de de los valores de umbrales a utilizar para un modelo.

Nuevo formato para datasets

Considero que el uso del formato JSON para el almacenamiento de datasets hace que estos ficheros tengan un tamaño muy elevado de lo necesario. A pesar de que JSON reduce el tamaño de los ficheros, si lo comparamos con los documentos de lenguaje de marcado, XML [17], creo que la solución más adecuada debería apostar por el diseño de un propio DSL, del inglés Domain Specific Language, y en español conocido como Lenguaje específico de dominio.

Referencias

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017..

[2] Gherrity, "A learning algorithm for analog, fully recurrent neural networks," International 1989 Joint Conference on Neural Networks, 1989, pp. 643-644 vol.1, doi: 10.1109/IJCNN.1989.118645.

[3] R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 2017, pp. 1597-1600, doi: 10.1109/MWSCAS.2017.8053243.

[4] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[5] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language Models are Few-Shot Learners. *ArXiv, abs/2005.14165*.

[6] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv, abs/1810.04805*.

[7] Kushilevitz, G., Markovitch, S., & Goldberg, Y. (2020). A Two-Stage Masked LM Method for Term Set Expansion. 6829–6835.

https://doi.org/10.18653/v1/2020.acl-main.610

[8] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. 1. http://arxiv.org/abs/1907.11692

[9] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2–6. http://arxiv.org/abs/1910.01108

[10] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. Advances in Neural Information Processing Systems, 32(NeurIPS), 1–18.

[11] Li, X., Zhang, W., Ding, Q., & Li, X. (2019). Diagnosing rotating machines with weakly supervised data using deep transfer learning. *IEEE transactions on industrial informatics*, *16*(3), 1688-1697.

[12] Rojas-Carulla, M., Schölkopf, B., Turner, R., & Peters, J. (2018). Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, *19*(1), 1309-1342.

[13] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

[14] Gu, J., Hassan, H., Devlin, J., & Li, V. O. (2018). Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.

[15] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

[16] Otegi, A., Agirre, A., Campos, J. A., Soroa, A., & Agirre, E. (2020, May). Conversational Question Answering in Low Resource Scenarios: A Dataset and Case Study for Basque. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 436-442).

[17] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (1997). Extensible markup language (XML). World Wide Web Journal, 2(4), 27-66.