UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Recuperación de información para la búsqueda de respuestas en idioma Español

Lucía Bouza Heguerte

Programa de Posgrado en Ciencia de Datos y Aprendizaje Automático
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Marzo de 2023

UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Recuperación de información para la búsqueda de respuestas en idioma Español

Lucía Bouza Heguerte

Tesis de Maestría presentada al Programa de Posgrado en Ciencia de Datos y Aprendizaje Automático, Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Magíster en Ciencia de Datos y Aprendizaje Automático.

Director:
Guillermo Moncecchi

Director académico:
Guillermo Moncecchi

Montevideo – Uruguay
Marzo de 2023

Bouza Heguerte, Lucía

Recuperación de información para la búsqueda de respuestas en idioma Español / Lucía Bouza Heguerte. - Montevideo: Universidad de la República, Facultad de Ingeniería, 2023.

XVIII, 97 p. 29, 7cm.

Director:

Guillermo Moncecchi

Director académico:

Guillermo Moncecchi

Tesis de Maestría – Universidad de la República, Programa en Ciencia de Datos y Aprendizaje Automático, 2023.

Referencias bibliográficas: p. 83 – 88.

1. PLN, 2. Recuperación de Información, 3. Búsqueda de respuestas, 4. Temporalidad en colecciones de documentos, 5. Modelos de Lenguaje. I. Moncecchi, Guillermo, . II. Universidad de la República, Programa de Posgrado en Ciencia de Datos y Aprendizaje Automático. III. Título.

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

Aiala Rosá

Lorena Etcheverry

Álvaro Cabana

Montevideo – Uruguay
Marzo de 2023

RESUMEN

En esta tesis se presenta un análisis de diferentes técnicas de recuperación de información que son utilizadas en el contexto de búsqueda de respuestas, en documentos de prensa y en documentos enciclopédicos en el idioma español. El trabajo identifica e intenta cuantificar algunos problemas en la evaluación estándar de esta tarea.

Como parte central de los aportes de este trabajo, se propone un método manual para obtener una mejor estimación del rendimiento real del sistema, haciendo foco en problemas que involucren temporalidad, pero que puede ser utilizado también en otros contextos. El análisis realizado mediante el método manual permite medir cuánto subestima el rendimiento del sistema el método de evaluación utilizado habitualmente.

Como parte del trabajo, se desarrolló el módulo de recuperación de información del sistema DPR (Karpukhin et al., 2020), para ser utilizado para el idioma español, que queda disponible como un nuevo recurso para la investigación del problema.

Palabras claves:
PLN, Recuperación de Información, Búsqueda de respuestas, Temporalidad en colecciones de documentos, Modelos de Lenguaje.

Lista de tablas

Tabla de contenidos

Capítulo 1

Introducción

El problema de Recuperación de Información (Information Retrieval - IR) focalizado en la tarea de Búsqueda de Respuestas (Question Answering - QA) es un área activa de investigación. Diferentes métodos y técnicas se estudian para la mejora del rendimiento de la tarea a nivel de calidad de resultados y eficiencia computacional, con el objetivo de construir sistemas fiables y útiles para los usuarios, aunque los mayores avances se han alcanzado para preguntas y documentos en idioma inglés.

Definimos la recuperación de información, como «*la tarea de encontrar material de naturaleza no estructurada en alguna colección, que satisfaga las necesidades de información del usuario*» (Manning et al., 2009). La definición es amplia y puede instanciarse en diferentes tareas, como recuperación de imágenes, videos, sitios web, documentos, etc.. Entonces, en los sistemas de IR se ingresa una consulta que es la representación de la necesidad de información del usuario, y el sistema devuelve objetos de la colección que maneja, que satisfacen la consulta, según algún criterio preestablecido.

La tarea más común de IR es llamada Ad Hoc Retrieval e implica que un usuario ingrese una consulta en lenguaje natural en el sistema, y éste le devuelva un conjunto ordenado de documentos de una cierta colección (Jurafsky y Martin, 2020) [1]. Llamamos *documento* a cualquier elemento indexado por el sistema, perteneciente a la colección. Pueden ser porciones de texto, páginas web, publicaciones científicas, etc.. Las consultas son conjuntos de términos,

[1] Otro ejemplo de tarea de IR es Image Retrieval, donde la colección que maneja el sistema son imágenes y las consultas pueden ser expresadas en lenguaje natural o mediante imágenes. Dependiendo del sistema, el criterio de satisfacción de la necesidad de información del usuario pueden ser imágenes similares o imágenes que contengan en su metadata términos de la consulta. En este trabajo trabajaremos con IR restringida a textos.

siendo éstos palabras o conjuntos de ellas. Un documento es relevante si satisface la consulta. Para ejemplificar, supongamos que tenemos una colección con 3 documentos con el contenido indicado en la tabla 1.1

ID	Contenido documento
1	*Uruguay es un país de Sudamérica.*
2	*Valverde juega en España.*
3	*Valverde es un jugador uruguayo.*

Tabla 1.1: Ejemplos de documentos de una colección

Ante la consulta del usuario: «*¿Dónde queda Uruguay?*», El sistema deberá ordenar los 3 documentos según relevancia. El orden devuelto dependerá del método, pero una posible salida podría ser: documento 1, 3 y 2, siendo el primer documento el relevante, ya que es el único que contiene la respuesta a la pregunta.

Question Answering (QA) es la tarea de encontrar en el texto de documentos, respuesta a una pregunta planteada por un usuario. Para llevar a cabo esta tarea, uno de los paradigmas principales es Búsqueda de Respuestas basado en Recuperación de Información (IR based QA). En este paradigma dada una consulta del usuario, un módulo de IR (Ad Hoc Retrieval) se encarga de encontrar los documentos relevantes para esa consulta. Luego otro módulo se encarga de buscar y extraer la respuesta a la consulta en ese subconjunto de documentos preseleccionados (Jurafsky y Martin, 2020).

Casi todos los sistemas de IR tienen como particularidad que asignan un puntaje a los elementos de la colección, representando qué tan bien se ajustan a la consulta del usuario. En el contexto de IR para QA, esto es necesario ya que el sistema de IR debe elegir los documentos donde sea más probable encontrar la respuesta, para pasarlos a los algoritmos de comprensión de lectura y extraer la respuesta.

1.1. Recuperación de Información: Ad Hoc Retrieval

La tarea de recuperación de información más común es Ad Hoc Retrieval, por lo que muchas veces suele omitirse la referencia correspondiente. A continuación se describirá brevemente el marco teórico de la recuperación de información, siempre considerando el caso más habitual, que es justamente aquel que interesa para la búsqueda de respuestas.

Existe una larga trayectoria de investigación en recuperación de información, que data casi desde los inicios de la computación. Sin embargo, en 1960 se hace la primera caracterización del problema de recuperación de información, definiendo también la noción de relevancia de un documento (Maron y Kuhns, 1960).

Salton (1971) plantea la arquitectura base para los sistemas de recuperación de información, aún utilizada hoy en día, proponiendo modelar documentos y consultas como vectores basados en conteos de términos. El ordenamiento se realiza utilizando la distancia coseno entre los vectores que representan a los documentos y la consulta. Para el ordenamiento de documentos fue muy utilizado en los inicios y aún se mantiene con vigencia el esquema de ponderación de términos tf-idf (Jones, 1972). El trabajo de Salton et al., (1975), reúne todas estas ideas definiendo el modelo clásico de los sistemas de recuperación de información.

En los años siguientes, la investigación se basó en la búsqueda de mejores esquemas de ponderación de términos. Un esquema muy popular llamado BM25 fue propuesto por Robertson et al., (1994). Este esquema, así como tf-idf, requiere coincidencia exacta de términos, lo cual los hace poco adecuados para los casos en que los términos de la consulta y de los documentos no coinciden.

Con el advenimiento del aprendizaje automático, la investigación en recuperación de información se centró en aplicar técnicas de aprendizaje automático para entrenar modelos que aprendan a ordenar documentos. Estos métodos, llamados *learning-to-rank*, habitualmente se basan en métodos de relevance feedback (Rocchio, 1971) o en modelos de ajuste de hiperparámetros de métodos tradicionales de recuperación de información (Taylor et al., 2006).

Un hito importante en el Procesamiento del Lenguaje Natural provoca un cambio de enfoque en el desarrollo de modelos: los vectores densos. Los

embeddings o vectores densos son vectores de números reales que representan palabras o documentos. Estas representaciones encapsulan el significado, de tal manera que palabras o documentos con significados similares, se encuentran cercanos en el espacio vectorial de las representaciones. Los métodos para generar los vectores densos incluyen redes neuronales, métodos de reducción de dimensionalidad, modelos probabilísticos, etc.. La posibilidad de representar documentos y consultas con vectores densos y no con conteo de términos, permite el desarrollo de dos tipos de modelos. Los modelos basados en representaciones se basan en aprender independientemente los vectores que representan documentos y consultas, para luego comparar su similitud con la distancia coseno (Mitra et al., 2016). Los modelos basados en interacciones comparan representaciones de términos de las consultas y documentos generando una matriz, sobre la que se aplica un proceso para calcular el puntaje de relevancia del documento respecto a la consulta (Guo et al., 2016).

Los enfoques más recientes se centran en la utilización de la arquitectura Transformers (Vaswani et al., 2017). BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), uno de los principales ejemplos de la utilización de la arquitectura Transformers, es un modelo de red neuronal para generar vectores densos contextuales de las secuencias de entrada en inglés. BERT considera tanto el contexto anterior como el posterior de cada token de entrada durante el procesamiento. Esto permite que el modelo capture mejor el significado y las relaciones de las palabras dentro del contexto más amplio del texto. El resultado del procesamiento de BERT es una secuencia de vectores densos contextuales. Cada vector denso contextual es la representación de un token en función de su contexto en la secuencia de entrada. Estos vectores densos contextuales son ricos en información semántica y capturan las relaciones sintácticas y semánticas entre las palabras en el texto. [1]. De esta manera el modelo logra capturar características complejas del lenguaje como sintaxis, semántica y polisemia. En la figura 1.1 puede verse un esquema de las entradas y las salidas de la arquitectura BERT.

Dentro de los métodos que utilizan BERT, encontramos los métodos multietapa (Nogueira y Cho, 2020), donde se utiliza BERT para una etapa de reranking de los documentos ordenados por un método clásico de recuperación

[1]En contraste con las representaciones dependientes del contexto, están las representaciones estáticas de tokens como Word2vec (Mikolov et al., 2013) o GloVe (Pennington et al., 2014)

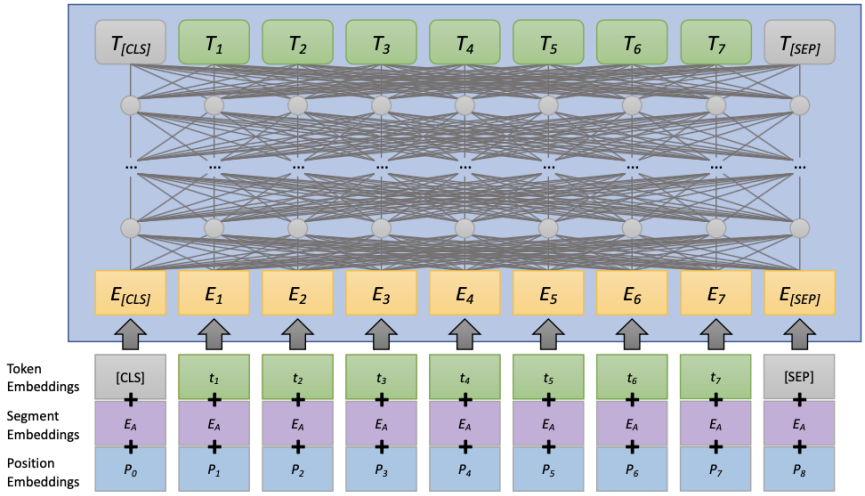

Figura 1.1: Esquema de entradas y salidas de la arquitectura BERT. Los vectores de entrada comprenden la suma de los embeddings que representan los tokens, segmentos y posiciones. La salida de BERT es una embedding contextual para cada token de entrada. El embedding contextual del token [CLS] se suele tomar como una representación agregada de toda la secuencia (imagen tomada de (Lin et al., 2021)).

de información. Como alternativa a la arquitectura multietapa se plantea el cálculo previo de embeddings para los documentos, utilizando como encoder un modelo basado en BERT. Al momento de inferencia se calcula el embedding de la consulta utilizando como encoder un modelo basado en BERT. Para ordenar se toma el producto interno entre los vectores de los documentos y el de la consulta. Este tipo de métodos se llaman bi-encoders y un ejemplo exitoso es el sistema DPR (Karpukhin et al., 2020).

Sin embargo, la utilización de estas arquitecturas, que hoy en día son el estado del arte en recuperación de información, está restringida al idioma inglés ya que BERT es un modelo del lenguaje del inglés. Para su utilización en español es necesario utilizar un modelo que siga la misma arquitectura que BERT, pero que haya sido entrenado en idioma español. Afortunadamente se cuenta con algunos modelos, como BETO (Cañete et al., 2020), RoBERTa (large y base) entrenado en el corpus de la Biblioteca Nacional de España (Gutiérrez-Fandiño et al., 2021) y multilingual BERT (mBERT) (Devlin et al., 2019). En el trabajo de estudio de modelos de lenguaje en español (Gutiérrez-Fandiño et al., 2021) se comparan dichos modelos sobre diferentes conjuntos de datos y diferentes tareas.

1.2. Objetivos

En esta tesis se estudiarán diferentes técnicas de recuperación de información para la búsqueda de respuestas en idioma español. Se trabajará en dos dominios diferentes de documentos y consultas con el fin de investigar potenciales diferencias de desempeño y cuestiones particulares al dominio.

El primer caso de estudio es el más usual e investigado, referido a documentos enciclopédicos como Wikipedia, con consultas de tipo deterministas. Estas preguntas normalmente tienen una única respuesta, como por ejemplo: «*¿En qué país se encuentra Normandía?*» o «*¿Cuál fue la capital de la dinastía Song?*». Este experimento es el más utilizado para benchmarking, por lo que pareció importante analizarlo y determinar la calidad de recursos disponibles en español para este problema.

El segundo caso de estudio es considerablemente menos estudiado independientemente del idioma, y refiere a recuperación de información sobre documentos de prensa. Este problema nace en el marco del proyecto CSIC [1] «Búsqueda de Respuestas a partir de Textos en español», presentado por el grupo de Procesamiento de Lenguaje Natural de la Facultad de Ingeniería, como parte de la línea Question Answering. Como dominio se eligió la enfermedad COVID-19 y hechos relacionados a la pandemia. En el marco de este proyecto, el equipo de trabajo construyó un corpus de preguntas y respuestas en español, a partir de un conjunto de noticias sobre COVID-19, el cual fue utilizado en este trabajo.

Los siguientes son los objetivos propuestos en esta tesis:

- Realizar un análisis en profundidad de diferentes técnicas de recuperación de información que son utilizadas en el contexto de búsqueda de respuestas y determinar su usabilidad para el idioma español.

- Estudiar la recuperación de información para la búsqueda de respuestas en documentos enciclopédicos en idioma español.

- Estudiar la recuperación de información para la búsqueda de respuestas en documentos de prensa en idioma español, analizando la forma de evaluación del sistema.

[1] Comisión Sectorial de Investigación Científica https://www.csic.edu.uy

1.3. Aportes

Las contribuciones de este trabajo son las siguientes:

1. Se realizó un análisis en profundidad de diferentes técnicas de recuperación de información que son utilizadas en el contexto de búsqueda de respuestas. El análisis incluye una descripción general de la evolución y el estado del arte de las diferentes técnicas, presentando sus fundamentos, fortalezas y debilidades así como las implementaciones prácticas.

2. Se reunieron seis conjuntos de datos para entrenar y probar el modelo de recuperación de Información para la búsqueda de respuestas en diferentes dominios. Además se reunieron 2 conjuntos de datos de documentos en español utilizados como fuente de información para la recuperación, los cuales también son de diferentes dominios. Se analizó en profundidad las características que deben tener los conjuntos de datos para su uso en IR.

3. Se adaptó el uso del módulo de recuperación de Información del sistema DPR (Karpukhin et al., 2020), para ser utilizado para el idioma español. Esto es importante ya que a la fecha de escritura de esta tesis, no se tenía a disposición una herramienta de recuperación de información que pudiese ser utilizada en idioma español, que hiciera uso de modelos basados en BERT.

4. Se realizó un análisis en profundidad de la recuperación de información para la búsqueda de respuestas en documentos de Wikipedia y en documentos de prensa. Se obtuvieron conclusiones muy útiles para avanzar en la mejora de la evaluación de IR para QA.

5. Se propuso un procedimiento de evaluación del modelo para la tarea de IR para QA en documentos de prensa. El procedimiento propuesto no tiene restricciones sobre las preguntas de evaluación. Los trabajos relacionados ajustan el conjunto de datos de evaluación para evitar ciertos problemas. Hasta donde llega nuestro conocimiento, aún no existía un procedimiento general de evaluación como el definido en este trabajo.

1.4. Organización del documento

El documento será organizado de la siguiente manera:

Capítulo 2: Fundamentos teóricos proporcionará una base teórica de los conceptos de Recuperación de Información, aplicado a la Búsqueda de Respuestas. Se mostrará el desarrollo del área a lo largo de los años, así como el desarrollo de la recuperación de información temporal (T-IR). Esto último es importante para el estudio y análisis de la tarea de IR para QA en documentos de prensa. También se expondrán las técnicas de evaluación y los recursos disponibles en inglés y español para llevar a cabo la tarea. Estos conceptos serán necesarios para comprender el resto del documento.

Capítulo 3: Recuperación de información para la búsqueda de respuestas en idioma español detallará los dos problemas estudiados en el marco de este trabajo y las características particulares de cada uno. Estos experimentos permitirán un análisis en profundidad del funcionamiento y la evaluación de los sistemas de IR para QA en diferentes dominios. En este capítulo también se presentará el módulo original elegido para la realización de los experimentos, así como los conjuntos de datos y los modelos del lenguaje utilizados para los experimentos.

Capítulo 4: Presentación y análisis de resultados presentará y analizará en profundidad los resultados de los experimentos realizados para cada dominio. Se estudiarán las causas que provocan los errores del sistema, así como el grado de ajuste del método de evaluación utilizado para la tarea estudiada. También se proporcionará un procedimiento manual de evaluación para ser utilizados sobre conjuntos de datos de prensa o con características temporales.

Capítulo 5: Conclusiones y trabajos a futuro expondrá las conclusiones obtenidas, los desafíos y los caminos futuros de investigación.

Capítulo 2

Fundamentos teóricos

Este capítulo presentará un estudio de los diferentes enfoques de IR para la tarea de Ad Hoc Retrieval y su aplicación en el contexto de la búsqueda de respuestas. Se profundizará el marco teórico presentando los métodos de evaluación, las diferentes métricas utilizadas, así como recursos disponibles tanto en inglés como en español para poder llevar a cabo el ciclo completo de IR-QA.

2.1. Arquitectura clásica Ad Hoc Retrieval y mejoras

La arquitectura clásica para la recuperación de información consiste en modelar la consulta y los documentos como vectores basados en conteos de unigramas, y luego ordenar los documentos según la distancia coseno entre los vectores (Salton, 1971). Este es el primer esquema completo definido de recuperación de información para la tarea de Ad Hoc Retrieval.

Se puede detectar entonces dos subproblemas: el primero consiste en encontrar todos los documentos que contengan al menos un término de la consulta. El segundo consiste en ordenar los documentos para devolverlos en orden de relevancia para la consulta dada.

Para la búsqueda de documentos que contengan al menos un término de la consulta, se utiliza un índice invertido. Esta estructura de datos consiste en un diccionario de términos, donde cada término apunta a una lista de IDs de documentos que lo contienen.

Para el ranking de documentos se utiliza algún esquema de pesos de térmi-

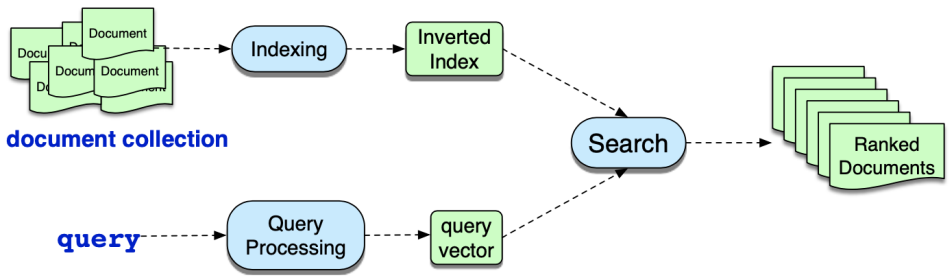

Figura 2.1: Arquitectura de un sistema IR de tipo Ad hoc. Las consultas y los documentos se representan como vectores basados en conteos de unigramas. Se ordenan los documentos según la distancia coseno entre éstos y la consulta. Imagen tomada de Jurafsky y Martin, 2020.

nos de la consulta, típicamente tf-idf (Term Frequency-Inverse Document Frequency)(Jones, 1972). La técnica tf-idf devuelve un valor numérico que representa la importancia relativa del término en el contexto de la colección de documentos. Se compone de dos componentes: tf calcula la frecuencia con la que un término específico aparece en un documento. Cuanto más frecuente sea el término en el documento, mayor será su valor tf. idf calcula la rareza de un término en la colección de documentos. Los términos que aparecen en pocos documentos tienen un valor idf más alto, lo que significa que se consideran más relevantes. La fórmula general para calcular tf-idf es la multiplicación de ambos competentes. Calculando esta medida para cada par consulta-documento, se realiza el ordenamiento.

Este modelo, ampliamente utilizado y eficiente, sufre de algunos problemas. No toma en cuenta información sintáctica, ni el orden de las palabras. Además no tiene herramientas para solucionar ambigüedades semánticas ni considerar sinonimia entre términos.

En los años siguientes, la investigación se basó en la búsqueda de mejores esquemas de ponderación de términos, para mejorar el método de ranking tf-idf. Un esquema muy popular llamado BM25 fue propuesto por Robertson et al., (1994).

Esta variación agrega dos parámetros: k ajusta balance entre frecuencia de término e idf mientras que el parámetro b controla la importancia de la normalización de los largos de los documentos. Los valores sugeridos son: $k \in [1.2, 2]$ y $b = 0.75$ (Manning et al., 2008). El puntaje del documento d dada una consulta q es:

$$\sum_{t \in q} \log\left(\frac{N}{df_t}\right) \frac{tf_{t,d}}{tf_{t,d} + k(1 - b + b\frac{|d|}{|d_{avg}|})} \tag{2.1}$$

Cuando $k = 0$, la puntuación BM25 es solo la suma del valor de *idf* para cada término de la consulta. El valor de b debe estar entre 0 y 1, indicando la importancia de la normalización en el cálculo del puntaje BM25.

Sin embargo, el método sufre de los mismos problemas que tf-idf: No toma en cuenta información sintáctica, ni el orden de las palabras. Además no tiene herramientas para solucionar ambigüedades semánticas ni considerar sinonimia entre términos.

2.1.1. Conteo de bigramas

Un trabajo destacado que hace uso de la arquitectura clásica con algunas mejoras es DrQA (Chen et al., 2017). DrQA es un sistema completo de IR-QA que utiliza como base de conocimiento los artículos de Wikipedia. Implementa el pipeline clásico, que consiste en una etapa de IR para obtener los documentos relevantes y luego un módulo de extracción de respuestas (en este trabajo implementado mediante redes neuronales).

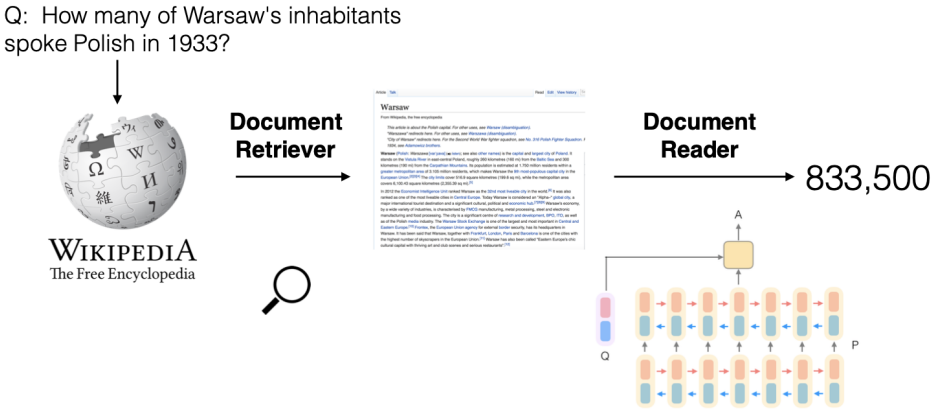

Figura 2.2: Pipeline de sistema DrQA. Consiste en una etapa de IR sobre el conjunto de documentos de Wikipedia, para obtener los documentos relevantes para la consulta (Document Retriever), seguido de una etapa de lectura y extracción de dichos documentos para obtener la respuesta (Document Reader). Imagen tomada de Chen et al., (2017).

Para la implementación del módulo de IR, los autores propusieron una variante del método clásico basado en tf-idf, utilizando n-gramas en lugar de

unigramas, para poder considerar el orden de las palabras. Para implementarlo utilizaron como estructura de datos para la búsqueda de documentos el Hash Murmurhash3 (Weinberger et al., 2010), en lugar de índice invertido. MurmurHash es un hash no criptográfico que sirve para búsquedas generales. El nombre viene de las dos operaciones básicas utilizadas en el ciclo interno: multiplicación (MU) y rotación (R). Este tipo de hash no fue diseñado para ser difícil de revertir, por eso no es adecuado para ser utilizado en criptografía. Murmurhash3 es la versión actual de MurmurHash.

Los autores evaluaron el desempeño del módulo IR contra los conjuntos de datos SQuAD (Rajpurkar et al., 2016), CuratedTREC (Baudiš y Šedivý, 2015), WebQuestions (Bordes et al., 2015) y WikiMovies (Miller et al., 2016). La tabla 2.1 compara el desempeño del motor de búsqueda de Wikipedia basado en ElasticSearch a la fecha de publicación, la recuperación de información utilizando conteo de unigramas (método clásico) y conteo de bigramas. La métrica utilizada corresponde al porcentaje de preguntas para las cuales alguna de las posibles respuestas aceptadas aparece en uno de los 5 primeros artículos devueltos.

Puede verse que el desempeño del sistema utilizando bigramas es en promedio mejor que los otros dos métodos. El trabajo indica también que se realizaron experimentos comparando el sistema contra BM25 y vectores densos, obteniéndose mejores resultados, pero no se muestran métricas.

Sistema	SQuAD	CuratedTREC	WebQuestions	WikiMovies
Wiki Search	62.7	81.0	73.7	61.7
DR unigramas	76.1	85.2	**75.5**	54.4
DR bigramas	**77.8**	**86.0**	74.4	**70.3**

Tabla 2.1: Resultados de WikiSearch y Document Retrieval. La métrica utilizada corresponde al porcentaje de preguntas para las cuales la respuesta aparece en uno de los 5 primeros documentos devueltos por el método.

Los resultados del sistema completo IR-QA en cada conjunto de datos pueden verse en la tabla 2.2. La métrica utilizada corresponde al porcentaje de respuestas devueltas por el sistema que coinciden exactamente con lo indicado como respuesta en el conjunto de datos.

Conjunto de datos	% Exact Match
SQuAD	29.8
CuratedTREC	25.4
WebQuestions	20.7
WikiMovies	36.5

Tabla 2.2: Resultados de DrQA sobre diferentes conjuntos de datos. La métrica utilizada corresponde a porcentaje de respuestas devueltas por el sistema que coinciden exactamente con lo indicado como respuesta en el conjunto de datos.

2.1.2. Expansión de consultas

Uno de los problemas de la arquitectura clásica de Salton es ignorar la sinonimia de los términos. Este problema se llama *vocabulary mismatch problem* (Furnas et al., 1987). Voorhees (1994) propone expandir las consultas agregando términos relacionados de forma léxica-semántica, utilizando wordNet. El modelo luego utiliza tf-idf e índice invertido. Los resultados no aportan beneficios significativos a la performance del sistema de IR.

Con el advenimiento del aprendizaje automático, la investigación en recuperación de información se centró en aplicar técnicas de aprendizaje automático para entrenar modelos que aprendan a ordenar documentos. Una rama de estos métodos llamados *learning-to-rank*, se basan en *Relevance feedback* (Rocchio, 1971).

Relevance feedback sirve como forma de expandir consultas. Se agregan términos de los documentos que se saben relevantes a la consulta gracias a un usuario que ya ha indicado previamente que el documento contiene la respuesta a la consulta. *Relevance feedback* mejora el rendimiento a costa de tener más de una etapa de recuperación de documentos en el sistema. En la primera recuperación, el usuario proporciona información sobre la relevancia de los documentos. En la siguiente iteración de recuperación, la consulta se mejora con esa información. *Pseudo-relevance feedback* (Croft y Harper, 1979) supone que los primeros documentos devueltos son relevantes y expande la consulta con términos de estos documentos. Un ejemplo de los métodos de *learning-to-rank* es el método RM3 (Jaleel et al., 2004).

2.2. Métodos con vectores densos pre-BERT

Al irrumpir Deep Learning con la posibilidad de representar palabras mediante vectores densos, surge a continuación la necesidad de representar porciones de texto más largas como oraciones y textos. Muchos investigadores adoptaron un enfoque jerárquico, componiendo representaciones de palabras para lograr representaciones de oraciones (Socher et al., 2013). Posteriormente varios trabajos se enfocaron en métodos de agregación simples para agregar representaciones de palabras en representaciones de textos más largas. Estas propuestas resultaron ser efectivas y fáciles de calcular. Algunas de las propuestas fueron: promedio ponderado de vectores densos con pesos aprendidos (Boom et al., 2016) y promedio ponderado de vectores densos seguido por una modificación con PCA (Arora et al., 2017).

Representando documentos y consultas con vectores densos y no con conteo de términos, dos tipos de modelos se desarrollaron.

Los modelos basados en representaciones se basan en aprender mediante ejemplos de entrenamiento los vectores que representan documentos y consultas, de forma independiente. Esto permite que las representaciones de documentos puedan hacerse previamente, y no en tiempo de inferencia. Al momento de la consulta, se computa la representación de la consulta y se calcula el puntaje de similitud documento-consulta (por ejemplo con la distancia coseno) para realizar el ranking. Un ejemplo de este tipo de modelo es el propuesto por Mitra et al., (2016).

Los modelos basados en interacciones comparan representaciones de términos de las consultas y documentos generando una matriz, para luego calcular el puntaje de relevancia del documento respecto a la consulta. El proceso implica la construcción de una matriz de similitud con filas correspondientes a los términos de consulta y columnas correspondientes a los términos del documento. Cada entrada de la matriz $m_{i,j}$ se completa con la similitud coseno entre la representación del término i-ésimo de consulta y la representación del j-ésimo término del documento. Luego, los modelos extraen características de la matriz, utilizando diferentes enfoques. Un ejemplo es el del modelo DRMM (Guo et al., 2016), que suma la similitud entre cada término de la consulta y documento. El último paso para calcular el puntaje de similitud implica combinar todas las características extraídas e introducirlas en una red neuronal para obtener el puntaje.

Se ha demostrado que los modelos basados en interacciones dan mejores resultados pero son más lentos que los modelos basados en representaciones.

2.3. Arquitectura de IR utilizando BERT

Actualmente las tareas de IR que involucran la utilización de vectores densos para la representación de consultas y documentos, se centran en la arquitectura Transformers (Vaswani et al., 2017), específicamente utilizando modelos tipo BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019).

BERT es un modelo de red neuronal que como resultado de un proceso de entrenamiento auto supervisado, aprende vectores densos contextuales de las secuencias de entrada en inglés. Fue entrenado con dos objetivos:

- Masked language modeling (MLM): tomando una oración, el modelo enmascara aleatoriamente el 15 % de las palabras en la entrada, luego procesa la oración enmascarada completa a través del modelo y tiene que predecir las palabras enmascaradas.
- Next sentence prediction (NSP): el modelo concatena dos oraciones enmascaradas como entradas durante el pre-entrenamiento. A veces corresponden a oraciones que estaban consecutivas en el texto original, a veces no. Luego, el modelo tiene que predecir si las dos oraciones eran consecutivas originalmente.

BERT toma como entrada una secuencia de representaciones derivadas de tokens y genera una secuencia de vectores densos contextuales, que proporcionan representaciones dependientes del contexto de los tokens de entrada. De esta manera el modelo logra capturar características complejas del lenguaje como sintaxis, semántica y polisemia. La característica distintiva de BERT respecto a otros modelos es que utiliza las mejoras provistas por la arquitectura Transformers, como el mecanismo de attention para capturar relaciones de largo alcance en el texto, en lugar de otras redes neuronales, como LSTM.

El primer token de cada secuencia de entrada a BERT es un token especial llamado [CLS]; la representación de salida de BERT de este token especial se suele tomar como una representación de toda la secuencia. El token

[CLS] va seguido de la representación de las entradas: por ejemplo, los tokens de una oración. Si la tarea involucra más de una entrada, como es el caso de la Recuperación de información (documento y consulta) se agrega un token especial [SEP] entre cada entrada. Toda la secuencia finaliza con un token [SEP]. En la figura 2.3 puede verse un ejemplo donde la tarea requiere la entrada de la consulta y el documento. La entrada a BERT es entonces la secuencia de tokens: $E_{[CLS]}, E_1, E_2, E_3 E_{[SEP]}, F_1 ... F_m, E_{[SEP]}$. La salida será una representación contextual de cada uno de los tokens de entrada: $T_{[CLS]}, U_1, U_2, U_3 T_{[SEP]}, V_1 ... V_m, V_{[SEP]}$.

2.3.1. Enfoque multietapa

El uso de BERT en este enfoque se utiliza para estimar un puntaje de relevancia de un documento respecto a una consulta. Los vectores de entrada del modelo son la representación de la consulta, el documento y el vector [CLS], que normalmente resume toda la representación de la entrada. Durante el entrenamiento de BERT, el vector espacial [CLS] se entrena para capturar información contextual y semántica sobre toda la secuencia. Entonces, se considera que encapsula la información global de la entrada y se utiliza como una representación de la misma. La salida del modelo correspondiente a la entrada [CLS] se introduce en una capa totalmente conectada, lo que produce un puntaje de relevancia para el documento con respecto a la consulta.

Puede verse en la figura 2.3 un esquema de esta arquitectura llamada monoBERT (Nogueira y Cho, 2020). DuoBERT (Nogueira et al., 2019) extiende monoBERT, formulando el problema de ranking como clasificación por pares de documentos. Yan et al., (2019) usa el enfoque multietapa y agrega en el primer paso del pipeline generación de consultas equivalentes para realizar expansión de los documentos de forma de subsanar los problemas de Vocabulary Mismatch. Multipassage BERT (Z. Wang et al., 2019) sigue esta arquitectura normalizando puntajes de preguntas.

Uno de los problemas de BERT para IR es el tamaño máximo de tokens de entrada (512), lo cual dificulta la clasificación de documentos de mayor tamaño. Para ello existen técnicas para particionar los documentos al momento de entrenamiento e inferencia.

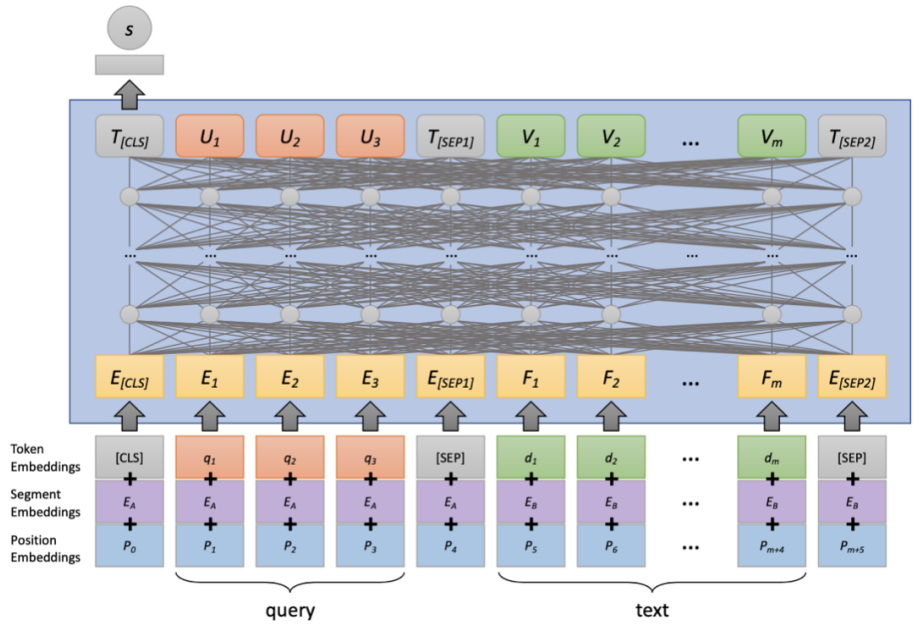

Figura 2.3: Arquitectura monoBERT. Las entradas son las representaciones de la consulta y el texto (como suma de los embeddings que representan los tokens, segmentos y posiciones). La salida del modelo BERT es el embedding contextual para cada token de entrada. La representación final del token [CLS] se envía a una capa completamente conectada que produce la puntuación de relevancia de ese texto con respecto a la consulta. Imagen tomada de Lin et al., (2021).

Computacionalmente es muy costoso realizar estos cálculos para cada combinación consulta-documento, si la colección de documentos es muy grande. Se quiere conseguir un balance entre la efectividad del sistema (referida a la calidad de la información devuelta), así como eficiencia (referida al tiempo de respuesta de la tarea), por lo que aplicar una solución que involucre solamente BERT no satisface el objetivo de eficiencia.

Es por eso que normalmente se utiliza BERT como una etapa de reranking de documentos, luego de la aplicación de los métodos clásicos o variantes sobre la colección original.

La primer etapa de este enfoque es un paso de IR mediante el método tradicional o variantes, la cual tiene como salida una lista ordenada de documentos. En una segunda etapa se seleccionan los primeros X documentos de la lista devuelta. Para cada documento, se introduce en BERT los tokens del par consulta-documento como se muestra en la figura 2.3, produciendo un puntaje de relevancia para el par. Con los nuevos puntajes de relevancia para cada documento, se reordena la lista original. De esta manera tendremos un ordenamiento de documentos más acertado que el devuelto por la primer etapa

de IR, y el costo computacional de la etapa de reranking se verá reducido, ya que se realizará sobre un conjunto de documentos acotado. Esta arquitectura puede agrandarse, aplicando sucesivas etapas de reranking, pero es necesario tener en cuenta el costo computacional de la tarea. Puede verse en el diagrama 2.4 la arquitectura del enfoque multietapa.

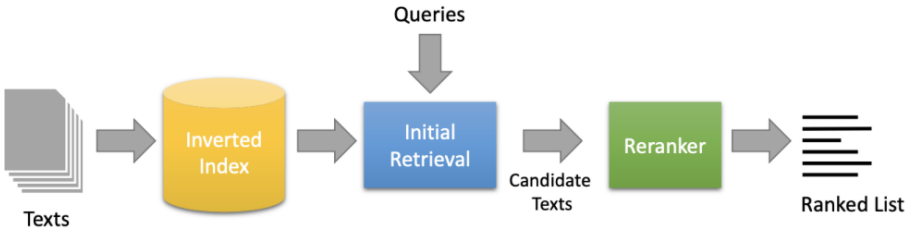

Figura 2.4: Diagrama de una arquitectura multietapa. Para la generación de candidatos se utilizan los métodos clásicos y luego esos documentos son reordenados con un modelo basado en BERT. Imagen tomada de Lin et al., (2021).

Con esta arquitectura, al tener una primer etapa donde se seleccionan documentos utilizando el método clásico, se mantiene el problema de sinonimia. De esta manera, se podría estar dejando fuera documentos relevantes en la primera etapa, que luego no serían considerados por BERT.

Es inusual que un documento relevante no tenga algún término de la consulta, por lo que una posible solución es agrandar lo suficiente el conjunto devuelto por la primera etapa, para que luego BERT realice un reordenamiento de los documentos. Esta solución se puede aplicar hasta cierto punto, ya que aumenta el tiempo de respuesta de la etapa de reranking.

Otras propuestas incluyen la expansión de consultas (tal como se vio en el punto 2.1.2) o la expansión de documentos, que implica agregar términos relacionados a los documentos. Esta última técnica puede realizarse en paralelo sobre cada documento, y no en tiempo de consulta, por lo que no agrega latencia en la tarea. Sin embargo, al no saber de antemano la consulta, la elección de términos a agregar a los documentos puede ser difícil.

2.3.2. Enfoque con bi-encoders

Como alternativa a la arquitectura multietapa, se plantea el cálculo previo de embeddings para los documentos, utilizando como encoder un modelo basado en BERT. Luego, al momento de la consulta, se calcula el embedding

de ésta también utilizando como encoder un modelo basado en BERT. Una vez se tiene la representación de los documentos y las consultas, es necesario hallar los más relevantes para devolver una lista ordenada de documentos. Para esto se utiliza como función de comparación el producto interno (u otras ditancias, como la distancia coseno) entre el vector que representa la consulta y los vectores que representan los documentos. Puede verse en la figura 2.5 un diagrama de la arquitectura.

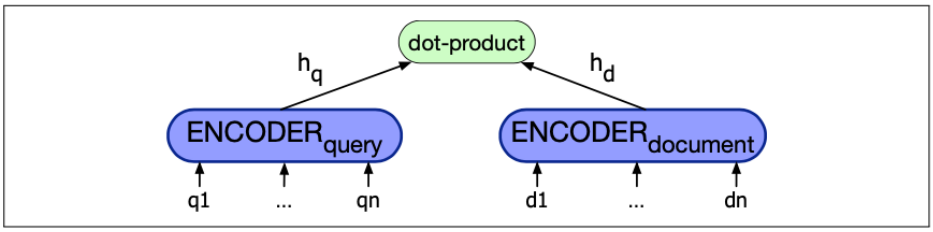

Figura 2.5: Diagrama de bi-encoder. Se codifican documentos y consultas con encoders diferentes, basados en BERT. Para ordenar se toma el producto interno entre los vectores los vectores de los documentos y el de la consulta. Imagen tomada de Jurafsky y Martin (2020).

Esta Arquitectura se propone en SentenceBERT (Reimers y Gurevych, 2019). Algunos trabajos que se basan en esta arquitectura son ORQA (Lee et al., 2019), CLEAR (Gao et al., 2021) que complementa con bi-encoders el retriever clásico con BM25. Uno de los trabajos en el que profundizaremos es Dense Passage Retriever (DPR) (Karpukhin et al., 2020).

Al usar este enfoque, se debe encontrar la representación de documentos más similares a la representación de la consulta. Si la colección de documentos es grande, para que el problema sea computacionalmente realizable, se realiza una aproximación de los vecinos más cercanos. Los algoritmos más populares para resolver este problema son los algoritmos basados en grafos HNSW (Malkov y Yashunin, 2018) y los basados en Product quantization (PQ) (Jégou et al., 2011) como los usados en FAISS (Johnson et al., 2017).

Dense Passage Retriever (DPR) (Karpukhin et al., 2020) es un ejemplo de uso de este enfoque con aproximación KNN. DPR aplica de forma previa un encoder entrenado basado en BERT a todos las porciones de documentos de la colección y se indexan utilizando FAISS. Esta biblioteca almacena y organiza los vectores densos para permitir una búsqueda eficiente. Se pueden crear diferentes tipos de índices, como el índice de vecinos más cercanos o

índices exactos.

En tiempo de inferencia, se aplica otro encoder basado en BERT a la consulta y se calculan los k elementos más cercanos indexados previamente. Se utiliza como función de similitud el producto interno de los vectores. Los autores estudiaron otras funciones de similitud, como la distancia coseno y la distancia L2, pero éstas tenían un rendimiento similar al del producto interno, pero son más complejas de calcular.

Los autores mostraron que su enfoque de IR, seguido por un componente de búsqueda de respuestas basado en BERT, alcanza el estado del arte en varios conjuntos de datos de referencia. En las tablas 2.3 y 2.4 se muestra el desempeño del módulo de IR y del pipeline completo IR-QA respectivamente.

En el trabajo también se muestra cómo la selección de ejemplos de entrenamiento negativos impacta en la efectividad de la recuperación. Otros trabajos que estudian el módulo DPR son RocketQA (Qu et al., 2021) y el estudio planteado por Ma et al., (2021).

Training	Retriever	Top 20				
		NQ	TriviaQA	WQ	TREC	SQuAD
None	BM25	59.1	66.9	55.0	70.9	68.8
Single	DPR	78.4	79.4	73.2	79.8	63.2
Single	BM25 + DPR	76.6	79.8	71.0	85.2	**71.5**
Multi	DPR	**79.4**	78.8	**75.0**	**89.1**	51.6
Multi	BM25 + DPR	78.0	**79.9**	74.7	88.5	66.2

Tabla 2.3: Resultados de desempeño de DPR. La métrica utilizada corresponde al porcentaje de preguntas para las cuales la respuesta aparece en uno de los 20 artículos devueltos por el método. Single and Multi indican si DPR fue entrenado usando solo un conjunto de datos o una combinación de todos salvo SQuAD. BM25 + DPR indica una combinación lineal de los puntajes de ambos métodos. Tabla tomada de Karpukhin et al., 2020.

2.4. Recuperación de información temporal

Uno de los objetivos de este trabajo es estudiar la recuperación de información para la búsqueda de respuestas en documentos de prensa en idioma español. Realizar la tarea sobre el dominio de prensa conlleva desafíos adicionales al problema tradicional de recuperación de información sobre documentos de Wikipedia (en las siguientes secciones 2.5 y 2.6 se explica en detalle este

Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2.
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	**56.5**
Single	REALM (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	BM25	32.6	52.4	29.9	24.9	38.1
Single	DPR	**41.5**	56.8	34.6	25.9	29.8
Single	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	**41.5**	56.8	**42.4**	49.4	24.1
Multi	BM25 + DPR	38.8	**57.9**	41.1	**50.6**	35.8

Tabla 2.4: Resultados de Sistema completo QA basado en DPR sobre diferentes conjuntos de datos. La métrica utilizada corresponde Average Exact Match. Single and Multi indican si DPR fue entrenado usando solo un conjunto de datos o una combinación de todos salvo SQuAD. BM25 + DPR indica una combinación lineal de los puntajes de ambos métodos. Tabla tomada de Karpukhin et al., 2020.

problema). Las consultas y los documentos contienen información temporal, que es necesario explotar para poder devolver los documentos relevantes a la consulta de un usuario.

Para ejemplificar supongamos que el conjunto de documentos contiene dos artículos con la información indicada en la tabla 2.5

ID	Timestamp	Contenido documento
1	20-3-2020	*Hoy hubo 3 contagios por COVID.*
2	21-3-2020	*Hoy hubo 2 contagios por COVID.*

Tabla 2.5: Ejemplos de documentos de una colección de Prensa.

Si un usuario consulta «*¿Cuántos contagiados hubo hoy por COVID?*», Dependerá del día en que fue realizada la consulta para determinar cuál de los dos documentos es el que contiene la respuesta. Si por el contrario el usuario consulta «*¿Cuántos contagiados hubo el 20 de marzo del 2020 por COVID?*», el documento relevante deberá ser el primero. Pero para que la clasificación se realice de forma correcta, se debe tener en cuenta información temporal de la consulta, de los documentos, así como metadatos de todas las entidades.

El propósito de la recuperación de información temporal (T-IR) es mejorar la efectividad de los métodos de recuperación de información mediante la explotación de información temporal en documentos y consultas. En general, se combina la relevancia del documento con la relevancia temporal para ordenar

los documentos devueltos.

Uno de los primeros trabajos en hacer un resumen del estado del arte de T-IR es el de Alonso et al., (2011). En este trabajo se definen conceptos y se describen algunos de los desafíos del área para resolver problemas relacionados a la temporalidad. En 2011, cuando fue publicado el artículo, aún no existía BERT y muchas de las técnicas descritas no son aplicables a las arquitecturas más recientes. Sin embargo, sienta las bases sobre algunos conceptos que pueden ser de utilidad.

La fecha de creación de un documento se encuentra en su metadata y es información particularmente importante en el dominio de las noticias. Pero también dentro del documento existen muchas referencias temporales, las cuales aportan mucha información para resolver consultas. El etiquetado temporal es una tarea específica en el reconocimiento y normalización de entidades con nombre. Los objetivos de los tags temporales son la extracción de expresiones temporales y la normalización de estas expresiones a algún formato estándar. Luego, se utiliza la extracción de relaciones temporales para QA, clasificación, así como para IR. En el caso de QA e IR, también es importante tener en cuenta que existe información temporal en la consulta. Los usuarios pueden requerir documentos que describan el pasado, documentos que contengan la información más reciente, o información relacionada con el futuro. Las técnicas de tagging son aún las más utilizadas (Campos et al., 2014).

Luego del advenimiento de BERT, destacan dos trabajos de IR-QA sobre artículos de noticias del período 1987-2007 (Jiexin et al., 2021; J. Wang et al., 2020). Las preguntas sobre los artículos a menudo están relacionadas con eventos. Este sistema completo de QA utiliza BERT solamente para el módulo de extracción de respuestas pero no para el módulo de IR, donde se utiliza BM25. El sistema implementado (QANA) cuenta con un módulo adicional entre el módulo de IR y el de extracción de respuestas, para reordenar los documentos mediante la explotación de información temporal, como puede verse en la Figura 2.6.

Los resultados presentados por los autores son del sistema completo IR-QA. Cuando el sistema utiliza los 5 primeros documentos devueltos para buscar respuestas, la métrica de Exact Match es de 24.40 %.

Respecto al conjunto de datos utilizado para evaluación, los autores crearon uno con preguntas a partir de otros conjuntos de datos y de pruebas de historia. Para la primera versión del sistema utilizaron un conjunto de datos de 200

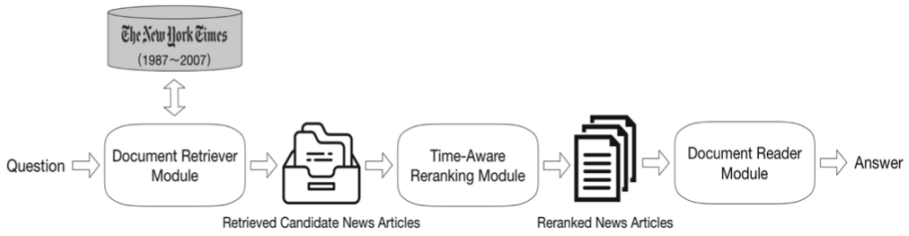

Figura 2.6: Arquitectura de QANA. Cuenta con un módulo adicional entre el módulo de IR y el de extracción de respuestas, para reordenar los documentos mediante la explotación de información temporal. Imagen tomada de J. Wang et al., (2020).

preguntas, donde se aseguraron que la respuesta estuviese contenida en alguno de los documentos del conjunto de datos de noticias. Para la segunda versión del sistema crearon un conjunto de datos de 1000 preguntas, pero en este caso no verificaron la pertenencia de la respuesta a algún documento del conjunto de noticias.

Algunas de las preguntas y sus respuestas pueden verse en la tabla 2.6. Se observa que las preguntas son deterministas, es decir, tienen casi siempre una sola posible respuesta. Si bien el sistema QANA trabaja con noticias, las preguntas del conjunto de datos de evaluación son relacionadas a eventos históricos, y están expresadas con tal completitud que no admiten una respuesta diferente a la indicada en el conjunto de datos.

Pregunta	Respuesta
Which sheep was the first cloned mammal?	*Dolly*
Which role does Robin Williams play in the movie hook?	*Peter Pan*
Who won the Booker Prize in 2002?	*Yann Martel*
In which state did the Acteal massacre of 1997 occur?	*Chiapas*
Which city hosted the 1998 Winter Olympics?	*Nagano*

Tabla 2.6: Ejemplos de preguntas utilizadas para evaluar QANA.

Como último trabajo relacionado al T-IR se tiene TimeBERT (J. Wang et al., 2022). TimeBERT es un modelo del lenguaje entrenado en una colección temporal de artículos de noticias, con el objetivo de incorporar información temporal y mejorar el desempeño en tareas que requieran hacer uso de esta información. Los autores remplazan el módulo de re-ranking temporal de QANA por TimeBERT. Los resultados indican que cuando el sistema utiliza los

5 primeros documentos devueltos para buscar respuestas, la métrica de Exact Match es de 29.20 %, lo cual supera a QANA original.

T-IR es un área en donde hay aún mucho por investigar. Pueden distinguirse dos líneas aún no muy exploradas. Por un lado, todavía no se ha incorporado completamente la temporalidad en la recuperación de información con arquitecturas basadas en BERT, las cuales son el estado del arte para la recuperación de información. La aproximación más cercana y muy reciente se da con TimeBERT, donde se utiliza como un módulo de re-ranking (como la arquitectura multietapa), y no como el mecanismo principal para la recuperación. Por otro lado, no hay a la fecha ningún trabajo que plantee los problemas reales de evaluación de la tarea, sino que se tienen conjuntos de datos de evaluación cuidadosamente elegidos para poder realizar un cálculo de métricas de forma automática.

En este trabajo de tesis se pretende trabajar con consultas que se asemejen a las que una persona pudiese realizar. El ejemplo de la tabla 2.5, muestra un caso de estudio cuyos documentos y consultas son extremadamente sencillos, pero que plantea problemas aún no resueltos para la evaluación de sistemas de IR para QA en un dominio con características de temporalidad.

En la próxima sección se explica cómo se realiza la evaluación de sistemas de IR para QA en los trabajos de investigación.

2.5. Evaluación

Los sistemas de QA tienen como objetivo, dada una pregunta, extraer la respuesta de un documento perteneciente a una colección potencialmente grande. El pipeline clásico contiene un módulo de IR que selecciona los documentos relevantes a la pregunta, y otro de comprensión de texto para extraer la respuesta de los documentos seleccionados.

En las siguientes secciones se explicarán los métodos de evaluación de cada uno de los módulos.

2.5.1. Método de evaluación

La evaluación del módulo de IR, del de extracción de respuestas y del sistema completo, se realiza utilizando los mismos corpus. Se explicará como

Contexto	«*Uruguay, oficialmente República Oriental del Uruguay, es un país soberano de América del Sur, situado en la parte oriental del Cono Sur. Abarca 176 215 km² y es el segundo país más pequeño de Sudamérica, después de Surinam.*»
Pregunta	*¿Cuál es el nombre oficial de Uruguay?*
Indicador	Verdadero
Índices	[22,52]
Respuesta	*República Oriental del Uruguay*

Tabla 2.7: Ejemplo de conjunto de datos de evaluación.

se utilizan los corpus para evaluar el desempeño de los módulos.

Módulo de extracción de respuestas

El módulo de extracción de respuestas tiene como entrada una porción de texto y una pregunta. Lo esperado como salida es una bandera que indique si se pudo encontrar respuesta en el texto, y en caso afirmativo, los índices de inicio y fin dentro de éste. Para evaluar el módulo se debe disponer de un conjunto de datos donde cada pregunta disponga de un contexto (al menos un párrafo de un documento), un indicador para saber si hay respuesta o no dentro del contexto, y los índices de la posición de la respuesta en el contexto. El sistema es exitoso para una pregunta cuando la bandera y los índices devueltos coinciden con los indicados en el conjunto de datos.

Supongamos que se tiene un ejemplo de evaluación como el mostrado en la tabla 2.7. El módulo de extracción de respuestas será exitoso si ante la pregunta «*¿Cuál es el nombre oficial de Uruguay?*» devuelve la bandera *Verdadero* y los índices de inicio y fin [22,52]. Si devuelve los índices [9,52] no será exitoso ya que si bien dentro de la respuesta devuelta, se encuentra la respuesta indicada en el conjunto de evaluación, el módulo no devolvió el span esperado.

Módulo de IR

Para la evaluación del módulo de IR, será necesario disponer de una técnica para clasificar los documentos como relevantes o no relevantes, ante una consulta. En el contexto de IR para QA, se utiliza el mismo conjunto de datos descrito anteriormente, donde un documento es relevante para la consulta si

Contexto	«*Suárez tiene 36 años*»
Pregunta	*¿Cuántos años tiene Suárez?*
Indicador	Verdadero
Índices	[13,14]
Respuesta	*36*

Tabla 2.8: Ejemplo de conjunto de datos de evaluación.

contiene en el texto la respuesta a ésta. Aquí el problema se simplifica, y no se utilizan los índices ni el contexto, lo que puede llevar a una clasificación conceptualmente errónea.

Por ejemplo, supongamos el conjunto de datos de evaluación de la tabla 2.8 y la colección de documentos de la tabla 2.9. En este ejemplo, ambos documentos deberían ser catalogados como relevantes para el módulo de IR para la consulta «*¿Cuántos años tiene Suárez?*» aunque claramente el segundo documento no lo es.

ID	Contenido documento
1	«*Suárez tiene 36 años porque nació el 24 de enero del 1987.*»
2	«*La máxima temperatura en Noruega fue de 36 grados.*»

Tabla 2.9: Ejemplos de documentos de una colección.

El módulo de IR devuelve un listado ordenado de documentos que son clasificados con la técnica descrita anteriormente como relevantes o no para la consulta dada. Luego, se utilizan las métricas descritas en la siguiente sección.

Para evaluar el sistema completo, también se utiliza el mismo conjunto de datos y la simplificación respecto a no utilizar los índices. El sistema es exitoso para una pregunta cuando el texto devuelto como respuesta es igual al indicado en el conjunto de datos.

2.5.2. Medidas de evaluación de IR para QA

A continuación se verán métricas con diferentes capacidades para la evaluación de la tarea de IR en el contexto de QA.

Top k

Esta métrica es la utilizada en la mayoría de los trabajos aquí nombrados, y corresponde al porcentaje de consultas para las cuales existe al menos un documento relevante en los primeros k devueltos. Esta métrica no indica la calidad del orden de los documentos devueltos.

Las tres siguientes métricas pueden utilizarse para evaluar también el orden de los documentos devueltos, aunque no son utilizadas normalmente en los artículos.

Interpolated Precision

Interpolated Precision se calcula como el máximo valor de precisión alcanzado para cualquier valor de Recall igual o menor al que se está calculando.

Precisión en este contexto es la porción de documentos relevantes devueltos del total de documentos devueltos. Recall es la porción de documentos relevantes devueltos, del total de documentos relevantes.

$$InterPrecision(r) = \max_{i \leq r} \text{Precision}(i) \tag{2.2}$$

Para comparar los sistemas, se comparan las curvas de la gráfica Recall-InterpolatedPrecision, siendo mejores los sistemas que tienen mayor área bajo la curva. Esta evaluación considera que la relevancia es binaria, es decir: un documento es relevante o no.

Mean Average Precision

Para una consulta, Average Precision (AP) promedia las medidas de precisión considerando solamente los documentos relevantes listados, para cierto límite de listado r. Mean Average Precision promedia todos los AP de las consultas realizadas al sistema para su evaluación. Esta evaluación considera que la relevancia es binaria, es decir: un documento es relevante o no.

$$AP = \frac{1}{|R_r|} \sum_{d \in R_r} Precision_r(d) \tag{2.3}$$

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q) \tag{2.4}$$

Normalized Discounted Cumulative Gain (nDCG)

Esta métrica permite tomar en cuenta relevancia en escalas, y no solamente binaria. Considera el grado de relevancia del documento y también la posición en el ranking.

$$DCG(R,q) = \sum_{(i,d)\in R} \frac{2^{rel(q,d)} - 1}{\log_2(i+1)} \tag{2.5}$$

IDCG representa la lista ideal de documentos ordenados. La métrica DCG normalizada es entonces la siguiente, estando sus valores entre 0 y 1:

$$nDCG(R,q) = \frac{DCG(R,q)}{IDCG(R,q)} \tag{2.6}$$

2.5.3. Medidas de evaluación del módulo de extracción de respuestas y pipeline completo

A continuación veremos las métricas más usuales para la evaluación del módulo de extracción de respuestas.

Average Exact Match

Para la evaluación del módulo de extracción de respuestas, para cada consulta, si los índices de inicio y fin de la respuesta devuelta coinciden con los indicados en el conjunto de datos de evaluación EM=1, en caso contrario EM=0. Para el caso del pipeline completo, EM=1 si el span is (el texto) devuelto es igual al indicado en el conjunto de datos de evaluación. Average Exact Match promedia los valores de EM para cada consulta del conjunto de evaluación.

Average F1

Para esta métrica se calcula la precisión y el recall en base a los tokens de la predicción frente a la respuesta real. La cantidad de tokens compartidos entre la predicción y respuesta real es la base de la puntuación F1: la precisión es la relación entre la cantidad de tokens compartidos y la cantidad total de tokens en la predicción. El recall es la relación entre la cantidad de tokens compartidos y la cantidad total de tokens en la respuesta real.

$$F1 = 2.\frac{Precision.Recall}{Precision + Recall} \tag{2.7}$$

Average F1 promedia los valores de F1 para cada consulta del conjunto de evaluación.

2.6. Recursos

Una vez descrito el marco teórico, y la evolución de las técnicas de IR para Ad Hoc Retrieval, en esta sección se expondrán los recursos necesarios y disponibles para llevar a cabo la tarea.

El objetivo del sistema completo IR-QA es encontrar respuestas a preguntas en documentos, por lo que los recursos indispensables son los documentos, y un conjunto de datos de evaluación como se describió en la sección 2.5. En la siguiente sección se profundiza en estos conjuntos de datos, detallando los recursos disponibles para investigación, tanto en inglés como en español.

2.6.1. Conjuntos de datos

Uno de los Conjuntos de datos requeridos para llevar a cabo la tarea de recuperación de información para la búsqueda de respuestas es el corpus de documentos. Los documentos son la fuente de información para la búsqueda de respuestas. La finalidad del sistema a construir debe ser coherente con el tipo de documentos y la información que éstos contienen. Es decir, si se desea construir un sistema que busque respuestas sobre hechos históricos, entonces los documentos deben hablar sobre esta temática, para que el sistema sea capaz de encontrar las respuestas. Por otro lado, si por ejemplo, se desea que el sistema busque respuestas sobre noticias, entonces los documentos deberían ser artículos de prensa.

En un buscador web, los documentos son todas las páginas indexadas por los buscadores. Al realizar diferentes preguntas, puede verse que Google en algunos casos realiza una búsqueda de respuestas, pero en otros solamente puede devolver una lista ordenada de documentos (páginas web). Si se pregunta: «*¿Cuántos casos de la viruela del mono hubo el 31 de agosto del 2022 en Francia?*» el buscador devuelve una lista de páginas relevantes, pero no contesta la pregunta. Sin embargo, si se pregunta «*¿Quien fue Juana de Ibarbourou?*», el buscador devuelve un párrafo de una página web, resaltando en negrita la respuesta:

«También conocida como Juana de América, ***fue una poetisa reconocida como una de las voces más personales de la lírica hispanoamericana de principios del siglo XX****. Juana Fernández Morales, quien se convertiría más tarde en Juana de Ibarbourou, nació en Melo, Cerro Largo, el 8 de marzo de 1892».*

Para la investigación, es muy común, y casi exclusiva la utilización de documentos de Wikipedia [1]. En ocasiones se utilizan los documentos de Wikipedia depurados con herramientas disponibles como wikiextractor (Attardi, 2015), o particionados. Los dumps de Wikipedia pueden ser descargados por idioma, lo que pone a disposición la información independientemente del lenguaje. Wikipedia cuenta con más de seis millones y medio de artículos en inglés, siendo éste el idioma que cuenta con más artículos en la plataforma. En español existen casi un millón ochocientos mil artículos, siendo el sexto idioma con mayor cantidad de artículos en Wikipedia [2].

Para la investigación de recuperación de información temporal, suelen usarse documentos de prensa. Uno de ellos es el New York Times Anotated Corpus (Sandhaus, 2008), que cuenta con 1.8 millones de artículos publicados por el New York Times entre 1987 y 2007, escritos en inglés.

En el marco del proyecto CSIC «Búsqueda de Respuestas a partir de Textos en español», presentado por el grupo de Procesamiento de Lenguaje Natural de la Facultad de Ingeniería, como parte de la línea Question Answering, se crearon diversos conjuntos de datos. QuALES_Docs [3] es un corpus de 1250 artículos de prensa de los medios La Diaria y Montevideo Portal de noticias relacionadas al COVID-19, escritas en español.

Por otro lado, necesitamos información para evaluar el desempeño del sistema y, en algunos casos, también para entrenarlo. Tal como se describió en la sección 2.5 el conjunto de datos se compone de un listado de preguntas, donde cada pregunta dispone de un contexto (al menos un párrafo de un documento), un indicador para saber si hay respuesta o no dentro del contexto, y los índices de la posición de la respuesta en el contexto. En los últimos años se han construido varios corpus de este tipo (corpus de QA) en inglés.

[1] https://dumps.wikimedia.org
[2] Información extraída en setiembre 2022
[3] https://github.com/pln-fing-udelar/covid19-qa/tree/master/data/annotated_articles

Probablemente el más popular es **Stanford Question Answering Dataset (SQuAD)** (Rajpurkar et al., 2016). La primer versión del conjunto de datos contenía más de 100.000 pares de preguntas-respuestas. La versión 2.0 de SQuAD expande dicho conjunto de datos con 50.000 preguntas sin respuesta en el contexto dado, creadas de tal manera que fuesen similares a las preguntas que sí tenían respuesta en dicho párrafo.

Para la creación del conjunto de datos, los autores tomaron 536 artículos de los 10.000 artículos principales de Wikipedia en inglés. De cada uno de los 536 artículos se tomaron 23.215 párrafos. Para la anotación (realizada por los trabajadores de Mechanical Turk), para cada párrafo, se pidió a los anotadores que pensaran y respondieran 5 preguntas sobre el contenido del párrafo, seleccionando la respuesta contenida en éste. Realizar la anotación siguiendo esta metodología genera que muchas de las preguntas no cuenten con la información suficiente para ser respondidas, en ausencia del párrafo de contexto provisto. Además, al realizar el conjunto de datos solo en base a 536 artículos, la distribución de ejemplos de entrenamiento está extremadamente sesgada (Karpukhin et al., 2020) (Lee et al., 2019).

Natural Questions (NQ) (Kwiatkowski et al., 2019) contiene 307.373 ejemplos de entrenamiento, 7.830 de desarrollo y 7.842 de test. Fue creado a partir de consultas reales de la búsqueda de Google. Cada ejemplo contiene la pregunta, la página correspondiente de Wikipedia, un fragmento largo que contiene la respuesta, y uno o varios fragmentos cortos que efectivamente responden a la pregunta. Sin embargo, los fragmentos pueden estar vacíos. Si todos los fragmentos están vacíos, quiere decir que no hay respuesta a la pregunta.

Para la creación del conjunto de datos, a los anotadores se le dio una pregunta junto con una página de Wikipedia en inglés de los 5 primeros resultados de búsqueda. El anotador debía anotar una respuesta larga (normalmente un párrafo) y una respuesta corta (una o más entidades) o indicar si no había respuesta presente. Al ser preguntas reales, éstas cuentan con el contexto necesario.

Web Questions (WQ) (Bordes et al., 2015) contiene 6.642 pares de preguntas-respuestas. Utiliza la base de conocimiento Freebase para definir las

respuestas como entidades. Fue creado tomando preguntas de la API sugerencias de Google y anotando respuestas mediante los trabajadores de Mechanical Turk. Este conjunto de datos fue utilizado en varios trabajos, aunque en noviembre 2020 se dió de baja la base de datos Freebase, parte esencial de este conjunto de datos.

TriviaQA (Joshi et al., 2017) contiene más de 650.000 ejemplos de entrenamiento formados por pregunta-respuesta-contexto. Para cada par pregunta-respuesta (aproximadamente 95.000) se recogieron en promedio 6 contextos diferentes de Wikipedia y la web, donde se encuentra la respuesta. Las preguntas y respuestas fueron tomadas de 14 sitios de Trivia de la web, mientras que los contextos fueron obtenidos de forma independiente mediante un proceso llevado a cabo por los autores.

NewsQA[1] (Trischler et al., 2016) contiene cerca de 100.000 pares pregunta-respuesta sobre un conjunto de 10.000 artículos de CNN. La anotación de preguntas fue realizada por un grupo que solo veía el título de la noticia y un pequeño resumen. La anotación de respuestas fue realizada por otro grupo de personas. Lo particular de este conjunto de datos en relación a los anteriores es que el dominio son artículos de prensa. Las preguntas de este conjunto de datos, son de un estilo diferente a las anteriores.

Para la tarea en español se cuenta con algunos conjuntos de datos creados originalmente en español y se cuenta también con traducciones de algunos de los conjuntos de datos mencionados previamente creados en idioma inglés.

El conjunto de datos **Spanish Question Answering Corpus (SQAC)** (Gutiérrez-Fandiño et al., 2021) fue creado originalmente en español. Contiene 18.817 preguntas con respuestas extraídas de Wikipedia en español, Wikinews y el corpus AnCora (Taulé et al., 2008). El conjunto de datos fue creado siguiendo la guía de SQUAD 1.0. SQAC fue realizado con la versión en línea de Wikipedia de marzo-abril 2021.

El conjunto de datos **MultiLingual Question Answering (MLQA)** [2]

[1] https://www.microsoft.com/en-us/research/project/newsqa-dataset/
[2] https://github.com/facebookresearch/MLQA

(Lewis et al., 2019) es un conjunto de datos para evaluación en varios idiomas (inglés, árabe, alemán, español, hindú, vietnamita y chino simplificado). Las preguntas fueron realizadas en inglés y luego traducidas por traductores profesionales a los otros idiomas. Los contextos utilizados fueron los artículos de Wikipedia en los respectivos idiomas (No se utilizaron traducciones de los documentos). La anotación se realizó generando preguntas sobre los párrafos seleccionados. Para el español cuenta con 5.253 ejemplos de test y 500 de desarrollo.

También existen diversas traducciones automáticas de SQuAD. La traducción utilizada en este trabajo fue la traducción automática del conjunto de datos SQuAD 2.0 con el método **Translate-Align-Retrieve** (Carrino et al., 2019) [1]. El proceso implica la traducción automática de la pregunta, respuesta y contexto. Este conjunto de datos fue utilizado para la evaluación de BETO para Question Answering (Cañete et al., 2020).

El conjunto de datos **Question Answering Learning from Examples in Spanish (QuALES)**[2] fue creado en el marco del proyecto CSIC «Búsqueda de Respuestas a partir de Textos en español». El conjunto de datos de QA contiene 1.000 pares de preguntas-respuestas para entrenamiento, 800 para desarrollo y 800 para evaluación sobre un conjunto de artículos de noticias de La Diaria y Montevideo Portal referidas a la pandemia de COVID-19. Los anotadores fueron el equipo de proyecto, otros integrantes del Grupo PLN y estudiantes de grado de la carrera Ingeniería en Computación que participaron de un módulo taller. El corpus se creó seleccionando artículos del conjunto de noticias. Cada anotador recibió un conjunto de artículos y para cada uno de ellos debió pensar preguntas mirando solamente el título de la noticia, y luego también leyendo el texto completo. A continuación se debió marcar el span de texto correspondiente a la respuesta, o indicar que ésta no estaba presente.

En el marco del proyecto CSIC también se realizó la traducción al español del corpus NewsQA, llamado **NewsQA-ES**. Se debió traducir todos los textos del corpus y las preguntas y alinear la respuesta en el texto original con la versión traducida. Esto se logró entrenando un modelo neuro-

[1] https://github.com/ccasimiro88/TranslateAlignRetrieve
[2] https://github.com/pln-fing-udelar/covid19-qa/tree/master

nal de alineación, llamado Mask-Align. También se creó un subcorpus de 2.000 ejemplos corregido manualmente y se dividió en dos partes. Una de ellas fue utilizada para ajustar los hiperparámetros del modelo, y la otra para evaluarlo. Los pasos a seguir para reproducir el procedimiento llevado adelante para generar corpus traducido están disponibles en el repositorio https://github.com/pln-fing-udelar/newsqa-es.

2.6.2. Modelos del Lenguaje

Para la utilización de arquitecturas basadas en modelos del lenguaje, será necesario disponer de estos modelos pre-entrenados.

BERT es un modelo de transformers pre-entrenado en un gran corpus de textos en inglés no etiquetados, y con un proceso automático para generar entradas y etiquetas para dichos textos. De esta manera el modelo aprende una representación del lenguaje, que podemos utilizar para la tarea de recuperación de información. En la plataforma Hugginface pueden encontrarse los modelos para ser utilizados. Los modelos base de BERT son bert-base-cased [1] y bert-base-uncased [2] que se diferencian por distinguir minúsculas y mayúsculas o no, respectivamente.

Multilingual BERT (mBERT) es un modelo entrenado con los artículos de Wikipedia de los 104 idiomas más populares. No utiliza ningún indicador del lenguaje del artículo. Puede encontrarse su versión sensible a mayúsculas [3] y no sensible a mayúsculas [4].

Para el español contamos con BETO (Cañete et al., 2020), que es un modelo BERT entrenado en un gran corpus en español. El tamaño del modelo es similar al de los indicados anteriormente y fue entrenado de forma similar. Como en inglés, puede encontrarse su versión sensible a mayúsculas [5] y no sensible a mayúsculas [6].

Existe también un método optimizado de entrenamiento de BERT llamado RoBERTa (A Robustly Optimized BERT Pretraining Approach) (Liu et al., 2019). RoBERTa se entrena utilizando una variante de la tarea de "llenado de espacios en blanco"llamada .[en]mascaramiento dinámicoz un enfoque de entre-

[1] https://huggingface.co/bert-base-cased
[2] https://huggingface.co/bert-base-uncased
[3] https://huggingface.co/bert-base-multilingual-cased
[4] https://huggingface.co/bert-base-multilingual-uncased
[5] https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased
[6] https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased

namiento más extensivo en términos de datos y tiempo de entrenamiento. Para este modelo existen modelos de lenguaje en español, como RoBERTa-large-bne [7] (Gutiérrez-Fandiño et al., 2021), que es un modelo de lenguaje entrenado con un corpus de español provisto por la Biblioteca Nacional de España (BNE). Este corpus de 570 GB fue creado rastreando todos los sitios con dominio .es desde 2009 a 2019, y luego aplicando un proceso de depurado riguroso.

2.7. Conclusiones

Como se mostró en este capítulo, existe una vasta historia de investigación en recuperación de información, aún abierta en diferentes líneas. El balance entre la performance a nivel de calidad de resultados y de tiempo de respuesta es el objetivo a alcanzar por los diferentes enfoques para resolver la tarea. Los métodos clásicos y sus variantes aún son ampliamente utilizados: como componente principal de la tarea de IR, o en ocasiones como la primera componente de la arquitectura multietapa, seguida por rerankings basados en BERT. El advenimiento de Transformers y BERT da respuesta a los problemas más importantes de los métodos clásicos, pero introdujo el problema de la latencia. Arquitecturas híbridas intentan capturar lo mejor de los dos mundos, para lograr el balance necesario entre calidad y rapidez.

Las posibles líneas de investigación que se visualizan sobre recuperación de información a nivel general implican optimizar la arquitectura en capas planteada en el inciso 2.3.1 para mejorar la calidad de los documentos devueltos por los métodos clásicos o para disminuir el tiempo de ejecución del módulo basado en BERT.

Por otro lado, las arquitecturas como los Bi-encoders descritos en la sección 2.3.2 tienden a maximizar el potencial de los Transformers para la tarea de IR. Siguiendo este camino, es posible investigar usar modelos BERT preentrenados con dominios específicos o en diferentes idiomas. También es posible evaluar la aplicación de mejoras de BERT como RoBERTa para optimizar el entrenamiento o LongFormers (Beltagy et al., 2020) para usar documentos más largos como elementos de la colección.

A nivel particular de IR para QA con características temporales, también hay muchas líneas de trabajo abiertas. Incorporar la temporalidad en la re-

[7] https://huggingface.co/PlanTL-GOB-ES/roberta-large-bne

cuperación de información con arquitecturas basadas en BERT es imperativo para poder aprovechar todas las ventajas de dicha arquitectura para en este contexto.

Otra línea de trabajo fundamental para el desarrollo del área es el análisis y estudio de la evaluación de los sistemas de IR para QA.

En el siguiente capítulo se detallarán los dos problemas abordados en el marco de este trabajo. Estos problemas permitirán un análisis en profundidad del funcionamiento de la recuperación de información para la búsqueda de respuestas en diferentes dominios. Para la implementación se configuró un módulo de IR para QA en español que incorpora la arquitectura BERT, utilizando modelos del lenguaje para español.

Capítulo 3

Recuperación de información para la búsqueda de respuestas en idioma español

Uno de los objetivos de este trabajo es estudiar diferentes técnicas de recuperación de información para la búsqueda de respuestas en idioma español. Se trabajará en dos problemas diferentes, adaptando y evaluando modelos y sistemas que son el estado del arte para el idioma inglés.

El primer caso de estudio es el más usual e investigado, referido a documentos enciclopédicos como Wikipedia, con consultas de tipo deterministas, es decir preguntas normalmente tienen una única respuesta. En los ejemplos presentados en la tabla 3.1 pueden verse preguntas de este tipo, y porciones de documentos de Wikipedia que responden a la pregunta, así como las respuestas extraídas. Este experimento es el más utilizado para benchmarking de sistemas de IR y QA.

El segundo caso de estudio es considerablemente menos estudiado independientemente del idioma, y refiere a recuperación de información sobre documentos de prensa. En los ejemplos presentados en la tabla 3.2 pueden verse ejemplo de preguntas que se realizan sobre noticias, porciones de noticias que podrían responder a la pregunta, y las respuestas extraídas. Como dominio para el estudio se eligió la enfermedad COVID-19 y hechos relacionados a la pandemia.

El objetivo de este capítulo es presentar los dos problemas abordados en este trabajo. Se realizará un análisis de las particularidades de la recuperación

Ejemplo 1	
Contexto	*«Normandía (en francés, Normandie; en normando, Normaundie) es una entidad histórica, geográfica y cultural en el noroeste de Francia, bordeada por el canal de la Mancha.»*
Pregunta	*¿En qué país se encuentra Normandía?*
Respuesta	*Francia*
Ejemplo 2	
Contexto	*«David Herbert Richards Lawrence (Eastwood, Inglaterra; 11 de septiembre de 1885-Vence, Francia; 2 de marzo de 1930) fue un escritor inglés, autor de novelas, cuentos, poemas, obras de teatro, ensayos, libros de viaje, pinturas, traducciones, y críticas literarias.»*
Pregunta	*¿En qué año nació David Herbert?*
Respuesta	*1885*

Tabla 3.1: Ejemplos de porciones de documentos de Wikipedia, una pregunta, y su respuesta extraída del documento.

de información para la búsqueda de respuestas en los diferentes dominios. Los resultados se presentarán en el capítulo 4.

Para la implementación de la recuperación de información para ambos dominios se configuró un módulo de IR para QA que incorpora la arquitectura de Transformers, utilizando un modelo BERT entrenado en idioma español.

3.1. Módulo de IR para QA en español

En esta sección se realizará una introducción a la herramienta elegida para la implementación de la recuperación de información de los problemas elegidos. Se realizará también una justificación de la elección de la herramienta.

Luego de analizar los diferentes enfoques y arquitecturas para llevar a cabo la tarea, se decidió adaptar el sistema Dense Passage Retriever (DPR) (Karpukhin et al., 2020) para el idioma español.

Se eligió este sistema ya que los resultados en inglés para Wikipedia eran el estado del arte al momento de inicio del proyecto. DPR utiliza arquitectura BERT, y es fácilmente adaptable al idioma español. Además el sistema permite la evaluación del módulo de IR de forma separada a la evaluación del pipeline

Ejemplo 1	
Contexto	*«El ministro de Defensa Nacional, Javier García, agregó que este domingo a las seis de la mañana va a salir para Lima un vuelo de la Fuerza Aérea con 60 ciudadanos peruanos y traerá a 75 uruguayos.»*
Pregunta	*¿Cuántos uruguayos arribarán hacia Uruguay desde Lima?*
Respuesta	*75*
Ejemplo 2	
Contexto	*«De acuerdo con el balance oficial actualizado por el Ministerio de Salud, con estos nuevos casos, las muertes de personas infectadas por covid-19 ya son 3.313 y el número de infectados en la actualidad es de 20.861, aunque las cifras reales son mucho mayores, ya que además de que en Brasil no se está realizando pruebas de diagnóstico en forma masiva.»*
Pregunta	*¿Cuántos casos de Covid hay en Brasil?*
Respuesta	*20.861*

Tabla 3.2: Ejemplos de porciones de documentos de prensa, una pregunta, y su respuesta extraída del documento.

completo, lo que facilita el análisis de la tarea que nos proponemos estudiar.

DPR es un sistema completo de QA. Cuenta con un módulo de recuperación de información y otro de extracción de respuestas, llamados Retriever y Reader respectivamente como se muestra en la figura 3.1. Dado que el objetivo de esta tesis se basa en la investigación de técnicas de recuperación de información para la búsqueda de respuestas, nos centraremos en el primer módulo.

El módulo de IR de DPR es un ejemplo de arquitectura de bi-encoders, donde tanto documentos como consultas se representan como vectores densos. La forma de determinar la relevancia de un documento, es midiendo la similitud de la representación de la consulta con la representación de los documentos, tal como se muestra en la figura 3.2

DPR entrena dos encoders (P y Q) basados en BERT. Uno de ellos se utiliza para mapear pasajes de documentos con vectores densos. Se utilizan pasajes de documentos de menos de 512 tokens, y no los documentos originales ya que BERT tiene una limitante en el tamaño de la entrada. Una vez se tiene

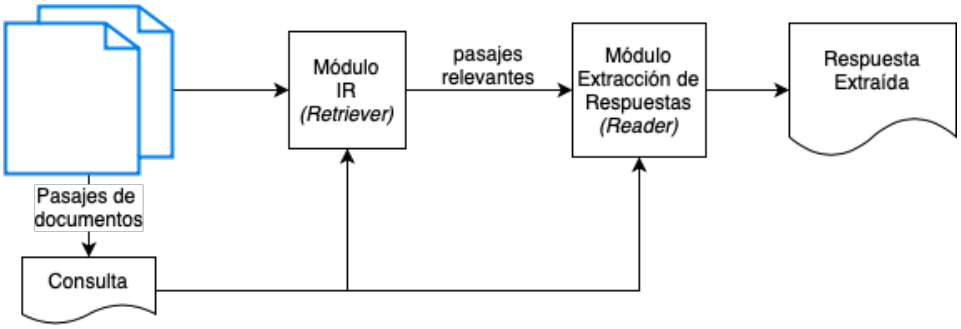

Figura 3.1: Sistema DPR. El módulo Retriever recupera los pasajes relevantes. Estos pasajes son enviados al módulo Reader, donde se extrae de los pasajes la respuesta a la consulta, o se indica que no hay respuesta posible.

el encoder P entrenado, DPR realiza el cálculo previo de vectores densos para los pasajes de documentos de la colección. Se utiliza como representación del pasaje, la salida del encoder para el token [CLS] de dimensión 768. Luego se indexan dichos vectores densos.

En tiempo de inferencia, se calcula el vector denso de la representación de la consulta utilizando el encoder Q. Se utiliza como representación de la consulta la salida del encoder Q para el token [CLS] de dimensión 768.

Para medir la similitud de la consulta con las porciones de documentos se realiza una aproximación de los vecinos más cercanos mediante FAISS (algoritmo basado en Product quantization mencionado en la sección 2.3.2), usando como medida de similitud la distancia coseno. De esta manera se genera el listado ordenado de los documentos más similares a la consulta.

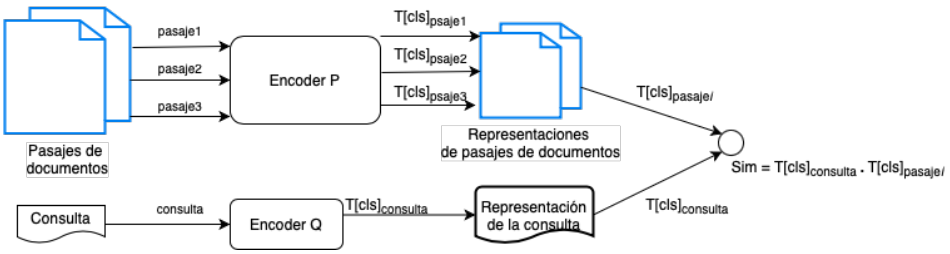

Figura 3.2: Módulo IR de DPR. DPR realiza el cálculo previo de embeddings para porciones de documentos de la colección. En tiempo de inferencia, se calcula el embedding de la consulta. Se calcula el puntaje de similitud calculando el producto interno entre vectores.

Dado que BERT es un modelo del lenguaje para el idioma inglés, es necesario utilizar para los encoders un modelo base que siga la misma arquitectura que BERT pero que haya sido pre-entrenado en textos en español. Además,

para el entrenamiento requerido para el cálculo de vectores densos y consultas serán necesarios conjuntos de datos de consultas en idioma español. La configuración detallada para la adaptación al sistema para el español puede encontrarse en el Anexo 2. A continuación se profundizará en cada una de las etapas.

3.1.1. Entrenamiento de encoders

Los encoders son los responsables de transformar las porciones de documentos y consultas en embeddings contextuales. Los modelos del lenguaje como BERT son modelos pre-entrenados que pueden realizar esta tarea. Se realiza entonces un nuevo entrenamiento de estos modelos con conjuntos de datos de QA como los descritos en la sección 2.6.1 para que los modelos puedan aprender además las características de la tarea.

DPR permite utilizar además de ejemplos positivos, ejemplos negativos. Es decir, para una pregunta, indicar pasajes donde se puede encontrar la respuesta (ejemplos positivos), y otros en los que no (ejemplos negativos). En este trabajo no se incluyeron pasajes negativos, debido a que se requería un procesamiento adicional del dataset de QA que se decidió no realizar.

El entrenamiento de ambos encoders se realiza de forma conjunta. La entrada del encoder P (Pasajes) son los pasajes positivos y negativos. Los pasajes positivos son los contextos del conjunto de datos de QA. La entrada del encoder Q (Consultas) son las preguntas del mismo dataset. Se optimiza la función de pérdida conjunta como el logaritmo negativo de la verosimilitud del pasaje positivo. El objetivo de la optimización es maximizar la similitud entre la consulta i-esima y el pasaje positivo i-esimo y disminuir la similitud entre aquellos que no son relevantes, es decir, entre la consulta i-esima y los pasajes negativos i-esimos.

El entrenamiento de los encoders es uno de los procesos que consume mayor tiempo. Como resultado de este paso obtenemos nuevos modelos de encoders, que se utilizarán en las siguientes etapas. En la tabla 3.3 se resume lo requerido en esta etapa y la salida deseada.

3.1.2. Generación de Embeddings

DPR realiza el cálculo previo de embeddings para porciones de documentos de la colección. Los documentos originales son segmentados en tiras de como

Entrada 1	Modelo BERT entrenado en idioma español
Entrada 2	Conjunto de datos de QA
Salida esperada	Modelos BERT en idioma español con capacidades para realizar tareas de QA.

Tabla 3.3: Resumen de Entradas y Salidas esperada de la etapa «Entrenamiento de encoders».

máximo 512 tokens, y luego procesadas por el encoder entrenado en el paso anterior. Los embeddings son indexados con la biblioteca FAISS.

El proceso de generación de embeddings para el conjunto de datos de documentos es un proceso altamente paralelizable. DPR permite utilizar varios servidores de GPU para la generación de embeddings de forma independiente.

3.1.3. Inferencia de Retriever

En tiempo de inferencia, para cada pregunta del conjunto de datos de validación, se devuelve una lista de ordenada de porciones de documentos. Para cada porción, se tiene el puntaje de similitud con la consulta y una bandera que indica si el texto contiene alguna de las posibles respuestas indicadas en el conjunto de datos de evaluación.

Los resultados se ordenan por su puntuación de similitud, de más relevante a menos relevante. Además, el sistema calcula el desempeño del sistema mediante la métrica Top k (para $k \in [1..100]$ por defecto, pero puede configurarse la cota superior).

De forma predeterminada, se utiliza un proceso de búsqueda exhaustivo, pero puede elegirse utilizar índices con pérdida. La puntuación de similitud es el producto para la búsqueda exhaustiva (indexador plano) y para índice HNSW es la distancia L2 en un espacio de representaciones modificado.

En el marco de este trabajo se abordaron dos problemas de recuperación de información en idioma español, referido al tipo de documentos a utilizar y las preguntas relacionadas a éstos. Se presentarán a continuación los dos problemas abordados.

3.2. Recuperación de información de Wikipedia

El primer problema investigado correspondió a la recuperación de documentos para la búsqueda de respuestas en los artículos de Wikipedia. Este problema es interesante ya que es uno de los más recurrentes en los trabajos de investigación sobre IR y QA, y es utilizado habitualmente como medida de referencia en Question Answering.

Muchos de los recursos disponibles para la investigación se basan en artículos de Wikipedia. Los conjuntos de datos de QA en inglés más utilizados y de mayor volumen son SQuAD 1.0, SQuAD 2.0 y NQ. Estos conjuntos de datos utilizan como contextos porciones de artículos de Wikipedia exclusivamente. Otros conjuntos de datos como TriviaQA utiliza contextos de Wikipedia pero también de otros sitios de Internet.

Para la investigación en idioma español, normalmente se utilizan traducciones de SQuAD, por lo que se permanece en el dominio de Wikipedia. MLQA también utiliza exclusivamente documentos de Wikipedia, y SQAC utiliza contextos de Wikipedia pero también de otras fuentes.

Además el conjunto de artículos de Wikipedia es libre, de gran tamaño, crece constantemente, y es posible encontrarlo es múltiples idiomas. Los recursos disponibles llevan a pensar que este es un experimento interesante a realizar.

Realizar recuperación de información sobre documentos de Wikipedia con preguntas pensadas para este tipo de documentos, tiene características particulares. Los documentos de Wikipedia son de índole enciclopédico. Una Enciclopedia es una obra que se expone el conjunto de los conocimientos humanos. La enciclopedia reúne conocimientos de una forma objetiva y universal, en artículos separados. Las preguntas de los conjuntos de datos de QA basados en contextos de Wikipedia, son preguntas relacionadas a conocimiento, las cuales casi siempre tienen una única respuesta.

3.2.1. Adaptación de DPR para la implementación

A continuación se detallan los pasos necesarios para la implementación del problema utilizando DPR.

Procesamiento de conjuntos de datos

Para la ejecución de los pasos necesarios para la utilización del módulo IR de DPR necesitaremos tres conjuntos de datos diferentes. El primero es el conjunto de documentos sobre los cuales se desea recuperar información. Los otros dos son conjuntos de datos de QA, que se utilizan para el entrenamiento de los encoders y el último para la evaluación del módulo de IR. Como se vio en la sección 2.5, para la evaluación del módulo de IR no se requiere toda la información del conjunto de datos tradicional de QA. En la tabla 3.4 se muestra un resumen de los datasets utilizados para este problema.

Para la creación del conjunto de documentos se utilizó el dump de fecha 2021/08/20 con todos los artículos de Wikipedia escritos en español. Se depuró el conjunto de datos utilizando WikiExtractor. Luego se ejecutó el script processWikipedia [1] modificado para: (1) poder contar en el conjunto de datos final con los ID de los documentos, títulos, y no quitar las stop words de las oraciones, (2) poder guardar porciones de texto de máximo 100 tokens con oraciones completas.

Como conjunto de datos de entrenamiento de los encoders se utilizó TAR20, la traducción automática de SQUAD 2.0 con el método Translate-Align-Retrieve (Carrino et al., 2019).

Para la validación, se encontraron problemas para cumplir con la restricción de inclusión de los spans de texto de las respuestas en algún ítem de la colección de documentos. Actualmente, no existe un conjunto de datos de preguntas y respuestas creado a partir de una versión específica registrada de Wikipedia. Eso implica que los spans de texto de las respuestas indicadas en el conjunto de datos de validación, pueden no encontrarse en el conjunto de artículos utilizado, ya que Wikipedia es un conjunto de documentos que cambia constantemente.

Por ejemplo, ante la pregunta «*¿Quién fue Joaquim Renart?*» el conjunto de datos SQAC, da la respuesta: «*un pintor, decorador y coleccionista, fundador de Fomento de las Artes Decorativas*». Pero en las versiones de Wikipedia disponibles dicho span de texto no aparece, y en su lugar dice «*un dibujante, pintor y decorador español*».

Al no contar con el conjunto de documentos con los que fue creado el conjunto de datos, éste no puede ser utilizado satisfactoriamente para la evaluación

[1] Tomado de https://gist.github.com/snakers4/e0b0e68904db65671ca979639b337f7b

automática de la tarea de recuperación de información. Se profundizará en este problema en el análisis.

Se decidió proceder con el experimento, tomando en consideración que las métricas de evaluación no serán satisfactorias. En el caso de SQAC, se quitaron las preguntas a documentos que no corresponden a Wikipedia para mantener la consistencia.

Entrenamiento de encoders

Esta etapa tiene como objetivo obtener modelos del lenguaje en idioma español, que hayan aprendido las características de la tarea de QA sobre documentos de Wikipedia (ver tabla 3.3). Para esta etapa se utilizó el modelo de encoder pre-entrenado para el español BETO, que no distingue minúsculas y mayúsculas [1]. Como conjuntos de datos de QA para el entrenamiento se utilizaron los conjuntos de datos TAR20_train y TAR20_dev. Este proceso es el que insumió más tiempo y recursos. Se utilizaron los recursos de ClusterUY, utilizando un GPU y 32 GB de memoria. El entrenamiento demoró 120 hs y alcanzó las 29 épocas.

Generación de Embeddings

Para la generación de embeddings de las porciones de documentos se utilizó como modelo el encoder entrenado en el paso anterior. Dado que el tamaño de los conjuntos de datos de documentos no es muy grande, se decidió ejecutar los trabajos en un único GPU, sin necesidad de paralelizar el proceso. Este proceso también fue llevado a cabo en ClusterUY, utilizando un GPU y 32 GB de memoria. Se generaron 1.074.999 embeddings en tres horas.

Inferencia de Retriever

Para la evaluación se utilizaron los conjuntos de datos SQAC y MLQA, con un total de 867 preguntas para SQAC y 5252 para MLQA. Se utilizaron los recursos de ClusterUY, utilizando un GPU y 32 GB de memoria. El proceso de inferencia demoró dos minutos.

[1] modelo bert-base-spanish-wwm-uncased

	Conjuntos de datos
Colección	Dump Wikipedia (08/2021)
Entrenamiento	TAR20_train y TAR20_dev
Evaluación	SQAC (solo Wikipedia) y MLQA (solo español)

Tabla 3.4: Resumen de datasets utilizados para el problema de recuperación de información para la búsqueda de respuestas sobre Wikipedia.

3.2.2. Análisis

En esta sección se analizarán algunos de los problemas detectados en la evaluación de los sistemas de IR. El método utilizado normalmente en los trabajos de investigación se encuentra descrito en la sección 2.5.1, y presenta varios problemas, dependiendo los conjuntos de datos que se utilicen y el dominio donde se desea recuperar información. En esta sección se presentarán los problemas detectados para evaluación de la recuperación de información de Wikipedia, utilizando los conjuntos de datos mencionados anteriormente.

El problema de la coherencia de versiones

Wikipedia es un conjunto de documentos que cambia constantemente, ya que los artículos son editados, actualizados y hasta eliminados. En línea pueden encontrarse versiones de los cuatro meses más recientes. Para el idioma inglés, hay mayor disponibilidad de dumps en el sitio Internet Archive [1].

El problema surge porque que los conjuntos de datos disponibles de evaluación, como SQAC y MLQA, no fueron realizados extrayendo las respuestas de alguna de las versiones a disposición de Wikipedia. Es por eso que en ocasiones las respuestas indicadas en el conjunto de datos de evaluación no coinciden con el contenido de los artículos de Wikipedia que se están utilizando en un momento dado.

La performance de la extracción de respuestas se mide evaluando el módulo para cada pregunta en un determinado contexto, que está contenido en el conjunto de datos de evaluación. El contexto es un párrafo de Wikipedia, pero las triplas contexto-pregunta-respuesta fueron recolectadas en un momento dado. Es por eso que los conjuntos de datos de QA se ajustan perfectamente a la

[1] https://archive.org/details/wikimediadownloads

evaluación de la extracción de respuestas.

Para la evaluación de la tarea de IR, o para la del pipeline completo IR-QA, es necesario contar con una coherencia entre el conjunto de datos de documentos y el de evaluación. Esto con el fin de asegurar que las respuestas contenidas en el conjunto de datos de evaluación efectivamente se encuentren en los artículos del conjunto de documentos.

Los autores de los conjuntos de datos utilizados para la evaluación (SQAC y MLQA) no proporcionan el dump de Wikipedia con el que éstos fueron creados. Es importante destacar que tampoco lo hacen los conjuntos de datos de QA en inglés, por lo que el problema no es exclusivo del idioma español. Sin embargo para la evaluación en inglés podría utilizarse una versión del dump de Wikipedia cercana a la fecha de creación del conjunto de datos de QA a utilizar.

Este problema es muy relevante, porque hace que los sistemas no sean evaluados bajo las mismas condiciones. Los sistemas más antiguos llevarán ventaja ante este problema ya que la versión de Wikipedia utilizada habrá sufrido menos cambios y por ende el problema se presentará en menos ocasiones.

Utilizando un ejemplo: SQuAD 1.0 fue creado en 2016, con una versión de Wikipedia de ese año o anterior. Suponga que se desarrolla un sistema de IR-QA en 2017. Para ello se utilizará alguna de las versiones de Wikipedia disponibles: alguna versión de 2017. Al evaluar el sistema, algunos artículos de Wikipedia habrán cambiado (ya que se están utilizando artículos al menos un año más nuevos), y con seguridad tendremos el problema de la discrepancia entre artículo-respuesta. Ahora bien, suponga que se desarrolla en 2020 otro sistema de IR-QA y se utiliza alguna de las versiones disponibles de Wikipedia: alguna versión de 2020. En el lapso 2016-2020 habrán cambiado más documentos que en el lapso 2016-2017, por lo que se tendrán aún más discrepancias artículo-respuesta. Entonces, aunque ambos sistemas se evalúan utilizando Wikipedia y SQuAD 1.0, en realidad no es la misma Wikipedia, y los sistemas más antiguos llevarán ventaja ya que probablemente su versión de Wikipedia habrá sufrido menos cambios que la utilizada por los sistemas más nuevos.

No hay registro ni estudio de este problema en ningún trabajo de investigación encontrado. En la web puede encontrarse apenas un problema reportado por una persona [1], que no tiene seguimiento. Para el inglés el problema podría solventarse al menos en parte utilizando la versión más cercana a la fecha de creación del conjunto de datos de QA. Para español, al día de hoy no hay una solución con los conjuntos de datos disponibles de QA.

El problema de la falta de contexto

Los conjuntos de datos SQAC y MLQA se crearon solicitándole a los anotadores generar preguntas sobre párrafos que debían leer previamente. Realizar la anotación siguiendo esta metodología genera que muchas de las preguntas no cuenten con la información suficiente para ser respondidas, en ausencia del párrafo provisto. En la tabla 3.5 se muestran algunas de las preguntas presentes en los conjuntos de datos de evaluación que cuentan con esta característica.

Pregunta	Respuesta
«*¿Qué grupos se dividieron?*»	«*la facción de Magdiwang y la facción de Magdalo*»
«*¿Cuándo fueron transferidos?*»	«*En marzo de 1942*»
«*¿Qué pasó tras completarse el proyecto?*»	«*se abandonara*»

Tabla 3.5: Ejemplos de preguntas en el conjunto de datos de evaluación sin información suficiente para ser respondidas.

Para la evaluación de la tarea de extracción de respuestas, esto no es un problema, ya que la performance del sistema se mide evaluando la extracción de la respuesta para cada pregunta en un determinado contexto. Pero para la evaluación de la tarea de IR o del pipeline completo IR-QA, no se cuenta con el contexto de la pregunta, por lo que para el sistema es muy desafiante poder recuperar los documentos correctos.

En el capítulo 4 se mostrarán y analizarán los resultados de este caso de estudio. Se utilizará la herramienta DPR configurada para ser utilizada en idioma español y los conjuntos de datos presentados en esta sección.

[1] https://pythontechworld.com/issue/facebookresearch/drqa/112

3.3. Recuperación de información de prensa

El segundo problema investigado consistió en la recuperación de documentos para la búsqueda de respuestas en el conjunto de noticias de dominio específico. Este problema nace en el marco del proyecto CSIC «Búsqueda de Respuestas a partir de Textos en español», presentado por el grupo de Procesamiento de Lenguaje Natural de la Facultad de Ingeniería, como parte de la línea Question Answering. El objetivo es poder contestar preguntas sobre noticias de un dominio específico. Dada la relevancia mundial y local de la pandemia, se eligió como dominio la enfermedad COVID-19 y hechos relacionados a la pandemia.

Este problema es interesante ya que no existen muchos trabajos de investigación referidos a IR para QA sobre documentos de prensa. Los documentos de prensa, así como preguntas asociadas a ellos tienen algunas particularidades. Las referencias temporales de las consultas y los documentos es información fundamental para la búsqueda de respuestas. Las preguntas que se realizan sobre noticias normalmente no tienen una única respuesta, y casi siempre existe más de una noticia que contiene una posible respuesta. Además, la respuesta a una pregunta puede cambiar con el tiempo. Estas características hacen que sea interesante estudiar el sistema de IR en este contexto.

A diferencia del dominio de Wikipedia, no hay demasiados conjuntos de datos de prensa, y tampoco existe una estandarización. Como conjuntos de datos de documentos, en varios trabajos de investigación es utilizado New York Times Anotated Corpus. Como conjunto de datos de QA se tiene NewsQA. Para la evaluación, las respuestas indicadas en el conjunto de datos de evaluación deben estar incluidas en los documentos. Dado que NewsQA se basa en artículos de la CNN, entonces no es posible utilizarlo para evaluar sistemas de IR sobre los artículos del New York Times Anotated Corpus.

En idioma español, hasta donde sabemos no hay más recursos que los generados en el marco del proyecto CSIC «Búsqueda de Respuestas a partir de Textos en español». Se tiene NewsQA-ES para ser utilizado para entrenamiento (pero no para la evaluación) y se cuenta con QuALES_docs y QuALES, como conjuntos de datos de documentos y QA respectivamente. Estos dos últimos, aunque pequeños, son de gran importancia porque pueden utilizarse para el

pipeline completo IR-QA y también para IR.

3.3.1. Adaptación de DPR para la implementación

A continuación se detallan los pasos necesarios para la implementación del problema utilizando DPR.

Procesamiento de conjuntos de datos

Para la recuperación de información de prensa de un dominio específico, como conjuntos de datos de documentos se utilizó QuALES_Docs. Los archivos XML correspondientes a cada noticia fueron recopilados y procesados para que puedan ser utilizados por la arquitectura BERT, aplicando una transformación que permite la subdivisión de los artículos.

Para el entrenamiento de los encoders, dado que el tamaño del conjunto de datos QuALES es pequeño, se requiere un conjunto de datos adicional. Idealmente, las preguntas y contextos de este conjunto de datos deben ser de la misma índole que las que se utilizarán para la evaluación. Es decir, sobre documentos que son artículos enciclopédicos, normalmente se hacen ciertas preguntas, pero sobre noticias normalmente se hacen otro tipo de preguntas. Se requiere entonces conjuntos de datos de QA sobre noticias en español. Se utilizaron NewsQA-ES y QuALES_train para hacer el fine tunning del encoder.

Para validación se utiliza el conjunto de datos de evaluación de QuALES. El conjunto de datos original de 800 preguntas debió ser reducido a 656 preguntas eliminando las preguntas sin respuesta, ya que en el marco de la evaluación de recuperación de información estas preguntas no son de utilidad.

En la tabla 3.6 se muestra un resumen de los datasets utilizados para este problema.

Entrenamiento de encoders

Esta etapa tiene como objetivo obtener modelos del lenguaje en idioma español, que tengan la capacidad de realizar tareas de QA sobre documentos de prensa (ver tabla 3.3). Para esta etapa se utilizó el modelo de encoder pre-entrenado para el español BETO, que no distingue minúsculas y mayúsculas. Se entrenó con el conjunto de datos NewsQA-ES y luego se realizó un segundo entrenamiento con QuALES_train. Se utilizaron los recursos de ClusterUY, utilizando un GPU y 32 GB de memoria. El primer entrenamiento con

NewsQA-ES demoró 120 hs alcanzando las 36 épocas. El siguiente entrenamiento con QuALES_train demoró 10 minutos alcanzando las 39 épocas. La gran diferencia en el tiempo incurrido se debe al tamaño de los conjuntos de datos.

Generación de Embeddings

Para la generación de embeddings de las porciones de documentos de prensa se utilizó como modelo el encoder entrenado en el paso anterior. Dado que el tamaño de los conjuntos de datos de documentos no es muy grande, se decidió ejecutar los trabajos en un único GPU, tal como en el experimento previo. Se utilizaron los recursos de ClusterUY, utilizando un GPU y 4 GB de memoria. El proceso demoró un minuto y procesó 1249 documentos.

Inferencia de Retriever

Para la evaluación se utilizó el conjunto de datos de test de QuALES, con un total de 656 preguntas. Se utilizaron los recursos de ClusterUY, utilizando un GPU y 6 GB de memoria. El proceso de inferencia demoró un minuto.

	Conjuntos de datos
Colección	Noticias de La Diaria y Montevideo Portal (QuALES_Docs)
Entrenamiento	NewsQA_es y QuALES_train
Evaluación	QuALES_test

Tabla 3.6: Resumen de datasets utilizados para el problema de recuperación de información para la búsqueda de respuestas sobre Prensa.

3.3.2. Análisis

En esta sección se presentarán los problemas detectados para evaluación de la recuperación de información de prensa, utilizando los conjuntos de datos mencionados anteriormente. Se verá que algunos problemas que se tenía para la evaluación de la recuperación de información de Wikipedia no se tienen, que otros se mantienen, y también que surgen nuevos, inherentes a la naturaleza de los documentos y consultas.

En la sección 4.2.2 se propone un método para subsanar cada uno de los problemas encontrados.

El problema de la coherencia de versiones

El conjunto de datos QuALES fue creado utilizando los artículos de Quales_Docs. De esta forma podemos asegurar que los spans de texto de las respuestas del conjunto de datos están contenidas en al menos un documento de la colección de documentos. El problema de coherencia de versiones mencionado en el experimento de Wikipedia, no está presente en este experimento.

El problema de la falta de contexto

Por la naturaleza de anotación del conjunto de datos QuALES, existen preguntas realistas (aquellas realizadas mirando solamente el título de la noticia), y otras a las que quizá pueda faltarle contexto, ya que fueron realizadas mirando el artículo, tal como sucede en SQuAD. Sin embargo, en la guía de anotación del conjunto de datos se solicitó a los anotadores tener especial cuidado en incluir en las preguntas todo el contexto posible para poder responderlas. De esta manera se intentó tener una gran cantidad de preguntas con respuestas (muchas de las preguntas realizadas mirando solamente el título de la noticia, no tienen respuesta), y disminuir el problema de la falta de contexto.

Al analizar las preguntas del conjunto de datos de evaluación, puede verse que efectivamente el problema de falta de contexto está presente. En la tabla 3.7 se muestran algunas de las preguntas presentes en QuALES test que cuentan con esta característica.

Pregunta	Respuesta
«*¿Cuándo retoma las actividades el resto del plantel?*»	«*el próximo lunes*»
«*¿A quiénes se busca proteger con las medidas solicitadas?*»	«*mujeres, niñas, niños y adolescentes*»
«*¿Qué edad tenía el fallecido?*»	«*73*»
«*¿Quién debe tomar las medidas?*»	«*las autoridades*»
«*¿Dónde se realizarán los actos?*»	«*la plaza Libertad, el intercambiador Belloni, la plaza Colón y la plaza Lafone*»

Tabla 3.7: Ejemplos de preguntas en el conjunto de datos QuALES test sin información suficiente para ser respondidas.

Análisis de preguntas incompletas

Analizando las preguntas con falta de contexto, puede verse que en muchos casos la información faltante refiere al momento en que se realizó la pregunta, o sobre qué período de tiempo se está preguntando. Algunos ejemplos pueden verse en la tabla 3.8.

Pregunta
«*¿Cuáles son las nuevas restricciones para las visitas a los reclusos?*»
«*¿Cuál es el porcentaje de vacunados en Portugal?*»
«*¿Qué medidas se flexibilizaron en Austria?*»

Tabla 3.8: Ejemplos de preguntas sin contexto temporal en el conjunto de datos QuALES

Para poder responder los dos primeros ejemplos, es necesario saber el momento en el que la consulta fue efectuada. Para responder la última pregunta, deberíamos saber sobre qué lapso se está realizando la pregunta. Por ejemplo, la pregunta podría reformularse de esta manera y así transformarse en una pregunta completa: «*¿Qué medidas se flexibilizaron en Austria en mayo de 2022?*». Otra forma de transformar la pregunta podría ser saber el momento en el que la pregunta fue efectuada, y asumir que lo que se espera saber son las últimas medidas flexibilizadas en Austria.

En el contexto de Question Answering sobre una colección de documentos con características de temporalidad se propone que una pregunta que no contenga ninguna expresión temporal, puede ser mapeada al momento presente, o al período cuando ocurrió el evento al que hace referencia la pregunta (J. Wang et al., 2022).

El problema de la simplificación de la evaluación

Como se vio en la sección 2.5.1, la evaluación de los sistemas de IR para QA no utilizan los índices ni el contexto del conjunto de datos de evaluación, lo que puede llevar a una clasificación conceptualmente errónea. Dado que se define como relevante los documentos que contengan el span de texto de la respuesta, es posible que un documento sea catalogado como relevante, aunque no responda conceptualmente la pregunta.

Se analizaron los casos en los que documentos son clasificados erróneamente como relevantes. Las respuestas indicadas en el conjunto de datos de evaluación son spans muy comunes, que suceden en muchos de los documentos. Por ende, aunque el sistema no encuentre el documento que responda la pregunta, es probable que el span se encuentre en algún documento devuelto, y el sistema lo clasifique como relevante.

Por ejemplo, el span «*Italia*» aparece en muchos documentos. Ante la pregunta: «*¿Cuál es el segundo país más afectado por el Covid?*», el sistema devuelve 5 documentos, entre los cuales aparece mencionado el país, pero no en un contexto que responda la pregunta. Otros ejemplos de spans de respuestas en el conjunto de datos son: «*seis*», «*julio*», «*el gobierno*», «*12*». Puede verse que son spans muy comunes, tal como «*Italia*».

Al analizar los ejemplos donde el sistema devuelve documentos clasificados erróneamente como relevantes, puede verse también que el sistema clasifica como no relevantes documentos que podrían responder la pregunta. Estos documentos son clasificados como no relevantes porque la respuesta no coincide con la indicada en el conjunto de datos de evaluación.

Veamos el ejemplo de la pregunta «*¿Cuál es el segundo país más afectado por el Covid?*». En este ejemplo el conjunto de datos de evaluación indica como respuesta «*Italia*». Sin embargo, en uno de los 5 documentos devueltos se menciona que Macedonia del Norte es el segundo país con más muertes por COVID, pero no coincide con lo indicado como respuesta en el conjunto de datos de evaluación. Esto se debe a que la pregunta del conjunto de datos de evaluación fue realizada basándose en un artículo posiblemente de una fecha diferente al documento recuperado por el sistema.

Dado que los artículos no indican la fecha de publicación y la pregunta no indica un período de tiempo específico, no se puede decir que la respuesta «*Macedonia del Norte*» sea incorrecta. El análisis de este tipo de problemas se verá a continuación en la sección 4.2.5.

El problema de los pasajes y documentos

El sistema DPR utilizado para los experimentos devuelve pasajes de los documentos originales. El sistema utiliza pasajes en lugar de documentos completos porque utiliza una arquitectura basada en BERT. Dado que BERT no puede procesar entradas de más de 512 tokens, los artículos fueron procesados y divididos en pasajes de hasta 450 tokens (Los pasajes pueden tener menos tokens ya que fueron procesados de tal forma que no se cortaran las oraciones).

Para la evaluación, el sistema verifica si el span de texto de la respuesta se encuentra en el pasaje, en lugar de verificar en el documento completo original que contiene el pasaje. Este método, particular de DPR, puede llevar a una evaluación equivocada del sistema.

El problema de la metodología de evaluación

A lo largo de este análisis se observa que la metodología de evaluación estándar de la tarea de IR tiene varias falencias. Por un lado, la simplificación de la evaluación conlleva a clasificar erróneamente los documentos. Por otro lado, las características del conjunto de datos a utilizar en la evaluación inciden fuertemente en los resultados.

En este experimento existen dificultades particulares inherentes a la naturaleza de las preguntas que se realizan sobre noticias. También existen dificultades para definir las posibles respuestas correctas a las preguntas.

Profundizando en esto último, los artículos pueden tener diferentes respuestas a la misma pregunta. Esto puede darse por temporalidad, variaciones del lenguaje, o por la naturaleza de la pregunta. Para mostrar más claramente el problema, a continuación se muestran dos ejemplos.

Ejemplo 1: El conjunto de datos de evaluación cuenta con la siguiente pregunta y respuesta asociada: «*¿Dónde empezó la pandemia?*» «*China*». Si el Sistema IR recupera en primer lugar un documento que dice que la pandemia empezó en Wuhan, éste es catalogado como no relevante porque no tiene la respuesta «*China*» que es lo que se tiene en el conjunto de datos de test.

Ejemplo 2: El conjunto de datos de evaluación cuenta con la siguiente pregunta y respuesta asociada: «*¿Cuántos contagiados hubo hoy por covid?*» «*20*» Si el Sistema IR recupera en primer lugar un artículo que indica que hubo

30 contagiados (porque corresponde a un día diferente al artículo con el que se hizo la pregunta del conjunto de datos de evaluación), éste es catalogado como no relevante. Aquí tenemos un problema de temporalidad. En principio no puede saberse cuál es la respuesta correcta porque no se tiene los datos temporales de los artículos. Se considerarán como relevantes cualquiera de los 2 documentos.

En el próximo capítulo, se propone un método manual de evaluación de sistemas de IR, focalizándonos en el dominio de prensa. El objetivo es poder obtener una mejor estimación del valor real del rendimiento del sistema, y así poder saber cuánto subestima el rendimiento del sistema el método automático tradicional de evaluación.

En el próximo capítulo se detallará el método y se aplicará, mostrando los resultados.

Capítulo 4

Presentación y análisis de resultados

El objetivo de este capítulo es presentar los resultados de los dos casos de estudio planteados en el capítulo 3 y realizar el análisis.

El primer caso de estudio es la recuperación de información de documentos de Wikipedia, con consultas que normalmente tienen una única respuesta. Para el entrenamiento y la evaluación se utilizaron conjuntos de datos disponibles como traducciones de conjuntos de datos de referencia, o conjuntos de datos originales.

El segundo caso de estudio es la recuperación de información de documentos de prensa sobre la enfermedad COVID-19 y hechos relacionados a la pandemia. Para el entrenamiento y la evaluación se trabajó junto al grupo de PLN en la creación de conjuntos de datos originales y traducción de conjuntos de datos de noticias.

El método de evaluación normalmente utilizado es el descrito en la sección 2.5.1. La métrica normalmente utilizada es Top k, descrita en la sección 2.5.2. La utilización de este método es sencilla y puede ser realizado de forma automática, sin intervención de un humano. En las secciones 3.2.2 y 3.3.2 se mostraron los problemas detectados en este método de evaluación.

En el caso de estudio de Wikipedia, se mostrarán y analizarán los resultados cuantitativos de la aplicación del método automático. Para el caso de estudio de prensa, se mostrará cuantitativamente cuánto afecta cada uno de los problemas detectados en el rendimiento reportado, utilizando el método automático. También se propondrá un método manual con el objetivo de obtener

una mejor estimación del valor real del rendimiento del sistema.

4.1. Recuperación de información de Wikipedia

El primer experimento correspondió a la recuperación de documentos para la búsqueda de respuestas en el conjunto de artículos de Wikipedia. Este problema es interesante ya que es uno de los más recurrentes en los trabajos de investigación sobre IR y QA, y es utilizado habitualmente como medida de referencia en Question Answering.

Como se explicó en el capítulo anterior, los conjuntos de datos disponibles para la validación de este problema no indican la versión de Wikipedia con la que fueron creados. Eso implica que los spans de de texto de las respuestas indicadas en el conjunto de datos de validación, pueden no encontrarse en el conjunto de artículos utilizado, ya que Wikipedia es un conjunto de documentos que cambia constantemente.

Al no contar con el conjunto de documentos con los que se crean los conjuntos de datos de evaluación, éstos no pueden ser utilizados satisfactoriamente para la evaluación de la tarea de recuperación de información.

Se presenta en la tabla 4.1 los resultados de los dos experimentos utilizando el método automático presentado en la sección 2.5.1 y las métricas Top 1, Top 5 y Top 20. Se utilizaron dos conjuntos de datos de evaluación diferentes.

Para la evaluación con el conjunto de datos MLQA, Puede leerse que para un 1,5 % de las preguntas el sistema devuelve en primer lugar un documento que contiene la respuesta indicada en el conjunto de datos de evaluación. Si se miran los primeros 5 documentos, aproximadamente un 4,4 % de las veces el sistema devuelve al menos un documento que contiene la respuesta indicada en el conjunto de datos de evaluación. En el caso de medir sobre los primeros 20 documentos, 10,3 % de las veces el sistema devuelve documentos que contienen la respuesta indicada en el conjunto de datos de evaluación. Para la evaluación con el conjunto de datos SQAC, el desempeño es ligeramente más bajo que con el conjunto de datos MLQA.

En este experimento, el sistema de recuperación de información no obtuvo buenos resultados. Si bien no se tiene una referencia del desempeño de otros sistemas de IR para el problema de recuperación de información de Wikipe-

Conjunto de datos	Cant. preguntas	Top 1	Top 5	Top 20
MLQA	5252	1.5	4.4	10.3
SQAC	867	1.1	4.2	9.6

Tabla 4.1: Resultado del experimento de Recuperación de información de Wikipedia en español.

dia en idioma español, puede tomarse para la comparación el desempeño de DPR para el mismo problema en idioma inglés. DPR tiene un desempeño para la métrica Top 20 de 63.2, utilizando como conjunto de datos de evaluación SQuAD, como se muestra en la tabla 2.3. Utilizando esta referencia, los resultados obtenidos en los experimentos realizados tienen un desempeño seis veces más bajos.

Al hacer un análisis de los principales problemas en los que incurre el sistema, vemos que es principalmente el método de evaluación y los recursos, los que no se amoldan al problema.

Para este caso de estudio no se realizó un análisis cuantitativo de los problemas detectados en el análisis de la sección 3.2.2. Sin embargo al observar las salidas del sistema se observa que:

- En varios casos el sistema efectivamente no es capaz de devolver documentos relevantes. La metodología de creación de los conjuntos de datos genera que muchas preguntas sean imposibles de contestar sin el contexto correspondiente. Si bien existe el conjunto de datos Natural Questions, creado de forma diferente para prevenir este problema, no hay un conjunto de datos similar en español, ni tampoco una traducción del mismo.
- En varios casos el sistema devuelve documentos que responden a la pregunta, pero el span de texto indicado como respuesta en el conjunto de datos de evaluación no pertenece a ningún documento del conjunto. Este es el problema de la coherencia de versiones entre los conjuntos de datos de documentos y de QA.

El origen de ambos problemas es el conjunto de datos de evaluación. Al utilizar conjuntos de datos de QA para evaluar la tarea de IR, se introducen nuevos problemas que no se presentan al momento de evaluar la extracción de respuestas. En la sección 4.2.2 se propone un método manual para poder

obtener una mejor estimación del valor real del rendimiento del sistema. Este método será aplicado para el caso de estudio de la recuperación de información de prensa, pero no para este caso de estudio.

4.2. Recuperación de información de prensa

El segundo experimento consistió en la recuperación de documentos para la búsqueda de de respuestas en el conjunto de noticias de dominio específico. El objetivo es poder contestar preguntas sobre noticias de un dominio específico. Dada la relevancia mundial y local de la pandemia, se eligió como dominio la enfermedad COVID-19 y hechos relacionados a la ésta.

La utilización del método automático descrito en la sección 2.5.1 es sencilla y puede ser realizada sin intervención de un humano. En la sección 3.3.2 se mostraron los problemas detectados en este método de evaluación aplicado a este caso de estudio.

Se presentarán a continuación los resultados aplicando el método automático.

4.2.1. Resultados utilizando el método automático

Se presenta en la tabla 4.2 los resultados de los experimentos. En todos los experimentos se utilizó el mismo conjunto de datos de evaluación, pero se utilizaron diferentes métodos de indexación de los documentos. Puede verse que el método de indexación no presenta diferencias notorias en el rendimiento.

Puede leerse que para un 12,3 % de las preguntas el sistema devuelve en primer lugar un documento que contiene la respuesta indicada en el conjunto de datos de evaluación. Si se miran los primeros 5 documentos, aproximadamente un 26 % de las veces el sistema devuelve al menos un documento que contiene la respuesta indicada en el conjunto de datos de evaluación. En el caso de medir sobre los primeros 20 documentos, 50.7 % de las veces el sistema devuelve documentos que contienen la respuesta indicada en el conjunto de datos de evaluación.

Si bien los resultados son mejores que para el caso de estudio de recuperación de información de Wikipedia, vemos que los valores para la métricas son bajos: aproximadamente 1 de cada 4 veces el sistema devuelve un documento

Conjunto de datos	cant. preguntas	Index	Top 1	Top 5	Top 20
QuALES	656	flat	12.3	26.5	50.7
QuALES	656	hnsw	12.3	26.5	50.7
QuALES	656	hnsw_sq	12.3	26.8	50.7

Tabla 4.2: Resultado del experimento de Recuperación de información de noticias de COVID

que responde la pregunta en los primeros 5 listados. Lamentablemente no contamos con una línea base para este problema, de forma de poder comparar los resultados con otros sistemas u otros idiomas. Si utilizamos como referencia el desempeño de DPR en idioma inglés utilizando documentos de Wikipedia y el conjunto de evaluación SQuAD, DPR tiene un desempeño Top 20 = 63.2, que es un poco más alto al desempeño obtenido en este experimento (24 % mejor).

Al realizar el análisis del caso de estudio, se detectaron diversos problemas que afectan el valor de rendimiento reportado por el método automático. A continuación se presentará un método que intenta remediarlos, de forma de obtener una mejor estimación del rendimiento.

4.2.2. Método de evaluación MEMTIR

Nos interesa poder obtener una mejor estimación del valor real del rendimiento del sistema, y así poder saber cuánto subestima el rendimiento del sistema el método automático tradicional de evaluación. Se propone entonces un método manual de evaluación de sistemas de IR, focalizándonos en el dominio de prensa. Llamaremos a este método MEMTIR: Manual Evaluation Method for Temporal Information Retrieval.

Se plantea el siguiente procedimiento manual de evaluación para ser utilizado sobre conjuntos de datos de prensa o con características temporales. Este mismo procedimiento también puede aplicarse para la evaluación en otros contextos, como en el caso de Wikipedia.

Precondición: En el conjunto de datos de evaluación para la tarea de IR, todas las preguntas deben tener respuestas y ser completas. Las preguntas son completas cuando pueden ser respondidas sin necesidad de información extra. Si a la pregunta le faltase contexto temporal, ésta puede ser mapeada al mo-

mento presente, o al período cuando ocurrió el evento al que hace referencia la pregunta.

Primer paso: evaluación automática. Se realiza en primera instancia una evaluación automática, analizando si las respuestas contenidas en el conjunto de datos de evaluación se encuentran o no en los documentos listados por el sistema. Con dicha información se utiliza la métrica de evaluación Top k. Este paso se corresponde a lo realizado en la sección 4.2.1.

Segundo paso: evaluación de falsos positivos. Respecto a las preguntas del conjunto de datos de evaluación que recibieron algún documento relevante en los primeros k devueltos por el sistema, se analizan los documentos clasificados como relevantes. En ellos se determina si el span encontrado responde a la pregunta, o si pertenece al documento pero en un contexto donde no responde a la pregunta. Para el primer caso, el documento permanece clasificado como relevante. Para el segundo caso, el documento se clasifica como no relevante, ya que no responde a la pregunta.

Un ejemplo: supongamos que tenemos en el conjunto de datos la pregunta: «*¿Donde inició la pandemia?*» y la respuesta asociada en el conjunto de datos de evaluación es «*China*». Ahora supongamos que el sistema devuelve los pasajes ordenados tal como indica la tabla 4.3:

Orden	Contenido documento	Clasificación
1	«*China es un país de Asia*»	Relevante
2	«*La pandemia nació en China*»	Relevante

Tabla 4.3: Ejemplos de documentos devueltos por el sistema de IR para la pregunta «*¿Donde inició la pandemia?*».

El primer documento devuelto contiene el span de texto indicado como respuesta en el conjunto de datos de evaluación, pero no responde a la pregunta planteada. Este documento debería reclasificarse como no relevante. El segundo documento contiene el span de texto indicado como respuesta en el conjunto de datos de evaluación y responde correctamente a la pregunta, por lo que este documento permanece clasificado como relevante.

Tercer paso: evaluación manual. Un humano evalúa para cada pregunta, si en los primeros k elementos devueltos por el sistema para ésta, se encuentra una respuesta a la pregunta, aunque la respuesta no se encuentre explícita en el conjunto de datos de evaluación. Para cada pregunta, se reclasifican los documentos devueltos, catalogando como relevantes aquellos documentos donde se encuentren respuestas a la pregunta.

Un ejemplo: supongamos que tenemos en el conjunto de datos la pregunta: «*¿Donde inició la pandemia?*» y la respuesta asociada en el conjunto de datos de evaluación es «*China*». Ahora supongamos que el sistema devuelve los pasajes ordenados tal como indica la tabla 4.4:

Orden	Contenido documento	Clasificación
1	«*La pandemia nació en Asia*»	No relevante
2	«*El virus fue visto por primera vez en Wuhan*»	No relevante

Tabla 4.4: Ejemplos de documentos devueltos por el sistema de IR para la pregunta «*¿Donde inició la pandemia?*».

Todos los documentos devueltos responden a la pregunta planteada, por lo que deberían considerarse como relevantes, aunque no contengan la respuesta indicada en el conjunto de datos de evaluación.

Cuarto paso: recalcular métrica Top k. teniendo la nueva clasificación de todos los pasajes, se recalcula la métrica Top k, para el k elegido.

Veamos a continuación un ejemplo completo de aplicación del método MEMTIR. Supongamos que en el conjunto de datos de evaluación se tienen solamente tres elementos, presentados en la tabla 4.5. Los resultados del sistema de IR, para cada ejemplo puede verse en la tabla 4.6.

Precondición: Para asegurarnos que todas las preguntas sean completas, mapeamos al momento presente la pregunta del ejemplo uno. Las preguntas del ejemplo dos y tres no requieren mapeo temporal. No es necesario eliminar ninguna pregunta, ya que todas tienen respuesta y son completas.

Ejemplo 1	
Pregunta	*¿Cuántos trabajadores de la educación pasaron al teletrabajo?*
Respuesta	*50 %*

Ejemplo 2	
Pregunta	*¿Cuándo aparecieron los primeros casos de Covid en el sudeste asiático?*
Respuesta	*enero*

Ejemplo 3	
Pregunta	*¿Dónde nació la pandemia?*
Respuesta	*China*

Tabla 4.5: Ejemplos del conjunto de datos de evaluación. El conjunto cuenta con tres ejemplos.

Primer paso: Se realiza la evaluación automática. Considerando la métrica Top 2, podemos decir que el sistema tiene un desempeño del 33.3 %: solo en 1 de los tres ejemplos el sistema devuelve documentos relevantes en las primeras dos posiciones.

Segundo paso: Se analiza el ejemplo 1, que fue el que obtuvo un documento relevante en los primeros dos devueltos. Se reclasifica el primer documento devuelto como No relevante, ya que si bien el span de texto aparece en el documento devuelto, éste no responde correctamente a la pregunta.

Tercer paso: Se analizan para todos los ejemplos los documentos devueltos clasificados como No relevantes. Se reclasifican como Relevantes el primer documento devuelto para el ejemplo 2, y ambos documentos devueltos para el ejemplo 3, ya que los tres documentos responden a las preguntas respectivas, aunque el span de texto no coincida con el definido en el conjunto de datos de evaluación. La nueva clasificación puede verse en la tabla 4.7.

Cuarto paso: Se recalcula la métrica Top 2, tomando la nueva clasificación de los documentos devueltos para cada ejemplo. El desempeño del sistema es ahora de 66.6 % ya que para dos de los tres ejemplos, el sistema devuelve documentos que responden la pregunta en las primeras dos posiciones.

Documentos devueltos para el ejemplo 1		
Orden	Contenido documento	Clasificación
1	«*El 50 % de las personas podría ser portador de COVID.*»	Relevante
2	«*Los educadores tuvieron que adaptarse.*»	No relevante
Documentos devueltos para el ejemplo 2		
Orden	Contenido documento	Clasificación
1	«*En el primer mes del 2020 aparecieron las primeras personas en el sur de Asia con la enfermedad.*»	No relevante
2	«*El virus fue visto por primera vez en Wuhan*»	No relevante
Documentos devueltos para el ejemplo 3		
Orden	Contenido documento	Clasificación
1	«*La pandemia nació en Asia*»	No relevante
2	«*El virus fue visto por primera vez en Wuhan*»	No relevante

Tabla 4.6: Ejemplos de documentos devueltos por el sistema de IR para el conjunto de datos de la tabla 4.5.

A continuación se mostrarán los resultados del sistema en el dominio de prensa, aplicando diferentes métodos. Cada método da solución a alguno de los problemas identificados en el capítulo anterior. De esta manera podremos realizar un análisis cuantitativo del impacto de cada problema detectado y determinar cuanto subestima en total el método automático utilizado actualmente, respecto al rendimiento real del sistema.

4.2.3. Resultados utilizando el método automático discriminado por preguntas

En la sección 3.3.2 se mostraron los problemas detectados para este caso de estudio. En esta sección se analizará cuánto afecta la falta de contexto en el rendimiento reportado por el método automático.

Recordemos que en el conjunto de datos de evaluación existen preguntas que un humano no es capaz de contestar sin un contexto, como por ejemplo las presentadas en la tabla 3.7. Además, dentro de las preguntas sin contexto,

Documentos devueltos para el ejemplo 1		
Orden	Contenido documento	Clasificación
1	«*El 50 % de las personas podría ser portador de COVID.*»	No relevante
2	«*Los educadores tuvieron que adaptarse.*»	No relevante

Documentos devueltos para el ejemplo 2		
Orden	Contenido documento	Clasificación
1	«*En el primer mes del 2020 aparecieron las primeras personas en el sur de Asia con la enfermedad.*»	Relevante
2	«*El virus fue visto por primera vez en Wuhan*»	No relevante

Documentos devueltos para el ejemplo 3		
Orden	Contenido documento	Clasificación
1	«*La pandemia nació en Asia*»	Relevante
2	«*El virus fue visto por primera vez en Wuhan*»	Relevante

Tabla 4.7: Reclasificación de documentos devueltos por el sistema de IR, luego de la ejecución de los primeros 3 pasos del método MEMTIR.

hay un subconjunto de ellas donde la única información faltante refiere a la temporalidad de la pregunta, como las presentadas en la tabla 3.8.

Se desea analizar cuantitativamente el impacto de la falta de contexto general y temporal en el rendimiento reportado del sistema mediante el método automático. Las conclusiones de esta sección aportaron a definir las precondiciones del método MEMTIR.

Se analizaron todas las preguntas del subconjunto de QuALES utilizado para la evaluación. Del total de 656 preguntas, 59 % son preguntas completas, con todo el contexto necesario para poder contestarlas. Tener 41 % de preguntas incompletas, que un humano no podría contestar, es un número demasiado grande para evaluar un sistema. Llamaremos de aquí en adelante QuALES_Comp al subconjuntos de QuALES de preguntas completas y QuALES_Incomp al subconjuntos de QuALES de preguntas incompletas.

Observando las preguntas completas, el desempeño medido en Top 5 es

de 28 %. Para las preguntas incompletas, la métrica Top 5 es de 25 %. Esta información expuesta en la tabla 4.8, muestra que efectivamente el sistema falla más cuando no se tiene el contexto suficiente en la pregunta, para poder recuperar documentos relevantes.

La primera conclusión del experimento es entonces: **La falta de contexto en las preguntas afecta el rendimiento de la recuperación de información.**

Subconjunto QuALES	% QuALES	Top 5
Preguntas completas	59	28
Preguntas incompletas	41	25

Tabla 4.8: Performance del sistema analizando preguntas completas e incompletas.

En la siguiente sección se analizarán las preguntas incompletas con el fin de entender el comportamiento del sistema dependiendo de la información faltante.

Análisis de preguntas incompletas

En el contexto de Question Answering sobre una colección de documentos con características de temporalidad se propone que una pregunta que no contenga ninguna expresión temporal, puede ser mapeada al momento presente, o al período cuando ocurrió el evento al que hace referencia la pregunta (J. Wang et al., 2022). Este mapeo se realiza asumiendo el tiempo de la pregunta, sin editarla explícitamente.

Si se realiza ese mapeo de forma manual, entonces el 86 % de las preguntas del conjunto de datos son completas. Para este nuevo conjunto el Top 5 es de 28 %, igual valor que para las preguntas completas originales. Para las 89 preguntas (14 %) que permanecen sin contexto aún luego de la suposición temporal, el el Top 5 es de 18 %. Llamaremos a dichos subconjuntos QuALES_Mapeo_Comp y QuALES_Mapeo_InComp respectivamente. Puede verse un resumen de esta información en la tabla 4.9.

En la figura 4.1 se muestra un esquema de los subconjuntos mencionados en esta sección. La primer subdivisión se realizó entre ejemplos donde con preguntas que pueden ser respondidas por un humano (completas) y las que

Subconjunto QuALES	% QuALES	Top 5
Preguntas completas (considerando mapeo temporal)	86	28
Preguntas incompletas (luego del mapeo temporal)	14	18

Tabla 4.9: Performance del sistema analizando preguntas luego del mapeo temporal

no (incompletas). Luego, considerando el mapeo temporal, algunas de las preguntas incompletas se transformaron en completas, y otras permanecieron sin poder ser respondidas, por falta de contexto.

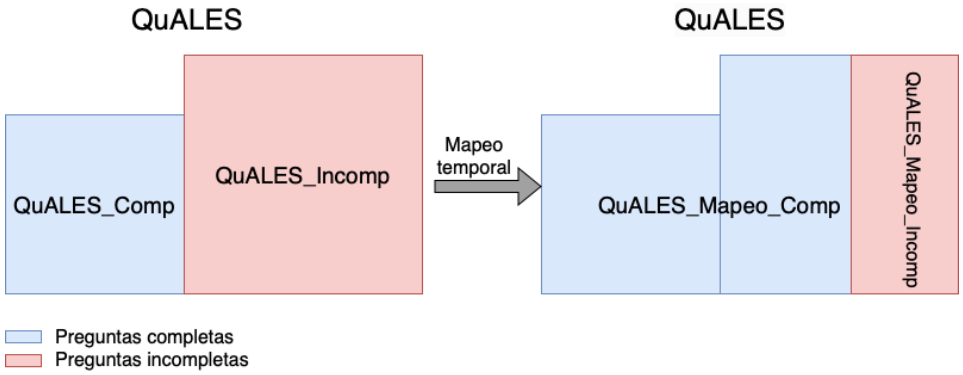

Figura 4.1: Diagrama de relación de los subconjuntos de QuALES.

Al analizar estos resultados, se ve el mismo desempeño del sistema en las preguntas completas originales y en las preguntas a las que solo les falta el contexto temporal. También puede verse que el desempeño en las preguntas con falta de contexto de otro tipo, es considerablemente menor. El resultado es interesante ya que el entrenamiento no se realizó con ninguna configuración particular para que el modelo aprendiera el mapeo en las preguntas sin expresiones temporales.

Podemos observar de estas estadísticas que: **El mapeo temporal es de alguna manera modelado por el sistema de recuperación de información, sin necesidad de ninguna configuración particular.**

4.2.4. Resultados utilizando método automático y análisis semántico manual

En la sección 3.3.2 se mostraron los problemas detectados para este caso de estudio. En esta sección se analizará cuánto afecta la simplificación de la evaluación en el rendimiento reportado por el método automático.

Como se mostró en la sección 2.5.1, la evaluación de los sistemas de IR para QA no utilizan los índices ni el contexto del conjunto de datos de evaluación, lo que puede llevar a una clasificación conceptualmente errónea. Dado que se define como relevante los documentos que contengan el span de texto de la respuesta, es posible que un documento sea catalogado como relevante, aunque no responda conceptualmente la pregunta.

Se desea analizar cuantitativamente el impacto de la simplificación de la evaluación en el rendimiento reportado del sistema mediante el método automático. Las conclusiones de esta sección aportaron a definir el segundo paso del método MEMTIR.

Analizaremos si efectivamente los documentos catalogados como relevantes responden correctamente a la pregunta (es decir, con la respuesta adecuada con el significado adecuado), o si simplemente contienen el span de texto, pero en un contexto donde no responde a la pregunta.

Se realizó este análisis estudiando el comportamiento en los subconjuntos de QuALES con preguntas completas e incompletas originales (Quales_Comp y Quales_Incomp), así como en los subconjuntos de QuALES con preguntas completas e incompletas luego del mapeo temporal (Quales_Mapeo_Comp y Quales_Mapeo_Incomp). Se analizaron los primeros 5 documentos devueltos por el sistema para cada pregunta.

Para el subconjunto de QuALES de preguntas completas originales, el 90 % de las veces que un documento (de los primeros 5 devueltos) es catalogado como relevante, éste contesta correctamente la pregunta. Si consideramos como relevantes los documentos que realmente son capaces de responder la pregunta, entonces la métrica Top 5 (llamemos a esta nueva métrica Top 5 semántico) para este subconjunto desciende de 28 % a 25 %.

Para el subconjunto de QuALES de preguntas incompletas originales, el 82 % de las veces que un documento (de los primeros 5 devueltos) es catalogado como relevante, éste contesta correctamente la pregunta. La métrica Top 5 semántico es para este subconjunto 21 %, cuando Top 5 es 25 %.

Para el subconjunto QuALES de preguntas completas luego del mapeo temporal, el 88 % de las veces que un documento (de los primeros 5 devueltos) es

catalogado como relevante, éste contesta correctamente la pregunta. La métrica Top 5 semántico es para este subconjunto 21 %, cuando Top 5 es 25 %.

Para el subconjunto QuALES de preguntas que permanecen incompletas luego del mapeo temporal, el 75 % de las veces que un documento (de los primeros 5 devueltos) es catalogado como relevante, éste contesta correctamente la pregunta. La métrica Top 5 semántico es para este subconjunto 13 %, cuando Top 5 es 18 %.

En la tabla 4.10 puede verse la información anterior de forma resumida.

Subconjunto QuALES	% clasificación correcta	Top 5	Top 5 semántico
QuALES_Comp	90	28	25
QuALES_Mapeo_Comp	88	28	24.9
QuALES_Incomp	82	25	21
QuALES_Mapeo_InComp	75	18	13

Tabla 4.10: Performance del sistema analizando Top 5 semántico

La simplificación de la evaluación del módulo de IR conlleva a un porcentaje de error para nada despreciable. En el caso de las preguntas incompletas, la clasificación incorrecta alcanza el 18 % en el conjunto original y un 25 % en el subconjunto de preguntas incompletas aún luego del mapeo temporal, lo que son valores muy altos. En el caso de los subconjuntos de preguntas completas (con y sin mapeo temporal), vemos que las estadísticas son prácticamente iguales, reforzando las conclusiones de la sección 4.2.3.

Se desprenden dos conclusiones. Por una lado, se dan más datos que refuerzan la conclusión de la sección 4.2.3. Además, podemos agregar que: **La falta de contexto temporal en las preguntas no es aquello que más perjudica el desempeño.**

Por otro lado, puede verse una **falencia en el modo en que se clasifican los documentos como relevantes, utilizando el proceso de evaluación estándar.**

4.2.5. Resultados utilizando el método MEMTIR

En la sección 3.3.2 se mostraron los problemas detectados para este caso de estudio. Uno de los problemas más importantes refiere a la metodología de

evaluación normalmente utilizada.

En el caso de estudio de prensa, los artículos pueden tener diferentes respuestas a la misma pregunta. Esto puede darse por temporalidad, variaciones del lenguaje, o por la naturaleza de la pregunta.

En esta sección se evaluará cada pregunta para determinar si los documentos devueltos contienen una respuesta, aunque ésta sea diferente a la del conjunto de datos de evaluación. Las conclusiones de esta sección aportaron a definir el tercer y cuarto paso del método MEMTIR.

Se desea saber entonces el desempeño real del sistema en el subconjunto de preguntas completas luego del mapeo temporal, QuALES_Mapeo_Comp (567 preguntas). Para ello, para los ejemplos en los que el sistema no devuelve en el Top 5 documentos relevantes (72 %), se analizan los primeros 5 documentos.

Para cada pregunta se leyeron los primeros 5 documentos devueltos buscando una respuesta a la pregunta, aunque la respuesta no se encuentre explícita en el conjunto de datos de evaluación. Como resultado se tiene que para el 23.4 % de las preguntas el sistema sí devuelve documentos relevantes en los primeros 5 puestos del listado. Para estos casos, la respuesta encontrada no es la del conjunto de datos de evaluación. Los motivos de esta diferencia pueden clasificarse en tres conjuntos:

- Para el 53.7 % de las preguntas el sistema devuelve documentos relevantes, pero éstos no son considerados como tales debido a problemas como el siguiente: El conjunto de datos de evaluación cuenta con la siguiente pregunta y respuesta asociada: «*¿Dónde empezó la pandemia?*» «*China*». Si el Sistema IR recupera en primer lugar un documento que dice que la pandemia empezó en Wuhan, éste es catalogado como no relevante porque no tiene la respuesta «*China*» que es lo que se tiene en el conjunto de datos de test.

- Para el 44.2 % de las preguntas el sistema devuelve documentos relevantes, pero éstos no son considerados como tales debido a problemas de temporalidad como el siguiente: El conjunto de datos de evaluación cuenta con la siguiente pregunta y respuesta asociada: «*¿Cuántos contagiados hubo hoy por covid?*» «*20*». Si el Sistema IR recupera en primer lugar un artículo que indica que hubo 30 contagiados (porque corresponde a un

día diferente al artículo con el que se hizo la pregunta del conjunto de datos de evaluación), éste es catalogado como no relevante. En principio no puede saberse cuál es la respuesta correcta porque no se tiene los datos temporales de los artículos. Se consideran como relevantes cualquiera de los 2 documentos.

- Para el 2.1 % de las preguntas el sistema devuelve documentos relevantes, pero éstos no son considerados como tales debido a que los spans de texto no coinciden exactamente. Ejemplo: a las pregunta «*¿Quién es Pablo Mieres?*» El conjunto de datos de evaluación indica «*El ministro de Trabajo y Seguridad Social*». Pero un artículo devuelto (diferente al utilizado para la creación del conjunto de datos) contiene el span «*El ministro de Trabajo*», y como no coinciden los spans, este documento es considerado como no relevante.

Entonces, luego de hacer una clasificación manual de los documentos, se tiene que la métrica Top 5 del sistema es el 43 %. En la tabla 4.11 puede verse este resultado y también el resultado de la evaluación automática quitando los falsos positivos (Top 5 semántico).

Top 5	Top 5 semántico	Top 5 manual
28	24.9	43

Tabla 4.11: Desempeño del sistema clasificando manualmente los documentos

Puede verse que el desempeño real del sistema es mucho mejor que lo que muestra la evaluación automática o semi-automática.

La evaluación manual arroja que el problema más importante que justifica esta diferencia es que el conjunto de datos de evaluación no tiene todas las respuestas posibles a una pregunta. Al ser creado utilizando solo un artículo, la respuesta está sesgada a lo que diga éste y las expresiones que se utilicen en él. Aquí hay una gran diferencia con los conjuntos de datos de preguntas sobre enciclopedias, ya que en este caso las preguntas casi siempre tienen una única respuesta. Un conjunto de datos de preguntas más abiertas, no debería crearse siguiendo la misma metodología que para conjuntos de datos de preguntas enciclopédicas.

El siguiente problema en importancia es que aunque realicemos un mapeo

de la temporalidad de la pregunta, es importante contar con la información temporal de la creación del artículo. Al no contar con esa información no se puede determinar exactamente si un documento es relevante o no.

A continuación se verá el último problema encontrado, intrínseco al sistema que se utilizó para los experimentos, pero que posiblemente no se encuentre en otros sistemas.

4.2.6. Resultados utilizando el método MEMTIR considerando artículos en lugar de pasajes

En la sección 3.3.2 se mostraron los problemas detectados para este caso de estudio. En esta sección se analizará cuanto afecta la implementación particular de DPR respecto a pasajes y documentos.

El sistema DPR utilizado para los experimentos devuelve pasajes de los documentos originales. Para la evaluación, el sistema verifica si el span de texto de la respuesta se encuentra en el pasaje, en lugar de verificar en el documento completo original que contiene el pasaje. Este método, particular de DPR, puede llevar a una evaluación equivocada del sistema.

Se desea analizar cuantitativamente el impacto de esta implementación en el rendimiento reportado del sistema mediante el método automático.

Se estudiaron los ejemplos en los que aún después de la evaluación manual, el sistema no devuelve documentos relevantes. En el 11 % de estos ejemplos, el sistema devuelve un pasaje del artículo utilizado para extraer la respuesta para el conjunto de datos de evaluación, pero no el pasaje que contiene la respuesta. Es decir, si el sistema devolviese todo el artículo del pasaje que clasifica como relevante, y no solo el pasaje, entonces la métrica de Top 5 sería de 49.2 %.

Esta estadística se obtuvo estudiando la pertenencia de los pasajes devueltos al artículo indicado en el conjunto de datos de evaluación. Es decir, no se leyeron los artículos de los pasajes devueltos para encontrar una respuesta a la pregunta, aunque ésta fuese diferente a la indicada en el conjunto de datos de evaluación. Extrapolando las estadísticas, es posible que el Top 5 aumentara un 10 % adicional si estudiáramos los artículos completos devueltos, y no solamente los pasajes.

4.2.7. Resumen del análisis

Como pudo verse en los incisos anteriores, la evaluación de la tarea de recuperación de información para conjuntos de datos de noticias no es trivial. Se puede ver que al analizar manualmente los resultados, el rendimiento real del sistema es casi el doble de lo que se obtiene al evaluar automáticamente el sistema. Siendo precisos, utilizando la métrica Top 5, el método de evaluación automático subestima el rendimiento del sistema un 75 %.

Para obtener el valor final de la métrica, se requirió analizar y seleccionar las preguntas útiles para la evaluación del sistema. Además, sobre ciertas preguntas se tomaron suposiciones sobre la temporalidad. También se requirió estudiar todos los pasajes devueltos para cada pregunta (en nuestro caso elegimos estudiar la métrica Top 5, por lo cual analizamos los 5 primeros pasajes devueltos). En caso de clasificar automáticamente un pasaje como relevante, se debió validar que no era un falso positivo y en caso de clasificar un pasaje como no relevante, se debió validar que no existiera alguna posible respuesta. Para finalizar, también se analizó la pertenencia de los pasajes devueltos por el sistema a los artículos completos.

En la tabla 4.12 se resumen todas estas etapas y sus resultados. El punto de partida de los documentos a utilizar es el conjunto de datos QuALES test sin las preguntas cuya respuesta es vacía, evaluado de forma automática. Luego se quitan las preguntas incompletas, y se utilizan solamente aquellas que tienen todo el contexto para ser contestadas, considerando también el supuesto temporal. En este caso la métrica mejora, pero al ajustar los falsos positivos, baja. Cuando estudiamos manualmente los resultados, vemos que el sistema tiene un desempeño mucho más alto de lo interpretado con el cálculo automático. Luego, si además analizamos los artículos de donde se obtienen las respuestas, ampliaríamos aún más el rendimiento final del sistema.

Podemos decir entonces que en casi la mitad de los casos (49.2 %), cuando se formula una pregunta completa, el sistema devuelve en los primeros cinco puestos la referencia a un artículo que contiene una respuesta.

Sintetizando algunas de las conclusiones del análisis de este experimento, tenemos que:

Conjunto de datos	Evaluación	Top 5
QuALES	Método automático	26.8
QuALES_Mapeo_Comp	Método automático	28
	Método automático y análisis semántico	24.9
	Método MEMTIR	43
	Método MEMTIR c/artículos	49.2

Tabla 4.12: Métricas del sistema de IR. Método automático y análisis semántico refiere a la evaluación automática y verificación que el span encontrado en los documentos responda correctamente a la pregunta. Método MEMTIR c/artículos refiere a la evaluación manual, pero considerando que la búsqueda de respuestas en el artículo, y no en el pasaje devuelto. Las métricas refieren al Top 5 de elementos devueltos.

- La falta de contexto en las preguntas afecta el desempeño de la recuperación de información.
- El mapeo temporal de las preguntas es de alguna manera modelado por el sistema de recuperación de información, sin necesidad de ninguna configuración particular.
- Se puede decir que la simplificación de la evaluación del módulo de IR conlleva a un porcentaje de error significativo.
- La falta de contexto temporal en las preguntas no es aquello que más perjudica el rendimiento.
- Existe una falencia en el modo en que se clasifican los documentos como relevantes, utilizando el proceso de evaluación estándar.
- El desempeño real del sistema es mucho mejor que lo que muestra la evaluación automática o semi-automática.
- El problema más importante que justifica la diferencia en la evaluación manual es que el conjunto de datos de evaluación no tiene todas las respuestas posibles a una pregunta.
- El siguiente problema en importancia refiere a la falta de información temporal de la creación de los artículos.

4.3. Conclusiones

En este capítulo se presentaron los resultados de los dos problemas abordados. Los problemas tienen características e inconvenientes diferentes, aunque

para ambos casos llegamos a la conclusión que el método de evaluación no se ajusta al problema.

Se propone un método manual para una mejor estimación del valor real del rendimiento del sistema, aplicado en el dominio de prensa. Sin embargo aquí hay una investigación abierta, para proponer un mecanismo más automatizado para la evaluación que no subestime tanto el rendimiento del sistema, y se acerque más a los resultados devueltos mediante el método MEMTIR.

En el próximo capítulo se realizarán las conclusiones finales del trabajo.

Capítulo 5

Conclusiones y trabajos a futuro

En esta tesis se realizó un análisis profundo de la tarea de recuperación de información para la búsqueda de respuestas en dos dominios diferentes: documentos enciclopédicos y documentos de prensa. Ambos problemas fueron abordados utilizando y adaptando recursos en idioma español. El estudio y análisis de estos problemas permitieron la detección de falencias en el método de evaluación de los sistemas de recuperación de información que es utilizado hoy en día en los trabajos de investigación en el área. Se propuso entonces el método manual de evaluación MEMTIR, focalizado en el dominio de prensa, con el fin de tener una mejor estimación del desempeño real de los sistemas de IR. La aplicación del método permitió medir cuantitativamente la afectación de cada falencia detectada en el desempeño reportado por la evaluación estándar.

En este capítulo se resumen los diferentes pasos que se siguieron para abordar los problemas, se comentan las principales dificultades encontradas y se obtienen conclusiones con respecto a la evaluación de la tarea de recuperación de información. Finalmente, presentamos algunas posibles líneas futuras de investigación, en base a estas conclusiones.

5.1. Resumen del proceso

Antes de profundizar en las tareas, primero se estudió la literatura sobre recuperación de información para la búsqueda de respuestas, tratando de caracterizar el problema. Del estudio, se desprende que el balance entre el desempeño a nivel de calidad de resultados y de tiempo de respuesta es el objetivo a alcanzar por los diferentes enfoques para resolver la tarea. Hoy en día se han

logrado los mejores resultados con arquitecturas basadas en BERT. Del estudio también se concluye que la investigación se centra en el dominio de los documentos enciclopédicos, sin plantearse fuertemente el análisis del desempeño de los sistemas en otros dominios con características diferentes, como el dominio de prensa. Además, hasta donde llega nuestro conocimiento, no hay ninguna línea de investigación que se plantee la forma de evaluación de los sistemas de IR.

Se definió entonces abordar el problema de recuperación de información para la búsqueda de respuestas en idioma español, en los dominios enciclopédicos y de prensa. Para realizar los experimentos se adaptó el módulo de IR del sistema DPR para ser utilizado en español, y se trabajó en la creación y adaptación de conjuntos de datos en español para el entrenamiento y la evaluación del sistema.

Del análisis de los dos problemas y sus resultados se observaron diversos inconvenientes en la evaluación. Nos focalizamos en el problema de prensa, menos estudiado generalmente, proponiendo un método de evaluación manual para estimar mejor el desempeño real del sistema. Al aplicar este método, pudo calcularse que el desempeño reportado por el método automático normalmente utilizado en los trabajos de investigación, subestima al menos un 75 % el desempeño del sistema. Este resultado es uno de los aportes más importantes de este trabajo.

5.2. Evaluación de los sistemas de IR

Los dos problemas abordados en esta tesis tienen características e inconvenientes diferentes, aunque para ambos casos llegamos a la conclusión que el método de evaluación no se ajusta al problema. Creemos que hay dos conclusiones muy relevantes producto de este trabajo que aportan valor a la hora de interpretar los resultados de la evaluación de los sistemas de IR.

La coherencia de versiones entre documentos y respuestas

Para realizar la evaluación estándar de los sistemas de IR o del pipeline completo IR-QA, es necesario contar con conjuntos de datos de evaluación de QA que provean el conjunto de documentos con el que fueron realizados. Esta restricción es importante ya que permite una mejor evaluación del sistema

(eliminando el problema de coherencia de versiones), y permite también una evaluación idéntica de todos los sistemas que deseen verificar su desempeño con el conjunto de datos. Es curioso que esta restricción no esté declarada ni se cumpla en ningún trabajo de investigación de los estudiados para la creación de este trabajo, independientemente del idioma. Esta nueva restricción es un aporte de esta tesis.

El ajuste del método de evaluación estándar a la tarea de IR

Quizá la conclusión más importante de este trabajo es que el método de evaluación estándar no se ajusta a la evaluación de la recuperación de información ni a la evaluación del pipeline completo. Existe una falencia en el modo en que se clasifican los documentos como relevantes utilizando conjuntos de datos de QA.

Clasificar los documentos como relevantes por tener el span de texto de la respuesta en su contenido, genera falsos positivos (cuando el span se encuentra pero no responde correctamente a la pregunta) y también falsos negativos (cuando el documento responde a la pregunta pero con un span de texto diferente al que se encuentra en el conjunto de datos). La simplificación de la evaluación del módulo de IR lleva a un porcentaje de error significativo.

Como parte de los aportes de este trabajo, se propuso el método manual MEMTIR para obtener una mejor estimación del rendimiento real del sistema, para experimentos que involucren temporalidad. El procedimiento declara de forma concisa cada paso para que pueda ser ejecutado de forma objetiva para poder comparar sistemas de esta índole. Aquí hay una investigación abierta, para proponer un mecanismo más sencillo de evaluación.

5.3. Otras contribuciones

Además de las conclusiones detalladas en la sección anterior, como producto de esta tesis se entrega la comunidad una herramienta de IR para ser utilizada en idioma español. Si bien existen diferentes enfoques que pueden ser aplicados independientemente del idioma subyacente de documentos y consultas, al momento de la escritura de esta tesis no existía ningún sistema de IR que utilizara modelos del lenguaje como BERT en idioma español.

Los enfoques que utilizan BERT en el contexto de IR para el inglés son el

estado del arte actual para la tarea de IR ad hoc. Con la adaptación de DPR se pretendió aportar una mejora a la tarea de IR para el español.

También, hasta donde llega nuestro conocimiento, este es el primer trabajo que estudia la recuperación de información para la búsqueda de respuestas trabajando con información temporal y modelos del lenguaje. Si bien existen trabajos de investigación sobre las particularidades de temporalidad de las noticias, hasta donde llega nuestro conocimiento, no existe un trabajo que utilice modelos del lenguaje para la recuperación de información y que aborde e investigue el problema de la temporalidad, independientemente del idioma.

En los trabajos existentes, los conjuntos de datos utilizados para la evaluación son generados cuidadosamente, y no comprenden preguntas naturales que un usuario pudiese hacer a un sistema para satisfacer una necesidad. Los conjuntos de datos utilizados en estos trabajos utilizan preguntas que casi siempre tienen una única respuesta, omitiendo algunos de los problemas importantes para la evaluación.

En esta tesis, al utilizar un conjunto de datos que contiene preguntas reales, se pudo realizar un análisis en profundidad del comportamiento del sistema de IR. Pudo verse que la falta de contexto en las preguntas afecta fuertemente el desempeño de la recuperación de información, aunque la falta de contexto temporal en las preguntas no es aquello que más perjudica el desempeño. En el módulo de IR de DPR adaptado al español, el mapeo temporal de las preguntas es de alguna manera modelado por el sistema, sin necesidad de ninguna configuración particular. Esto es muy relevante, ya que el sistema no se entrenó explícitamente para que aprendiera estas particularidades. Sin embargo, aunque realicemos un mapeo de la temporalidad de la pregunta, es importante contar con la información temporal de la creación del artículo. Al no contar con esa información no se puede determinar exactamente si un documento es relevante o no.

El problema más importante que justifica la diferencia en la evaluación manual y la automática es que el conjunto de datos de evaluación no tiene todas las respuestas posibles a una pregunta. Este problema se evita al utilizar para la evaluación un conjunto de datos con preguntas que casi siempre tienen una única respuesta. El siguiente problema en importancia que justifica la diferencia en la evaluación manual y la automática es la falta de información temporal de la creación del artículo.

5.4. Trabajo futuro

Los resultados obtenidos permiten observar que existen varias líneas de investigación abiertas en el área que pueden servir como orientación para futuros trabajos.

El módulo de IR de DPR permite configuraciones adicionales a las realizadas en este trabajo, que podrían llegar a mejorar el desempeño del sistema.

La primera de ellas refiere a los pasajes negativos. El sistema original DPR utiliza, además de ejemplos positivos, ejemplos negativos. Es decir, para una pregunta, se indican pasajes donde se puede encontrar la respuesta (ejemplos positivos), y otros en los que no (ejemplos negativos). En los conjuntos de datos utilizados en los experimentos no se incluyeron pasajes negativos, pero sería recomendable realizar experimentos con ellos. Según los autores de DPR, la inclusión de ejemplos negativos, y la buena selección de éstos, son trascendentes para la mejora de performance del sistema.

Respecto a los modelos de lenguaje utilizados, en este trabajo se utilizó el modelo BETO uncased, pero existen otros modelos para ser utilizados como encoders, los cuales podrían ser explorados. En particular los modelos de RoBERTa entrenados con la Biblioteca Nacional de España, podrían aportar mejoras respecto al modelo utilizado.

Por último, las limitaciones de BERT respecto al largo del token de entrada al modelo generan que los artículos originales deban ser subdivididos en pasajes, los cuales luego son transformados en vectores densos. El sistema DPR devuelve una lista ordenada de pasajes, y no de artículos. Los experimentos muestran que en ocasiones el sistema devuelve artículos que contienen las respuestas, pero la respuesta no se encuentra en el pasaje devuelto. Una mejora al sistema podría ser devolver los artículos completos que contienen los pasajes ya que de hecho es eso lo que se espera de un sistema de IR.

Por otro lado, focalizándonos en el problema de la temporalidad en la recuperación de información, podría ser muy útil el entrenamiento del modelo timeBERT con conjuntos de datos en español, como newsQA_ES. Este modelo incorpora información temporal, pero fue entrenado en idioma inglés. Sería interesante poder utilizarlo en idioma español en el módulo de recuperación de información.

Como línea más importante, se tiene que avanzar en la automatización de la evaluación de los sistemas de recuperación de información es clave. El método MEMTIR permite una mejor estimación del desempeño del sistema, pero es un método manual, costoso de implementar cuando el conjunto de evaluación es grande.

En contextos con información temporal, dos caminos son posibles: el primero consiste en definir un método para la creación del conjunto de datos de evaluación y una vez que éste cumpla con las condiciones, poder utilizar métricas automáticas. Este camino implica que el conjunto de datos contenga menos preguntas realistas. El segundo camino implicaría repensar la forma de evaluación de la tarea, de forma de no incurrir en problemas solamente para reutilizar recursos.

5.5. Conclusiones finales

En este trabajo se analizaron diferentes técnicas de recuperación de información que son utilizadas en el contexto de búsqueda de respuestas, en documentos de prensa y en documentos enciclopédicos en el idioma español.

El análisis en documentos de prensa, que involucran temporalidad, permitió ver que existen mejoras a desarrollar para este caso de estudio. Los enfoques que son el estado del arte para la recuperación de información, aún no están completamente adaptados para manejar y combinar la información temporal presente en metadatos, documentos y consultas.

Como parte central de los aportes de este trabajo, el estudio deja a la vista un gran vacío referido a la evaluación de los sistemas de IR. Se propuso el método manual MEMTIR, para obtener una mejor estimación del rendimiento real de los sistemas de IR, pero se requiere un método automatizado, que sea sencillo de aplicar y pueda estimar de forma similar el desempeño de los sistemas.

Referencias bibliográficas

Alonso, O., Strötgen, J., Baeza-Yates, R. y Gertz, M. (2011). Temporal Information Retrieval: Challenges and Opportunities. *TWAW Workshop, WWW 2011*, *707*.

Arora, S., Liang, Y. y Ma, T. (2017). A Simple but Tough-to-Beat Baseline for Sentence Embeddings. *International Conference on Learning Representations*.

Asai, A., Hashimoto, K., Hajishirzi, H., Socher, R. y Xiong, C. (2020). Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering.

Attardi, G. (2015). WikiExtractor.

Baudiš, P. y Šedivý, J. (2015). Modeling of the Question Answering Task in the YodaQA System. En J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. Jones, E. San Juan, L. Capellato y N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (pp. 222-228). Springer International Publishing.

Beltagy, I., Peters, M. E. y Cohan, A. (2020). Longformer: The Long-Document Transformer.

Boom, C. D., Canneyt, S. V., Demeester, T. y Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, *80*, 150-156.

Bordes, A., Usunier, N., Chopra, S. y Weston, J. (2015). Large-scale Simple Question Answering with Memory Networks.

Campos, R., Dias, G., Jorge, A. y Jatowt, A. (2014). Survey of Temporal Information Retrieval and Related Applications. *ACM Computing Surveys*, *47*, 1-41.

Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H. y Pérez, J. (2020). Spanish Pre-Trained BERT Model and Evaluation Data. *PML4DC at ICLR 2020*.

Carrino, C. P., Costa-jussà, M. R. y Fonollosa, J. A. R. (2019). Automatic Spanish Translation of the SQuAD Dataset for Multilingual Question Answering.

Chen, D., Fisch, A., Weston, J. y Bordes, A. (2017). Reading Wikipedia to Answer Open-Domain Questions.

Croft, W. B. y Harper, D. J. (1979). Using Probabilistic Models of Document Retrieval without Relevance Information. *J. Documentation*, *35*, 285-295.

Devlin, J., Chang, M.-W., Lee, K. y Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Furnas, G. W., Landauer, T. K., Gomez, L. M. y Dumais, S. T. (1987). The Vocabulary Problem in Human-System Communication. *Commun. ACM*, *30*(11), 964-971.

Gao, L., Dai, Z., Chen, T., Fan, Z., Durme, B. V. y Callan, J. (2021). Complementing Lexical Retrieval with Semantic Residual Embedding.

Guo, J., Fan, Y., Ai, Q. y Croft, W. B. (2016). A Deep Relevance Matching Model for Ad-hoc Retrieval. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*.

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., Gonzalez-Agirre, A., Armentano-Oller, C., Rodriguez-Penagos, C. y Villegas, M. (2021). Spanish Language Models.

Guu, K., Lee, K., Tung, Z., Pasupat, P. y Chang, M.-W. (2020). REALM: Retrieval-Augmented Language Model Pre-Training.

Jaleel, N., Allan, J., Croft, W., Diaz, F., Larkey, L., Li, X., Smucker, M. y Wade, C. (2004). UMass at TREC 2004: Novelty and hard.

Jégou, H., Douze, M. y Schmid, C. (2011). Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*, 117-128.

Jiexin, W., Jatowt, A., Faerber, M. y Yoshikawa, M. (2021). Improving question answering for event-focused questions in temporal collections of news articles. *Information Retrieval Journal*, *24*, 1-26.

Johnson, J., Douze, M. y Jégou, H. (2017). Billion-scale similarity search with GPUs.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*, 11-21.

Joshi, M., Choi, E., Weld, D. S. y Zettlemoyer, L. (2017). TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension.

Jurafsky, D. y Martin, J. H. (2020). *Speech and Language Processing (3rd ed. draft)*.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D. y Yih, W.-t. (2020). Dense Passage Retrieval for Open-Domain Question Answering.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q. y Petrov, S. (2019). Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, *7*, 453-466.

Lee, K., Chang, M.-W. y Toutanova, K. (2019). Latent Retrieval for Weakly Supervised Open Domain Question Answering.

Lewis, P., Oğuz, B., Rinott, R., Riedel, S. y Schwenk, H. (2019). MLQA: Evaluating Cross-lingual Extractive Question Answering. *arXiv preprint arXiv:1910.07475*.

Lin, J., Nogueira, R. y Yates, A. (2021). Pretrained Transformers for Text Ranking: BERT and Beyond.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. y Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Ma, X., Sun, K., Pradeep, R. y Lin, J. (2021). A Replication Study of Dense Passage Retriever.

Malkov, Y. A. y Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs.

Manning, C. D., Raghavan, P. y Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Manning, C. D., Raghavan, P. y Schütze, H. (2009). An Introduction to Information Retrieval (Draft).

Maron, M. E. y Kuhns, J. L. (1960). On Relevance, Probabilistic Indexing and Information Retrieval. *J. ACM*, *7*(3), 216-244.

Mikolov, T., Chen, K., Corrado, G. y Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space.

Miller, A., Fisch, A., Dodge, J., Karimi, A.-H., Bordes, A. y Weston, J. (2016). Key-Value Memory Networks for Directly Reading Documents. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1400-1409.

Mitra, B., Nalisnick, E., Craswell, N. y Caruana, R. (2016). A Dual Embedding Space Model for Document Ranking.

Nesmachnow, S. y Iturriaga, S. (2019). Cluster-UY: Collaborative Scientific High Performance Computing in Uruguay. En M. Torres y J. Klapp (Eds.), *Supercomputing* (pp. 188-202). Springer International Publishing.

Nogueira, R. y Cho, K. (2020). Passage Re-ranking with BERT.

Nogueira, R., Yang, W., Cho, K. y Lin, J. (2019). Multi-Stage Document Ranking with BERT.

Pennington, J., Socher, R. y Manning, C. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543.

Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W. X., Dong, D., Wu, H. y Wang, H. (2021). RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering.

Rajpurkar, P., Zhang, J., Lopyrev, K. y Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383-2392.

Reimers, N. y Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982-3992.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. y Gatford, M. (1994). Okapi at TREC-3. *TREC*.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. En G. Salton (Ed.), *The Smart retrieval system - experiments in automatic document processing* (pp. 313-323). Englewood Cliffs, NJ: Prentice-Hall.

Salton, G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc.

Salton, G., Wong, A. y Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Commun. ACM*, *18*(11), 613-620.

Sandhaus, E. (2008). *The New York Times Annotated Corpus.*

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. y Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631-1642.

Taulé, M., Martı, M. A. y Recasens, M. (2008). AnCora: Multilevel Annotated Corpora for Catalan and Spanish. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08).*

Taylor, M., Zaragoza, H., Craswell, N., Robertson, S. y Burges, C. (2006). Optimisation Methods for Ranking Functions with Multiple Parameters. *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, 585-593.

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P. y Suleman, K. (2016). NewsQA: A Machine Comprehension Dataset.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. y Polosukhin, I. (2017). Attention Is All You Need.

Voorhees, E. (1994). Query Expansion Using Lexical-Semantic Relations., 61-69.

Wang, J., Jatowt, A., Färber, M. y Yoshikawa, M. (2020). Answering Event-Related Questions over Long-Term News Article Archives. En J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva y F. Martins (Eds.), *Advances in Information Retrieval* (pp. 774-789). Springer International Publishing.

Wang, J., Jatowt, A. y Yoshikawa, M. (2022). TimeBERT: Enhancing Pre-Trained Language Representations with Temporal Information.

Wang, Z., Ng, P., Ma, X., Nallapati, R. y Xiang, B. (2019). Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5878-5882.

Weinberger, K., Dasgupta, A., Attenberg, J., Langford, J. y Smola, A. (2010). Feature Hashing for Large Scale Multitask Learning.

Yan, M., Li, C., Wu, C., Bi, B., Wang, W., Xia, J. y Si, L. (2019). IDST at TREC 2019 Deep Learning Track: Deep Cascade Ranking with

Generation-based Document Expansion and Pre-trained Language Modeling. En E. M. Voorhees y A. Ellis (Eds.), *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*. National Institute of Standards; Technology (NIST).

APÉNDICES

Apéndice 1

Seteo de entorno en ClusterUY

En este apéndice se detallan los pasos necesarios para la configuración del entorno en ClusterUY para la utilización del sistema DPR.

- crear entorno virtual Conda con Python 3.7: `conda create --name DPRLucia python=3.7`
- Activar entorno virtual: `conda activate DPRLucia`
- Instalar pytorch 1.2.0+: `conda install pytorch torchvision torchaudio -c pytorch`
- Instalar RUST: `curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh`
- exportar PATH: `export PATH="`$HOME/.cargo/bin$`:PATH"`
- Instalar compilador C: `conda install -c conda-forge c-compiler`
- Instalar transformers: `pip install transformers`
- Instalar wget (necesario para bajar los conjuntos de datos): `pip install wget`
- Instalar hydra (necesario para entrenar encoder): `pip install hydra`
- Instalar jsonlines (necesario para entrenar encoder): `pip install jsonlines`
- Instalar Spacy (necesario para entrenar encoder): `pip install spacy`
- Descargar módulo de Spacy (necesario para entrenar encoder): `python -m spacy download en_core_web_sm`
- Instalar DPR: `git clone https://github.com/facebookresearch/DPR.git ; cd DPR ; pip install .`

Apéndice 2

Adaptación de DPR para su utilización en idioma español

En este apéndice se detallan las etapas ejecutadas para la adaptación del módulo DPR para la utilización en idioma español.

2.1. Seteo de entorno

El sistema DPR está testeado en Python 3.6+ y PyTorch 1.2.0+. DPR se basa en bibliotecas de terceros para la implementación del encoder. Actualmente es compatible con los modelos de encoder de Huggingface (versión <= 3.1.0) BERT, Pytext BERT y Fairseq RoBERTa. Debido a la generalidad del proceso de tokenización, DPR utiliza tokenizadores Huggingface.

Para la adaptación del sistema, debido al intenso consumo de memoria y GPU, el entorno de ejecución se configuró en ClusterUY (Nesmachnow y Iturriaga, 2019). Se utilizó un entorno virtual de Conda con Python 3.7, donde se instalan los requerimientos del sistema (Pytorch, RUST, C, transformers, wget, hydra, jsonlines y Spacy). Finalmente se clonó el repositorio del proyecto DPR: https://github.com/facebookresearch/DPR.git (detalles de versiones y comandos en el apéndice A)

2.2. Formato de conjuntos de datos

El sistema DPR requiere que los conjuntos de datos cumplan con un formato específico, detallado a continuación.

Conjunto de documentos: Archivo separado por tabuladores tsv con los campos `[id sentence title]`. BERT tiene como limitación no poder procesar entradas de más de 512 tokens. Habitualmente los documentos son más largos de los que BERT es capaz de procesar, por lo que se realiza una división de los documentos. Se dividen los artículos en secciones de como máximo 450 tokens (el valor es cota superior, ya que no se cortan oraciones) y en cada fila del conjunto de datos se mantiene el id del documento original y el título de éste, para posterior indexado.

Conjunto de datos de preguntas, contextos y respuestas: Se espera como entrada un archivo json con el siguiente formato.

```
[
  {
   "question": "....",
   "answers": ["...", "...", "..."],
   "positive_ctxs": [{
      "title": "...",
      "text": "...."
   }],
   "negative_ctxs": ["..."],
   "hard_negative_ctxs": ["..."]
  },
  ...
]
```

Cada ítem cuenta con una pregunta y potencialmente varias respuestas. Por cada pregunta se cuenta con una lista de contextos donde está la respuesta `positive_ctxs`, de los cuales se tiene el texto y el título. Las secciones `negative_ctxs` y `hard_negative_ctxs` tienen el mismo formato que la sección `positive_ctxs` pero refieren a contextos donde no aparece la respuesta. En los experimentos las secciones referentes a contextos negativos no fueron utilizadas.

Conjunto de datos de preguntas y respuestas: Se espera un archivo tsv con los campos `[pregunta [respuesta1, respuesta2, ...]]`, donde la

lista de respuestas debe tener al menos un elemento. Para poder realizar la evaluación correctamente, los spans de texto de las respuestas deben estar contenidos en algún ítem de la colección de documentos.

2.3. Formato de salida de evaluación

Al ejecutar el script de inferencia, para cada pregunta del conjunto de datos de validación, se devuelve una lista de ordenada de documentos. Se indica para cada uno un puntaje de similitud, y si efectivamente contiene o no alguna de las posibles respuestas a la pregunta, indicadas en el conjunto de datos de evaluación. La herramienta escribe los resultados recuperados en un json con el siguiente formato:

```
[
    {
        "question": "...",
        "answers": ["...", "...", ... ],
        "ctxs": [
            {
                "id": "...", # id del documento de la base de datos
                    (tsv)
                "title": "",
                "text": "....",
                "score": "...", # puntaje de retrieval
                "has_answer": true|false
     },
]
```

Los resultados se ordenan por su puntuación de similitud, de más relevante a menos relevante. Además, el script devuelve en la salida estándar la medida de evaluación del desempeño del sistema mediante la métrica Top K (para $k \in [1..100]$ por defecto, pero se puede configurar la cota superior en el archivo `dense_retriever.yaml` en la clave `n_docs`).

2.4. Cambios de configuración

Para la adaptación del sistema DPR (módulo IR) se debió configurar los conjuntos de datos a utilizar para cada uno de los experimentos, así como configurar el modelo de encoder pre-entrenado a utilizar para el idioma español. DPR es compatible con los modelos de encoder de Huggingface (versión $\leq$ 3.1.0) BERT, Pytext BERT y Fairseq RoBERTa. Algunos de los modelos entrenados para el español disponibles en Huggingface y compatibles con DPR son:

- BETO fine tunned con squad [1]
- BETO cased [2]
- BETO uncased [3]
- RoBERTa large (para español) [4]
- RoBERTa base (para español) [5]

Para los experimentos se utilizó el modelo de encoder BETO uncased. Las modificaciones se realizaron en los siguientes archivos:

- `conf/ctx_sources/default_sources.yaml`: se configura en la clave `dpr_articles` la ruta correspondiente al conjunto de datos con el split de Artículos.

- `conf/datasets/encoder_train_default.yaml`: se configura una clave por cada conjunto de datos de train y dev, aportando el tipo de archivo (json o csv), así como la ruta al conjunto de datos.

- `conf/datasets/retriever_default.yaml`: se configura el conjunto de datos de validación a utilizar, aportando el tipo de archivo (json o csv), así como la ruta al conjunto de datos.

- `conf/encoder/hf_bert.yaml`: se configura el tipo de modelo, el nombre de configuración en HuggingFace para apuntar al modelo correcto, así

[1] https://huggingface.co/mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es
[2] https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased
[3] https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased
[4] https://huggingface.co/BSC-TeMU/roberta-large-bne
[5] https://huggingface.co/BSC-TeMU/roberta-base-bne

como otros parámetros. Pueden crearse otros archivos de configuración y para apuntar a éstos, o modificar el indicado. En este trabajo se optó por modificar el archivo original.

- `conf/dense_retriever.yaml`: aquí se pueden cambiar configuraciones referidas al tipo de indexación a utilizar, la cantidad de documentos a devolver para cada pregunta y la cantidad de workers de validación (puede ser necesario este cambio dependiendo de la capacidad del hardware y software donde se está ejecutando el proceso).

- `conf/biencoder_train_cfg.yaml`: si se desea partir de un checkpoint, este puede configurarse en las claves `model_file` y `checkpoint_file_name`. También es necesario setear en true: `ignore_checkpoint_optimizer`.