Trabajo Fin de Máster: Aplicación de modelos basados en Transformers a la validación de respuestas

Trabajo Fin de Máster

Juan Manuel López García

Trabajo de investigación para el

Máster Universitario en Tecnologías del Lenguaje

Universidad Nacional de Educación a Distancia

Dirigido por el

Prof. Dr. D. Álvaro Rodrigo Yuste

Septiembre 2022

Resumen

En la Validación de Respuestas se valora la corrección de las soluciones generadas por un sistema de Búsqueda de Respuestas. Se trata de decidir si las respuestas a una pregunta son apropiadas de acuerdo con los contenidos de un texto determinado. Apenas hay evidencia de que esta funcionalidad esté siendo implementada a partir de modelos de Aprendizaje Profundo y, aún menos, por medio de redes neuronales que incorporen mecanismos atencionales, como en los modelos *Transformer*. Las técnicas de Aprendizaje Profundo y, en particular, los modelos *Transformer*, han adquirido gran relevancia en prácticamente todas las áreas del Procesamiento del Lenguaje Natural debido a sus cualidades y rendimiento.

En este trabajo proponemos el uso de modelos *Transformer* en la implementación de subsistemas que utilicen la Implicación Textual para realizar tareas de Validación de Repuestas. Los modelos implementados se evalúan sobre los conjuntos de datos de prueba desarrollados durante los ejercicios del *Answer Validation Exercise* (AVE). Los resultados obtenidos en las evaluaciones indican que estos modelos pueden ofrecer rendimientos superiores a los de las técnicas probadas durante los AVE por lo que deben ser considerados métodos adecuados para llevar a cabo labores de Validación de Respuestas.

Abstract

Answer Validation systems assess the accuracy of Question Answering responses by deciding whether the answers to a question are appropriate according to the content of a given text. There is hardly any evidence that this functionality is being implemented from Deep Learning models and, even less, through neural networks that incorporate attentional mechanisms, as in *Transformer* models. Deep Learning techniques and in particular, *Transformer* models, have acquired great relevance in basically all areas of Natural Language Processing due to their features and performance.

In this thesis we propose the use of *Transformer* models in the implementation of subsystems using Textual Entailment to perform Answer Validation tasks. The implemented models are evaluated on test datasets developed during the three *Answer Validation Exercise* (AVE) editions. The results obtained in the evaluations indicate that these models can provide superior performance than those tested during the AVEs and therefore should be considered as suitable methods for performing Answer Validation tasks.

Índice general

1. Introducción....1
1.1. Propuesta y objetivos....2
1.2. Estructura del documento....3

2. Trabajos relacionados....5
2.1. Validación de Respuestas e Implicación Textual....5
2.2. Evaluación de sistemas de Validación de Respuestas....7
2.3. Métodos de validación....9
2.3.1. Solapamiento léxico....10
2.3.2. Uso de entidades nombradas....11
2.3.3. Análisis semántico....12
2.3.4. Representación lógica....13
2.3.5. Validación basada en recursos web....14
2.3.6. Aprendizaje automático....15
2.3.7. Aprendizaje profundo....16
2.4. Recapitulación....19

3. Marco de evaluación....21
3.1. Colecciones de evaluación....21
3.2. Métricas de evaluación....23

4. Propuestas....27
4.1. Modelos empleados....27
4.2. Colecciones de datos....32
4.3. Estrategias utilizadas....36
5. Evaluación....41
5.1. Resultados de los modelos ajustados con RTE, AVE o SNLI....41
5.2. Resultados de los modelos ajustados con SNLI, MNLI o Hans....45
5.3. Resultados de los modelos ajustados con SNLI, MNLI, FEVER-NLI y ANLI....47
5.4. Resultados de los modelos ajustados con SICK_ES, AVE o XNLI_ES....50
6. Discusión....55
7. Conclusiones y trabajo futuro....59
7.1. Conclusiones....59
7.2. Trabajo futuro....59
Bibliografía....61

Índice de tablas

Tabla 1. Composición de los conjuntos de prueba para inglés y castellano en AVE 2006....22

Tabla 2. Composición de los conjuntos de prueba para inglés y castellano en AVE 2007....22

Tabla 3. Composición de los conjuntos de prueba para inglés y castellano en AVE 2008....23

Tabla 4. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa RTE 1, 2 y 3 (Dev & Test), fracciones de SNLI o una combinación de ambas....42

Tabla 5. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa RTE 1, 2 y 3 (Dev & Test), AVE 07 Dev, fracciones de SNLI o una combinación de ellos....43

Tabla 6. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa RTE 1, 2 y 3 (Dev & Test), AVE 08 Dev, fracciones de SNLI o una combinación de ellos....44

Tabla 7. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, Hans o una combinación de ellos....45

Tabla 8. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, Hans o una combinación de ellos.....46

Tabla 9. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, Hans o una combinación de ellos.....47

Tabla 10. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, FEVER-NLI y ANLI (R1, R2, R3).....48

Tabla 11. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, FEVER-NLI y ANLI (R1, R2, R3).....49

Tabla 12. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, FEVER-NLI y ANLI (R1, R2, R3).....50

Tabla 13. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con el conjunto de entrenamiento en lengua española SICK_ES.....51

Tabla 14. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con el conjunto de entrenamiento en lengua española SICK_ES combinado, en algunos casos, con AVE 07 Dev.....51

Tabla 15. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con el conjunto de entrenamiento en lengua española SICK_ES combinado, en algunos casos, con AVE 08 Dev.....52

Tabla 16. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con el conjunto de entrenamiento en lengua española XNLI_ES.....53

Tabla 17. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con el conjunto de entrenamiento en lengua española XNLI_ES.....53

Tabla 18. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con el conjunto de entrenamiento en lengua española XNLI_ES.....54

Capítulo 1

Introducción

Un sistema de Búsqueda de Respuestas (*Question Answering* -QA-) es un sistema automático capaz de responder preguntas formuladas en lenguaje natural ofreciendo respuestas cortas y precisas. El objetivo de la búsqueda de respuestas es el de identificar y presentar al usuario una respuesta real y sucinta, en lugar de devolver, como hacen los sistemas de Recuperación de Información, una serie de documentos relacionados con la pregunta y que pueden contener o no la respuesta buscada [24].

La mayoría de los sistemas de Búsqueda de Respuestas ofrecen siempre un resultado, independientemente de si el conjunto de respuestas candidatas que haya reunido el sistema contiene resultados correctos o no [33]. Esto puede malograr en gran medida la experiencia de usuario, especialmente cuando estos tienen dificultad para valorar si las respuestas que reciben del sistema son ciertas o no. La Validación de Respuestas aborda este problema determinando si el conjunto de respuestas candidatas contiene alguna respuesta correcta y, solo en ese caso, permite que el sistema devuelva un resultado [46].

La Validación de Respuestas (también conocida en inglés como *Answer Validation* o *Answer Triggering*) trata, por tanto, sobre el desarrollo y la evaluación de subsistemas que pretenden validar la corrección de las respuestas generadas por un sistema de Búsqueda de Respuestas. La validación de

respuestas automática es útil para mejorar el rendimiento de los sistemas de Búsqueda de Respuestas y para ayudar a los humanos a valorar los resultados producidos por estos [2][20]. También puede utilizarse cuando simultáneamente se reciben respuestas de varios sistemas y es necesario discernir cuál es la más adecuada. Es apropiada, asimismo, para validar respuestas aportadas por humanos.

Tradicionalmente se han utilizado técnicas como el solapamiento de textos, el reconocimiento de entidades nombradas, la similaridad semántica o la inferencia lógica para llevar a cabo la Validación de Respuestas, pero apenas hay evidencia de que esta funcionalidad se esté implementando a partir de modelos de Aprendizaje Profundo y, menos aún, por medio de redes neuronales que incorporen mecanismos atencionales, como en los modelos basados en arquitectura *Transformer*.

La tendencia dominante actual al abordar problemas de Procesamiento del Lenguaje Natural (NLP) es la de utilizar modelos de Aprendizaje Profundo y, en particular, aquellos que utilizan mecanismos atencionales. De hecho, se puede observar que en las primeras posiciones de los *rankings*[1] de resultados aparecen sistemas basados en *Transformers*. Estos últimos están consiguiendo resultados que representan o han representado el estado del arte en problemas como la traducción automática [32], la búsqueda de respuestas [53], realización de inferencias basadas en el sentido común [70] u otras tareas que requieren de la comprensión del lenguaje natural [39].

1.1. Propuesta y objetivos

El objetivo de este trabajo es el de evaluar diferentes modelos neuronales con arquitectura *Transformer* para valorar cómo de bien rinden en las tareas de Validación de Respuestas. Nuestra hipótesis inicial de trabajo presume que estos modelos (convenientemente preentrenados y ajustados), podrían ofrecer

1 https://gluebenchmark.com/leaderboard

rendimientos superiores a las técnicas probadas durante los *Answer Validation Exercise* (AVE) cuando son evaluados sobre los conjuntos de prueba propuestos en las diferentes ediciones de ese mismo ejercicio.

En el proceso de verificación de esta hipótesis se utilizarán versiones básicas del modelo BERT, así como modelos derivados, que aunque preentrenados sobre conjuntos de datos con distribuciones diferentes a las seguidas en los AVE, han sido convenientemente ajustados (*fine-tuned*) para realizar labores de inferencia textual. Esto nos permitirá realizar una evaluación sobre los conjuntos de prueba propuestos en los AVE y comparar los resultados con los obtenidos durante la celebración de aquellos ejercicios.

1.2. Estructura del documento

A lo largo de los siete capítulos de este documento se muestran las aproximaciones más populares utilizadas en la Validación de Respuestas y se presentan los experimentos realizados y sus resultados, así como los modelos y conjuntos de datos empleados.

Más concretamente, en el capítulo 2 se introducen las características generales tanto de las técnicas que tradicionalmente se han usado para resolver el problema de la Validación de Respuestas, como de las aproximaciones más actuales.

En el capítulo 3 se presentan los conjuntos de datos del AVE con los que se va a realizar la evaluación y se justifica el uso de las métricas que se van a aplicar en este proceso.

En el capítulo 4 se establecen qué modelos se emplean en los experimentos y las colecciones de datos utilizadas durante su entrenamiento o ajuste. También se detallan las estrategias y decisiones tomadas cuando, tras vincular modelos y colecciones, se procede a la evaluación.

En el capítulo 5 se presentan los resultados obtenidos en los distintos experimentos y se reflexiona sobre ellos.

En el capítulo 6 se hace una recapitulación de los resultados de las evaluaciones y se realiza un análisis de los mismos.

Finalmente, en el capítulo 7, se resume y discuten los resultados básicos obtenidos en este trabajo y se bosquejan algunas líneas de trabajo futuras.

Capítulo 2

Trabajos relacionados

La Validación de Respuestas surgió a partir de la necesidad de mejorar la exactitud y corrección de los resultados ofrecidos por los sistemas de Búsqueda de Respuestas. En este capítulo se describe la relación que, desde el principio, se estableció entre las técnicas de Validación de Repuestas y el problema de la Implicación Textual. Asimismo se presentan los ejercicios que se han celebrado para promover la mejora de los subsistemas de Validación de Respuestas y se introducen las técnicas principales que tradicionalmente se han aplicado, bien para implementarlos, bien para complementar su función.

2.1. Validación de Respuestas e Implicación Textual

En la Validación de Respuestas se valora la corrección de las soluciones generadas por un sistema de Búsqueda de Respuestas. Se trata de identificar si una respuesta extraída de un documento es una respuesta válida a una pregunta dada, de acuerdo con los contenidos del documento. La validación automática de respuestas puede aportar beneficios o resultar útil en los siguientes casos [2]:

- Mejora del rendimiento de los sistemas de Búsqueda de Respuestas.
- Asistencia a los usuarios en la valoración de la validez de los resultados

producidos por sistemas de Búsqueda de Respuestas.

- Mejora del grado de confianza en las respuestas candidatas que generan los sistemas de Búsqueda de Repuestas.
- Asistencia en la elección de un resultado cuando se reciben múltiples respuestas candidatas que han sido producidas concurrentemente por varios sistemas de Búsqueda de Respuestas.
- Validación de respuestas aportadas por humanos.

La manera fundamental con la que los diferentes sistemas han tratado de abordar el problema de la Validación de Respuestas es a través de la implicación textual. Se dice que un texto T implica o vincula una hipótesis H si el significado de H puede ser inferido del significado de T [20]. El objetivo del Reconocimiento de la Implicación Textual es el de determinar si la semántica o significado de un texto puede deducirse de la semántica o significado de otro texto [1]. De esta forma cuando un sistema de Búsqueda de Respuestas devuelve una respuesta y el extracto de texto de donde proviene la misma -su texto de soporte o *snippet*-, se construye una hipótesis en forma afirmativa a partir de la pregunta y la respuesta. Si el extracto de texto implica semánticamente a la hipótesis generada, entonces, la respuesta se considera correcta [3][24].

Se han utilizado numerosas técnicas diferentes para abordar el problema de la Implicación Textual. La comunidad dedicada al Procesamiento del Lenguaje Natural ha dedicado muchos esfuerzos en conseguir avances en este ámbito y, a lo largo del tiempo, ha ido proponiendo aproximaciones que trabajan a nivel léxico, sintáctico y semántico. Desde hace unos años, la tendencia dominante para tratar el problema de la Implicación Textual (o Inferencia del Lenguaje Natural) se dirige hacia el uso de redes neuronales profundas que, para trabajar, transforman frases en vectores codificados [36]. El Aprendizaje Profundo es un tipo de Aprendizaje Automático que logra gran rendimiento y flexibilidad aprendiendo a representar el lenguaje como una jerarquía anidada de conceptos y representaciones.

2.2. Evaluación de sistemas de Validación de Respuestas

En los años 2006, 2007 y 2008, y bajo el nombre de *Answer Validation Exercise* (AVE), se realizaron una serie de evaluaciones de sistemas que identificaban si una respuesta extraída de un documento era una respuesta válida a una pregunta dada [2][6][11]. Las tres ediciones del AVE formaron parte de la tarea sobre Búsqueda de Respuestas (*Question Answering*) de la iniciativa *Cross-Language Evaluation Forum* (CLEF) [19] y tenían como finalidad promocionar el desarrollo y evaluación de subsistemas destinados a validar la corrección de los resultados devueltos por sistemas de Búsqueda de Respuestas.

En la edición del AVE de 2006 se planteó la tarea de la Validación de Respuestas como un problema de Implicación Textual [2]. Los sistemas evaluados aceptaban textos e hipótesis como entrada y devolvían un valor booleano como salida: 'YES', si la hipótesis puede deducirse del texto, y 'NO', en caso contrario. En esta edición, entonces, los participantes no necesitaron construir las hipótesis a partir de las preguntas y respuestas; estas ya estaban incluidas en los conjuntos de desarrollo SPARTE/ENGARTE (para castellano e inglés, respectivamente) y en el conjunto de prueba. Para generar las hipótesis del conjunto de prueba se recurrió al uso de patrones, creando uno para cada una de las preguntas e instanciándolo para todas las respuestas disponibles en cada pregunta [2][3].

En el ejercicio AVE de 2007 los sistemas podían devolver a la salida los valores: 'SELECTED', 'VALIDATED' o 'REJECTED', teniendo en cuenta que, para cada pregunta, debían seleccionar exactamente una respuesta de entre las candidatas válidas. Este cambio, con respecto a 2006, permitía obtener evidencias sobre la ganancia en rendimiento que un subsistema de validación puede aportar en la Búsqueda de Respuestas. Se dejó, además, abierto el problema de la Generación Automática de Hipótesis para aquellos sistemas basados en Implicación Textual, es decir, los sistemas recibían ahora tripletas de la forma (pregunta, respuesta, texto de soporte) en lugar de un par (hipótesis,

texto) y se veían obligados a generar las hipótesis, haciendo que la evaluación adquiriese un carácter más realista [6].

En la última edición del AVE (2008) se siguió una estrategia de evaluación similar a la de 2007 pero la complejidad de los conjuntos de datos aumentó con el objetivo de tratar de valorar la habilidad de los sistemas de validación para enfrentarse a situaciones en las que todas las respuestas candidatas fueran incorrectas. En estos casos, los sistemas participantes debían etiquetar todas ellas como 'REJECTED', abriendo la posibilidad de que solicitasen al sistema de Búsqueda de Respuestas nuevos resultados para tratar de obtener la respuesta correcta [11].

En ninguna de las ediciones de los AVE los sistemas participantes necesitaron contemplar la posibilidad de que las hipótesis pudiesen tener una relación neutral o desconocida con los extractos de texto ya que los pares o tripletas etiquetados, en los conjuntos de prueba, como *neutral* fueron ignorados en las evaluaciones. La composición de los conjuntos de entrenamiento y prueba para todas las ediciones de los AVE se describe con detalle en los apartados 3.1 y 4.2 de este trabajo; No obstante, incluyen una proporción no balanceada de respuestas correctas e incorrectas con el propósito de procurar una evaluación más realista, con salidas más cercanas a las de un sistema de Búsqueda de Respuestas auténtico.

Todos los grupos de investigación participantes en los AVE 2006 y 2007 utilizaron implementaciones basadas en la Implicación Textual [2][6], no así en 2008 [11]. Es una aproximación muy utilizada que suele ir acompañada de técnicas de procesamiento léxico, lematización y etiquetado PoS, análisis sintáctico y, en menor medida, análisis semántico. Para tomar la decisión de la validación se suele aplicar Aprendizaje Automático (SVMs o árboles de decisión) utilizando la similaridad léxica como característica principal, seguido de la similaridad sintáctica y, por último, de características semánticas [6][11][13][16]. También podrían usarse atributos que caractericen el tipo de pregunta que se formula y su formato [8].

El uso de más recursos en el proceso de validación no implica mejor rendimiento, de hecho, los sistemas que empleaban análisis semántico no alcanzaron los mejores resultados en las lenguas para los que están diseñados con lo que su utilización decreció [11], en contraposición al uso de Entidades Nombradas. No obstante, el uso de características semánticas para el Reconocimiento de Implicación Textual siguió vigente tras los AVE. Así, en [24] se expresan las propiedades semánticas de los términos en el *Lenguaje de Interconexión Universal* (UNL).

Terminadas las tres ediciones del AVE, en los años 2009 y 2010, se desarrollaron las evaluaciones de la tarea ResPubliQA – CLEF que valoraba el desempeño de sistemas de Búsqueda de Respuestas cuando trabajaban sobre textos legislativos europeos [20]. En ellas fue posible analizar las sinergias que se producen entre las técnicas de Validación de Respuestas y los sistemas de Búsqueda de Respuestas, al mejorar las primeras la confianza atribuida a los distintos resultados candidatos generados y facilitar, de esta forma, la selección de la respuesta correcta. Posteriormente, durante los años 2011 a 2013, se organizó una nueva campaña de evaluación de sistemas de Búsqueda de Respuestas en CLEF llamada *Question Answering for Machine Reading Evaluation* (QA4MRE). En esta tarea también se trató de promover la inclusión de módulos de validación en sistemas de Búsqueda de Respuestas pero, a pesar de ello, los participantes apenas hicieron uso de la parte de validación.

2.3. Métodos de validación

Como ya hemos mencionado, la forma más extendida de acometer la Validación de Respuestas es por medio de la Implicación Textual. En los siguientes apartados se introducen las técnicas principales empleadas en los subsistemas de Validación de Respuestas. Casi todas ellas implementan alguna forma de reconocer la implicación textual entre dos textos.

2.3.1. Solapamiento léxico

Esta técnica puede aplicarse en diferentes lenguas, tiene un coste computacional bajo y es, por ello, rápida. Es una forma de estimar la implicación textual entre las hipótesis construidas y los extractos de texto devueltos por los sistemas de Búsqueda de Respuestas. Podría implementarse cuantificando una serie de atributos de los textos que previamente han pasado por un etiquetador PoS para obtener los lemas. Estos atributos se usarían como entradas en clasificadores de aprendizaje automático SVM y kNN que habrían sido previamente entrenados con datos (como los proporcionados por los organizadores de AVE) para las distintas lenguas [1]. Veamos algunos de los atributos que podrían cuantificarse:

- N-gramas: número de unigramas coincidentes (independientemente de la posición que ocupen) entre la hipótesis y el texto de soporte, sin entrar a valorar si dos palabras son sinónimas o pueden representar conceptos similares.
- Secuencias de palabras comunes no consecutivas: con esta característica se intenta, a diferencia del atributo anterior, tener en cuenta el orden de las palabras y la estructura a nivel de frase. Una secuencia común más larga indica mayor similaridad entre dos textos.
- Skip-grams: número de palabras coincidentes en ambos textos en el mismo orden en el que aparecen en sus frases, permitiendo huecos arbitrarios entre ellas.
- Números coincidentes: se trata de identificar y contar los números que aparecen tanto en las hipótesis como en los textos de soporte que se les pasa a los sistemas de Validación de Respuestas.
- Fragmentos clave coincidentes: se identifican los fragmentos clave de preguntas y textos de soporte usando módulos CRF y basados en reglas y se asocia un peso según el nivel de coincidencia [18][22].
- Otras características incluyen: número de palabras y caracteres, diferencia en el número de palabras y caracteres (para reducir sesgos en

cuando a la longitud), razón en la longitud de las palabras y caracteres (para reducir sesgos y mejorar la exactitud), la distancia Levenshtein de edición (el número mínimo de operaciones de edición necesarias para modificar una cadena y conseguir la otra), subcadena común más larga, coeficiente de Dice (similaridad de dos frases según el número de bigramas comunes), etc. [30].

También es posible incluir ciertos métodos que extienden y mejoran los anteriores. Así, por ejemplo, se podrían considerar solo "palabras de contenidos" para calcular tanto n-gramas como las secuencias no consecutivas; Estas secuencias no consecutivas también podrían cuantificarse utilizando etiquetas PoS; Se podrían realizar transformaciones sintácticas sobre las hipótesis para considerar tanto su forma activa como pasiva; Se puede utilizar patrones léxicos para tratar textos de soporte que contengan frases yuxtapuestas; Finalmente, también se podrían incluir características que impongan restricciones sobre la respuesta a causa del tipo de pregunta que se formula [8][18].

Podría comprobarse también la similaridad sintáctica entre dos frases (en este caso cada par formado por la pregunta y una de las respuestas candidatas) utilizando un método de correspondencia difusa entre ellas como el utilizado en [31]. Se trataría de encontrar correspondencias parciales entre secuencias (tras lematizar y eliminar stopwords) empleando el coeficiente de similaridad de Jaccard.

2.3.2. Uso de entidades nombradas

Reconocer las Entidades Nombradas de los textos e hipótesis que se le pasan al sistema de Validación de Respuestas proporciona muy buenos resultados en la tarea de valorar la implicación textual. Es un técnica que consigue una exactitud que ronda el 70% y que es similar al solapamiento léxico [5]. Implica identificar las expresiones numéricas, los nombres propios y las expresiones temporales tanto de la hipótesis como del texto de soporte de cada resultado y así determinar las relaciones de vinculación entre las Entidades Nombradas de

una y otra. La decisión final con el que se aplica el modelo aprendido que resuelve la existencia o no de Implicación Textual puede, de nuevo, recaer sobre un clasificador SVM [4][5].

Solo si este clasificador diagnostica que puede existir vinculación, se aplica un segundo clasificador que es entrenado teniendo en cuenta atributos adicionales [4][13] que cuantifican el grado de:

- Solapamiento léxico: se realiza un etiquetado léxico y se obtienen con *Freeling* los lemas de las hipótesis y de los textos de soporte. Se calculan los porcentajes de palabras, unigramas, bigramas y trigramas coincidentes.
- Solapamiento sintáctico: se obtienen árboles de dependencias del texto de soporte y de la hipótesis y se buscan ramas coincidentes. A mayor coincidencia entre los árboles sintácticos, mayor similitud semántica se asume que existe entre ellas.

Otros modelos dependen únicamente de la información recabada a través de las Entidades Nombradas con lo que necesitan identificarlas de manera robusta [7]. Otros, sin embargo, añaden comparaciones detalladas a nivel sintáctico entre hipótesis y textos de soporte, incluyendo relaciones sujeto-verbo, sujeto-sujeto, objeto-verbo, dependencias preposicionales, análisis de determinantes y resolución de anáforas, entre otras, incluyendo o no el uso de *WordNet* en ellas [22].

2.3.3. Análisis semántico

Esta técnica trata de mejorar el reconocimiento de la implicación textual entre hipótesis y textos de soporte cuantificando la coocurrencia entre sus términos teniendo en cuenta que una palabra o concepto podría ser expresada con un sinónimo o término similar. El análisis semántico de los textos e hipótesis puede realizarse haciendo uso de recursos como *WordNet* (que permite la búsqueda de términos similares relacionados) [11][13][18][22] o construyendo espacios

vectoriales que reflejen en una "matriz conceptual" las relaciones semánticas de similaridad entre todos los términos presentes en un corpus [1]. Con este método cada par hipótesis-texto de soporte recibirá una puntuación de similaridad.

Como ya dijimos, los sistemas que emplean análisis semántico no alcanzan los mejores resultados en las lenguas para los que están diseñados y su utilización, de hecho, ha estado decreciendo [11]. A pesar de ello, han aparecido iniciativas posteriores al AVE [24] que emplean UNL (*Lenguaje de Interconexión Universal*) para expresar la información o el conocimiento en la forma de una red semántica con hipernodos (conceptos) y enlaces (relaciones entre conceptos). Con ello, el sistema puede calcular, siguiendo un conjunto de reglas, una "puntuación de similaridad" entre las hipótesis y los texto de soporte a los resultados del sistema de Búsqueda de Respuestas.

También pueden hacerse comprobaciones de similaridad semántica haciendo uso de soluciones basadas en Aprendizaje Profundo. En [31] se presenta un sistema LSI (*Latent Semantic Indexing*) no supervisado que aprende a plasmar las características de fragmentos de texto de longitud variable en representaciones vectoriales de longitud fija. En este modelo, la concatenación o media de las representaciones, junto a un contexto de tres palabras, se usa para predecir una cuarta palabra. El modelo se entrena con un algoritmo de gradiente estocástico y propagación hacia atrás utilizando páginas de la *Wikipedia*, lo que ayuda a aprender sobre la similaridad semántica (similaridad coseno) entre los pares pregunta-respuestas candidatas. El sistema se complementa con otros módulos que permiten eliminar las frases o respuestas candidatas que no casan con el tipo de pregunta que se ha formulado (hay que clasificar entonces la pregunta) o que contienen declaraciones manifiestamente falsas (es necesario recopilar un conjunto de este tipo de respuestas no plausibles).

2.3.4. Representación lógica

Algunos sistemas de Validación de Respuestas usan inferencia lógica para trabajar con las relaciones semánticas entre términos y así resolver el problema

de implicación textual entre hipótesis y textos de soporte. Son sistemas que pueden obtener muy buenos resultados y que pueden adaptarse para aplicarse en tiempo real [2][24]. A veces pueden utilizar Aprendizaje Automático para asignar puntuaciones en el proceso de validación usando características tanto basadas en representaciones lógicas como en otras más superficiales (como las de tipo sintáctico) [9][12][24]. También puede ocurrir que la hipótesis generada a partir de la pregunta formulada por el usuario y el resultado devuelto por el sistema de Búsqueda de Respuestas se exprese directamente en forma lógica (en lugar de hacerlo de forma textual). Esta representación puede realizarse usando lógica de primer orden y la validación de la respuesta puede resolverse con un demostrador automático de teoremas [6][23]. Para conseguir que estos sistemas no sean computacionalmente costosos de aplicar, es importante que el demostrador automático solo use un análisis de la pregunta y otros de los textos que dan soporte a los resultados devueltos por el sistema de Búsqueda de Respuestas [24]. Se trataría de comprobar si estos textos contienen una respuesta correcta.

Los sistemas basados en razonamiento lógico necesitan de una base de conocimiento amplia para conseguir resultados realmente buenos, pero si han de operar en un entorno de dominio abierto, esta aproximación no es realista. No obstante, podrían complementarse a través de técnicas de Aprendizaje Automático con otras fuentes de conocimiento como los sistemas de razonamiento basados en casos (CBR) que permitirían ir incrementando la calidad de los resultados que valida el sistema a partir de conocimiento basado en la experiencia. Para ello, necesitarían disponer de retroalimentación por parte de los usuarios, extendiendo así el caso base sobre el que operan [23].

2.3.5. Validación basada en recursos web

Una aproximación práctica a la Validación de Respuestas y diferente del reconocimiento de implicación textual, consiste en hacer uso de recursos online como la *Wikipedia* o *Google*. Su funcionamiento se basa en el hecho de que la

frecuencia de las repuestas correctas en la Web son mayores que la de las respuestas incorrectas. La consulta original se envía a un buscador web como Google para obtener un listado de documentos relevantes [15]. Las respuestas candidatas se comparan con los n primeros resultados obtenidos. De esta manera, si una respuesta candidata puede encontrarse en al menos el 20% de estos resultados, se considerará una respuesta correcta. En [16], por su parte, se busca en la Wikipedia aquellas páginas cuyo título contengan la respuesta candidata. Si alguna de estas páginas contiene además el tipo de respuesta esperado entonces se dará por válida la respuesta.

Otro tipo de heurística para la Validación de Respuestas que también utiliza la Web, apuesta por la generación de una serie de patrones de respuesta para la pregunta formulada por el usuario y su comparación con respuestas candidatas extraídas de la Web. Estos patrones se producen tras identificar el tipo de pregunta y tras haberla reformulado varias veces sustituyendo sus términos clave por sinónimos o añadiendo palabras similares (*query expansion*) haciendo uso de recursos como *WordNet*. Los patrones de respuestas se representan entonces como árboles sintácticos y se comparan con las frases candidatas previamente obtenidas de la Web. Según la similitud observada, a las frases candidatas se les asigna una puntuación. Si esta puntuación supera un determinado umbral, podrán considerarse respuestas correctas [15].

2.3.6. Aprendizaje automático

Ya se ha mencionado en apartados anteriores cómo pueden emplearse técnicas de Aprendizaje Automático para complementar o implementar distintas formas de reconocer la implicación textual entre dos fragmentos de texto. Estas técnicas tratan de calcular la similaridad entre ellos basándose en características más o menos elaboradas.

Tradicionalmente, al aplicar Aprendizaje Automático al problema de la Validación de Respuestas, se ha optado por extraer distintas características de los textos a relacionar (hipótesis y texto de soporte) para que, a partir de ellas,

un clasificador automático pueda determinar si existe implicación textual.

Las características extraídas suelen provenir de la comparación léxica de los textos (número de palabras compartidas, stopwords, n-gramas, entidades nombradas, cantidades, nombres o verbos) pero también de sus estructuras sintácticas, representaciones semánticas o incluso representaciones lógicas. Para ello, los textos son preprocesados por analizadores léxicos y sintácticos y se resuelven correferencias. De los resultados se extraen características, incluyendo dependencias, paráfrasis y características semánticas. Tras la extracción y selección de características, se elige un clasificador (árboles de decisión, SVMs -los más populares-, entropía máxima, etc.), se alimenta el clasificador con las características extraídas (con SVM todas ellas pueden adoptar la forma de un vector) y el clasificador permite finalmente decidir si existe o no vinculación.

En un conjunto de datos, las relaciones que se establecen entre las hipótesis y los textos de soporte son complejas, no es sencillo establecer las características que permitan discriminar si existe implicación textual entre ellas o no, con lo que el rendimiento de los clasificadores no es ideal [28].

2.3.7. Aprendizaje profundo

Como decíamos en apartados anteriores, la aplicación de técnicas de Aprendizaje Automático ha estado presente desde el inicio en el problema del reconocimiento de la implicación textual entre textos. Sin embargo, conforme se van desarrollando los métodos propios del Aprendizaje Profundo, los investigadores se están valiendo de Redes Neuronales Recurrentes (RNN) o Redes Neuronales Convolucionales (CNN) para realizar aproximaciones holísticas a la Validación de Respuestas.

Así, en [33], abordan de forma conjunta el problema de puntuar las posibles respuestas candidatas y el de establecer si existen respuestas correctas entre ellas. Utilizan para ello redes neuronales profundas que optimizan una función objetivo que penaliza los falsos positivos, los falsos negativos y el otorgar

puntuaciones más altas a las respuestas incorrectas que a las correctas. El sistema funciona en tres pasos: (1) se procesan y codifican las preguntas y las respuestas candidatas, obteniendo sus imágenes en dos espacios vectoriales diferentes; (2) para cada par pregunta-respuestas candidata se concatenan sus representaciones y se obtiene una puntuación (a través de una red neuronal) para establecer el grado de correspondencia entre ambas; (3) se obtiene el par pregunta-respuesta candidata con la máxima puntuación.

En [34], por su parte, se pone en práctica y evalúa un modelo HGRNT (*Hierarchical Gated Recurrent Neural Tensor*) para capturar tanto la información contextual como las relaciones "profundas" entre las respuestas candidatas y la pregunta con la intención de predecir la respuesta correcta. Incluir información contextual incrementa la precisión y exhaustividad del sistema en su conjunto. El modelo consiste en cuatro capas: (1) se codifican por separado las respuestas candidatas y la pregunta en vectores de longitud fija; (2) los vectores de las respuestas pasan por redes BiGRNN (*Bidirectional Gated Recurrent Neural Network*) junto con el párrafo que las contienen para así permitir que la información contextual fluya; (3) una red neuronal determina la similitud entre las respuestas candidatas en su contexto y la pregunta; (4) con regresión logística se calcula una puntuación a la confianza otorgada sobre cada respuesta. Se establece un umbral para decidir si la respuesta con la mayor puntuación se elige o no como respuesta final.

Las técnicas de Aprendizaje Profundo también se han combinado con prácticas más tradicionales. Así ocurre en [35], donde encuentran limitaciones en las redes CNN para capturar la similitud entre frases e investigan la incorporación y uso de características lingüísticas (computables a partir de los grafos de dependencia) en esas redes convolucionales para obtener mejoras en el rendimiento.

Por otra parte, dos conceptos que han adquirido gran relevancia en prácticamente todas las áreas del Procesamiento del Lenguaje Natural son las redes LSTM (*Long Short Term Memory*) y los mecanismos atencionales. LSTM

es un tipo de Red Neuronal Recurrente (RNN) bien conocida por sus cualidades para manejar secuencias más largas y detectar las dependencias que se producen en ellas. Los mecanismos atencionales, por su parte, permiten a la red neuronal aprender qué partes de la secuencia son relevantes para el problema. Estos métodos ofrecen a las RNNs acceso a todos los estados de la secuencia de entrada, asignándoles un peso o puntuación al utilizarlos.

Podemos clasificar, entonces, estos modelos más novedosos en dos grandes categorías: (1) Los modelos basados en construir una representación de las frases codificándolas a partir de codificadores secuenciales como RNN o LSTM y (2) los modelos que directamente codifican la relación existente entre hipótesis y texto de soporte utilizando diferentes mecanismos atencionales [38].

Ambos conceptos se están aplicando en la implementación de sistemas de Búsqueda de Respuestas [46], pero apenas hay evidencia de que se estén usando en la Validación de Respuestas. A este respecto, en [47] experimentan con tres arquitecturas diferentes: (1) Se emplea como verificador un modelo secuencial con un descodificador multicapa *Transformer* (reemplazando a LSTM, ya que permite mayor grado de paralelismo en la computación y menor tiempo de entrenamiento) que toma las entradas como secuencias largas. El modelo *Transformer* es una adaptación del OpenAI GPT (*Generative Pre-trained Transformer*) que se ajusta (afina) tras el entrenamiento para esta tarea objetivo, consiguiendo manejar las dependencias de largo alcance en la secuencia e incorporar mejor el conocimiento. (2) Se usa como verificador un modelo interactivo que intenta capturar y codificar las interacciones entre dos frases (y también las correlaciones dentro de cada frase) para así reconocer su grado de vinculación. Se utiliza LSTM bidireccional (BiLSTM) y mecanismos atencionales para producir estas representaciones. (3) Se experimenta con un modelo híbrido entre el modelo secuencial y el interactivo concatenando los vectores de salida producidos por ambos modelos.

2.4. Recapitulación

Un subsistema de Validación de Respuestas comienza a realizar su trabajo cuando un usuario formula una pregunta a un sistema de Búsqueda de Respuestas y este devuelve una respuesta candidata junto al documento o extracto de texto que da soporte al resultado [24]. Típicamente, el subsistema combinará en una hipótesis la pregunta formulada (expresada en modo afirmativo) con la respuesta obtenida y comprobará si existe vinculación entre esta hipótesis y el extracto de texto [20]. Como ya se ha mencionado, las relaciones que se establecen entre las hipótesis y los textos de soporte son complejas y no es sencillo establecer las características que permitan discriminar si existe implicación textual entre los dos fragmentos de texto.

La complejidad en la investigación en torno a la Validación de Respuestas y, en particular, en el problema de la Implicación Textual ha ido creciendo en los últimos años. Inspirados por el éxito conseguido en otras áreas del Procesamiento del Lenguaje Natural, se han ido sustituyendo las técnicas de Aprendizaje Automático y otras más tradicionales por la aplicación de los métodos propios del Aprendizaje Profundo. Entre estos métodos, los *Transformers* parecen estar en auge al ir reemplazando a RNNs como LSTM.

No se ha podido documentar el empleo del modelo *Transformer* preentrenado BERT (*Bidirectional Encoder Representations from Transformers*) en un subsistema de Validación de Respuestas. En este trabajo pretendemos, en consecuencia, aplicarlo y evaluarlo (junto a otros modelos derivados o similares) en la tarea de Validación de Respuestas para valorar su rendimiento y compararlo con el de las aproximaciones que participaron en las distintas ediciones de los AVE.

Capítulo 3

Marco de evaluación

Nuestra hipótesis inicial de trabajo presume que los modelos *Transformer* (una vez preentrenados y ajustados), podrían conseguir resultados mejores a los ofrecidos por las técnicas que fueron probadas durante los AVE cuando son evaluados sobre los mismos conjuntos de prueba. En este capítulo se presentan los conjuntos de datos del AVE con los que se va a realizar la evaluación y se justifica el uso de las métricas que se van a aplicar en este proceso.

3.1. Colecciones de evaluación

Utilizaremos las colecciones de datos de prueba de las tres ediciones del *Answer Validation Exercise* (AVE) para evaluar nuestros modelos de Validación de Respuestas y comparar sus resultados con los de otras aproximaciones no basadas en Aprendizaje Profundo. Describamos los conjuntos de prueba utilizados en los experimentos:

- **AVE 2006** [2]: En la edición de 2006 del *Answer Validation Exercise* los participantes recibieron conjuntos de prueba conteniendo pares de *texto* - *hipótesis* junto a un valor *VERDADERO/FALSO* que, para inglés y castellano, tenían la composición que muestra la tabla 1.

	Inglés	**Español**
Pares con vinculación (TRUE)	198 (9,5%)	671 (28%)
Pares sin vinculación (FALSE)	1048 (50%)	1615 (68%)
Total de pares	2088	2369

Tabla 1. Composición de los conjuntos de prueba para inglés y castellano en AVE 2006

Ambas colecciones contienen también pares con vinculación desconocida (UNKNOWN) pero se han obviado en la tabla 1 porque no serán tenidos en cuenta en el proceso de evaluación. Durante las evaluaciones de las distintas ediciones de los AVE, los pares con vinculación desconocida fueron ignorados o no contabilizados para el cálculo de las métricas de evaluación. En nuestros experimentos, por tanto, hemos de actuar de igual forma.

- **AVE 2007** [6]: En la edición de 2007 del AVE, los conjuntos de prueba y desarrollo facilitados no contenían pares *texto - hipótesis* sino tripletas del tipo *pregunta – respuesta - texto de soporte*. De esta forma, se obligaba a los participantes a generar las hipótesis a partir de las preguntas y respuestas. Los conjuntos de prueba para inglés y castellano seguían la composición de la tabla 2.

	Inglés	**Español**
Pares con vinculación (VALIDATED)	21 (10,4%)	127 (22,5%)
Pares sin vinculación (REJECTED)	174 (86,1%)	424 (75,2%)
Total de pares	202	564

Tabla 2. Composición de los conjuntos de prueba para inglés y castellano en AVE 2007

Estos conjuntos se utilizarán en nuestras pruebas y también contienen pares con vinculación desconocida. Estos han sido excluidos de la tabla 2 porque

serán ignorados durante la evaluación.

- **AVE 2008** [11]: El formato propuesto para el conjunto de prueba (y el de entrenamiento) en esta edición coincide con la de la edición anterior: están formados por tripletas *pregunta – respuesta - texto de soporte*, con lo que era necesario que los participantes generasen las hipótesis. Los conjuntos de prueba para inglés y castellano tienen la composición mostrada en la tabla 3.

	Inglés	**Español**
Pares con vinculación (VALIDATED)	79 (7,5%)	153 (10%)
Pares sin vinculación (REJECTED)	940 (89,1%)	1354 (88,6%)
Total de pares	1055	1528

Tabla 3. Composición de los conjuntos de prueba para inglés y castellano en AVE 2008

Como en las ediciones anteriores, en AVE 2008, durante las evaluaciones se ignoraron aquellos pares de los conjuntos de prueba etiquetados como *UNKNOWN*. En nuestros experimentos procederemos de igual forma y, por ello, no se han incluido en la tabla 3.

3.2. Métricas de evaluación

En este trabajo evaluamos diferentes modelos de Aprendizaje Profundo basados en arquitectura *Transformer* utilizando los conjuntos de prueba propuestos en las tres ediciones del AVE. Nuestra evaluación, al igual que en los AVE, se basará en la detección de los pares *hipótesis - texto de soporte* (edición de 2006) o tripletas *pregunta – respuesta - texto de soporte* (ediciones de 2007 y 2008) donde realmente haya vinculación [12]. De esta forma, se valorará más positivamente aquellos sistemas que validen una respuesta cuando tengan suficiente evidencia de su corrección.

Además, como decíamos en el apartado 2.2, un subsistema de Validación de Respuestas, en un entorno real de explotación, no va a recibir necesariamente proporciones equilibradas entre pares o tripletas en los que existe vinculación (respuestas correctas) y pares en los que no (respuestas incorrectas) y debe evitarse el uso de métricas (como la exactitud) cuyo resultado depende de esta distribución [26][30].

Por tanto, el lugar de usar la exactitud global como la medida de evaluación, se utilizará la precisión, exhaustividad y medida F_1 (media armónica) solo sobre las respuestas correctas (donde se haya detectado vinculación entre hipótesis y texto). Las fórmulas usadas son:

$$precisión = \frac{|respuestas\ positivas\ verdaderas|}{|respuestas\ positivas\ verdaderas| + |respuestas\ positivas\ falsas|}$$

$$exhaustividad = \frac{|respuestas\ positivas\ verdaderas|}{|respuestas\ positivas\ verdaderas| + |respuestas\ negativas\ falsas|}$$

$$F_1 = \frac{2 * exhaustividad * precisión}{exhaustividad + precisión}$$

Con estas medidas es posible comparar el rendimiento entre diferentes aproximaciones, pero si pretendemos establecer una comparación entre sistemas que se han evaluado con colecciones de datos distintas, debemos tener presente algún sistema básico de referencia, como puede ser un sistema que siempre acepte todas las respuestas (dé por válidas el 100% de las respuestas).

Estos sistemas básicos devuelven mejores resultados en nuestras métricas de evaluación cuando la proporción de respuestas válidas es mayor. Dicho con otras palabras, estas métricas son muy dependientes de la distribución de las respuestas correctas en la colección de prueba.

El valor F_1, como hemos visto, es una medida que combina tanto la precisión como la exhaustividad. Es una media armónica entre ambas, que es una forma de calcular la media más adecuada que la media aritmética cuando se trabaja

con razones (como la precisión y la exhaustividad) ya que pondera las dos razones de forma balanceada, obligando a que tanto precisión como exhaustividad crezcan si queremos que el valor F_1 crezca también.

A pesar de sus inconvenientes, se considera que el valor F_1 es la medida más informativa para evaluar los módulos de Validación de Respuestas puesto que penaliza en mayor medida la validación incorrecta de resultados.

Métricas como la precisión, exhaustividad o F_1 son apropiadas para evaluar y comparar los rendimientos de subsistemas de Validación de Respuestas pero no son suficientes para valorar en qué medida mejoran el rendimiento de los sistemas de Búsqueda de Respuestas en los que puedan incorporarse.

Capítulo 4

Propuestas

En este capítulo se presentan los modelos empleados en los experimentos y las colecciones de datos utilizadas para su ajuste. También se detallan las estrategias y decisiones aplicadas que nos permiten obtener modelos habilitados para realizar tareas de Validación de Respuestas bajo las mismas condiciones que se siguieron en los AVE. Solo así podremos comparar los resultados que obtengamos con los conseguidos durante la celebración de aquellos ejercicios.

4.1. Modelos empleados

Los modelos de Aprendizaje Profundo requieren de una gran cantidad de datos etiquetados para poder ser entrenados. Como la elaboración de estos conjuntos de datos etiquetados resulta enormemente costosa, para construir modelos del lenguaje que ofrezcan resultados precisos se suele recurrir al preentrenamiento de sus redes neuronales. En este preentrenamiento el modelo procesa una vasta cantidad de texto sin etiquetar, requiriéndole que complete una frase o rellene sus huecos para así aprender la estructura del lenguaje. Solo después podemos adaptar este aprendizaje a las distintas tareas de Procesamiento del Lenguaje Natural, aplicándole un entrenamiento más específico (*fine-tuning*).

En nuestro trabajo se han empleado fundamentalmente modelos de tipo BERT

que consisten en redes neuronales *Transformer* que permiten integrar el contexto de una palabra en la generación de su representación vectorial. Estas representaciones de palabras se conocen como *Word Embeddings* y establecen correspondencias entre las palabras (y su contexto) y un espacio vectorial predefinido que puede tener cientos de dimensiones [45]. Nuestros modelos usan redes neuronales prealimentadas (*feedforward*), en lugar de redes recurrentes, lo que permite crear modelos del lenguaje de gran tamaño que ofrecen mejor rendimiento.

Describamos los modelos empleados en los experimentos:

- **BERT** [39][54]: Es un modelo del lenguaje basado en *Transformers,* como ya hemos dicho, y que ha sido preentrenado de forma autosupervisada a partir de un conjunto amplio de textos (*BooksCorpus* -con 800 millones de palabras- y *Wikipedia* en inglés -con 2.500 millones de palabras-). El proceso es autosupervisado porque las etiquetas asociadas a las entradas del conjunto de entrenamiento no vienen establecidas por humanos. Un proceso automático genera, a partir de los textos, tanto las secuencias de entrada como sus etiquetas asociadas. Durante el preentrenamiento se enmascara de forma aleatoria el 15% de las palabras de las entradas (es decir, se sustituyen por el *token* especial *[MASK]*) y el modelo debe predecirlas. También es entrenado para aprender si dos frases dadas aparecen consecutivamente en los textos.

  BERT es considerado un modelo del lenguaje *autoencoder* y durante su preentrenamiento aprende tanto del contexto previo de la palabra enmascarada como del posterior. BERT es ajustado o afinado, en nuestras pruebas, a partir de distintos conjuntos de datos etiquetados (RTE, AVE y SNLI). Para ello, hemos añadido un perceptrón multicapa (MLP) con tres capas de nodos (capa de entrada, capa oculta y capa de salida) cuyos parámetros han de ajustarse para conseguir que realice una tarea específica: en nuestro caso, discernir si existe o no implicación textual entre secuencias de texto. Las entradas al modelo se construyen de la siguiente forma:

  [CLS] + Secuencia de texto de apoyo + [SEP] + Hipótesis

Cuando BERT procesa las entradas, da una representación determinada al token especial *[CLS]* (abreviatura de *classification*). El perceptrón multicapa transforma esa representación que codifica la información de tanto el texto de soporte como de la hipótesis en dos salidas: *entailment* y *contradiction* (etiquetas de las clases a las que pueden pertenecer las entradas). No se consideran resultados neutrales porque en los AVE no se contemplaron durante el proceso de evaluación.

En nuestros experimentos se usan las versiones *base* y *large* de BERT. BERT-base tiene unos 110 millones de parámetros (aproximadamente el mismo tamaño que OpenAI GPT, modelo al que BERT superó en todas las tareas evaluables en GLUE -*General Language Understanding Evaluation*-). BERT-large tiene unos 340 millones de parámetros.

- **BETO**: Es una versión de BERT, de tamaño similar a BERT-base, que ha sido preentrenado desde cero en español con la técnica del enmascaramiento de palabra completa. El corpus utilizado durante el preentrenamiento comprende casi tres mil millones de *tokens* procedentes de la *Wikipedia* en castellano y quince fuentes más.

  En nuestra pruebas, para ajustar BETO en nuestro propio equipo, se ha añadido una última capa que aplica una transformación lineal a las salidas del modelo original. Esta capa ofrece una sola salida (función logit) y, cuando queremos obtener las predicciones del modelo sobre el conjunto de prueba, se aplica sobre ella la función logística (su función inversa) para así obtener probabilidades. De esta forma, el modelo ajustado considerará que no hay vinculación entre el texto de soporte y la hipótesis cuando una probabilidad sea menor a 0.5, y considerará que sí la hay en caso contrario. BETO, una vez ajustado, no devolverá, por tanto, resultados neutrales.

- **ALBert** [52]: Como BERT, es un modelo basado en *Transformers y* preentrenado de forma autosupervisada. En su diseño se adoptan técnicas que permiten reducir el número de parámetros requeridos ofreciendo así un menor consumo de memoria y mayor velocidad de entrenamiento. Durante su

preentrenamiento, al igual que BERT, se enmascara de forma aleatoria el 15% de las palabras de las entradas. El modelo es entrenado para aprender a predecir las palabras enmascaradas y conocer la disposición que guardan segmentos consecutivos de texto con el objetivo de ofrecer mayor coherencia en las predicciones.

En nuestros experimentos se usan las versiones *base*, *large* y *xx-large* de ALBert. ALBert es un modelo mucho más ligero que BERT. La versión *large* tiene 17 millones de parámetros, lo que viene a ser unas 18 veces menos que en *BERT-large*. *ALBert-xxlarge* tiene 223 millones de parámetros y eso supone menos del 70% de los parámetros de *BERT-large*.

- **BART** [51]: Modelo que utiliza una arquitectura estándar de traducción automática basada en *Transformers* que incorpora un codificador bidireccional (similar al de BERT) y un descodificador autoregresivo (similar al de GPT). Para preentrenarlo se corrompe primero el texto del corpus con una función que introduce ruido aleatorio y se le pide, después, al modelo que aprenda a reconstruir el texto original.

  En nuestras pruebas se usa la versión *large* de BART (406 millones de parámetros) y muestra un rendimiento similar al de RoBERTa cuando es ajustado con recursos de entrenamiento comparables.

- **RoBERTa** [50]: Como BERT y ALBert, se trata de un modelo basado en *Transformers* y preentrenado de forma autosupervisada a partir de un amplio conjunto de textos. RoBERTa comparte arquitectura con BERT pero el preentrenamiento se realiza durante más tiempo, con lotes mayores, sobre más datos y sobre secuencias más largas. También durante este proceso se va cambiando el patrón de enmascaramiento que se aplica sobre las entradas. El modelo se entrena para predecir las palabras enmascaradas pero no para discernir si dos frases aparecen en los textos de forma consecutiva.

  En nuestros experimentos se usa la versión *large* de RoBERTa con 355 millones de parámetros.

- **Bertín**: Se trata de un modelo RoBERTa-base que ha sido entrenado desde cero en español. El conjunto de entrenamiento para el modelo base es *mC4*, variante multilingüe del conjunto de datos C4.

  En nuestras pruebas se usa la versión *base* de Bertín para español que utiliza 125 millones de parámetros.

- **ELECTRA** [67]: Con ELECTRA se practica un preentrenamiento diferente sobre BERT, que es el modelo subyacente y sobre el que apenas se realizan cambios en su arquitectura. Se entrenan dos modelos basados en *Transformers*: un generador y un discriminador. La función del modelo generador es la de reemplazar algunos *tokens* de la secuencia de entrada con alternativas plausibles (en lugar de reemplazarlos, como hace BERT, con el *token [MASK]*). La función del modelo discriminador es la de intentar identificar qué *tokens* fueron reemplazados por el modelo generador en la secuencia. Con esta aproximación se consiguen mejorar los resultados de BERT para modelos de tamaño similar.

  En nuestros experimentos se usa la versión *large discriminator* de ELECTRA cuyo rendimiento puede compararse al de RoBERTa y XLNet a pesar de utilizar menos parámetros y usar mucho menos tiempo para el entrenamiento.

- **Electricidad**: Se trata de un modelo ELECTRA-small para la clasificación de secuencias que ha sido entrenado en un amplio corpus en español sin etiquetar con casi tres mil millones de palabras.

  En nuestras pruebas se usa la versión *small discriminator* de Electricidad para español que utiliza unos 17 millones de parámetros.

- **XLNet** [49]: Modelo del lenguaje, basado en *Transformer-XL*, que utiliza un método autoregresivo para aprender sin supervisión a predecir palabras a partir de su contexto. Su arquitectura autoregresiva solo permitiría que este contexto esté formado bien por las palabras anteriores a la que se ha de predecir, o bien por las posteriores, pero no ambas simultáneamente. Para que

XLNet pueda aprender de un contexto bidireccional se realizan permutaciones sobre los *tokens* que conforman el contexto. XLNet es uno de los pocos modelos que no establece un límite para la longitud de las secuencias con las que puede trabajar.

En nuestros experimentos se usa la versión *large cased* de XLNet para inglés que utiliza 340 millones de parámetros.

4.2. Colecciones de datos

Como se mencionaba en el apartado anterior, se requieren grandes cantidades de datos en el preentrenamiento de un modelo de Aprendizaje Profundo para que asimile la estructura de un lenguaje. Conseguido un modelo base, es posible añadirle capas neuronales adicionales y aplicarle un nuevo entrenamiento (*fine-tuning*) para adaptar su aprendizaje a una tarea más específica, como puede ser el reconocimiento de la implicación textual, que es la que necesitamos para nuestros experimentos.

El ajuste de un modelo para las tareas requeridas en este trabajo precisa de conjuntos de entrenamiento amplios que sigan una estructura similar a aquellos utilizados durante los AVE y que nos permitan realizar labores de implicación o inferencia textual. Describamos los conjuntos de entrenamiento utilizados:

- **AVE 2006** [2]: En la edición de 2006 del *Answer Validation Exercise* no hubo conjunto de entrenamiento propio y los participantes pudieron utilizar SPARTE o ENGARTE (para español e inglés respectivamente) porque son apropiados para sistemas de reconocimiento de implicación textual [3]. Estas colecciones contienen las preguntas formuladas, las hipótesis ya elaboradas y los textos de soporte sobre los que han de contrastarse las hipótesis. No hay pares *texto - hipótesis* cuya vinculación sea desconocida.

- **AVE 2007** [6]: Los conjuntos de entrenamiento en AVE 2007 contienen tripletas del tipo *pregunta – respuesta - texto de soporte*, obligando, de esta

forma, a generar las hipótesis a partir de las preguntas y respuestas. Constan, para los idiomas inglés y castellano, de 1121 (130 *validated*, 991 *rejected*) y 1817 tripletas (265 *validated*, 1552 *rejected*) respectivamente y no son, por tanto, lo suficientemente amplios como para entrenar con ellos modelos estadísticos como los utilizados en nuestros experimentos. Optamos, por tanto, por utilizarlos junto a otros conjuntos de entrenamiento similares.

- **AVE 2008** [11]: El formato propuesto para los conjuntos de entrenamiento y prueba en esta edición de 2008 coincide con la del año anterior: se facilitan unas tripletas *pregunta – respuesta - texto de soporte*, con lo que los participantes tenían que generar las hipótesis.

  Los conjuntos de entrenamiento facilitados en AVE 2008 para inglés y castellano constan de 195 (21 *validated*, 174 *rejected*) y 551 tripletas (127 *validated*, 424 *rejected*) respectivamente. Son conjuntos pequeños para entrenar o ajustar modelos estadísticos como los aquí utilizados y han de ser utilizados junto a otros conjuntos de datos de estructura similar como RTE, SNLI, MNLI y otros.

- **RTE-1, RTE-2 y RTE-3**: Provienen de las tres primeras ediciones del *PASCAL Recognizing Textual Entailment Challenge*. Se elaboraron manualmente y recogen pares de textos (de una o dos frases) e hipótesis (de una frase). No es, por tanto, necesario generar las hipótesis.

  RTE-1 (2005) proporciona 567 pares texto-hipótesis en el conjunto de entrenamiento y 800 en el conjunto de prueba. RTE-2 (2006) aporta 800 pares texto-hipótesis en el conjunto de entrenamiento y otros 800 en el conjunto de prueba. RTE-3 (2007) proporciona 800 pares texto-hipótesis en el conjunto de entrenamiento y otros 800 en el conjunto de prueba. A partir de RTE-4 las clases de salida son diferentes y menos apropiadas para nuestro propósito. En nuestros experimentos, como el número de pares facilitados por los conjuntos RTE es pequeño, se opta por mezclar conjuntos de desarrollo y prueba en uno cuando necesitamos ajustar (*fine-tune*) un modelo. Para la evaluación siempre se emplearán los conjuntos de prueba proporcionados en AVE.

- **SNLI** [38]: El conjunto *Stanford Natural Language Inference* (SNLI) es una colección de 570.000 pares clasificados manualmente con las etiquetas *entailment*, *contradiction*, y *neutral*. Al estar formado por un vasto número de frases puede ser utilizado para el entrenamiento de modelos de Aprendizaje Profundo.

  Todas sus frases pertenecen al mismo género (descripciones cortas y sencillas de escenas visuales) con lo que puede producir modelos con problemas de rendimiento al mostrar dificultades con otros estilos de frases.

- **MNLI** [38]: El corpus NLI multigénero (MNLI) se creó para remediar las limitaciones de SNLI. Consiste en 433.000 pares de frases recogidas y etiquetadas de manera similar a SNLI, pero, a diferencia de este, las frases proceden de diez géneros diferentes en inglés escrito y hablado, cubriendo casi toda la complejidad del lenguaje.

  Todos los géneros (transcripciones telefónicas, cartas, guías de viaje, revistas, etc.) están presentes tanto en el conjunto de desarrollo como en el de prueba, pero solo cinco de ellos se incluyen en el conjunto de entrenamiento. Esto se hace para evaluar la capacidad de los modelos para trabajar fuera de los géneros ahí recogidos.

- **HANS** [55]: Es un conjunto pensado para evaluar si un modelo de inferencia del lenguaje natural ha aprendido heurísticas no válidas sobre la sintaxis del lenguaje. El conjunto HANS (*Heuristic Analysis for NLI Systems*) contiene muchos ejemplos donde estas heurísticas fallan; es una colección compleja cuyo etiquetado puede representar, a veces, un reto también para humanos. De hecho, modelos de tipo BERT, que han sido entrenados con MNLI, pueden ofrecer un bajo rendimiento con HANS.

- **FEVER-NLI**: El conjunto *Fact Extraction and VERification* (FEVER-NLI) consiste en 185.445 afirmaciones generadas alterando frases extraídas de la *Wikipedia*. Las afirmaciones se clasifican y etiquetan manualmente como *Supported*, *Refuted* o *NotEnoughInfo*. En el conjunto FEVER original la

vinculación se establecía sobre la *Wikipedia*, pero para facilitar su uso se incluyeron textos de soporte junto a las afirmaciones.

- **ANLI (R1, R2, R3)**: *Adversarial NLI* (ANLI) es un conjunto que consiste en tres series de datos que ofrecen mayor dificultad y complejidad de forma progresiva. Se trata, por diseño, de un conjunto de datos más complicado que otros conjuntos anteriores (SNLI, MNLI, ...). Sus textos son también más largos que los que contiene SNLI.

- **SICK_ES** [71]: El conjunto SICK (*Sentences Involving Compositional Knowldedge*) comprende unos 10.000 pares de frases en inglés que incluyen muchos ejemplos de fenómenos léxicos, sintácticos y semánticos que los modelos del lenguaje han de tener en cuenta. Las frases son usualmente cortas (de longitud menor a 100 caracteres) porque se generaron a partir de leyendas asociadas con imágenes o vídeos; SICK_ES es una traducción al castellano de las mismas. Se clasificó manualmente cada par de frases para dos tareas semánticas: la relación del par de frases en cuanto a sus significados y la implicación textual entre las dos frases del par (usando, para este caso, las etiquetas *entailment*, *contradiction* y *neutral*).

- **XNLI_ES** [40]: El corpus *Cross-lingual Natural Language Inference* (XNLI) es la extensión del conjunto MNLI a 15 lenguas diferentes. Se creó traduciendo manualmente los conjuntos de validación y prueba de MNLI a esas quince lenguas. El conjunto de entrenamiento, por su parte, se construyó con traductores automáticos. Incluyen frases largas, superando, a veces, los 500 caracteres. XNLI contiene 122.000 pares de texto en su conjunto de entrenamiento, 2490 en el de validación y 5010 en el de prueba; todos ellos etiquetados manualmente siguiendo el formato habitual de *entailment*, *contradiction* y *neutral*.

4.3. Estrategias utilizadas

En este trabajo se evaluarán, en el problema de la Validación de Respuestas, distintas versiones de los modelos *Transformer* propuestos en el apartado 4.1. No todas ellas tienen el mismo origen ni precisan del mismo tratamiento para acometer la evaluación. Así, por un lado tenemos los modelos BERT-base y BETO que han sido ajustados por nosotros mismos sobre colecciones relativamente pequeñas (por limitaciones técnicas) para realizar tareas de inferencia textual y, por otro, tenemos los modelos procedentes de la plataforma *Hugging Face* (incluyendo un segundo BERT-base) que son utilizados directamente porque fueron ya ajustados sobre grandes colecciones de datos (principalmente SNLI y MNLI).

Como ya se ha expuesto, los modelos que han sido ajustados en nuestro equipo no devuelven resultados neutrales debido a que la última capa neuronal añadida es una capa lineal que, en ningún caso, llega a tener tres nodos, con lo que sólo quedan representadas la *vinculación* y la *no vinculación* entre los textos. Sin embargo, el resto de modelos empleados sí fueron ajustados por sus promotores para contemplar la posibilidad de que no se pueda determinar la implicación entre la hipótesis y el texto de soporte. Han sido entrenados, entonces, para devolver tres posibles respuestas: *entailment*, *contradiction* y *neutral*.

En estos casos, para que la evaluación del modelo sea semejante al procedimiento utilizado durante los AVE, se hace necesario decidir cómo serán considerados los resultados neutrales o desconocidos cuando se obtengan. Se ofrecerán, por este motivo, dos tipos de evaluaciones para cada modelo: en una asimilando los resultados neutrales como respuestas correctas (es decir, contabilizándolos como pares o tripletas donde hay implicación textual) y en otra, considerando los resultados neutrales como respuestas incorrectas (en las que no hay implicación textual entre texto de soporte e hipótesis).

En este sentido, estaremos aplicando una suerte de *transfer learning*, en tanto en cuanto, para adaptarnos a la evaluación de los AVE, emplearemos modelos que han sido entrenados para una tarea distinta (aunque muy similar).

Asimismo, los conjuntos utilizados en su proceso de ajuste siguen una distribución (en términos de género o temática de las fuentes, estilo y longitud de sus textos) diferente a la de los conjuntos de evaluación creados para las distintas ediciones de los AVE. No obstante, son conjuntos que fueron diseñados para construir sistemas de clasificación de pares de oraciones (o clasificación de secuencias) y, por tanto, su uso es adecuado para nuestra labor.

Esta aproximación también es apropiada por la razón de que no se dispone en los AVE de conjuntos de entrenamiento lo suficientemente grandes: cuando se ha experimentado con ajustar un modelo exclusivamente con uno de los conjuntos de desarrollo de los AVE hemos obtenido sistemas que no devuelven ningún verdadero positivo. Con todo, debe evitarse intentar aumentar el conjunto de entrenamiento de un modelo mezclando varios de los conjuntos de desarrollo construidos para los AVE: estos pueden incluir pares o tripletas que formen parte del conjunto de prueba que estemos evaluando, quedando alterados, en consecuencia, los resultados que se obtuviesen.

Por otra parte, el uso de conjuntos de entrenamiento para la clasificación de secuencias que incluyen pares o tripletas en los que la vinculación es desconocida o neutral plantea, de nuevo, la necesidad de decidir cómo interpretarlos durante el proceso de ajuste manual de nuestros modelos. En estos casos se ha optado también por efectuar tres tipos de evaluaciones: en la primera, realizando el ajuste del modelo considerando que existe implicación textual en estos pares; en la segunda, rechazando, en el proceso de ajuste, que exista vinculación entre ellos; y finalmente, en la tercera, ignorando los pares etiquetados como *neutral* durante el procedimiento de ajuste del modelo.

Un último aspecto a tener en cuenta para poder valorar nuestros modelos según el procedimiento establecido en los AVE es el de la construcción de las hipótesis. Los conjuntos de prueba y desarrollo de los AVE 2007 y 2008 no incluyen hipótesis ya construidas o elaboradas, como sí ocurría en la edición de 2006. Necesitamos, por tanto, crearlas para utilizarlas en los procesos de ajuste o evaluación. El problema de la Generación Automática de Hipótesis no es trivial

y, al no formar parte del objetivo principal de este trabajo, se ha optado por una confección muy rudimentaria: las hipótesis correspondientes a las tripletas de los conjuntos, tanto en los de prueba como en los de desarrollo, se formarán a partir de la concatenación de la pregunta, un carácter 'espacio' y la respuesta.

Describamos cómo se han utilizado, en nuestros experimentos, los modelos introducidos en 4.1 en combinación con los conjuntos de entrenamiento del apartado 4.2:

- **BERT base**: El modelo utilizado procede de la biblioteca *d2l* que es iniciativa de los autores del libro *Dive into Deep Learning* [69]. Los conjuntos empleados para su ajuste son los conjuntos de desarrollo y prueba procedentes de RTE 1, 2 y 3, la propia colección del AVE correspondiente a cada evaluación (cuando existe) y parte de SNLI . Estos conjuntos se utilizan por separado o bien combinados en diferentes entrenamientos.

  El proceso de ajuste se realiza en una máquina que cuenta con una tarjeta gráfica Nvidia GeForce RTX 2060 Mobile con 6,3 GiB de memoria. Es necesario disminuir el tamaño de los lotes a 64 y truncar la longitud de la secuencia de textos del conjunto de entrenamiento a 60 *tokens* para no agotar la memoria disponible durante el proceso de ajuste. La falta de memoria es también la responsable de que utilicemos SNLI de forma parcial (10.000 y 50.000 pares). El entrenamiento o ajuste se realiza durante cinco tandas (*epochs*) con una tasa de aprendizaje de $5{\cdot}10^{-5}$. Siempre se emplearán estos parámetros para facilitar la comparación entre los resultados.

  BERT-base también ha sido aplicado en otros dos casos de estudio. En ambos se emplean modelos de *Hugging Face* ya preajustados: el primero sobre los conjuntos completos de SNLI y MNLI, y el segundo sobre SNLI, MNLI y Hans.

- **BETO** (BERT base): Se hace uso de la biblioteca *transformer* de *Hugging Face* para descargar este modelo preentrenado. En una implementación *PyTorch,* el modelo se ajusta sobre el conjunto SICK_ES en combinación, en

algunos casos, con la colección del AVE propia de cada evaluación (si existe).

Se disminuye el tamaño de los lotes a 64 y se trunca la longitud de la secuencia de textos del conjunto de entrenamiento a 70 *tokens* para no agotar la memoria disponible durante el afinamiento. Este se realiza durante cinco tandas (*epochs*) con una tasa de aprendizaje de $5{\cdot}10^{-5}$. Siempre se emplearán estos parámetros para facilitar la comparación entre los resultados.

- **BERT large, ALBert base y ALBert large**: Son modelos que se obtienen en *Hugging Face* y que ya fueron ajustados sobre las colecciones SNLI y MNLI completas. En algunos de ellos (y en los BERT-base preajustados antes mencionados) se hace necesario limitar a 512 caracteres la longitud máxima de los textos del conjunto de prueba (particularmente AVE 2008 Test) para evitar problemas de memoria en la GPU.

- **ALBert xx-large, BART large, RoBERTa large, ELECTRA large discriminator y XLNet large cased**: Son modelos de *Hugging Face* ya ajustados sobre la combinación de los conjuntos SNLI, MNLI, FEVER-NLI y ANLI (R1, R2, R3).

  RoBERTa-large se ha empleado, además, en un segundo caso de estudio: habiendo sido ajustado únicamente sobre la colección MNLI.

- **Bertin base** (RoBERTa base) **y Electricidad small** (ELECTRA for sequence classification): Son modelos procedentes de *Hugging Face* que fueron ajustados exclusivamente sobre el conjunto XNLI_ES.

Capítulo 5

Evaluación

El objetivo de este trabajo es evaluar el rendimiento de diferentes modelos *Transformer* al realizar labores de Validación de Respuestas. En este capítulo se presentan los resultados obtenidos en los experimentos, agrupándose estos según las colecciones de datos empleadas durante el proceso de ajuste de los modelos.

5.1. Resultados de los modelos ajustados con RTE, AVE o SNLI

Aplicamos varias combinaciones de las colecciones en lengua inglesa RTE 1, 2 y 3, AVE y SNLI para ajustar, en nuestro propio equipo, un modelo BERT-base. SNLI es utilizado solo de forma parcial, con lo que los experimentos de este apartado emplean colecciones de entrenamiento pequeñas. Al evaluar los distintos modelos generados sobre los conjuntos de prueba de los AVE se obtienen los resultados recogidos en las tablas 4, 5 y 6.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
BERT base	RTE 1, 2 y 3 (Dev & Test), primeros 10000 pares de SNLI (vinculación en pares neutrales)	Sin resultados neutrales	0.4491	0.3191	0.7576

BERT base	RTE 1, 2 y 3 (Dev & Test), primeros 50000 pares de SNLI no neutrales	Sin resultados neutrales	0.4444	0.3359	0.6566
BERT base	Primeros 10000 pares de SNLI. Contradicción en pares neutrales.	Sin resultados neutrales	0.4423	0.3218	0.7071
BERT base	RTE 1, 2 y 3 (Dev & Test), primeros 10000 pares de SNLI (contradicción en pares neutrales)	Sin resultados neutrales	0.4407	0.3292	0.6667
BERT base	RTE 1, 2 y 3 (Dev & Test), primeros 10000 pares de SNLI no neutrales	Sin resultados neutrales	0.4361	0.3105	0.7323
BERT base	Primeros 50000 pares de SNLI no neutrales	Sin resultados neutrales	0.4194	0.2780	0.8535
BERT base	RTE 1, 2 y 3 (Dev & Test)	Sin resultados neutrales	0.4157	0.3136	0.6162
BERT base	Primeros 10000 pares de SNLI no neutrales	Sin resultados neutrales	0.4014	0.2601	0.8788
BERT base	Primeros 10000 pares de SNLI. Vinculación en pares neutrales.	Sin resultados neutrales	0.3039	0.1795	0.9899
Resultado de referencia en ejercicio AVE 2006			0.4559	0.3261	0.7576
Modelo de referencia 100% validación			0.2742	0.1589	1

Tabla 4. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa RTE 1, 2 y 3 (Dev & Test), fracciones de SNLI o una combinación de ambas.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 07 Dev, primeros 10000 pares de SNLI (contradicción en pares neutrales)	Sin resultados neutrales	0.3636	0.5000	0.2857
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 07 Dev	Sin resultados neutrales	0.3448	0.6250	0.2381
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 07 Dev, primeros 10000 pares de SNLI no neutrales	Sin resultados neutrales	0.3256	0.3182	0.3333

BERT base	Primeros 10000 pares de SNLI no neutrales	Sin resultados neutrales	0.2970	0.1875	0.7143
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 07 Dev, primeros 50000 pares de SNLI no neutrales	Sin resultados neutrales	0.2899	0.2083	0.4762
BERT base	Primeros 50000 pares de SNLI no neutrales	Sin resultados neutrales	0.2880	0.1731	0.8571
BERT base	Primeros 10000 pares de SNLI. Vinculación en pares neutrales.	Sin resultados neutrales	0.2548	0.1471	0.9524
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 07 Dev, primeros 10000 pares de SNLI (vinculación en pares neutrales)	Sin resultados neutrales	0.2353	0.3077	0.1905
BERT base	Primeros 10000 pares de SNLI. Contradicción en pares neutrales.	Sin resultados neutrales	0.2245	0.1429	0.5238
BERT base	RTE 1, 2 y 3 (Dev & Test)	Sin resultados neutrales	0.2105	0.2353	0.1905
Resultado de referencia en ejercicio AVE 2007			0.55	0.44	0.71
Modelo de referencia 100% validación			0.19	0.11	1

Tabla 5. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa RTE 1, 2 y 3 (Dev & Test), AVE 07 Dev, fracciones de SNLI o una combinación de ellos.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 08 Dev, primeros 10000 pares de SNLI no neutrales	Sin resultados neutrales	0.4859	0.4388	0.5443
BERT base	Primeros 10000 pares de SNLI. Contradicción en pares neutrales.	Sin resultados neutrales	0.3776	0.2462	0.8101
BERT base	RTE 1, 2 y 3 (Dev & Test)	Sin resultados neutrales	0.3601	0.2414	0.7089
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 08 Dev	Sin resultados neutrales	0.3402	0.2870	0.4177

BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 08 Dev, primeros 50000 pares de SNLI no neutrales	Sin resultados neutrales	0.3094	0.2161	0.5443
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 08 Dev, primeros 10000 pares de SNLI (vinculación en pares neutrales)	Sin resultados neutrales	0.3057	0.2333	0.4430
BERT base	Primeros 50000 pares de SNLI no neutrales	Sin resultados neutrales	0.2590	0.1488	1.0000
BERT base	RTE 1, 2 y 3 (Dev & Test), AVE 08 Dev, primeros 10000 pares de SNLI (contradicción en pares neutrales)	Sin resultados neutrales	0.2424	0.3019	0.2025
BERT base	Primeros 10000 pares de SNLI no neutrales	Sin resultados neutrales	0.2385	0.1357	0.9873
BERT base	Primeros 10000 pares de SNLI. Vinculación en pares neutrales.	Sin resultados neutrales	0.1948	0.1080	0.9873
Resultado de referencia en ejercicio AVE 2008			0.64	0.54	0.78
Modelo de referencia 100% validación			0.14	0.08	1

Tabla 6. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa RTE 1, 2 y 3 (Dev & Test), AVE 08 Dev, fracciones de SNLI o una combinación de ellos.

Todos los resultados superan el del modelo base de referencia que valida el 100% de los pares o tripletas. Ninguno, sin embargo, mejora el rendimiento del modelo más destacado de cada una de las ediciones de los AVE para lengua inglesa. Los mejores resultados se obtienen ajustando con RTE, AVE (cuando hay conjunto de desarrollo) y una fracción pequeña de SNLI. Aumentar los pares de SNLI durante el ajuste o entrenar solo con SNLI suele arrojar peores resultados, al alejarse este conjunto en mayor medida, del tipo y estilo de textos que se utilizan en los AVE.

5.2. Resultados de los modelos ajustados con SNLI, MNLI o Hans.

Cuando se ha experimentado con modelos de *Hugging Face* ajustados sobre combinaciones de los conjuntos SNLI, MNLI y Hans en lengua inglesa se consiguen los resultados de las tablas 7, 8 y 9.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
RoBERTa large	MNLI	Contradicción	0.5899	0.4507	0.8535
BERT large	SNLI, MNLI	Contradicción	0.5437	0.4000	0.8485
ALBert large	SNLI, MNLI	Contradicción	0.5423	0.3963	0.8586
BERT base	SNLI, MNLI	Contradicción	0.5329	0.3951	0.8182
BERT base	SNLI, MNLI, Hans	Contradicción	0.5108	0.3673	0.8384
ALBert base	SNLI, MNLI	Contradicción	0.5103	0.3718	0.8131
RoBERTa large	MNLI	Vinculación	0.3508	0.2144	0.9646
ALBert large	SNLI, MNLI	Vinculación	0.3437	0.2098	0.9495
ALBert base	SNLI, MNLI	Vinculación	0.3346	0.2042	0.9242
BERT base	SNLI, MNLI	Vinculación	0.3345	0.2026	0.9596
BERT large	SNLI, MNLI	Vinculación	0.3255	0.1968	0.9394
BERT base	SNLI, MNLI, Hans	Vinculación	0.3325	0.2011	0.9596
Resultado de referencia en ejercicio AVE 2006			0.4559	0.3261	0.7576
Modelo de referencia 100% validación			0.2742	0.1589	1

Tabla 7. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, Hans o una combinación de ellos.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
RoBERTa large	MNLI	Contradicción	0.4615	0.5000	0.4286

ALBert base	SNLI, MNLI	Contradicción	0.4364	0.3529	0.5714
BERT large	SNLI, MNLI	Contradicción	0.3556	0.3333	0.3810
RoBERTa large	MNLI	Vinculación	0.2941	0.1852	0.7143
ALBert base	SNLI, MNLI	Vinculación	0.2881	0.1753	0.8095
BERT large	SNLI, MNLI	Vinculación	0.2521	0.1531	0.7143
BERT base	SNLI, MNLI	Vinculación	0.2435	0.1489	0.6667
BERT base	SNLI, MNLI	Contradicción	0.2326	0.2273	0.2381
BERT base	SNLI, MNLI, Hans	Vinculación	0.2241	0.1368	0.6190
ALBert large	SNLI, MNLI	Vinculación	0.2222	0.1500	0.4286
ALBert large	SNLI, MNLI	Contradicción	0.1935	0.3000	0.1429
BERT base	SNLI, MNLI, Hans	Contradicción	0.1081	0.1250	0.0952
Resultado de referencia en ejercicio AVE 2007			0.55	0.44	0.71
Modelo de referencia 100% validación			0.19	0.11	1

Tabla 8. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, Hans o una combinación de ellos.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
ALBert base	SNLI, MNLI	Contradicción	0.6154	0.5172	0.7595
BERT base	SNLI, MNLI	Contradicción	0.5683	0.5000	0.5683
BERT base	SNLI, MNLI, Hans	Contradicción	0.5488	0.5294	0.5696
BERT large	SNLI, MNLI	Contradicción	0.5000	0.4455	0.5696
RoBERTa large	MNLI	Contradicción	0.4941	0.4615	0.5316
ALBert large	SNLI, MNLI	Contradicción	0.2787	0.3953	0.2152
ALBert base	SNLI, MNLI	Vinculación	0.2239	0.1271	0.9367
BERT base	SNLI, MNLI	Vinculación	0.2016	0.1128	0.9494
BERT base	SNLI, MNLI, Hans	Vinculación	0.1895	0.1057	0.9114
RoBERTa large	MNLI	Vinculación	0.1874	0.1060	0.8101

BERT large	SNLI, MNLI	Vinculación	0.1808	0.1012	0.8481
ALBert large	SNLI, MNLI	Vinculación	0.1155	0.0677	0.3924
Resultado de referencia en ejercicio AVE 2008			0.64	0.54	0.78
Modelo de referencia 100% validación			0.14	0.08	1

Tabla 9. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, Hans o una combinación de ellos.

Solo para AVE 2006 varios modelos mejoran el resultado más destacado que se obtuvo en aquella edición para lengua inglesa. Las tablas 7, 8 y 9 muestran que los mejores rendimientos se alcanzan cuando los resultados neutrales de los modelos son contabilizados, en el cálculo de las métricas, como resultados incorrectos. Asimismo, los modelos BERT-base que incorporan Hans entre sus colecciones de entrenamiento obtienen siempre resultados inferiores al de sus modelos homólogos que fueron ajustados sin ella. Hans es un conjunto en el que sus pares adoptan una sintaxis compleja. La diferencia entre los estilos de los textos seguidos por esta colección y por los conjuntos de prueba de los AVE explicaría esta pérdida de rendimiento al evaluar.

5.3. Resultados de los modelos ajustados con SNLI, MNLI, FEVER-NLI y ANLI

Al experimentar con grandes modelos procedentes de *Hugging Face* que han sido ajustados con las colecciones de entrenamiento SNLI, MNLI, FEVER-NLI y ANLI se obtienen, para cada una de las ediciones de los AVE, los resultados de las tablas 10, 11 y 12.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
RoBERTa large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.6365	0.5075	0.8535

BART large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.5927	0.4631	0.8232
XLNet large cased	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.5919	0.4531	0.8535
ALBert xx-large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.5859	0.4563	0.8182
ELECTRA large discriminator	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.4795	0.3552	0.7374
BART large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.3266	0.1977	0.9394
XLNet large cased	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.3232	0.1950	0.9444
RoBERTa large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.3214	0.1933	0.9545
ALBert xx-large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.3101	0.1847	0.9646
ELECTRA large discriminator	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.2885	0.1718	0.8990
Resultado de referencia en ejercicio AVE 2006			0.4559	0.3261	0.7576
Modelo de referencia 100% validación			0.2742	0.1589	1

Tabla 10. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, FEVER-NLI y ANLI (R1, R2, R3).

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
ALBert xx-large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.5818	0.4706	0.7619
BART large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.5581	0.5455	0.5714
RoBERTa large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.5556	0.4545	0.7143
XLNet large cased	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.4068	0.3158	0.5714

ELECTRA large discriminator	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.3371	0.2206	0.7143
RoBERTa large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.2745	0.1591	1.0000
BART large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.2685	0.1562	0.9524
XLNet large cased	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.2521	0.1531	0.7143
ELECTRA large discriminator	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.2395	0.1370	0.9524
ALBert xx-large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.2262	0.1293	0.9048
Resultado de referencia en ejercicio AVE 2007			0.55	0.44	0.71
Modelo de referencia 100% validación			0.19	0.11	1

Tabla 11. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, FEVER-NLI y ANLI (R1, R2, R3).

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
BART large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.6734	0.5583	0.8481
RoBERTa large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.6667	0.5440	0.8608
XLNet large cased	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.4823	0.3350	0.8608
ALBert xx-large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.3636	0.2923	0.4810
ELECTRA large discriminator	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Contradicción	0.3082	0.2024	0.6456
XLNet large cased	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.2095	0.1177	0.9494
RoBERTa large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.1761	0.0967	0.9873

BART large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.1754	0.0967	0.9367
ALBert xx-large	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.1709	0.0937	0.9747
ELECTRA large discriminator	SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3)	Vinculación	0.1282	0.0702	0.7342
Resultado de referencia en ejercicio AVE 2008			0.64	0.54	0.78
Modelo de referencia 100% validación			0.14	0.08	1

Tabla 12. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con los conjuntos de entrenamiento en lengua inglesa SNLI, MNLI, FEVER-NLI y ANLI (R1, R2, R3).

Todos los resultados, excepto uno, superan el del modelo base de referencia que valida el 100% de los pares o tripletas. Asimismo, se obtienen mejoras en los resultados para todas las ediciones de los AVE en lengua inglesa (con respecto al modelo más destacado en cada una). En las tres tablas, los mejores rendimientos se logran, invariablemente, cuando los resultados neutrales de los modelos son contabilizados como incorrectos al calcular las métricas. Destacan los modelos BART-large y RoBERTa-large al ofrecer y mantener, en todas las pruebas, rendimientos altos. BART-large y RoBERTa-large son los modelos, de entre todos los evaluados en este trabajo, que emplean un mayor número de parámetros en su diseño. Como se sostuvo en el apartado 4.1, ambos muestran un rendimiento similar cuando son ajustados con recursos de entrenamiento comparables (en este caso, con las mismas colecciones).

5.4. Resultados de los modelos ajustados con SICK_ES, AVE o XNLI_ES

BETO es un BERT-base preentrenado en español y que hemos ajustado para este trabajo, en nuestro propio equipo, utilizando varias combinaciones de los conjuntos SICK_ES y AVE. Al evaluar estas variantes sobre las colecciones de prueba en castellano de los diferentes AVE se obtienen los resultados de las

tablas 13, 14 y 15.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
BETO	SICK_ES (contradicción en pares neutrales)	Sin resultados neutrales	0.6063	0.6137	0.5991
BETO	SICK_ES (pares no neutrales)	Sin resultados neutrales	0.4455	0.2961	0.8986
BETO	SICK_ES (vinculación en pares neutrales)	Sin resultados neutrales	0.4329	0.2827	0.9239
Resultado de referencia en ejercicio AVE 2006			0.6063	0.5270	0.7139
Modelo de referencia 100% validación			0.4538	0.2935	1

Tabla 13. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con el conjunto de entrenamiento en lengua española SICK_ES.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
BETO	SICK_ES (contradicción en pares neutrales), AVE 07 Dev	Sin resultados neutrales	0.5428	0.6867	0.4488
BETO	SICK_ES (contradicción en pares neutrales)	Sin resultados neutrales	0.5092	0.6179	0.4330
BETO	SICK_ES (pares no neutrales), AVE 07 Dev	Sin resultados neutrales	0.4365	0.6142	0.3385
BETO	SICK_ES (pares no neutrales)	Sin resultados neutrales	0.3762	0.2379	0.8976
BETO	SICK_ES (vinculación en pares neutrales)	Sin resultados neutrales	0.3733	0.2299	0.9921
Resultado de referencia en ejercicio AVE 2007			0.53	0.38	0.86
Modelo de referencia 100% validación			0.37	0.23	1

Tabla 14. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con el conjunto de entrenamiento en lengua española SICK_ES combinado, en algunos casos, con AVE 07 Dev.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
BETO	SICK_ES (contradicción en pares neutrales), AVE 08 Dev	Sin resultados neutrales	0.4560	0.3933	0.5424
BETO	SICK_ES (contradicción en pares neutrales)	Sin resultados neutrales	0.3603	0.5797	0.2614
BETO	SICK_ES (pares no neutrales), AVE 08 Dev	Sin resultados neutrales	0.3557	0.2549	0.5882
BETO	SICK_ES (pares no neutrales)	Sin resultados neutrales	0.1970	0.1104	0.9150
BETO	SICK_ES (vinculación en pares neutrales)	Sin resultados neutrales	0.1688	0.0932	0.8888
Resultado de referencia en ejercicio AVE 2008			0.44	0.32	0.67
Modelo de referencia 100% validación			0.18	0.10	1

Tabla 15. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con el conjunto de entrenamiento en lengua española SICK_ES combinado, en algunos casos, con AVE 08 Dev.

Los peores resultados se obtienen al considerar, durante el ajuste, los pares neutrales de SICK_ES como resultados positivos (no llegando a superar a los modelos base de referencia para las ediciones de 2006 y 2008). De hecho, si para ajustar estos mismos modelos, aumentamos el corpus de entrenamiento añadiéndoles los conjuntos de desarrollo de los AVE, no se obtiene ninguna respuestas positiva verdadera. Sin embargo, se logran resultados mucho mejores cuando estimamos que no hay vinculación entre los textos de estos pares. De esta forma, se consigue igualar o mejorar los resultados de los modelos más destacados de todas las ediciones de los AVE, a pesar de contar SICK_ES con menos de 10.000 pares. Añadir a estos modelos el conjunto de desarrollo de cada AVE durante su ajuste (cuando existe), arroja mejor resultado. No obstante, ignorar los pares neutrales de SICK_ES reduce el conjunto de entrenamiento a más de la mitad, ofreciendo, las pruebas, rendimientos intermedios.

Cuando se ha experimentado con modelos de *Hugging Face* preentrenados en español y ajustados sobre la colección XNLI_ES en castellano se consiguen los

resultados de las tablas 16, 17 y 18.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
Bertin base (RoBERTa base)	XNLI_ES	Contradicción	0.6141	0.4833	0.8420
Electricidad small (ELECTRA for sequence classification)	XNLI_ES	Contradicción	0.5934	0.4463	0.8852
Bertin base (RoBERTa base)	XNLI_ES	Vinculación	0.4965	0.3420	0.9061
Electricidad small (ELECTRA for sequence classification)	XNLI_ES	Vinculación	0.4901	0.3291	0.9598
Resultado de referencia en ejercicio AVE 2006			0.6063	0.5270	0.7139
Modelo de referencia 100% validación			0.4538	0.2935	1

Tabla 16. Resultados obtenidos sobre el conjunto de prueba de AVE 2006 por modelos ajustados con el conjunto de entrenamiento en lengua española XNLI_ES.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
Electricidad small (ELECTRA for sequence classification)	XNLI_ES	Contradicción	0.4650	0.3407	0.7323
Bertin base (RoBERTa base)	XNLI_ES	Contradicción	0.4111	0.4127	0.4094
Bertin base (RoBERTa base)	XNLI_ES	Vinculación	0.3965	0.2551	0.8898
Electricidad small (ELECTRA for sequence classification)	XNLI_ES	Vinculación	0.3869	0.2518	0.8346
Resultado de referencia en ejercicio AVE 2007			0.53	0.38	0.86
Modelo de referencia 100% validación			0.37	0.23	1

Tabla 17. Resultados obtenidos sobre el conjunto de prueba de AVE 2007 por modelos ajustados con el conjunto de entrenamiento en lengua española XNLI_ES.

Modelo	Conjunto(s) de datos usado(s) en el ajuste del modelo	Implicación considerada con resultados neutrales	Medida F_1	Precisión	Exhaustividad
Electricidad small (ELECTRA for sequence classification)	XNLI_ES	Contradicción	0.2814	0.1783	0.6667
Bertin base (RoBERTa base)	XNLI_ES	Contradicción	0.2766	0.2332	0.3399
Electricidad small (ELECTRA for sequence classification)	XNLI_ES	Vinculación	0.2085	0.1188	0.8497
Bertin base (RoBERTa base)	XNLI_ES	Vinculación	0.1963	0.1114	0.8235
Resultado de referencia en ejercicio AVE 2008			0.44	0.32	0.67
Modelo de referencia 100% validación			0.18	0.10	1

Tabla 18. Resultados obtenidos sobre el conjunto de prueba de AVE 2008 por modelos ajustados con el conjunto de entrenamiento en lengua española XNLI_ES.

Siempre se consigue superar el resultado del modelo base de referencia que valida el 100% de los pares o tripletas, pero solo para AVE 2006 se logra, en una ocasión, mejorar el rendimiento de su modelo más destacado para lengua española. En las tres tablas, los mejores registros se alcanzan cuando los resultados neutrales de los modelos se juzgan incorrectos al calcular las métricas.

Capítulo 6

Discusión

Las técnicas de Aprendizaje Profundo se están aplicando con éxito en muchas áreas del Procesamiento del Lenguaje Natural como la traducción automática o el análisis de sentimientos [38]. También han empezado a ser aplicadas en los problemas de implicación textual para los que ya se han diseñado varios sistemas [36]. La implicación o inferencia textual mantiene una fuerte relación con muchas tareas del Procesamiento del Lenguaje Natural como la validación de respuestas o la elaboración de resúmenes.

En este trabajo se han evaluado diferentes modelos neuronales de Aprendizaje Profundo con arquitectura *Transformer* para valorar su rendimiento en tareas de Validación de Respuestas (a través de la inferencia textual). Se han empleado versiones del modelo BERT y otros modelos derivados o similares. Todos ellos han sido ajustados bajo colecciones de datos adecuadas para realizar un entrenamiento en labores de validación de respuestas.

Se ha mostrado que los modelos evaluados pueden ofrecer rendimientos superiores a los de las técnicas probadas durante los AVE. En particular, tres de ellos sobresalen por los resultados conseguidos en todas las ediciones del AVE: BART-large y RoBERTa-large para lengua inglesa (ambos ajustados en la suma de las colecciones SNLI, MNLI, FEVER-NLI y ANLI) y BETO para lengua española (ajustado en SICK_ES). En consecuencia, se logran buenos resultados

tanto con grandes modelos que han sido afinados con colecciones muy amplias, como con modelos con pocos parámetros en los que se ha empleado un corpus pequeño.

No obstante, en las tablas de resultados del apartado 4, se observa, muy a menudo, cómo los mejores rendimientos de los modelos que devuelven resultados neutrales se alcanzan cuando estos son contabilizados como resultados negativos. La estrategia de considerar los resultados neutrales (no contemplados en los AVE) como resultados negativos o positivos es equivalente a la que podría seguirse en otros modelos (como, por ejemplo, BETO) adecuando a conveniencia el valor umbral que les permite, en la evaluación, decantarse por un resultado, bien positivo, bien negativo. Se ajustarían, entonces, los hiperparámetros para así adaptarlos a las características del conjunto de datos con el que se trabaja.

Dicho esto, dada la naturaleza desbalanceada de los conjuntos de prueba de los AVE (en los que encontramos más respuestas negativas que positivas), la decisión de considerar los resultados neutrales como negativos podría conseguir disminuir el número de respuestas negativas falsas, afectando al alza a la exhaustividad ofrecida por el modelo, aunque nunca a su precisión. De igual forma, cuando contabilizamos como positivos los resultados neutrales devueltos por un modelo, al ser mayor la proporción real de respuestas negativas, es más probable que estemos cometiendo errores con esta decisión, con lo que obtendremos más respuestas positivas falsas, influyendo así a la baja en la precisión del modelo. En consecuencia, es natural que los modelos arrojen mejores rendimientos cuando contabilizamos como negativos sus resultados neutrales.

Asimismo, la estrategia de considerar que no hay vinculación en los pares que están etiquetados como *neutral* en los conjuntos de desarrollo de SNLI o SICK_ES, sería equivalente a la de juzgar, durante el proceso de creación de las colecciones de desarrollo, que no habrá vinculación entre sus pares a menos que esta sea fehaciente e incuestionable. Esta decisión discrecional, aplicada durante el proceso de ajuste manual de modelos como BERT-base o BETO, también

debería favorecer la exhaustividad de los modelos cuando evaluamos sobre los conjuntos de prueba de los AVE. Actuando de esta forma, entrenamos al modelo para que dé una respuesta negativa ante una mayor variedad de pares y, dado que, en los conjuntos de prueba de los AVE, la proporción de respuestas negativas es mayor, es probable que se obtengan así menos respuestas negativas falsas durante las evaluaciones. Análogamente, si ajustamos los modelos estimando que existe vinculación en los pares etiquetados como *neutral,* tendremos más posibilidades de obtener más respuestas positivas falsas y conseguir menos precisión. Estas circunstancias podrían explicar parcialmente el patrón observado en las tablas 13, 14 y 15 con BETO con respecto a la consideración dada a los pares *neutral* de SICK_ES. Sin embargo, no queda tan claro su impacto en las métricas de las tablas 4, 5 y 6 con BERT-base y SNLI, especialmente al evaluar para AVE 2007.

En consecuencia, teniendo en cuenta que los pares etiquetados como *neutral* en SICK_ES representan casi el 57% del total y en SNLI aproximadamente el 33%, las estrategias aplicadas con las que se han logrado las mejores métricas han podido llevarnos hacia unos modelos bien adaptados a la evaluación de los AVE pero que no generalizarían muy bien. Los resultados de los modelos BERT-base o BETO cuando han sido ajustados ignorando los pares *neutral* (de SNLI y SICK_ES respectivamente), sugieren que necesitaríamos conjuntos de entrenamiento mucho mayores y de temática variada para mejorarlos.

A este respecto, los factores que han desfavorecido los resultados que se han alcanzado, en general, son:

- Los conjuntos de entrenamiento para realizar los ajustes (tanto en inglés como en castellano) no siguen una distribución (en cuanto a temática de las fuentes, estilo o longitud de los textos) que incluya la adoptada en los AVE.
- La confección de las hipótesis para la evaluación de los modelos sobre los conjuntos de prueba de AVE 2007 y 2008 es excesivamente rudimentaria. No se forman frases con estructuras gramaticales elaboradas, simplemente se concatenan preguntas y respuestas. Los modelos no se han ajustado para trabajar con este tipo de oraciones.

- En los modelos con los que se ha practicado un ajuste en nuestro propio equipo, nos hemos visto obligados a limitar el tamaño de las secuencias y el de los lotes durante el proceso de ajuste para evitar agotar la memoria de la tarjeta gráfica. Los modelos recibirían un mejor entrenamiento sin estas limitaciones.

Con todo, en todas las pruebas realizadas, siempre podemos encontrar modelos que ofrecen rendimientos superiores o muy superiores al modelo de referencia (*baseline*) en el que se valida el 100% de las respuestas. Asimismo, para todas las ediciones de los AVE y tanto para lengua inglesa como española, se han logrado resultados que han igualado o superado los de la aproximación de referencia que ofrecía el mejor rendimiento. Por todo ello, las técnicas de Aprendizaje Profundo con arquitectura *Transformer* deben ser consideradas adecuadas para acometer tareas de Validación de Respuestas.

La incorporación de modelos *Transformer* a los sistemas de Búsquedas de Respuestas como verificadores de las respuestas ofrecidas por su generador, tiene los inconvenientes de añadir complejidad al sistema y de dilatar el proceso de respuesta. Puede ser ventajosa si el rendimiento del modelo de validación es apropiado, si el coste de dar una respuesta errónea es alto o si el generador de respuestas no ofrece valores de confianza asociados a sus salidas [48].

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

En este trabajo se han evaluado distintos modelos basados en arquitectura *Transformer* en el problema de la Validación de Respuestas. Se han empleado versiones básicas de BERT y modelos derivados o similares. Los mejores rendimientos alcanzados vienen de la mano de aquellos modelos que presentan un mayor número de parámetros y que han sido ajustados a partir de una combinación de grandes colecciones de datos. De esta forma, el corpus de entrenamiento resultante incluye textos de longitud variable que pertenecen a géneros o temáticas diversas, presentando así mayor heterogeneidad en su vocabulario y en sus rasgos estilísticos y sintácticos. Los modelos así ajustados muestran mayor facilidad para adaptarse a la complejidad del lenguaje y, en consecuencia, arrojan mejores resultados al ser evaluados en los conjuntos de prueba de los AVE.

7.2. Trabajo futuro

Podrían explorarse varios aspectos para tratar de mejorar los resultados

obtenidos durante este trabajo:

- La generación de colecciones sintéticas amplias y variadas en temáticas y estilos con las que realizar los ajustes de modelos preentrenados en castellano e inglés (y mayores a los aquí utilizados).
- Experimentar con otros enfoques para la generación de la hipótesis durante el ajuste y evaluación de los modelos con los conjuntos de desarrollo y prueba de los AVE 2007 y 2008.

Bibliografía

1. Kozareva, Z., Vázquez, S., & Montoyo, A. (2006, Septiembre). Adaptation of a Machine-learning Textual Entailment System to a Multilingual Answer Validation Exercise. In CLEF (Working Notes). http://clef.isti.cnr.it/2006/working_notes/workingnotes2006/kozarevaCLEF2006.pdf

2. Peñas, A., Rodrigo, Á., Sama, V., Verdejo, F. (2007). Overview of the Answer Validation Exercise 2006. In Evaluation of Multilingual and Multi-modal Information Retrieval. CLEF 2006. Lecture Notes in Computer Science, vol 4730. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-74999-8_32

3. Peñas, A., Rodrigo, A., & Verdejo, F. (2006, Febrero). Sparte, a test suite for recognising textual entailment in Spanish. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 275-286). Springer, Berlin, Heidelberg. DOI: 10.1007/11671299_29

4. Rodrigo, Á., Penas, A., Herrera, J., & Verdejo, F. (2006, Septiembre). The effect of entity recognition on answer validation. In Workshop of the Cross-Language Evaluation Forum for European Languages (pp. 483-489). Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-74999-8_57

5. Rodrigo, Á., Peñas, A., Herrera, J., & Verdejo, F. (2007, Junio). Experiments of UNED at the third recognising textual entailment challenge. In Proceedings of the ACL-PASCAL Workshop on Textual

Entailment and Paraphrasing (pp. 89-94). DOI: 10.3115/1654536.1654555

6. Peñas, A., Rodrigo, Á., Verdejo, F. (2007, Septiembre). Overview of the Answer Validation Exercise 2007. In Advances in Multilingual and Multimodal Information Retrieval. CLEF 2007. Lecture Notes in Computer Science, vol 5152. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-85760-0_28

7. Rodrigo, Á., Penas, A., & Verdejo, F. (2007, Septiembre). UNED at answer validation exercise 2007. In Workshop of the Cross-Language Evaluation Forum for European Languages (pp. 404-409). Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-85760-0_53

8. Téllez-Valero, A., Montes-y-Gómez, M., & Pineda, L. V. (2007). INAOE at AVE 2007: Experiments in Spanish Answer Validation. In CLEF (Working Notes). 1173. http://clef.isti.cnr.it/2007/working_notes/tellez_valeroCLEF07_AVE.pdf

9. Glöckner, I. (2007). University of Hagen at CLEF 2007: Answer Validation Exercise. In CLEF (Working Notes). http://ims-sites.dei.unipd.it/documents/71612/86371/CLEF2008wn-QACLEF-Glockner2008.pdf

10. Wang, R., & Neumann, G. (2007). DFKI-LT at AVE 2007: Using Recognizing Textual Entailment for Answer Validation. In CLEF (Working Notes). http://clef.isti.cnr.it/2007/working_notes/WangCLEF2007.pdf

11. Rodrigo, Á., Peñas, A., Verdejo, F. (2009). Overview of the Answer Validation Exercise 2008. In Evaluating Systems for Multilingual and Multimodal Information Access. CLEF 2008. Lecture Notes in Computer Science, vol 5706. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-04447-2_35

12. Peñas, A., Rodrigo, A., Sama, V., & Verdejo, F. (2008). Testing the reasoning for question answering validation. In Journal of Logic and Computation, 18(3), 459-474. DOI: 10.1093/logcom/exm072

13. Castillo, J. J. (2008, Septiembre). The Contribution of FaMAF at QA@ CLEF 2008. Answer Validation Exercise. In Working Notes of the CLEF 2008 Workshop. 17-19. http://clef.isti.cnr.it/2008/working_notes/castillo-paperCLEF2008.pdf

14. Rodrigo, Á., Pérez-Iglesias, J., Peñas, A., Garrido, G., & Araujo, L. (2009, Septiembre). Approaching question answering by means of paragraph validation. In Workshop of the Cross-Language Evaluation Forum for European Languages (pp. 245-252). Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-15754-7_27

15. Ray, S. K., Singh, S., & Joshi, B. P. (2009, Agosto). World Wide Web based Question Answering System-a relevance feedback framework for automatic answer validation. In 2009 Second International Conference on the Applications of Digital Information and Web Technologies (pp. 169-174). IEEE. DOI: 10.1109/ICADIWT.2009.5273942

16. Grappy, A., & Grau, B. (2010, Enero). Answer type validation in question answering systems. Recherche d'Information Assistée par Ordinateur, RIAO 2010: Adaptivity, Personalization and Fusion of Heterogeneous Information, 9th International Conference, Bibliotheque Nationale de France, Paris, France, April 28-30, 2010, Proceedings. https://hal.archives-ouvertes.fr/hal-02282099/document

17. Wu, Y., Kashioka, H., & Nakamura, S. (2010). An Unsupervised Model of Redundancy for Answer Validation. IEICE transactions on information and systems, 93(3), 624-634. DOI: 10.1587/transinf.E93.D.624

18. Pakray, P., Pal, S., Bandyopadhyay, S., & Gelbukh, A. (2010, Agosto). Automatic answer validation system on English language. In 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE) (Vol. 6, pp. V6-329). IEEE. DOI: 10.1109/ICACTE.2010.5579166

19. Pakray, P., Gelbukh, A., & Bandyopadhyay, S. (2011). Answer validation using textual entailment. In International Conference on Intelligent Text

Processing and Computational Linguistics (pp. 353-364). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19437-5_29

20. Pakray, P. (2011). Answer validation through textual entailment. In International Conference on Application of Natural Language to Information Systems (pp. 324-329). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-22327-3_48

21. Babych, S., Henn, A., Pawellek, J., & Padó, S. (2011). Dependency-Based Answer Validation for German. In CLEF (Notebook Papers/Labs/Workshop). http://clef2011.clef-initiative.eu/resources/proceedings/Babych_Clef2011.pdf

22. Pakray, P., Bhaskar, P., Banerjee, S., Pal, B. C., Bandyopadhyay, S., & Gelbukh, A. F. (2011, Enero). A Hybrid Question Answering System based on Information Retrieval and Answer Validation. In CLEF (Notebook Papers/Labs/Workshop). http://nlp.cic.ipn.mx/publications/2011/pakray_clef2011.pdf

23. Glöckner, I., & Weis, K. H. (2012, Diciembre). An integrated machine learning and case-based reasoning approach to answer validation. In 2012 11th International Conference on Machine Learning and Applications (Vol. 1, pp. 494-499). IEEE. DOI: 10.1109/ICMLA.2012.90

24. Pakray, P., Barman, U., Bandyopadhyay, S., & Gelbukh, A. (2012). Semantic answer validation using universal networking language. In International Journal of Computer Science and Information Technologies, 3(4), 4927-4932. http://www.academia.edu/download/30768420/ijcsit2012030476.pdf

25. Zhikov, V., Tolosi, L., Osenova, P., Simov, K., & Georgiev, G. (2012). Cross-Lingual Answer Validation. http://ceur-ws.org/Vol-1178/CLEF2012wn-QA4MRE-ZhikovEt2012.pdf

26. Rodrigo, Á., & Penas, A. (2014). On evaluating the contribution of validation for question answering. IEEE Transactions on Knowledge and Data Engineering, 27(4), 1157-1161. DOI: 10.1109/TKDE.2014.2373363

27. Yu, L., Hermann, K. M., Blunsom, P., & Pulman, S. (2014). Deep learning for answer sentence selection. In Proceedings of the Deep Learning and Representation Learning Workshop: NIPS-2014. https://doi.org/10.48550/arxiv.1412.1632

28. Lyu, C., Lu, Y., Ji, D., & Chen, B. (2015, Noviembre). Deep learning for textual entailment recognition. In 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 154-161). IEEE. DOI: 10.1109/ICTAI.2015.35

29. Weis, K. H. (2015). A case based reasoning approach for answer reranking in question answering. CoRR, vol. abs/1503.02917

30. Day, M. Y., & Tsai, C. C. (2016, Julio). A study on machine learning for imbalanced datasets with answer validation of question answering. In 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI) (pp. 513-519). IEEE. DOI: 10.1109/IRI.2016.76

31. Acheampong, K. N., Pan, Z. H., Zhou, E. Q., & Li, X. Y. (2016, Diciembre). Answer triggering of factoid questions: a cognitive approach. In 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP) (pp. 33-37). IEEE. DOI: 10.1109/ICCWAMTIP.2016.8079800

32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

33. Zhao, J., Su, Y., Guan, Z., & Sun, H. (2017, Septiembre). An end-to-end deep framework for answer triggering with a novel group-level objective. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 1276-1282). DOI: 10.18653/v1/D17-1131

34. Li, W., & Wu, Y. (2017). Hierarchical gated recurrent neural tensor network for answer triggering. In Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data

(pp. 287-294). Springer, Cham. DOI: 10.1007/978-3-319-69005-6_24

35. Gupta, D., Kohail, S., & Bhattacharyya, P. (2018). Combining graph-based dependency features with convolutional neural network for answer triggering. In 19th International Conference on Computational Linguistics and Intelligent Text Processing http://www.cicling.org/2018/intranet/pre-print/papers/paper_10.pdf

36. Yang, H. (2017). Recognizing textual entailment using deep learning techniques. Master's thesis, Universitat Politècnica de Catalunya. https://upcommons.upc.edu/bitstream/handle/2117/ 109802/126227.pdf

37. Tan, C., Wei, F., Zhou, Q., Yang, N., Lv, W., & Zhou, M. (2018). I know there is no answer: modeling answer validation for machine reading comprehension. In CCF International Conference on Natural Language Processing and Chinese Computing (pp. 85-97). Springer, Cham. DOI: 10.1007/978-3-319-99495-6_8

38. Mishra, A., & Bhattacharyya, P. (2018). Deep learning techniques in textual entailment. Survey Paper, Center For Indian Language Technolgy. https://www.cfilt.iitb.ac.in/resources/surveys/Anish-Survey-Entailment.pdf

39. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT (pp. 4171-4186). https://deepsense.ai/wp-content/uploads/2020/05/1810.04805.pdf

40. Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S. R., Schwenk, H., & Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 2475-2485). DOI: 10.18653/v1/D18-1269

41. Prakash, T., Tripathy, B. K., & Banu, K. S. (2018). ALICE: A Natural Language Question Answering System Using Dynamic Attention and Memory. In International Conference on Soft Computing Systems (pp.

274-282). Springer. https://doi.org/10.1007/978-981-13-1936-5_30

42. Bhatt, G., Sharma, S., & Raman, B. (2018). Attentive recurrent tensor model for community question answering. https://doi.org/10.48550/arXiv.1801.06792

43. Kumar, S., Garg, S., Mehta, K., & Rasiwasia, N. (2019). Improving answer selection and answer triggering using hard negatives. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 5911-5917). DOI: 10.18653/v1/D19-1604

44. Rodrigo, A., Herrera, J., & Peñas, A. (2019). The effect of answer validation on the performance of Question-Answering systems. In Expert Systems with Applications, 116, 351-363. https://doi.org/10.1016/j.eswa.2018.09.014

45. Rodrigues, R., Couto, P., & Rodrigues, I. (2019). IPR: The Semantic Textual Similarity and Recognizing Textual Entailment Systems. In ASSIN@ STIL (pp. 39-48). http://ceur-ws.org/Vol-2583/4_IPR.pdf

46. Trembczyk, M. (2019). Answer Triggering Mechanisms in Neural Reading Comprehension-based Question Answering Systems. Master's Thesis in Language Technology. Uppsala University. https://pdfs.semanticscholar.org/46be/fe55ff162f2589d9ca33d9b05912d80fbc9d.pdf

47. Hu, M., Wei, F., Peng, Y., Huang, Z., Yang, N., & Li, D. (2019, Julio). Read+ verify: Machine reading comprehension with unanswerable questions. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 6529-6537). https://www.aaai.org/ojs/index.php/AAAI/article/view/4619/4497

48. Marshland, K. (2019) BERT + Verify. Final Project Reports for 2019. CS224n: Natural Language Processing with Deep Learning. Stanford / Winter 2019. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15763126.pdf

49. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in neural information processing systems, 32. https://spartee.github.io/NeurIPS-2019/xlnet.pdf

50. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. https://www.cs.princeton.edu/~danqic/papers/roberta_paper.pdf

51. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 7871-7880). http://aclanthology.lst.uni-saarland.de/2020.acl-main.703.pdf

52. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. In International Conference on Learning Representations. https://static.aminer.cn/upload/pdf/63/1809/1502/5e5e18e493d709897ce3a0f2_0.pdf

53. Shao, T., Guo, Y., Chen, H., & Hao, Z. (2019). Transformer-based neural network for answer selection in question answering. In IEEE Access, 7, 26146-26156. DOI: 10.1109/ACCESS.2019.2900753

54. Mozafari, J., Fatemi, A., & Nematbakhsh, M. A. (2019). BAS: an answer selection method using BERT language model. In Journal of Computing and Security, 8(2), 1-18. https://iranjournals.nlai.ir/handle/123456789/874668

55. McCoy, R. T., Pavlick, E., & Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3428-3448). https://aclanthology.org/P19-1334.pdf

56. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3982-3992). https://aclanthology.org/D19-1410.pdf

57. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F. and Liu, Q. (2019). Tinybert: Distilling BERT for natural language understanding. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 4163-4174). https://aclanthology.org/2020.findings-emnlp.372.pdf

58. Back, S., Chinthakindi, S. C., Kedia, A., Lee, H., & Choo, J. (2019). NeurQuRI: Neural question requirement inspector for answerability prediction in machine reading comprehension. In International Conf-rence on Learning Representations. http://hdl.handle.net/10203/280390

59. Acheampong, K. N., Tian, W., Sifah, E. B., & Opuni-Boachie, K. O. A. (2020). The Emergence, Advancement and Future of Textual Answer Triggering. In Science and Information Conference (pp. 674-693). Springer, Cham. DOI: 10.1007/978-3-030-52246-9_50

60. Acheampong, K. N., & Tian, W. (2020). Advancement of Textual Answer Triggering: Cognitive Boosting. In IEEE Transactions on Emerging Topics in Computing. DOI: 10.1109/TETC.2020.3022731

61. Reddy, R. G., Sultan, M. A., Kayi, E. S., Zhang, R., Castelli, V., & Sil, A. (2020). Answer Span Correction in Machine Reading Comprehension. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 2496-2501). https://aclanthology.org/2020.findings-emnlp.226.pdf

62. Doxolodeo, K., & Mahendra, R. (2020). UI at SemEval-2020 Task 4: Commonsense Validation and Explanation by Exploiting Contradiction. In Proceedings of the Fourteenth Workshop on Semantic Evaluation (pp. 614-619). https://aclanthology.org/2020.semeval-1.78

63. Wu, X. (2020). Machine Reading Comprehension with Enhanced Linguistic Verifiers. https://openreview.net/pdf?id=EVV259WQuFG

64. Zhang, S., Zhao, H., Wu, Y., Zhang, Z., Zhou, X., & Zhou, X. (2020). DCMN+: Dual co-matching network for multi-choice reading comprehension. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 9563-9570). https://ojs.aaai.org/index.php/AAAI/article/view/6502

65. Zhang, Z., Wu, Y., Zhou, J., Duan, S., Zhao, H., & Wang, R. (2020). SG-Net: Syntax-guided machine reading comprehension. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 9636-9643). DOI: 10.1609/aaai.v34i05.6511.

66. Zhang, Z., Yang, J., & Zhao, H. (2020). Retrospective reader for machine reading comprehension. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 16, pp. 14506-14514). https://ojs.aaai.org/index.php/AAAI/article/view/17705

67. Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2016). Electra: Pre-training text encoders as discriminators rather than generators. In ELECTRA, 85, 90. https://www-cs.stanford.edu/~kevclark/resources/electra.pdf

68. Chakravarti, R., & Sil, A. (2021). Towards Confident Machine Reading Comprehension. https://doi.org/10.48550/arXiv.2101.07942

69. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning (https://d2l.ai/). https://doi.org/10.48550/arXiv.2106.11342

70. Gabriel, S., Bhagavatula, C., Shwartz, V., Le Bras, R., Forbes, M., & Choi, Y. (2021). Paragraph-level commonsense transformers with recurrent memory. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 14, pp. 12857-12865). https://doi.org/10.1609/aaai.v35i14.17521

71. Huertas-Tato, J., Martín, A., & Camacho, D. (2022). SILT: Efficient transformer training for inter-lingual inference. In Expert Systems with Applications, 200, 116923. https://www.sciencedirect.com/science/article/pii/S0957417422003578