TRABAJO FIN DE MÁSTER

Búsqueda de Respuestas en foros usando modelos basados en Transformers

Trabajo Fin de Máster

Javier Lasheras Navas

Trabajo de investigación para el

Máster en Tecnologías del Lenguaje

Universidad Nacional de Educación a Distancia

Dirigido por

Prof. Victor Fresno

Prof. Alvaro Rodrigo Yuste

Resumen

Durante los últimos años se ha hecho más necesaria la búsqueda de información en internet. Debido a dicha necesidad, se han creado sitios web donde los usuarios tratan de ayudar a otros usuarios contestando sus preguntas. Estos sitios se llaman foros de respuesta colaborativa. Esto ha generado un problema: muchas de las preguntas que se plantean ya se han respondido previamente y es deseable poder recuperar respuestas a preguntas similares. Es por ello por lo que surgió la tarea de Community Question Answering (CQA), buscando permitir un acceso más cómodo a la información publicada en estos sitios web. Dentro de esta tarea han surgido varias subtareas y en este trabajo se aborda una de la más recurrentes: Dada una pregunta y un conjunto de respuestas, encontrar las respuestas que mejor encajan con la pregunta. Para ello se usarán técnicas de Deep Learning, y más en particular una nueva arquitectura de redes neuronales, los Transformers, que utilizan los mecanismos de atención. Mediante dicha técnica se ha intentado clasificar y ordenar las respuestas de distintos foros en función de la pregunta formulada. Se ha probado su validez en función de 3 colecciones distintas: SemEval 2015, 2017 y AmazonQA. Se ha concluido que los resultados extraídos con este tipo de métodos permiten superar los resultados anteriores, abriendo la posibilidad hacia una mayor implantación de estas tecnologías en sistemas reales.

Abstract

During the last few years it has become more necessary to search for information on the Internet. Due to this need, websites have been created where users try to help other users by answering their questions. These sites are called collaborative response forums. This has created a problem: many of the questions asked have already been answered previously and it is desirable to be able to retrieve answers to similar questions. This is why the task of Community Question Answering (CQA) arose, seeking to allow more convenient access to the information posted on these websites. Within this task, several subtasks have emerged and this paper addresses one of the most recurrent ones: Given a question and a set of answers, find the answers that best fit the question. For this purpose, Deep Learning techniques will be used, and more in particular a new neural network architecture, the Transformers, which use the attention mechanisms. By means of this technique, an attempt has been made to classify and order the answers from different forums according to the question asked. Its effectiveness has been tested on the basis of 3 different collections: SemEval 2015, 2017 and AmazonQA. It has been concluded that the results extracted with this type of methods allow to overcome the previous results, opening the possibility towards a greater implementation of these technologies in real systems.

Índice general

Índice de Ilustraciones

Índice de tablas

1. Introducción

A pesar de todos los motores de búsqueda y otros sistemas de búsqueda de información en internet, todavía hay algunas preguntas que la gente encuentra difíciles de responder. Por eso, cada vez más personas buscan las respuestas a sus preguntas en plataformas de preguntas y respuestas en línea, llamados foros. Estas plataformas permiten a personas de todo el mundo hacer preguntas y obtener respuestas de otras personas que tienen el mismo interés o experiencia. Algunos foros famosos son Stack Overflow o forococoches.

En este contexto surgió la tarea de Community Question Answering (CQA), en particular, en el SemEval de 2015 [1]. Esta tiene características diferentes a las de las tareas genéricas de un sistema de Question Answering (QA) general. Por ejemplo, en este caso la respuesta se extrae de foros que están escritos por la comunidad, por lo que cualquiera puede escribir la respuesta. Esto permite tener una gran cantidad de respuestas, pero, por el contrario, al no tener controlados a los autores puede haber respuestas sin relación con la pregunta.

Por tanto, es necesaria una solución para recorrer la gran cantidad de respuestas discriminando las que no son útiles. La solución a este problema puede beneficiar a los usuarios de manera que se creen herramientas que permitan navegar entre las respuestas de cualquier foro [4].

Dentro de la tarea global de cQA se pueden distinguir varias subtareas como recuperar mensajes relevantes dada una pregunta, buscar preguntas similares ya publicadas, decidir si un comentario es cierto o falso, etc [1,4]. En este trabajo nos centramos en comprobar la relevancia de las respuestas publicadas a un mensaje con una pregunta inicial.

Para abordar esta tarea se han probado varios enfoques basados en texto e información relativa al contexto de las respuestas. Sin embargo, en los últimos años se han realizado grandes avances en procesamiento del lenguaje natural mediante nuevas técnicas de Deep Learning llamadas Transformers. Estos modelos utilizan sistemas de atención para focalizar la importancia del contexto para cada palabra de la frase o del texto y han supuesto una revolución en multitud de tareas, incluido el QA. Sin embargo, estas técnicas no se han aplicado a la hora de realizar la tarea de CQA y comprobar la similitud semántica entre preguntas y respuestas.

En este trabajo se proponen y evalúan enfoques centrados en determinar la relevancia entre una pregunta y una respuesta dentro del mismo foro mediante Transformers. Concretamente vamos a evaluar si con los enfoques propuestos se puede mejorar a las aproximaciones anteriores propuestas en el marco de distintas tareas del SemEval. También se va a tratar de utilizar dichas aproximaciones sobre el dataset de AmazonQA, ya que este no es un dataset en el que comúnmente se haya evaluado la tarea de CQA pero que está orientado a su aplicación en un contexto real.

1.2 Propuesta y objetivos

El objetivo general de este trabajo es evaluar el uso de modelos basados en Transformers para la recuperación de respuestas en foros. Para abordar este objetivo general, se ha descompuesto en los siguientes objetivos específicos:

- Evaluar el comportamiento de arquitecturas de Transformers basadas en Cross-Encoders para buscar la similitud semántica entre preguntas y respuestas.
- Evaluar el comportamiento de arquitecturas de Transformers basadas en Bi-Encoders para buscar la similitud semántica entre preguntas y respuestas.
- Evaluar el comportamiento de modelos de composicionalidad semántica no supervisados para buscar la similitud semántica entre preguntas y respuestas.
- Comprobar si crear un ensemble mediante técnicas de machine learning, como One Versus All (OVA), permite una mejor clasificación de las respuestas.
- Comprobar si la tarea de similitud semántica y la clasificación son compatibles y mejora los resultados mediante técnicas multitask.
- Comprobar si las propuestas desarrolladas permiten realizar además de la tarea de clasificación de las preguntas, la tarea de ranking sin realizar otro entrenamiento diferente.

1.3 Estructura del documento

El presente trabajo se ha dividido en los siguientes capítulos:

- **Capítulo 2. Preliminares:** En este capítulo se detallan los fundamentos teóricos básicos para poder entender correctamente las diferentes propuestas planteadas a lo largo de este trabajo.
- **Capítulo 3. Estado del arte:** En este capítulo se realiza una revisión del estado del arte, describiendo las técnicas propuestas anteriormente para resolver la tarea.
- **Capítulo 4. Marco Experimental:** En este capítulo se detallan ciertos aspectos de los experimentos como las colecciones y las métricas utilizadas, además de realizar un esquema de las propuestas realizadas.
- **Capítulo 5. Análisis de Resultados:** En este capítulo se ofrecen los resultados obtenidos, analizando los mismos y comparándolos entre si.
- **Capítulo 6. Conclusiones y Líneas Futuras:** En este capítulo se hará una síntesis del trabajo, concluyendo las aportaciones llevadas a cabo por este trabajo junto a las líneas futuras que han surgido.

2. Preeliminares

Para dar solución a este trabajo, se han utilizado técnicas de Machine Learning y de Deep Learning. El Machine learning es el proceso por el cual los ordenadores consiguen "aprender" a partir de los ejemplos que les están disponibles. Grosso modo, aprender es el proceso que convierte la experiencia en conocimiento. La entrada de un algoritmo de aprendizaje son datos de entrenamiento, experiencia representada y la salida es alguna habilidad, que normalmente toma forma de un programa informático que puede realizar alguna tarea.

Pero… ¿Para qué necesitamos machine learning? ¿No se puede hacer un algoritmo que resuelva las cosas y sea mucho más simple? Para solucionar cualquier problema de manera computacional hay que tener en cuenta dos aspectos: La complejidad del problema y la necesidad de adaptabilidad:

- Tareas que son demasiado complicadas de programar: Estas a su vez se dividen en dos.
    - Tareas realizadas por animales/humanos: Hay muchas tareas que los humanos realizamos de manera rutinaria pero que no comprendemos suficientemente bien para programarlas. Por ejemplo, conducir, reconocer el habla y entender imágenes.
    - Tareas que están más allá de las capacidades humanas: Hay otra familia igualmente grande de tareas que se benefician de las técnicas de machine learning, ya que se basan en analizar grandes cantidades de datos que además son complejos: Datos astronómicos, predicción del tiempo o analizar datos genómicos.
- Adaptabilidad. Una limitación de las herramientas programadas es su rigidez. Una vez que el programa ha sido escrito e instalado, permanece igual para siempre. Sin embargo, hay muchas tareas que cambian a lo largo del tiempo o de un usuario a otro. Las herramientas que se basan en machine learning tratan de solucionar este problema, ya que, por naturaleza, son herramientas que basan su comportamiento en el entorno que le rodea. Por ejemplo, los programas que descodifican texto escrito a mano, que se ajustan a la letra de cada usuario, los detectores de spam, que se ajustan a que el spam cambia a lo largo del tiempo, y los programas de reconocimiento del habla.

En nuestro ejemplo, el tratamiento del lenguaje natural para calcular la similitud semántica entre dos frases es una tarea que hacemos sin darnos cuenta pero que no entendemos suficientemente bien como para programarlo satisfactoriamente. Si que ha habido distintos intentos, pero siempre un algoritmo de machine learning ha funcionado mejor. Por otra parte, la variabilidad que tenemos a la hora de meter textos en el algoritmo hace que sea necesario que el algoritmo sea capaz de generalizar. Esta es la mejor capacidad del machine learning.

2.1 Tipos de aprendizaje y de tareas

Por otra parte, aprender es un terreno muy amplio. De hecho, el campo del machine learning se ha desglosado en diferentes subtipos dependiendo de la manera en la que la máquina aprende. En nuestro caso, solo nos interesa entender una clasificación de todas estas, aunque es bueno saber que existen más.

En este trabajo se van a explotar algoritmos tanto de aprendizaje **supervisado** como **no supervisado.** El aprendizaje define una interacción entre el algoritmo que aprende y el entorno, así que se puede clasificar

el aprendizaje según esta relación. Para ejemplificar esta clasificación, vamos a usar como ejemplos el sistema de detección de spam y un sistema de detección de anomalías. Para la detección de Spam, consideramos un escenario en el que el algoritmo recibe correo etiquetado como spam o no spam. Basándose en este etiquetado, el algoritmo tiene que obtener una regla para etiquetar el siguiente correo. Por otra parte, para la tarea de detección de anomalías, el algoritmo tiene como entrenamiento un montón de correos electrónicos, sin etiquetas, y este tiene que detectar los mensajes que no son normales.

Cambiando de punto de vista, viendo el aprendizaje como un proceso por el cual usamos experiencia para obtener una habilidad, el aprendizaje supervisado describe un escenario donde la experiencia, cada ejemplo de entrenamiento, contiene información significativa (en nuestro ejemplo, Spam o no spam) que falta en los ejemplos de test, donde la habilidad aprendida tiene que ser demostrada. En resumen, el aprendizaje supervisado trata de predecir la información que falta para los datos de test. Podemos pensar en este tipo de aprendizaje como un profesor que supervisa el aprendizaje del alumno dándole información extra (las etiquetas).

Por otra parte, en el aprendizaje no supervisado no hay distinción entre los datos de entrenamiento y los de test. El aprendiz procesa los datos de entrada con el objetivo de obtener algún resumen, o alguna versión comprimida de los datos. Convertir unos datos en subconjuntos de objetos similares es un ejemplo típico de este aprendizaje.

En este trabajo se van a utilizar algoritmos para la similitud semántica que están preentrenados con otros datos y se usan en este trabajo o que simplemente no necesitan ser entrenados. Estos son clases de machine learning auto-supervisado, es decir, es un algoritmo no-supervisado si solo observamos el trabajo actual pero ha sido preentrenado con una tarea de enmascaramiento. Sin embargo, cuando se usan los datos de train para realizar un fine-tuning de los algoritmos preentrenados estaríamos hablando de aprendizaje supervisado.

Además de los tipos de aprendizaje, hay diferentes tareas que ejecutan los algoritmos de machine learning. Dos tareas típicas del machine learning son las siguientes:

- La **clasificación** que trata de resolver el problema de decidir si el conjunto de datos que se le pasan al modelo como entrada pertenece a un tipo discreto concreto o a otro. En este caso se podría encontrar el caso del detector de Spam del que hemos hablado antes.
- La **regresión** que trata de predecir un valor continuo a partir de los datos de entrada. Dentro de este grupo se podría encontrar un modelo que, a partir de la ciudad, el tamaño, el número de ventanas y el número de baños de una casa estimase su precio.

En este trabajo, la tarea que se trata de resolver es la clasificación. Esta se puede resolver de manera no supervisada calculando la similitud semántica con métodos no supervisados o puede ser una clasificación supervisada, utilizando la similitud semántica de algún método supervisado o utilizando directamente un modelo supervisado de clasificación.

2.2 Técnicas de machine learning: One Versus All (OVA) y multitask

Dentro del machine learning, existen métodos para mejorar el rendimiento de ciertos algoritmos o para dotarles de características de las que carecen. Por ejemplo, las máquinas de soporte vectorial son solamente capaz de hacer clasificaciones entre dos clases y mediante un ensemble se puede conseguir que hagan clasificaciones con más de dos clases. Estos métodos se consideran ensembles.

En nuestro caso, se utiliza un ensemble para mejorar algunos resultados y también para convertir la similitud semántica que solo se aplica entre dos pares de frases, a multiclase. Se llama One versus All (OVA). La idea en este caso consiste en entrenar un clasificador por cada clase, de tal forma que cada clasificador sepa si la instancia es mas de esa clase o del resto de clases. Para agregar los resultados de los múltiples clasificadores, se coge la mayor confianza de la predicción de que pertenezca a una clase.

Por otro lado, en este trabajo también se usa otro tipo de estrategia de machine learning llamada multitask. En este caso, el algoritmo de machine learning trata de resolver varias tareas al mismo tiempo, aprovechando las similitudes y disonancias entre dichas tareas. Esto permite que en algunos casos la eficiencia del aprendizaje sea mayor a la vez que mejora la precisión frente a modelos que solo se especifican en una tarea.

En este trabajo, se va a usar multiclass ya que la tarea de clasificación entre pregunta y respuesta puede ser vista como una tarea de similitud semántica o como una clasificación. Dichas tareas están relacionadas por lo que se pueden agregar en un mismo modelo, intentando que este aprenda las dos.

2.3 Deep Learning

En este apartado trataremos un tipo especial de machine learning que se llama Deep Learning. Para explicar que es el Deep Learning, primero hay que entender cómo funciona el cerebro humano.

En 1888, Ramón y Cajal demuestra que el sistema nervioso está compuesto por una red de células individuales que están interconectadas entre sí. Esto es un paso muy grande a la hora de empezar a entender nuestro cerebro. También demuestra que la información fluye desde las dendritas hasta el axón atravesando el soma. Esto se puede observar en la ilustración 1.

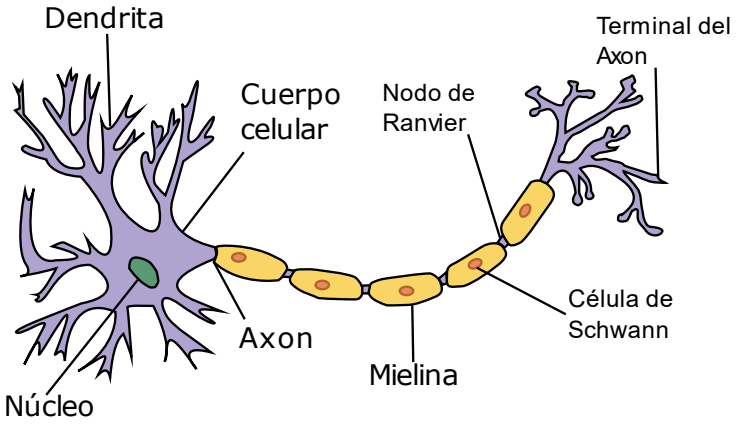

Ilustración 1: Neurona. Fuente: https://ast.wikipedia.org/wiki/Neurona

Nuestro cerebro está compuesto por 10^8 neuronas. Cada neurona se comporta de manera individual como un pequeño procesador sencillo. Se puede imaginar que las dendritas podría ser el canal de entrada de la información, el núcleo es el órgano de cómputo y el axón es el canal de salida que envía impulsos a otras neuronas.

El cerebro de un ser vivo se modela durante su desarrollo. Sí que hay algunas cualidades que son innatas, pero también hay otras muchas que son adquiridas por la influencia de la información que entra por sus sensores. Esta información transforma el sistema nervioso mediante creación de nuevas conexiones, ruptura de otras, mediante el modelado de las intensidades en las que se pasa la información de una neurona a otra o mediante la muerte y creación de nuevas neuronas.

Este simple funcionamiento permite que nuestro cerebro tenga una cantidad de cómputo inimaginable. Es por eso, que distintos expertos en computación han tratado de emular esta estructura en un ordenador con el fin de alcanzar una funcionalidad similar.

Para hacer esta simulación, hacen falta 3 conceptos clave:

- **Procesamiento en paralelo:** Cuando vemos una imagen las neuronas no actúan de manera secuencial, sino que actúan todas simultáneamente para procesarla lo más rápido posible.
- **Memoria distribuida:** Nuestro cerebro no guarda la información en un solo punto, sino que los datos están distribuidos por todo nuestro cerebro de manera redundante, para evitar pérdidas de información.
- **Adaptabilidad al entorno:** Nuestro cerebro es capaz de coger ejemplos individuales, que se ganan a través de la experiencia y generalizarlos de manera que podamos aplicar este aprendizaje en otros casos.

La idea de estos científicos fue copiar nuestro cerebro de la manera más exacta que pudieran. Para ello, observaron que nuestro cerebro era un conjunto de neuronas conectadas organizadas por capas. Así pues, un sistema que simulase un cerebro humano tendría que estar construido por un conjunto de neuronas, en este caso artificiales, organizadas en capas con una entrada, que simularía nuestros sentidos, y una salida [86].

2.3.1 Perceptrón

Estás neuronas artificiales se denominaron perceptrones. Un Perceptrón es un dispositivo de computación con umbral U y n entradas reales $X_1, \dots, X_n$ a través de arcos con pesos $W_1, \dots, W_n$ y que tiene salida 1 cuando $\sum_i w_i x_i \geq U$ y 0 en caso contrario. Todo esto se muestra en la ilustración 2.

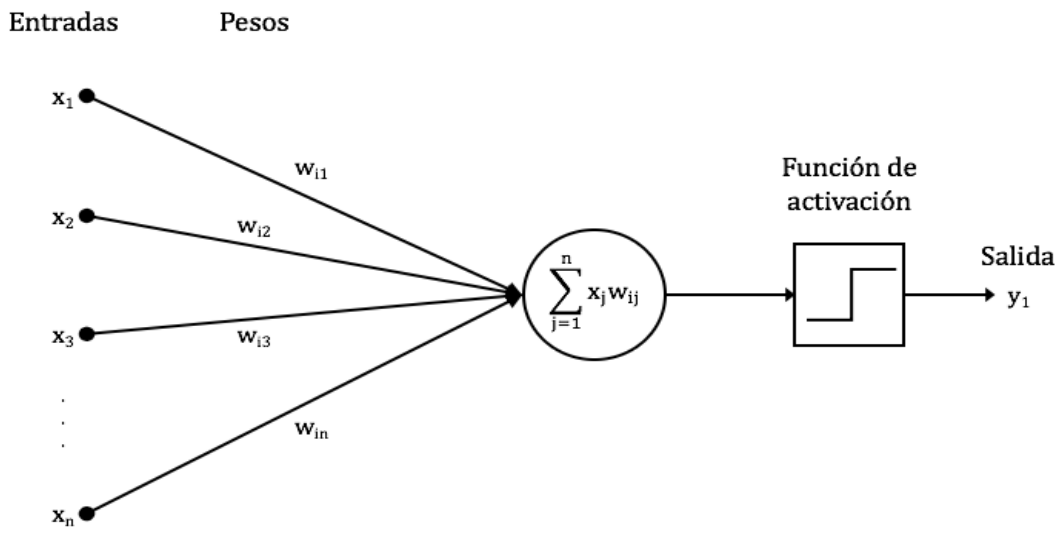

Ilustración 2: Neurona artificial. Fuente: https://www.avansis.es/inteligencia-artificial/que-son-las-redes-de-neuronas-artificiales-parte-i/

El origen de las entradas es irrelevante. Puede venir de otros perceptrones o de cualquier clase de unidad de computación. Geométricamente, es fácil identificar cual es la forma que dibuja un Perceptrón en el espacio. Crea un hiperplano que separa los puntos cuya agregación no supera el umbral de los que sí lo superan.

La salida de la neurona responde a una función de activación que depende de la agregación de todos los valores de entrada y del valor del umbral. Normalmente esta función devuelve un 0 o un 1 dependiendo de si el valor de esta agregación es positiva o negativa, pero puede dar diferentes valores. Estas funciones de activación se suelen utilizar en las redes neuronales multicapa de las que ya hablaremos posteriormente.

Un Perceptrón simple como este puede separar dos conjuntos de puntos linealmente separables. Dos conjuntos de puntos A y B en un espacio n-dimensional son linealmente separables si existen n+1 números reales $w_1, \dots, w_{n+1}$ tales que cada punto $(x_1, \dots, x_n)$ A satisface $\sum_i w_i x_i \geq w_{n+1}$ y que cada punto $(x_1, \dots, x_n)$ B satisface que $\sum_i w_i x_i < w_{n+1}$.

Si las clases a separar son linealmente separables, el algoritmo del Perceptrón converge a una solución correcta en un número finito de pasos para cualquier elección inicial de pesos.

Sin embargo, algo de lo que no tenemos que olvidarnos es que esto es machine learning, luego detrás tiene que haber un algoritmo de entrenamiento. Este algoritmo es supervisado y usa n vectores que se pueden clasificar en dos conjuntos P y N en el espacio extendido n+1-dimensional. Se considera que el umbral es una entrada de tendencia con valor fijo 1, ya que $\sum_i w_i x_i < w_{n+1}$ equivale a $\sum_i w_i x_i - w_{n+1} < 0$ [87].

El algoritmo de entrenamiento es el mostrado en la ilustración 3.

Algorithm 1: PerceptronFit(X,P,N)

Entradas: X - vectores de entrada con al menos un elemento, P - indices de los vectores x cuya salida tiene que ser 1, N - indices de los vectores x cuya salida tiene que ser 0

Salidas: w - vector que separa los dos conjuntos de vectores

```
1 w ← random(0, 1, len(X_0))
2 while queden vectores mal clasificados do
3     r ← randint(0, len(X))
4     aux ← x_r
5     if x ∈ P ∧ w * x < 0 then
          /* Hay un error                    */
6         w ← w + aux
7     if x ∈ N ∧ w * x ≥ 0 then
          /* Hay un error                    */
8         w ← w − aux
9 return w
```

Ilustración 3:Algoritmo de entrenamiento del Perceptrón.

2.3.2 Adaline

Por otro lado, a la vez que surgía el perceptrón, también aparecía el adaline y su regla de aprendizaje llamada *least mean square*.

El adaline es prácticamente idéntico al perceptrón, excepto en su función de transferencia, la cual es una función de tipo lineal en vez de hacer un corte fuerte como sí hace el perceptrón. El adaline presenta el mismo problema que el perceptrón, no puede resolver problemas cuyos datos no sean linealmente separables. Sin embargo, el algoritmo de aprendizaje de este tipo de neuronas es mucho más potente que el de los perceptrones ya que se basa en el error cuadrático medio. De hecho, esta regla sirvió de inspiración para más algoritmos posteriores.

El termino Adaline es una sigla, aunque su significado ha cambiado ligeramente desde los años sesenta cuando decayó el estudio de las redes neuronales. Inicialmente se llamaba *ADAptive Linear NEuron* (Neurona lineal adaptativa), para pasar después a ser *adaptative linear element* (Elemento lineal adaptativo). Este cambio se produjo ya que este tipo de neuronas son dispositivos que constan únicamente de un único elemento de procesamiento y por tanto no es como tal una red neuronal.

Esta unidad de procesamiento de lo que se encarga es de realizar las sumas y los productos entre los valores de entrada y los pesos, y aplica una función de salida, la cual es una función lineal en este caso, para obtener un único valor de salida.

El adaline es **adaptativo** ya que existe un procedimiento bien definido para modificar los pesos con el objetivo de hacer posible que el dispositivo consiga obtener la salida esperada dada la entrada correspondiente. Estos valores dependerán de la función de salida que tenga la neurona. No será lo mismo si tiene la función identidad, en el que los valores de la entrada son sumados y multiplicados por los pesos y expulsados por la neurona, que si tiene otra función más complicada. En redes neuronales más complejas, la función de salida o de activación será clave para la correcta adaptación de las redes.

Por otra parte, el adaline es **lineal** porque la salida es una función lineal sencilla de los valores de la entrada.

Por último, es una **neurona** tan solo en el sentido del elemento de procesamiento. Como ya hemos explicado, quizás sería más correcto llamarlo elemento lineal, evitando por completo la definición de neurona [87].

2.3.3 Redes neuronales multicapa y algoritmo backpropagation

Lo interesante de estos dos modelos previamente expuestos es que no permiten más que el tratamiento de datos lineales. Es por eso, que se pensó cómo mejorar el funcionamiento de las neuronas para poder tratar la no linealidad, que se encuentra en la mayor parte de lugares de la naturaleza. Para ello contábamos con dos estrategias:

- **El aumento del número de neuronas:** Al aumentar el número de unidades de procesamiento, a pesar de que cada una de ellas tuviese individualmente un funcionamiento lineal, conseguimos que el conjunto tenga un funcionamiento no lineal.
- **Introducir funciones de activación no lineales:** En el modelo adaline se contaba con funciones de activación lineales. Esto como es obvio, provocaba que la salida fuese lineal. Sin embargo, en este caso contamos con más funciones que podemos incluir como funciones de activación y que no llevan consigo este lastre.

Mediante estas redes, a pesar de que constan de una morfología diferente a las que se encuentran en el interior de nuestro cerebro, conseguimos las siguientes características:

- **Auto-organización y adaptabilidad:** Utilizan algoritmos de aprendizaje adaptativo y auto-organización, por lo que ofrecen mejores posibilidades de procesado robusto y adaptativo.
- **Procesado no Lineal:** Aumenta la capacidad de la red para aproximar funciones, clasificar patrones y aumenta su inmunidad frente al ruido.
- **Procesado paralelo:** Normalmente se usa un gran número de nodos de procesado, con alto nivel de interconectividad.

La morfología de las redes depende de los datos a tratar. No usaremos el mismo tipo de redes en el caso de usar como datos de entrada imágenes que si usamos series temporales [88][89].

Como ya hemos dicho en el caso del adaline, la función que se usa en este caso para mejorar el rendimiento de la red neuronal y hacerla “aprender”, es minimizar el error cuadrático medio entre los valores esperados y los valores de salida. Esta etapa se llama **fase de entrenamiento** y se basa en

determinar los pesos que definen el modelo de red neuronal de manera iterativa. Para entrenar la red hace falta un algoritmo especial que se llama *backpropagation.*

Este algoritmo surgió ante la necesidad de un algoritmo eficiente que nos permita adaptar todos los pesos de una red multicapa y no solo los de la capa de salida, como en el caso del adaline o del perceptrón simple. En estos casos simples, era fácil saber cuáles eran los pesos que se tenían que poner en la capa de salida, pero al aumentar el número de capas y el número de neuronas, se hace mucho menos evidente cuales son las características que hay que poner para que la red neuronal funcione correctamente.

Entonces si realmente no sabemos cuáles son las características que tienen que aprender las capas ocultas de la red… ¿Cómo entrenamos estas redes? La respuesta es que hay que cambiar de prisma nuestra observación. En vez de fijarnos en los cambios de los pesos, nos fijaremos en los cambios de las salidas. Nuestro objetivo será acercar lo más posible las salidas de nuestra red a las salidas esperadas. Esta estrategia funcionará en la mayoría de los casos, solo que empezará a ser más problemática en los problemas no convexos.

Para explicar este algoritmo, usaremos el filtro lineal, aunque es aplicable a cualquier filtro interno de las neuronas, aunque elegiremos éste porque es el más "visual" y el más sencillo de ejemplificar.

$$y = \sum_i w_i x_i = W^T X \quad (1)$$

Nuestro objetivo es minimizar la suma de los errores de la y real contra la y predicha por la red en los ejemplos de entrenamiento. Se podría resolver el problema analíticamente, pero la solución sería difícil de generalizar además de ser poco eficiente. Es por tanto que diseñaremos un algoritmo iterativo. Para explicar el algoritmo iterativo vamos a poner un ejemplo de la vida real.

Imaginemos que todos los días desayunamos en la cafetería de la universidad. Tomamos para desayunar café, zumo y tostadas, pero sólo nos pasan la factura al final de la semana, sin decirnos cuánto cuesta cada producto. ¿Seríamos capaces de saber cuánto vale cada uno de los productos? La respuesta es que sí, tras varias semanas.

Lo primero que tendríamos que hacer es empezar con una estimación, que puede ser aleatoria, de los precios, y posteriormente irlos ajustando de tal manera que encajen con los importes facturados

Cada semana, el total nos impone una restricción lineal sobre los precios de las cantidades consumidas:

$$total = w_{cafe} x_{cafe} + w_{zumo} x_{zumo} + w_{tostada} x_{tostada}$$

Los precios son los pesos $w = (w_{cafe}, w_{zumo}, w_{tostada})$ y las entradas corresponderían a las cantidades consumidas de cada producto.

La primera semana, la cuenta nos ha costado 8,50 euros. Lo que nos deja la situación de la ilustración 4, estos serían los pesos reales que debería tener nuestra red para predecir la cuenta.

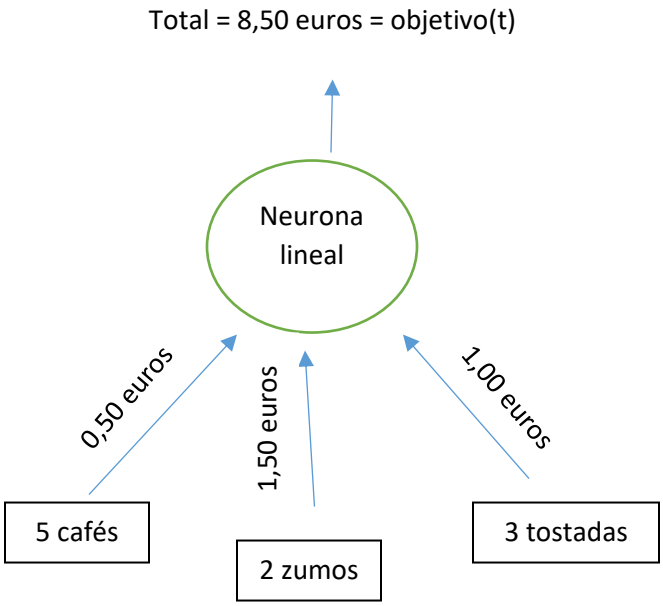

Ilustración 4: Precios que estimamos la primera semana de universidad.

Sin embargo, en la primera semana los precios que habíamos estimado eran 0.50 euros por cada producto. Lo que nos deja con un error residual de 3 euros y medio. Esto se puede observar en la ilustración 5. Para corregir los pesos de nuestra calculadora de facturas usamos la regla delta:

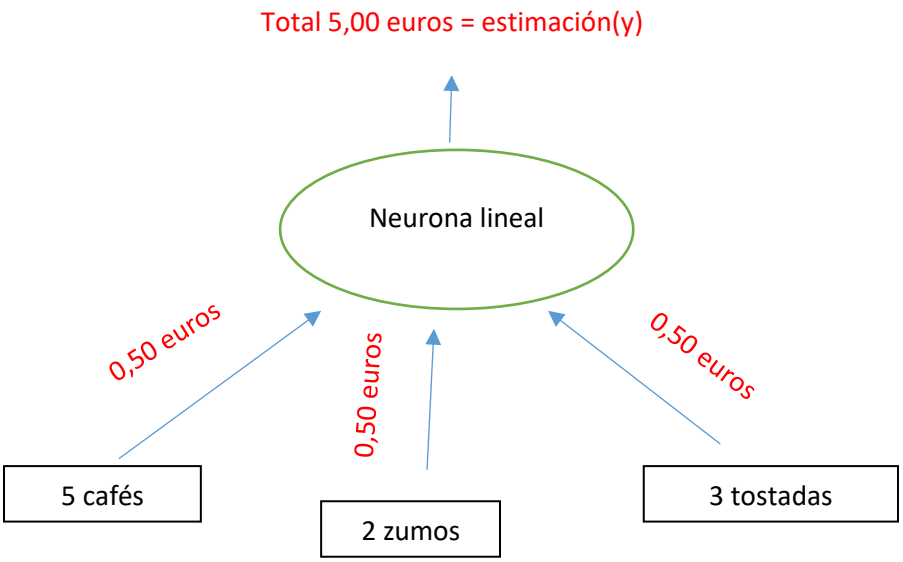

Ilustración 5: Precios que estimamos la segunda semana de universidad.

$$\Delta w_i = \eta x_i (t - y) \quad (2)$$

η es la tasa de aprendizaje utilizada.

Entonces para ajustar los pesos de nuestro calculador de facturas, usaremos la regla delta. Como el error residual es 3,50 euros (esto es la parte de la regla delta $(t - y)$) y siendo la tasa de aprendizaje $\eta = \frac{1}{35}$, los nuevos pesos son los que se muestran en la ilustración 6.

De esta manera, de manera iterativa y tras una serie finita de pasos, se llegaría a unos pesos que se corresponderían con los precios originales de los alimentos.

Si escribimos esto de una manera más formal nos queda:

- $Error = \frac{1}{2}\sum_{n \in training}(t^n - y^n)^2 \quad (3)$
- Derivada del error: $\frac{\partial E}{\partial w_i} = \frac{1}{2}\sum_n \frac{\partial y^n dE^n}{\partial w_i dy^n} = -\sum_n x_i^n (t^n - y^n) \quad (4)$
- Ajuste de los pesos en proporción al error:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} = \eta \sum_n x_i^n (t^n - y^n) \quad (5)$$

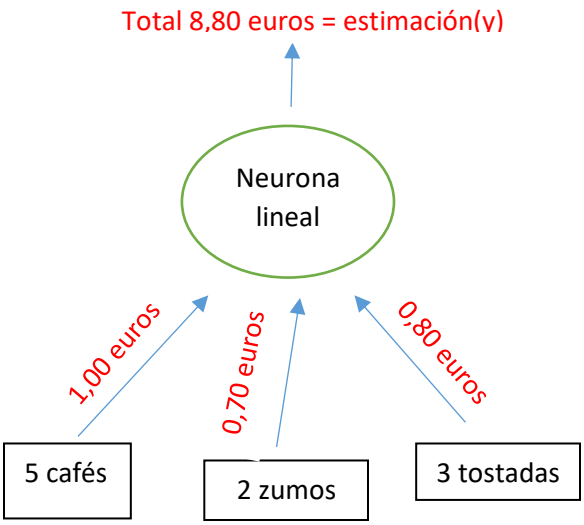

Ilustración 6: Precios que estimamos la tercera semana de la universidad.

Con estas fórmulas, como hemos ejemplificado anteriormente, nos iremos acercando poco a poco, iteración tras iteración a la mejor solución posible. La tasa de aprendizaje tiene que ser pequeña para que no nos pasemos el punto mínimo de la función que se puede observar en la Ilustración 7 [90].

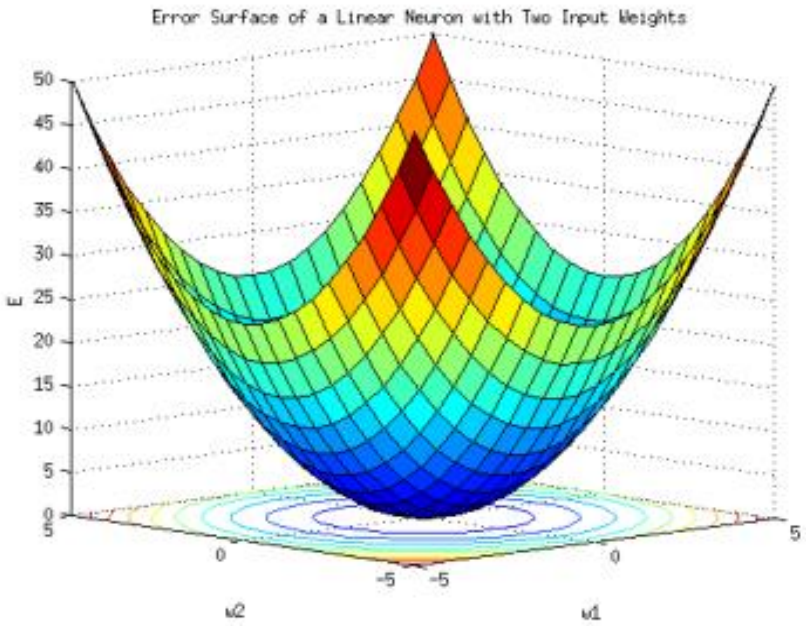

Ilustración 7: Descenso por gradiente del error. Fuente: Fernando Berzal, Backpropagation.

Después de que la red "aprenda" o se entrene mediante el algoritmo que acabamos de explicar, hay que hacer **una fase de prueba**. Esta fase hay que hacerla ya que en ocasiones el modelo se ajusta demasiado a las peculiaridades presentes en los patrones de entrenamiento, perdiendo su habilidad de generalizar el aprendizaje a casos nuevos.

Para realizar esta separación en dos fases, hay que dividir el conjunto de los datos en dos datasets, uno de train y uno de test. Con el dataset de train se realiza la fase de entrenamiento y con el dataset de test la fase de prueba. En ocasiones, el dataset de train se divide a su vez en dos, en train y en validación. El dataset de validación sirve para ajustar los híper-parámetros.

Normalmente, la fase de prueba consiste en probar a predecir o clasificar con el dataset de test y sacar una métrica, que nos de información sobre la desviación de las predicciones sobre la solución teórica, y, que simboliza la capacidad de generalización de la red. Sin embargo, en nuestro caso la métrica no nos sirve para testear la red porque lo que estamos buscando son outliers que no están etiquetados.

2.3.4 Funciones de base y activación

Una red neuronal se puede clasificar por la función de base y de activación que usa. Cada neurona suministra un valor y_j a su salida. Este valor se propaga a través de la red mediante conexiones unidireccionales hacia otros nodos de la red. Asociada a cada conexión hay un peso sináptico denominado $\{w_{ij}\}$, que determina el efecto de la neurona j-ésima sobre la neuronal i-ésima.

La función base es la que se encarga de tratar todas las entradas provenientes de otras neuronas junto con umbral θ_i obteniendo u_i. La salida final y_i se obtiene aplicando la función de activación sobre u_i.

La función base suele ser una de dos:

- **Una función lineal de tipo hiperplano**, en la que el valor de red es una combinación lineal de las entradas,

$$u_i(w,x) = \sum_{j=1}^{n} w_{ij}x_j \quad (6)$$

- **Una función radial de tipo hiperesférico,** que es una función de base de segundo orden no lineal,

$$u_i(w,x) = \sqrt{\sum_{j=1}^{n}(x_j - w_{ij})^2} \quad (7)$$

Por otra parte, como ya hemos dicho la función de activación transforma este valor que expulsa la función de base, para añadir no linealidad a la salida. Las funciones de activación más comunes son las siguientes:

- Función sigmoide

$$f(u_i) = \frac{1}{1 + e^{-u_i}} \quad (8)$$

- Función gaussiana

$$f(u_i) = c * e^{-{u_i}^2} \quad (9)$$

2.3.5 Funciones de loss o de pérdida

Como ya se ha explicado en la sección 4.2.2, cualquier modelo de Deep Learning utiliza la regla delta para ejercer el algoritmo de backpropagation y hacer aprender a la red. Sin embargo, necesita de una función de error que le diga cuales son las características del modelo que tiene que optimizar, es decir, cual es el error que tiene que minimizar. Estas funciones se ejecutan al final de la red en cada iteración del aprendizaje, actualizando de esta manera los pesos internos.

Dentro del Deep Learning, hay una gran cantidad de funciones de error que se pueden utilizar. Las más comunes son las siguientes:

- **El error absoluto medio** calcula la media de las diferencias absolutas entre el valor predicho y el valor real. Se hace el valor absoluto porque si no los errores positivos podrían contrarrestar a los errores negativos. Se expresa como:

$$\text{loss}(\text{y}, \hat{y}) = |\hat{y} - y| \quad (10)$$

  Esta función se usa en tareas de regresión con outliers, ya que se considera más robusta a los mismos.
- **El error cuadrático medio** calcula la media de las diferencias cuadradas entre el valor predicho y el valor real. Esto hace que los errores grandes se cometan menos y se cometan más errores pequeños. Es la función de loss por defecto en regresión y se define como:

$$loss(y, \hat{y}) = (y - \hat{y})^2 \quad (11)$$

- **La función de loss de cross-entropy** calcula la diferencia entre C distribuciones de probabilidad para un conjunto de apariciones o de variables aleatorias. Se utiliza para elaborar una puntuación que resume la diferencia media entre los valores predichos y los valores reales. Para mejorar la precisión del modelo, hay que intentar minimizar la puntuación.La puntuación de entropía cruzada está entre 0 y 1, y un valor perfecto es 0. Este loss penaliza mucho estar muy seguro de una predicción y estar equivocado. Se define como:

$$loss(x, clase) = -log\left(\frac{\exp(x[clase])}{\sum_j \exp(x[j])}\right) \quad (12)$$

Se usa para crear modelos de clasificación y será la que se emplee en los modelos de clasificación de este TFM.

- **El loss de la similitud del coseno** se utiliza para medir si dos tensores son similares o diferentes y se utiliza para aprendizaje semi supervisado y para aprender embeddings no lineales. La función de loss se escribe para cada instancia como:

$$loss(y, \hat{y}) = \begin{cases} 1 - \cos(y, \hat{y}), & si\ y = 1 \\ max(0, \cos(y, \hat{y}) - margin), & si\ y = -1 \end{cases} \quad (13)$$

Esta función representa la función de loss utilizado en este trabajo para aprender los embeddings no lineales de pregunta y de respuesta.

2.3.6 Redes neuronales recurrentes

Estas redes son especiales para tratar series numéricas. Para optimizar este propósito se disponen de distintas técnicas:

- Presentan retro-alimentación, esto significa que la salida de una neurona se usa como entrada de sí misma o como entrada de otra neurona que está conectada a sí misma.
- La salida de la neurona se calcula usando valores de entrada y salida obtenidos en tiempos anteriores. Es decir, de predicciones pasadas.

Las diferencias de estas redes con respecto a las redes neuronales habituales es que estas segundas aceptan una entrada de tamaño fijo (por ejemplo, una imagen) y producen un vector de salida de tamaño fijo (por ejemplo, probabilidades de las distintas clases), mientras que las redes recurrentes operan sobre secuencias de vectores que no tienen tamaño fijo. La transformación recurrente es fija y se puede aplicar tantas veces como sea necesario.[91]

Además, otra cosa que tienen de especial las redes neuronales recurrentes es que cuentan con un estado, y para hacer la siguiente predicción usan el estado anterior. Para calcular, el estado de la neurona recurrente se usa la siguiente formula:

$$h_t = f_W(h_{t-1}, x_t) \ (14)$$

Donde h_t es el estado actual, h_{t-1} el estado anterior, x_t es la entrada en el momento actual y f_w es una función cualquiera. Una función muy usada para esto es la tangente hiperbólica.

Por tanto, las redes neuronales recurrentes (RNNs) se podrían definir como unas redes neuronales especiales que aceptan como entrada un vector x, que tienen internamente un vector de estados h, y que

combinan x y h mediante una función (fija pero cuyos parámetros se van aprendiendo) y que producen como salida un vector y [92].

Lo más importante que hay que ver en esta definición es que la salida no solo está influenciada por la entrada x, sino por toda la historia de las entradas que hubo en el pasado y que afectan al estado h.

Este tipo de redes se pueden utilizar, por ejemplo, para predecir la siguiente letra de una palabra. Esto se puede observar en la Ilustración 8.

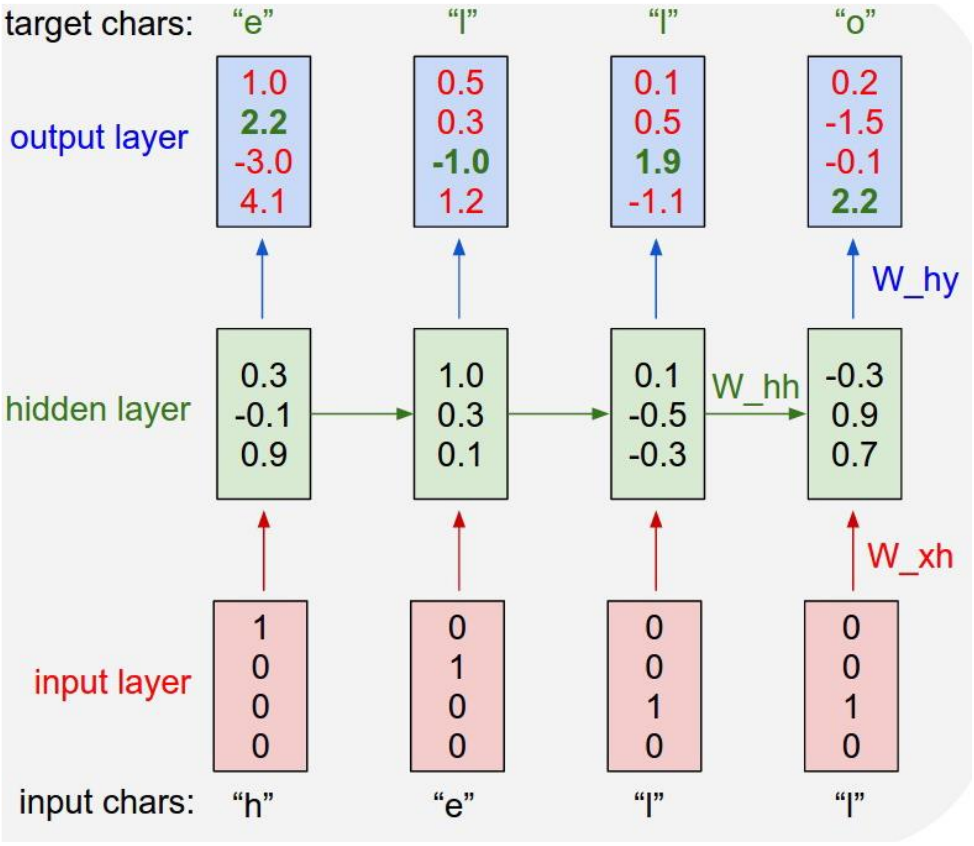

Ilustración 8: Predicción de una red neuronal recurrente.

Para inicializar una red recurrente, hay que definir 3 matrices W de manera aleatoria, que afectan a la entrada, al estado oculto y a la salida. El estado oculto h es inicializado con el vector nulo, ya que el estado es nulo al no tener ninguna historia la red.

Para predecir el siguiente valor, primero se actualiza el vector de estados con el estado anterior y la entrada actual y después se calcula la salida actual.

Para hacer entrenar la red, se usa el mismo algoritmo que en el caso de las redes neuronales multicapas normales: El algoritmo *backpropagation*.

Sin embargo, este tipo de redes neuronales recurrentes convencionales tienen un problema. ¿Qué pasaría si un ejemplo en particular es muy relevante, pero ha pasado hace muchas iteraciones? La red va olvidando su estado de la historia más lejana.

Para entenderlo mejor voy a poner un ejemplo. Imaginemos un sistema de texto predictivo. Si tratamos de predecir la siguiente palabra en una frase como “las nubes están en el”, es obvio que la siguiente palabra va a ser “cielo”. Si estamos en un caso así las redes neuronales recurrentes convencionales no necesitan más ayuda para adivinar la siguiente palabra.

Pero imaginemos una biografía, un texto más largo. En una de las primeras frases pone “Yo crecí en Francia”, y hacia la mitad de la biografía escribimos “Hablo”. Una red neuronal recurrente convencional sabrá que la siguiente palabra tendrá que ver sobre el lenguaje, pero si tiene que decidir qué lenguaje necesitamos el contexto de que crecí en Francia, de mucho más atrás, al ser este hueco tan grande, las redes neuronales recurrentes no son capaces de recordar esa información.

Con el desarrollo del procesamiento del lenguaje natural con Deep Learning, se han desarrollado redes que permiten tener en cuenta el contexto independientemente de la distancia que haya entre las palabras en el texto. Para esto se usó la **atención**. Al principio, este mecanismo fue aplicado directamente a las redes recurrentes para solucionar el problema de la distancia entre palabras. Sin embargo, esto dio a lugar a los Transformers, que son los modelos utilizados en este trabajo y que son los que mejor métricas dan en técnicas de procesamiento de lenguaje natural.

2.3.7 Transformers y configuraciones

Un Transformer es un modelo de Deep Learning que adopta el mecanismo de la atención para calcular la importancia de cada parte de la entrada. Lo bueno de este tipo de estructuras es que permite identificar el contexto que tiene cada parte de la entrada. Además, a diferencia de las redes neuronales recurrentes permite analizar los elementos de la oración en desorden, lo que hace que la velocidad de entrenamiento sea mucho más rápida.

En el caso del procesamiento del lenguaje, permite tener en cuenta el contexto de cada una de las palabras a la hora de codificarlas. Esto hace que sean ideales para analizar la semántica de cualquier tipo de texto. En nuestro caso, con estos Transformers podemos modelar el contexto de las preguntas y de las respuestas para poder encontrar la similitud semántica entre ellas o por lo menos clasificarlas como mas o menos similares de manera satisfactoria.

Antes de los Transformers, para el análisis de textos se utilizaban redes neuronales recurrentes, como se ha explicado en la sección anterior. El procesamiento secuencial de este tipo de redes permitía tener un vector de estado en cada token que representaba todos los tokens anteriores. De esta manera, se pretendía mantener un vector de estado que se actualizaba a cada token nuevo para poder guardar todo el contexto de la frase. Sin embargo, este tipo de redes cuenta con el problema de que este vector de estado no conseguía reflejar toda la información de la frase porque las palabras más viejas o que habías aparecido antes iban perdiendo relevancia en dicho vector.

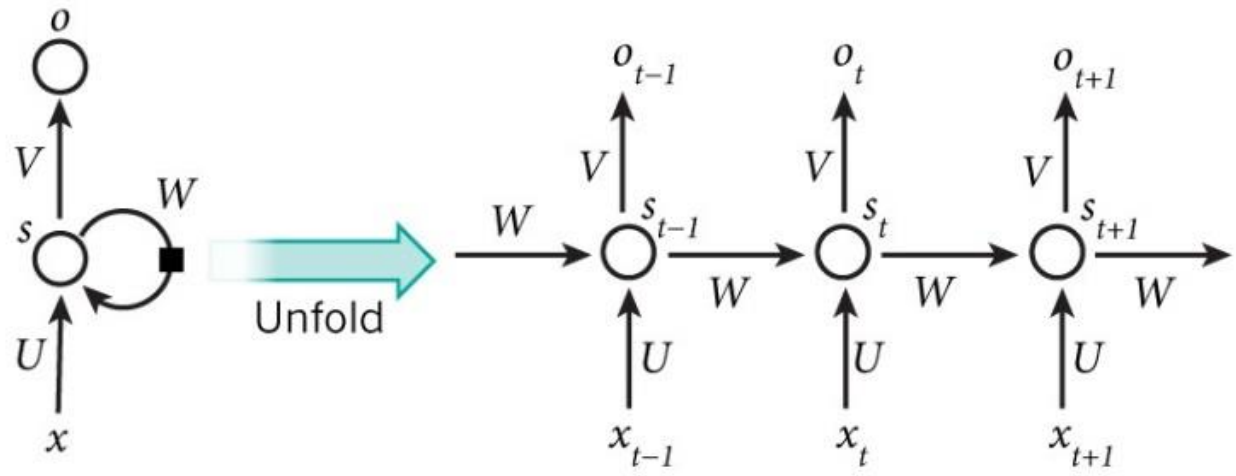

Ilustración 9: Red neuronal recurrente. Fuente: https://ia-latam.com/2019/02/06/entendiendo-las-redes-neuronales-de-la-neurona-a-rnn-cnn-y-deep-learning/

Para solucionar este problema se inventó el mecanismo de atención. Dicho mecanismo permite al modelo extraer información del estado de cualquier punto de la secuencia. Esto permite ponderar el peso de cada uno de los tokens del texto, ponderándolos en función de la importancia para el token actual y no en función de la distancia con este. En este trabajo no se va a entrar en la estructura de los mecanismos de atención.

Otra cosa interesante de los Transformers es la necesidad de un codificador de la posición. A diferencia de las redes neuronales recurrentes en las que un token va después de otro, en el caso de los Transformers la posición de las palabras no está implícita en la estructura del modelo. Esto hace que hagan falta codificadores que conviertan los tokens de manera que los tokens más cercanos en el texto estén más cerca en la codificación y los más lejanos, más lejos. Sobre este tema hay muchos papers que crean distintos tipos de codificadores para distintas tareas por ejemplo el artículo de Shiv et al. [93] en el que crean un codificador posicional en forma de árbol para diferentes tipos de datos o en el artículo de Ke et al. [94] en el que también explican diferentes codificadores para preentrenar Transformers.

Hablando del preentrenamiento de los Transformers, la gran cantidad de capas y por tanto pesos de los Transformers hace que sea necesario una gran potencia computacional para entrenarlos. Es por esto, que normalmente se usan Transformers preentrenados para abordar problemas de procesamiento del lenguaje natural. A estos Transformers preentrenados se les hace un fine tuning, un entrenamiento de pocas épocas con los datos de train del problema específico, de manera que se adaptan perfectamente al problema que se lleva a cabo.

Este es el caso de los Transformers usados en este trabajo, se han usado modelos ya preentrenados para las tareas que nos interesa solucionar. Se han usado los siguientes:

- Distilbert [95] entrenado para calcular la similitud semántica con el dataset de similitud semántica STS.
- MiniLM-L6 [96] entrenado con multitud de datasets para calcular la similitud semántica.

- Distilbert [97] entrenado para encontrar las respuestas más relevantes a una determinada pregunta con el dataset MS MARCO [98].

Lo que tienen en común estos modelos es la destilación. La destilación es el proceso de transferencia de conocimiento entre un modelo más grande y uno más pequeño sin reducir el valor de las métricas. Aunque el modelo más grande tiene más capacidad de guardar conocimiento, muchas veces esa capacidad está siendo infrautilizada y se puede transmitir a un modelo más pequeño. Esta técnica se usa muy frecuente en el caso de los Transformers debido al gran tamaño de estos. Gontijo et al. explican la distilación sin añadir más datos en su paper de 2017 [99].

Respecto a las configuraciones de los Transformers, en este trabajo se utilizan dos tipos:

- Por un lado, se usa una configuración de Cross-Encoder. La idea en este caso es utilizar un Transformer al que se le pasan las dos frases simultáneamente y el Cross-Encoder devuelve un embedding de las dos frases. Este embedding se pasa por una red lineal que actúa como clasificador. En este caso, se usa un cross entropy loss.
- Por otro lado, se usa una configuración de Bi-Encoder. En este caso, hay un Transformer que codifica la pregunta y otro la respuesta generando dos embeddings. Los pesos y la arquitectura se comparten entre ambos Transformers. Esta configuración se usa mucho en Deep Learning y se denomina red siamesa, porque tiene dos cabezas iguales. Estos embeddings pueden compararse mediante una función de similitud como la similitud del coseno o pueden concatenarse y pasar por una red lineal que haga de clasificador. Si lo que se quiere es obtener dos embeddings

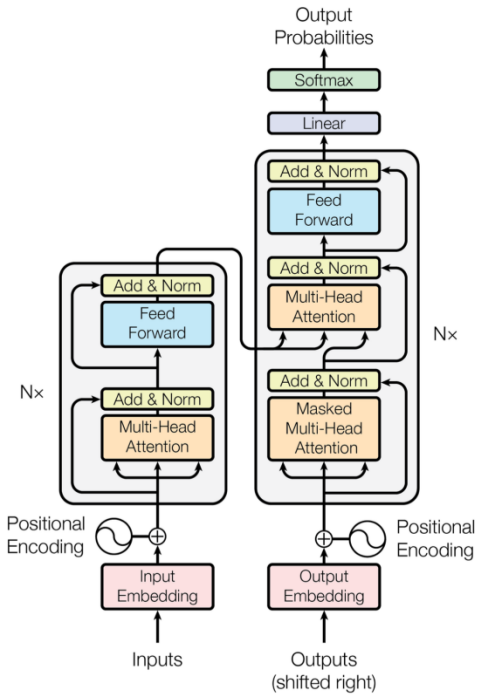

Figure 1: The Transformer - model architecture.

Ilustración 10: Estructura de un Transformer. Fuente: https://arxiv.org/abs/1706.03762

comparables mediante la similitud del coseno se usa el cosine loss. Si por el contrario se quiere utilizar un clasificador se usa el cross entropy loss.

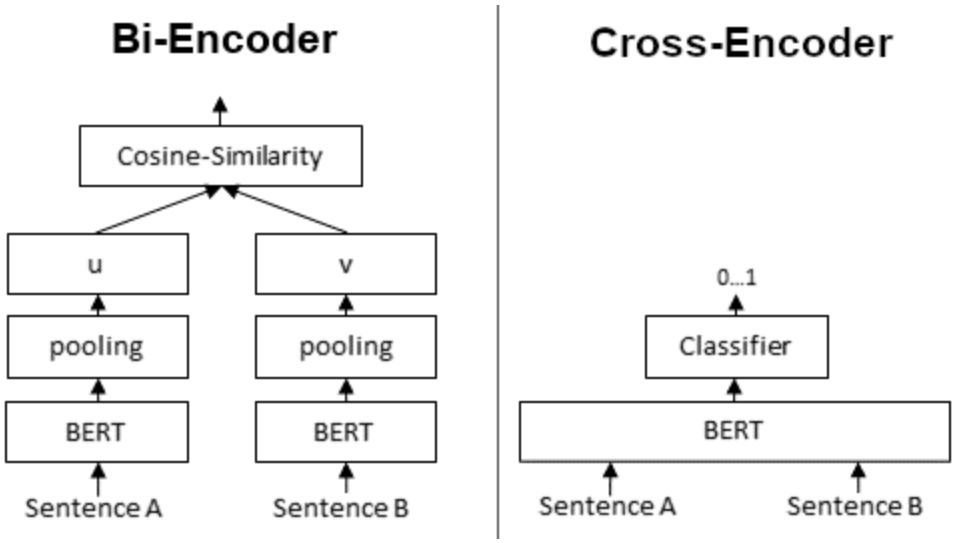

Ilustración 11: Bi-Encoder junto a Cross-Encoder. URL: https://www.sbert.net/

Realmente, la idea de usar dos Transformers en una red siamesa se explica en el artículo de Devlin et al. [100]. Para realizar la clasificación, se concatena el vector de una de las cabezas con el de la otra cabeza y con el valor absoluto de la diferencia de estas. Este es el método que se sigue en este trabajo para realizar la clasificación con los Bi-Encoders.

2.4.8 Mecanismos de atención

Como se ha tratado en secciones anteriores, el mayor problema de las redes recurrentes es que no prestan mucha atención a las palabras lejanas. Para dar con una solución, Luong et al. [101] trataron de crear un método para que la atención no se limitase a las palabras anteriores, sino que se calculase en la propia red. De esta manera, la atención que presta una palabra sería independiente de la distancia a la que se encuentra.

Aunque su primera aplicación fue en traducción automática de textos, actualmente ese mecanismo se ha trasladado a todos los campos del Deep Learning como, por ejemplo, al campo de la visión artificial con papers como este de Tao et al. [102]. En este paper, se explica cómo se usa la atención para mejorar la segmentación semántica, mejorando los resultados de la Unet [103].

Debido a la existencia de tantos mecanismos de atención, en este trabajo nos limitaremos a escrutar el mecanismo que usan los Transformers para lograr sus tan buenos resultados. Realmente, las ideas generales son las que se aplican en todos los casos por lo que servirá para entender cualquier otro mecanismo de atención. En particular, se va a explicar el mecanismo de atención del Transformers del paper de Vaswani et al. [81].

Este Transformer tienen dos partes: un conjunto de encoders y un conjunto de decoders. En concreto, en este artículo se detalla que el conjunto de encoders consta de 6 y el conjunto de decoders tiene que ser del mismo tamaño. Cada uno de los encoders, consta de una capa de atención y una capa neuronal sencilla. Se puede ver esto en la Ilustración 12.

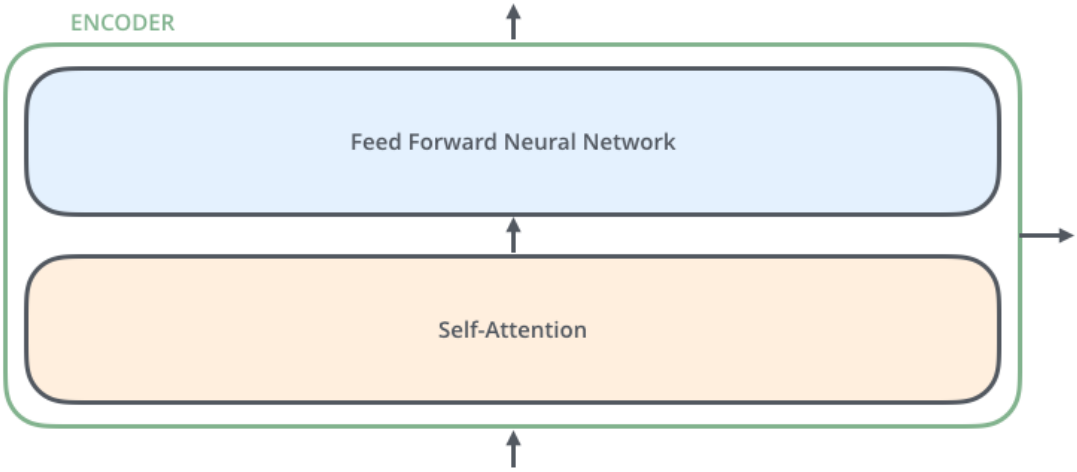

Ilustración 12: Encoder de un Transformer. Fuente: https://jalammar.github.io/illustrated-Transformer/

La capa que nos interesa para aplicar la atención es la de self-attention. Dicha capa ayuda al encoder a ver otras palabras de la oración de entrada mientras codifica una palabra en específico. La salida de esta capa de atención se introduce directamente a una red neuronal. En esta red, la posición de las palabras es irrelevante.

El decoder tiene las mismas dos capas que el encoder pero con una capa intermedia que ayuda al decoder a fijar la atención en partes relevantes de la oración que se introduce como entrada. Se puede ver en la ilustración 13.

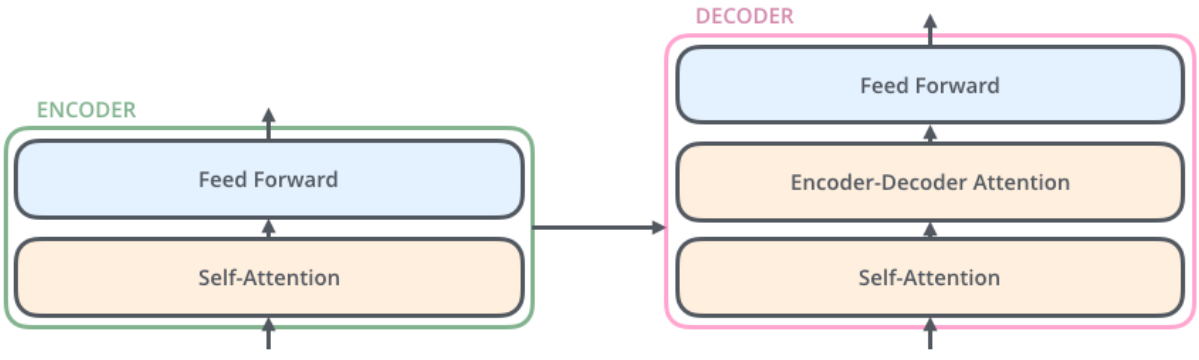

Ilustración 13: Transmisión de la atención entre un encoder y un decoder. Fuente: https://jalammar.github.io/illustrated-Transformer/

Vistos los componentes del Transformer en los que se encuentran los mecanismos de atención, veamos cómo funcionan. Para saber cómo funcionan, primero hay que ver cuál es su entrada. En NLP, las palabras se transforman en vectores con unos algoritmos de embedding para poder introducirlo en las redes. Al primer encoder, se le introduce una lista de vectores de palabras de longitud 512 cada uno mientras que al resto de encoders se le introduce la salida de los encoders anteriores.

Lo más relevante para tener en cuenta es que cada vector fluye a través del encoder de manera independiente. Las dependencias entre los vectores se producen en la capa de self-attention. En la capa de feed forward no existen estas dependencias por lo que se pueden ejecutar sobre varias de las palabras de manera paralela, acelerando el proceso. Se puede ver como varían las palabras a través del encoder en la Ilustración 14.

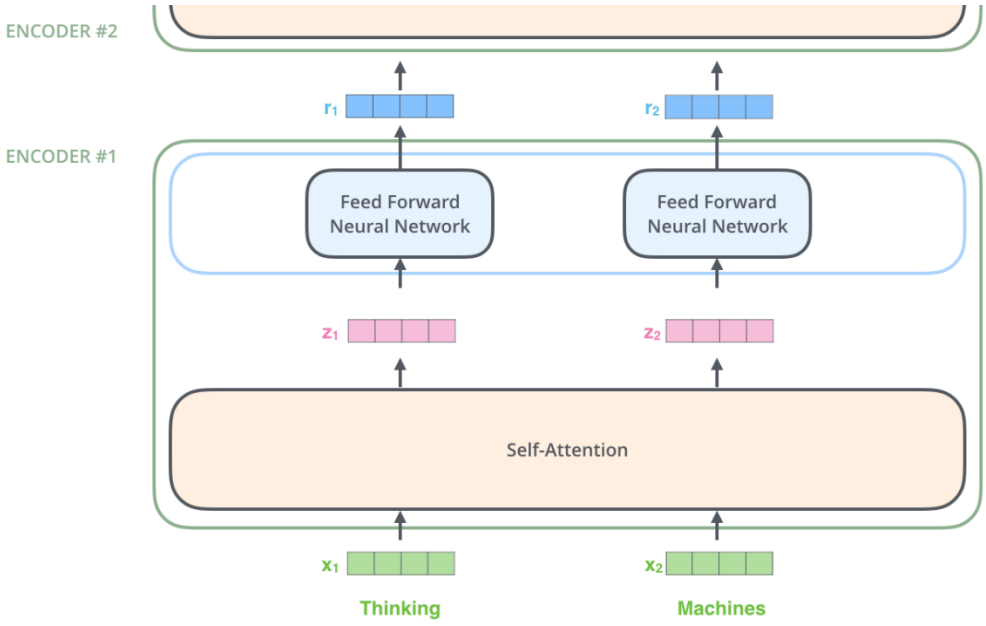

Ilustración 14: Palabras a través del encoder. Fuente: https://jalammar.github.io/illustrated-Transformer

Ya explicado cómo se introducen los datos y como pasan a través del encoder, vamos a explicar cómo funciona la capa de self-attention. Como ya hemos hablado posteriormente, la atención permite analizar otras palabras mientras se analiza la palabra dada. Veamos el ejemplo de la atención en la frase "Tom is 5 years old and he plays in the park.".

La atención permite asociar las palabras más relevantes entre ellas. Por ejemplo, ¿A qué se refiere el pronombre “he”? Para un ser humano es obvio que se refiere a Tom pero para una maquina no lo es tanto. Con la atención se puede asociar la palabra “he” con Tom como se puede ver en la ilustración 15.

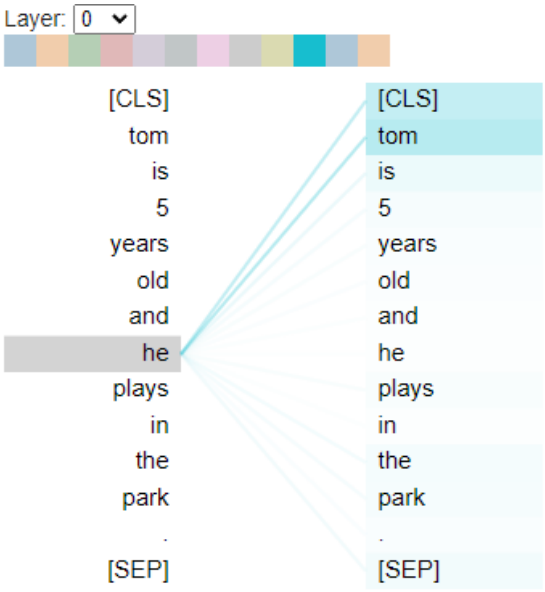

Ilustración 15: Atención del Transformer relacionando he con Tom.

Veamos cómo se implementa esta atención. Lo primero que hace falta es tener 3 vectores por cada uno de los vectores de entrada del encoder. Es decir, para cada palabra creamos un vector Query, un vector Key y otro vector Value. Estos tres vectores se crean pultiplicando el embedding por 3 matrices que se entrenan durante el proceso de entrenamiento. El tamaño de estos 3 vectores puede ser de cualquier tamaño independientemente del tamaño de los embeddings. Normalmente, el tamaño es de 64.

Estos tres vectores son abstracciones que son útiles para calcular y pensar sobre la atención. La segunda parte de la atención es calcular una puntuación entre cada palabra de la oración. La puntuación indica cuanto nos tenemos que fijar en una palabra en particular cuando nos estamos fijando en otra. Para calcular la puntuación de una palabra con respecto a otra, es decir, lo importante que es la segunda palabra a la hora de procesar la primera, hay que hacer el producto escalar del vector query de la primera palabra con el vector key de la segunda palabra.

Por ejemplo, imaginemos que en el ejemplo anterior queremos saber la relevancia de la palabra “Tom” para la palabra “he”. Para ello, haríamos la multiplicación escalar del vector query de la palabra he con el vector key de la palabra Tom.

Los siguientes dos pasos son dividir el valor de puntuación entre la raíz cuadrada de la longitud del vector Key. De esta manera, al realizar el descenso por gradiente, los resultados son más estables. Después, se pasan estos valores por una función softmax para que todos los valores de puntuación para una palabra dada sean positivos y sumen 1.

El último paso es multiplicar cada vector Value, el vector Value respectivo al vector Key en el computo de la puntuación, por su puntuación pasada por el softmax y hacer la suma ponderada. Este vector resultante es el vector que representa a la palabra a la salida de la capa de atención. Este vector se pasa a la siguiente red neuronal.

Para acelerar el proceso, se realizan todos estos pasos mediante multiplicaciones vectoriales. Para calcular los vectores Query, Key y Value, se multiplica una matriz X con tantas filas como palabras se introducen a la capa y tantas columnas como la dimensión de cada palabra por tres matrices, una por cada tipo de vector (W^Q, W^K y W^V).

Después de extraer los vectores, se realizan todos los demás pasos anteriores mediante multiplicaciones matriciales.

Aunque ya hemos explicado grosso modo la atención, faltan de explicar dos detalles más sobre la implementación de los Transformers. El primero es que las capas de atención no solo tienen una atención. Tienen multihead attention. Es decir, existen múltiples matrices W^Q, W^K y W^V, múltiples vectores Query, Key y Value y múltiples vectores salida de la capa de atención. Esto mejora a los Transformers porque:

1. Permite fijarse en diferentes posiciones de la oración. Siguiendo con nuestro ejemplo, es verdad que “he” ha de referirse a “Tom”, pero también tiene relevancia con la palabra “plays”, más incluso que la palabra “Tom” con plays.
2. Se generan diferentes subespacios de representaciones. De esta manera, conseguimos que en cada uno de estos subespacios se prioricen unas cosas por encima de otras. Salen diferentes subespacios porque las matrices W^Q, W^K y W^V se inicializan de manera aleatoria.

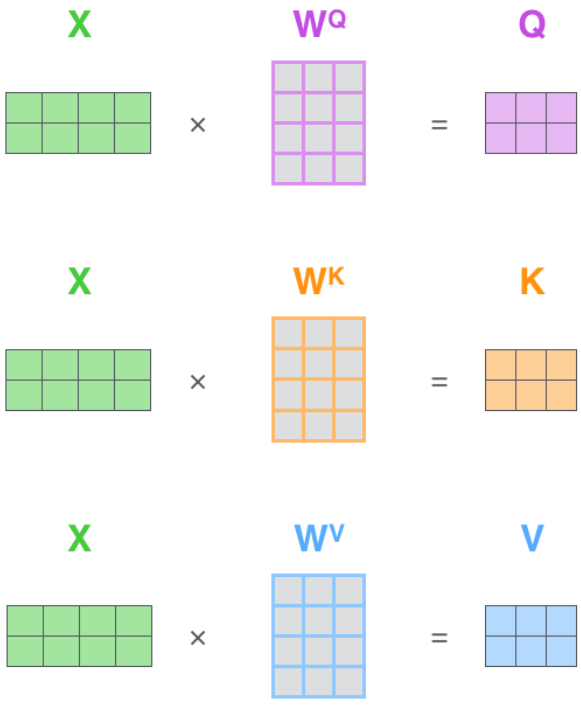

Ilustración 16: Multiplicación del vector X, con tantas columnas como palabras en la frase, por las matrices generadoras de los vectores Q,K y V. Fuente: https://jalammar.github.io/illustrated-Transformer/

En el Transformer que estamos explicando existen 8 subespacios de representaciones de las palabras. Sin embargo, la red neuronal posterior solo está esperando una representación por palabra. Para reducir todas estas representaciones a solo una, se concatenan todas las representaciones de las diferentes cabezas de atención y se multiplican por la matriz W^O. Esta matriz, que se entrena a la vez que el Transformer, permite ponderar las diferentes representaciones para pasárselas a la siguiente red.

El proceso completo de calcular los vectores después de la capa de atención quedaría de la manera que se ilustra en la Ilustración 17.

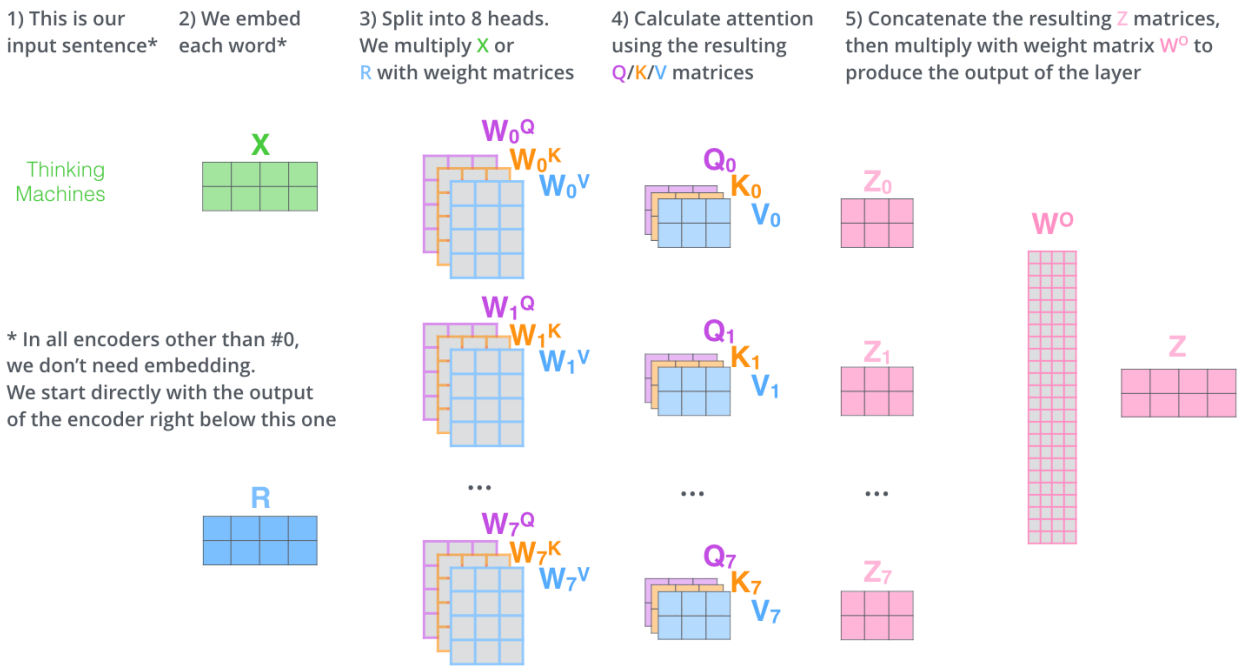

Ilustración 17: Proceso de calculo de los vectores de cada palabra con atención. Fuente: [The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time. (jalammar.github.io)]

La última cosa que no se ha explicado, es como el modelo tiene en cuenta el orden de las palabras. Para solucionar este tema, el Transformer suma un vector a cada embedding de entrada. Estos vectores siguen un patron especifico que aprende el modelo y ayuda a determinar la posición de cada palabra o la posición relativa entre palabras. A este vector de posición se le llama positional encoding.

3. Estado del arte

En esta sección, se va a hacer un estudio preliminar del estado del arte relativo a la tarea de CQA y de todos los asuntos relacionados con la misma. De esta manera, en la parte de experimentación se podrá entender de una mejor manera la metodología seguida y la finalidad del trabajo realizado.

Este capítulo va a estar dividido en las siguientes secciones:

1. En primer lugar, se abordará el Question Answering con su estructura principal y sus diferentes retos.
2. En segundo lugar, se tratará el tema principal de este trabajo, el Community Question Answering. Este se dividirá en las siguientes secciones:
    a. Diferencias con el Question Answering.
    b. Tareas que ha habido y como se han resuelto.
    c. Colecciones usadas para realizar esta tarea.
    d. Representación del significado de las preguntas y de los comentarios recientemente para solucionar el problema.
3. Por último, se extraerá una conclusión de todo lo revisado.

3.1 Question Answering (QA)

La tarea del **Question Answering** se encarga de responder una pregunta en lenguaje natural del usuario, con información que se encuentra dentro de un texto, mediante frases o pequeños pasajes del texto. Este es un campo muy amplio que se encarga de recuperar información de cualquier tipo, analizarla e incluso almacenarla [6].

Sus soluciones han tenido mucho impacto, ya que resolvían un problema que había sido clave durante muchos años. De hecho, en muchos congresos, como en el TREC [7] o en conferencias de inteligencia artificial, ha sido clave este tema y ha tenido su consecuente relevancia en dichos congresos.

En los primeros años de este campo, las técnicas utilizadas eran muy similares a las de **recuperación de información (IR)**. Sin embargo, con el auge de las redes neuronales, se han propuesta distintas técnicas basadas en estos modelos, que son específicos de la búsqueda de respuestas.

Esta sección se va a dividir en 4 partes: En la primera se van a hablar de las tareas en las que se divide un proceso de Question Answering tradicional, en los que están basados los sistemas de Question Answering actuales; en la segunda parte, se hará una clasificación de los distintos sistemas de Question Answering, ubicando dentro de este espectro el Community Question Answering; en la tercera parte, se hablará de la evaluación de dichos sistemas; y en la última parte de esta sección, se hablarán de los avances que ha habido en este área enlazándolos con el Community Question Answering.

3.1.1 Tareas del QA

El Question Answering es una tarea muy amplia. Llega desde la parte del procesamiento de la pregunta junto con los datos para responderla, hasta la generación de la respuesta. Por esta razón, para abordar este problema se ha dividido en diferentes partes, que siendo independientes unas de otras, permiten solucionarlo de una mejor manera. Allam et al.[8] y Carmona y Aldon [9] se hace un resumen de las principales tareas establecidas:

1. **Procesamiento de la pregunta**: Se intenta encontrar la pregunta para poder introducirla en un sistema de recuperación de la información y encontrar el tipo de respuesta, por ejemplo, si la respuesta tiene que ser una lista, una descripción… Dentro de este módulo, hay 3 grandes bloques que se tienen que realizar para llevar a cabo un procesamiento global de la pregunta:
    a. **Análisis de la pregunta:** Este análisis lo que busca es encontrar el foco de las preguntas formuladas, definido por Moldovan et al. [10] dentro del TREC 8 [7] como una palabra o una secuencia de palabras que indique que información necesita la pregunta para ser contestada. Esto es especialmente importante en las preguntas que exigen un desarrollo. Por ejemplo, en la pregunta "¿Quién es la persona más vieja de España?", el foco es "persona más vieja" porque es sobre lo que se va a responder. Si se sabe de qué tipo es la pregunta y cuál es su foco, el sistema lo tiene más fácil para responderla.
    b. **Clasificación del tipo de pregunta:** Para responder correctamente la pregunta, el sistema tiene que entender que tipo de respuesta busca. Esta respuesta puede ser de tipo persona, de tipo ciudad, de tipo fecha… Clasificar una pregunta para decidir qué tiene que devolver una persona, una localización o una fecha es relativamente más sencillo que detectar que tiene que devolver una razón o una descripción.
    c. **Reformulación de la pregunta**: Cuando ya se ha encontrado el foco de la pregunta y el tipo, el sistema extrae palabras clave para pasárselas al sistema de IR. Para expandir el número de palabras clave relacionadas con una pregunta, se utilizan ontologías como WordNet [11].
2. **Procesamiento de los documentos**: Este módulo usa varios sistemas de IR para obtener los documentos de los que se van a extraer las respuestas, que en su mayor parte del tiempo se encuentran en la red. El principal objetivo de esta parte de un sistema de Question Answering es obtener un conjunto de párrafos ordenados conforme a su probabilidad de contener la respuesta. Para llegar a este objetivo, esta subtarea se divide a su vez en diferentes apartados:
    a. **IR**: El objetivo de este submódulo es obtener resultados precisos que respondan a la pregunta realizada por el usuario, y ordenar dichos resultados en función de la importancia a la hora de responder a la pregunta. Es mejor que un sistema de Question Answering no dependa exclusivamente de otro de IR.
    b. **Filtrado de párrafos**: El número de documentos recuperados por el submódulo de IR puede ser muy grande. El objetivo de este sistema es filtrar los documentos y reducir la cantidad de texto por documento con probabilidad de contener la respuesta.
    c. **Ordenamiento de los párrafos**: Lo que se pretende en este submódulo es ordenar los párrafos según su probabilidad de contener información útil para la pregunta. Para llevar a cabo este ordenamiento, tradicionalmente se han usado métricas para ordenar los párrafos pero existen más aproximaciones como embeddings [12].
3. **Procesamiento de la respuesta**: Este módulo es el encargado de identificar, extraer y validar la respuesta del conjunto de párrafos ordenados que se han obtenido en la subtarea anterior. Los pasos que se requieren para llevar a cabo esta subtarea son identificar la respuesta de entre los

párrafos ordenados, extraer la respuesta eligiendo solo la palabra o la frase necesaria y validar la respuesta aumentando la confianza en la validez de esta. Tradicionalmente, este proceso se llevaba a cabo mediante un sistema de identificación de respuestas mediante NER[1] seguido de una búsqueda de patrones del tipo de respuesta o de N-gramas para extraer la frase o la palabra concreta. Hoy en día hay sistemas como BERT [13] que realizan todos estos pasos automáticamente.

3.1.2 Sistemas de QA

Los sistemas de QA se pueden clasificar dependiendo de diferentes factores. Por ejemplo, Jurafsky et al. [14] clasificaron los sistemas de QA según sus fuentes de datos; Yogish, Majunath y Hegadi [15] según sus fuentes de conocimiento y las técnicas usadas en el sistema… [16] De todas estas clasificaciones, los criterios fundamentales a partir de los que se pueden clasificar los sistemas de QA, extraídos de Mishra y Jain [17] y de Ojokoh y Adebisi [16] debido a su buen trabajo de síntesis, son:

- **El dominio:** Existen dos tipos de sistemas de QA, los de dominio abierto y los de dominio cerrado. Un sistema de dominio abierto devuelve respuestas a cualquier tipo de pregunta. Este tipo de sistemas usan colecciones generales de texto estructurado para producir respuestas. Por otro lado, los sistemas de dominio cerrado solo responden preguntas de un tema concreto. Se suelen crear para temas que requieren más precisión que la que dan los sistemas de dominio abierto. Por ejemplo, existen sistemas de QA de dominio cerrado para temas médicos, geoespaciales… En concreto, el **CQA** es un sistema de dominio cerrado ya que el tipo de datos tienen unas características particulares. Las colecciones de estos sistemas de dominio cerrado son del tema concreto que se requiere contestar.
- **El tipo de pregunta:** Generar respuestas para el usuario está directamente relacionado con el tipo de pregunta planteada. Los tipos de respuestas que puede haber son:
    - **Preguntas de hechos**: Estas preguntas responde a una pregunta en concreto como por ejemplo “¿En qué país esta Pamplona?”. Normalmente la respuesta suele ser una entidad nombrada.
    - **Preguntas de tipo lista**: El sistema devuelve una enumeración.
    - **Preguntas de definición**: El sistema tienen que procesar documentos para elaborar una respuesta que consista en el resumen de dichos documentos.
    - **Preguntas de tipo hipotético**: Esto requiere de información asociada a cualquier evento. Para esto se usan colecciones que tengan bases de conocimiento asociadas para producir respuestas.
    - **Preguntas causales**: Mientras las respuestas de hechos son entidades nombradas de los documentos, este tipo de preguntas necesita de una elaboración más compleja acerca de eventos o de objetos.
    - **Preguntas de confirmación**: Solo responden un si o un no. Para responderlas se necesitan mecanismos de inferencia, razonamiento y conocimiento del mundo.

[1] Reconocimiento de entidades nombradas

- **Según el tipo de fuentes de datos**: En , Jurafsky et al. [14] identificaban el QA basado en information retrieval y en conocimiento como los dos paradigmas más importantes del QA. Basándonos en las fuentes de datos, existen 3 tipos de sistemas de QA:
    - **QA basado en IR**: Usan la información que se puede acceder desde la red para responder a preguntas. En esta categoría se incluiría el Community Question Answering.
    - **QA basado en conocimiento:** Se basan en devolver una respuesta mapeándola sobre una ontologia como DBpedia.
    - **Usando múltiples fuentes de información**: Hay algunos sistemas que involucran bases estructuradas de conocimiento y datasets de texto para devolver respuestas a preguntas.
- **Clasificación según la forma de las respuestas generadas por el sistema de QA**: Hay dos tipos de categorías de respuestas según esta clasificación:
    - **Respuestas extraídas:** Devuelve una oración, un párrafo o multimedia extraída directamente de la fuente de datos. Se hace un ranking de la información según la pregunta y se devuelve la que más relación tenga.
    - **Respuestas generadas**: Pueden ser de tipo confirmación, que dicen sí o no según una inferencia llevada a cabo a partir de los datos; de opinión, que da un valor de rango a un objeto; y de dialogo, que devuelve las respuestas en forma de dialogo.
- **Clasificación basada en el lenguaje:** Divide los sistemas de QA según el número de lenguajes usado para procesar la pregunta. Hay de tres tipos:
    - **Sistemas de QA monolingüe**: En estos sistemas tanto la pregunta, como los recursos del documento y la respuesta del sistema están expresadas en el mismo lenguaje.
    - **Sistemas de QA cross-lingual**: En estos sistemas los documentos están en diversos lenguajes y la pregunta se convierte al lenguaje de los documentos antes de la búsqueda. Google Knowledge Graph traduce las preguntas de todos los lenguajes al inglés y devuelve la respuesta en inglés.
    - **Sistemas de QA multilingües**: En estos sistemas las preguntas del usuario y los documentos recurso están expresados en el mismo lenguaje y la búsqueda de la pregunta se realiza en el mismo lenguaje que la pregunta.
- **Clasificación basada en las aproximaciones**:
    - **Aproximación lingüística:** Consigue un análisis sintáctico completo de la colección del texto mediante técnicas de conocimiento lingüísticas. Mediante este análisis, se puede explotar la información lingüística para usarla en QA. Este tipo de técnicas tienen como limitación que hay que reescribir las reglas por cada lenguaje, además de que crear una base de conocimiento es un proceso muy lento, por lo que estas técnicas se usan en dominios específicos.
    - **Aproximación estadística**: Este tipo de técnicas usan una gran cantidad de datos para entrenar un sistema que pueda buscar y crear respuestas en forma de lenguaje natural, y han experimentado un boom últimamente con el auge del Deep Learning. Hablaremos de estas técnicas recientes posteriormente.
    - **Aproximación basada en patrones**: Este método busca patrones que consigan hacer frente a la influencia de la comunicación de diferentes dominios**.** Cambia el procesamiento clásico de otros métodos computacionales para poder aprender conocimiento lingüistico que luego le permita obtener las respuestas.

3.1.3 Evaluación en los sistemas de QA

Los sistemas de evaluación de QA tienen en cuenta una serie de parámetros para juzgar lo buena que es una respuesta. Estos parametros son los siguientes, según Allam et al. [8] y Hirschman et al. [18]:

- **Relevancia:** La respuesta debe ser una respuesta a la pregunta.
- **Lo correcta que sea:** La respuesta debe ser correcta.
- **Concisión**: La respuesta debe tener la información necesaria para responder a la pregunta, pero no más.
- **Completitud**: La respuesta debe ser completa, una respuesta parcial no debe tener toda la puntuación.
- **Justificación**: La respuesta debe tener el suficiente contexto para permitir al usuario determinar por qué se ha elegido esa respuesta.

Basados en estos criterios, hay 3 diferentes juicios para una respuesta extraída de un documento [19]:

- **"Correcta"**: Si la respuesta contesta a la pregunta de una manera correcta.
- **"Inexacta"**: Si falta alguna información o se le ha añadido alguna información innecesaria a la respuesta.
- **"No soportada"**: Si la respuesta no está soportada según otros documentos.

Hay muchas métricas para evaluar todo esto. Sin embargo, las más utilizadas son las siguientes:

- **La Precisión, el Recall y el F-measure:** La precisión y el recall son medidas tradicionales que han sido usadas en IR y la F-measure es la media harmónica de las dos medidas anteriores. Estas tres métricas se expresan como:

$$Precision = \frac{Numero\ de\ respuestas\ correctas}{Numero\ de\ preguntas\ contestadas} \quad (15)$$

$$Recall = \frac{Numero\ de\ respuestas\ correctas}{Numero\ de\ preguntas\ contestadas} \quad (16)$$

$$F - measure = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (17)$$

- **Rango reciproco medio (MRR):** Se usa para calcular la relevancia de una respuesta. Se empezó a usar en el TREC8 y su fórmula

$$MRR = \sum_{i=1}^{n} \frac{1}{r_i} \quad (18)$$

Donde n es el número de preguntas de test y ri es la relevancia de la primera respuesta correcta para la pregunta de test i-ésima.

- **Puntaje de confianza pesado (CWS)**: La confianza sobre lo correcta que es una respuesta se evalúa usando la CWS, que se definió en el TREC11:

$$CWS = \sum_{i=1}^{n} \frac{P_i}{n} \quad (19)$$

Donde n es el número de preguntas de test y p es la precisión de las respuestas en las posiciones 1 a i en una lista de respuestas ordenada.

3.1.4 Avances actuales en QA

En los últimos años, una de las ramas de la informática que más ha crecido han sido las redes neuronales. Estos son sistemas de aprendizaje automático que permiten capturar la semántica y otras relaciones del lenguaje con una precisión nunca vista. A partir del trabajo de Mikolov et Al. [20] representando palabras mediante vectores con redes neuronales, los trabajos en este campo han aumentado exponencialmente.

La idea de Mikolov era utilizar redes neuronales para crear representaciones vectoriales de las palabras, de tal manera que la red tuviera como entrada tanto la palabra como el contexto, creando una representación vectorial de la palabra contextualizada. De esta manera, Mikolov era capaz de predecir la palabra mediante su contexto y predecir un contexto a partir de la palabra, lo que permitía relacionar palabras según su similitud en este espacio vectorial.

Un ejemplo de la evolución del uso de las redes neuronales para el procesamiento del lenguaje y el QA es el artículo de Danqi Chen et Al. [21] donde se utilizan redes neuronales recurrentes para codificar los párrafos de los documentos y las preguntas para encontrar en qué párrafo se encuentra la respuesta más correcta a las preguntas. Cada palabra se representa mediante su embedding, estas representaciones vectoriales de Mikolov. Para encontrar los párrafos importantes de la Wikipedia se usa un hashing basado en bigrama. A pesar de la existencia de algunas estrategias diferentes, la arquitectura en líneas generales es como la que se ha explicado en esta sección.

En Kratzwald et al. [22] se aumentan los pasos tradicionales de procesado del texto que son recuperar los pasajes del texto con más probabilidad de que contengan la respuesta y, posteriormente, extraer la respuesta. En este trabajo, se añade una tercera fase que mejora el rendimiento de las respuestas haciendo un re-ranking de las respuestas según un conjunto de características.

En Wang et Al. [23], se usa un sistema basado en aprendizaje por refuerzo para extraer respuestas de los documentos leídos, además de usar un sistema neuronal para extraer que partes de los pasajes tienen más probabilidad de tener la respuesta.

Uno de las últimas y más importantes creaciones en el campo de las redes neuronales para procesamiento del lenguaje natural, fue la creación de los Transformers, siendo BERT [24] el primero y más importante de ellos. La idea es crear una red muy densa que permita codificar un texto con su contexto en todas las capas. De esta manera, se crea una red entrenada que es capaz de codificar cualquier texto, y, añadiendo

una última capa a la red depende el uso que se le quiera dar, permite dar resultados de estado del arte en cualquier campo del procesamiento del lenguaje natural.

Uno de los campos en los que ha ayudado BERT, ha sido precisamente en el campo del QA, ayudando a resolver el problema como en artículos como el de Zhang et al. [13]. En este artículo, se utiliza BERT para extraer la respuesta de los textos, una vez los textos ya han sido recuperados, filtrados y ordenados.

3.2 Community Question Answering (CQA)

El CQA ha crecido en popularidad recientemente, ya que, sitios como Stack Overflow, Quora… han elevado mucho sus visitas. Estos foros tienen la peculiaridad de permitir a la gente hacer sus preguntas y a expertos que están en la red responderlas. Esta capacidad de resolver dudas ha hecho que este tipo de páginas web hayan ganado tanta fama.

Esta sección se va a dividir en las siguientes secciones: En primer lugar, se hablará de las diferencias entre el CQA y el QA; para continuar, se verá cómo se ha abordado el problema mediante diferentes tareas como en el caso del QA; después, se hablará de las colecciones usadas para este problema; por último, se tratará del problema de la representación del significado y de las diferentes soluciones dadas a este.

En general, un foro hace que haya que tener unas consideraciones particulares, que no deben tener en cuenta los sistemas de QA más generales. Estas características se describen en el artículo de Patra [25] y son las siguientes:

- El usuario sube una pregunta al foro, que es filtrada por si tiene contenido inapropiado o esta duplicada, y se hace visible para todos los usuarios para ser contestada.
- Los demás usuarios interactúan de dos maneras:
    - Subiendo respuestas, que pueden ser relevantes o irrelevantes, basadas en sus opiniones y conocimientos.
    - Dando más o menos puntuación a las respuestas de otros usuarios, en función de su validez o su concordancia con la pregunta formulada.
- Algunos foros permiten a los usuarios también poner puntuación a la pregunta e incluso, preguntar al usuario que ha hecho la pregunta por más detalles.
- Por último, cuando el usuario que ha hecho la pregunta está satisfecho con una respuesta, la marca como la correcta, y la pregunta queda archivada.

En Bian et al. [26] se indica que los usuarios usan foros como estos cuando buscan opiniones y respuestas a preguntas complejas, ya que las preguntas simples las pueden buscar en Google. De hecho, estos sitios son tan populares principalmente porque permiten obtener respuestas precisas de manera rápida a preguntas formuladas en lenguaje natural.

Dentro de Blooma et al. [27], se describen las diferencias fundamentales entre los sistemas de QA y de CQA, que son las que se van a exponer a continuación.

La primera diferencia es acerca del tipo de la pregunta. En el caso del QA, las preguntas que se manejan son preguntas de una sola frase cuya respuesta es un hecho de la realidad, normalmente. Esto se ha

estado cambiando en los últimos años desde 2006, en el TREC y en otros grupos de investigación de QA, pero tradicionalmente ha sido así. Sin embargo, en el CQA nos encontramos preguntas de mas de una frase. Esto hace que el procesamiento de la pregunta sea más complejo que en el caso del QA convencional.

La segunda diferencia se refiere a la fuente de las respuestas. Esta diferencia es muy importante ya que determina la complejidad de procesado necesario para responder preguntas. En el caso de QA, los datos se extraen de documentos de un corpus, que se pueden encontrar en la red o en otro sitio. Por el contrario, en CQA los datos necesarios para responder a las preguntas son los que proveen los usuarios al contestar las preguntas. Esto hace que haya muchas diferencias como la longitud del contenido, la estructura y el estilo de escritura, el ruido del contenido ya que hay muchas respuestas invalidas, la probabilidad de que haya spam…

La tercera esta relacionada con la calidad de las respuestas, que es la que al final determina la calidad del sistema. Es evidente que los sistemas de QA extraen su información de corpus reputado, lo que eleva mucho la fiabilidad de las respuestas. Por otro lado, en CQA las respuestas obtenidas son de diferentes usuarios, cuya validez respondiendo preguntas varía. También varía la calidad de las respuestas según el foro en el que se esta basando el sistema de CQA. Además, la calidad de las respuestas es muy importante cuando hay varias respuestas para una pregunta en particular, ya que permite decidir cuál es mejor.

La cuarta diferencia es la disponibilidad de metadatos. Los servicios de CQA permiten obtener metadatos como la puntuación de las respuestas, junto a su autoría y a otros datos. Esto permite de alguna manera filtrar las respuestas que no son muy fiables. Todos estos metadatos no están disponibles en el caso del QA, lo que hace que en CQA hagan falta nuevas tareas que los aprovechen.

La última diferencia es el factor de tiempo a la hora de responder preguntas. En el caso de los sistemas de QA, las respuestas se generan instantáneamente a partir del corpus. Sin embargo, en el caso de CQA se requiere de respuestas por parte de los usuarios para contestar a la pregunta, lo que inherentemente tiene una espera a que los usuarios suban sus respuestas.

Como resumen, se incluye la Tabla 1 con las diferencias sintetizadas extraída de Blooma et al. [27].

	Sistemas de QA	Sistemas de CQA
Tipo de pregunta.	Preguntas de una sola frase acerca de hechos.	Preguntas de más de una frase.
Fuente de las respuestas.	Extractos de documentos de un corpus.	Generadas por la comunidad.
Calidad de las respuestas.	Normalmente muy buena.	Depende de la comunidad varía mucho.
Disponibilidad de metadatos.	Ninguna.	Mejor respuesta seleccionada y la puntuación de estas.
Demora.	Automático e inmediato.	Normalmente hay que esperar a alguna respuesta de la comunidad.

Tabla 1: Comparativa entre QA y CQA

3.2.1 Tareas del CQA

Al ser el CQA un tipo de QA, tiene tareas similares que han sido derivadas de las tareas de QA aplicadas al caso concreto del CQA. En general, las tareas diferentes entre el CQA y el QA convencional se definieron en los SemEval 2015 [1], 2016 [5] y 2017 [4], aunque en esta sección se verán las semejanzas y diferencias entre ambos tipos de sistemas junto a ejemplos de estos. Para empezar, hay que decir que los bloques en los que se dividen las tareas del QA también se pueden aplicar en el caso del CQA, como se puede ver en las siguientes secciones.

3.2.1.1 Procesamiento de la pregunta

En este caso, los pasos que se llevan a cabo se cruzan con los del caso del QA. El análisis, clasificación y reformulación de la pregunta son esenciales para la posterior búsqueda de las respuestas adecuadas. Sin embargo, con los avances en redes neuronales de los que se han hablado en la sección 2.4, todas estas se combinan usando embeddings o Transformers, aunque ya se hablará de este tema posteriormente.

Debido a las diferencias con respecto al QA, en el procesamiento de la pregunta se añade una tarea que no existía en el QA: **La similitud entre preguntas**, como se ve en la subtarea B del SemEval de 2016 [5]. Un gran problema en el caso de los foros es que una pregunta no se repita, es decir, saber si dos preguntas son similares o no para combinarlas. Por ejemplo, "¿Cuál es la ciudad más grande de España?" y "¿Cómo se llama la ciudad con la mayor extensión de España?" son dos preguntas con la misma semántica pero que se escriben de diferente manera. Estás dos preguntas deberían ser identificadas como la misma. De esta forma, evitamos tener redundancia en el foro, evitando así que una persona que ya haya respondido tenga que volver a hacerlo. También se consigue reducir el lag para obtener respuestas del que se hablaba en la sección 3.1, ya que, si se hace la segunda pregunta habiendo hecho antes la primera, se pueden devolver estas respuestas instantáneamente.

Algunas aproximaciones que se han seguido para solucionar este problema, en orden cronológico, han sido las siguientes:

1. **Okapi BM25** [28]: En este método de 1996, se compara la similitud entre dos preguntas basándose en su similitud de tokens. Se usa un alineamiento ponderado entre los tokens para computar la similitud entre las dos oraciones. Para computar este alineamiento ponderado, se usa la frecuencia inversa del documento (IDF) [29]. La función de similitud se define como:

$$Puntuación(q^1, q^2) = \sum_{i=1}^{|q^1|} IDF(q_i^1) freq\left(q_i^1, q^2\right) \quad (20)$$

$$freq(q_i^1, q^2) = \frac{\#(q_i^1, q^2)(k_1 + 1)}{D^r(q_i^1, q^2)} \quad (21)$$

$$D^r(q_i^1, q^2) = \#(q_i^1, q^2) + k_1\left(1 - b + b \cdot \frac{|q^2|}{longitud_{media(q^2)}}\right) \quad (22)$$

Siendo k_1 y b parámetros que se pueden modificar según la aplicación. Como la medida de similitud no es simétrica, se hace la media de $Puntuación(q^1, q^2)$ y $Puntuación(q^2, q^1)$. Esta medida pese a no ser extraordinariamente buena fue aplaudida en congresos como el TREC-3 por ser muy útil en IR.

2. **TRLM (Translation-Based Language Model)** [30]**:** Esta es una función de similitud usada por primera vez en el artículo de 2008 de Xue et al. [30]. Este sistema se basa en un modelo de traducción que computa la probabilidad $P(q^1|q^2)$ y $P(q^2|q^1)$, es decir, la probabilidad de generar la primera pregunta dada la segunda. Para crear esta probabilidad condicionada, se hace una versión suavizada del algoritmo de aprendizaje automático que genera las palabras de q^1 a partir de q^2. A esto se le añade la probabilidad de generar q^1 a partir de q^2, como una traducción. Las ecuaciones se definen de la siguiente manera:

$$P(q^1|q^2) = \prod_{w \in q^1} P(w|q^2) \quad (23)$$

$$P(w|q^2) = \frac{|q^2|}{|q^2| + \lambda} \cdot P_{mx}(w|q^2) + \frac{\lambda}{|q^2| + \lambda} \cdot P_{ml}(w|C) \quad (24)$$

$$P_{mx}(w|q^2) = (1 - \beta)P_{ml}(w|q^2) + \beta \sum_{t \in q^2} P_{trans}(w|t) \cdot P_{ml}(t|q^2) \quad (25)$$

Siendo $P_{ml}(w|C)$ la máxima probabilidad de parecido, donde C es el corpus de fondo y siendo el cálculo de $P_{ml}(w|C)$ cómo $\frac{\#(w,C)}{|C|}$, siendo λ un factor de suavizado y siendo β la ponderación entre P_{ml} y P_{trans}. P_{trans} es la probabilidad de traducir una palabra en otra, no necesariamente en distintos idiomas, simplemente la probabilidad de que se puede sustituir una palabra con la otra. Estas funciones obtenían daban mejores resultados en las de Okapi [28].

3. **Smoothed Partial Tree Kernels (SPTK)** son la base de KeLP [31], un sistema de 2016 aunque su origen proviene de un artículo de Croce et al. [32] de 2011. SPTK usa un kernel que computa la similitud de las preguntas en función de las subestructuras que comparten sus árboles sintácticos. Además, SPTK tiene en cuenta las relaciones entre palabras.
   Hay múltiples maneras de computar la similitud entre dos preguntas pero la manera básica, explicada en Kunneman et al. [33] se define de la siguiente manera dados los árboles sintácticos T_{Q_1} y T_{Q_2}:

$$TK(T_{Q_1}, T_{Q_2}) = \sum_{n_1 \in N_{T_{Q_1}}} \sum_{n_2 \in N_{T_{Q_2}}} \Delta(n_1, n_2) \quad (26)$$

$N_{T_{Q_1}}$ y $N_{T_{Q_2}}$ son los nodos de los árboles sintácticos T_{Q_1} y T_{Q_2} respectivamente. $\Delta(n_1, n_2)$ se calcula de distintas maneras dependiendo de 3 condiciones: Si las reglas de producción de T_{Q_1} en n_1 y T_{Q_2} en n_2 son diferentes, entonces $\Delta(n_1, n_2) = 0$; si n_1 y n_2 son terminales similares, entonces $\Delta(n_1, n_2) = Sim(w_{n_1}, w_{n_2})$, donde Sim puede ser cualquier función de similitud entre palabras; si las reglas de producción de T_{Q_1} en n_1 y T_{Q_2} en n_2 son iguales y ninguno es terminal entonces:

$$\Delta(n_1, n_2) = \prod_{j=1}^{hijo(n_1)} \Delta\big(hijo(n_1)_j, hijo(n_2)_j\big) \quad (27)$$

Por tanto, dados un par de árboles sintácticos de las preguntas $p = \langle T_{Q_1}, T_{Q_2} \rangle$ y un conjunto de entrenamiento de árboles C, las características se extraen de la siguiente manera:

$$SPTK\big(T_{Q_1}, T_{Q_2}\big) = \big\{TK\big(T_{Q_1}, C\big) + TK\big(T_{Q_2}, C\big)\big\} \quad (28)$$

Con este kernel se usan máquinas de soporte vectorial, cuya salida es la similitud entre T_{Q_1} y T_{Q_2}. Este método es algo más complejo que los anteriores, ya que usa la sintaxis de las frases para computar la similitud entre las preguntas.

4. **Neural Network Attention with token alignment** [34]**:** En este método de 2016, es donde se empiezan a introducir embeddings para computar la similitud entre preguntas. Dadas las preguntas $q^1 = [q_1^1, q_2^1, \dots, q_n^1]$ y $q^2 = [q_1^2, q_2^2, \dots, q_n^2]$, usando embeddings para cada token, se generan los vectores $\hat{q}^1 = [\hat{q}_1^1, \hat{q}_2^1, \dots, \hat{q}_n^1]$ y $\hat{q}^2 = [\hat{q}_1^2, \hat{q}_2^2, \dots, \hat{q}_n^2]$. Después de esto, se genera una matriz affine $L = \sigma(\hat{q}^{1T} \cdot \hat{q}^2) \in \mathbb{R}^{mxn}$. Dicha matriz se normaliza por filas y después por columnas para obtener los coeficientes de atención. Ahora, la j-esima palabra de q^1 se representa por $G[\hat{q}_j^1; \hat{v}_j]$ donde $\hat{v}_j$ es la representación ponderada de atención de $\hat{q}^2$, definido por L y G es no lineal. De esta manera, se obtienen las representaciones nuevas para cada token de q^2. El vector de la pregunta es obtenido como la suma de los vectores para cada token de la pregunta. Por último, se concadenan las dos representaciones de las preguntas y se aplica una red densa para generar la predicción.
5. **SoftCosine:** Es la función de ordenamiento usada por SimBOW [35], el sistema de similitud de preguntas que obtuvo los mejores resultados el SemEval 2017 [4]. Usa embeddings como en el caso anterior. Dadas X e Y como las tf-idf bag-of-words para las preguntas Q_1 y Q_2, se define la métrica SoftCosine cómo:

$$SoftCos(X, Y) = \frac{X^t M Y}{\sqrt{X^t M Y}\sqrt{Y^t M Y}} \quad (29)$$

$$X^t MY = \sum_{i=1}^{n}\sum_{j=1}^{n} X_i M_{ij} Y_j \quad (30)$$

$$M_{ij} = max\big(0, coseno(V_i, V_j)\big)^2 \quad (31)$$

Siendo V_i y V_j embeddings de las palabras de las preguntas.

Estas han sido aproximaciones que se han dado al problema de ver si dos preguntas son parecidas o no. Como se puede ver, cada vez el uso de los embeddings, que al final son aproximaciones neuronales, es cada vez mayor. Desde 2019, con la creación de BERT [24], se han podido hacer otras aproximaciones para similitud de frases con BERT como se puede ver en este artículo de Li et al.[36]. Sin embargo, no parece haberse aplicado al problema particular de question similarity.

3.2.1.2 Procesamiento de los documentos

En este caso, no tenemos documentos sino respuestas de un foro. Se ha heredado el nombre para seguir con el símil con el QA. Es por eso por lo que no se necesita una recuperación de documentos como en el QA original, sino un sistema por el cual se ordenen las respuestas en función de su utilidad para responder la pregunta. En el SemEval 2016 [5] esta subtarea se definió como la subtarea A: **Similitud pregunta-comentario**. También está relacionada con la subtarea C, que trata de establecer esta similitud, pero con comentarios externos al foro donde se encuentra la pregunta, pero las técnicas usadas son muy similares.

Esta subtarea es útil porque, entre otras cosas, la respuesta más votada puede ser ruido (puede haber sido votada por ser una broma y no contestar a la pregunta). En esta sección se va a hacer un pequeño resumen de los métodos usados para esta subtarea. Muchos métodos se solapan con la subtarea de similitud de preguntas, debido al parecido de la tarea:

1. **Okapi BM25** [28]: Este método de 1996 descrito previamente también se puede usar para calcular la similitud entre pregunta y respuesta. Esto es porque si una pregunta tiene relación con la respuesta, también la tendrá la cantidad de tokens que comparte. Sin embargo, a la hora de la verdad las preguntas y las respuestas no comparten demasiados tokens, por lo que el método no funciona demasiado bien.
2. **TRLM (Translation-Based Language Model)** [30]**:** Este método de 2008 descrito previamente se puede usar para puntuar las respuestas. Dada una pregunta q y un pool de respuestas A, encontrar la respuesta mejor se puede modelar como la probabilidad de convertir la respuesta en la pregunta, mediante un sistema de traducción:

$$a^* = argmax_{a \in A} P_{TransLM}(q|a) \quad (32)$$

3. **Smoothed Partial Tree Kernels (SPTK)** [31]: En su artículo de 2016, se utiliza este método para modelar la similitud en todas las subtareas del cQA en el SemEval 2016 [5]. El cambio entre las

diferentes subtareas el kernel que se usa para aplicar el SVM. En este caso, el kernel se define, dados los pares de comentarios $p = \langle q, a \rangle$, como:

$$PTK^{+}(P_a, P_b) + LK_A(P_a, P_b) \quad (33)$$

$$PTK^{+}(P_a, P_b) = TK(q_1, q_2) + TK(c_1, c_2) \quad (34)$$

Siendo LK_A un kernel lineal que opera sobre las medidas de similitud entre q y c, en características heurísticas y en características basadas el foro y en los metadatos que contiene cada respuesta. Para las medidas de similitud se suele utilizar la medida del coseno sobre sus embeddings.

4. **Un método basado en embeddings y en CNNs** [37]: En este método de 2016, se crean un embedding de la pregunta y de la respuesta usando una CNN. Dadas las preguntas $q = q_1, \dots, q_n$ y las respuestas $a = a_1, \dots, a_n$, se generan las matrices $\hat{q} = [\hat{q}_1, \dots, \hat{q}_n] \in \mathbb{R}^{dxn}$ y $\hat{a} = [\hat{a}_1, \dots, \hat{a}_n] \in \mathbb{R}^{dxm}$, donde d es el tamaño del embedding. Estos embeddings pueden estar aprendidos a priori usado un modelo como Word2Vec, o se pueden aprender como parte del modelo.

   Posteriormente a la creación de estos embeddings, se aplica un filtro convolucional de tamaño m a lo largo de todas las direcciones del vector, seguido de una capa de max pooling y una capa densa, generando una matriz de $\mathbb{R}^d$. El modulo de CNN se comparte entre la pregunta y la respuesta.

   Dado un vector para una pregunta y otro para una respuesta, la red es entrena usando el método de máximo margen:

$$\mathcal{L} = \sum_{(q,a)\in C} \sum_{(q,a')\in C'} \max(0, \gamma - s(q,a) + s(q,a')) \quad (35)$$

   Donde C es un conjunto de preguntas con su respuesta correcta y C' es un conjunto de preguntas con una respuesta incorrecta, γ es un umbral y s es una función de similitud (como la similitud del coseno).

5. **Un método basado en atención usando CNN y LSTM** [38]**:** En 2015, usando el trabajo anterior como base, los autores intentaron mejorar este modelo. Para ello, en vez de usar simplemente embeddings, se les pasa la pregunta y la respuesta a través de una capa bidireccional LSTM. Esto permitía codificar el contexto. Después, se usó una capa convolucional y una capa de max pooling, como en el trabajo anterior. Esto permitía codificar las dependencias lejanas mejor.

   Por otro lado, para mejorar el modelo se aplicó un modelo basado en atención al que le seguía una capa de max pooling de la pregunta. Este vector resultante se usaba para asistir a los vectores respuesta y calcular la atención de las respuestas. Posteriormente se aplicaba una capa de max pooling sobre los vectores de la pregunta ponderados, y el resultado se usaba como el embedding de la respuesta. Esto además permitía pesar las diferentes palabras de la respuesta según el contexto antes del max pooling.

   Finalmente, los autores de este trabajo combinaron ambas ideas: Usar la CNN para generar el embedding de la pregunta, usándolo para generar los pesos de atención de la respuesta. Esta respuesta ponderada por atención se le pasaba como entrada al módulo de la CNN para generar

el embedding final de la respuesta. El modelo se entrenaba usando el loss del margen máximo, que se ha definido previamente.

6. **Un método basado en CNN profundas** [39]**:** En este método de 2015, los autores usan redes convolucionales para generar embeddings para preguntas y respuestas. Una pregunta $q = q_1, \dots, q_m$ se transforma a una matriz usando embeddings de las palabras, que se aprenden como parte de este método. Se consigue una matriz entrada $s = \mathbb{R}^{dxl_q}$ donde d es la dimensión de la capa del embedding y l_q es la longitud de q. Cada fila de la matriz se convoluciona con un filtro $\mathbb{R}^m$, donde m es la anchura del filtro, y la matriz resultante se forma con las dimensiones $\mathbb{R}^{dx(l_q-m+1)}$.

   Para convertir la red convolucional en profunda, el modelo usa capas de pool k-max. Estas capas seleccionan los k mayores valores sobre una dimensión y devuelve la subsecuencia sin cambiar su orden relativo. De esta manera, se consigue que la pregunta tenga longitud independiente del embedding después del pool k-max. Esta capa consigue coger las características que contribuyen el máximo a lo largo de una dimensión, mientras se preserva el orden de las características. El valor k se elige como $\max\left(k_{maximo}, \left\lceil \frac{D-d}{D} \cdot l_q \right\rceil\right)$ donde D es la profundidad máxima y d es la profundidad actual. Este proceso de convoluciones seguidas de una capa k-max pool se hace D veces. Una capa densa finalmente convierte los vectores $k_{maximos}$ a un vector $\mathbb{R}^{n_s}$. Posteriormente, se extrae un vector de manera similar para la respuesta.

   Para medir la puntuación de una respuesta y una respuesta, se crea una función que tiene en cuenta las interacciones multiplicativas y aditivas entre vectores:

$$s(q,a) = U^T \sigma\left(v_q^T M^{[1:r]} v_a + V\left[v_q; v_a\right] + b\right) \quad (36)$$

   Donde $M^{[1:r]} \in \mathbb{R}^{n_s x n_v x r}$ es un tensor que captura las interacciones multiplicativas y U, V y b son parámetros.

   Al igual que los modelos para la tarea de buscar la similitud entre pregunta y pregunta sirven en esta tarea, este método se puede aplicar para esta subtarea B del SemEval [5]. Dada una pregunta q, encontrar una que este semánticamente la más cerca se puede modelar como:

$$q^* = argmax_{(t,a)\in C} \alpha v_q^T v_t + (1-\alpha) s(t,a) \quad (37)$$

   Donde C es la colección de los pares respuesta pregunta y alfa controla la contribución de la puntuación de similitud de la pregunta y la puntuación de similitud entre pregunta y respuesta.

Estos métodos tienen unos años, pero sientan las bases de lo que ahora conocemos como CQA. En la sección 3.4 se hablará más en detalle de la similitud semántica y de los diferentes métodos que existen.

3.2.1.3 Procesamiento de la respuesta

Normalmente, los sistemas de QA requieren de un proceso posterior a la recuperación antes de devolver la respuesta al usuario. Afortunadamente, en el CQA ya se tienen las respuestas en lenguaje natural por lo que normalmente no se realiza ningún tipo de procesamiento posterior a la recuperación.

3.2.2 Colecciones para evaluar el CQA

Los datasets que se emplean para evaluar el CQA son los siguientes:

1. **Yahoo! Answers Dataset**: En Qiu y Huang [39], se recopilan las preguntas resueltas en la categoría de ordenador e internet de Yahoo! Answers. Se extrajeron 312000 preguntas con sus respectivas respuestas.
2. **Antique:** En [40], se genera un dataset de 2626 preguntas de domino abierto que no solo son acerca de hechos y de diversas categorías. Las preguntas recogidas fueron hechas en servicios de CQA por usuarios reales. Cada pregunta tiene asociadas respuestas con etiquetas que definen su relevancia para contestar la pregunta.
3. **CQASUMM** [41]**:** Este dataset está preparado para responder las preguntas del CQA mediante un resumen de las mejores respuestas. El dataset tiene 100001 preguntas y 120174 respuestas ordenadas por hilos de sus respectivos foros. Estas preguntas y respuestas han sido extraídas de Yahoo! Answers.
4. **AmazonQA** [42]**:** Este dataset consta de 923 mil preguntas, 3.6 millones de respuestas y 14 millones de reseñas sobre 156 mil productos. Las preguntas están etiquetadas en función de si han sido respondidas o no, según las reseñas disponibles.
5. **ComQA** [43]**:** En este gran dataset de preguntas reales de los usuarios se observan aspectos complicados de tratar como la composicionalidad, el razonamiento temporal y las comparaciones. ComQA contiene 11214 preguntas agrupadas en 4834 clústeres de párrafos. Las preguntas de este clúster están relacionadas semánticamente pero no tienen por qué ser iguales.
6. **CQADupStack** [44]**:** Este dataset se usa para investigación en CQA exclusivamente. Contiene hilos de 12 subforos de StackExchange, anotados con información de la pregunta duplicada. Se proveen conjuntos de train y test predefinidos, para recuperación y clasificación, para poder comparar los resultados entre distintos sistemas.
7. **LinkSO** [45]**:** Este dataset está diseñado para ordenar preguntas similares en Stack Overflow. Stack Overflow contiene una cantidad masiva de preguntas respondidas por la comunidad de gran calidad, lo que resulta una oportunidad de oro para evaluar algoritmos de recuperación de respuestas para el CQA y para aprender cómo obtener respuestas similares.

3.2.3 Representación y similitud semántica

Después de todas las tareas que se han visto en este trabajo, se puede concluir que el núcleo del CQA es una buena representación y métrica de la similitud semántica entre textos. Medir la similitud semántica entre componentes como textos, oraciones o documentos juega un rol muy importante en una gran cantidad de tareas de NLP, por lo que ha avanzado mucho en los últimos años.

Al principio, se usaban técnicas como Bags of Words (BoW), la frecuencia de los términos o la frecuencia inversa del documento (TF-IDF) para representar textos como vectores para calcular su similitud semántica. Sin embargo, estas técnicas no tenían en cuenta los diferentes significados de las palabras y que se podían usar varias palabras para representar el mismo concepto. Por ejemplo, las frases “Javier y

Miguel hacen pan y cerveza" y "Javier hace pan y Miguel hace cerveza" tienen las mismas palabras, pero no significan lo mismo. Por otro lado, las frases "Manuel es intolerante a la lactosa" y "Manuel es alérgico a los productos derivados de la leche" tienen el mismo significado, pero las palabras varían.

La similitud textual semántica (STS) se define como la medida de equivalencia semántica entre dos bloques de texto. Los métodos de similitud semántica normalmente dan una puntuación de similitud entre textos, más allá de una decisión binaria de si es similar o no.

Este tema ha crecido tanto que se han creado diferentes familias de algoritmos para medir la similitud semántica entre textos. En esta sección se expondrán estas diferentes familias y algunos ejemplos representativos de cada uno, para entenderlas mejor.

3.1.3.1 Métodos basados en conocimiento

Los métodos basados en conocimiento calculan la similitud semántica entre dos términos en función de la información derivada de una o más fuentes de conocimiento como ontologías, tesauros, diccionarios… Estas bases de conocimiento permiten una representación estructurada de los términos y conceptos conectados por relaciones semánticas.

Las bases de datos utilizadas para este tipo de métodos son:

- **WordNet** [46]: Es una de las bases de datos léxica más utilizada para calcular la similitud semántica con métodos basados en conocimiento. Tiene incluidos más de 100000 conceptos en inglés.
- **Wiktionary**: Es una base de datos léxica que tiene 6.2 millones de palabras en 4000 lenguajes.
- **BabelNet** [47]: Es un recurso léxico que combina WordNet con los datos disponibles en la Wikipedia para cada synset.

Mediante estas bases de datos, se aplican una serie de métodos para calcular la similitud entre palabras, que se dividen en 3 tipos:

1. **Métodos basados en contar aristas:** La manera más directa de calcular la similitud entre dos palabras es considerar que la ontología en la que se basa el método es un grafo que conecta las palabras por su significado y que contar las aristas entre dos términos mide la similitud entre ellos. Una medida de estas es *path* que fue propuesta por Rada et al. [48] donde la similitud es inversamente proporcional al camino más corto entre dos términos. La medida *path* se define como:

$$sim_{path}(t_1, t_2) = \frac{1}{1 + \min\_long(t_1, t_2)} \quad (38)$$

   Por otro lado, se propuso la medida *wup* por Wu y Palmer [49] en el que se tenía en cuenta la profundidad de los dos términos dentro de la ontología y tenía también en cuenta la profundidad de ancestro común (LCS). Cuanto más profundo es un término dentro de la ontología, más específico es. La medida *wup* se define como:

$$sim_{path}(t_1, t_2) = \frac{2 \cdot profundidad(t_{lcs})}{profundidad(t_1) + profundidad(t_2)} \quad (39)$$

Por último, Li et al. [50] propusieron una medida que tenía en cuenta tanto la distancia del camino mínimo entre términos como su profundidad. La medida *li* se define cómo:

$$sim_{li}(t_1, t_2) = e^{-\alpha \cdot \min\_long(t_1, t_2)} \cdot \frac{e^{\beta \cdot profundidad(t_{lcs})} - e^{-\beta \cdot profundidad(t_{lcs})}}{e^{\beta \cdot profundidad(t_{lcs})} + e^{-\beta \cdot profundidad(t_{lcs})}} \quad (40)$$

2. **Métodos basados en características:** Los métodos basados en características calculan la similitud como una función de propiedades de las palabras como el glos, conceptos de vecindad… [51] El glos se define como el significado de una palabra en un diccionario y una colección de gloses se define como glosario. Hay varias maneras de comparar la similitud entre glosarios, por ejemplo, se puede ver la cantidad de palabras en común entre estos. La medida Lesk [52] asigna un valor de similitud entre dos términos en función de las palabras que se solapan entre sus gloses y entre los gloses de las palabras relacionadas con estos. Jiang et al. [53] proponen un método basado en características donde la similitud semántica se mide usando gloses de conceptos presentes en la Wikipedia.
3. **Métodos basados en contenido de la información**: El contenido de la información (IC) se define cómo la información derivada de un concepto cuando aparece en un contexto [54].Si el valor del IC es muy alto indica que la palabra es muy específica y clara, mientras que si el valor de IC es bajo significa que hay más palabras abstractas en el significado de la palabra. La especificidad de la palabra se puede calcular mediante la frecuencia inversa del documento (TF-IDF), que usa el concepto de que una palabra es más especifica si aparece menos en un documento. Resnik y Philip [55] propusieron una medida de similitud semántica llamada *res* que media la similitud con la idea de que si dos conceptos tenían un ancestro común comparten más información ya que el valor de IC del LCS es mayor. *Res* se define cómo:

$$sim_{res}(t_1, t_2) = IC_{t_{lcs}} \quad (41)$$

D. Lin [56] propuso una extensión de la medida *res* teniendo en cuenta el valor de IC de los dos términos. La medida *lin* se define como:

$$sim_{lin}(t_1, t_2) = \frac{2 \cdot IC_{t_{lcs}}}{IC_{t_1} + IC_{t_2}} \quad (42)$$

Jiang y Conrath [57] calcularon una medida de distancia basándose en la diferencia entre la suma de los IC de los términos y el valor de IC del ancestro común. La medida de distancia reemplaza a la longitud del camino mínimo. La medida *jcn* se define cómo:

$$dis_{jcn}(t_1, t_2) = IC_{t_1} + IC_{t_2} - 2 \cdot IC_{t_{lcs}} \quad (43)$$

$$sim_{jcn}(t_1, t_2) = \frac{1}{1 + dis_{jcn}(t_1, t_2)} \quad (44)$$

4. **Métodos combinados**: Se han propuesto medidas combinando varios de métodos basados en conocimiento que previamente han sido explicados.
   Gao et al. [58] propusieron una medida de similitud semántica basada en la ontología de WordNet donde se utilizaban 3 estrategias para poner pesos a las aristas del grafo y se usaba la distancia del camino mínimo para medir la similitud. La primera estrategia era que las profundidades de los términos de WordNet en el camino entre dos términos se debían tener en consideración añadiéndolas como pesos a las aristas. La segunda estrategia era que solo se debía añadir como peso la profundidad del LCS de los términos. La tercera estrategia era que el valor de IC de los términos se añadía a los pesos.
   Por otro lado, Zhu e Iglesias [59] propusieron una medida de camino ponderado llamado *wpath*, que añadía el valor IC del LCS como un peso de la longitud del camino mínimo. Quedaba así:

$$sim_{wpath}(t_1, t_2) = \frac{1}{1 + \min \_long(t_1, t_2) \cdot k^{IC_{t_{lcs}}}} \quad (45)$$

   Siendo k un hiperparametro que se puede modificar depende del dominio.

Los métodos basados en conocimiento son computacionalmente muy simples. Además, las bases de conocimiento que se encuentran por debajo hacen que los modelos de similitud sean muy robustos y permiten gestionar de manera eficiente problemas como el de sinónimos o frases hechas. Además, estos métodos son muy sencillos de agregar.

Sin embargo, son muy dependientes de las bases de conocimiento y tienen que estar siempre actualizadas, lo que hace que se requiera mucho tiempo y una gran cantidad de esfuerzo humano.

3.2.3.2 Métodos de similitud semántica basados en corpus

Los métodos de similitud semántica basados en corpus miden la similitud semántica entre términos usando la información recuperada de un gran corpora. El principio de este tipo de métodos se llama "hipótesis distribucional" [60] que utilizan la idea de que las palabras que tienen similitud semántica suelen aparecer cerca. Hay muchas medidas de similitud semántica, cómo se exploran en el paper de Mohammad y Hurst [61]. Algunos de los estos métodos son:

1. **Embeddings de palabras**: La aproximación más utilizada hoy en día. Son representaciones vectoriales de las palabras que tienen en cuenta las relaciones lingüísticas entre las palabras. Estos vectores se computan usando diferentes aproximaciones como redes neuronales [20], matrices

de coocurrencia [62] o representaciones en función del contexto donde aparece la palabra [63]. Algunos de los embeddings preentrenados más utilizados son:

a. **Word2vec** [20]: Es un modelo basado en redes neuronales para producir una representación vectorial distribuida de palabras en función de un corpus. Hay dos tipos de word2vec: Una basada en Bag Of Words continua (CBOW) y otra basada en un skip-gram. La arquitectura es una red densa con una capa de entrada, una oculta y otra de salida. La entrada del modelo es un corpus de palabras grande y la salida es la representación vectorial de las palabras. El modelo CBOW predice la palabra actual usando las palabras del contexto, mientras el skip-gram predice el contexto a partir de la palabra objetivo.
b. **Glove** [62]: Este embedding está desarrollado por la universidad de Stanford y se basa en la coocurrencia de las palabras en una matriz basada en el corpus. Estima la similitud basada en el principio de que las palabras similares entre ellas aparecen juntas.
c. **fastText** [64]**:** Este embedding desarrollado en Facebook crea vectores de palabras basados en skip-grams donde cada palabra se representa como una colección de n-gramas de caracteres.
d. **Bert** [24]: Este es un Transformer preentrenado que puede ser ajustado añadiendo una última capa para crear embeddings para diferentes tareas de procesamiento de lenguaje natural. Esto permite incurrir tener en cuenta cualquier palabra aunque este fuera del vocabulario, palabras denominadas OOV (Out Of Vocabulary). En el preentrenamiento el modelo se entrena usando un corpus de 3300M de palabras del corpus Book y de la Wikipedia en inglés. La primera tarea que tiene que resolver es predecir las palabras que faltan en el corpus, porque están enmascaradas. La segunda tarea, es predecir de pares de oraciones aleatorias que se le pasa al modelo, adivinar cuales están seguidas y cuáles no. En el ajuste fino, el modelo se entrena con la última capa para abordar una tarea de PLN en particular.

El mayor problema de los embeddings a la hora de computar la similitud semántica es la deficiencia en la confluencia de significados. Hay palabras polisémicas que, cómo pueden ir acompañadas de diferentes grupos de palabras, generan malos embeddings. De esto se libra Bert, ya que determina el significado concreto en base al contexto de la palabra.

2. **Latent Semantic Analysis (LSA)** [65]: Este era uno de los métodos para la similitud semántica basada en corpus anterior a los embeddings. Se crea una matriz de coocurrencia donde las filas representan las palabras y las columnas representan los párrafos, y las celdas representan la frecuencia de cada palabra en su respectivo párrafo. A esta matriz se le aplica el singular value decomposition (SVD). Este método reduce el número de filas, pero mantiene el número de columnas que corresponde al número de párrafos. Entonces cada concepto se representa como su respectiva fila y la similitud entre palabras se representa como la similitud del coseno entre los vectores de dichas palabras.
3. **Hyperspace Analogue to Language (HAL)** [66]: HAL crea una matriz de coocurrencia de palabras donde las filas y las columnas representan las palabras del vocabulario y los elementos de la matriz son los valores de asociación entre esas palabras. Estos valores se calculan con una ventana móvil de tamaño variable sobre el corpus. La similitud semántica se calcula midiendo la distancia euclídea o manhattan entre los vectores de las palabras.

4. **Word-Alignment models** [67]: Los modelos de alineamiento de palabras calculan la similitud semantica de las frases en función de su alineamiento sobre un corpus grande. La segunda, tercera y quintas posiciones del SemEval 2015 usaron métodos basados en alineamiento de palabras. El problema de estos modelos es que los resultados pueden estar condicionados por un sesgo en la colección y en cómo se generan los pares de frases similares.
5. **Latent Dirichlet Allocation (LDA)** [68]: El LDA es usado para representar un tema o la idea general de un documento como un vector. Esta técnica es muy usada en topic modelling debido a la reducción de dimensionalidad de los vectores producidos. También puede ser empleado para calcular la similitud semántica entre textos mediante la similitud del coseno aplicada sobre estos vectores.
6. **Distancia normalizada de Google** [69]: La distancia normalizada de Google mide la similitud entre dos términos en función de los resultados que se obtienen cuando se introducen dichos términos en el buscador Google. El principio detrás de este método es que dos palabras ocurren juntas más frecuentemente en páginas web si son similares.
7. **Modelos basados en dependencias** [70]:Los modelos basados en dependencias calculan el significado de una palabra en función de sus vecinos en ventanas de un determinado tamaño. Para cada palabra se generan plantillas sintácticas usando los nodos anteriores y posteriores a la misma. La representación vectorial se genera añadiendo cada ventana en la ubicación que tiene la palabra de que aparece como raíz, así como la frecuencia que aparece esa estructura sintáctica a lo largo del corpus. Para calcular la similitud, se usa la medida del coseno sobre estos vectores.
8. **Modelos basados en kernels** [71]: Los métodos basados en kernel se usan para encontrar patrones en los datos textuales que permiten detectar similitudes entre partes del texto. Los dos tipos de kernels más usados son los kernels de secuencia [72] y los kernels tipo árbol [73]. El SPTK [31] que se presenta en la sección 3.1 es un modelo basado en kernels de tipo árbol. Estos kernels son muy buenos a la hora de identificar la estructura en las oraciones de entrada basándose en su dependencia y las reglas gramaticales del lenguaje. Suelen usarse junto a algoritmos de aprendizaje automático como las máquinas de soporte vectorial para adaptar los datos del texto a varias tareas como etiquetado de roles semánticos.
9. **Modelos basados en atención en las palabras** [74]: La idea en este caso es que dada una palabra y su contexto, no todos los términos de su contexto tienen la misma relevancia. Este tipo de modelos permite modelar la semántica de la palabra en función de su contexto, ponderando las palabras este en función de cómo afecten al termino principal.

A diferencia de los sistemas basados en conocimiento, los sistemas basados en corpus son independientes del lenguaje y del dominio. Sin embargo, estos métodos no tienen en cuenta el significado real de las palabras, además del tiempo y recursos necesarios para crear un corpora grande con el que entrenar estos modelos.

3.2.3.3 Métodos basados en Deep Learning

Los avances actuales en Deep Learning no han dejado de lado a la similitud semántica. Las técnicas más usadas incluyen redes neuronales convolucionales (CNN), long short term memory (LSTM), bididirectional long short term memory (Bi-LSTM) y LSTM con árboles recursivos. En general, estas aproximaciones tienen

dos operaciones fundamentales: convoluciones y pooling. Las convoluciones en texto se definen como la suma de los productos elemento a elemento de un vector de la oración y de una matriz de pesos. Por otro lado, las operaciones de pooling se encargan de eliminar las características que tienen un impacto negativo o que no son relevantes para la tarea que se quiere resolver. Estos métodos, aunque crean representaciones vectoriales de los textos, se basan en Deep Learning por lo los distinguimos de los métodos basados en corpus. Algunos ejemplos son:

1. **CNN:** Wang et al. [75] proponen un modelo que estima la similitud semántica entre dos frases basado en la descomposición léxica y la composición de la misma. Este modelo usa los embeddings preentrenados de word2vec para crear los embedding de las oraciones s_1 y s_2. Se crea una matriz de similitud M de dimensión i x j donde i y j son el número de palabras de la oración 1 y de la oración 2 respectivamente. Las celdas de la matriz representan la similitud del coseno entre las palabras de los índices de la matriz. Para calcular la similitud semántica entre los vectores $\overrightarrow{s_1}$ y $\overrightarrow{s_2}$ se usan tres funciones: Una global, una local y otra de máximo. La función global construye el vector de matching con s_1 cogiendo la suma ponderada de los vectores de todas las palabras de s_2, la función local coge solo los vectores de palabras en una ventana dada y la función máxima solo coge los vectores de las palabras con máxima similitud. La segunda fase del algoritmo utiliza funciones de descomposición para calcular el componente de similitud y de disimilitud entre los vectores de las oraciones y los vectores de matching. Estos componentes se pasan a través de una red convolucional de dos canales con una capa de max pooling. La similitud se calcula usando una capa sigmoide que da un valor entre 0 y 1. Este modelo obtuvo el mejor MAP en el dataset QASent [2] y el mejor MAP y MRR en el dataset WikiQA [76].
2. **LSTM:** Las redes LSTM también han dado unos resultados muy buenos. Estas redes permiten tener en cuenta palabras anteriores, aunque estén muy alejadas en el texto. Tien et al. [77] usa una red combinando LSTM y CNN para crear un sentence embedding de word embedding preentrenados seguido de una LSTM para calcular su similitud.
3. **CNN:** He y Lin [78] desarrollaron una arquitectura hibrida entre una Bi-LSTM y una CNN que estimaba la similitud del modelo. Un LSTM bidireccional son dos redes LSTM: Una que analiza el texto en de derecha a izquierda y otra en sentido inverso. Con estas redes se puede capturar el contexto como en este artículo. Se crea un modelo de interacción de palabras con las medidas de similitud del coseno, distancia euclídea y manhattan sobre los estados ocultos de la Bi-LSTM. Estas sirven para crear un vector ponderado de las palabras de un texto en función de su importancia. Estos vectores ponderados se pasan a una red convolucional que devuelve un valor entre 0 y 1 de similitud entre textos. Este modelo supero a los mejores modelos del dataset SICK.
4. **Atención**: Lopez-Gazpio et al. [79] propusieron una extensión de modelo de atención descomponible (DAM) propuesto por Parikh et al. [80]. Se uso una secuencia de 3 capas consecutivas: Una capa de atención, otra de comparación y la última de agregación. La capa de atención calculaba dos vectores de atención por cada frase. La capa de comparación concatenaba los vectores de atención con los vectores de la oración para crear un vector representativo por cada oración. Por último, la capa de agregación aplana los vectores y calculan la distribución de probabilidad sobre ellos, para calcular la similitud. Este modelo daba mejor rendimiento en el dataset SICK y en el benchmark STS que DAM y otros modelos como Bi-LSTM.
5. **Transformers**: Vaswani et al. [81] proponen un modelo que usa mecanismos de atención para capturar las propiedades semánticas de las palabras en los embeddings. El Transformer tiene dos partes: Un “encoder” y un “decoder”. El encoder consiste en capas de atención multi cabeza

seguidas de una capa densa. El decoder es parecido al encoder pero con una capa adicional de atención multi capeza que captura los pesos de atención de la salida del encoder. En sus orígenes, este modelo fue pensado para la traducción pero Devlin et al. [24] usaron este modelo para crear BERT embedding de palabras. En Sun et al. [82], se propone un framework que hace un entrenamiento fino de estos embeddings preentrenados cuando se presenta una nueva tarea. Esta idea mejoro a BERT. A partir de aquí se propusieron muchas mejoras. Una de las más relevantes es la de Lan et al. [83] que utilizaba dos técnicas para reducir la complejidad computacional de BERT llamadas "parametrización factorizada de los embeddings" y "compartición de la capa cruzada de parámetros". Este modelo funcionaba mucho mejor que BERT. Por último, Raffel et al. [84] propusieron un Transformer con el corpus "Colosal Clean Crawled Corpus" creando un modelo llamado T5-11B. Este modelo tenía un enforque distinto que BERT donde la secuencia de entrada estaba vinculada a un token que decía cuál era la tarea de NLP que se iba a realizar, eliminando las fases de preentrenamiento y de entrenamiento fino. Este modelo mejoro a todos los demás Transformers propuestos anteriormente. Se ha visto que aumentando el número de parámetros este modelo funciona cada vez mejor, pero se necesita una capacidad computacional altísima para llevar a cabo estos entrenamientos.

Los modelos basados en Deep Learning han mejorado por mucho todos los métodos tradicionales. Sin embargo, estas implementaciones de Deep Learning necesitan una cantidad de recursos computacionales muy grandes. Otra desventaja es que estos modelos son cajas negras que no tienen una interpretación clara.

3.4 Conclusión

El QA es una tarea con tanto recorrido que tiene una taxonomía muy amplia, así como unas métricas muy bien establecidas, que se han podido leer durante las secciones anteriores de este trabajo. Además, esta tarea, a pesar de ser una tarea con mucho recorrido, sigue despertando el interés de la comunidad tal y como se han podido leer en la sección 2.3.

Del QA nace el CQA y aunque ambas tareas se parecen, tienen muchas diferencias debido a la naturaleza tan diferente de los datos y de los tipos de preguntas.

Tanto dentro del QA como del CQA es fundamental realizar 3 pasos: El procesamiento de la pregunta que permite al sistema entender que es lo que el usuario está buscando, el procesamiento de los documentos para recuperar los datos de los que se van a extraer las respuestas y, por último, el procesamiento de la respuesta permite extraer exactamente la respuesta que espera el usuario.

El CQA tiene dos tareas que el QA no tiene: La similitud entre preguntas, para no mantener en un foro preguntas duplicadas y responder antes a los usuarios, y, la similitud entre preguntas y comentarios, para ordenar los comentarios de dichos foros en función de si contestan o no a la pregunta. Esta tarea, la del CQA, a pesar de no tener tantos años como el QA, tienen una serie de colecciones que se han usado para evaluarla en las diferentes ediciones de SemEval en los que se ha planteado.

En la sección 3.3, se ha llevado a cabo una pequeña revisión de los diferentes métodos de similitud semántica que se han llevado a cabo durante los años, ya que, al ser una de las tareas más necesarias en

el caso del procesamiento del lenguaje natural, ha tenido un largo recorrido. Este análisis se ha llevado a cabo ya que el análisis de la similitud es fundamental en el CQA. Dentro de esta revisión de métodos, se ha visto que todos los tipos de métodos tienen sus pros y sus contras. Los métodos basados en conocimiento son muy rápidos, pero dependen mucho de sus fuentes de conocimiento, los métodos basados en corpus funcionan bien, además de ser independientes del lenguaje y el dominio, pero requieren de mucho tiempo para crear el corpus correcto además de no reflejar el significado real de las palabras.

Las últimas aproximaciones propuestas de similitud semántica desarrolladas se han creado mediante Deep Learning. Estas arquitecturas han dado un golpe encima de la mesa dentro del campo de la similitud semántica, ya que han obtenido resultados de similitud nunca imaginados. Sin embargo, tienen problemas por cantidad de recursos necesarios para entrenar estos modelos y por ser una caja negra sin interpretabilidad.

Por último, parece a la hora de aplicar métodos de Deep Learning, se puede observar que en Question answering se ha desarrollado mucho más que en Community Question Answering. Es por eso que se puede ver que muchos de los métodos más novedosos de Deep Learning, aquellos que utilizan la atención y los Transformers, se han utilizado en Question Answering pero no en Community Question Answering. Además, en Question Answering estos métodos han dado muy buenos resultados tanto en eficiencia como en interpretabilidad. Por tanto, parece lógico hacer una traslación de estos métodos de una tarea a la otra.

Por tanto, las hipótesis que se pueden plantear es que los Transformers y demás métodos basados en atención pueden codificar la información semántica nunca vista. Esto se debe a que los embeddings contextuales extraídos por los Transformers permiten codificar la información semántica de una manera nunca vista.

4. Marco experimental

En esta sección se detalla lo necesario para poder reproducir las propuestas de este trabajo. Concretamente se van a detallar las colecciones sobre las que se van a evaluar las propuestas realizadas, así como las métricas que se van a utilizar.

4.1 Colecciones

En esta sección se van a desgranar las 3 colecciones sobre los que se ha realizado el análisis de este trabajo. A excepción de la colección AmazonQA, las colecciones sobre las que se trabaja pertenecen a congresos particulares y por tanto están vinculados a tareas muy específicas. Estas tareas también se presentarán en esta sección. Por último, se estudiará la estructura interna de las colecciones para comprender que estructuras de datos son más adecuadas a la hora de procesarlos.

4.1.1 AmazonQA

Los clientes de las aplicaciones de compras por internet crean miles de preguntas de productos al día. Por ejemplo, casi todas las preguntas de Amazon se responden en un periodo de tan solo un par de días [42]. Además, casi la mitad de las preguntas de los usuarios se pueden responder a partir de las opiniones realizadas por otros clientes.

Por este motivo, resulta interesante crear un sistema que responda automáticamente a las preguntas de los usuarios a partir de las opiniones disponibles de un producto dado. Por ello, la colección AmazonQA pretende dar solución a lo que los autores denominan *review-based Community Question Answering*. En este trabajo no se pretende dar solución a este problema. Sin embargo, como veremos a continuación, este dataset también se puede usar para solucionar la tarea de *Community Question Answering* al uso.

Para facilitar el uso de este dataset, los creadores de este realizaron un preprocesamiento. La primera parte del preprocesamiento consistió en eliminar de las reviews contenido repetido que era innecesario. Por otra parte, se eliminaron preguntas, respuestas y reviews que eran particularmente largas comparándolas con la mediana de la longitud de estas.

La segunda parte del preprocesamiento fue extraer los extractos de las reviews relevantes para responder a cada pregunta, dividiendo además dichos fragmentos en partes de 100 tokens. También se convirtió todo el texto a minúsculas menos abreviaciones de empresas o entidades, las cuales necesariamente deben ir en mayúsculas.

Con respecto a las estadísticas del dataset, contiene 923 mil preguntas, 3.6 millones de respuestas y 14 millones de reviews. Las anotaciones del dataset son múltiples, ya que quieren tratar de resolver una tarea que es diferente a la que nos interesa en este trabajo. Sin embargo, la anotación que nos interesa a nosotros es la anotación de **helpful** que hay en cada una de las respuestas. Esta anotación consta de dos números que se calculan en función de los likes y la cantidad de respuestas encadenadas a esa respuesta. De esta manera, una respuesta con mayor valor indica que tiene más utilidad para los usuarios.

Como se ha abordado esta tarea como una clasificación, se ha dividido en función del parámetro helpful las respuestas en malas, potencialmente buenas y buenas. Si la suma de ambos números es 0, se asume que la respuesta es mala. Por el contrario, si la suma de ambos números supera el número 4 se considera que la respuesta ha sido muy útil. En el resto de los casos, las respuestas son etiquetadas como potencialmente buenas. Se ha realizado de esta manera en primer lugar teniendo en cuenta que, si no tiene ningún like y ninguna respuesta encadenada, esa respuesta ha pasado desapercibida y por tanto es poco relevante. Por otro lado, la división entre respuestas potenciales y buenas se ha realizado viendo ejemplos y fijando este umbral en función de estos.

El dataset se ha dividido en subconjuntos de entrenamiento, validación y test, para que todas las preguntas del mismo producto vayan a la misma partición. Los porcentajes de división han sido 80%-10%-10%. En este trabajo, se usa validación para ajustar el número de épocas que entrenar los modelos


```
"answers": [
    {
        "helpful": [2, 2],
        "answerType": "NA",
        "answerText": "hmmm...I imagine so, but I used mine on s
    },
    {
        "helpful": [1, 1],
        "answerType": "NA",
        "answerText": "I have not used it on tall boots that are
    }
],
"questionText": "Does it work for tall boots that are too tight
"questionType": "descriptive",
"category": "Clothing_Shoes_and_Jewelry",
"asin": "B0010TN0UW",
"review_snippets": [
    "I ordered a pricey pair of fitted ...",
    "For some odd reason my left boot seems tighter then my righ
    "I took off the boots, put on some socks, and lo and behold,
    "..my darn big calves made the last two inches of zipping th
    "..they were very tight across the instep. I thought I ..",
    "I could not even put my feet inside my JS shoes .."
]
```


Ilustración 18: Extracto del dataset AmazonQA.

supervisados. En la ilustración 18 se muestra un extracto del dataset.

4.1.2 SemEval 2015

Este dataset [1] salió precisamente para el Community Question Answering, por lo que lo hace excelente para este trabajo. Salió para resolver la tarea de encontrar la mejor respuesta a una pregunta dada dentro de los foros. Para ello se enfocó la subtarea A, es decir, la búsqueda de las respuestas más acordes a una pregunta dada, de la siguiente manera: Se pretendía clasificar cada una de las respuestas en buena, potencialmente buena o mala en función de su relevancia con la pregunta. Otra subtarea era contestar a respuestas de si o no, aunque en este trabajo no nos centramos en esta.

Todas las preguntas del dataset han sido extraídas del foro Qatar Living[2]. Es un foro sobre preguntas del día a día en Qatar.

[2] https://www.qatarliving.com/

Existen dos datasets homólogos: Uno en inglés, que es el que se usa en este trabajo, y otro en árabe. El dataset en inglés esta guardado en XML con formato UTF-8. El dataset está organizado como un conjunto de preguntas.Cada una de ellas tiene una serie de atributos como en el siguiente ejemplo.

<Question QID="Q1" QCATEGORY="Pets and Animals" QDATE="2009-03-07 19:24:00" QUSERID="U1" QTYPE="YES_NO" QGOLD_YN="Yes">

Estos atributos son los siguientes:

- **QID**: Identificador único de la pregunta.
- **QCATEGORY**: La categoría de la pregunta según la taxonomía de Qatar living, el foro del que se han extraído las preguntas.
- **QDATE**: Fecha en la que fue publicada la pregunta.
- **QUSERID**: Id del usuario que publico la pregunta.
- **QTYPE**: Tipo de la pregunta según si es una pregunta general o una de si o no.
- **QGOLD_YN**: La etiqueta que predecir en la subtarea B. Es decir, si la respuesta a la pregunta es afirmativa o negativa.

La estructura de las preguntas es las siguiente:

<Question ...>

<QSubject> text </QSubject>

<QBody> text </QBody>

<Comment> ... </Comment>

<Comment> ... </Comment>

...

<Comment> ... </Comment>

</Question>

El QSubject es un breve resumen de la pregunta publicada por el usuario, mientras que el QBody corresponde con la pregunta completa.

Cada uno de los comentarios tiene unos atributos como en el siguiente ejemplo:

<Comment CID="Q1_C1" CUSERID="U4" CGOLD="Good" CGOLD_YN="No">

Cada uno de los atributos son los siguientes:

- **CID**: Identificador interno del comentario.
- **CUSERID**: Identificador del usuario que ha publicado el comentario.
- **CGOLD**: El ground truth del comentario. Dice si la respuesta es buena, potencialmente buena o mala para la pregunta. Es la etiqueta a predecir.
- **CGOLD_YN**: Dice si la respuesta está respondiendo positiva, neutral o negativamente a la pregunta. Es para la segunda subtarea del dataset.

Los comentarios a su vez se estructuran de la siguiente manera:

<Comment ...>

<CSubject> text </CSubject>

<CBody> text </CBody>

</Comment>

El CSuject y el CBody funcionan igual que en el caso de las preguntas.

El dataset está dividido en train, validación y test. Cada uno de ellos tienen 2600, 300 y 329 preguntas respectivamente. A su vez, cada uno ellos cuentan con 16541, 1645 y 1976 comentarios respectivamente. Las clases buena y mala están balanceadas. Sin embargo, la clase potencial está muy reducida y probablemente sea más complicada de predecir.

4.1.3 SemEval 2017

Este dataset [4] es la evolución del dataset SemEval 2015 [1]. Sin embargo, en el SemEval 2015, las tareas estaban más enfocadas a tareas más convencionales de Question Answering como por ejemplo selección de respuestas. En el SemEval 2017 se crean subtareas más específicas al problema como por ejemplo la subtarea c. Esta consiste en devolver un ranking de las respuestas que hay en el foro para una pregunta completamente nueva. A diferencia de la tarea que estamos abordando en este trabajo, las respuestas no están unidas a la nueva pregunta, sino que hay que encontrarlas entre las respuestas a otras preguntas previas. A pesar de crearse esta nueva subtarea, se mantiene la tarea que interesa en este trabajo que es la de question-comment simmilarity.

Los comentarios están etiquetados en el dataset de la misma manera que el SemEval 2015, como respuestas buenas, potenciales y malas. Algo interesante es que en el dataset de test no hay respuestas potencialmente buenas, aunque si existen en los datasets de validación y de train. Esto es porque se agrupan, dentro del dataset de test, las respuestas potenciales y buenas. A pesar de ello, en nuestros experimentos sobre esta colección, incluiremos también algunas propuestas centradas en detectar tres clases para estudiar su comportamiento y por homogeneidad con los resultados sobre la colección de SemEval 2015.

El dataset es la unión de las preguntas y las respuestas de la tarea 3 de los SemEval 2015 y 2016 para train y validación, aunque tiene nuevos registros para test. Es por eso por lo que el número de preguntas y respuestas es más elevado que en los otros dos SemEval. Tiene 6959 preguntas entre train y validación y 880 preguntas para test. Por otra parte, el número de comentarios para train y validación son 41908 comentarios y para test son 2930.

La estructura del dataset es la misma que en el caso del SemEval de 2015. Se añaden nuevas etiquetas para diferentes subtareas, que no son relevantes para este trabajo. El resto es igual al SemEval 2015.

4.2 Propuestas

Las propuestas que se han realiando en este trabajo están organizadas de manera incremental para evaluar la contribución de cada propuesta.

Para realizar dicho análisis, se ha planteado un baseline basado en el modelo **Bag of words**. Para ello, en este caso se representa cada comentario y cada pregunta como la frecuencia de las palabras que aparecen en el mismo. Se crea un vector en el que cada número representa las veces que aparece dicha palabra en el texto. Para medir la similitud entre ambas palabras, se calcula el coseno entre sus vectores de representaciones. Esto hace que este método sea no supervisado y más rápido de las propuestas.

Por otro lado, la primera parte del resto de propuestas se han organizado en 2 grupos principales:

- Por una parte, se encuentran todas las propuestas basadas en Bi-Encoders, agrupando en el mismo grupo todas las arquitecturas con la misma arquitectura neuronal.
- Por otra parte, se encuentran las propuestas con una arquitectura de Cross-Encoder.

Además de esta división, también existen propuestas que no se encuentran dentro de estos grupos y se introducirán en una Sección titulada otras propuestas. Dentro de esta se encuentran el uso de **funciones de composicionalidad** y **la mezcla de Cross-Encoders con Bi-Encoders**.

Por último, debido a las peculiaridades de los datasets hay ciertas propuestas que se entrenan utilizando 2 clases en vez de 3. Esto se debe a que en el dataset del SemEval 2017 el conjunto de test solo contiene 2 clases, buena y mala, a diferencia de en su conjunto de entrenamiento que tiene 3 clases. Por este motivo, se plantean distintos experimentos para comprobar si la propuesta obtiene mejor resultados al ser entrenada con 2 o 3 clases .

En primer lugar, las propuestas que usan Bi-Encoder son las siguientes:

1. **Bi-Encoder preentrenado con los pesos de msmarco-distilbert-base-v4:** Esta es la primera propuesta realizada con Transformers. Se usa una configuración de Bi-Encoder, explicada anteriormente, con cada uno de los Transformers entrenados sobre el dataset de Question Answering de msmarco. Se usa un modelo Bert destilado para que ocupe menos y sea más rápido a la hora de la inferencia. Esto hace que este método sea no supervisado, ya que no requiere de entrenamiento.
2. **Entrenar Bi-Encoder con capa de clasificación y pesos de msmarco-distilbert-base-v4: E**n esta propuesta se entrena un Transformer con estructura de Bi-Encoder y luego se usa una última capa para clasificar el resultado de los dos Bi-Encoders. Los pesos que se han usado en este caso han sido los de la arquitectura msmarco-distilbert-base-v4. El número de épocas durante las que se ha entrenado este modelo han sido 10, aunque la época en la que se han guardado los pesos del modelo lo ha determinado el accuracy del dataset de validación. El learning rate utilizado ha sido el típico usado para los fine-tuning de estos Transformers, 2e-5 y un weight decay de 0.01.
3. **Entrenar Bi-Encoder y pesos de msmarco-distilbert-base-v4 como multitask:** En esta propuesta se ha entrenado un Bi-Encoder para realizar la predicción de las clases. La diferencia con respecto al modelo anterior radica en cómo se ha entrenado dicho Bi-Encoder. A diferencia del experimento anterior, en este caso se ha entrenado el modelo para realizar dos tareas simultáneamente. Por un lado, el Transformer ha utilizado el loss del coseno para tratar de crear

un espacio vectorial en el que las respuestas de la misma clase se encuentren cerca. Por otro lado, a la salida de esta tarea se le ha aplicado una red neuronal para realizar clasificación usando cross-entropy loss. Al igual que en todas las propuestas que requieren entrenamiento, el número de épocas que se han utilizado para realizar el fine-tuning ha sido 10, aunque se han guardado los pesos del modelo que mejor actuaba en validación. El learning rate utilizado ha sido el típico usado para los fine-tuning de estos Transformers, 2e-5 y un weight decay de 0.01.

4. **Entrenar Bi-Encoder y pesos de msmarco-distilbert-base-v4 con OVA:** En esta propuesta se estudian si se pueden mejorar los resultados del Bi-Encoder msmarco-distilbert-base-v4 con un ensemble realizado mediante OVA. Como ya se ha explicado, la idea del One Vs All es crear un clasificador por cada para una de las clases. En este caso, se ha realizado el ensemble con los pesos del modelo preentrenado, entrenándolo por un lado como clasificador y, por otro, utilizando la función coseno para calcular la similitud. El número de épocas que se ha realizado en cada uno de ellos han sido 10 épocas, guardándose los pesos de la época que mejor accuracy consiga en validación. El learning rate está fijado a 2e-5 y el weight decay es de 0.01.
5. **Bi-Encoder preentrenado con los pesos de msmarco-distilbert-base-v4 con 2 clases:** En este experimento se ha probado el Bi-Encoder preentrenado con los pesos de msmarco-distilbert-basev4. La diferencia con el anterior experimento en el que usa este Transformer con estos pesos de manera preentrenada es que, en este caso, la clase potencial y la clase buena se unen en una misma clase. De esta manera se trata de solucionar todos los problemas que aporta la clase potencial, ya que en muchos casos no queda clara su distinción con la clase buena.
6. **Bi-Encoder entrenado con los pesos de msmarco-distilbert-base-v4 con 2 clases:** En este caso, se ha cogido el modelo preentrenado del Bi-Encoder del experimento anterior y se le ha introducido un fine-tuning. En este caso, al igual que en el experimento anterior, solo hay dos clases: la negativa y una nueva clase positiva que une la clase potencial y la clase buena. De esta manera, evitamos los problemas de la clase potencial en los experimentos, como se comenta en la propuesta anterior. Se usa el loss de similitud del coseno para aprender embedding en vez de aplicar una capa de clasificación. El número de épocas utilizado en este experimento ha sido 10, aunque se han guardado los pesos del modelo que obtuviesen mejor accuracy en validación. El learning rate es de 2e-5 y el weight decay de 0.01.

Las propuestas que utilizan la arquitectura de Cross-Encoder son las siguientes:

1. **Cross-Encoder entrenado:** En este caso se va a hacer un fine-tuning de dos arquitecturas con sus pesos. Como los tipos de errores son muy similares y la distribución del error también, se van a explicar ambos en el mismo apartado en vez de hacer uno para cada uno. Las dos arquitecturas de Transformer que se van a usar son paraphrase-MiniLM-L6-v2 y stsb-distilbert-base, arquitecturas explicadas en los preliminares. Para entrenarlo, se ha usado un cross-entropy loss y 10 épocas. Para decidir el mejor número de épocas, se ha utilizado el accuracy en validación: El modelo se queda con los pesos que mayor accuracy consigan en el dataset de validación. El learning rate utilizado ha sido el típico usado para los fine-tuning de estos Transformers, 2e-5 y un weight decay de 0.01.
2. **Entrenar Cross-Encoder con OVA:** En esta propuesta se va a entrenar un Cross-Encoder en forma de OVA con los pesos de stsb-distilbert base por un lado y con los pesos de paraphrase-MiniLM-L6-v2 por otro. Lo que permite el OVA (One versus all) es descomponer el problema multiclase en varios problemas binarios en los que se compara una de las clases con el resto. De esta forma, se

crea un ensemble que muchas veces funciona de manera muy satisfactoria. En este caso, al haber 3 clases, se divide el problema en 3 problemas diferentes.

3. **Entrenar Cross-Encoder con 2 clases:** En esta propuesta se ha probado a entrenar un Cross-Encoder como en otro experimento anterior, pero mezclando la clase potencial y la clase positiva. De esta manera, se pasa de tener 3 clases a tan solo tener 2 y desaparece la clase potencial, consiguiendo así los problemas generados a la hora de diferenciar entre ambas clases. El número de épocas utilizadas han sido 10 en todos los casos, guardando los pesos que mejor accuracy diesen en validación. Por otra parte, el learning rate en ambos casos ha sido de 2e-5 y el weight decay de 0.01.

Por último, las propuestas que no entran dentro de ninguno de los grupos son las siguientes:

1. **Función de composicionalidad basada en teoría de la información:** Además de probar modelos basados en Deep Learning, en este trabajo se han probado modelos que utilizan funciones matemáticas para combinar embeddings. De esta manera, se prueba la utilidad de una nueva de una nueva medida de composicionalidad en esta tarea. Se está hablando de la función de composicionalidad de Amigo et al. [85], que permite combinar los embeddings de varias palabras para crear el vector semántico de toda la oración. Los embeddings que se han usado para realizar dicha combinación han sido los vectores de Word2Vec con dimensión 200.
2. **Entrenar mezcla Bi-Encoder para separar clase buena-potencial y mala y Cross-Encoder para separar clase buena de la clase potencial:** En esta propuesta se han mezclado las dos arquitecturas de Transformers para realizar un ensemble. Por un lado, se encuentra la estructura en forma de Bi-Encoder con el coseno. Esta se ha usado para dividir los ejemplos entre los ejemplos malos y los buenos o potenciales. Por otra parte, se ha entrenado un Cross-Encoder para separar la clase potencial de la buena. En este Cross-Encoder se han usado dos modelos preentrenados diferentes para realizar el fine-tuning: El stsb-distilbert-base-v4 y el paraphrase-MiniLM-L6-v2. Se pueden ver las métricas de ambos modelos en las Tablas 117 y 119 respectivamente. Se ha entrenado durante 10 épocas, solo salvando los pesos del modelo que diese mejor accuracy en validación. Ambos modelos se han entrenado con un learning rate de 2e-5 y con un weight-decay de 0.01.

4.3 Métricas

En este trabajo se han usado varias métricas dependiendo de la intención de la tarea asociada. Estas han sido las propias que se han utilizado sobre cada dataset analizado en la sección anterior. Por un lado, sobre la coleccioón de SemEval 2015 se trata de categorizar la respuesta entre unos tipos dados, de modo que se trabaja sobre un problema de clasificación y la métrica va orientada en esta dirección. Por otro lado, si queremos ordenar las respuestas acordes a su a adecuación a la pregunta, como en el SemEval 2017,se trabaja sobre un problema de ranking y se tienen que usar unas métricas asociadas a este objetivo. En el caso del dataset de Amazon, no existe ningún precedente de abordar esta tarea y se enfocará como en estos otros dos datasets con las mismas métricas.

Dicho esto, las métricas usadas en este trabajo de fin de máster son las siguientes:

1. ***Mean Average Precision*** **(MAP)**: Dado un conjunto de búsquedas, se define cómo la media de la precisión media de cada una de las búsquedas. La precisión media se calcula como:

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \quad (46)$$

   Donde R_n y P_n son el recall y la precisión en el umbral n. Esta métrica se usa para calcular si los documentos o las respuestas obtenidas están correctamente ordenadas. Es decir, la vamos a utilizar para evaluar la creación de rankings de respuestas. Esta es la métrica principal en el SemEval 2017.
2. ***Accuracy***: El accuracy de un clasificador esta determinado por el porcentaje de predicciones correctamente etiquetadas. Se calcula como:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (47)$$

   Donde TP son los verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos. Por tanto, vamos a usar esta métrica para evaluar las propuestas que realizan clasificación.
3. **F1-score:** Es la media armónica entre la precisión y el recall. Se define como:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \quad (48)$$

   Donde TP son los verdaderos positivos, FP los falsos positivos y FN los falsos negativos. Vamos a usar esta métrica para evaluar las propuestas que realizan clasificación. Esto se debe a que da una mejor visión del comportamiento de un clasificador que el accuracy, evitando sesgar los resultados por un dataset desbalanceado. Esta es la métrica principal usada en el SemEval 2015.

4.4 Conclusión del marco experimental

En esta sección se han definido las bases de la experimentación de este trabajo. Esto sirve para poder comprender la experimentación llevada a cabo de una manera más profunda. Lo presentado han sido las propuestas de manera general para poder entenderlas de una manera global, los datasets utilizados de los que van a depender los resultados obtenidos y las métricas utilizadas.

Los datasets que se han utilizado han sido tres: AmazonQA, SemEval 2015 Task 3 y SemEval 2017 Task 3. Los dos datasets de SemEval se han utilizado debido que al haber un desarrollo previo sobre la misma temática que el TFM es interesante comparar los resultados obtenidos con los anteriores. De esta forma, se puede analizar qué tan bien funcionan las propuestas de este trabajo. Por otro lado, el dataset de AmazonQA se ha utilizado porque no está diseñado particularmente para esta tarea, pero es sencillo adaptarlo.

Por último, las métricas usadas se han seleccionado por los congresos de los que se han extraído los datasets. En estos congresos son especialistas a la hora de evaluar tareas como la de este trabajo y por tanto su selección de métricas es muy adaptada y es complicado encontrar mejores. Es cierto que para el ranking hay otras métricas buenas como el mean reciprocal rank (MRR) pero no veíamos necesario incluir más métricas de las utilizadas y explicadas en esta sección.

5. Análisis de resultados

En esta sección se van a evaluar con detalle las propuestas que se han presentado en el capítulo anterior, realizando un análisis de los resultados de cada una de ellas. La estructura del análisis que se va a realizar va a ser la siguiente:

1. Primero se explican aspectos generales del modelo junto con las métricas generales.
2. Posteriormente se realiza un análisis por dataset. Este análisis consta de 3 partes:
    a. En primer lugar, se analiza la matriz de confusión.
    b. Luego se muestran las métricas por clase.
    c. Finalmente se muestran 4 ejemplos para comprobar de manera más visual cuales son los principales errores que comete la propuesta.

En ciertas propuestas, la evaluación se realiza sobre 2 clases. Por tanto, la matriz de confusión no tiene sentido ni las métricas por clase. Por tanto, estos experimentos estarían organizados de la siguiente manera:

1. Una exposición de los resultados del experimento en los 3 datasets junto con una explicación breve de la propuesta.
2. Por cada dataset, una explicación del rendimiento de la propuesta en cada dataset y unos ejemplos fallados para ver los errores más comunes.

5.2 Baseline: Bag of words

Este modelo es el que peor funciona. Esto es debido a que una respuesta buena no tiene por qué contener las mismas palabras que la pregunta. Es más, que una respuesta tenga las mismas palabras que la pregunta puede ser sinónimo de que no es una respuesta buena al no aportar información nueva.

La dificultad proviene al separar las diferentes clases según su coseno. Para este experimento se ha tomado los resultados del coseno por debajo de 0.33 como clase negativa, los resultados entre 0.33 y 0.66 como clase potencial y los resultados por encima de 0.66 como clase buena.

Los resultados en los diferentes datasets en este experimento se puede ver en la Tabla 2.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.22142636	0.401315789	0.34297415	0.172533517	0.811084474
SemEval 2017	0.216740548	0.441818182	0.30534141	0.351383543	0.675379321
Amazon QA	0.323323859	0.58931097	0.33152508	0.326464857	0.743852377

Tabla 2: Resultados macro de BOW.

Como se puede observar, los mejores resultados se obtienen sobre el dataset de AmazonQA, aunque tampoco es muy bueno. En el caso de SemEval 2015 y 2017, comparándolo con los resultados del congreso de otras soluciones, es un resultado muy malo. De hecho, en el baseline (Poner todas las respuestas como buenas) del SemEval 2015 se obtuvo un F1-score de **0.22142636** y en el SemEval 2017 (Ordenar las respuestas de manera aleatoria) se obtuvo un Map de **0.623**.

SemEval 2015

La matriz de confusión del baseline sobre esta colección se puede observar en la Tabla 3. Se observa que casi todos los ejemplos han sido detectados como negativos y muy pocos han sido predichos como positivos. Esto se debe principalmente a dos motivos: Porque las respuestas no contienen necesariamente las mismas palabras que las preguntas y porque los umbrales han sido fijados suponiendo que la distribución de las 3 clases en test está balanceada.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	782	25	5
	Potencial	156	11	0
	Positive	919	78	0

Tabla 3: Matriz de confusión BOW en SemEval 2015.

Debido a estas predicciones casi en su mayoría negativas, las métricas por clase solo salen bien en la clase negativa, como se puede observar en la Tabla 4. Como no hay ningún resultado predicho como positivo que realmente sea de clase positiva, las métricas de la clase positiva son nulas.

	F1-score	Recall	Precision
Clase negativa	0.58598727	0.96305419	0.42110932
Clase potencial	0.07829181	0.06586826	0.09649123
Clase positiva	0	0	0

Tabla 4: Métricas por clase en SemEval 2015.

Algunos ejemplos mal predichos sobre la colección del SemEval 2015 por el modelo son los representados en la Tabla 5. En el ejemplo 1, la repuesta es exactamente igual que la pregunta. Eso hace que el coseno sea muy alto entre ambos, aunque realmente su validez como respuesta es nula. En el segundo ejemplo, se piden aclaraciones sobre la pregunta formulada con otra pregunta. Al compartir muchas palabras en común, el coseno es muy alto, aunque responder una pregunta con otra pregunta no da mucha información sobre la pregunta formulada. En el tercer y cuarto ejemplo, se responde claramente a la pregunta con un si y una explicación. Sin embargo, como se usan otras palabras su coseno es muy bajo y son clasificados como mala.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	I saw a little girl running by the streets , and she had a cat attached to her .....is that normal in this country?	I saw a little girl running by the streets , and she had a parent attached to her .....is that normal in this country?\n\n	Mala	Buena
2	I am looking for a restaurant with a nice view preferably a sea view and much expensive???\nthis is urgent!\nthank you	Are you looking for the most expensive restaurant with a sea view?	Mala	Buena
3	I have Windows Vista Home Premium edition and would like to upgrade, not since i want to, since i am fed-up with Vista-mania.\n\nIf any1 of u guys has upgraded, i can take your advice while i will upgrade mine.\n\nThanks	Yes, you can upgrade directly from Vista to Windows 7. No need for clean-install. Very easy if you have the installation disk.	Buena	Mala
4	Hello Friends,\n\nMy family is here in Qatar in visit visa for last 5 months in my personal sponsorship. Now they need to go back and come again on visit visa after 3-4 months. Do i need to get the exit permit for them?? Your answers will be highly appretiated.\n\nThanks in Advance.	Yes Ofcourse You Must need the Exit Permit	Buena	Mala

Tabla 5: Ejemplos mal clasificados por el BOW en SemEval 2015.

SemEval 2017

La matriz de confusión se puede observar en la Tabla 6. Se puede apreciar que no hay ningún ejemplo en el test de la clase potencial. Sin embargo, se han predicho casi 1000 ejemplos como potenciales. Por otra parte, se tiene el mismo problema que en el SemEval 2015: los rangos entre clases, en los que según el coseno se clasifica como una clase u otra, están mal definidos y por tanto no se clasifica casi ningún ejemplo como positivo. Además, como ya se ha comentado antes y se ha podido ver en los ejemplos, compartir muchas palabras no implica que la respuesta responda mejor a la pregunta, sino todo lo contrario.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3869	359	15
	Potencial	0	0	0
	Positive	3942	596	19

Tabla 6: Matriz de confusion SemEval 2017 en BOW

Debido a estas predicciones, las métricas por cada clase son las que se muestran en la Tabla 7. Como no hay ejemplos de la clase potencial, sus métricas son 0. Se puede ver que los resultados son bajos para todas las métricas y que casi se pueden mejorar clasificando los ejemplos de manera aleatoria.

	F1-score	Accuracy	Recall
Clase negativa	0.64194458	0.91185482	0.4953271
Clase potencial	0	0	0
Clase positiva	0.00827706	0.00416941	0.55882353

Tabla 7: Métricas por clase de BOW en SemEval 2017.

Se representan en la Tabla 8 algunos ejemplos mal clasificados por el método de bag of words sobre la colecció SemEval 2017. Se puede ver que las respuestas clasificadas como malas, realmente no comparten palabras con las preguntas y por eso resultan erróneas. En las respuestas predichas como buenas y que son realmente buenas, se puede ver que o son respuestas que no aportan ninguna información o que repiten de alguna manera palabras de la pregunta.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	about freelance visa?	never heard about freelance visa.	Mala	Buena
2	How many of us seen this and where.....?	seen.....?	Mala	Buena
3	Will I get a GCC ban Ban if I go on leave and doesn't come back during my contract period in Qatar..? please give me an answer...	ur right PP only ban here no probs in going to other GCC	Buena	Mala
4	Is there anybody who can tell which bank is the best to invest here in Doha?	Amongst foreign banks HSBC is best and from local, CBQ and Doha Bank excels in customer services. Try them out. Good luck.	Buena	Mala

Tabla 8: Ejemplos del SemEval 2017 clasificados por el Bag Of Words

AmazonQA

Se puede ver la matriz de confusión en la Tabla 9. Se puede ver que al ser un dataset mucho más grande, la clase potencial la acierta más en proporción que en los casos anteriores. Sin embargo, falla catastróficamente al predecir la clase potencial. Los motivos de esta clasificación errónea son los mencionados en el caso de los otros datasets.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	195236	44988	5658
	Potencial	72059	16934	1692
	Positive	19026	4669	332

Tabla 9: Matriz de confusion AmazonQA en BOW

Consecuentemente con estas predicciones, las métricas por cada clase se pueden observar en la Tabla 10. En este caso, al menos ninguna clase tiene métricas nulas. A pesar de esto, sí que se puede ver que las métricas empeoran desde la clase negativa hacia la clase positiva paulatinamente. Esto es por las causas mencionadas en los datasets anteriores.

	F1-score	Accuracy	Recall
Clase negativa	0.73368996	0.79402315	0.68187803
Clase potencial	0.21534118	0.1867343	0.25429863
Clase positiva	0.02094043	0.01381779	0.04321791

Tabla 10: Métricas por clase de BOW en AmazonQA

Por último, se observan en la Tabla 11 ejemplos mal clasificados del dataset AmazonQA. Se puede observar en el ejemplo 3, como ya hemos visto anteriormente también que no se reconoce una respuesta afirmativa. En el ejemplo 1, se responde de una manera muy elaborada a la pregunta con una opinión, esto es clasificado como malo, aunque la clase real es buena porque no hay muchas palabras en común. A su vez, en el ejemplo 2, la contestación es la reformulación de la pregunta en forma de respuesta. Esto tiene muchas palabras en común, se predice como bueno, aunque la verdadera clase es mala. En el ejemplo, se responde correctamente a la pregunta, aunque la clase real es mala y se predice como potencial. Esto pasa a veces en este dataset, al estar etiquetado por seres humanos en función de su opinión subjetiva, la respuesta puede parecer buena, aunque esta etiquetada como mala.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	what color would you recommend for nighttime walking?	Jason, When I purchased the NiteIze I bought several because I didn't know how bright the different colors would be.The white is the brightest and the red is better when it is blinking.Hope this answer helps.	Buena	Mala
2	Is there a disconnect between the cable and the strap?	There is a disconnect between the cable and the strap	Mala	Buena
3	do they work in belgium?	Yes,it will.	Buena	Mala
4	Does anyone know the difference between the "digital timer" and "cooks digital timer" Two prices, two names, looks the same	my digital timer has kitchenaid on the face and is black. the same timer comes in red. the times goes to 9 hours and 59 minutes. works like a charm and the buzzer is loud enough to hear in the next room.	Mala	Potencial

Tabla 11: Ejemplos clasificados incorrectamente por BOW en AmazonQA.

5.3 Propuestas basadas en Bi-Encoders

Las propuestas de esta Sección son las basadas en Bi-Encoders. Como se ha explicado en la sección de propuestas del marco experimental, se encuentran tanto las propuestas que predicen 3 clases como las que predicen 2 clases. Para diferenciarlas, se explicará en el inicio del desarrollo de la propuesta.

5.3.1 Bi-Encoder preentrenado con los pesos de msmarco-distilbert-base-v4

Al ser un modelo entrenado para calcular similitud semántica y no específico entre pregunta y respuesta, los resultados no tienen por qué ser óptimos. El Transformer realiza otra tarea que no es la que requiere la nuestra. Sin embargo, es interesante ver como de buenos son los resultados para comprobar si la tarea general de encontrar una similitud semántica se aproxima a la similitud entre pregunta y respuesta.

Este Transformer crea un embedding de la pregunta y de la respuesta de forma separada a los que posteriormente se le realiza el coseno. Si el coseno es muy alto es que su similitud es alta por lo que se le asigna buena o potencial. Por otro lado, si el coseno es muy bajo, la relevancia para el usuario se reduce y esa respuesta no sería candidata para ser devuelta por el sistema.

Los umbrales que se han tomado para discernir entre las diferentes clases han sido los mismos que en el caso del baseline. Se ha hecho así para ver si en este caso los umbrales funcionan tal y como están definidos. Los resultados de este experimento para los 3 datasets son los que se muestran en la Tabla 12. Se puede observar que son mejores que los obtenidos por el baseline, con un F1-score de **0.22142636** y en el SemEval 2017 con un Map de **0.684**. Una de las razones por las que ocurre esto es porque la clase potencial no tiene ningún miembro. Se explicará con más detalle cuando se analice dicho conjunto de datos.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.359465793	0.40840081	0.425902713	0.459904093	0.829555613
SemEval 2017	0.212002665	0.332045455	0.229110673	0.371656277	0.696995915
Amazon QA	0.33041919	0.474195355	0.345336877	0.34415276	0.766834852

Tabla 12: Métricas de los resultados de los datasets con el Bi-Encoder preentrenado masmarco-distilbert-base-v4

SemEval 2015

La matriz de confusión del dataset SemEval 2015 aparecen en la Tabla 13 y las métricas por cada una de las clases aparecen en la Tabla 14. Como se puede ver, en este caso, la clase con más errores es la clase potencial. Sin embargo, las métricas son más equilibradas que en el caso del baseline. La clase negativa sigue siendo la mayoritaria pero no parece que el umbral sea un gran problema a la hora de clasificar, sino que ya es cosa del modelo.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	544	203	65
	Potencial	80	69	18
	Positive	297	506	194

Tabla 13: Matriz de confusión de SemEval 2015 con el Bi-Encoder preentrenado msmarco-distilbert-base-v4

	F1-score	Recall	Precision
Clase negativa	0.62781304	0.66995074	0.59066232
Clase potencial	0.14603175	0.41317365	0.08868895
Clase positiva	0.30455259	0.19458375	0.70036101

Tabla 14: Matriz de confusión de SemEval 2015 con el Bi-Encoder preentrenado msmarco-distilbert-base-v4

Se puede ver también que la clase positiva se clasifica en su mayor parte como potencial. Esto es porque las respuestas no tienen por qué tener similitud semántica tal cual con la pregunta.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Does anyone know where or have a contact for the chevrolet service centre?	Thank you MD, rock, vignale, nafaz and raynermac for your responses, greatly appreciated.	Mala	Buena
2	Will it be a ny problem if i sign in hindhi language in indian passport\n'	It should be HINDI...right?	Mala	Buena
3	.\nI have been charged QR.2800 on a 3 year R/P renewal whilst the tariff says a 20% discount is available for 3 year renewals! Any one knows what is the calculation for 3 year discounted R/P charges ? (No explanation received from counter staff)	Obviously you get 20% on the third year only. Instead of 1.000 you pay only 800.	Buena	Mala
4	My wife is currently on vacation in Australia to visit her family. I am planning to surprise her by sending a dozen of red roses. Any trusted website?	gardenia florists are an agent here for interflora.....maktab hollandi are also very good and they also are partners of Interflora...	Buena	Mala

Tabla 15: Ejemplos clasificados incorrectamente en SemEval 2015 por el Bi-Encoder preentrenado msmarco-distilbert-base-v4

En la Tabla 15, se pueden ver ejemplos mal clasificados por el modelo en el SemEval 2015. Se puede ver como las respuestas que mejor valora son las que tienen mayor similitud con la pregunta. Realmente, el modelo no ha aprendido a priorizar las respuestas que aportan información, ya que es un modelo no supervisado.

SemEval 2017

En la Tabla 16 se puede ver la matriz de confusión del modelo en el SemEval 2017 y en la Tabla 17 las métricas consecuentes con esta matriz de confusión. Como ya se ha explicado posteriormente, la clase potencial no tiene ningún ejemplo por lo que sus métricas son nulas. Se puede ver que la clase que mejor predice es la negativa. De hecho, casi todos los ejemplos los predice como clase negativa. Esto se debe probablemente a que las respuestas muy similares a la pregunta no aportan ninguna información y por tanto su coseno es muy alto, pero su clase es negativa o como mucho potencial.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	2840	1341	62
	Potencial	0	0	0
	Positive	2366	2109	82

Tabla 16: Matriz de confusión de SemEval 2017 con el Bi-Encoder preentrenado msmarco-distilbert-base-v4

	F1-score	Recall	Precision
Clase negativa	0.60112181	0.66933773	0.54552439
Clase potencial	0	0	0
Clase positiva	0.03488619	0.01799429	0.56944444

Tabla 17: Matriz de confusión de SemEval 2017 con el Bi-Encoder preentrenado msmarco-distilbert-base-v4

En la Tabla 18, se pueden ver ejemplos mal clasificados por el clasificador en el SemEval 2017. Se puede ver que el ejemplo 1 se responde correctamente a la pregunta, pero no le asigna su clase sino una potencial. Este no sería un fallo muy importante ya que el modelo podría entender que la respuesta es ironica o es una mentira. Sin embargo, en el ejemplo 4 se ve un gran error. Se pregunta sobre un salario mínimo para emigrar a Doha y se responde, pero el sentence bert no entiende que esa respuestas sea satisfactoria para la pregunta en cuestión, por tanto se clasifica dicha respuesta como negativa. Los errores se deben a lo mismo, que al no estar entrenado el modelo entiende que, si la respuesta no es similar, su importancia como respuesta es baja y la clasifica como mala.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	can a tourist visa drive a car here in doha? what requirements and for how long?	everyone can drive here. :)	Buena	Potencial
2	Hello everyone, I'm already here in Qatar a month ago under business visa with my sponsor. Do I still need an NOC in order to apply for another company here in Qatar if ever I decide to go home on the 3 month period of the business visa? Thanks.	Hi every one I was cabin crew in Qatar airways I left the company 10 month Ago .now I m sales manager in Dubai I must go To Doha for a job for 15 days..do I need NOC?and is it possible To get it or impossible???plz answer me as soon as possible Thank you	Mala	Buena
3	Hey everyone, Any one knows of nice, new, unique baby names (girls & boys) of muslim or pakistani origin ?	Maham Aleya Faryal Palwasha	Buena	Mala
4	Is there any salary limitations for bringing wife on Family visit visa ?? Morever while staying on visit visa is it possible to transfer into Residence >> Appreciate your help in this regards	Qrs.4,000/-	Buena	Mala

Tabla 18: Ejemplos clasificados incorrectamente en SemEval 2017 por el Bi-Encoder preentrenado msmarco-distilbert-base-v4

AmazonQA

En las Tablas 19 y 20 se muestran la matriz de confusión y las métricas por clase de AmazonQA. Como se puede ver, el patrón de que se predice la clase negativa más que el resto se reafirma. Esto es por las razones que se han explicado para los otros datasets y se puede ver en las métricas que la clase negativa es la que mejor resultados obtiene mientras que la positiva tiene los peores resultados.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	127513	105731	12638
	Potencial	43926	42238	4521
	Positive	10234	12552	1241

Tabla 19: Matriz de confusión de AmazonQA con el Bi-Encoder preentrenado msmarco-distilbert-base-v4

	F1-score	Recall	Precision
Clase negativa	0.59647531	0.51859429	0.70188195
Clase potencial	0.33628178	0.46576611	0.26313068
Clase positiva	0.05850048	0.05165023	0.06744565

Tabla 20: Matriz de confusión de AmazonQA con el Bi-Encoder preentrenado msmarco-distilbert-base-v4

En la Tabla 21, se muestran unos ejemplos mal clasificados del dataset AmazonQA. Se puede ver que el ejemplo 1 se clasifica como respuesta buena porque tiene mucha semántica en común, habla sobre niños y utiliza el verbo fit. Sin embargo, los usuarios han clasificado esta respuesta como negativa debido a su carácter más personal y no tan objetivo. Por el contrario, en el ejemplo 3 se responde con detalle a la pregunta explicando cuál es el tamaño exacto del producto, pero al no tener muchas palabras en común o expresiones, se clasifica como mala. Se puede ver que el modelo no puede ser capaz de saber si una respuesta es buena o mala porque carece de las reviews para determinarlo; se verá en los siguientes experimentos con entrenamiento si mejoran están métricas o no.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Does this car fit one or two kids?	My kids are 6 and 2. So it fits for two.	Mala	Buena
2	how many screws or bolts attach each leg to the table?	There are 2 bolts per leg.	Mala	Buena
3	what is the size	Outside dimensions are 20.25 x 10.5 x 1, grill surface is approximately 17 x 10.25 with about .25 lip on the edge.	Buena	Mala
4	If I want to use this to spread on my lawn for Flea and tick control, how much will I need for an acre of land?	My research online has recommended 2 pounds per acre. Some have said that a push style spreader worked great. One cautioned to be sure to get food grade DE as they experienced very bad reactions.	Buena	Mala

Tabla 21: Ejemplos clasificados incorrectamente en AmazonQA por el Bi-Encoder preentrenado msmarco-distilbert-base-v4

5.3.2 Entrenar Bi-Encoder con capa de clasificación y pesos de msmarco-distilbert-base-v4

Para entrenar un Bi-Encoder con una capa de clasificación se usa un cross entropy loss al final de la capa de clasificación. A la capa de clasificación, para realizar la clasificación de ambos embeddings simultáneamente, se le introducen los dos embeddings concatenados unido al valor absoluto de la resta de ambos. Esta idea se extrae del trabajo de Reimers et al. [100].

Lo positivo de esta aproximación respecto a la preentrenada es que esta sí que aprende la relación que se está buscando. Uno de los problemas más recurrentes con el Bi-Encoder preentrenado era que no entendía bien cuál era exactamente la relación buscada al aplicar la similitud del coseno. Con el entrenamiento y una capa de clasificación sustituyendo a la similitud del coseno evitamos esto.

Una mejora con respecto al Cross-Encoder es que la predicción es mucho más rápida porque se puede guardar un embedding de la pregunta y uno por cada respuesta, y lo único que hay que hacer por cada pareja es aplicar la capa de clasificación.

En la Tabla 5, se pueden ver las métricas globales por cada uno de los datasets. Se puede ver cómo el entrenamiento sirve para algo ya que todas las métricas aumentan con respecto a las métricas de los sistemas no supervisados.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.5901513	0.73076923	0.5919335	0.59166607	0.93160157
SemEval 2017	0.52580292	0.72147727	0.48235858	0.57977312	0.88513471
Amazon QA	0.3654387	0.58084161	0.36398209	0.37066445	0.7623257

Tabla 22: Métricas de Bi-Encoder con capa de clasificación para todos los datasets.

SemEval 2015

En la Tabla 57 se muestra la matriz de confusión de la red en el dataset SemEval 2015. Como se puede observar, las clases positiva (o buena) y la clase negativa (o mala), son las clases que mejor se predicen. Por el contrario, en la clase potencial comete más fallos de los que acierta. Esto es debido a que probablemente la clase potencial sea difusa y el modelo no entienda bien como diferenciarla de las otras dos clases. Posteriormente, en otros experimentos, se intentará ver si uniendo las clases potencial y positiva se mejora la eficacia de los modelos. Por otro lado, en la Tabla 58, se pueden ver las métricas de este modelo sobre SemEval 2015. Reafirmando lo dicho, este modelo funciona de manera excepcional en las clases negativa y potencial, pero falla en la clase potencial.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	604	87	121
	Potencial	47	38	82
	Positive	100	95	802

Tabla 23: Matriz de confusión de Bi-Encoder con capa de clasificación de SemEval 2015.

	F1-score	Recall	Precision
Clase negativa	0.77287268	0.74384236	0.80426099
Clase potencial	0.19638243	0.22754491	0.17272727
Clase positiva	0.8011988	0.80441324	0.79800995

Tabla 24: Métricas por clase de Bi-Encoder con capa de clasificación para SemEval 2015.

En la Tabla 59, se pueden observar ciertos ejemplos mal predichos por el modelo en el SemEval 2015. En el ejemplo 1, se puede ver que la respuesta concuerda de manera perfecta a la pregunta, aunque parece que hay cierto de ironía al final de la respuesta. Esto hace que esté clasificada como potencial, aunque al parecer del modelo es una respuesta buena. Este fallo puede caber dentro de la subjetividad del etiquetador.

En el ejemplo 2, se responde a la pregunta con una sugerencia sobre cómo puede ahorrar dinero. El usuario le responde correctamente. Sin embargo, el modelo predice la clase como mala. Esto se produce porque el modelo no entiende la pregunta ni que significan las siglas QL.

En el ejemplo 4, se responde sobre si un nombre es femenino. De igual modo que en el ejemplo 2, la respuesta es una palabra que el modelo no consigue entender.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	I saw a little girl running by the streets , and she had a cat attached to her .....is that normal in this country?	Unimaginable! A cat who is attached to a girl using a "leech"...hahahahaha.	Potencial	Buena
2	So, I have seen that I have been spending money more than usual. So, I need to cut back on expenses!!\n\nWhat are your tips on saving money here?	stick on QL .... its free !	Buena	Mala
3	Hi Everyone,\n\nJust moved here a few months ago and i'm thinking about buying a new car, more specifically a Jaguar XF.\nDoes anyone have any idea or recommendations about Jaguars in this country ??	lol Khattak...\n\nand no Fuel Tank....:)	Buena	Mala
4	Tan Sze Peng is it a female name?	donno !	Mala	Buena

Tabla 25: Ejemplos mal clasificados por el Bi-Encoder con capa de clasificación para SemEval 2015.

SemEval 2017

En las Tablas 60 y 61 se puede ver la matriz de confusión y los resultados por clase respectivamente del dataset SemEval 2017. Se puede ver, que al igual que en el SemEval 2015, las clases positiva y negativa son las que mejor predichas están y están muy bien predichas en comparación con otros modelos previamente expuestos. Sin embargo, hay ejemplos que asigna a la clase potencial sin ser de ella. Esto se debe a la ambigüedad sobre la clase potencial que parece que depende más de la interpretación de un humano que de un criterio objetivo.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3315	467	461
	Potencial	0	0	0
	Positive	490	1033	3034

Tabla 26: Matriz de confusión de Bi-Encoder con capa de clasificación de SemEval 2017.

	F1-score	Recall	Precision
Clase negativa	0.82380716	0.78128683	0.87122208
Clase potencial	0	0	0
Clase positiva	0.75360159	0.6657889	0.86809728

Tabla 27: Métricas por clase de Bi-Encoder con capa de clasificación para SemEval 2017.

En la Tabla 62, se pueden ver ejemplos mal clasificados por el Bi-Encoder en el dataset SemEval 2017. En el ejemplo 3, se falla porque la pregunta es muy compleja. Se pregunta sobre a qué país pertenece Kerala y en la respuesta se le responde que a United Kerala. El modelo no ha entendido la referencia y por tanto la clasifica como mala a pesar de ser de clase positiva.

En el ejemplo 2, se clasifica la respuesta como buena, aunque la clase de la etiqueta es clase positiva. Sin embargo, parece que se responde correctamente a la pregunta. Este es un fallo de etiquetado que se puede dar a menudo a lo largo del dataset y por el cual se debe tener cuidado.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Anybody knows where to play badminton in Doha?How much is the fee? I heard its crowded in QBC.other location? Anybody knows anybody plays badminton & would like to play with other team? Anybody friendly enough to play with?? on weekends?? Me and my friends are trying to lose weight and we find going to the gym boring! Thanks thanks!! :)	I am a good player , so i would like to join if it is possible,,,but i dont know any place to play it, Thankes	Buena	Mala
2	Can anyone help me in giving some information regarding Qatar Prometric exam for Nurses and what the requirement to sit for the exam?	https://www.prometric.com/en-us/Pages/home.aspx here schedule the test online and pay using a credit card.. it's a 70 items test and you have to get at least half to pass. You'll get the result right after the exam and it's valid for 3 years..:)	Mala	Buena
3	Whenever i ask a keralite if he is from india his answer will be no am from Kerala,so am wondering if Kerala is not part of India and its in the US or Thialand or wherever,I feel there is so much hate between keralite and other part of India specialy Mumbai and Delhi? Also in a Company where the Manager is a keralite the preference of hiring emlpyees goes to Keralite only and not other indian.. Any Idea?	Kerala is part of UK..... Dont miss interpret... its not United Kingdom.... its United Kerala.....:) :).	Buena	Mala

4	I'm checking a flight in December for Bangkok and Qatar Airway is quoting me the following: Fare: 3180 Taxes: 780 Total: 3960QAR I then checked with Emirates Airlines and they quote: Airfare: 3030 Taxes: 120 Total: 3150 QAR 810 QAR cheaper with Emirates Airlines than with Qatar Airways! Why is Qatar Airways charging 780 taxes and Emirates only charging 120 taxes? Why is Qatar Airways such a rip-off compared to other airlines?	If you try ETIHAD airlines you will be surprised from Manila to Doha Qatar airways = over 700 USD from Manila to Doha Etihad airways= 570 USD	Mala	Buena

Tabla 28: Ejemplos mal clasificados por el Bi-Encoder con capa de clasificación para SemEval 2017.

AmazonQA

En las Tablas 63 y 64 se pueden ver la mátriz de confusión y las métricas por clase del dataset AmazonQA. Se puede apreciar que está claramente desbalanceado, priorizando de una forma cuantiosa la clase negativa sobre el resto. De hecho, la clase positiva es la que peores resultados obtiene. Esto probablemente se deba a las características de dataset desbalanceado que caracterizan a AmazonQA.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	1866460	48733	10689
	Potencial	65339	20369	4977
	Positive	15136	6272	2619

Tabla 29: Matriz de confusión de Bi-Encoder con capa de clasificación de AmazonQA.

	F1-score	Recall	Precision
Clase negativa	0.72719898	0.75833123	0.69852211
Clase potencial	0.24532244	0.22461267	0.27023907
Clase positiva	0.12379467	0.10900237	0.14323216

Tabla 30: Métricas por clase de Bi-Encoder con capa de clasificación para AmazonQA.

En la Tabla 63, se pueden observar cuatro ejemplos incorrectamente predichos por el modelo en el dataset AmazonQA. En el ejemplo 1, la pregunta es la palabra color. El usuario entiende a que se refiere y contesta de manera correcta, aunque el modelo predice la clase negativa. Esto predicción errónea probablemente sea porque falta contexto para responder bien dicha pregunta.

En el ejemplo 2 y 4, la pregunta hace referencia a si el producto tiene o no leche y si es vegetariano. En ambas respuestas se responde de una manera correcta, se predice como clase positiva, aunque esta etiquetada como clase negativa. Esto se produce debido al mal etiquetado del dataset e influye en los resultados de evaluación del sistema.

En el ejemplo 3, se responde correctamente a la pregunta, aunque el usuario indica cierta incertidumbre. Esto es entendido por el modelo y se indica como potencial. Sin embargo, la clase real es clase mala. Esto puede ser porque realmente ese dato no se corresponda con la review del producto y sobre lo que se dice en ella. Esto puede haber hecho que los usuarios hayan categorizado esta respuesta como clase mala.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	color	I have purchased this listing at least 3-4 times... Color varies. I have never gotten more than 2 of the same color per 5 pcs ordered.. But the colors makeup is different every time.	Buena	Mala
2	Is this product milk free and vegetarian?	It is made from organic plant sources and it lactose free.	Mala	Buena
3	Does this car fit one or two kids?	I would say only one child.	Mala	Potencial
4	Is this product milk free and vegetarian?	Yes, it just tastes awful.	Mala	Buena

Tabla 31: Ejemplos mal clasificados por el Bi-Encoder con capa de clasificación para AmazonQA.

5.3.3 Entrenar Bi-Encoder y pesos de msmarco-distilbert-base-v4 como multitask

Este tipo de forma de entrenar un modelo se llama multitask, ya que trata de resolver dos tareas que son muy similares simultáneamente. En este caso, parece intuitivo que la tarea de acercar en el espacio vectorial los pares pregunta-respuesta parecidos es compatible con la tarea de clasificar dichos pares según sus clases. Este experimento además de tratar de conseguir una buena métrica pone de manifiesto si estas dos tareas son compatibles o no.

La diferencia con respecto a la propuesta de un Bi-Encoder entrenado únicamente para la tarea de clasificación es que en este caso se puede utilizar para resolver las dos tareas: La de clasificar la clase correcta de un par de pregunta y respuesta y la de ver cuál es la similitud semántica entre pregunta y respuesta. Esto puede ser usado incluso para realizar un ensemble, mejorando así los resultados.

Con respecto al Cross-Encoder, esta solución mejora en velocidad, pero pierde relaciones entre pregunta y respuesta ya que cada una de ellas va a un encoder independiente.

Se han sacado las predicciones y las métricas de dos maneras. En la Tabla 66 se pueden ver las métricas en los datasets dada por la clasificación del Bi-Encoder. Esta clasificación, como ya se ha explicado anteriormente en otros experimentos, se da pasando a una red lineal el vector pregunta, el vector respuesta y el valor absoluto de la diferencia de ambos concatenado. Por otro lado, en la Tabla 67 se puede ver la salida del modelo por la clasificación dada por el coseno entre pregunta y respuesta. En este caso, a la clase potencial se le ha asignado el ground truth 0.5. Dado esto y que los datasets estan balanceados, menos AmazonQA, se supone que los ejemplos cuyos cosenos vayan de -1 a 0.33 son pares malos, si van de 0.33 a 0.66 son pares potencialmente buenos y si están por encima de 0.66 son pares buenos o de la clase positiva.

Se puede ver que en todos los datasets da mejores resultados la clasificación que el coseno. Esto puede ser probablemente porque no es necesario poner umbrales, ya que poner umbrales a 3 clases es bastante complicado. Sin embargo, los resultados con el coseno son bastante buenos, en cualquier caso. Lo que se puede deducir de este experimento es que ambas tareas están relacionadas. A pesar de esta conclusión, hay mejores resultados si se entrena por separado cada una de las tareas por lo que aún siendo compatibles, las dos tareas no encajan a la perfección.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.5603403	0.70293522	0.56092306	0.56092515	0.91105189
SemEval 2017	0.5140736	0.705	0.47141115	0.56763817	0.86994928
Amazon QA	0.36869613	0.59431383	0.36641563	0.38022499	0.76118747

Tabla 32: Métricas en todos los datasets por Bi-Encoder multitask con clasificación

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.52208546	0.63714575	0.52635435	0.55272973	0.91131416
SemEval 2017	0.51460744	0.70352273	0.47031407	0.57084212	0.86872087
Amazon QA	0.29819163	0.66689684	0.33792243	0.35132529	0.77049342

Tabla 33: Métricas en todos los datasets por Bi-Encoder multitask con coseno

SemEval 2015

En las Tablas 68 y 69, se pueden ver las matrices de confusión del Bi-Encoder entrenado como multitask tanto con la capa de clasificación como con el coseno. Además, en las Tablas 70 y 71, se pueden ver las métricas por clase de dicho modelo con los dos métodos de predicción en el SemEval 2015. Como se ve, el modelo del coseno funciona peor en todas las clases con el loss del coseno que con la clasificación. Esto reafirma la suposición de que el problema del coseno es fijar los umbrales para separar las clases, umbrales que con la clasificación no es necesario fijar sino que se aprenden automáticamente. Sin embargo, se puede ver que las diferencias no varían mucho entre ambos métodos de predicción.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	591	77	144
	Potencial	52	31	84
	Positive	137	93	767

Tabla 34: Matriz de confusión SemEval 2015 Bi-Encoder multitask clasificación.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	446	210	156
	Potencial	36	43	88
	Positive	82	145	770

Tabla 35: Matriz de confusión SemEval 2015 Bi-Encoder multitask coseno.

	F1-score	Recall	Precision
Clase negativa	0.74246231	0.72783251	0.75769231
Clase potencial	0.16847826	0.18562874	0.15422886
Clase positiva	0.77008032	0.76930792	0.77085427

Tabla 36: Métricas por clase SemEval 2015 Bi-Encoder multitask clasificación.

	F1-score	Recall	Precision
Clase negativa	0.64825581	0.54926108	0.79078014
Clase potencial	0.15221239	0.25748503	0.1080402
Clase positiva	0.76578817	0.77231695	0.75936884

Tabla 37: Métricas por clase SemEval 2015 Bi-Encoder multitask coseno.

En la Tabla 72, se pueden ver los ejemplos mal clasificados por el modelo en el SemEval 2015. En el ejemplo 1, se puede ver que se pregunta por un número y se da ese número. Sin embargo, el Transformer no entiende dicho número y por tanto clasifica esta respuesta como negativa a pesar de ser positiva. Es un error típico de los Transformers no entender ciertos números o palabras.

Por otro lado, en el ejemplo 2 se puede ver se responde afirmativamente sobre si el nombre formulado en la pregunta es femenino. Sin embargo, al haber otro nombre en la respuesta el Transformer se equivoca y la clasifica como mala.

En el ejemplo 4, se pregunta si llovió en Qatar ayer. La respuesta va un poco con sorna diciendo que es bueno que la gente disfrutara de la lluvia que nadie ve en Doha más que una vez cada año. Se predice la respuesta como buena. La clase real es clase negativa. Es cierto que no es una respuesta completamente positiva pero sí que se responde que ha llovido, por tanto, es una respuesta válida para el usuario. Creo que, en este caso, la clase real debería ser potencial.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	you know the Telephone Number of Wakra Technical Inspection?	Dial 180	Buena	Mala
2	Tan Sze Peng is it a female name?	Yes its Female BIRTS name too....	Buena	Mala
3	hi,can anyone give me an idea about grade 109 pay scale in hamad medical cooperation? If HR give me an offer can i give it for a review? will it affect the current offer?	pay scale of 108 is 6500 so how can it be the same for 109?	Mala	Buena
4	Was there any rain in Qatar yesterday?	Good that most of you had enjoyed rains yesterday which is a rare seen in DOHA once in a year...:)	Mala	Buena

Tabla 38: Ejemplos mal clasificados por el Bi-Encoder multitask en SemEval 2015.

SemEval 2017

En las Tablas 73 y 74, se pueden observar las matrices de confusión de Bi-Encoder multitask con la clasificación y el coseno respectivamente con el SemEval 2017. En las Tablas 75 y 76, se pueden observar las métricas de dichas matrices de confusión. Se observa que al igual que en el caso del SemEval 2015, la capa de clasificación funciona mejor que el coseno. Esto como ya se ha explicado se debe a que con la capa de clasificación no hace falta poner umbrales. También hay que destacar que en ningún par de pregunta respuesta pertenece a la clase potencial, por lo que las métricas globales se reducen mucho a pesar de ser una buena predicción.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3252	491	500
	Potencial	0	0	0
	Positive	584	1021	2952

Tabla 39: Matriz de confusión SemEval 2017 Bi-Encoder multitask clasificación.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3225	575	443
	Potencial	0	0	0
	Positive	603	988	2966

Tabla 40: Matriz de confusión SemEval 2017 Bi-Encoder multitask coseno.

	F1-score	Recall	Precision
Clase negativa	0.80505013	0.76643884	0.84775808
Clase potencial	0	0	0
Clase positiva	0.73717068	0.6477946	0.85515643

Tabla 41: Métricas por clase SemEval 2017 Bi-Encoder multitask clasificación.

	F1-score	Recall	Precision
Clase negativa	0.79915748	0.76007542	0.84247649
Clase potencial	0	0	0
Clase positiva	0.74466483	0.6508668	0.87004987

Tabla 42: Métricas por clase SemEval 2017 Bi-Encoder multitask coseno.

En la Tabla 77, se pueden ver ejemplos mal clasificados por el Bi-Encoder multitask. En el ejemplo 1, se clasifica la respuesta como mala porque contiene cierta ironía. Sin embargo, la respuesta estaba etiquetada como buena porque aportaba información. Es otro ejemplo de la subjetividad a la hora de etiquetar estos datasets.

En el ejemplo 2, se puede ver que se responde con desdén a la pregunta. En este caso, sí que se da una respuesta valida a la pregunta, pero se clasifica como de clase negativa. Esto probablemente sea porque el Transformer no entiende la palabra “kinky”.

En el ejemplo 3, la clase se predice como buena a pesar de que esta etiquetada como mala. Esto es posible que ocurra debido a que realmente la respuesta es buena. Leyendola uno se puede fijar en que responde correctamente y además con una referencia al usuario que pregunta. De nuevo, un ejemplo de mal etiquetado del dataset.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Where can I find Birth Control Pills..? Many Thanks :)	Are you kidding? Just go to "SOUK HARAAG" very simple!!!!!!!!!.	Buena	Mala
2	i need a help guys....what is the good gift for my bf?..it is our 1st yr. anniversary... need some suggestion...thank you	Get him something kinky...	Buena	Mala
3	Can anyone help me in giving some information regarding Qatar Prometric exam for Nurses and what the requirement to sit for the exam?	https://www.prometric.com/en-us/Pages/home.aspx here schedule the test online and pay using a credit card.. it's a 70 items test and you have to get at least half to pass. You'll get the result right after the exam and it's valid for 3 years..:)	Mala	Buena
4	Is there any option of Multiple Entry Visit Visa for Qatar? Someone is saying, there is only Single Entry visa option. Also, where can i find the list of required documents as I couldn't see any such document in moi.gov.qa website.	Usually Visit Visas, are of two types 1. Simple Visit Visa, Extendable upto Six months 2. Business Visit Visa- extendable upto 3 months but both the types are valid for single entry and exit only, if you have passport from indian subcontinent or east Asia. Not confirmed about the EU Passports. For detailed info visit: moi.gov.qa	Mala	Buena

Tabla 43: Ejemplos mal clasificados por Bi-Encoder multitask en SemEval 2017.

AmazonQA

En las Tablas 78 y 79, se pueden ver las matrices de confusión del Bi-Encoder multitask con clasificación y con capa del coseno en el dataset AmazonQA. Se puede ver que el modelo de clasificación es mucho mejor que el modelo del coseno. De hecho, la diferencia de rendimiento es bastante más grande que en el caso de los otros datasets. Esto, como ya se ha comentado, es por los umbrales. En los casos anteriores, el coseno funcionaba un poco peor, pero al ser el dataset balanceado no bajaba tanto el rendimiento con respecto a la clasificación. En este caso, el dataset esta tan desbalanceado que empeoran las métricas en las clases minoritarias mucho más con el coseno. Esto se puede ver en las Tablas 80 y 81 en las que se encuentran las métricas por clase en el dataset AmazonQA por el modelo con clasificación y con coseno respectivamente.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	192655	44924	8303
	Potencial	67420	19135	4130
	Positive	15634	5877	2516

Tabla 44: Matriz de confusión AmazonQA Bi-Encoder multitask clasificación.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	235823	9193	866
	Potencial	85924	4547	214
	Positive	21090	2828	109

Tabla 45: Matriz de confusión AmazonQA Bi-Encoder multitask coseno.

	F1-score	Recall	Precision
Clase negativa	0.73872057	0.78352624	0.6987621
Clase potencial	0.23826274	0.21100513	0.2736073
Clase positiva	0.12910509	0.10471553	0.16830557

Tabla 46: Métricas por clase AmazonQA Bi-Encoder multitask clasificación.

	F1-score	Recall	Precision
Clase negativa	0.80113942	0.95909013	0.68785749
Clase potencial	0.08479017	0.0501406	0.27444471
Clase positiva	0.0086453	0.00453656	0.09167368

Tabla 47: Métricas por clase AmazonQA Bi-Encoder multitask coseno.

En la Tabla 82, se pueden ver ejemplos mal clasificados por el modelo en el dataset AmazonQA. En los ejemplos 1 y 2, se puede ver que la respuesta esta etiquetada como buena, pero se predice como mala. En este caso el etiquetado es correcto, pero se predice de esta manera probablemente debido al sobreentrenamiento en la clase negativa provocado por el desbalanceo de clases.

En el ejemplo 3, se puede ver que la respuesta esta etiquetada como mala porque es la misma pregunta, pero sin el signo de interrogación. Sin embargo, se predice de la clase buena. Esto probablemente sea porque el Transformer, al sobreentrenar tanto o, al menos, clasificar tantos ejemplos como de la clase negativa, solo se quede como clase positiva aquellos cuya similitud semántica sea muy parecida. En este caso, las palabras son iguales y en el mismo orden, por esto, el vector creado en ambos casos sea prácticamente el mismo.

En el ejemplo 4, se puede ver que se responde a la pregunta con otra pregunta. Por esto la clase de este par esta etiquetada como mala. Sin embargo, se predice la clase positiva. Posiblemente, esto ocurre porque tiene muchas palabras en común y, como se ha explicado en el ejemplo 3, el Transformer no ha aprendido por el desbalanceo a encontrar vectores cuya similitud dependa de su cercanía respecto a pregunta y respuesta.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	what scent should i get if i have never used this product before?	I only use the "bar" versions. Most scents are fairly mild with the citrus and the almond being my favorites. The teaberry is fairly strong (and sort of medicinal). Hope this helps.	Buena	Mala
2	Size chart for Chicago roller skates?	The size is about a half to a full size bigger, compared to regular shoes.	Buena	Mala
3	is this slow cooker insert lead free?	is the Cuisinart slow cooker lead free	Mala	Buena
4	Can this device attach to a TV to make a TV bluetooth enabled to pair with bluetooth speakers?	Will this pair with a TV soundbar that has a bluetooth woofer?	Mala	Buena

Tabla 48: Ejemplos mal clasificados por Bi-Encoder multitask en AmazonQA.

5.3.4 Entrenar Bi-Encoder y pesos de msmarco-distilbert-base-v4 con OVA.

En el caso de los modelos con el clasificador, se ha entrenado con el loss cross entropy mientras que en el caso del Bi-Encoder tradicional se ha utilizado el cosine embedding loss. Para clasificar la similitud entre dos embeddings se introduce en una red neuronal el vector pregunta, el vector respuesta y la diferencia absoluta entre ambos vectores concatenados, como se hace en Reimers et al. [100]. Se puede ver observar los resultados obtenidos con las diferentes métricas de ambas opciones en las Tablas 100 y 101 respectivamente. Se puede ver que la clasificación funciona mejor que el coseno y puede ser por el hecho de no usar umbrales.

A la hora de agregar los modelos del ensemble se ha procedido de la siguiente manera:

- En el caso de la clasificación, se ha clasificado cada ejemplo como la clase que tuviese mayor probabilidad entre los 3 modelos.
- En el caso del coseno, se ha clasificado cada ejemplo como la clase que tuviese mayor coseno entre pregunta y respuesta.

Las métricas son prácticamente las mismas que las obtenidas sin utilizar ningún tipo de ensemble, empeorando en algunos casos. Es, por tanto, que no es necesario utilizar un ensemble para mejorar los resultados, porque no se consigue.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.56146031	0.75910931	0.56679558	0.56949874	0.93666551
SemEval 2017	0.53361909	0.73954545	0.49438961	0.58155126	0.88820624
Amazon QA	0.28420117	0.68228811	0.34004458	0.46939988	0.72151429

Tabla 49: Métricas de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA clasificación.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.55680949	0.78947368	0.57889447	0.63730399	0.92161791
SemEval 2017	0.53673139	0.74818182	0.50011664	0.58132193	0.87009369
Amazon QA	0.31231007	0.6612201	0.34261466	0.38803641	0.76708923

Tabla 50: Métricas de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA coseno.

SemEval 2015

En las Tablas 102 y 103, se pueden ver las matrices de confusión del modelo con clasificación y coseno respectivamente en el SemEval 2015. Se aprecia que la clasificación predice mejor la clase potencial que el coseno. Esto puede ser debido a los umbrales fijados para determinar la clase potencial, que con la clasificación no hay que fijar. Se pueden ver estas observaciones en las Tablas 104 y 105, donde se ven las métricas del modelo con clasificación y coseno con SemEval 2015.

Actual value	Predicted value: Negative	Potencial	Positive
Negative	639	35	138
Potencial	58	10	99
Positive	123	23	851

Tabla 51: Matriz de confusión de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA clasificación en SemEval 2015.

Actual value	Predicted value: Negative	Potencial	Positive
Negative	709	1	102
Potencial	81	2	84
Positive	145	3	849

Tabla 52: Matriz de confusión de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA coseno en SemEval 2015.

	F1-score	Recall	Precision
Clase negativa	0.78455019	0.78694581	0.78216912
Clase potencial	0.08510638	0.05988024	0.14705882
Clase positiva	0.81472436	0.85356068	0.77926829

Tabla 53: Métricas por clase de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA clasificación en SemEval 2015.

	F1-score	Recall	Precision
Clase negativa	0.81167716	0.87315271	0.75828877
Clase potencial	0.02312139	0.01197605	0.33333333
Clase positiva	0.83562992	0.85155466	0.82028986

Tabla 54: Métricas por clase de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA coseno en SemEval 2015.

En la Tabla 106, se pueden observar ejemplos mal clasificados por el modelo tanto con clasificación como con el coseno en el SemEval 2015. En el ejemplo 1, se puede ver que la pregunta es sobre si alguien conoce a alguien que pueda reparar un coche y se responde que si que es para un land rover. Esta respuesta no es buena y por tanto se clasifica como mala. Sin embargo, el modelo entiende que aporta información y se clasifica como una clase buena.

En el el ejemplo 2, se pregunta sobre renovar el visado y se responde como que no es posible pero no parece claro si el que responde tiene clara o no la pregunta por su expresión. La clase de este par

pregunta y respuesta es negativa, pero se clasifica como buena porque a simple vista parece que la respuesta responde a la pregunta correctamente.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Does anyone know who can rebuild an automobile differential? I have all the new parts,,,bearings, shims etc.	yes, very scarce,,,its for a Land Rover Defender...	Mala	Buena
2	can i renew my wife visit visa (six month)under my sponosrship in post office	post office?it's not possible!	Mala	Buena
3	This is aimed at ex-pats already living in Qatar.\n\nI am moving out in early October and would like to ask, what one thing you would do differently (if you had your time again) regarding your move?\n\nAll help is much appreciated.\n\nThanks\n\nScott	i wouldn't have moved!	Buena	Mala
4	Hi All, I noticed that people selling their cars say that they are "expat" or "expat car". Well, what does that mean? Do "expat cars" bring more value than local folks' cars. I know expat take care of their cars and spend money on it. But, really .. who is an expat. To me .. when I read the advert, I expect an native english speaker or europeans. Is that so? or expat is just everyone who is not local?	Anyone who doesn't smell of mangos	Buena	Mala

Tabla 55: Ejemplos mal clasificados por Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA en SemEval 2015.

SemEval 2017

En la Tabla 107 y 108, se pueden ver las matrices de confusión del modelo con la última capa de clasificación y de coseno respectivamente con el SemEval 2017. En este caso, parece que las clases positiva y negativa están más balanceadas que en el SemEval 2015. Como ya se ha dicho en otras ocasiones, no existe ningún ejemplo que pertenezca realmente a la clase potencial. Se puede ver también por las métricas en las Tablas 109 y 110, que, en este caso, funciona mejor en todas las clases el coseno que la clasificación. Esto probablemente sea por el desbalanceo con respecto a la clase potencial que no existe en test.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3389	403	451
	Potencial	0	0	0
	Positive	502	936	3119

Tabla 56: Matriz de confusión de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA clasificación en SemEval 2017.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3420	392	431
	Potencial	0	0	0
	Positive	539	854	3164

Tabla 57: Matriz de confusión de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA coseno en SemEval 2017.

	F1-score	Recall	Precision
Clase negativa	0.83329235	0.79872732	0.87098432
Clase potencial	0	0	0
Clase positiva	0.76756491	0.68444152	0.87366947

Tabla 58: Métricas por clase de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA clasificación en SemEval 2017.

	F1-score	Recall	Precision
Clase negativa	0.83394294	0.80603347	0.86385451
Clase potencial	0	0	0
Clase positiva	0.77625123	0.69431644	0.88011127

Tabla 59: Métricas por clase de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA coseno en SemEval 2017.

En la Tabla 111, se pueden ver los ejemplos mal clasificados por el Bi-Encoder con OVA en el SemEval 2017. En el ejemplo 1, se pregunta sobre cuánto tiempo se tarda en conseguir el visado Schengen en Grecia. Sin embargo, se responde que no es necesario ningún tipo de pasaporte porque hicieron un viaje y nadie se lo pidió. El par pregunta y respuesta esta etiquetado como malo, pero se predice la clase como positiva. Esto es porque el Transformer entiende que es una respuesta que aporta información y que por tanto es una respuesta positiva.

En el ejemplo 3, se pregunta si alguien conoce a alguien que le pueda recomendar productos de Herbalife. La respuesta es que sí que coma sano y haga ejercicio. La etiqueta en este caso es que la clase es positiva. Sin embargo, no responde a la pregunta entonces el modelo lo predice como de la clase negativa.

En el ejemplo 4, el usuario cuenta que ha perdido mucho peso muy rápido y si alguien quiere contar su experiencia. Sin embargo, la repuesta es que probablemente haya perdido peso tan rápido porque era peso de agua. Esta respuesta esta clasificada como buena pero no responde realmente a la pregunta por lo que se predice como de la clase negativa.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Do you know how long it will take to get Schengen Visa from Embassy of Greece? 3 days? 1 week? any idea? Thanks.	We took euro train from London to Paris, arrived in Paris nobody asked our passport.	Mala	Buena
2	Does anyone know of any decent tennis clubs/ courts in Qatar? I have just moved here for a few months, and am keen to play on a regular basis. Any recommendations for a decent sports shop wld also be appreciated. cheers, Sambo	Al Dana Club(next to Khalifa stadium) and the sheraton i know for sure have indoor courts if you want to beat the heat and almost all the hotels have regular socials so...yeh :) As for sports shops Sportmart (opposite the Centre...next ot ramada) is good and also Sun and Sand sport on the 3rd floor of the City Centre(Carrefour side) infact that whole 3rd floor is full of sports shops on both sides :)	Mala	Buena
3	Does anybody know is there someone who can recomend HERBALIFE products?	Yes - eat healthily Do exercise	Buena	Mala
4	I don't believe that i lost 5 kg in 10 days.. by simply just counting calories and walking for 30 minutes everyday. any one would like to share his/her experience?	Zac,probably just water weight.	Buena	Mala

Tabla 60: Ejemplos mal clasificados por Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA en SemEval 2017.

AmazonQA

En las Tablas 112 y 113, se ven las matrices de confusión del Bi-Encoder con clasificación y coseno respectivamente en el dataset AmazonQA. Como en el resto de los casos en este dataset, sigue pareciendo que este dataset está mal configurado porque existe una clase mucho más superior que el resto, la clase negativa. Es por eso, que todos los ejemplos se clasifican mayoritariamente como de la clase negativa. En este caso, el coseno ha ido un poco mejor ya que aun habiendo mucho desbalanceo entre clases, la clase potencial tiene unas métricas ligeramente superiores que, en el caso de la clasificación, como se pueden ver en las Tablas 114 y 115.

Actual value \ Predicted value	Negative	Potencial	Positive
Negative	201827	38187	5868
Potencial	70524	17200	2961
Positive	16412	5824	1791

Tabla 61: Matriz de confusión de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA clasificación en AmazonQA.

Actual value \ Predicted value	Negative	Potencial	Positive
Negative	231731	12981	1170
Potencial	83942	6325	418
Positive	20660	2991	376

Tabla 62: Matriz de confusión de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA coseno en AmazonQA.

	F1-score	Recall	Precision
Clase negativa	0.81118213	0.99844234	0.68307063
Clase potencial	0.00026452	0.00013233	0.27272727
Clase positiva	0.04115684	0.02155908	0.45240175

Tabla 63: Métricas por clase de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA clasificación en AmazonQA.

	F1-score	Recall	Precision
Clase negativa	0.79603239	0.942448	0.68899275
Clase potencial	0.11196474	0.06974693	0.28367045
Clase positiva	0.02893309	0.01564906	0.19144603

Tabla 64: Métricas por clase de Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA coseno en AmazonQA.

En la Tabla 116, se pueden ver ejemplos mal clasificados por el modelo en el dataset AmazonQA. En el ejemplo 1, se pude ver como se pregunta si sirve para pintar un gato En la respuesta se responde que si si tu gato está dentro de algún lugar. Realmente se esta contestando a la pregunta, pero la etiqueta es que la respuesta es de clase negativa. Esto es debido a que existen problemas de etiquetado con este dataset. Sin embargo, el modelo entiende que si es una respuesta relevante y la clasifica como de la clase positiva.

En el ejemplo 2, se pregunta si esa cubierta vale para su portátil. Se responde que depende del portátil pero que se puede llamar a la compañía para consultarlo. La clase real es que este par de pregunta

respuesta es de la clase mala sin embargo se responde que es de la clase buena porque parece que aporta nueva información.

En el ejemplo 3, se pregunta si el aparato funciona sin detección de movimiento. El usuario responde que ni siquiera sabe cómo funciona. Sin embargo, se etiqueta la respuesta como buena, a pesar de que la respuesta no aporta ninguna información. Vuelve a ser problema del mal etiquetado del dataset. Sin embargo, el modelo predice que la clase es negativa porque no aporta nada.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	does this really work for cat spraying in one spot?	does this product work for spraying, if you cat is indoor outdoor cat?	Mala	Buena
2	I have a Lenovo Ideapad A1 (7 inch) will this cover work for it?	how are the cut outs of your Idea pad, but what you can do is go on line and find the number to this company and ask them.	Mala	Buena
3	can this run continually without the motion detection? I want to set it up and leave it on. For how long will it record all alone being on	don't even know how to use it	Buena	Mala
4	Is this all metal construction? And is it meant for long term use (i.e., not a one-month or one-time use gun)?	Most of the gun you are asking about is mostly plastic (main body, handle, etc) The only metal parts are the nozzle tube, the aluminum block it connects to inside the main body, the can connector, and the control rod inside the dispenser tube. Good maintenance should let this gun last quite a while.....I had one work flawlessly for 3 years.....sparing use of cleaner and only clean tip with fingernail or credit card - type material is suggested	Buena	Mala

Tabla 65: Ejemplos mal clasificados por Bi-Encoder con pesos de msmarco-distilbert-base-v4 con OVA en AmazonQA.

5.3.5 Bi-Encoder preentrenado con los pesos de msmarco-distilbert-base-v4 con 2 clases

La motivación de esta propuesta es ver si un encoder pre-entrenado puede mejorar dar un buen rendimiento sobre dos clases: Buena y Mala. Hemos elegido los pesos de msmarco-distilbert-base-v4 para el encoder debido a la similitud de la tarea propuesta. De hecho, los resultados han demostrado que se comporta mejor que una propuesta entrenada en AmazonQA. Esto se debe a que el encoder es capaz de aprender la estructura general de los datos y, por lo tanto, es capaz de rendir en el pipeline propuesto, evitando el sobreentrenamiento.

Se pueden ver en la Tabla 139 las métricas por dataset de dicho modelo preentrenado. Se puede comprobar que los resultados mejoran respecto a sus homologos con 3 clases y que en AmazonQA incluso mejoran el resto de los experimentos con 2 clases supervisados.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.76879433	0.67004049	0.93127148	0.65458937	0.82955561
SemEval 2017	0.59752149	0.59034091	0.58722844	0.60818182	0.69699115
Amazon QA	0.41353315	0.5202	0.53174908	0.33831955	0.76683485

Tabla 66: Métricas por dataset Bi-Encoder preentrenado con 2 clases.

SemEval 2015

En la Tabla 140, se pueden ver ejemplos mal clasificados por el modelo en el SemEval 2015. En el ejemplo 1, se plantea la duda de como cambiar la fecha de cumpleaños del ID Qatari. En la respuesta se indica como cambiarlo. Por tanto, la clase real es positiva. Sin embargo, como el Bi-Encoder no ha sido entrenado y la pregunta y la respuesta no son similares semanticanente, se predice que la clase es negativa.

En el ejemplo 2, pasa todo lo contrario al ejemplo 1. Se pregunta una cuestión y la respuesta es la misma cuestión reformulada. Como no aporta nueva información se etiqueta como de la clase mala, pero al realizar la predicción, como son muy similares se predice de la clase positiva.

En el ejemplo 3, pasa igual que en el ejemplo 1. La respuesta es satisfactoria pero como aporta nueva información y semánticamente no son iguales se predice la clase como negativa.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Dear All,\n\nI have changed birthday date in passport due to wrong birthday was written my passport. What is the process to change in Qatari ID. Can you help me to know how i can change it in qatari ID. Thanks.	go to immigration office at markhiya .....meet captain....\n	Buena	Mala
2	hi,can anyone give me an idea about grade 109 pay scale in hamad medical cooperation? If HR give me an offer can i give it for a review? will it affect the current offer?	pay scale of 108 is 6500 so how can it be the same for 109?	Mala	Buena
3	I am looking for an Avery labels, I wonder if anyone know where could I buy them.\n\nMany thanks...CRM	Agree with asiha Office 1 - is the place I bought laser lables for a seminar from them. They on left side on salwa road if you leaving doha - beyond ramada i think.\n \n	Buena	Mala
4	Is it better to transfer from family to work visa? I got a job offer from Barwa and it includes work visa. Presently, I'm under the sponsorship of my husband. Please reply to enlighten me.. Thanks.\n	> cheekylady, did u manage to transfer your visa from family to work?\nHow long did it takes, what are the requirements?\nI'm in the same situation now.\nThanks ;)\n	Mala	Buena

Tabla 67: Ejemplos mal clasificados por Bi-Encoder preentrenado con 2 clases en SemEval 2015.

SemEval 2017

En la Tabla 141, se pueden ver los ejemplos mal clasificados por el modelo en el SemEval 2017. En el ejemplo 1, se ve como se pregunta sobre un recinto. Se responde que se encuentra detrás de un hotel. Evidentemente la respuesta es buena y esta etiquetada como tal pero como el Bi-Encoder no ha sido entrenado y no son iguales semánticamente, se clasifica la respuesta como mala.

En el ejemplo 2, se pregunta cuanto tarda en realizarse un visado. La respuesta es otra pregunta sobre visados en la misma zona. Como se parecen mucho ambos textos se predice la clase como buena, aunque evidentemente la clase real es mala.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Can someone tell some information about this compound?? thankyou a	I think it is behind Ramada hotel	Buena	Mala
2	Do you know how long it will take to get Schengen Visa from Embassy of Greece? 3 days? 1 week? any idea? Thanks.	Just curious... is it possible to get a UK visa while visiting one of the schengen countries??	Mala	Buena
3	Can anyone help me to choose and cook some of the fish I see in the supermarkets, but have never heard of? I also find many of the vegetables new - can anyone list the most frequent fish and recommend a recipe? Many thanks	throw it on the bbq.	Buena	Mala
4	hello. i am new here in doha, qatar. i need a "KABAYAN" help on where to send money from Doha to Philippines in the most safest and cheapest way. Western Union is good but its money exchange rate is too low. I also have a bank account in my born-town. which is which? mga kababayan, suggestions nman diyan.	sabi ng mga kasamahan ko sa al fardan. magpapadala rin ako gusto mo sama tau? Bago lang din ako rito e.. :)	Mala	Buena

Tabla 68: Ejemplos mal clasificados por Bi-Encoder preentrenado con 2 clases en SemEval 2017.

AmazonQA

En la Tabla 142, se pueden ver ejemplos mal clasificados por el Bi-Encoder preentrenado con 2 clases en el dataset AmazonQA. Se puede ver en el ejemplo 2, como aunque la clase real es mala porque la respuesta no responde correctamente a la pregunta, se clasifica como buena porque están semánticamente muy relacionadas.

En el ejemplo 3, se responde a sobre si la cinta de correr se puede inclinar. Sin embargo, esto añade nueva información por lo que la cantidad de información común se reduce y se predice como de la clase negativa a pesar de estar etiquetado como de la clase positiva.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Remote damaged, is it possible to find a new one? Does logitech sell just the remote for these speakers? Where can I find a new one? If anybody know, please. Thanks.	IGregorio-Simply Google the remote's model number, among the links will be a collection of vendors that sell 'original' replacement remotes. Be aware that the prices will be all over the place. When I did this, my remote came back with prices from 25 to 95.best,a.	Buena	Mala
2	HI I would like to find out if it works with polar F4?	Sorry not much help. Mine is used with F55 so I am unsure. Good luck.	Mala	Buena
3	Does the treadmill incline?	Yes, it does, it has two settings! Very good value for the money!	Buena	Mala
4	I have just purchased The Little Giant Titan X 17 foot ladder. will this ladder leveler fit my ladder ?	I'm pretty sure it will work with your ladder. I have a Little Giant ladder (I'm not sure of the size) and I bought one of these leg levelers myself, and it works great.	Mala	Buena

Tabla 69: Ejemplos mal clasificados por Bi-Encoder preentrenado con 2 clases en AmazonQA.

5.3.6 Bi-Encoder entrenado con los pesos de msmarco-distilbert-base-v4 con 2 clases

Se puede ver que los resultados de esta propuesta son bastantes buenos en la Tabla 143, menos en el caso de AmazonQA en los que el F1-score se reduce mucho con respecto al modelo preentrenado. Esto probablemente es debido al desbalanceo de clases durante el entrenamiento.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.80948487	0.76417004	0.85051546	0.77223089	0.89899757
SemEval 2017	0.83711466	0.81806818	0.90278692	0.78034901	0.86045987
Amazon QA	0.07591576	0.68531922	0.04063219	0.57671368	0.78685507

Tabla 70: Métricas por dataset Bi-Encoder con 2 clases.

SemEval 2015

En la Tabla 144, se pueden ver ejemplos mal clasificados por parte del modelo en el SemEval 2015. En el ejemplo 1, se pueden ver una pregunta y una respuesta que no tienen ningún lazo. Por un lado, la pregunta dice que el coche es un honda civic y la respuesta habla sobre si se conduce el coche en automático o en manual. Probablemente, con los metadatos de la respuesta se podría determinar que la respuesta corresponde con esta pregunta ya que la clase real es buena. Sin embargo, como el modelo solo cuenta con los textos se predice la clase como negativa.

En el ejemplo 2, se puede ver que la repuesta es una reformulación de la pregunta para buscar aclaraciones. Esto hace que la respuesta no sea buena, ya que no aporta nueva información. Por su parte, el modelo como venía entrenado con unos pesos que buscan la similitud semántica entre dos textos, clasifica la respuesta como buena, a pesar del fine-tuning.

En el ejemplo 4, se pregunta como ahorrar dinero. La respuesta carece de cualquier sentido y no esta remotamente relacionada con la pregunta. Por tanto, la etiqueta real es de la clase mala. A pesar de ello, el modelo lo predice como de la clase buena probablemente porque no entiende ciertas palabras de la respuesta.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Car is Honda Civic	manual your driving but automatic your like just sitting :) \n	Buena	Mala
2	hi,can anyone give me an idea about grade 109 pay scale in hamad medical cooperation? If HR give me an offer can i give it for a review? will it affect the current offer?	pay scale of 108 is 6500 so how can it be the same for 109?	Mala	Buena
3	Hi.\nDoes anybody know if it is possible to buy car parts from the scrapyard located at your right hand when you drive from Mesaieed to Al Wakra ? I have been down there a couple of times, but everytime I meet a man who doesn`t speak english.\n\n	Would you tell me the exact location of the scrap yard you are telling in Masseid?	Mala	Buena
4	So, I have seen that I have been spending money more than usual. So, I need to cut back on expenses!!\n\nWhat are your tips on saving money here?	GYM?! Darude.. hummmmm!!!!!!!!!!!!!!!!!!!	Mala	Buena

Tabla 71: Ejemplos mal clasificados por Bi-Encoder con 2 clases en SemEval 2015.

SemEval 2017

En la Tabla 145, se pueden ver ejemplos mal clasificados con el Bi-Encoder con 2 clases en el SemEval 2017. En el ejemplo 1, se pregunta si una serie de parques están abiertos. Se responde que no, que se olvide de ir. Sin embargo, la manera de contestar es con acrónimos en inglés, es decir, en vez de decir *forget* se dice *4get*. Esto la red no lo entiende y etiqueta la respuesta como de la clase negativa.

En el ejemplo 2, se pregunta sobre una agresión y cuál es la multa en general por estos. La respuesta por otro lado parece un mensaje filosófico sobre que no se puede predecir el futuro. Probablemente por problemas de etiquetado, la clase real es la positiva a pesar de que la respuesta no da ninguna información extra. El modelo no se deja engañar tan fácil y predice la clase como la negativa que es la que realmente parece que es.

En el ejemplo 3, se pregunta sobre detalles de la inmigración filipina, si hace falta una carta del patrocinador catarí. La respuesta parece que responde a la pregunta, pero realmente es una respuesta escrita por el mismo usuario que formula la pregunta inicial. Al estar escrito dicho comentario por el autor de la pregunta, la clase real es buena. Por otra parte, el modelo al no saber que la respuesta pertenece al mismo usuario que la pregunta, clasifica la clase como positiva.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Aladdin's Kingdom is it open ? if not is there any similar places ? Aqua Park is it open ? if not is there any similar places ? Doha Zoo is it open ? if not is there any similar places ? Thank you for your feedback	No 4get	Buena	Mala
2	I lost my temper with my colleague. He is so rude and arrogant. Besides, he is hated by all. I slapped his face with a paper punch and got a cut (not serious). The case is pending in the court. Do you know of the common penalty given by the court on this particular case? Pls. reply.	"The best way to predict the future is to create it".	Buena	Mala
3	Hi, i took the business visa of my friend in one of the company here in Qatar. but just recently, she said that the need of authenticated letter of sponsor is need in the Philippine Immigration? does anybody know how true is this? Please help...	Thanks Jolena but according to her (my friend) that there is a new policy, a letter from the sponsor must be needed. Immigration Officer in the Phils will ask for that. Do u have any idea regarding that? This is very new to my knowledge, that's why I opt to ask here in QL.	Mala	Buena
4	Hello everyone, I will be joining Qatar University by the end of August. Anyone have some idea, how is academic environment there? Also, I will be staying at University housing (yet don't know where), How is university housing facility-wise and for a family of three. Anyone of you living there?, plz write about your experience. I will appreciate you feed back. Thanks	Thanks for information and advice, it really helps a lot. I have been teaching in US for a while and was thinking the academic environment is similar to US universities or somehow different. They system they are using at QU just looks like a copy of US system.	Mala	Buena

Tabla 72: Ejemplos mal clasificados por Bi-Encoder con 2 clases en SemEval 2017.

AmazonQA

En la Tabla 146, se pueden ver ejemplos mal clasificados por parte del Bi-Encoder con 2 clases en el dataset AmazonQA. En el ejemplo 1, se pregunta si se puede acceder a unas notas dentro del tablero y se responde que no pero que se puede ver con el smarphone. Evidentemente, la clase real es buena, pero se predice como de la clase negativa, probablemente por el sesgo creado por el desbalanceo del dataset.

En el ejemplo 2, se pregunta que hacen unas barras y se responde solo con interrogantes. La respuesta no aporta nada y la clase real es mala. Sin embargo, el modelo lo clasifica como de la clase buena porque no ha visto un ejemplo así en el dataset de train y se equivoca, probablemente.

En el ejemplo 3, se pregunta que si las baterías mueren que si se puede abrir manualmente el producto. La respuesta explica claramente como se puede abrir y su experiencia. Por tanto, la clase etiquetada es buena. A pesar de ello, el modelo clasifica esta respuesta como mala probablemente por el sesgo del modelo hacia la clase negativa.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Is there any way to navigate to the saved notes within the board?	No, but if you connect a smart phone or tablet you can look through past notes.	Buena	Mala
2	what do sway bars do?	???????	Mala	Buena
3	if the batteries die, can you still open it manually?	Some times. The first one we got was DOA, the second one failed a little later. When it failed, (not dead batteries) sometimes the lid was locked open, sometimes (most frequent) locked closed, sometimes free-wheeling.	Buena	Mala
4	what do sway bars do?	Reduces body roll. When I put these on, it significantly reduced the number of times my car rolled over. Well worth the $$$.	Mala	Buena

Tabla 73: Ejemplos mal clasificados por Bi-Encoder con 2 clases en AmazonQA.

5.4 Propuestas basadas en Cross-Encoders

En esta Sección se presentan las propuestas basadas en Cross-Encoders. Tal y como se ha indicado anteriormente, se encuentran tanto las propuestas que hacen una predicción en 3 clases como en 2. El número de propuestas con Cross-Encoders es menor que con Bi-Encoders debido a la imposibilidad de hacer realizar propuestas con multitask o preentrenadas como en el caso de estos últimos.

5.4.1 Cross-Encoder entrenado

La primera propuesta con cross-encoders es la más básica: Entrenar un modelo preentrenado con cada uno de los datasets. De esta manera, se puede comprobar la calidad de sus resultados con respecto al resto de bi-encoder.

Los pesos han sido entrenados anteriormente para calcular la similitud semántica, pero al entrenarlo con un fine-tuning conseguimos cambiar el problema para encontrar la similitud entre pregunta y respuesta. Una desventaja de este modelo es que necesita hacer un fine-tune sobre el modelo y entrenar la última capa siempre que se quiera aplicar a un nuevo problema. Esto con los Bi-Encoders no es necesario ya que la clasificación depende únicamente del coseno. También es algo más lento que en el caso de los Bi-Encoder y requiere de más memoria ya que el modelo tiene que codificar la pregunta junto con la respuesta simultáneamente.

Con respecto a la solución anterior, lo que se consigue es modelar también las relaciones entre pregunta y respuesta. Esto permite crear un vector en el que estén codificadas las dependencias entre ambas. La ventaja de esta idea es que no se tratan las dos por separado, aprendiendo más características.

En las Tablas 22 y 23, se ven las métricas de los dos datasets con los diferentes pesos. Se puede apreciar que en los dos SemEval se tiene mejores métricas con el Cross-Encoder paraphrase-MiniLM-L6-v2, aunque en AmazonQA mejoran con stsb-distilbert-base. Además, la ventaja de paraphrase-MiniLM-L6-v2 es que es mucho más rápida de entrenar e inferir al contener menos parámetros.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.576450306	0.726214575	0.576545	0.579071	0.92674989
SemEval 2017	0.5240266	0.716023	0.47631	0.5847979	0.892255
Amazon QA	0.2944267	0.682404588	0.345408505	0.70275868	0.790203933

Tabla 74: Métricas del Cross-Encoder con stsb-distilbert-base fine-tuneado.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.584404782	0.747975709	0.582918	0.587087	0.93608346
SemEval 2017	0.527926465	0.721023	0.481996	0.5855307	0.901374
Amazon QA	0.2729418	0.681284214	0.33388173	0.30757311	0.785050394

Tabla 75: Métricas del Cross-Encoder con paraphrase-MiniLM-L6-v2 fine-tuneado.

SemEval 2015

En las Tablas 24 y 25, se pueden ver las matrices de confusión del SemEval 2015 con las dos arquitecturas. A su vez, en las Tablas 26 y 27 se ven las métricas del SemEval 2015 con ambas arquitecturas. Se puede observar que tanto la clase negativa como la positiva se predice de manera muy satisfactoria. La clase potencial, que a su vez es la más escasa, es la que peor se predice y en la que se obtienen peores métricas. Si se compara este experimento con los otros dos anteriores se ve que los resultados mejoran sustancialmente. Por lo tanto, se puede sacar en claro que un experimento con entrenamiento es mejor que sin él.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	616	93	103
	Potencial	52	30	85
	Positive	105	103	789

Tabla 76: Matriz de confusión en SemEval 2015 con stsb-distilbert-base fine-tuneado.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	620	76	116
	Potencial	46	25	96
	Positive	86	78	833

Tabla 77: Matriz de confusión en SemEval 2015 con paraphrase-MiniLM-L6-v2 fine-tuneado.

	F1-score	Recall	Precision
Clase negativa	0.77728706	0.75862069	0.79689521
Clase potencial	0.15267175	0.17964072	0.13274336
Clase positiva	0.7993921	0.79137412	0.80757421

Tabla 78:Métricas por clase en SemEval 2015 con stsb-distilbert-base fine-tuneado.

	F1-score	Recall	Precision
Clase negativa	0.79283888	0.7635468	0.82446809
Clase potencial	0.14450867	0.1497006	0.1396648
Clase positiva	0.8158668	0.83550652	0.79712919

Tabla 79: Métricas por clase en SemEval 2015 con paraphrase-MiniLM-L6-v2 fine-tuneado.

En la Tabla 28, se pueden ver algunos ejemplos mal clasificados por ambas arquitecturas en el SemEval 2015. En el primer ejemplo, se habla del salario en el grado 108. Esto hace que el modelo suponga que la clase del comentario es bueno, aunque realmente se pregunta por el grado 109 y por tanto es mala respuesta. A pesar de esto, se puede ver que el modelo ahora entiende que tiene que encontrar la similitud entre pregunta y respuesta, no entre dos textos cualesquiera. En el ejemplo 3, se pregunta sobre una chica corriendo con un gato agarrado a ella y sobre si es esto normal. La respuesta es que no es normal y por tanto la clase es buena. Sin embargo, se predice que es mala probablemente por la sencillez de ésta. En el ejemplo 4 no se responde a la pregunta directamente, aunque sí se menciona que está abierto. Por tanto, la clase es potencial, aunque se predice como clase mala porque no está contestada de manera directa. Viendo los errores se puede observar que éstos tienen más sentido que en los casos anteriores.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	hi,can anyone give me an idea about grade 109 pay scale in hamad medical cooperation? If HR give me an offer can i give it for a review? will it affect the current offer?	pay scale of 108 is 6500 so how can it be the same for 109?	Mala	Buena
2	why supermarkets not keeping coins?always they are giving chewing gum or chocolate.is it dificult to keep coins in qatar? or this is a business trick?\nif we argue they will ask coin from us.\nanyway.... this is not a good habit.	Princess, Old QLers are present, they just need a small electrical shock from 440v connection ;)\nevery one is in reading mode now a days..\nand recent comment tab absence is its one of the cause also :(\n\nnow you don't go to hibernation..	Mala	Buena
3	I saw a little girl running by the streets , and she had a cat attached to her .....is that normal in this country?	its not normal!	Buena	Mala
4	Does anyone know if or when the Doha Zoo will reopen?\n\nThanks!	yesterday i went there.... ya too much crowded... but still worth seeing	Potencial	Mala

Tabla 80: Ejemplos mal clasificados de SemEval 2015 con las dos arquitecturas del Cross-Encoder

SemEval 2017

En las Tablas 29 y 30, se pueden ver las matrices de confusión del SemEval 2017 con las dos arquitecturas. A su vez, en las Tablas 31 y 32 se ven las métricas del SemEval 2017 con ambas arquitecturas. Se puede ver que al igual que en el SemEval 2015, la clase positiva y la clase negativa se predicen muy bien. La clase potencial no tiene ningun ejemplo real, aunque tiene falsos positivos tanto negativos como positivos. Se puede ver que se confunde más la clase potencial con la clase positiva que con la negativa. Esto puede dar idea de que realmente se parecen estas dos clases y la diferencia es sutil. Las métricas por clase son buenas, excepto en la clase potencial porque no existen ejemplos en test de dicha clase. Se puede ver que el modelo actúa mejor que lo que aparenta en las métricas globales. Esto es porque la clase potencial hace que se disimule el buen funcionamiento de este sistema.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3480	333	430
	Potencial	0	0	0
	Positive	696	829	3032

Tabla 81: Matriz de confusión en SemEval 2017 con stsb-distilbert-base fine-tuneado.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3302	528	413
	Potencial	0	0	0
	Positive	467	1047	3043

Tabla 82: Matriz de confusión en SemEval 2017 con paraphrase-MiniLM-L6-v2 fine-tuneado.

	F1-score	Recall	Precision
Clase negativa	0.82645661	0.77893	0.88015979
Clase potencial	0	0	0
Clase positiva	0.74562319	0.65	0.87423402

Tabla 83: Métricas por clase en SemEval 2017 con stsb-distilbert-base fine-tuneado.

	F1-score	Recall	Precision
Clase negativa	0.82426361	0.77822296	0.87609445
Clase potencial	0	0	0
Clase positiva	0.75951579	0.66776388	0.88049769

Tabla 84: Métricas por clase en SemEval 2017 con paraphrase-MiniLM-L6-v2 fine-tuneado.

En la Tabla 33, se pueden ver algunos ejemplos mal clasificados por ambas arquitecturas en el SemEval 2015. En el ejemplo 2, se clasifica dicho comentario como uno bueno, aunque es de la clase mala. Esto es porque el comentario sí que está relacionado con la pregunta y probablemente sea una respuesta valida si el autor de la respuesta es del mismo país. La pregunta es cuánto cuesta sacar el visado y en la respuesta se menciona que realmente no es necesario. Así que este error es entendible. En el ejemplo 1, se clasifica la respuesta como bueno, aunque realmente es de la clase mala porque no ofrece ningún grupo. A pesar de esto, sí que habla sobre jugar a Badminton por lo que la clasificación de la red es entendible. Como se puede observar, estos fallos son fallos que podría cometer un humano al etiquetar las muestras y que indican que el modelo está realmente aprendiendo lo que se necesita para solucionar esta tarea.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Hello! Any badminton group or club here that plays regularly? I would like to join you. Kindly PM me the details. Thank you!	Hi, i would like to play batminton plz call me 66430353.	Mala	Buena
2	Do you know how long it will take to get Schengen Visa from Embassy of Greece? 3 days? 1 week? any idea? Thanks.habit.	We took euro train from London to Paris, arrived in Paris nobody asked our passport.	Mala	Buena
3	hi everyone! I work here in doha under a gov't sponsorship...i heard that aquiring a US visit visa with this sponsorship will make it all easy. Any truth about this? Thanks ,,,	my collegues got the visa in a weeks time. we applied it online got interview date after 4 days . went for the interview and got the passport stamped with vissa the other day for 5 years. they both are indians.	Buena	Mala
4	What was wrong with the old version, new one keeps doing strange things & I'm getting annoyed with it!! What say you guys that are FB users?! Torque	the new look is like some web ads page. i dont think the layout team did a thorough user survey before launching the new FB.	Buena	Mala

Tabla 85: Ejemplos mal clasificados de SemEval 2017 con las dos arquitecturas del Cross-Encoder.

AmazonQA

En las Tablas 34 y 35, se pueden ver las matrices de confusión del AmazonQA con las dos arquitecturas. A su vez, en las Tablas 36 y 37 se ven las métricas del AmazonQA con ambas arquitecturas. Se puede ver en este caso que sobreaprende la clase negativa. Debido a que tanto en test como en train la clase negativa es la mayoritaria parece que las demás clases no las predice bien. De hecho, las métricas salen solo bien en el caso de la clase negativa. Se verá si en otros experimentos se lidia mejor con el desbalanceo de clases.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	245126	0	756
	Potencial	90159	1	525
	Positive	23083	0	944

Tabla 86: Matriz de confusión en AmazonQA con stsb-distilbert-base fine-tuneado.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	245305	577	0
	Potencial	90323	362	0
	Positive	23455	572	0

Tabla 87: Matriz de confusión en AmazonQA con paraphrase-MiniLM-L6-v2 fine-tuneado.

	F1-score	Recall	Precision
Clase negativa	0.81133967	0.99692535	0.68400638
Clase potencial	2.2054E-05	1.1027E-05	1
Clase positiva	0.07191833	0.03928913	0.42426966

Tabla 88: Métricas por clase en AmazonQA con stsb-distilbert-base fine-tuneado.

	F1-score	Recall	Precision
Clase negativa	0.81097253	0.99765335	0.68314289
Clase potencial	0.00785284	0.00399184	0.23957644
Clase positiva	0	0	0

Tabla 89: Métricas por clase en AmazonQA con paraphrase-MiniLM-L6-v2 fine-tuneado.

En la Tabla 38, se pueden observar ejemplos mal clasificados por el modelo en el dataset AmazonQA. En el ejemplo 1, se predice la clase como buena porque se pregunta sobre qué coche es mejor y se responde un coche que le parece mejor. Sin embargo, en la clasificación real la respuesta es mala porque a los usuarios no les ha gustado que se mencione el racismo. En el ejemplo 2 y 3, se puede ver el sobre entrenamiento mencionado anteriormente porque las respuestas son claramente buenas, pero se clasifican como malas. Por tanto, tenemos un ejemplo de dataset en el que se sobreentrena.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	hello everyone. I'm sick and tired of using the roads and wait in traffic jams so I deided to buy a used 4x4 to discover some new places that only a 4x4 can take u there. The two best are landcruiser and 129hevro patrol but witch is better in terms of quality and service? Please if u r an owner Of one of these two cars I need your advice.	Go for 129hevrolet Tahoe ---------- No human can stop racism.	Mala	Buena
2	Is the pocket clip reversable for use on a cap?	Yes, it is reversible. You just have to clip it onto the thinner part of the barrel rather than the very end. I put it on my cap this past weekend and it stayed on nicely although it did fall once when I was jolted hard. So great for walking or projects, but I might not take it running.	Buena	Mala

3	Please recommend perfect “closed” version of the 595s? I’d buy these right now except I need them for the office and don’t want to disturb my co-workers. What are the “perfect” ~$200 headphones for the office that have the rave reviews like the 595s?	Now, right here on Amazon Estes 4615 Proto X Rotor Blade Set (4)	Buena	Mala
4	Will it work with the Yongnuo Speedlite YN560?	Yes, should fit with no problem. It handles my SB-910 flash with room to spare	Buena	Potencial

Tabla 90: Ejemplos mal clasificados de AmazonQA con las dos arquitecturas del Cross-Encoder.

5.4.2 Entrenar Cross-Encoder con OVA.

Esta propuesta se base en realizar un ensemble de un modelo. Se va a usar un ensemble del tipo One Vs All. Cada uno de los modelos va a realizar una clasificación en la que se tiene que decidir si la muestra pertenece a una clase en particular o a cualquiera de las demás. Por este motivo, se crean 3 modelos, uno para cada clase. Para cada clasificación, se ha utilizado un modelo con los pesos mencionados anteriormente. El número de épocas que se ha utilizado han sido 10. Los pesos guardados han sido los que han obtenido mejor accuracy en validación. Para mezclar los resultados, la clase de cada ejemplo es la clase cuya probabilidad es máxima entre los modelos del OVA. Se pueden ver los resultados obtenidos con distintas métricas con stsb-distilbert-base y paraphrase-MiniLM-L6-v2 en las Tablas 83 y 84 respectivamente.

La ventaja de este modelo con respecto al resto es que, al hacer un modelo por clase, aprende mejor los ejemplos de la clase minoritaria. Esto hace que este tipo de ensembles mejoren el desbalanceo entre clases.

Se pude ver en las métricas de las Tablas 83 y 84 que los resultados con el OVA no mejoran mucho, sino que oscilan aproximadamente por los mismos resultados que el cross encoder sin realizar ensembles. A pesar de ello, los resultados son buenos por lo que este experimento es satisfactorio. Se puede ver que el paraphrase-MiniLM-L6-v2 mejora los resultados del stsb-distilbert-base en casi todos los casos.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.56089442	0.76973684	0.61710116	0.56414204	0.93907363
SemEval 2017	0.52844339	0.72238636	0.48321442	0.58623653	0.88808907
Amazon QA	0.34644306	0.64355757	0.35609882	0.45927511	0.77684889

Tabla 91: Métricas por dataset en el Cross-Encoder con OVA con pesos de stsb-distilbert-base.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.56236571	0.77327935	0.5705626	0.57889724	0.9454376
SemEval 2017	0.52734071	0.72045455	0.48193843	0.58575408	0.88493079
Amazon QA	0.28420117	0.68228811	0.34004458	0.46939988	0.72151429

Tabla 92: Métricas por dataset en el Cross-Encoder con OVA con pesos de paraphrase-MiniLM-L6-v2.

SemEval 2015

En las Tablas 85 y 86, se pueden ver las matrices de confusión del Cross-Encoder OVA con stsb-distilbert-base y paraphrase-MiniLM-L6-v2 respectivamente con el SemEval 2015. A su vez, en las Tablas 87 y 88, se pueden ver las métricas por clase con dichos modelos en el SemEval 2015. Se puede ver que se detectan muy bien las clases negativa y positiva, aunque la clase potencial se detecta muy mal. Esto se debe a que esta es la clase menos clara de todas, no está bien definida la separación entre la clase potencial y el resto de las clases. La clase que mejor se detecta es la positiva, como en el resto de los experimentos en este dataset.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	670	19	132
	Potencial	61	7	99
	Positive	134	10	844

Tabla 93: Matriz de confusión del SemEval 2015 con Cross-Encoder OVA stsb-distilbert-base.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	631	11	154
	Potencial	56	7	104
	Positive	96	27	890

Tabla 94: Matriz de confusión del SemEval 2015 con Cross-Encoder OVA paraphrase-MiniLM-L6-v2.

	F1-score	Recall	Precision
Clase negativa	0.79904591	0.82512315	0.77456647
Clase potencial	0.06896552	0.04191617	0.19444444
Clase positiva	0.81467182	0.84653962	0.78511628

Tabla 95: Métricas por clase SemEval 2015 con Cross-Encoder OVA stsb-distilbert-base.

	F1-score	Recall	Precision
Clase negativa	0.79122257	0.7770936	0.80587484
Clase potencial	0.06603774	0.04191617	0.15555556
Clase positiva	0.82983683	0.89267803	0.77526132

Tabla 96: Métricas por clase SemEval 2015 con Cross-Encoder OVA paraphrase-MiniLM-L6-v2.

En la Tabla 89, se pueden observar ejemplos mal clasificados por el Cross-Encoder con OVA en el SemEval 2015. En el ejemplo 3, se puede apreciar cómo la respuesta es satisfactoria pero la pregunta es tiene tantas palabras mal escritas que el Transformer no entiende el significado de la pregunta.

En el ejemplo 4, se puede ver que pregunta algo cuyo significado no está claro porque ni siquiera está en forma de pregunta. La respuesta también es ambigua. A pesar de ello, se clasifica este par como de la clase positiva y se predice como de la clase negativa. Esto puede ser un error del etiquetado ya que el Transformer realiza la predicción más lógica.

En el ejemplo 1, la pregunta es sobre si va a haber lluvia en Qatar y la respuesta no tiene nada que ver con dicha pregunta. Sin embargo, se predice la clase como positiva. Esto puede ser porque el Transformer entiende que se está aportando información relacionada pero no tiene claro si esta información es importante o no.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Was there any rain in Qatar yesterday?	Gtim.dont worry ..tiger wont stay in one place for a long.tomorrow he will go to wakra...u enjoy ur weekend in sheraton park.	Mala	Buena
2	Do you know of a store where i can buy one?\n\n	my skin costs a life :D\n\nKaput check with dwell in landmark.\n\n\n<img src="http://www.qatarliving.com/files/images/Da.gif">\n\n	Mala	Buena
3	hi\nwhat is the diffrenc between fake ed-hardy clothes and caps --and the origin ed hardy??\nis ther is a diffrence?\nfro m where can i buy fake ed hardy stuff?	where they come from and price .....may be	Buen a	Mala
4	you for your inputs!!	of course gold..	Buen a	Mala

Tabla 97: Ejemplos mal clasificados Cross-Encoder OVA SemEval 2015.

SemEval 2017

En las Tablas 90 y 91, se pueden ver las matrices de confusión del Cross-Encoder OVA con stsb-distilbert-base y paraphrase-MiniLM-L6-v2 respectivamente con el SemEval 2017. A su vez, en las Tablas 92 y 93, se pueden ver las métricas por clase con dichos modelos en el SemEval 2017. Como en todos los casos de este dataset, es notorio que no hay ningún caso etiquetado como potencial por lo que sus métricas son 0. Se puede ver que entre la clase negativa y la positiva hay cierto balance ganando ligeramente la clase negativa en métricas.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3365	1080	390
	Potencial	0	0	0
	Positive	485	488	2992

Tabla 98: Matriz de confusión del SemEval 2017 con Cross-Encoder OVA stsb-distilbert-base.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	3359	1072	377
	Potencial	0	0	0
	Positive	504	507	2981

Tabla 99: Matriz de confusión del SemEval 2017 con Cross-Encoder OVA paraphrase-MiniLM-L6-v2.

	F1-score	Recall	Precision
Clase negativa	0.83329235	0.79872732	0.87098432
Clase potencial	0	0	0
Clase positiva	0.76756491	0.68444152	0.87366947

Tabla 100: Métricas por clase SemEval 2017 con Cross-Encoder OVA stsb-distilbert-base.

	F1-score	Recall	Precision
Clase negativa	0.82426361	0.77822296	0.87609445
Clase potencial	0	0	0
Clase positiva	0.75951579	0.66776388	0.88049769

Tabla 101: Métricas por clase SemEval 2017 con Cross-Encoder OVA paraphrase-MiniLM-L6-v2.

En la Tabla 94 se pueden ver ejemplos mal clasificados por el modelo en el SemEval 2017. En el ejemplo 1, se puede ver que en la pregunta se pregunta cuál es su sitio favorito chill de Doha y se responde roof. La clase real de este ejemplo es mala. Sin embargo, la clase predicha es buena porque el modelo entiende que roof es un sitio chill.

En el ejemplo 4, se pregunta cuál es el mejor sitio para salir en Qatar. La respuesta va en broma y dice que lo mejor es dar una vuelta por los caminos del estadio a la noche. La clase real es buena pero la clase predicha es mala.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	What's your favorite chill out place in Doha?	roof	Mala	Buena
2	I know of people who are illegally in this country, runaways , expired visas and they up to no good. Where do I report this anonymously?	ok follow the laws...there is no law asking you to report these issues unless they ran away from your sponsorship... if they did to another person then he will report it.	Mala	Buena
3	Is it possible to block a cell phone number which is calling you in Qtel? I am being disturbed by a person, being miss called several times. I would like to hear also your advices. I dont want to change my number because I am using it since 3 years.	Service available from 3yrz in Pakistan, dont know here, neither can hope =p Y ousr I	Buena	Mala
4	which place is the best to hang out in qatar?	riding around the paths of the stadium in the eve...	Buena	Mala

Tabla 102: Ejemplos mal clasificados Cross-Encoder OVA SemEval 2017.

AmazonQA

En las Tablas 95 y 96, se pueden ver las matrices de confusión del Cross-Encoder OVA con stsb-distilbert-base y paraphrase-MiniLM-L6-v2 respectivamente con el dataset AmazonQA. A su vez, en las Tablas 96 y 97, se pueden ver las métricas por clase con dichos modelos en el dataset AmazonQA. Se puede ver como en casos anteriores, la clase negativa tiene muchos ejemplos con respecto al resto de clases. Es por esto por lo que se crea un sobreentrenamiento y que la clase que mejor se predice, o la única que tiene buenas métricas, es la clase negativa. Este es el único caso en el que el stsb-distilbert-base tiene algo positivo que es que las clases están más equilibradas en cuanto a métricas.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	217607	28219	50
	Potencial	79436	13668	1181
	Positive	14562	6889	782

Tabla 103: Matriz de confusión del AmazonQA con Cross-Encoder OVA stsb-distilbert-base.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	245499	13	364
	Potencial	90410	12	263
	Positive	23496	19	518

Tabla 104: Matriz de confusión del AmazonQA con Cross-Encoder OVA paraphrase-MiniLM-L6-v2.

	F1-score	Recall	Precision
Clase negativa	0.78068144	0.88503022	0.69834378
Clase potencial	0.1985863	0.15071952	0.29100664
Clase positiva	0.06006145	0.03254672	0.38847491

Tabla 105: Métricas por clase AmazonQA con Cross-Encoder OVA stsb-distilbert-base.

	F1-score	Recall	Precision
Clase negativa	0.81118213	0.99844234	0.68307063
Clase potencial	0.00026452	0.00013233	0.27272727
Clase positiva	0.04115684	0.02155908	0.45240175

Tabla 106: Métricas por clase AmazonQA con Cross-Encoder OVA paraphrase-MiniLM-L6-v2.

En la Tabla 99, se ven los ejemplos mal clasificados por el Cross-Encoder con OVA en el dataset AmazonQA. En el ejemplo 1, se explica perfectamente lo que se pregunta. Sin embargo, a pesar de que la etiqueta es que es de la clase buena, la clase predicha es negativa. Esto es debido a cierto sobreentrenamiento por parte del modelo. Además, el modelo como no cuenta con las reviews no puede saber si la respuesta es buena o no lo es. En el ejemplo 2, se puede ver otro ejemplo de lo mismo.

En el ejemplo 3, se da una respuesta a si ese producto tiene los mismos beneficios que el aceite de coco. La respuesta es muy explicativa sobre el tema y es por esto por lo que se clasifica como buena. Sin embargo, el ground truth es negativo. Esto es probablemente por la mala etiquetación de este dataset.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	what is the eye slit thingy for? sorry if that sounds like a dumb question	It's used for sighting the course you intend to follow. As you turn the compass to the magnetic direction you want to go the sight wire shows you where that direction is in the distance or on the horizon. It gives you a more precise reading to follow.	Buena	Mala
2	I know of people who are illegally in this country, runaways , expired visas and they up to no good. Where do I report this anonymously?	ok follow the laws...there is no law asking you to report these issues unless they ran away from your sponsorship... if they did to another person then he will report it.	Buena	Mala
3	Does this have the same health benefits as coconut oil?	This comes from the sap of either the coconut flower or tree, usually the flower. While the benefits are not the same as coconut oil, it does have slightly fewer calories than cane sugar or brown (cane) sugar. One huge benefit is the lower glycemic index it has, which is the very best benefit to those who need to watch those numbers (diabetics and those sensitive to processed sugar). I love this, you can bake with it 1:1 to processed sugar, it tastes great in coffee, on oats, in cereal. I use it to make sugar-free ice cream. The only thing is that if used in something normally "light" in color it will look like you added caramel to it! That does not bother me as I prefer SWEETNESS to color! I love coconut palm sugar and wish I had found it years ago!	Mala	Buena

4	How does Lithium battery compare to NiMH battery? I can get 2 XRC batteries for about $100, is it worth twice the price?	You asked two questions, first a comparison between NiMN and the Lithium. There really is no comparison, the Lithium by far is a better battery. Your second question is it worth twice the price. That depends on several factors. First do you make your living with this tool? Another factor how do you use this tool? Lets say your working upon a scaffold, your getting up and down and if you have co-workers upon the scaffold helping you, then it might be worth the additional price. But if your working in your shop and a charger is handy, then most likely the price is too high. If you still want one, wait until they go on sale. I got my Lithium around Christmas. The sale price justified the additional cost.	Mala	Buena

Tabla 107: Ejemplos mal clasificados Cross-Encoder OVA AmazonQA.

5.4.3 Entrenar Cross-Encoder con 2 clases

En este caso, se va a entrenar un Cross-Encoder para diferenciar la clase buena y la mala. Como en otras propuestas, se ha unido la clase potencial y la buena en una sola para tratar de evitar los problemas a la hora de diferenciar estas dos clases. Se han usado dos pesos para el Transformer inicial: el stsb-distilbert-base-v3 y el paraphrase-MiniLM-L6. Se puede ver en las Tablas 134 y 135 como en todas los datasets paraphrase-MiniLM-L6 ofrece mejores métricas que el otro.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.85113541	0.8208502	0.86941581	0.83360791	0.92738416
SemEval 2017	0.85432823	0.84204545	0.8944481	0.81765296	0.88412582
Amazon QA	0.0608933	0.68513619	0.03208906	0.59486102	0.78661344

Tabla 108: Métricas por dataset con Cross-Encoder stsb-distilbert-base-v4 y 2 clases.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.86116867	0.83046559	0.89261168	0.83186549	0.9413247
SemEval 2017	0.85008855	0.83647727	0.89532587	0.8092027	0.90151069
Amazon QA	0.35845482	0.59908096	0.35208174	0.36506287	0.76003994

Tabla 109: Métricas por dataset con Cross-Encoder paraphrase-MiniLM-L6-v2 y 2 clases.

SemEval 2015

En la Tabla 136, se pueden ver ejemplos mal clasificados por el Cross-Encoder en el SemEval 2015. En el ejemplo 1, la respuesta parece provenir de la misma persona que pregunta. En esta respuesta se aporta más información, pero no responde a la pregunta. Sin embargo, la etiqueta asociada es que es de la clase positiva, aunque el modelo clasifica la respuesta como mala.

En el ejemplo 2, pasa lo mismo que en el ejemplo 1. La respuesta es una reformulación por parte del que formula la pregunta. En este caso, sí que esta etiquetada la respuesta como mala pero el modelo predice dicha respuesta como buena entendiendo que aporta información.

En el ejemplo 3, se pregunta dónde se puede encontrar a una persona que haga tatuajes. La respuesta no es más que una reformulación de la pregunta inicial. Al igual que en el ejemplo 1, el par pregunta respuesta se clasifica como de la clase positivo, aunque el modelo no se deja engañar y lo clasifica como negativo.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	If I lost my cat, how can i find her?	I have this persian cat that lived with me for over 7years, and after we moved out the house, I placed her only for a night in our neighbours. The next day I came and my cat was gone. Shes been away for past 4 days now. I hope no1 took her and sell her.\nShes a female gray perian cat, without a tail.\n\nSo how can I find her? Any Ideas?	Buena	Mala
2	hi,can anyone give me an idea about grade 109 pay scale in hamad medical cooperation?	hi,can anyone give me an idea about grade 109 pay scale in hamad medical cooperation? If HR give me an offer can i give it for a review? will it affect the current offer?	Mala	Buena
3	where I can find a tattooist?	I would like to get a tattoo on the back, does anyone know a pro tattooist here in qatar? please pm me if you do.\nthanks	Buena	Mala
4	Tips on saving money in Qatar	So, I have seen that I have been spending money more than usual. So, I need to cut back on expenses!!\n\nWhat are your tips on saving money here?	Mala	Buena

Tabla 110: Ejemplos mal clasificados por Cross-Encoder con 2 clases en SemEval 2015.

SemEval 2017

En la Tabla 137, se pueden ver ejemplos mal clasificados por el modelo en el SemEval 2017. En el ejemplo 1, se pregunta sobre en qué habitación pasa la gente más tiempo. Se responde que en la cama porque se pasa todo el día durmiendo. La respuesta da la información que requiere la pregunta y está clasificada como buena. Sin embargo, por el estilo de comunicación el modelo entiende que es una broma y lo clasifica como de la clase negativa.

En el ejemplo 2, se pregunta sobre cuanto tarda en procesarse el visado en Grecia. La respuesta tiene relación, pero no contesta sobre el tiempo que tarda, sino sobre el porqué tarda tanto. Por tanto, la clase real se ha clasificado como mala, aunque el modelo, entendiendo que la respuesta aporta información adicional relevante para el usuario que ha planteado la pregunta, clasifica la respuesta como buena.

En el ejemplo 3, se pregunta sobre cuál es la habitación en la que pasa más tiempo. En la respuesta se responde que, en el baño, pero se incluye una broma. Al igual que en el ejemplo 1, la etiqueta real es que es de la clase positiva, pero al incluir una broma el modelo se confunde y clasifica la respuesta como de la clase negativa.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	What room of your home do you spend the most time in?	my bedroom coz i like sleeping and daydreaming. LOL!!!	Buena	Mala
2	ou know how long it will take to get Schengen Visa from Embassy of Greece? 3 days? 1 week? any idea? Thanks.	es UKEng, only joined recently. I think that is why their visa application process took ages.	Mala	Buena
3	What room of your home do you spend the most time in?	for me its my bathroom, I spent 15 to 30 minutes using..im just playing basketball always...LOLZ! Im serious...	Buena	Mala
4	Do you know how long it will take to get Schengen Visa from Embassy of Greece? 3 days? 1 week? any idea? Thanks.	You right UK is not part of the Schengen Novi. Uk maintains Border Controls. Even EU citizens arriving from within EU have their passport checked at their departure point before flying to the UK. ---------- HE WHO DARES WINS	Mala	Buena

Tabla 111: Ejemplos mal clasificados por Cross-Encoder con 2 clases en SemEval 2017.

AmazonQA

En la Tabla 138, se ven ejemplos incorrectamente clasificados por el modelo en AmazonQA. En el ejemplo 1, se pregunta si se necesita comprar elementos adicionales para el producto. En la respuesta se responde que el producto viene con ciertas cosas. Esta información aporta y por eso la clase real es positiva pero como faltan las reviews, el modelo no es capaz de vincular la respuesta con la pregunta y le asigna la clase negativa.

En el ejemplo 2, la pregunta es sobre la diferencia entre dos productos. En la respuesta el usuario dice que ha mirado las especificaciones y enumera las diferencias. La clase real es negativa, aunque realmente la respuesta es útil. El modelo entiende que la respuesta es útil y clasifica la respuesta como buena.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Do I need to purchase any additional items for this product?	It comes with some laminating sheets. I was not impressed with them. Now I am buying smaller sheets at Staples that are actually Staples brand and I like them so much more.	Buena	Mala
2	What is the difference between AC1200 and AC600?	I don't have either, but from a direct look at the specs, there's a few differences. The AC600 is USB 2, while the AC1200 is USB 3. USB 3 supports much higher data rates and is probably needed to maximize the full network bandwidth potential of a solid 802.11 AC router. The AC1200 spec sheet also has a line about supporting "both 802.11 AC and 802.11 N", and the AC 600 doesn't have that line, so I'm guessing the AC1200 has dual-radios and can support a wider variety of configurations.	Mala	Buena
3	What ISO range is covered when in set on auto-ISO? Could I set on shutter preferre at 1/400 sec where the auto ISO will adjust? What range covered?	I believe it is 12800 or 25600 by default. I set mine to 12800 as max due to noise. But you can change the auto ISO range. Yes you can set the shutter speed to any setting and the auto ISO will compensate for the shutter speed. My only warning is at high shutter speeds except in good light, the ISO may get pushed to a point a lot of noise is introduced.	Buena	Mala

4	Is this product good for "seniors" and also for slightly oily skin?	Hi Katie:The Specific Beauty product line is designed for all age skin types and it can be used for oily skin. The Daily Hydrating lotion is a non-oily day lotion with spf which works wonderfully by itself or under foundation. The Skin Brightening Serum and Night Treatment both contain retinol so they will help to dry the skin, exfoliate then hydrate while erasing dark spots and evening skin tone. I hope that I have answered your inquiry to your satisfaction. Please let me know how I can be of further assistance. Have a great night!	Mala	Buena

Tabla 112: Ejemplos mal clasificados por Cross-Encoder con 2 clases en AmazonQA.

5.5 Otras propuestas

En esta Sección, se presentan propuestas para atajar el CQA que no se pueden clasificar ni como Bi-Encoders ni como Cross-Encoders. En este caso, la predicción es siempre sobre 3 clases.

5.5.1 Función de composicionalidad basada en teoría de la información

Esta aproximación se basa en la aproximación de Amigo et al. [85]. Esta se basa en que los vectores de significado de una unidad sintáctica más compleja se pueden formar mediante la combinación de los vectores de las palabras que la forman. Para ello, se ha aplicado la función de composicionalidad de manera secuencial de izquierda a derecha. Hay otras maneras de aplicarla como de derecha a izquierda o siguiendo estructuras sintácticas. La fórmula se escribe de la siguiente manera:

$$F_{\lambda,\mu}(\overrightarrow{v_1},\overrightarrow{v_2}) = \frac{\overrightarrow{v_1}+\overrightarrow{v_2}}{\|\overrightarrow{v_1}+\overrightarrow{v_2}\|}\cdot\sqrt{\lambda(\|\overrightarrow{v_1}\|^2+\|\overrightarrow{v_2}\|^2)-\mu\langle\overrightarrow{v_1},\overrightarrow{v_2}\rangle} \quad (49)$$

En nuestro experimento se ha elegido $\lambda = 1$ y $\mu = \frac{min(\|\overrightarrow{v_1}\|,\|\overrightarrow{v_2}\|)}{max(\|\overrightarrow{v_1}\|,\|\overrightarrow{v_2}\|)}$ debido a que satisface las propiedades de composición expuestas en Amigo et al. Esto permite crear un modelo no supervisado que pone en un mismo espacio las representaciones de las palabras y la de las frases. Las propiedades que sigue esta función y que permite construir un espacio semántico composicional son las siguientes:

1. **La composición del elemento neutro:** Si se añade el elemento neutro no se modifica la composición.
2. **Límite inferior de la norma de composición**: La norma del vector composición siempre es mayor o igual que la norma de cada componente. Con esto se consigue que no se reduzca el contenido de información.
3. **Monotonía de la norma de la composición**: La norma del vector composición es monótona con respecto al ángulo de los vectores que lo componen. Es decir, a igual norma de los vectores que componen otro, la norma del vector composición será menor cuanto menor sea el ángulo entre los vectores que lo componen.
4. **Sensible a la estructura**: No cumple la propiedad asociativa por lo que el orden de composición es importante.

Para calcular la similitud entre los vectores de pregunta y respuesta, se han usado dos métricas. Por un lado, se ha usado la distancia del coseno que ya se usado anteriormente en este trabajo. Por otro lado, se ha usado la métrica de ICM propuesta por Amigo et al. [85]. Se pueden ver las métricas de ambas aproximaciones en las Tablas 39 y 40. Como se puede ver la función coseno para calcular la similitud funciona mejor en todas las métricas y en todos los datasets. Su funcionamiento es bastante malo debido a probablemente a dos razones:

- En primer lugar, como ya se ha explicado anteriormente, la relación entre pregunta y respuesta no es exactamente la misma que entre dos textos similares. Es por eso por lo que empeora con respecto a otros métodos entrenados que buscan estos otros tipos de relaciones entre ambos.
- En segundo lugar, el modelo trata de componer el significado de una frase a partir del significado de las partes. Esto no suele ser cierto ya que cada palabra puede tener una acepción diferente dependiendo del contexto en el que se encuentran. En esto, los Transformers son muy útiles a la hora de extraer este tipo de información. No es un problema propiamente de la función de composición sino de Word2Vec, pero se arrastra a la función.

Para evitar el problema de las anteriores aproximaciones no supervisadas se ha cogido como clase negativa todas las respuestas cuyo coseno o ICM estuve entre el percentil 0 y el 33 de los cosenos. En el caso de la clase potencial, los cosenos han ido desde el percentil 33 hasta el percentil 66. Por último, en el caso de la clase buena los percentiles van desde 66 hasta el 1.

Una ventaja de este modelo, a pesar de sus malos resultados, es la gran velocidad de cálculo ya que no requiere entrenamiento ni complejas arquitecturas para extraer la predicción.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.30608799	0.33912155	0.34087604	0.3371204	0.8049374
SemEval 2017	0.26161763	0.32822507	0.21841661	0.32621603	0.6387611
Amazon QA	0.27574016	0.32294261	0.33741558	0.32545033	0.75405059

Tabla 113: Métricas de la composicionalidad con ICM.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.34594084	0.38610827	0.37345813	0.38412716	0.81238354
SemEval 2017	0.30424944	0.38254803	0.38107136	0.25323203	0.68521565
Amazon QA	0.30331894	0.35372112	0.37623562	0.35642166	0.76520077

Tabla 114: Métricas de la composicionalidad con coseno.

SemEval 2015

En las Tablas 41 y 42, se ven las matrices de confusión de SemEval 2015. Del mismo modo, en las Tablas 43 y 44 se pueden observar las métricas de dichas matrices de confusión para dicho dataset. Se puede observar que a pesar de que las clases están mejor distribuidas que en el resto de los modelos no supervisados, no logra una buena predicción. Se puede ver que funciona ligeramente mejor el coseno para todas las clases que el ICM.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	251	293	261
	Potencial	57	58	50
	Positive	338	295	355

Tabla 115: Matriz de confusión de composicionalidad con ICM sobre SemEval 2015.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	315	275	215
	Potencial	43	56	66
	Positive	288	315	385

Tabla 116: Matriz de confusión de composicionalidad con coseno sobre SemEval 2015.

	F1-score	Recall	Precision
Clase negativa	0.3459683	0.31180124	0.38854489
Clase potencial	0.14303329	0.35151515	0.08978328
Clase positiva	0.42926239	0.35931174	0.53303303

Tabla 117:Métricas de composicionalidad con ICM sobre SemEval 2015.

	F1-score	Recall	Precision
Clase negativa	0.43418332	0.39130435	0.4876161
Clase potencial	0.13810111	0.33939394	0.08668731
Clase positiva	0.46553809	0.38967611	0.57807808

Tabla 118: Métricas de composicionalidad con coseno sobre SemEval 2015.

En la Tabla 45, se pueden ver algunos ejemplos mal clasificados por el modelo de composicionalidad para el SemEval 2015. Es notable el ejemplo 3, se puede ver que en este dataset a veces existen ciertas preguntas que no tienen sentido. Sigue habiendo casos como en el ejemplo 3, que se pregunta algo claro y la respuesta es muy clara también, aunque se predice que la clase del comentario es malo.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	hi,can anyone give me an idea about grade 109 pay scale in hamad medical cooperation? If HR give me an offer can I give it for a review? Will it affect the current offer?	Pay scale of 108 is 6500 so how can it be the same for 109?	Mala	Buena
2	My 14 year old son is 149nterested in learning how to play a guitar. Where can I buy one cheaply and where can he get lessons at a reasonable price.	Guitars are available at Badie Studio & Stores at Old Salata...	Buena	Mala
3	Do you know of a store where I can buy one?	Recently bought a string with golden chain on front :D <img src=”http://www.qatarliving.com/files/images/Da.gif”>	Buena	Mala
4	lol 😊	hehehe	Mala	Potencial

Tabla 119: Ejemplos mal clasificados del SemEval 2015 por la composicionalidad.

SemEval 2017

En las Tablas 46 y 47, se pueden ver las matrices de confusión de la composicionalidad con ambas medidas en el SemEval 2017. A su vez, en las Tablas 47 y 48, se observan las métricas de dichas matrices de confusión. La métrica del coseno funciona mejor en tanto a la hora de balancear la calidad de la predicción entre las clases como en calidad de las predicciones. Es notable como llevamos repitiendo en este trabajo que en la clase potencial en este dataset no hay ningún ejemplo y por eso salen 0 también sus métricas.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	1311	1663	1237
	Potencial	0	0	0
	Positive	1575	1339	1559

Tabla 120: Matriz de confusión de composicionalidad con ICM sobre SemEval 2017.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	1609	1374	1228
	Potencial	0	0	0
	Positive	1277	1520	1736

Tabla 121: Matriz de confusión de composicionalidad con coseno sobre SemEval 2017.

	F1-score	Recall	Precision
Clase negativa	0.36945188	0.31132748	0.45426195
Clase potencial	0	0	0
Clase positiva	0.41540101	0.34392235	0.52438614

Tabla 122: Métricas por clase con composicionalidad e ICM sobre SemEval 2017.

	F1-score	Recall	Precision
Clase negativa	0.44962973	0.37672676	0.55751906
Clase potencial	0	0	0
Clase positiva	0.46311858	0.38296934	0.58569501

Tabla 123: Métricas por clase con composicionalidad y coseno sobre SemEval 2017.

En la Tabla 50 se pueden ver ejemplos mal clasificados por el SemEval 2017. Se puede ver que se clasifica el ejemplo 1 como clase buena a pesar de ser mala. Este ejemplo es poco interpretable porque no se entiende cual ha sido el criterio de Word2vec para determinar que esas palabras están de alguna manera cercana.

En el ejemplo 3, se pregunta si se le va a banear de entrar en GCC si vuelve a su casa durante su periodo de contratación en Qatar y se le responde que solo en Qatar. Esta predicho como clase negativa. Esto puede ser por la composicionalidad ya que, aunque tiene palabras en común, al ser la pregunta mucha más larga que la respuesta se pierde la semántica de las palabras que comparten. En un ejemplo como este un BOW hubiese funcionado de manera correcta, por ejemplo.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	o you know of a store where i can buy one?	FS i'll bet my life on it convince Da for me:-).	Mala	Buena
2	Just curious what types of prescription drugs i should think about bringing over in case they are difficult to obtain in Doha. Are Anxiety Suppressants and painkillers such as Xanax, Vallium and Vicodin available through health care providers? What about something like birth control?	I believe that Prozac is the only antidepressant/antianxiety med available. It is over-the-counter. Mandi	Buena	Mala
3	Will I get a GCC ban Ban if I go on leave and doesn't come back during my contract period in Qatar..? please give me an answer...	just a ban in Qatar. Not GCC	Buena	Mala
4	Anyone here who's a Filipino working for Hamad Medical Center? Was it ok to work there? As a Filipino who will be getting QR 6,500 is that enough?	hi there delldude... u might as well wanna check this.. http://qatarliving.com/group/filipino-expatriates-in-qatar	Mala	Buena

Tabla 124: Ejemplos mal clasificados del SemEval 2017 por la composicionalidad.

AmazonQA

En las Tablas 51 y 52, se pueden ver las matrices de confusión del experimento de composicionalidad con las diferentes clases en el dataset AmazonQA. Se puede observar que al igual que en el caso del SemEval 2017, esta más balanceado y con mejores resultados en el caso de la medida del coseno. Es cierto que no se el balanceo no se nota tanto debido a que la clase negativa está muy por encima del resto de clases en cantidad de ejemplos.

Actual value \ Predicted value	Negative	Potencial	Positive
Negative	78606	84221	81258
Potencial	30761	27975	31432
Positive	8830	6001	9088

Tabla 125: Matriz de confusión de composicionalidad con ICM sobre AmazonQA.

Actual value \ Predicted value	Negative	Potencial	Positive
Negative	85712	79759	78614
Potencial	30761	30464	32648
Positive	5429	7973	10517

Tabla 126: Matriz de confusión de composicionalidad con coseno sobre AmazonQA.

	F1-score	Recall	Precision
Clase negativa	0.43394924	0.32204355	0.66504226
Clase potencial	0.26851918	0.31025419	0.23668113
Clase positiva	0.12475205	0.37994899	0.0746276

Tabla 127: Métricas por clase con composicionalidad e ICM sobre AmazonQA.ç

	F1-score	Recall	Precision
Clase negativa	0.47317835	0.35115636	0.72516223
Clase potencial	0.29241136	0.33785822	0.25774138
Clase positiva	0.14436712	0.43969229	0.08636136

Tabla 128: Métricas por clase con composicionalidad y coseno sobre AmazonQA.

En la Tabla 55, se pueden observar algunos de los ejemplos mal clasificados por la composicionalidad con las dos medidas en el dataset AmazonQA. En el ejemplo 2, se ve claramente que se clasifica como mala una respuesta si no tiene palabras que tengan una cercanía en el espacio de word2vec con alguna palabra de la pregunta. En este caso, en la respuesta la única palabra con carga semántica es la palabra cheaper que significa barato y no hay ninguna palabra similar en la pregunta.

En cambio, en el ejemplo 4, se repiten las palabras con carga semántica block y blue en la pregunta y en la respuesta. Esto hace que la respuesta se clasifique como buena, aunque realmente la clase es mala.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Does this bottle include a tube to put in the top for hard to reach area's?	Yes. It comes with a straw-like tube and it works great for hard to reach areas.	Mala	Buena
2	Why does the automotive industry still use the old lead-acid battery witch should be in a museum and not a car?	Because it is cheaper	Buena	Mala
3	Is this opener compatible with "Home-Link" systems?	Did not work with my 2011 BMW 3 series (had to get the repeater). Did work with my 2014 Honda Odyssey	Buena	Mala
4	Do they block blue?	Sorry, I don't know what that means? Do they block blue?	Mala	Buena

Tabla 129: Ejemplos mal clasificados del AmazonQA por la composicionalidad.

5.5.2 Entrenar mezcla Bi-Encoder para separar clase buena-potencial y mala y crosencoder para separar clase buena de la potencial

La motivación de esta propuesta se debe a que en los experimento anteriores la separación entre clase buena y mala se ha realizado mejor con los Bi-Encoders mientras que la separación entre buena y potencial se realiza mejor con los Cross-Encoders. Por ello, en este experimento se va a probar el rendimiento de un Bi-Encoder seguido de un Cross-Encoder para lograr una mejor separación entre la clase buena y la mala y entre la clase buena y la potencial.

El Bi-Encoder se ha entrenado uniendo la clase potencial y la positiva. De esta manera, utilizando el cosine embedding loss, se consigue que los pares pregunta-respuesta mejores estén por encima de 0.5 de coseno y los pares negativos estén por debajo. Este Bi-Encoder se ha entrenado durante 10 épocas, guardando los pesos que diesen mejor accuracy en validación.

Por otro lado, el Cross-Encoder ha entrenado con el subconjunto de respuestas potenciales y positivas de conjunto de train. De esta manera y con el cross entropy loss, se consigue separar la clase potencial de la clase buena.

Se puede ver que, excepto en el MAP que coinciden ambos modelos porque solo depende del Bi-Encoder, las métricas salen mejor con el paraphrase-MiniLM-L6-v2 en SemEval 2015 y AmazonQA. En el caso del SemEval 2017, es un poco mejor el stsb-distilbert-base-v4. También se pude ver que este ensemble no mejora los resultados con respecto a los modelos más básicos con tan solo un Cross-Encoder para separar las 3 clases.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.54789118	0.74848178	0.55416599	0.57758637	0.92345248
SemEval 2017	0.51645048	0.72261364	0.48255414	0.55634225	0.85647433
Amazon QA	0.33666383	0.64320538	0.35120627	0.47287025	0.77684889

Tabla 130: Metricas por dataset de mezcla Bi-Encoder y Cross-Encoder stsb-distilbert-base-v4.

	F1-score	Accuracy	Recall	Precision	Map
SemEval 2015	0.54862902	0.75455466	0.55651636	0.57710795	0.92345248
SemEval 2017	0.51080767	0.70465909	0.47099683	0.56025972	0.85647433
Amazon QA	0.34644306	0.64355757	0.35609882	0.45927511	0.77684889

Tabla 131: Metricas por dataset de mezcla Bi-Encoder y Cross-Encoder paraphrase-MiniLM-L6-v2.

SemEval 2015

En las Tablas 119 y 120, se pueden ver las matrices de confusión con el modelo de stsb-distilbert-base-v4 y paraphrase-MiniLM-L6-v2 respectivamente en el SemEval 2015. En las Tablas 121 y 122 se pueden ver las métricas asociadas a dichas matrices de confusión en el SemEval 2015. Se puede observar que como el modelo de Bi-Encoder permanece inmutable la clase negativa no cambia. Sin embargo, en la clase positiva y potencial se ve que funciona mejor el modelo de paraphrase-MiniLM-L6-v2. La clase más problemática sin duda es la clase potencial que no se detecta en ninguno de los dos casos de manera satisfactoria.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	609	12	191
	Potencial	50	8	109
	Positive	116	19	862

Tabla 132: Matriz de confusión por dataset de mezcla Bi-Encoder y Cross-Encoder stsb-distilbert-base-v4 con SemEval 2015.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	609	23	180
	Potencial	50	7	110
	Positive	116	6	875

Tabla 133: Matriz de confusión por dataset de mezcla Bi-Encoder y Cross-Encoder paraphrase-MiniLM-L6-v2 con SemEval 2015.

	F1-score	Recall	Precision
Clase negativa	0.76748582	0.75	0.78580645
Clase potencial	0.0776699	0.04790419	0.20512821
Clase positiva	0.79851783	0.86459378	0.74182444

Tabla 134: Métricas por clase por dataset de mezcla Bi-Encoder y Cross-Encoder stsb-distilbert-base-v4 con SemEval 2015.

	F1-score	Recall	Precision
Clase negativa	0.76748582	0.75	0.78580645
Clase potencial	0.06896552	0.04191617	0.19444444
Clase positiva	0.80943571	0.8776329	0.75107296

Tabla 135: Métricas por clase por dataset de mezcla Bi-Encoder y Cross-Encoder paraphrase-MiniLM-L6-v2 con SemEval 2015.

En la Tabla 123, se puden ver ejemplos mal clasificados por el modelo en el SemEval 2015. En el ejemplo 1, se pude ver como se pregunta sobre donde se puede comprar Abaya. La respuesta es clara y se responde donde se puede comprar dicho componente. Sin embargo, la etiqueta es que el par pregunta-

respuesta es malo. Este es un problema de etiquetado. El modelo por el contrario da la clasificación que parece lógica, que el par pregunta y respuesta es de la clase positiva.

En el ejemplo 2, se pregunta sobre si alguien puede dar un idea sobre el rango de salario del grado 9 en una organización. En la respuesta se dice que en otro grado se paga un dinero y que no puede ser el mismo para el grado 109. Este par pregunta y respuesta están etiquetados como de la clase negativa. Sin embargo, la clase real predicha es buena porque no termina de entender el modelo a que se esta refiriendo.

En el ejemplo 3, se puede ver como el modelo se confunde cuando hay elementos desconocidos como html. En la respuesta, la cual es satisfactoria y esta etiquetada como de la clase buena, hay un código en html lo que hace que el modelo clasifique dicha pregunta como negativa.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	Iam new to Doha and want to buy Abaya, please tell me the address for any abaya stores .	yea go down town Souq Faleh or Souq Waqif	Mala	Buena
2	hi,can anyone give me an idea about grade 109 pay scale in hamad medical cooperation? If HR give me an offer can i give it for a review? will it affect the current offer?	pay scale of 108 is 6500 so how can it be the same for 109?	Mala	Buena
3	Is our manager still sleeping?	Yes azi getting better thanks<pre> \n \n \n \n \n \n \n \n \n \n \n [img_assist| nid=50852 |title=hmm| desc=|link=none |align=left| width=|height=0] \n \n</pre>\n	Buena	Mala
4	the car which i said is for sale has been sold out. How do I remove my advert ??	It's simple, Post that ADD again and WRITE IN CAPITAL "SOLD OUT" ..\n\nGood luck !	Buena	Mala

Tabla 136: Ejemplos mal clasificados de mezcla Bi-Encoder y Cross-Encoder en SemEval 2015.

SemEval 2017

En las Tablas 124 y 125, se pueden ver las matrices de confusión con el modelo de stsb-distilbert-base-v4 y paraphrase-MiniLM-L6-v2 respectivamente en el SemEval 2017. En las Tablas 126 y 127 se pueden ver las métricas asociadas a dichas matrices de confusión en el SemEval 2017. Se puede ver que no hay ningún ejemplo realmente de la clase potencial, por lo que las métricas son 0. Por otro lado, la clase positiva y la clase negativa están balanceadas en métricas. Funciona ligeramente mejor en la clase positiva el modelo

ststb-distilbert-base-v4, aunque probablemente pase esto debido a que ningún ejemplo pertenece a la clase potencial.

Actual value \ Predicted value	Negative	Potencial	Positive
Negative	3216	430	597
Potencial	0	0	0
Positive	665	749	3143

Tabla 137: Matriz de confusión de mezcla Bi-Encoder y Cross-Encoder stsb-distilbert-base-v4 con SemEval 2017.

Actual value \ Predicted value	Negative	Potencial	Positive
Negative	3216	509	518
Potencial	0	0	0
Positive	665	907	2985

Tabla 138: Matriz de confusión de mezcla Bi-Encoder y Cross-Encoder paraphrase-MiniLM-L6-v2 con SemEval 2017.

	F1-score	Recall	Precision
Clase negativa	0.79172821	0.75795428	0.82865241
Clase potencial	0	0	0
Clase positiva	0.75762324	0.68970814	0.84037433

Tabla 139: Métricas por clase de mezcla Bi-Encoder y Cross-Encoder stsb-distilbert-base-v4 con SemEval 2017.

	F1-score	Recall	Precision
Clase negativa	0.79172821	0.75795428	0.82865241
Clase potencial	0	0	0
Clase positiva	0.74069479	0.65503621	0.85212675

Tabla 140: Métricas por clase de mezcla Bi-Encoder y Cross-Encoder paraphrase-MiniLM-L6-v2 con SemEval 2017.

En la Tabla 128, se pueden ver ejemplos mal clasificados por el modelo en el SemEval 2017. En el ejemplo 1, se puede como se pregunta sobre la opinión sobre Facebook de los demás usuarios. La respuesta es la opinión de un usuario por lo que la clase real es buena. Sin embargo, se clasifica como de la clase negativa, probablemente por falta de entendimiento del modelo.

En el ejemplo 2, se pude ver como se pregunta sobre como conseguir el visado Schengen. Sin embargo, en la respuesta, solo se dice que no hay fronteras. La respuesta no responde correctamente a la pregunta, pero se clasifica de la clase positiva. Esto puede ser porque el modelo entiende que se esta dando más información al usuario que pregunta.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	What was wrong with the old version, new one keeps doing strange things & I'm getting annoyed with it!! What say you guys that are FB users?! Torque	the new look is like some web ads page. i dont think the layout team did a thorough user survey before launching the new FB.	Buena	Mala
2	Do you know how long it will take to get Schengen Visa from Embassy of Greece? 3 days? 1 week? any idea? Thanks.	I know that have recently joined the Eu a few years ago but I guess that means that there are also part of the countries within the EU zone without border controls. ---------- HE WHO DARES WINS	Mala	Buena
3	now i m eating food from restaurant.. one of the mess is available near my room... i m planning to join there...!! then i m confusioned is it good for my health, than restaurant food? But somebody told me to cook me, this is better than anyother...is it crrect ?	good Mess - Don't join. It already means confusion Confusioned: It is not good for ur health to eat. Next..	Buena	Mala
4	my friend is quite overweight these past months upto these days,.,,, and she want to loose so fast.... she s doing exercise and all, but she want ... u know, take some pills... so now, she find this xenical... i just want to know if anyone taking this pill and was it really effective? just want to know all the info before she buy something.... thank you :)	I checked on the Actrim...great info. I like that its all natural. Han, have you taken it? Actrim http://www.ameinfo.com/4263.html	Mala	Buena

Tabla 141: Ejemplos mal clasificados de mezcla Bi-Encoder y Cross-Encoder en SemEval 2017.

AmazonQA

En las Tablas 129 y 130, se pueden ver las matrices de confusión con el modelo de stsb-distilbert-base-v4 y paraphrase-MiniLM-L6-v2 respectivamente en el AmazonQA. En las Tablas 131 y 132 se pueden ver las métricas asociadas a dichas matrices de confusión en el AmazonQA. Se puede ver que como en todos los casos, la clase mejor predicha es la clase negativa. Esto ocurre debido al desbalanceo entre clases. Por otra parte, el Cross-Encoder determina mejor las respuestas que son positivas en el caso del paraphrase-MiniLM-L6. Esto se traduce en unas mejores métricas de manera general. Por tanto, el modelo que mejor funciona es el que de origen partia de los pesos del paraphrase-MiniLM-L6-v2.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	217613	28005	264
	Potencial	76513	13975	197
	Positive	17487	6192	348

Tabla 142: Matriz de confusión de mezcla Bi-Encoder y Cross-Encoder stsb-distilbert-base-v4 con AmazonQA.

		Predicted value		
		Negative	Potencial	Positive
Actual value	Negative	217613	27542	727
	Potencial	76513	13668	504
	Positive	17487	5758	782

Tabla 143: Matriz de confusión de mezcla Bi-Encoder y Cross-Encoder paraphrase-MiniLM-L6-v2 con AmazonQA.

	F1-score	Recall	Precision
Clase negativa	0.78068144	0.88503022	0.69834378
Clase potencial	0.20128622	0.15410487	0.29010629
Clase positiva	0.02802384	0.01448371	0.43016069

Tabla 144: Métricas por clase de mezcla Bi-Encoder y Cross-Encoder stsb-distilbert-base-v4 con AmazonQA.

	F1-score	Recall	Precision
Clase negativa	0.78068144	0.88503022	0.69834378
Clase potencial	0.1985863	0.15071952	0.29100664
Clase positiva	0.06006145	0.03254672	0.38847491

Tabla 145: Métricas por clase de mezcla Bi-Encoder y Cross-Encoder paraphrase-MiniLM-L6-v2 con AmazonQA.

En la Tabla 133, se pueden ver ejemplos mal clasificados por el modelo en el dataset AmazonQA. En el ejemplo 1, se pregunta cuantos carbohidratos se dan por cada comida. La respuesta responde a esta cuestión y por tanto la clase real es buena. Por el contrario, el modelo no entiende que se esté respondiendo a esa pregunta porque no tiene la review correspondiente y por ello la clasifica como de la clase mala.

En el ejemplo 3, se pregunta dónde esta producido el objeto en cuestión. En la respuesta no se responde específicamente eso, sino que se dice que no se ha producido en china. Es por esto por lo que la clase del ejemplo es positiva, pero se predice como mala. Es porque el Transformer no ve en la respuesta una respuesta valida a la pregunta formulada.

En el ejemplo 4, se pregunta si el objeto contiene plomo. La respuesta dice que no está seguro pero que lo más probable es que no contenga dicha sustancia. Como esta respuesta no ha sido de utilidad para el usuario, esta etiquetada como mala. Sin embargo, sí que aporta cierta información. Es por eso por lo que el modelo la clasifica como potencial.

	Pregunta	Respuesta	Clase Real	Clase Predicha
1	How many carbohydrates per serving?	The nutrition facts label as shown in one of the photos states there are 18.6 g carbs in each 3 lollipop serving (but, shhh, my kid doesn't know a serving equals 3!). If it helps make up your mind, I've been buying these for 2 or 3 years now as they are a favorite of my youngest son (age 13) and all my grandkids. I'm happy to be the grandma that always has organic lollipops in the cupboard, and I enjoy one myself every now and then!	Buena	Mala
2	Has anybody tried something different as a replacement for the mop sponge part? Something cheaper or better?	Have you tried the Libman Hardwood Microfiber Roller Mop Refill made of 100% sponge (not the old cellulose kind)?--it has the same dimentions as the refill for the Libman Nitty Gritty Roller Mop . I love 100% sponge mops bcuz they can be used on most any hard floor surface, dry fast/leave less water on the floor, are better for deep-clean scrubbing but don't scratch, and a lot less streaking than those little green dot things on a foam sponge. I discovered all this when I went looking for something more heavy-duty for my laminate floor; microfiber flat mop wasn't work'n for me in wanting squeaky "white socks" clean floors.	Mala	Buena
3	made in where?	What isn't made in China. .. but I can tell you this, it is by far one of the best sounding bluetooth systems out there,with amazing bass for its size, if you buy this you will not be disappointed. ..	Buena	Mala
4	I see that one comment says that this product contains lead - is that still true?	I really can't answer that forsure. It appears to be largely a plastic product. If there is lead I'm not sure where it would even be unless it's where the batteries go in the base of the unit. Sorry!	Mala	Potencial

Tabla 146: Ejemplos mal clasificados de mezcla Bi-Encoder y Cross-Encoder en AmazonQA.

6. Extracción de conclusiones sobre las propuestas

En esta sección, se comparan los resultados de los distintos experimentos de Community Question Answering de este trabajo. Los resultados de los experimentos de 2 y 3 clases no son comparables excepto en la métrica del Map. Por este motivo se han separado en distintas Tablas dichos experimentos.

Se muestran en la Tabla 147 los experimentos del dataset SemEval 2015 con 3 clases. La escala de color representa que cuanto más verde es una métrica más supera a otras propuestas. Por lo que se aprecia la métrica Map no está correlacionada por completo con el f1-score. En este dataset, la métrica importante era la F1-score, la que se evaluaba en el artículo. Se observa, que los métodos no supervisados funcionan peor que los supervisados.

Por otra parte, se puede observar que la arquitectura de Bi-Encoder y la de Cross-Encoder funcionan bien ambas. Sin embargo, en este caso, mejora las métricas el Bi-Encoder frente a los Cross-Encoders. Esto es porque en la clasificación los mejores resultados son con Bi-Encoders en todos los datasets. Probablemente, al modelar menos relaciones entre pregunta y respuesta tenga una mayor capacidad de generalización.

Experiments	Output	Learning type	F1-score	Accuracy	Map
Train Bi-Encoder + msmarco-distilbert-base-v4	Clasification	Supervised	**0.5901513**	0.73076923	0.93160157
Train Bi-Encoder + msmarco-distilbert-base-v4 + multitask	Clasification	Supervised	0.5603403	0.70293522	0.91105189
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA	Clasification	Supervised	0.56146031	0.75910931	0.93666551
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification	Clasification	Supervised	0.57645031	0.72621457	0.92674989
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + OVA + Clasification	Clasification	Supervised	0.56089442	0.76973684	0.93907363
BOW	Coseno	Unsupervised	0.22142636	0.40131579	0.81108447
Pretrained Sentence BERT + msmarco-distilbert-base-v4	Coseno	Unsupervised	0.35946579	0.40840081	0.82955561
Train Bi-Encoder + msmarco-distilbert-base-v4 trained multitask + clase potencial 0.5	Coseno	Supervised	0.52208546	0.63714575	0.91131416
Word2Vec + Composicionalidad + coseno	Coseno	Unsupervised	0.34594084	0.38610827	0.81238354
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA + cosine	Coseno	Supervised	0.55680949	**0.78947368**	0.92161791
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential	Coseno and Clasification	Supervised	0.54789118	0.74848178	0.92345248
Word2Vec + Composicionalidad + ICM	ICM	Unsupervised	0.30608799	0.33912155	0.8049374
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Cross entropy loss	Clasification	Supervised	0.58440478	0.74797571	0.93608346
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + OVA + Cross entropy loss	Clasification	Supervised	0.56236571	0.77327935	**0.9454376**
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential + sentence-Transformers/paraphrase-MiniLM-L6-v2	Coseno and Clasification	Supervised	0.54862902	0.75455466	0.92345248

Tabla 147: Comparación métricas experimentos con 3 clases SemEval 2015.

En la Tabla 148, se pueden ver los experimentos del dataset SemEval 2017 con 3 clases. La escala de color representa que cuanto más verde es una métrica más supera a otras propuestas. Realmente, esta comparación es la menos relevante, ya que en el artículo asociado se hace la inferencia sobre test y se calculan las métricas con 2 clases. Sin embargo, es interesante ver esta comparación para ver cual lidia mejor con la ausencia de dato en la clase potencial a la hora de predecir.

Se puede ver que a la hora de evitar el desbalanceo de clases en test, como en este caso que no hay ningún valor en la clase potencial, la mejor solución es un OVA con un Bi-Encoder para clasificar. El OVA es el que mejor funciona porque tiene que realizar una distinción con respecto a la clase potencial muy grande, debido a que no existe en el dataset de test. Por otra parte, es mejor utilizar la clasificación en lugar de los cosenos para evitar fijar umbrales.

Sin embargo, si lo que nos interesa es hacer el ranking que es lo que se primaba en este dataset, la mejor solución pasa por usar un Cross-Encoder. Esto se debe a que los Cross-Encoder extraen más información que los Bi-Encoders del par de frases. De hecho, después de los Cross-Encoders normales, la mejor solución es usar un OVA con Cross-Encoder, reafirmando que esta arquitectura es la que mejor funciona. Esto combina la mejor gestión del desbalanceo entre clases del OVA junto al Cross-Encoder.

Por último, se pude ver cómo los modelos no supervisados funcionan especialmente mal ya que no saben cuántos ejemplos de cada clase hay y, por tanto, se clasifican muchos como de la clase potencial, la cual no existe. Dentro de los métodos no supervisados, el que mejor funciona es el Bi-Encoder preentrenado.

Experiments	Output	Learning type	F1-score	Accuracy	Map
Train Bi-Encoder + msmarco-distilbert-base-v4	Clasificación	Supervised	0.52580292	0.72147727	0.88513471
Train Bi-Encoder + msmarco-distilbert-base-v4 + multitask	Clasificación	Supervised	0.5140736	0.705	0.86994928
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA	Clasificación	Supervised	0.53361909	0.73954545	0.88820624
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification	Clasificación	Supervised	0.5240266	0.71602273	0.89225532
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + OVA + Clasification	Clasificación	Supervised	0.52844339	0.72238636	0.88808907
BOW	Coseno	Unsupervised	0.21674055	0.44181818	0.67537932
Pretrained Sentence BERT + msmarco-distilbert-base-v4	Coseno	Unsupervised	0.21200266	0.33204545	0.69699591
Train Bi-Encoder + msmarco-distilbert-base-v4 trained multitask + clase potencial 0.5	Coseno	Supervised	0.51460744	0.70352273	0.86872087
Word2Vec + Composicionalidad + coseno	Coseno	Unsupervised	0.30424944	0.38254803	0.68521565
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA + cosine	Coseno	Supervised	**0.53673139**	**0.74818182**	0.87009369
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential	Coseno and Clasificación	Supervised	0.51645048	0.72261364	0.85647433
Word2Vec + Composicionalidad + ICM	ICM	Unsupervised	0.26161763	0.32822507	0.6387611
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Cross entropy loss	Clasificación	Supervised	0.52792646	0.72102273	**0.90137381**
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + OVA + Cross entropy loss	Clasificación	Supervised	0.52734071	0.72045455	0.88493079

Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential + sentence-Transformers/paraphrase-MiniLM-L6-v2	Coseno and Clasificatio n	Supervised	0.5108076 7	0.7046590 9	0.8564743 3

Tabla 148: Comparación métricas experimentos con 3 clases SemEval 2017.

En la Tabla 149, se puede ver la comparación de las métricas de los experimentos con 3 clases en AmazonQA. La escala de color representa que cuanto más verde es una métrica más supera a otras propuestas. Se puede ver que los resultados son mucho peores en todos los experimentos que en los datasets anteriores. Es difícil ver cuál es el experimento que mejor ha funcionado debido a que casi ninguno lo ha hecho. Sin embargo, se pude decir que los que mejor han funcionado han sido los basados en Bi-Encoders entrenados para la clasificación y el Cross-Encoder para la métrica Map. Esto puede ser porque a la hora de clasificar es más importante la capacidad de generalización y a al hora de ordenar resultados prima extraer más relaciones entre pregunta y respuesta. Este es un patrón que se repite en todos los datasets.

En líneas generales, en este dataset es en el que mejor funciona los métodos no supervisados. Esto es debido a que no tiene sobreaprendizaje en la clase mala y no le afecta el desbalanceo de clases. Por otro lado, el Map no sale tan mal, es decir, salen valores que podrían llegar a ser útiles, pero es verdad que están muy cerca del baseline.

Este dataset es muy complicado debido a los fallos de las etiquetas, del desbalanceo entre clases y de su gran tamaño. A pesar de ello, nos permite generalizar las afirmaciones sobre qué experimentos funcionan mejor en los datasets anteriores.

Experiments	Output	Learning type	F1-score	Accuracy	Map
Train Bi-Encoder + msmarco-distilbert-base-v4	Clasification	Supervised	0.3654387	0.58084161	0.7623257
Train Bi-Encoder + msmarco-distilbert-base-v4 + multitask	Clasification	Supervised	**0.36869613**	0.59431383	0.76118747
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA	Clasification	Supervised	0.28420117	**0.68228811**	0.72151429
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification	Clasification	Supervised	0.29442669	0.68240459	**0.79020393**
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + OVA + Clasification	Clasification	Supervised	0.34644306	0.64355757	0.77684889
BOW	Coseno	Unsupervised	0.32332386	0.58931097	0.74385238
Pretrained Sentence BERT + msmarco-distilbert-base-v4	Coseno	Unsupervised	0.33041919	0.47419536	0.76683485
Train Bi-Encoder + msmarco-distilbert-base-v4 trained multitask + clase potencial 0.5	Coseno	Supervised	0.29819163	0.66689684	0.77049342
Word2Vec + Composicionalidad + coseno	Coseno	Unsupervised	0.30331894	0.35372112	0.76520077
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA + cosine	Coseno	Supervised	0.31231007	0.6612201	0.76708923
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential	Coseno and Clasification	Supervised	0.33666383	0.64320538	0.77684889
Word2Vec + Composicionalidad + ICM	ICM	Unsupervised	0.27574016	0.32294261	0.75405059
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Cross entropy loss	Clasification	Supervised	0.27294179	0.68128421	0.78505039
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + OVA + Cross entropy loss	Clasification	Supervised	0.28420117	0.68228811	0.72151429
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential + sentence-Transformers/paraphrase-MiniLM-L6-v2	Coseno and Clasification	Supervised	0.34644306	0.64355757	0.77684889

Tabla 149: Comparación métricas experimentos con 3 clases AmazonQA.

En la Tabla 150, se puede ver la comparación de las métricas de los experimentos con 2 clases en SemEval 2015. La escala de color representa que cuanto más verde es una métrica más supera a otras propuestas. En este caso, se puede ver que el Cross-Encoder funciona mejor en todas las métricas que el Bi-Encoder. Puede ser que, en dos clases, el Cross-Encoder sea capaz de aprender más que el Bi-Encoder. Sin embargo, hay que recordar que el Cross-Encoder siempre necesita un fine-tuning para funcionar, mientras que el Bi-Encoder no. De hecho, el experimento no supervisado con 2 clases demuestra que podría ser incluso funcional utilizarlo sin entrenar para otros dominios.

Experiments	Output	Learning type	F1-score	Accuracy	Map
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification+2 Classes	Clasification	Supervised	0.85113541	0.8208502	0.92738416
Pretrained Sentence BERT + msmarco-distilbert-base-v4 + Coseno	Coseno	Unsupervised	0.76879433	0.67004049	0.82955561
Train Bi-Encoder + msmarco-distilbert-base-v4 + Cosine	Coseno	Supervised	0.80948487	0.76417004	0.89899757
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Borrar clase potencial + Cross entropy loss	Clasification	Supervised	**0.86116867**	**0.83046559**	**0.9413247**

Tabla 150: Comparación métricas experimentos con 2 clases SemEval 2015.

En la Tabla 151, se puede ver la comparación de las métricas de los experimentos con 2 clases en SemEval 2017. La escala de color representa que cuanto más verde es una métrica más supera a otras propuestas. En este caso, ocurre lo mismo que en el dataset del SemEval 2015. El Cross-Encoder funciona mejor que el Bi-Encoder en todas las métricas, lo que implica que aprende mejor la relación entre la pregunta y la respuesta.

En el fondo, el Bi-Encoder tiene el problema que al realizar el coseno se esta estableciendo una relación de cercanía entre embeddings. El problema es que quizás la relación entre una pregunta y una respuesta no se puede crear como una relación de cercanía. A pesar de esto, los Bi-Encoder dan bastante buen resultado.

Experiments	Output	Learning type	F1-score	Accuracy	Map
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification+2 Classes	Clasification	Supervised	**0.85432823**	**0.84204545**	0.88412582
Pretrained Sentence BERT + msmarco-distilbert-base-v4 + Coseno	Coseno	Unsupervised	0.59752149	0.59034091	0.69699115
Train Bi-Encoder + msmarco-distilbert-base-v4 + Cosine	Coseno	Supervised	0.83711466	0.81806818	0.86045987
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Borrar clase potencial + Cross entropy loss	Clasification	Supervised	0.85008855	0.83647727	**0.90151069**

Tabla 151: Comparación métricas experimentos con 2 clases SemEval 2017.

En la Tabla 151, se puede ver la comparación de las métricas de los experimentos con 2 clases en AmazonQA. La escala de color representa que cuanto más verde es una métrica más supera a otras

propuestas. En este caso, curiosamente el modelo que mejor funciona es el no supervisado. Esto indica que, como se ha ido viendo, los modelos sobreaprenden la clase negativa empeorando mucho los resultados.

A pesar de esto, con la métrica Map se ve que van mejor los modelos entrenados. Esto es porque al final la clase negativa es la que mejor se aprende y la mayoritaria. Por tanto, para ver cuál es el modelo que mejor generaliza es mejor ver el F1-score que el Map. La escala de color representa que cuanto más verde es una métrica más supera a otras propuestas.

Experiments	Output	Learning type	F1-score	Accuracy	Map
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification+2 Classes	Clasification	Supervised	0.0608933	0.68513619	0.78661344
Pretrained Sentence BERT + msmarco-distilbert-base-v4 + Coseno	Coseno	Unsupervised	**0.41353315**	0.5202	0.76683485
Train Bi-Encoder + msmarco-distilbert-base-v4 + Cosine	Coseno	Supervised	0.07591576	**0.68531922**	**0.78685507**
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Borrar clase potencial + Cross entropy loss	Clasification	Supervised	0.35845482	0.59908096	0.76003994

Tabla 152: Comparación métricas experimentos con 2 clases AmazonQA.

En la Tabla 153, se pueden ver los resultados de este trabajo en el ranking de soluciones del SemEval 2015. En este caso, además de que cuanto mejor es una métrica más verde es, se pueden ver 4 colores en las propuestas:

- El verde oscuro es la mejor solución de todas.
- El verde claro son las soluciones que han mejorado a aquellas del congreso pero que no son las mejores.
- En naranja se encuentran las propuestas peores que alguna solución propuesta en el congreso.
- Las propuestas sin fondo son aquellas que pertenecen al congreso.

Solo se han incluido los experimentos con 3 clases, ya que este ranking es de experimentos con clase buena, potencial y mala. Se puede ver como 3 de las soluciones expuestas en este trabajo mejoran a todas ellas.

Experiments	F1-score	Accuracy
Train Bi-Encoder + msmarco-distilbert-base-v4	**0.590151303**	0.730769231
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Cross entropy loss	0.584404782	0.747975709
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification	0.576450306	0.726214575
JAIST-primary	0.5719	0.7252
HITSZ-ICRC-primary	0.5641	0.6867

Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + OVA + Cross entropy loss	0.562365712	0.773279352
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA	0.561460314	0.759109312
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + OVA + Clasification	0.560894416	**0.769736842**
Train Bi-Encoder + msmarco-distilbert-base-v4 + multitask	0.560340298	0.702935223
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA + cosine	0.556809491	0.789473684
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential + sentence-Transformers/paraphrase-MiniLM-L6-v2	0.548629016	0.754554656
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential	0.547891185	0.748481781
QCRI-primary	0.5374	0.705
ECNU-primary	0.5347	0.7055
Train Bi-Encoder + msmarco-distilbert-base-v4 trained multitask + clase potencial 0.5	0.522085455	0.637145749
ICRC-HIT-primary	0.496	0.6786
VectorSLU-primary	0.491	0.6645
Shiraz-primary	0.4734	0.5683
FBK-HLT-primary	0.4732	0.6913
Voltron-primary	0.4607	0.6235
CICBUAPnlp-primary	0.404	0.5374
Pretrained Sentence BERT + msmarco-distilbert-base-v4	0.359465793	0.40840081
Word2Vec + Composicionalidad + coseno	0.345940842	0.386108274
Word2Vec + Composicionalidad + ICM	0.306087993	0.339121553
Baseline All True	0.2236	0.5046
BOW	0.22142636	0.401315789

Tabla 153: Resultados de este trabajo sobre los del ranking de soluciones de SemEval 2015.

En la Tabla 154, se pueden ver los resultados de este trabajo en el ranking de soluciones del SemEval 2017. En las métricas se ve en más verde cuanto mayor sea determinada métrica. En cuanto a los colores de las métricas, se ha seguido la siguiente lógica:

- El color verde es para la mejor propuesta.
- El color rosa es para las propuestas que han predicho 3 clases en lugar de 2.
- El color naranja es para las propuestas que han predicho 2 clases.
- Las propuestas sin fondo pertenecen a las presentadas en el congreso.

Se han incluido tanto los experimentos con 2 como con 3 clases. Esto es porque la métrica interesante en este competición, el Map, es independiente del número de clases que se tenga, ya que es una métrica de ranking. De hecho, es interesante ver como muchas de las soluciones con 3 clases mejoran a los experimentos con 2 clases. También se puede observar en dicha gráfica cómo 7 experimentos mejoran al mejor experimento del artículo del SemEval 2017.

Experiments	F1-score	Accuracy	Map
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Borrar clase potencial + Cross entropy loss	0.850088551	0.836477273	**0.901510693**
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + Cross entropy loss	0.527926465	0.721022727	0.901373812
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification	0.5240266	0.716022727	0.892255322
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA	0.533619088	0.739545455	0.888206238
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + OVA + Clasification	0.528443391	0.722386364	0.888089068
Train Bi-Encoder + msmarco-distilbert-base-v4	0.525802918	0.721477273	0.885134713
Train Cross-Encoder + sentence-Transformers/paraphrase-MiniLM-L6-v2 + OVA + Cross entropy loss	0.52734071	0.720454545	0.88493079
KeLP-primary*	0.6987	0.7389	0.8843
Train Cross-Encoder + sentence-Transformers/stsb-distilbert-base + Clasification+2 Classes	**0.854328233**	**0.842045455**	0.884125817
Beihang-MSRA-primary*	0.684	0.5198	0.8824
Train Bi-Encoder + msmarco-distilbert-base-v4 + OVA + cosine	0.536731391	0.748181818	0.870093693
Train Bi-Encoder + msmarco-distilbert-base-v4 + multitask	0.514073604	0.705	0.869949277
IIT-UHH-primary*	0.7394	0.727	0.8688
Train Bi-Encoder + msmarco-distilbert-base-v4 trained multitask + clase potencial 0.5	0.514607435	0.703522727	0.868720867
ECNU-primary*	0.7767	0.7843	0.8672
bunji-primary*	0.725	0.7498	0.8658
EICA-primary*	0.4501	0.6164	0.8653
SwissAlps-primary	0.433	0.613	0.8624
Train Bi-Encoder + msmarco-distilbert-base-v4 + Cosine	0.837114658	0.818068182	0.860459872
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential + sentence-Transformers/paraphrase-MiniLM-L6-v2	0.510807668	0.704659091	0.856474335
Train Bi-Encoder GOOD&Potential vs BAD + msmarco-distilbert-base-v4 + Cross-Encoder GOOD vs Potential	0.516450483	0.722613636	0.856474335
FuRongWang-primary	0.6204	0.6884	0.8426
FA3L-primary*	0.6596	0.6802	0.8342
SnowMan-primary	0.6737	0.7058	0.8184
Pretrained Sentence BERT + msmarco-distilbert-base-v4	0.212002665	0.332045455	0.696995915
Pretrained Sentence BERT + msmarco-distilbert-base-v4 + Coseno	0.597521492	0.590340909	0.696991147
Word2Vec + Composicionalidad + coseno	0.317930415	0.382548033	0.685215647
BOW	0.216740548	0.441818182	0.675379321
Word2Vec + Composicionalidad + ICM	0.261617632	0.328225069	0.638761101
Baseline Random	0.6254	0.527	0.623
Baseline All True	0.684	0.5198	0

Tabla 154: Resultados de este trabajo sobre los del ranking de soluciones de SemEval 2017.

Un análisis necesario de realizar es el de los errores más típicos que se han dado a lo largo de los experimentos. Estos son:

1. En los experimentos con modelos preentrenados o con uso de la cuenta de las palabras se da el error de **falta de palabras en común**. Al final una pregunta no tiene por qué tener palabras en común con una respuesta idónea. Al final el error general es que **falta aprendizaje sobre qué tipo de similitud buscar** y por tanto no clasifica dichos pares de pregunta-respuesta como buenos. La solución a este tipo de problemas es usar modelos contextuales que además estén entrenados para el tipo de tarea especifica que se quiere realizar.
2. También debido a la naturaleza subjetiva de la tarea, se da el caso de **fallo de etiquetado de pregunta y respuesta**. Realmente, esto no es un error al uso, simplemente que la persona que ha etiquetado dichos ejemplos entendió que una respuesta particular no encajaba con su respectiva pregunta. De hecho, de una muestra de 150 pares pregunta-respuesta elegidas aleatoriamente, 37 parecían mal etiquetadas, lo que supone un 24,7%. Esto no tiene una solución simple. La única manera sería que el sistema al llevarse a producción fuera reentrenado cada cierto tiempo en función de la valoración de los usuarios respecto a su uso.
3. El sistema **fracasa también al detectar funciones del lenguaje** menos literales, **como la ironía**. Esto da como resultado que respuestas que se plantean de manera sarcástica y sin aportar ninguna información relevante se asocien como muy relevantes para a la pregunta. Para hacer entender a un modelo este tipo de funciones del lenguaje, la solución pasa por aumentar el número de documentos y el número de parámetros del modelo aprendiendo mucho mejor el lenguaje natural.
4. Por último, estos sistemas **fallan al hilar fino**. Cuando una respuesta tiene apariencia de buena por parecer tratar de temas similares o utilizar lenguaje similar, se clasifica como buena respuesta, aunque realmente no tenga nada que ver. Esto se debe a que el modelo no sabe exactamente cuál es la respuesta esperada, solo el formato que debe tener. Al igual que en el caso anterior, la solución pasa por aumentar el número de ejemplos de entrenamiento y el número de parámetros el modelo.

7. Conclusiones y líneas futuras

En este trabajo se ha estudiado como aplicar modelos neuronales basados en Transformers a la tarea de Búsqueda de Respuestas Comunitaria (en inglés Community Question Answering, CQA). Las propuestas realizadas se han evaluado sobre las colecciones que se utilizaron en el marco del SemEval 2015 y 2017, así como sobre la colección AmazonQA, que no se ha utilizado previamente para evaluar CQA.

Durante el trabajo se ha visto que los modelos propuestos, utilizando únicamente texto, mejoran los resultados de los sistemas propuestas anteriormente, que hacían uso de texto y metadatos. Las dos arquitecturas propuestas que destacan en cuanto a resultados por encima de las demás son los Cross-Encoder y los Bi-Encoders. Ambas utilizan Transformers para codificar la información, aunque lo hacen de manera diferente.

También se han comprobado diferentes arquitecturas como los ensembles OVA (Obe versus All). Estos ensembles se pueden usar junto con los OVO (One Versus One) para tratar de separar mejor las clases complicadas o minoritarias y que el algoritmo aprenda mejor a diferenciarlas. En nuestro ejemplo, se ha visto que en el SemEval 2017 al haber la clase potencial en el entrenamiento y no en el test, si se usa el OVA las métricas evaluando con 3 clases en test funciona mejor.

Por otra parte, para comprobar si la tarea de clasificación es compatible con la de similitud semántica, se ha probado a hacer un multitask con estas dos tareas sobre Bi-Encoders. Lo que se ha podido apreciar es que ambas tareas son compatibles y los resultados son buenos, aunque en general funciona mejor entrenar un modelo para cada tarea por separado.

Respecto a los diferentes datasets, se ha visto que las propuesta realizadas no funciona muy bien sobre el dataset de AmazonQA debido a que las estrellas de cada respuesta dependen más de la coherencia de esta que de su adecuación con la pregunta. Sin embargo, los métodos de Transformers permiten mejorar los resultados del baseline.

También se ha observado que las aproximaciones que trabajan con 2 clases obtienen unos resultados extraordinarios con respecto a las de 3 clases. Esto lleva a pensar que a la hora de realizar esta tarea en un caso de uso real es mejor trabajar con 2 clases cuando no sea imprescindible añadir una clase potencial.

A partir de este trabajo hay ciertas líneas que parecen interesantes de seguir. Por ejemplo, los mejores sistemas del SemEval 2017 mejoran mucho sus resultados al incluir metadatos como el autor de la pregunta y de la respuesta, la fecha de publicación… Toda esta información no se está usando en las aproximaciones de este trabajo y podría ser interesante comprobar cómo se comporta.

Por otro lado, las representaciones composicionales con embeddings han funcionado de manera mediocre. Sin embargo, no se ha comprobado la validez de la función ICM en el caso de medir la similitud semántica entre representaciones extraídas por los Bi-Encoders. Esta función también podría ser aplicada a los embeddings de los Bi-Encoders para ver su similitud semántica. Estaría bien como futuro trabajo aplicarlo en estas arquitecturas y hacer una comparación con la función coseno.

Otro trabajo con relevancia que se deriva de este es ver cuál es la generalización de estos sistemas con trabajos en otro idioma, como el español. Esto es de interés ya que la gran mayoría de datasets y recursos

están en inglés, pero sin embargo existen numerosos foros en español como forococoches o cualquier foro de una universidad.

De hecho, al hilo de esto sería de interés comprobar cómo se comportan estos sistemas entrenados en otro idioma y en otro dominio para luego aplicarlo en un foro como el foro de estudiantes de UNED. Esto podría ser de mucho interés y de mucha utilidad para los estudiantes que igual tienen que buscar una respuesta entre 500.

Bibliografía

[1] P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. Glass, and B. Randeree, "SemEval-2015 task 3: Answer selection in Community Question Answering," *arXiv*, no. SemEval, pp. 269–281, 2015, doi: 10.18653/v1/s15-2047.

[2] M. Wang, N. A. Smith, and T. Mitamura, "What is the Jeopardy model? A quasi-synchronous grammar for QA," *EMNLP-CoNLL 2007 - Proc. 2007 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn.*, no. June, pp. 22–32, 2007.

[3] M. Heilman and N. A. Smith, "Tree edit models for recognizing textual entailments, paraphrases, and answers to questions," *NAACL HLT 2010 - Hum. Lang. Technol. 2010 Annu. Conf. North Am. Chapter Assoc. Comput. Linguist. Proc. Main Conf.*, no. June, pp. 1011–1019, 2010.

[4] P. Nakov *et al.*, "SemEval-2017 Task 3: Community Question Answering," *arXiv*, 2017, doi: 10.18653/v1/s17-2003.

[5] P. Nakov *et al.*, "SemEval-2016 Task 3: Community Question Answering," *arXiv*, 2016, doi: 10.18653/v1/s17-2003.

[6] M. N. Rusell, P. G. Jurafsky, Forsyth, and Norvig, *Speech and language processing*. 2009.

[7] E. M. Voorhees, "The TREC Question Answering track," *Nat. Lang. Eng.*, vol. 7, no. 4, pp. 361–378, 2001, doi: 10.1017/S1351324901002789.

[8] A. Allam, A. Mohamed, N. Allam, and M. H. Haggag, "The Question Answering Systems : A Survey," no. October, 2012.

[9] F. M. G. Carmona and M. M. Aldon, "AUTÓMATAS FINITOS, PROCESAMIENTO DE UNIDADES MORFOLÓGICO-LÉXICAS Y ETIQUETADO SINTÁCTICO," *Neoinstrumenta*, 2014.

[10] D. Moldovan, S. Harabagiu, and M. Pasca, "LASSO: A Tool for Surfing the Answer Net," *Proc. Eighth Text Retr. Conf. TREC 1999*, 1999, [Online]. Available: http://trec.nist.gov/pubs/trec8/papers/smu.pdf.

[11] A. M. N. Allam, M. M. Sakre, and M. M. Kouta, "Weighting Query Terms using WordNet Ontology," *Int. J. Comput. Sci. Netw. Secur.*, vol. 9, no. 4, pp. 349–358, 2009, [Online]. Available: http://paper.ijcsns.org/07_book/html/200904/200904047.html.

[12] Y. Hao, X. Liu, J. Wu, and P. Lv, "Exploiting sentence embedding for medical Question Answering," *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019*, pp. 938–945, 2019, doi: 10.1609/aaai.v33i01.3301938.

[13] Y. Zhang and Z. Xu, "BERT for Question Answering on SQuAD 2.0," 2019.

[14] D. Jurafsky and J. H. Martin, "Speech and Language Processing," 2020, doi: 10.4324/9780203461891_chapter_3.

[15] D. Yogish, P. M. T. N, and P. R. S. Hegadi, "A Survey of Intelligent Question Answering System Using NLP and Information Retrieval Techniques," *Int. J.*, vol. 5, no. 5, pp. 536–540, 2016, doi: 10.17148/IJARCCE.2016.55134.

[16] B. Ojokoh and E. Adebisi, "A review of Question Answering systems," *J. Web Eng.*, vol. 17, no. 8,

pp. 717–758, 2019, doi: 10.13052/jwe1540-9589.1785.

[17] A. Mishra and S. K. Jain, “A survey on Question Answering systems with classification,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 28, no. 3, pp. 345–361, 2016, doi: 10.1016/j.jksuci.2014.10.007.

[18] L. Hirschman and R. Gaizauskas, “Natural language Question Answering: The view from here,” *Nat. Lang. Eng.*, vol. 7, no. 4, pp. 275–300, 2001, doi: 10.1017/S1351324901002807.

[19] A. Allam, A. Mohamed, N. Allam, and M. H. Haggag, “The Question Answering Systems : A Survey,” no. October, 2016.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[21] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading wikipedia to answer open-domain questions,” *arXiv*, pp. 1870–1879, 2017.

[22] S. Kratzwald, Bernhard; Eigenmann, Anna; Feuerriegel, “RankQA: Neural Question Answering with Answer Re-Ranking,” 2019, [Online]. Available: https://doi.org/10.3929/ethz-a-010025751.

[23] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, and W. Zhang, “R 3 : Reinforced Ranker-Reader for Open-Domain Question Answering,” 2018.

[24] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional Transformers for language understanding,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[25] B. Patra, “A Survey of Community Question Answering,” *arXiv*, 2017.

[26] J. Bian, E. Agichtein, Y. Liu, and H. Zha, “Finding the right facts in the crowd: Factoid Question Answering over social media,” *Proceeding 17th Int. Conf. World Wide Web 2008, WWW’08*, no. January, pp. 467–476, 2008, doi: 10.1145/1367497.1367561.

[27] M. J. Blooma and J. C. Kurian, “Research issues in Community based Question Answering,” *PACIS 2011 - 15th Pacific Asia Conf. Inf. Syst. Qual. Res. Pacific*, 2011.

[28] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, “Okapi at TREC-3,” 1996.

[29] S. Robertson, “Understanding inverse document frequency: On theoretical arguments for IDF,” *J. Doc.*, vol. 60, no. 5, pp. 503–520, 2004, doi: 10.1108/00220410410560582.

[30] X. Xue, J. Jeon, and W. B. Croft, “Retrieval models for question and answer archives,” *ACM SIGIR 2008 - 31st Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, Proc.*, pp. 475–482, 2008, doi: 10.1145/1390334.1390416.

[31] S. Filice, D. Croce, A. Moschitti, and R. Basili, “KeLP at SemEval-2016 task 3: Learning semantic relations between questions and answers,” *SemEval 2016 - 10th Int. Work. Semant. Eval. Proc.*, pp. 1116–1123, 2016, doi: 10.18653/v1/s16-1172.

[32] D. Croce, A. Moschitti, and R. Basili, “Structured lexical similarity via convolution kernels on dependency trees,” *EMNLP 2011 - Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1034–1046, 2011.

[33] F. Kunneman, T. C. Ferreira, E. Krahmer, and A. Van Den Bosch, “Question similarity in

Community Question Answering: A systematic exploration of preprocessing methods and models," *Int. Conf. Recent Adv. Nat. Lang. Process. RANLP*, vol. 2019-Septe, pp. 593–601, 2019, doi: 10.26615/978-954-452-056-4_070.

[34] Robert B Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning, "STL: A Seasonal-Trend decomposition Procedure Based on Loess," *Journal of Official Statistics*, vol. 6, no. 1. pp. 3–73, 1990, [Online]. Available: http://www.nniiem.ru/file/news/2016/stl-statistical-model.pdf.

[35] D. Charlet and G. Damnati, "SimBow at SemEval-2017 Task 3: Soft-Cosine Semantic Similarity between Questions for Community Question Answering," pp. 315–319, 2018, doi: 10.18653/v1/s17-2051.

[36] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," *arXiv*, pp. 9119–9130, 2020, doi: 10.18653/v1/2020.emnlp-main.733.

[37] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying Deep Learning to answer selection: A study and an open task," *2015 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2015 - Proc.*, pp. 813–820, 2016, doi: 10.1109/ASRU.2015.7404872.

[38] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, "LSTM-based Deep Learning Models for Non-factoid Answer Selection," no. 1, pp. 1–11, 2015, [Online]. Available: http://arxiv.org/abs/1511.04108.

[39] X. Qiu and X. Huang, "Convolutional neural tensor network architecture for community-based Question Answering," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2015-Janua, no. Ijcai, pp. 1305–1311, 2015.

[40] H. Hashemi, M. Aliannejadi, H. Zamani, and W. B. Croft, "ANTIQUE: A non-factoid Question Answering benchmark," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12036 LNCS, pp. 166–173, 2020, doi: 10.1007/978-3-030-45442-5_21.

[41] T. Chowdhury and T. Chakraborty, "CQASUMm: Building references for Community Question Answering summarization corpora," *ACM Int. Conf. Proceeding Ser.*, pp. 18–26, 2019, doi: 10.1145/3297001.3297004.

[42] M. Gupta, N. Kulkarni, R. Chanda, A. Rayasam, and Z. C. Lipton, "Amazonqa: A review-based Question Answering task," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2019-Augus, pp. 4996–5002, 2019, doi: 10.24963/ijcai.2019/694.

[43] A. Abujabal, R. S. Roy, M. Yahya, and G. Weikum, "ComQA: A community-sourced dataset for complex factoid Question Answering with paraphrase clusters," *arXiv*, pp. 307–317, 2018.

[44] D. Hoogeveen, K. M. Verspoor, and T. Baldwin, "CQADupStack: A benchmark data set for community question-answering research," *ACM Int. Conf. Proceeding Ser.*, vol. 08-09-Dec-, no. December, 2015, doi: 10.1145/2838931.2838934.

[45] X. Liu, C. Wang, Y. Leng, and C. X. Zhai, "Linkso: A dataset for learning to retrieve similar question answer pairs on software development forums ∗," *NL4SE 2018 - Proc. 4th ACM SIGSOFT Int. Work. NLP Softw. Eng. Co-located with FSE 2018*, pp. 2–5, 2018, doi: 10.1145/3283812.3283815.

[46] A. Kilgarriff and C. Fellbaum, "WordNet: An Electronic Lexical Database," *Language (Baltim).*, vol. 76, no. 3, p. 706, 2000, doi: 10.2307/417141.

[47] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application

of a wide-coverage multilingual semantic network," *Artif. Intell.*, vol. 193, pp. 217–250, 2012, doi: 10.1016/j.artint.2012.07.001.

[48] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and Application of a Metric on Semantic Nets," *IEEE Trans. Syst. Man Cybern.*, vol. 19, no. 1, pp. 17–30, 1989, doi: 10.1109/21.24528.

[49] Z. Wu and M. Palmer, "Verb semantics and lexical selection," *n Proc. 32nd Annu. Meet. Assoc. Comput. Linguist. Assoc. Comput. Linguist.*, pp. 133–138, 1994, doi: 10.1152/ajplung.1998.274.3.l351.

[50] Y. Li, A. B. Zuhair, and D. McLean, "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 871–882, 2003, doi: 10.1109/ICCCI.2012.6158835.

[51] D. Sánchez, M. Batet, D. Isern, and A. Valls, "Ontology-based semantic similarity: A new feature-based approach," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 7718–7728, 2012, doi: 10.1016/j.eswa.2012.01.082.

[52] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 805–810, 2003.

[53] Y. Jiang, X. Zhang, Y. Tang, and R. Nie, "Feature-based approaches to semantic similarity assessment of concepts using Wikipedia," *Inf. Process. Manag.*, vol. 51, no. 3, pp. 215–234, 2015, doi: 10.1016/j.ipm.2015.01.001.

[54] D. Sánchez and M. Batet, "A semantic similarity method based on information content exploiting multiple ontologies," *Expert Syst. Appl.*, vol. 40, no. 4, pp. 1393–1399, 2013, doi: 10.1016/j.eswa.2012.08.049.

[55] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," vol. 1, 1995, [Online]. Available: http://arxiv.org/abs/cmp-lg/9511007.

[56] D. Lin, "An Information-Theoretic Definition of Similarity," *Icml*, pp. 296–304, 1998.

[57] J. J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," *Proc. Int. Conf. Res. Comput. Linguist.*, 1997, doi: 10.1152/ajplegacy.1959.196.2.457.

[58] J. B. Gao, B. W. Zhang, and X. H. Chen, "A WordNet-based semantic similarity measurement combining edge-counting and information content theory," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 80–88, 2015, doi: 10.1016/j.engappai.2014.11.009.

[59] G. Zhu and C. A. Iglesias, "Computing Semantic Similarity of Concepts in Knowledge Graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 72–85, 2017, doi: 10.1109/TKDE.2016.2610428.

[60] J. Gorman and J. R. Curran, "Scaling distributional similarity to large corpora," *COLING/ACL 2006 - 21st Int. Conf. Comput. Linguist. 44th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, vol. 1, no. July, pp. 361–368, 2006, doi: 10.3115/1220175.1220221.

[61] S. M. Mohammad and G. Hirst, "Distributional Measures of Semantic Distance: A Survey," 2012, [Online]. Available: http://arxiv.org/abs/1203.1858.

[62] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," *Proc. 2014 Conf. Empir. methods Nat. Lang. Process.*, 2014.

[63] O. Levy and Y. Goldberg, “Dependency-Based Word Embeddings,” *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist.*, pp. 351–358, 2014, doi: 10.1097/00006534-198205000-00031.

[64] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[65] T. K. Landauer and S. T. Dumais, “A Solution to Plato’s Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge We thank Karen Lochbanm for valuable help in analysis; George Fumas for early ideas and inspiration; Peter Foltz, Walter Kintsch,” *Psychol. Rev.*, vol. 104, no. 2, pp. 211–240, 1997, [Online]. Available: http://www.indiana.edu/~pcl/rgoldsto/courses/concepts/landauer.pdf%0Ahttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.333.7403&rep=rep1&type=pdf.

[66] K. Lund and C. Burgess, “Producing high-dimensional semantic spaces from lexical co-occurrence,” *Behav. Res. Methods, Instruments, Comput.*, vol. 28, no. 2, pp. 203–208, 1996, doi: 10.3758/BF03204766.

[67] M. A. Sultan, S. Bethard, and T. Sumner, “DLS$@$CU: Sentence Similarity from Word Alignment,” no. SemEval, pp. 241–246, 2015, doi: 10.3115/v1/s14-2039.

[68] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende, “Knowledge-enhanced document embeddings for text classification,” *Knowledge-Based Syst.*, vol. 163, pp. 955–971, 2019, doi: 10.1016/j.knosys.2018.10.026.

[69] R. L. Cilibrasi and P. M. B. Vitányi, “The Google similarity distance,” *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 370–383, 2007, doi: 10.1109/TKDE.2007.48.

[70] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, “A study on similarity and relatedness using distributional and wordnet-based approaches,” *NAACL HLT 2009 - Hum. Lang. Technol. 2009 Annu. Conf. North Am. Chapter Assoc. Comput. Linguist. Proc. Conf.*, no. January, pp. 19–27, 2009, doi: 10.3115/1620754.1620758.

[71] J. Shawe-Taylor and N. Cristianini, “Kernel Methods for Pattern Analysis,” *Kernel Methods Pattern Anal.*, 2004, doi: 10.1017/cbo9780511809682.

[72] N. Cancedda, E. Gaussier, C. Goutte, and J. M. Renders, “Word-Sequence Kernels,” *J. Mach. Learn. Res.*, vol. 3, no. 6, pp. 1059–1082, 2003, doi: 10.1162/153244303322533197.

[73] A. Moschitti, D. Pighin, and R. Basili, “Tree kernels for semantic role labeling,” *Comput. Linguist.*, vol. 34, no. 2, pp. 193–224, 2008, doi: 10.1162/coli.2008.34.2.193.

[74] Y. Le, Z. J. Wang, Z. Quan, J. He, and B. Yao, “ACV-tree: A new method for Sentence similarity modeling,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 4137–4143, 2018, doi: 10.24963/ijcai.2018/575.

[75] Z. Wang, H. Mi, and A. Ittycheriah, “Sentence similarity learning by lexical decomposition and composition,” *COLING 2016 - 26th Int. Conf. Comput. Linguist. Proc. COLING 2016 Tech. Pap.*, no. challenge 2, pp. 1340–1349, 2016.

[76] W. Y. C. Meek, “WIKI QA : A Challenge Dataset for Open-Domain Question Answering,” no. September 2015, pp. 2013–2018, 2018.

[77] N. H. Tien, N. M. Le, Y. Tomohiro, and I. Tatsuya, “Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity,” *arXiv*, 2018.

[78] H. He and J. Lin, “Pairwise word interaction modeling with deep neural networks for semantic similarity measurement,” *2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 - Proc. Conf.*, pp. 937–948, 2016, doi: 10.18653/v1/n16-1108.

[79] I. Lopez-Gazpio, M. Maritxalar, M. Lapata, and E. Agirre, “Word n-gram attention models for sentence similarity and inference,” *Expert Syst. with Appl. X*, p. 100002, 2019, doi: 10.1016/j.eswax.2019.100002.

[80] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, “A decomposable attention model for natural language inference,” *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 2249–2255, 2016, doi: 10.18653/v1/d16-1244.

[81] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[82] Y. Sun *et al.*, “Ernie 2.0: A continual pre-training framework for language understanding,” *arXiv*, 2019, doi: 10.1609/aaai.v34i05.6428.

[83] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” *arXiv*, pp. 1–17, 2019.

[84] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text Transformer,” *arXiv*, vol. 21, pp. 1–67, 2019.

[85] E. Amigo, A. Ariza, V. Fresno and M. A. Marti. “Information-Theoretic Compositional Distributional Semantics,” Computational Linguistics, 2022. (Under revision)

[86] R. L. Briega, “Introducción a Deep Learning,” *Introd. a Deep Learn.*, pp. 1–37, 2017, [Online]. Available: https://relopezbriega.github.io/blog/2017/06/13/introduccion-al-deep-learning/.

[87] BIBING, “El Perceptrón,” pp. 1–11, 2017.

[88] P. Larrañaga, “Tema 8. Redes Neuronales,” pp. 1–19.

[89] J. M. Marín Diazaraque, “Introducción a las redes neuronales aplicadas,” *Man. Data Min.*, pp. 1–31, 2007, [Online]. Available: halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema3dm.pdf.

[90] F. Berzal, “Backpropagation Backpropagation Introducción Introducción.”

[91] M. A. G. Naranjo, “Introducción a Deep Learning Deep Learning,” 2018.

[92] R. Y. Lstm and R. Matuk, “Rosana Matuk (DC-FCEyN-UBA) RNN y LSTM Redes Neuronales Profundas,” 2017.

[93] V. L. Shiv and C. Quirk, “Novel positional encodings to enable tree-based Transformers,” *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, 2019.

[94] G. Ke, D. He, and T.-Y. Liu, “Rethinking Positional Encoding in Language Pre-training,” pp. 1–14, 2020, [Online]. Available: http://arxiv.org/abs/2006.15595.

[95] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller,

faster, cheaper and lighter," pp. 2–6, 2019, [Online]. Available: http://arxiv.org/abs/1910.01108.

[96] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers," 2020, [Online]. Available: http://arxiv.org/abs/2002.10957.

[97] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," pp. 2–6, 2019.

[98] T. Nguyen *et al.*, "MS MARCO: A human generated MAchine reading COmprehension dataset," *CEUR Workshop Proc.*, vol. 1773, no. Nips 2016, pp. 1–11, 2016.

[99] R. G. Lopes, S. Fenu, and T. Starner, "Data-Free Knowledge Distillation for Deep Neural Networks," 2017, [Online]. Available: http://arxiv.org/abs/1710.07535.

[100] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 3982–3992, 2020, doi: 10.18653/v1/d19-1410.

[101] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process.*, pp. 1412–1421, 2015, doi: 10.18653/v1/d15-1166.

[102] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical Multi-Scale Attention for Semantic Segmentation," pp. 1–11, 2020, [Online]. Available: http://arxiv.org/abs/2005.10821.

[103] W. Weng and X. Zhu, "UNet: Convolutional Networks for Biomedical Image Segmentation," *IEEE Access*, vol. 9, pp. 16591–16603, 2015, doi: 10.1109/ACCESS.2021.3053408.