

Vicomtech en eHealth-KD Challenge 2020: Profundo Modelo de extremo a extremo para extracción de entidades y relaciones en Texto médico

Aitor García-Pablos, Naiara Perez, Montse Cuadros and Elena Zotova

Grupo SNLT en Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Mikeletegi Pasealekua 57, Donostia/San-Sebastián, 20009, Spain

Resumen Este

artículo describe la participación del equipo de PNL de Vicomtech en la tarea compartida eHealth-KD 2020 sobre detección y clasificación de entidades y relaciones en textos relacionados con la salud escritos en español. El sistema propuesto consta de una única red neuronal profunda de extremo a extremo con modelos BERT previamente entrenados como núcleo para la representación semántica de los textos de entrada. Hemos experimentado con dos modelos: BERT-Base Multilingual Cased y BETO, un modelo BERT previamente entrenado en texto en español. Nuestro sistema modela todas las variables de salida (entidades y relaciones) al mismo tiempo, modelando todo el problema de forma conjunta. Algunos de los resultados se retroalimentan a las últimas capas del modelo, conectando los resultados de las diferentes subtarefas en forma de canalización. Nuestro sistema muestra resultados robustos en todos los escenarios de la tarea. Ha conseguido la primera posición en el escenario principal de la competición y el top-3 en el resto de escenarios.

Palabras clave

Detección de entidades, extracción de relaciones, documentos de salud.

1. Introducción

Este artículo describe la participación de Vicomtech en el desafío eHealth Knowledge Discovery (eHealth-KD) 2020 (<https://knowledge-learning.github.io/ehealthkd-2020>). El desafío pasa por el reconocimiento y clasificación de entidades y extracción de relaciones en documentos sanitarios españoles, como se muestra en la Figura 1.

Los organizadores de la tarea proporcionaron un corpus de 800 frases con fines de formación y 200 frases con fines de desarrollo. Las oraciones se anotaron manualmente con un total de 4 tipos de entidades y 13 tipos de relaciones. Como muestra la Figura 1, las entidades constan de uno o más tokens, que pueden ser contiguos o no. Además, las entidades pueden superponerse entre sí, y pueden ser origen y/o destino de más de una relación. Todos estos matices impiden que los enfoques tradicionales de etiquetado de secuencias, basados en el etiquetado BIO o esquemas similares, capturen todas las entidades objetivo.

El desafío consta de cuatro escenarios de evaluación: en el escenario principal, se prueban los sistemas tanto para el reconocimiento como para la clasificación y extracción de relaciones de entidades; en la Tarea A, solo se evalúan entidades, mientras que en la Tarea B, se proporcionan entidades doradas y solo se evalúan relaciones; Finalmente, el escenario Transfer evalúa pipelines completos en documentos de un dominio diferente a los datos de capacitación y desarrollo. Se proporciona más información sobre eHealth-KD 2020 en el artículo de descripción general del desafío [1].

Actas del Foro Ibérico de Evaluación de Lenguas (IberLEF 2020) correo electrónico:

agarcia@vicomtech.org (A. García-Pablos); nperez@vicomtech.org (N. Pérez); mframes@vicomtech.org (METRO.

cuadrícula); ezotova@vicomtech.org (E. Zotova) orcid:

0000-0001-9882-7521 (A. García-Pablos); 0000-0001-8648-0428 (N. Pérez); 0000-0002-3620-1053 (M. Cuadros); 0000-0002-8350-1331 (E. Zotova)



© 2020 Copyright de este artículo por parte de sus autores.
Uso permitido bajo la Licencia Creative Commons Attribution 4.0 International (CC BY 4.0).

1613-0073 Actas del taller CEUR (CEUR-WS.org)

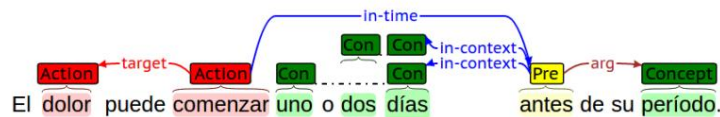


Figura 1: Ejemplo de anotaciones de eHealth-KD en la frase "El dolor puede comenzar uno o dos días antes de su período".

Ante el éxito generalizado de las arquitecturas basadas en Transformer [2] en prácticamente todas las tareas de Procesamiento del Lenguaje Natural (NLP), Vicomtech ha implementado un sistema con BERT [3] que aprende a reconocer y clasificar entidades y a establecer relaciones entre ellas de forma Moda multitarea de extremo a extremo. Nuestro sistema ha conseguido los mejores resultados en el escenario Principal y se sitúa entre los 3 primeros resultados en el resto de escenarios.

El artículo está organizado de la siguiente manera: la Sección 2 describe el modelo propuesto y el enfoque seguido para representar los datos para resolver la tarea; La sección 3 presenta los resultados obtenidos, incluyendo una comparación con otros sistemas competidores; Finalmente, las Secciones 4 y 5 comentan varias opciones de diseño y brindan algunas observaciones finales.

2. Descripción del sistema

Esta sección proporciona una descripción completa del sistema con el que se han obtenido los resultados reportados. Primero, presentamos la arquitectura de la red neuronal profunda. A continuación, explicamos cómo se han representado y manejado las entradas y salidas para resolver la tarea. Luego, describimos las reglas de posprocesamiento que ayudan a corregir posibles resultados incongruentes del modelo de red neuronal. Finalmente, presentamos los escenarios de entrenamiento.

2.1. Arquitectura

El modelo es una red neuronal profunda que recibe los tokens de entrada y emite conjuntamente predicciones para varias variables de salida diferentes. Estas predicciones se pueden agrupar en dos tareas: a) clasificar tokens individuales y b) clasificar relaciones (la presencia, ausencia o tipo de relación) entre pares de tokens. Las variables de salida que predecirá el modelo son las siguientes:

- Entidades: la clasificación de cada token individual en uno de los tipos de entidad de la tarea u 'O' (de 'Out', lo que significa que el token no es parte de ninguna entidad en absoluto, como "puede" en la Figura 1) .
- relaciones multipalabra : si los pares de tokens pertenecen a la misma entidad (como "uno" y "días" o "dos" y "días" en la Figura 1).
- Relaciones iguales : si los pares de tokens están relacionados por la relación iguales .
- Relaciones dirigidas: si los pares de tokens están relacionados por cualquiera de los otros tipos de relación descritos. en la tarea.

A diferencia del resto de relaciones consideradas, las relaciones multipalabra y iguales son bidireccionales.

En vista de varios experimentos preliminares, que indicaron que modelar todas las relaciones juntas causaba ruido al predecir relaciones dirigidas, decidimos modelar relaciones bidireccionales por separado.

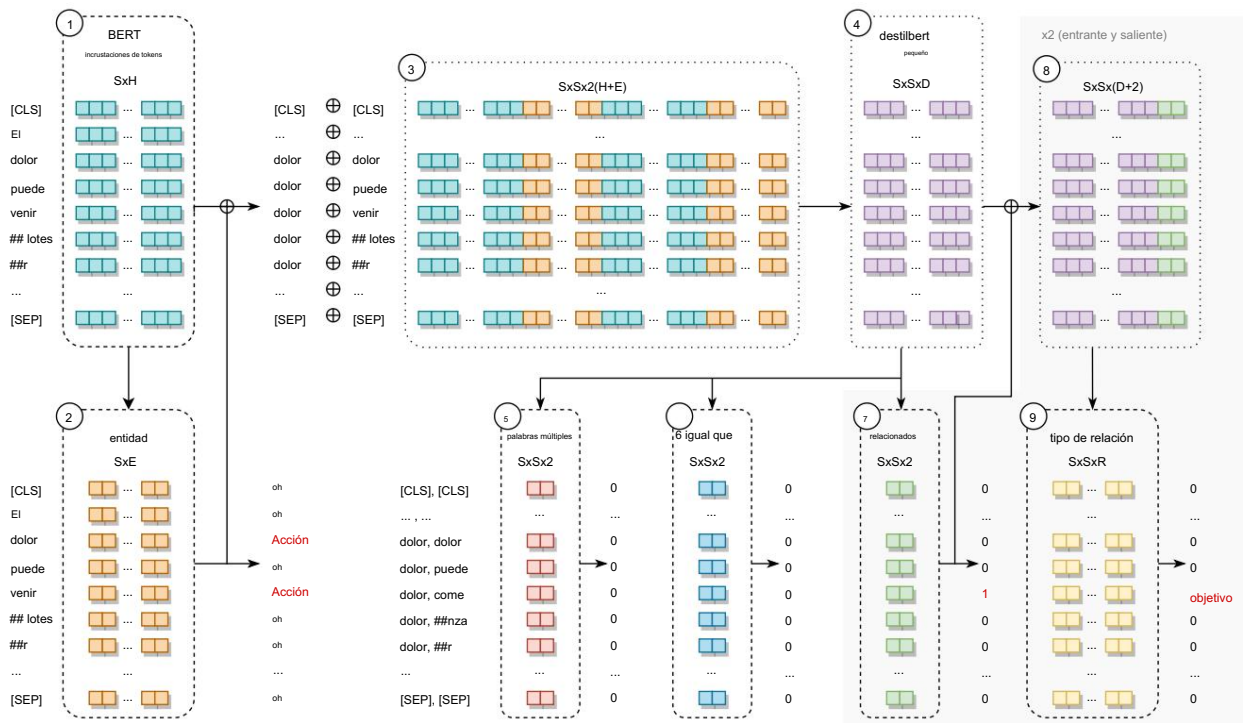


Figura 2: Diagrama de alto nivel del modelo, incluidas las formas generadas por cada capa: $H = 768$ (tamaño de incrustación contextual BERT); $E=5$ (número de tipos de entidades diferentes más 'O'); $D=768$ (incrustación contextual del modelo DistilBERT personalizado), $R=13$ (número de tipos de relaciones diferentes más 'O' y menos la relación igual que se modela por separado). [CLS] y [SEP] son tokens especiales utilizados por BERT.

En la Figura 2 se ofrece una descripción general del funcionamiento interno de la red. El cálculo del modelo comienza con los tokens de entrada. Los tokens se introducen en un modelo BERT para obtener sus incorporaciones contextuales 1. Estas incrustaciones se pasan a una capa de clasificación que emite logits con la predicción de si cada token es o no una entidad de un determinado tipo. 2 A

continuación, los logits de entidad se concatenan nuevamente a las incrustaciones contextuales y se realiza una operación tensor para obtener una Combinación contra todos de vectores simbólicos 3. Esto genera \times incrustaciones combinadas que representan todos los pares de tokens posibles, siendo la longitud de la secuencia de entrada. Además, estas incorporaciones se pasan a un pequeño modelo DistilBERT [4] inicializado aleatoriamente con sólo dos capas de dos cabezas de atención cada una 4. El objetivo de este modelo es capturar aún más las interacciones entre los pares de tokens a través de la autoatención.

Las representaciones de pares de tokens resultantes luego se pasan a varias capas de clasificación para hacer predicciones sobre la relación entre los tokens de cada par. Los pares se clasifican mediante cuatro clasificadores binarios que deciden respectivamente si los tokens que forman el par están conectados por una relación de varias palabras 5 una relación igual a 6, o una de las relaciones dirigidas 7.

Las relaciones dirigidas se modelan como arcos salientes y arcos entrantes. Es decir, si el par está vinculado por una relación de tipo, el par tendrá la misma relación. Esto se representa en la red neuronal como una bifurcación de dos ramas iguales de capas, una para modelar las relaciones salientes y otra para modelar las relaciones entrantes. Observe que la Figura 2 se ha simplificado para mostrar solo una de las ramas.

En este paso, la salida de las capas de clasificación indica si existe o no una relación entre un par de tokens. Las relaciones de varias palabras y del mismo nombre no requieren procesamiento adicional. En el

	0	1	2	3	4	5	6	7	8
simbólico	El	dolor	puede	comenzar	uno	oh	del	días	antes
entidad	oh	Acción	oh	Acción	Concepto	oh	Concepto	Concepto	Predicado
multi palabra	-	-	-	-	7	-	7	4,6	-
igual que	-	-	-	-	-	-	-	-	-
relacionado	-	-	-	1,8	-	-	-	-	4,6
relación tipo	-	-	-	objetivo, a tiempo	-	-	-	-	en contexto, en contexto

Figura 3: Ejemplo de representación de datos de entidades y sus relaciones.

Sin embargo, en el caso de relaciones dirigidas, se debe asignar un tipo de relación o etiqueta. Para ello, los logits de los clasificadores de relación dirigida se concatenan nuevamente con la capa 8 de DistilBERT y el resultado representado pasado a una capa de clasificación final para obtener el tipo de relación para cada par de tokens entre los tipos definidos en la tarea 9. De nuevo, esto se hace dos veces: una para el saliente y otra para el para los arcos entrantes.

En general, la red tiene siete clasificadores, que se construyen utilizando la misma pila de capas: una capa de transformación lineal completamente conectada, seguida de una capa de abandono y una función de activación no lineal. y una transformación lineal final que genera los logits para la variable de salida dada. Nosotros arbitrariamente decidí utilizar Mish [5], sin haber experimentado con otras funciones de activación.

2.2. Manejo de entradas y salidas

Los corpus de formación y desarrollo se han proporcionado en formato standoff Brat [6]. este formato se basa en caracteres, mientras que nuestra red funciona a nivel de token. Además, la salida de la red neuronal debe volver a convertirse al formato de Brat y al esquema de anotación propuesto en la tarea (es decir, las entidades de varias palabras deben reconstruirse a partir de las relaciones de varias palabras, etc.). En consecuencia, nuestro sistema se basa en un conjunto de pasos de transformación previos y posteriores al procesamiento, explica a continuación.

2.2.1. Representación de datos

A partir de la representación de Brat proporcionada, se deben adaptar los diferentes datos de forma manejable según nuestros objetivos. La figura 3 muestra un ejemplo de la información representada diseñada con todas las variables de salida de la red.

Como se mencionó en apartados anteriores, las entidades a detectar no necesariamente son continuas y incluso pueden superponerse. Por ejemplo, el intervalo de texto “uno o dos días” contiene dos entidades independientes: “un día” y “dos días”. Para representar esta información, asignamos a cada token individual su etiqueta de entidad correspondiente según los intervalos de las anotaciones Brat. Las fichas que pertenecen al mismo intervalo de entidad se marcan como vinculados por una relación de varias palabras. Este enfoque permite representar tokens como parte de una o más entidades independientemente de su posición original en el texto. Todo los tokens que forman parte de la misma entidad están interconectados mediante relaciones de varias palabras entre ellos.

Se sigue un enfoque similar para las relaciones iguales y las relaciones dirigidas. Cuando las entidades que abarcan varios tokens están conectados por una de estas relaciones, solo el primer token de cada entidad (en el

orden en que aparecen en el texto) se marca como parte de la relación. Además, las relaciones dirigidas se describen mediante una variable de salida adicional para indicar el tipo de relación entre los diferentes tipos de relación definidos en la tarea.

El modelo descrito realiza todas las tareas de un extremo a otro, utilizando sus propias predicciones de entidades como entrada para detectar relaciones. Sin embargo, en la Tarea B, los organizadores de la tarea proporcionan las anotaciones de entidades doradas; Los sistemas deben centrarse únicamente en las relaciones. En este caso, nuestro modelo acepta etiquetas de entidad doradas junto con los tokens de entrada, y reemplaza las entidades predichas con una codificación one-hot de las doradas como entrada para detectar relaciones.

2.2.2. Interpretar y reconstruir el resultado del modelo.

El resultado del modelo debe interpretarse para obtener una etiqueta o relación correcta y significativa. arco para cada uno de los tokens en el texto de entrada original. La salida del clasificador de entidades se interpreta directamente como una tarea regular de etiquetado de secuencias que selecciona la predicción más probable para cada ficha individual.

La reconstrucción de las relaciones es más elaborada. El resultado de la red para cada relación modelada La variable de ción forma una matriz \times , siendo la longitud de la secuencia del token, donde cada posición y ; $[0,]$ contiene la predicción de la relación entre y . Implementamos Dos estrategias para seleccionar la etiqueta más probable:

- Con el inferenciador 1, sólo se utilizan las relaciones salientes predichas, ignorando las relaciones de los arcos entrantes. predicciones En el caso ideal, deberían ser simétricos y , por tanto, redundantes.
- Con el inferenciador 2, los valores de predicción para cualquier arco saliente \rightarrow y su contraparte entrante arco \leftarrow se resumen antes de seleccionar el resultado más probable. Esto se hace para todos los relaciones modeladas. En el caso de los tipos de relación, esto sólo es posible porque la relación El vocabulario de tipos se comparte entre las relaciones salientes y entrantes, por lo que el mismo índice se refiere al mismo tipo de relación decodificada.

Finalmente, las posiciones simbólicas deben corregirse para tener en cuenta las desviaciones y compensaciones adicionales introducidas por Tokenización de BERT (BERT utiliza la tokenización de WordPiece [7], que divide los tokens originales en subtokens; además, requiere que se agreguen tokens especiales adicionales que distorsionan las posiciones de los tokens escribir la entrada original).

2.2.3. Reglas de posprocesamiento

Las predicciones obtenidas al interpretar el resultado del modelo todavía se basan en tokens y deben procesarse más para obtener una representación en formato Brat standoff que cumpla con eHealth-KD. esquema de anotación. Al hacerlo, también aplicamos varias reglas que corrigen posibles inconsistencias. producido por los diversos clasificadores de la red neuronal. El postprocesamiento consiste en lo siguiente pasos:

1. Alinear las anotaciones de entidades de tokens individuales con el texto original con una herramienta proporcionada por Brat desarrolladores (<https://github.com/nlplab/brat/blob/master/tools/annalign.py>).
2. Fusionar las anotaciones conectadas por un arco de varias palabras , generando efectivamente entidades de varias palabras. Decidimos mantener conjuntos de tokens separados: si token y ambos están conectados al token pero no entre sí, generamos las entidades de varias palabras y . Nosotros , , en lugar de , , También decidió ignorar los arcos de varias palabras hacia/desde tokens que no están clasificados como entidades.
3. Reasigne a esta última las relaciones iguales y dirigidas de los tokens en una entidad de varias palabras.

4. Dividir en dos entidades de varias palabras que contienen conjunciones o ciertos signos de puntuación, como comas, punto y coma, paréntesis, etc.
5. Descarte las entidades de varias palabras que comiencen o terminen con una palabra irrelevante.
6. Si a los tokens de una entidad multipalabra se les han asignado diferentes tipos de entidad, asigne a la entidad multipalabra el tipo más frecuente entre los tokens que componen la multipalabra; en caso de empate, elija la etiqueta más frecuente en el corpus (es decir, Concepto).
7. Descartar entidades que estén totalmente contenidas dentro de otra entidad.
8. Descartar las relaciones iguales y dirigidas desde/hacia un token que no sea una entidad o parte de una entidad.
9. Descartar las relaciones reflexivas, que podrían haber surgido durante la generación de palabras múltiples. notaciones.

2.3. Configuración de entrenamiento

El sistema ha sido implementado en Python 3.7 con la biblioteca de transformadores de HuggingFace [8] (<https://github.com/huggingface/transformers>). Hemos experimentado con dos modelos BERT diferentes previamente entrenados como núcleo para la representación semántica de los tokens de entrada: BERT-Base Multilingual Cased (en adelante, mBERT; <https://github.com/google-research/bert/blob/master/multilingual.md>) y BETO [9], un modelo BERT previamente entrenado en texto en español. No realizamos ningún ajuste del modelo de lenguaje en el dominio para los modelos base. En este sentido, el enfoque es general e independiente del dominio. El único recurso utilizado para ajustar todo el sistema son los datos proporcionados para la tarea, que constan de 800 frases de entrenamiento y 200 frases de desarrollo.

El entrenamiento de las diferentes variantes se realizó en 2 GPU Nvidia GeForce RTX 2080 con 11 GB de memoria. El modelo requiere una cantidad considerable de memoria para el entrenamiento, por lo que el tamaño del lote se ajustó a 2, mientras que la longitud de la secuencia se ajustó a aproximadamente 100 tokens (la longitud máxima encontrada en el conjunto de entrenamiento después de la tokenización de BERT Wordpiece, que varía de mBERT a BETO).). Aplicamos el optimizador AdamW [10] con una tasa de aprendizaje base de 2×10^{-5} , combinado con una programación LR lineal para calentar la tasa de aprendizaje durante los primeros 5000 pasos de entrenamiento. La probabilidad de abandono se estableció arbitrariamente en 0,2 en toda la red.

El entrenamiento monitoreó la puntuación F1 de varios de los clasificadores en el conjunto de desarrollo y se ejecutó durante un máximo de 500 épocas con una paciencia de detención temprana de 150 épocas. Finalmente, elegimos los puntos de control del modelo que tenían el mejor equilibrio de métricas de desarrollo, que para BETO estaba en la época 148 y para mBERT estaba en la época 262.

3. Resultados

Vicomtech participó en todos los escenarios, presentando las mismas tres ejecuciones:

- Ejecución 1: mBERT + inferenciador 1
- Ejecución 2: BETO + inferenciador 1
- Ejecución 3: BETO + inferenciador 2

Los resultados para cada escenario y ejecución se muestran en la Tabla 1. Proporcionamos los resultados de los datos de desarrollo y los resultados publicados oficialmente de los datos de entrenamiento. Además, también se incluyen por escenario para el benchmarking los mejores resultados obtenidos entre todos los participantes en el reto. propósitos.

tabla 1

Resultados de las ejecuciones enviadas y el mejor sistema en cada escenario.

	Desarrollo			Pruebas		
	PAG	R	F1	PAG	R	F1
Escenario 1 - Principal						
Ejecución 1: mBERT + inf 1	66,78	65,23	66,00	67,74	69,95	73,98
Ejecución 2: BETO + inf 1	66,26	65,09	65,67	67,74	69,95	73,98
Ejecución 3: BETO + inf 2 (mejor)	74,39	70,81	72,55	67,94	65,23	66,56
Escenario 2: Tarea A (reconocimiento y clasificación de entidades)						
Ejecución 1: mBERT + inf 1	84,69	85,40	85,04	82,16	82,01	82,09
Ejecución 2: BETO + inf 1	86,12	86,78	86,45	81,95	81,65	81,80
Ejecución 3: BETO + inf 2	85,97	87,09	86,52	82,16	82,01	82,09
SINAI [11] (mejor)	87,06	87,39	87,23	84,46	80,67	82,52
Escenario 3: Tarea B (extracción de relaciones)						
Ejecución 1: mBERT + inf 1	67,78	54,15	60,20	64,86	50,77	56,96
Ejecución 2: BETO + inf 1	69,50	59,05	63,85	65,61	51,73	57,85
Ejecución 3: BETO + inf 2	71,54	58,47	64,35	67,17	51,54	58,32
IXA-NER-RE [12] (mejor)	70,67	67,44	69,02	64,79	61,92	63,32
Escenario 4 - Transferencia						
Ejecución 1: mBERT + inf 1	-	-	-	58,45	52,18	55,14
Ejecución 2: BETO + inf 1	-	-	-	57,94	53,05	55,39
Ejecución 3: BETO + inf 2	-	-	-	59,40	53,55	56,33
Talp-UPC [13] (mejor)	-	-	-	60,47	56,38	58,35
Escenario 4: Transferencia, no oficial						
BETO + inf 2 + dominio FT	-	-	-	68,42	54,47	60,65

Además, después de enviar nuestras predicciones, supimos que los organizadores de la tarea habían proporcionado 100 sentencias fuera de dominio para afinar los sistemas para el escenario de Transferencia. Nuestros resultados oficiales para Este escenario no implicó ningún tipo de ajuste para el nuevo dominio y, por lo tanto, depende exclusivamente sobre el aprendizaje por transferencia de tiro cero. En aras de la integridad, hemos ajustado nuestro mejor rendimiento modele con las oraciones adicionales proporcionadas e informe los resultados en la parte inferior de la tabla de resultados.

Como muestra la Tabla 1, nuestro enfoque ha logrado las mejores puntuaciones del desafío en el escenario principal. obtener el mejor equilibrio entre la Tarea A (reconocimiento y clasificación de entidades) y la Tarea B (relación extracción—, a pesar de ser superados por otros participantes en las tareas individuales. La propuesta enfoque produce una mejor precisión y métricas de recuperación, mejorando el segundo mejor sistema en más más de 2 puntos de puntuación F1. En la Tarea A, nuestro sistema queda en segunda posición, habiendo mejorado la recuperación de el sistema ganador (82,01 vs 80,67) pero no su precisión (82,16 vs 84,46). En cuanto a la Tarea B, nuestro enfoque produce puntuaciones de recuperación notablemente más bajas que el mejor sistema (51,73 frente a 61,92), pero logra ganar el tercer lugar lugar en el escenario con mayor precisión (67,17). Finalmente, nuestro sistema ha conseguido el tercer puesto en el Tarea de transferencia, a pesar de no haberse afinado con los datos disponibles para el nuevo dominio. Nuestro El sistema habría logrado la mejor puntuación de F1 si se hubieran utilizado las 100 frases adicionales, ya que la última fila en la Tabla 1 se muestra.

En cuanto a las diferencias entre las ejecuciones presentadas, se observa poca diferencia. BETO parece

ser una opción ligeramente mejor que mBERT en todos los escenarios. Sin embargo, la variación de estas diferencias en el desarrollo y el conjunto de pruebas sugiere que las diferencias observadas pueden no ser estadísticamente significativas, especialmente debido al tamaño limitado de los conjuntos de datos. En cuanto a las diferencias entre el inferenciador 1 y 2, este último parece ayudar a mejorar las puntuaciones para la detección de relaciones. Específicamente, tener en cuenta los arcos entrantes y salientes parece ayudar a producir predicciones más precisas al descartar en su mayoría predicciones falsas positivas en comparación con el inferenciador 1.

4. Discusión

Debido a limitaciones de tiempo, el modelo presentado es el resultado de muchas elecciones de diseño arbitrarias y contiene componentes discutibles que pueden requerir más investigación y experimentación. Para enumerar algunos de los posibles defectos:

- El modelo DistilBERT personalizado intermedio es una adición basada en la intuición. No hemos realizado suficientes experimentos para demostrar su utilidad. Además, implica una cantidad no despreciable de requisitos computacionales y de memoria adicionales.
- No está claro si modelar las relaciones como arcos salientes y entrantes ayuda a mejorar los resultados. No hemos experimentado con otras variaciones de representación para reunir evidencia suficiente para llegar a una conclusión al respecto.
- Las relaciones dirigidas se detectan en dos pasos: 1) si existe o no una relación, y 2) el tipo de relación. Esto se puede hacer en un solo paso. No hemos realizado experimentos para saber qué enfoque produce mejores resultados.

Con todo, nuestro sistema parece ser un buen reconocedor de entidades con la capacidad de producir relaciones bastante precisas entre las entidades (aunque omite casi la mitad de ellas) y ser adecuado para escenarios de aprendizaje por transferencia. El modelado conjunto de ambas entidades y relaciones ha permitido al sistema lograr un buen equilibrio entre la Tarea A y B, pero el sistema no destaca en ninguna de ellas individualmente. La presencia de un modelo BERT previamente entrenado ayuda en el escenario de transferencia de dominio. Dado que los resultados obtenidos sugieren que una formación previa específica sobre textos en español (es decir, BETO) logra mejores puntuaciones, una formación previa adicional sobre datos más relevantes probablemente ayudaría a mejorar los resultados.

En general, la tarea está lejos de estar resuelta, en particular en lo que respecta a la extracción de relaciones, a pesar de los resultados razonablemente buenos obtenidos. Dejamos los temas y cuestiones abiertas discutidas para trabajos futuros.

5. Conclusiones

En estas notas de trabajo hemos descrito nuestra participación en la tarea compartida eHealth-KD 2020. Hemos presentado la arquitectura de nuestro sistema basada en el aprendizaje profundo de un extremo a otro, que se basa en modelos BERT previamente entrenados como base para la representación semántica de los textos y modela conjuntamente las entidades y relaciones propuestas en la competencia. Hemos descrito nuestra representación de datos, que permite modelar entidades discontinuas y superpuestas de manera integrada. También explicamos cómo interpretamos y posprocesamos la salida de la red neuronal. El sistema propuesto ha ganado la competición, logrando el primer puesto en el escenario Principal y ubicándose entre los 3 primeros en los otros tres escenarios. Aún así, se requiere más experimentación para comprender el impacto de los componentes de la red y cómo mejorarlos, lo que exploraremos en trabajos futuros.

Expresiones de gratitud

Este trabajo ha contado con el apoyo de Vicomtech y parcialmente financiado por el proyecto DeepReading (RTI2018-096846-B-C21, MCIU/AEI/FEDER,UE).

Referencias

- [1] Piad-Morffis, Y. Gutierrez, H. Cañizares-Díaz, S. Estevez-Velarde, Y. Almeida-Cruz, R. Muñoz y A. Montoyo, Descripción general del desafío de descubrimiento de conocimientos sobre eSalud en IberLEF 2020, en : Actas del Foro Ibérico de Evaluación de Lenguas compartido con la 36ª Conferencia de la Sociedad Española para el Procesamiento del Lenguaje Natural, IberLEF@SEPLN 2020,
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, AN Gomez, Ł. Kaiser, I. Polosukhin, La atención es todo lo que necesita, en: Actas de la trigésima primera conferencia sobre avances en sistemas de procesamiento de información neuronal (NeurIPS 2017), 2017, págs.
- [3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Formación previa de transformadores bidireccionales profundos para la comprensión del lenguaje, en: Actas de la Conferencia de 2019 del Capítulo Norteamericano de la Asociación de Lingüística Computacional: Tecnologías del lenguaje humano, Volumen 1 (artículos largos y breves), 2019, págs. 4171–4186.
- [4] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, una versión destilada de BERT: más pequeña, más rápida, más barata y más ligera, en: Actas del quinto taller sobre aprendizaje automático y Cognitive Computing (EMC2) ubicado conjuntamente con la Trigésima Tercera Conferencia sobre Sistemas de Procesamiento de Información Neural (NeurIPS 2019), 2019, págs.
- [5] D. Misra, Mish: una función de activación neuronal no monótona y autoregulada, arXiv:1908.08681 (2019) 1–13.
- [6] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, J. Tsujii, BRAT: una herramienta basada en web para la anotación de texto asistida por PNL, en: Actas de las manifestaciones en el 13 Conferencia del Capítulo Europeo de la Asociación de Lingüística Computacional (EACL '12), 2012, págs. 102–107.
- [7] Y. Wu, M. Schuster, Z. Chen, QV Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., traducción automática neuronal de Google sistema: cerrando la brecha entre la traducción humana y automática, arXiv:1609.08144 (2016) 1–23.
- [8] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Fun-towicz, J. Brew, HuggingFace's Transformers: procesamiento del lenguaje natural de última generación, arXiv:1910.03771 (2019) 1–11.
- [9] J. Cañete, G. Chaperon, R. Fuentes, J. Pérez, Modelo BERT preentrenado en español y datos de evaluación, en: Actas del taller práctico de aprendizaje automático para países en desarrollo en la Octava Conferencia Internacional sobre Representaciones del Aprendizaje (ICLR). 2020), 2020, págs. 1–9.
- [10] I. Loshchilov, F. Hutter, Regularización de la caída de peso desacoplada, en: Actas de la Séptima Conferencia Internacional sobre Representaciones del Aprendizaje (ICLR 2019), 2019, págs.
- [11] López-Úbeda P, Perea-Ortega JM, DG. Manuel C., MT Martin-Valdivia, LA Ureña-López, SINAI en eHealth-KD Challenge 2020: Combinando incrustaciones de palabras para el reconocimiento de entidades nombradas en registros médicos españoles Conferencia de la Sociedad Española de Procesamiento del Lenguaje Natural, Iber-LEF@SEPLN 2020,
- [12] E. Andrés, O. Sainz, A. Atutsa, O. López de Lacalle, IXA-NER-RE en eHealth-KD Challenge 2020: Aprendizaje por transferencia translingüe para la extracción de relaciones médicas, en: Proceedings of the Iberian

Foro de Evaluación de Lenguas coincidiendo con el 36º Congreso de la Sociedad Española de Ciencias Naturales
Procesamiento del Lenguaje, IberLEF@SEPLN 2020, 2020.

- [13] S. Medina, J. Turmo, TALP en eHealth-KD Challenge 2020: Redes neuronales convolucionales y recurrentes de
varios niveles para la clasificación conjunta de frases clave y relaciones en: Actas de ,
el Foro de Evaluación de Lenguas Ibéricas coincidiendo con el 36º Congreso de la Sociedad Española
para Procesamiento del Lenguaje Natural, IberLEF@SEPLN 2020, 2020.