

UH-MAJA-KD en eHealth-KD Challenge 2020: Profundo

Modelos de aprendizaje para el descubrimiento de conocimientos en Documentos españoles de eSalud

Alejandro Rodríguez-Pérez, Ernesto Quevedo-Caballero, Jorge Mederos-Alvarado, Rocío Cruz-Linares and Juan Pablo Consuegra-Ayala

a Facultad de Matemáticas e Informática, Universidad de La Habana, 10200 La Habana, Cuba

Resumen Este

artículo describe la solución presentada por el equipo de UH-MAJA-KD en IberLEF 2020: eHealth Knowledge Discovery Challenge. Se desarrollaron estrategias separadas para resolver las Tareas A y B, ambas basadas en modelos de aprendizaje profundo que utilizan incorporaciones contextuales obtenidas de un modelo BERT previamente entrenado y algunas otras características sintácticas. Proponemos una estrategia que utiliza un modelo híbrido para la Tarea A que utiliza capas LSTM bidireccionales apiladas como codificadores contextuales y campos aleatorios condicionales de cadena lineal como decodificadores de etiquetas. El sistema aborda la Tarea B en forma de consulta por pares, codificando información sobre la oración y el par de entidades dado utilizando estructuras sintácticas derivadas del árbol de análisis de dependencia, por medio de redes neuronales recurrentes basadas en LSTM. La salida se obtiene puntuando todas las relaciones posibles mediante un perceptrón multicapa con una función de activación sigmoidea. Nuestro modelo logró obtener un alto rendimiento en las cuatro tareas de la competencia. El sistema ocupó el tercer lugar en el escenario de evaluación principal, con una diferencia de 0,001 con el segundo lugar. Además, ocupó el segundo lugar en la evaluación encargada de medir el desempeño en la Tarea B, considerada la más difícil en ediciones anteriores del desafío.

Palabras clave

Descubrimiento de conocimientos, extracción de información, reconocimiento de entidades nombradas, extracción de relaciones, profundidad Apreciando

1. Introducción

IberLEF 2020: eHealth Knowledge Discovery Challenge [1] (o eHealth-KD 2020 para simplificar) es un evento que tiene como objetivo ampliar los límites del estado del arte en el área de descubrimiento de conocimiento, particularmente para textos de contenido médico. Hasta el momento cuenta con dos ediciones anteriores [2, 3], que definen tareas similares. El desafío se divide en dos tareas: A y B; uno para la extracción y clasificación de entidades, y el otro orientado a la extracción de relaciones semánticas entre pares de dichas entidades.

Este artículo describe la solución presentada por el equipo de UH-MAJA-KD en eHealth-KD 2020. Propone un modelo híbrido para la Tarea A que utiliza memorias apiladas bidireccionales de largo corto plazo.

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

a.rodriguez4@estudiantes.matcom.uh.cu (A. Rodríguez-Pérez); e.quevedo@estudiantes.matcom.uh.cu (Ml.

Quevedo-Caballero); jorge.mederos@estudiantes.matcom.uh.cu (J. Mederos-Alvarado); rociocl@matcom.uh.cu (R.

Cruz-Linares); jpmotherinlaw@matcom.uh.cu (JP Suegra) orcid: 0000-0002-0069-8 (R.

Cruz-Linares); 0000-0003-2009-393X (JP Consuegra-Ayala)



© 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).

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

ory (BiLSTM) como codificadores contextuales, debido a la estructura secuencial de la entrada y su uso generalizado en la literatura [4] para abordar el problema de reconocimiento de entidades nombradas (NER). Además, se utiliza un campo aleatorio condicional (CRF) de cadena lineal [5] como arquitectura decodificadora de etiquetas para el modelo, ya que se ha utilizado en muchos sistemas NER basados en aprendizaje profundo sobre un BiLSTM, con resultados exitosos [4]. El sistema aborda la tarea de Extracción de Relación (RE) en forma de consulta por pares, codificando información sobre la oración y el par de entidades dado usando estructuras sintácticas derivadas del árbol de análisis de dependencia y por medio de la Memoria a Corto y Largo Plazo . (LSTM) basadas en redes neuronales recurrentes (RNN) para lograr dicho propósito. La información de dependencia ha demostrado ser útil para resolver tareas de ER en varios conjuntos de datos de referencia [6, 7, 8]. El resultado se obtiene puntuando todas las relaciones posibles mediante un perceptrón multicapa (MLP) con una función de activación sigmoidea.

El resto del documento está organizado de la siguiente manera. La sección 2 explica en detalle el sistema propuesto. Los resultados del modelo en los distintos escenarios evaluados durante el evento eHealth-KD 2020 se presentan en la Sección 3. La Sección 4 analiza brevemente cuestiones de interés relacionadas con el desarrollo y desempeño de los modelos. Finalmente, las conclusiones del trabajo se muestran en la Sección 5.

2. Descripción del sistema

El sistema propuesto resuelve ambas tareas de forma separada y secuencial. Así, se definieron modelos independientes para resolver problemas de NER y RE.

La tarea NER se plantea como un problema de predicción de etiquetas que toma el texto sin formato de la oración de entrada y genera dos secuencias de etiquetas independientes: una en el sistema de etiquetas BMEWO-V para predicción de entidades [9] y otra con etiquetas correspondientes a tipos de entidades (Concepto, Acción, Referencia, Predicado) para fines de clasificación. La etiqueta Ninguno se incluye en este último para los casos en los que no hay ninguna entidad presente. Mientras tanto, la tarea RE se interpreta como una serie de consultas por pares entre las entidades presentes en la oración objetivo. Por tanto, predice la existencia de una determinada relación sobre características derivadas tanto de la frase como del par de entidades.

2.1. Preprocesamiento

Dada la oración de destino y las entidades resaltadas ingresadas como texto sin formato, se realiza algún preprocesamiento para derivar estructuras útiles a partir de dicho texto. Dado que ambos modelos utilizan información de fragmentos de palabras, la oración de entrada debe tokenizarse primero. Otros pasos de preprocesamiento incluyen la descomposición de palabras a nivel de caracteres, la extracción de características sintácticas y el análisis de dependencias.

Para obtener una representación de las entradas correspondientes, los modelos hacen uso de lo siguiente características de cada palabra:

Incrustación contextual: incrustaciones contextuales basadas en BERT sin hiperajuste adicional.

Debido al algoritmo de tokenización del modelo BERT, se necesita una determinada estrategia para fusionar palabras divididas en múltiples tokens BERT (por ejemplo, la palabra cáncer podría dividirse en [cán, cer]). En nuestro caso, se hace utilizando la media de los vectores dados. Cada modelo utiliza la concatenación de varios vectores de salida BERT.

Incrustaciones de caracteres: incrustaciones de caracteres basadas en CNN. La entrada a dicha CNN es una Secuencia de índices alfabéticos, los de los caracteres contenidos en la palabra.

Etiquetas POS e incrustaciones de dependencia: incrustaciones destinadas a codificar características sintácticas a nivel de palabra, como la etiqueta POS de la palabra dada y la dependencia con su antepasado. en el árbol de análisis de dependencia.

Etiquetas BMEWO-V y tipo de entidad: BMEWO-V y etiquetas de tipo entidad se utilizan en tareas RE y se obtienen de los resultados del modelo de la Tarea A.

Las incorporaciones contextuales se entrenan previamente sin ningún hiperajuste adicional; mientras que el resto Los pesos de las incrustaciones se optimizan al entrenar el modelo correspondiente.

2.2. Modelo de reconocimiento de entidades nombradas

El modelo recibe la oración como una secuencia de vectores de palabras. Una representación distribuida de cada palabra se obtiene concatenando incrustaciones contextuales, de caracteres y de etiquetas POS, como descrito en el inciso anterior.

En un segundo nivel, la secuencia de tokens es procesada en ambas direcciones por una capa BiLSTM, dando como resultado dos vectores de secuencia. Los vectores en posiciones complementarias de las dos secuencias. se concatenan dando como resultado una nueva secuencia con vectores dependientes del contexto asignados a cada ficha de la oración. Esta secuencia busca encapsular dependencias semánticas. entre las fichas de la frase. Se procesa la secuencia de salida del primer BiLSTM. en ambas direcciones mediante un BiLSTM apilado encima del primero, volviéndose más representativo potencia y dando como resultado la secuencia de vectores \mathbf{v}' .

$$\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n) \quad \mathbf{v}' = (\mathbf{v}'_1, \mathbf{v}'_2, \dots, \mathbf{v}'_n)$$

\mathbf{v}' se introduce en dos capas CRF de cadena lineal, que generan las secuencias de etiquetas más probables

Secuencia según el algoritmo de Viterbi [10]. Sean las salidas correspondientes a la sistema de etiquetas BMEWO-V y el tipo de entidad, respectivamente; \mathbf{y} y los respectivos capas CRF de cadena lineal, entonces:

$$\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n) \quad \mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n)$$

La Figura 1 muestra la arquitectura descrita.

La primera capa CRF produce una secuencia de etiquetas en el sistema de etiquetas BMEWO-V. Esta clasificación corresponde a B para el comienzo de una entidad, M para los tokens en el medio, E para el token final, W en el caso de tokens que son una entidad en sí mismos y O para tokens que no lo son representar cualquier cosa. También tiene en cuenta la posibilidad de superposición de entidades, incluidas la etiqueta V en tales casos.

Es necesario un proceso para transformar las secuencias de etiquetas obtenidas de las capas CRF en una lista de entidades esperadas como resultado de la Tarea A [9]. Este proceso de ahora en adelante se denominará descodificación. Hay un desafío importante en este proceso: los tokens que pertenecen a una entidad son no necesariamente continuo en la oración. Teniendo esto en cuenta, el proceso de decodificación es

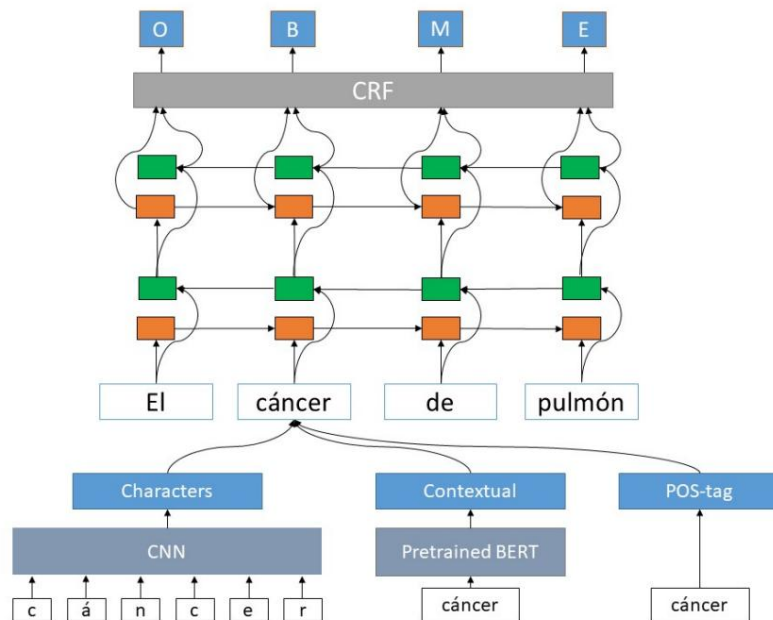


Figura 1: Arquitectura del modelo de Tarea A (solo muestra un resultado).

dividido en dos etapas. En primer lugar se detectan entidades discontinuas y luego, en un segundo momento, entidades continuas.

De acuerdo con el uso correcto del español, el conjunto de secuencias de etiquetas que deben interpretarse como un grupo de entidades discontinuas se redujo a aquellas que coinciden con las expresiones regulares $(V+)((M^*EO^*)+)(M^*E)$ y $((BO)+)(B)(V+)$. El primero corresponde a entidades que comparten sus tokens iniciales, y el segundo a aquellas que comparten sus tokens finales. Estos dos capturan la mayoría de las entidades discontinuas deseadas. Entre los ejemplos del primer caso se encuentra el fragmento *cáncer de pulmón* y *de mama*, etiquetado como VMEOME, donde se encuentran las entidades *cáncer de pulmón* y *cáncer de mama*. Y, como ejemplo de esto último, el fragmento *tejidos y órganos humanos*, etiquetado como BOBV, donde se encuentran entidades *tejidos humanos* y *órganos humanos*. Cuando se detecta una coincidencia y se extraen las entidades, todas las etiquetas de ese fragmento se configuran como etiqueta O.

Después de la detección de posibles entidades discontinuas, la segunda etapa comienza asumiendo que todas las entidades restantes aparecen como secuencias continuas de tokens. Para extraer entidades continuas, se lleva a cabo un proceso iterativo sobre la secuencia de etiquetas producida por el modelo. Debido a las limitaciones del sistema BMEWO-V, el procedimiento también supone que la profundidad máxima de superposición es 2. Asumir lo contrario sólo hace que el proceso sea más ambiguo y no captura mucha más información ya que las superposiciones más profundas no son frecuentes en las colecciones de capacitación y desarrollo. Ante esto, junto con el trámite, se mantienen dos entidades en construcción. En cada iteración, estas dos entidades se crean, amplían o emiten de acuerdo con reglas definidas considerando solo la etiqueta anterior y actual.

2.3. Modelo de extracción de relaciones

La información más completa para resolver la tarea RE se encuentra en la oración de entrada completa. Sin embargo, algunos autores afirman que el árbol de dependencia asociado con la oración de entrada condensa la información más importante y descarta la engañosa [6, 7, 8]. Para determinar una posible relación entre dos entidades, el sistema presentado utiliza como entrada estructuras derivadas del árbol de análisis de dependencia asociado con la oración de destino, para obtener información de tanto la oración como el par de entidades.

Según las observaciones, las entidades resaltadas en las colecciones de oraciones son frases nominales completas (o al menos subconstituyentes de ellas). Algunos de los criterios tomados en consideración establecer una relación de dependencia con un encabezado H en una construcción sintáctica C, es el hecho que H podría reemplazar a C [11]. Además, H podría determinar semánticamente a C. Por otro lado, Las entidades de varias palabras a menudo ocurren completamente en un subárbol de dependencia con raíz en uno de sus tokens. dada una frase 1 , definimos dicho subárbol correspondiente a una entidad e, como árbol relevante para e, y se denota más adelante como \mathcal{A}_e . La raíz se llama entidad e núcleo y se denota n_e .

Otra definición importante, ampliamente utilizada en la literatura para abordar esta tarea, es la dependencia camino entre dos tokens $1, 2$. De ahora en adelante se le denominará $(1, 2)$.

Las estructuras mencionadas anteriormente se introducen en una red neuronal profunda que genera un vector. cuya longitud es la misma que las relaciones establecidas. Cada componente de dicho vector es una puntuación que mide la fuerza con la que está presente la relación correspondiente entre la entrada entidades.

Para ello, el modelo primero codifica cada una de las estructuras \mathcal{A}_e en un vector v_e . v_e o (v_e) están formados por palabras de la oración de entrada. Se obtiene una representación distribuida de cada palabra concatenando contexto, carácter, etiqueta POS, dependencia,

BMEWO-V e incorporaciones de tipo de entidad, tal y como se describe en el apartado anterior.

Para calcular el vector de salida, una capa BiLSTM codifica la secuencia de vectores asociados a las palabras en $(1, 2)$ para incluir información bidireccional en la representación.

$$v_{(1,2)} = \text{BiLSTM}((v_1, v_2))$$

Esta salida se introduce en una capa LSTM unidireccional para enfatizar la dirección de la relación potencial, procesando la secuencia desde el origen hasta el destino. Esto resulta en un vector que codifica la información presente en $(1, 2)$.

$$v_{(1,2)} = \text{LSTM}(v_{(1,2)})$$

Al mismo tiempo, se aplica un ChildSum Tree-LSTM [12] de forma independiente sobre las v_1 y v_2 (es decir representaciones que se obtienen de forma independiente pero utilizando el mismo conjunto de pesos).

$$v_1 = \text{ChildSumTreeLSTM}(v_1) \quad v_2 = \text{ChildSumTreeLSTM}(v_2)$$

Los vectores que codifican las estructuras de entrada están concatenados. El resultado final se obtiene mediante aplicando una función sigmoidea a una transformación lineal de la misma.

¹Para simplificar, se omitirá cualquier notación relacionada con la oración siempre que quede clara en el contexto.

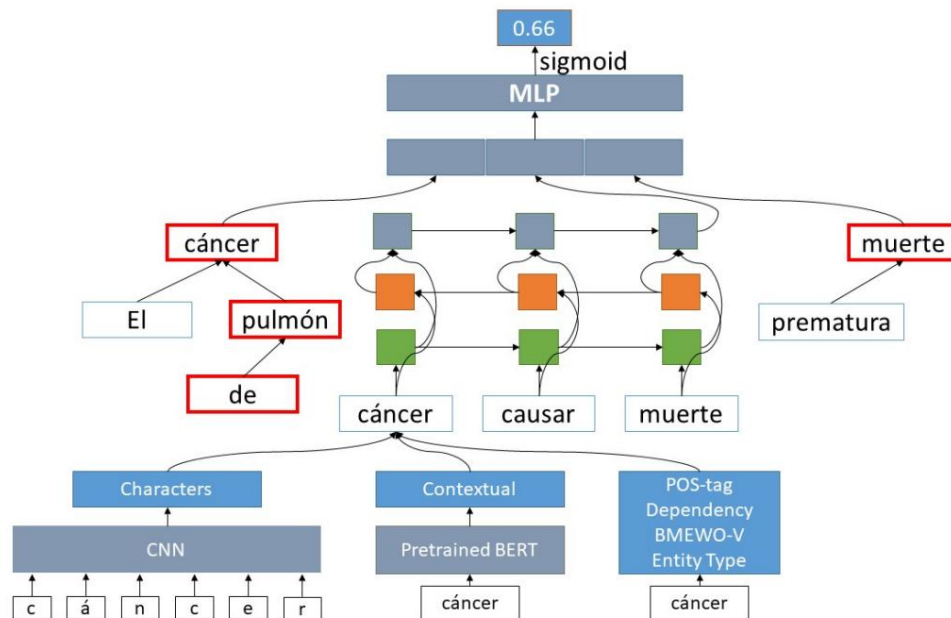


Figure 2: Task B model architecture. Input sentence is El cáncer de pulmón puede causar muerte prematura with the highlighted entities cáncer de pulmón and muerte.

$$= [\dots;]$$

$$= (\dots + \dots)$$

Según las puntuaciones presentes en el vector de salida. , si alguno de sus componentes excede un umbral dado, entonces se predice la relación con la puntuación máxima. Si ninguna de las puntuaciones es mayor que dicho umbral, entonces no se reporta ninguna relación. El valor umbral se añade como un hiperparámetro y optimizado utilizando la colección de desarrollo. Observe que este enfoque nos permite descartar el uso de una relación falsa none.

La Figura 2 muestra la arquitectura descrita.

2.4. Configuración y entrenamiento de parámetros

Para ambas subtarefas, el procedimiento de capacitación se llevó a cabo utilizando únicamente la colección de capacitación proporcionada a los concursantes.

Dado que la capa CRF está destinada a maximizar la probabilidad de obtener una secuencia de etiquetas deseada dado un vector de características de entrada, el modelo de la Tarea A se entrena para minimizar el registro negativo de la probabilidad (|). Sean y las matrices de emisiones y transición de CRF respectivamente. Entonces, esa probabilidad se define como la exponencial normalizada:

$$(|) = \frac{\exp(\sum_{i=1}^n (\dots) + \sum_{i=1}^{n-1} (\dots, +1))}{()}$$

siendo un factor de normalización dependiendo del vector de entrada. Y la función de pérdida es definido en términos de y de la siguiente manera:

$$L(y, \hat{y}) = - \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

En el caso del modelo de Tarea B, dado que cada componente de salida es independiente entre sí, el modelo está entrenado para minimizar una función binaria de entropía cruzada sobre el vector de salida. Sea el número de relaciones, el vector de salida y el vector objetivo, la función de pérdida se calcula de la siguiente manera:

$$L(y, \hat{y}) = - \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Como se explicó anteriormente, el resultado del modelo no utiliza la relación falsa none. Se utiliza una estrategia de muestreo negativo para optimizar el modelo con ejemplos en los que no existe ninguna relación. Una muestra negativa no es más que un ejemplo de entrenamiento donde la salida objetivo es el vector nulo. Dicho muestreo se realiza utilizando una proporción fija de pares de entidades no relacionadas.

Se utilizaron estrategias de abandono durante el procedimiento de entrenamiento en ambos modelos para reducir el sobreajuste. Para la Tarea A, se apilaron dos capas de abandono después del primer y segundo BiLSTM, y se agregó una exclusión espacial 2D después de la capa CNN utilizada para calcular la incrustación de caracteres de las palabras. En el modelo de Tarea B, se apilaron tres capas de abandono después de las capas BiLSTM, LSTM y TreeLSTM respectivamente.

El número de épocas se seleccionó empíricamente, basándose en la convergencia de los modelos, como lo mostraron las curvas de aprendizaje. Para el ajuste de hiperparámetros y la selección de modelos, se llevó a cabo un proceso de validación cruzada utilizando la colección de desarrollo. La Tabla 1 muestra la configuración de hiperparámetros para ambos modelos.

2.5. Implementación

Los sistemas se implementaron utilizando el lenguaje de programación Python, con la biblioteca PyTorch (v1.4.0) como marco de redes neuronales profundas. Las incrustaciones contextuales basadas en BERT se obtuvieron del modelo preentrenado bert-multilingual-uncased, utilizando la biblioteca de Python pytorch-pretrained-bert2 (v0.6.2). La etiqueta POS y el árbol de dependencias se obtuvieron utilizando la biblioteca Python spaCy3 (v2.2.1), específicamente el modelo es_core_news_md (93 MB).

Ambos modelos fueron entrenados en una computadora personal con las siguientes características: Intel(R) CPU Core(TM) i7-6500 a una frecuencia de 2.50GHz, con una memoria instalada de 8.00 GB sin GPU disponible para CUDA. El tiempo total de formación para el modelo de entidad tomó menos de 5 horas, mientras que el modelo de relación fue cercano a 12 horas.

3. Resultados

La evaluación en ambas tareas se realizó utilizando el corpus comentado propuesto en el desafío. Los resultados se midieron con una medida estándar F1 como se describe en detalle en la

²<https://pypi.org/project/pytorch-pretrained-bert/>

³<https://spacy.io/>

tabla 1

Configuración de hiperparámetros para los modelos NER (izquierda) y RE (derecha)

Parámetro	Valor	Parámetro	Valor
Incrustaciones de entrada			
Incrustación contextual 3072 (últimos cuatro)		Incrustación contextual 768 (última capa)	
Incrustación de personajes	50	Incrustación de personajes	50
Incrustación de etiquetas POS	50	Incrustación de etiquetas POS	50
		Incrustación de dependencia	50
		Incrustación de etiquetas BMEWO-V	50
		Incrustación de tipo de entidad	50
Red neuronal			
Tamaño oculto de CNN	100	Tamaño oculto de CNN	100
Abandono espacial 2D	0,5	Tamaño oculto BiLSTM	100
BiLSTM_1 tamaño oculto	300	Tasa de deserción escolar	0,2
Tasa de abandono_1	0,5	Tamaño oculto de LSTM	50
BiLSTM_2 tamaño oculto	300	Tasa de deserción escolar	0,5
Tasa de abandono_2	0,5	Árbol-LSTM oculto	50
		Tasa de deserción escolar	0,5
Capacitación			
Optimizador	Adán	Optimizador	Adán
Tasa de aprendizaje	0,001	Tasa de aprendizaje	0.001
Épocas	50	Épocas	30
Parámetros totales	4681528	Parámetros totales	689713

descripción general del desafío [1]. Además, se registraron y presentaron medidas de precisión y recuperación.

La Tabla 2 presenta los resultados oficiales de la competencia, dados por el escenario de evaluación 1. Como se muestra, nuestro sistema quedó clasificado como el tercero mejor, logrando una puntuación general de F1 de 0,625, bastante cerca al segundo mejor, y bastante lejos del primer y cuarto sistema clasificado.

La Tabla 3 muestra los resultados correspondientes a los escenarios 2 y 3, donde se evaluaron las Tarea A y B. independientemente. Con puntuaciones de F1 de 0,814 y 0,598, nuestro sistema pudo llegar al cuarto y segundas posiciones en los escenarios de evaluación de Tarea A y B, respectivamente.

Finalmente, en esta edición del desafío se propuso un escenario adicional de transferencia de aprendizaje. El escenario 4 evalúa las capacidades de generalización de los sistemas a dominios de temas generales.

La Tabla 4 compara los resultados de los sistemas participantes en este escenario de evaluación.

Como puede verse, todos los sistemas tuvieron un peor desempeño en este escenario que en el escenario 1, pero el Los pedidos se mantuvieron casi idénticos. El nuestro quedó tercero como en el escenario 1, con una puntuación F1 de 0,547.

Tabla 2

Resultados del escenario 1

Equipo	F1	Recuperación de precisión	
Vicomtech 0,665 Talp-UPC		0,679	0,652
0,626 UH-MAJA-KD 0,625 IXA-		0,626	0,626
NER-RE 0,557		0,634	0,615
		0,580	0,536
UH-MatCom 0.556		0,716	0.455
SINAI	0.420	0,651	0.310
FLIPBOARD	0,395	0,458	0.347
base	0,395	0,458	0.347
exsim	0.245	0.312	0.202

Tabla 3

Resultados del escenario 2 (izquierda) y 3 (derecha)

Equipo	F1	Recuperación de precisión		Equipo	F1	Recuperación de precisión	
SINAI	0,825	0.844	0.806	IXA-DOWN-RE	0.633	0,647	0,619
Vicomtech	0.820	0.821	0.820	UH-MAJA-KD 0,598 Vicomtech		0,629	0.571
Talp-UPC	0.815	0.807	0.824	0,583 Talp-UPC UH-MatCom		0,671	0.515
UH-MAJA-KD 0.814 UH-MatCom		0.820	0.808	0,545 SINAI	0.574	0,646	0.517
0.794 IXA-NER-RE HAPLAP		0.824	0,767	HAPLAP Línea base ExSim		0,682	0.453
base ExSim	0,691	0,726	0.660		0.461	0,627	0.365
	0.541	0.503	0.586		0.316	0.327	0.305
	0.541	0.503	0.586		0,131	0.527	0,075
	0.314	0,292	0,339		0,131	0.527	0,075

Tabla 4

Resultados del escenario 4

Equipo	F1	Recuperación de precisión	
Talp-UPC	0.583	0.604	0.563
Vicomtech	0.563	0,594	0.535
UH-MAJA-KD 0,547 IXA-NER-		0.608	0,498
RE 0,478 UH-MatCom 0,373		0.563	0.416
SINAI HAPLAP línea base		0,726	0.250
ExSim	0.281	0.626	0.181
	0,137	0.281	0.091
	0,137	0,281	0.091
	0,122	0,253	0.080

4. Discusión

Las incrustaciones contextuales basadas en BERT demostraron contener características útiles para resolver tanto NER y tareas de ER. También experimentamos con incrustaciones de palabras previamente entrenadas y entrenadas en un corpus de contenido médico extraído de Wikipedia, y el modelo con las características BERT superó

este último en ambas tareas. El uso combinado de ellos resultó ser bastante ineficaz. Además, se probaron modelos BERT con y sin carcasa y el modelo con carcasa mostró mejores resultados en el NER.

modelo. Sin embargo, la tarea de ER experimentó una disminución bastante insignificante en el rendimiento. Estos resultados se basan en el proceso de selección de modelos realizado utilizando la colección de desarrollo.

Vale la pena mencionar que nuestros experimentos demostraron que BERT no es suficiente para resolver ninguna de las tareas descritas en el desafío eHealth-KD 2020. Los modelos basados únicamente en BERT (es decir, modelos con entradas basadas en BERT y sus correspondientes capas de salida CRF o MLP) no lograron un buen desempeño en ambas tareas.

En ambos casos, todas las características utilizadas como entradas, como se describe en la Sección 2, resultaron determinantes para lograr el máximo rendimiento. En el caso particular del modelo RE, además de las incrustaciones contextuales basadas en BERT, la información obtenida de la tarea NER sobre las entidades (es decir, la etiqueta BMEOW-V y el tipo de entidad) son las más influyentes.

Finalmente, en cuanto al proceso de entrenamiento, cabe destacar el hecho de que el tiempo de entrenamiento del modelo NER es significativamente más corto que el correspondiente al modelo RE. Esto es algo esperado, ya que nuestro enfoque RE define como ejemplo de entrenamiento una oración y un par de entidades (por lo tanto, muchos más ejemplos de entrenamiento). Además, al igual que en eventos anteriores de eSalud, los sistemas siguen teniendo un desempeño significativamente peor en la tarea de ER. Estos hechos podrían llevar a la conclusión de que aún quedan muchos hallazgos por hacer.

5. Conclusiones

En este trabajo se describe el sistema propuesto por el equipo de UH-MAJA-KD en IberLEF eHealth- KD 2020: eHealth Knowledge Discovery Challenge. Para la subtarea A, se propuso un modelo híbrido Stacked-BiLSTM-CRF, que utiliza incorporaciones contextuales previamente entrenadas por BERT y otras características sintácticas. Este modelo obtuvo un desempeño competitivo en el Escenario 2, donde se ubicó en el cuarto lugar con una pequeña diferencia con respecto a los 3 primeros. La tarea B se abordó mediante una consulta por pares, codificando información sobre la oración y el par de entidades dado usando estructuras sintácticas derivadas del árbol de análisis de dependencia y mediante RNN basado en LSTM. La salida se obtiene puntuando todas las relaciones posibles mediante un perceptrón multicapa con una función de activación sigmoidea. Nuestro modelo obtuvo el segundo lugar en el Escenario 3. El sistema alcanzó la tercera posición en la clasificación general (Escenario 1), y también en el escenario de transferencia de aprendizaje (Escenario 4).

Se propone como trabajo futuro afinar las incorporaciones de BERT junto con el proceso de entrenamiento de los modelos propuestos, buscando incorporaciones contextuales para entrenar específicamente para estas tareas. Además, utilizar funciones específicas de dominio, como nomencladores, que buscan mejorar el rendimiento en el ámbito de la salud. Finalmente, desarrollar y evaluar el uso de modelos conjuntos para la resolución de ambas tareas.

Referencias

- [1] Piad-Morffis A, Gutiérrez Y, Cañizares-Díaz H, Estevez-Velarde S, Almeida-Cruz Y, Muñoz R, Montoyo A, Descripción general del eHealth Knowledge Discovery Challenge en IberLEF 2020, en: Proceedings of the Iberian Languages Foro de Evaluación ubicado conjuntamente con

- 36º Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural, IberLEF@SEPLN 2020, España, septiembre de 2020., 2020.
- [2] E. Cámara Martínez, Y. Almeida Cruz, MC Díaz Galliano, S. Estevez-Velarde, M.A. García Cumberas, M^a García Vega, Y. Gutiérrez, A. Montejo Ráez, A. Montoyo, R. Muñoz, et al.
- [3] Piad-Morffis A, Gutiérrez Y, Consuegra-Ayala JP, Estevez-Velarde S, Almeida-Cruz Y, Muñoz R, Montoyo A , en: Actas del Foro de Evaluación de Lenguas Ibéricas (IberLEF 2019). Actas del taller CEUR , CEUR-WS. org, 2019.
- [4] J. Li, A. Sun, J. Han, C. Li, Una encuesta sobre aprendizaje profundo para el reconocimiento de entidades nombradas, IEEE Transactions on Knowledge and Data Engineering (2020).
- [5] J. Lafferty, A. McCallum, FC Pereira, Campos aleatorios condicionales: modelos probabilísticos para segmentación y etiquetado de datos de secuencia (2001).
- [6] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, H. Wang, Una red neuronal basada en dependencia para clasificación de relaciones, preimpresión de arXiv arXiv:1507.04646 (2015).
- [7] S. Zhang, D. Zheng, X. Hu, M. Yang, Redes bidireccionales de memoria a corto plazo para clasificación de relaciones, en: Actas de la 29.^a conferencia de Asia Pacífico sobre lenguaje, información y computación, 2015, págs. 73–78.
- [8] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, Z. Jin, Clasificación de relaciones a través de redes de memoria a largo plazo a lo largo de caminos de dependencia más cortos, en: actas de la conferencia de 2015 sobre métodos empíricos. en el procesamiento del lenguaje natural, 2015, págs. 1785-1794.
- [9] JM Alvarado, EQ Caballero, A. Rodríguez, Uh-maja-kd en el desafío ehealth-kd (2019).
- [10] A. Viterbi, Límites de error para códigos convolucionales y un algoritmo de decodificación asintóticamente óptimo, Transacciones IEEE sobre teoría de la información 13 (1967) 260–269.
- [11] AM Zwicky, Heads, Revista de lingüística 21 (1985) 1–29.
- [12] KS Tai, R. Socher, CD Manning, Representaciones semánticas mejoradas a partir de redes de memoria a corto plazo con estructura de árbol, preimpresión de arXiv arXiv:1503.00075 (2015).