Detección Automática de Patologías Combinando Modelos de Lenguaje y Ontologías Médicas

Adam Lei Yi Chen Abolacio
Universidad de Sevilla, España
adacheabo@alum.us.es

Abstract

En este trabajo se presenta un enfoque híbrido para la detección automática de patologías dermatológicas en textos clínicos, combinando modelos de lenguaje con ontologías médicas. Se analizan resultados y arquitectura propuestos.

1 Introducción

El presente trabajo tiene como objetivo la aplicación de la metodología presentada en el artículo [1]. El artículo ofrece un nuevo método de clasificación de patologías médicas sobre enfermedades dermatológicas mediante el uso de grandes modelos del lenguaje (por sus siglas en inglés LLM). Como contribución adicional a la comunidad, hacen público el conjunto de datos generado llamado DermatES [2], que ya se encuentra enriquecido por las ontologías empleadas por los autores. Este conjunto de datos es muy valioso, ya que como apuntan los autores, en la actualidad el número de *datasets* disponibles en español es muy escaso, sobre todo en clasificación de textos, mientras que si realizamos una búsqueda en la red, nos daremos cuenta que la gran mayoría de *datasets* públicos disponibles están destinados a la tarea de *Named Entity Recognition (NER).*

2 Estado de la cuestión

Tras realizar una búsqueda exhaustiva en la red, no ha sido posible encontrar conjuntos de datos sobre clasificación de patologías médicas, al menos en español. El único *dataset* disponible para esta tarea (de manera pública) es el presentado en [1], por lo que, a priori, no era posible aplicar dicha metodología al no contar con conjuntos de datos para esta tarea, sin recurrir a la síntesis de datos.

Debido a esta problemática, se decidió recurrir a *datasets* sobre clasificación de patologías médicas en inglés, siendo el conjunto finalmente empleado Skin Disease Text Classification [3].

3 Metodología

En esta sección presentamos la descripción del conjunto de datos empleado y el conjunto de técnicas empleadas para el enriquecimiento de los datos con ontologías médicas y la nueva arquitectura basada en modelos en cascada para clasificación.

3.1 Descripción del conjunto de datos

El conjunto de datos empleado [3] se encuentra disponible en la plataforma Kaggle y está formado por las descripciones que dan los pacientes sobre los síntomas que tienen junto con la enfermedad finalmente asociada. Este *dataset* es interesante ya que guarda cierta relación con el *dataset* DermatES, aunque, en DermatES contamos con la nota clínica escrita por un médico, por lo que es posible que dichas notas clínicas sean más acertadas para la detección de una enfermedad que sólo contar con la descripción que nos da el paciente. El *dataset* se encuentra formado por 141 ejemplos con 13 enfermedades diferentes.

En este caso, los datos ya se encuentran anonimizados y balanceados, con un total de 7.7% de datos por cada clase.

3.2 Modelos de lenguaje

En este caso, al no contar con una gran cantidad de datos que normalmente son necesarios para entrenar un modelo de lenguaje, se ha optado por realizar *full fine-tuning* con modelos preentrenados. La opción más interesante sería emplear un modelo preentrenado con un corpus médico, sin embargo, debido a las limitaciones técnicas de utilizar la versión gratuita de google colab, se ha optado por utilizar los modelos basados en BERT que están disponibles en tamaños reducidos.

Se ha decidido optar por *full fine tuning* en vez de *fine tuning* al observar que el tiempo de entrenamiento es similar y se obtiene un rendimiento mucho mayor al *fine tuning*, ya que debido a los tamaños empleados, los modelos empleando *fine tuning* quedaban congelados con métricas muy bajas.

El modelo finalmente empleado es bert_small_en_uncased [4], un modelo transformer con dos capas encoding, un total de 28 millones de parámetros entrenables, trabaja únicamente con minúsculas y fue entrenado con un corpus en inglés. Este modelo ha sido extraído y entrenado a través de la librería Keras.

3.3 Ontologías utilizadas

El objetivo de emplear ontologías médicas es enriquecer el conjunto de datos aportando más información sobre las enfermedades a clasificar.

De manera análoga a [1], ambos conjuntos tratan sobre enfermedades de la piel, por lo que es una buena opción utilizas las mismas ontologías que emplean dichos autores, SNOMED, UMLS e ICD10. Para lograrlo, los autores utilizan la biblioteca *PyMedTermino* para acceder de manera automática a las patologías y asociarle a cada una de ellas el tipo de patología, la localización de la enfermedad y la severidad de la patología.

Sin embargo, para nuestro caso no ha sido posible realizar la extracción de las características necesarias de dichas ontologías, ya que las bases de datos no se pueden descargar sin una licencia o consentimiento previo. Por lo que, de manera semiautomática, se han extraído las características de las enfermedades disponibles en DermatES y se ha realizado la extracción de manera manual utilizando los *browsers* públicas de las ontologías SNOMED e ICD10. SNOMED nos indica el tipo y la localización, mientras que ICD10 nos indica la severidad de la enfermedad.

3.4 Modelo en cascada

La arquitectura empleada trata de cuatro modelos de lenguaje en cascada, donde la entrada de un modelo es la salida del anterior concatenando la descripción del paciente. El objetivo final de la arquitectura es la predicción de la enfermedad, por lo que podemos variar el orden en el que realizamos la predicción de las otras tres características: el tipo, sitio y gravedad.

En este caso nos hemos guiado por [1], donde los autores demuestran que se obtiene un mejor rendimiento siguiendo el esquema de predicción en el siguiente orden:

1. Tipo de enfermedad
2. Localización de la enfermedad
3. Gravedad de la enfermedad
4. Predicción final de la enfermedad

De esta manera nos evitamos realizar un total de seis entrenamientos por cuatro modelos en cascada. Si contásemos con el tiempo y computación suficientes, sería interesante validar si el mejor orden de los autores vuelve a repetirse con datos diferentes.

3.5 Modelos adicionales

Para realizar una comparativa entre este método de enriquecimiento mediante ontologías médicas y el modelo en cascada, se han seleccionado algunos modelos de *machine learning* clásico como son: *Random Forest*, *SVC* y *Logistic Regression* para comprobar sus rendimientos. Estos modelos han sido combinados utilizando *Bag of Words* para la vectorización de los textos.

Adicionalmente, hemos empleado el mismo modelo BERT para realizar la clasificación exclusivamente con la descripción del paciente.

3.6 Entrenamiento de modelo en cascada

El entrenamiento de los modelos en cascada se ha realizado utilizando la misma partición tanto de *train* como *validation* y *test*. Sin embargo, para cada modelo concreto se ha utilizado la información agregada correspondiente junto con la descripción del paciente.

De esta manera, durante el entrenamiento, los modelos reciben la información correcta de las ontologías, a excepción del modo predicción, donde los modelos utilizarán la información predicha por los modelos anteriores, partiendo del modelo predictor del tipo de patología hasta el modelo final que predice la enfermedad.

4 Resultados

En este apartado mostraremos y analizaremos los resultados obtenidos en todos los experimentos realizados.

Para los modelos basados en *transformers* se ha fijado un límite máximo de 40 *epochs* para los modelos en cascada y 30 *epochs* para el modelo *transformer* que trabaja exclusivamente con la descripción del paciente. Este valor máximo de *epochs* fijado es un valor elegido suficiente para permitir al modelo alcanzar el mejor rendimiento posible. Adicionalmente, se ha implementado un método de *EarlyStopping* con una paciencia de 5 *epochs* para evitar mínimos locales y se evalúan los resultados con los parámetros que menor valor de función de pérdida han obtenido los modelos en validación.

4.1 Modelos sin enriquecimiento ontológico

Como podemos ver en 1, el rendimiento de los modelos de *machine learning* clásicos se encuen-

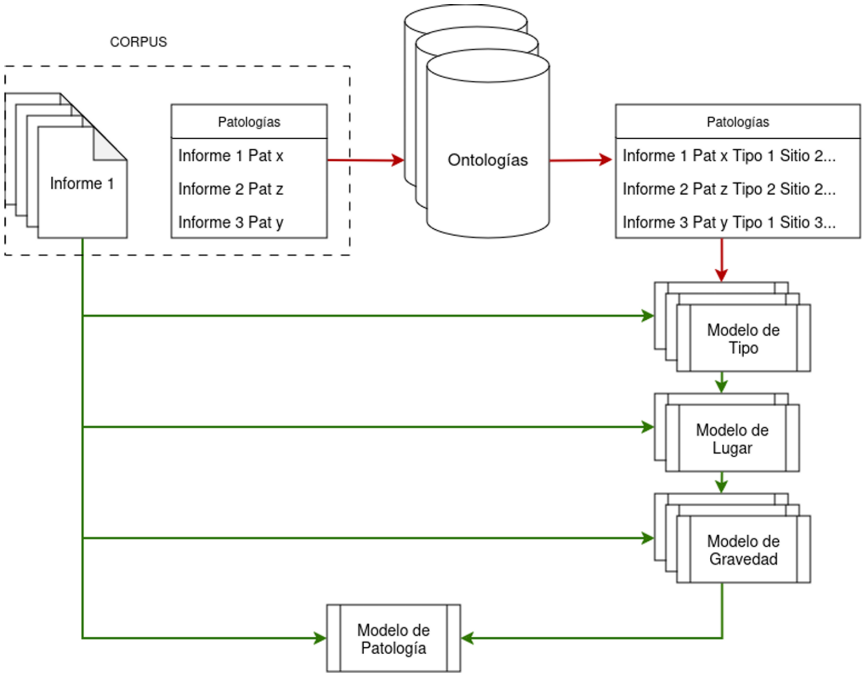

Figure 1: Arquitectura propuesta por modelos en cascada. La salida de un modelo es recibida por el siguiente concatenando la descripción del paciente.

Model	Acc	Prec	F1
Logistic Regression	68.2	72.0	66.7
Random Forest	63.6	71.6	62.7
SVC	40.9	42.4	36.2
BERT Small	77.3	83.3	78.5

Table 1: Resultados de los rendimientos obtenidos en la partición de *test* utilizando exclusivamente la descripción del paciente.

Model	Acc	Prec	F1
Type	68.2	65.3	64.4
Location	81.8	75.9	77.4
Severity	90.9	92.3	90.8
Disease	95.5	97.7	95.5

Table 2: Resultados de los rendimientos obtenidos por los modelos en cascada en la partición de *test* en modo entrenamiento

tra por debajo del rendimiento del modelo *transformer* BERT. De esta forma, podemos afirmar que el rendimiento obtenido por el modelo BERT puede ser aceptable, teniendo en cuenta que nos encontramos ante un problema de clasificación de 13 clases y que contamos únicamente con 99 ejemplos para el entrenamiento. Por el contrario, destacar que, pese a tener tanto un método de vectorización sencillo como es *BoW* y un modelo sencillo de regresión logística, podemos obtener un modelo *baseline* con cierta robustez.

4.2 Modelos con enriquecimiento ontológico

En este apartado mostramos los rendimientos obtenidos por los diferentes modelos en modo entrenamiento, los modelos se entrenan con la información de la ontología de la enfermedad correcta.

Los resultados obtenidos en 2, demuestran que el enriquecimiento mediante ontologías mejora enormemente el rendimiento del modelo final que predice la enfermedad, ya que también es el modelo que mayor información sobre la enfermedad. Otro aspecto interesante es el hecho de que el rendimiento de los modelos aumenta gradualmente a medida que contamos con más información.

Sin embargo, el rendimiento obtenido por el primero modelo, el modelo predictor del tipo de enfermedad, al ser mucho más bajo que el resto de modelos, nos puede hacer empeorar el rendimiento final del modelo en cascada. Otro aspecto importante es el desbalanceo de clases de cada modelo, ya que, a excepción del modelo final predictor de la enfermedad, el resto de modelos no cuenta con un conjunto de datos balanceado ya que las enfermedades analizadas, de manera general, no van a presentar dicho balanceo.

Model	Acc	Prec	F1
Type	68.2	79.5	72.0
Location	59.1	67.7	61.3
Severity	68.2	68.8	68.3
Disease	54.5	81.8	56.1

Table 3: Resultado de los modelos en cascada en modo oráculo.

4.3 Modelo en cascada

Finalmente, se ha comprobado el rendimiento obtenido por los modelos en lo que los autores denominan modo oráculo, utilizando las predicciones de las características de la ontología del modelo anterior.

El rendimiento final de la arquitectura de modelos en cascada 3 es el marcado por el modelo final *Disease*, lo que refleja un rendimiento que no es especialmente malo, con un 54.5% de *accuracy* sobre un total de 13 clases. Aun así, el rendimiento final sorprende teniendo en cuenta que durante el entrenamiento el modelo final *Disease* obtenía métricas muy altas, mientras que en modo oráculo el rendimiento se ve muy mermado. Este hecho demuestra la importancia de que, en una arquitectura como esta, mantener un balance en el rendimiento entre modelos es un aspecto muy importante.

5 Conclusiones y trabajo futuro

Tras observar y analizar los resultados obtenidos en todos los experimentos realizados, podemos concluir que el uso de ontologías resulta en modelos de clasificación mucho más precisos, tal y como podemos ver en el rendimiento obtenido durante el entrenamiento por *Disease* en 2 y el resto de modelos que trabajan sin el enriquecimiento ontológico 1.

Sin embargo, el rendimiento final del modelo en cascada ha sido decepcionante, teniendo en cuenta que empleando un único modelo obtenemos mejor rendimiento que toda la arquitectura de modelos en cascada. Además, estos modelos en cascada resultan en costes de entrenamiento mucho mayores y una complejidad intrínseca en el manejo de datos desbalanceados.

A pesar de esta problemática, esta arquitectura se podría seguir intentando explotar explorando nuevas opciones como cambiar el orden entre los modelos intermedios o cambiar la forma de realizar el entrenamiento. En vez de realizar un entrenamiento por separado con los datos correctos, podemos experimentar entrenando los modelos en cascada en forma de árbol, utilizando las predicciones de los modelos anteriores y no la información ontológica. No sólo el orden importa, también podemos analizar la inclusión, eliminación e incluso sustitución de ciertos modelos intermediarios para observar si una disminución del número de modelos resulta en un mejor rendimiento.

References

[1] Léon-Paul Schaub Torre, Pelayo Quirós, and Helena García Mieres. *Detección Automática de Patologías en Notas Clínicas en Español Combinando Modelos de Lenguaje y Ontologías Médicos*. 2024. arXiv: 2410.00616 [cs.CL]. URL: https://arxiv.org/abs/2410.00616.

[2] Fundación CTIC. *DermatES: Spanish Clinical Dermatology Reports Dataset*. https://huggingface.co/datasets/fundacionctic/DermatES. Accessed: 2025-05-31. 2024.

[3] Rafsun Ahmad. *Skin Disease Text Classification*. https://www.kaggle.com/datasets/rafsunahmad/skin-disease-text-classification/data. Kaggle dataset. 2023.

[4] Keras. *Keras BERT Small EN Uncased 3*. Accedido: 31 de mayo de 2025. 2025. URL: https://www.kaggle.com/models/keras/bert/keras/bert_small_en_uncased/3.