Informe Final

Identificación de entidades en prescripciones de medicamentos usando NLP

Integrantes: Daniel Carmona
Camilo Carvajal
Monserrat Prado
Martín Sepúlveda
Profesores: Constanza Contreras
Francisco Förster

Fecha de entrega: 23 de diciembre de 2022
Santiago de Chile

Índice de Contenidos

1. Introducción y Contexto

El Ministerio de Salud (MINSAL) es el ministerio encargado de la coordinación y gestión de la salud en Chile. Entre sus funciones se encuentran: formular y gestionar planes de salud, dictar normas en dicha materia, velar por el cumplimiento de normas sanitarias, establecer protocolos de atención y evaluar la situación en salud de la población. Actualmente, quien ocupa el cargo de ministra de la cartera es Ximena Aguilera.

Uno de los asuntos que el ministerio tiene que monitorear son las recetas médicas. Estas, para que logren cumplir adecuadamente su función de mejorar el estado de salud de un paciente, deben incluir información ciertos elementos importantes como el nombre del medicamento, la forma farmacéutica, la dosis, la vía de administración, la frecuencia y la duración del tratamiento. Sin embargo, existen algunas que carecen de alguno de estos datos, lo cual da pie a errores de medicación, a un posible empeoramiento en el estado del paciente.

Es por este motivo que surge este proyecto, el cual se sustenta a través de un Fondo de Fomento al Desarrollo Científico y Tecnológico (FONDEF). Aquí, el objetivo principal fue reconocer los elementos importantes de las prescripciones médicas utilizando algoritmos de Procesamiento de Lenguaje Natural (NLP). Estos, para la tarea a la cual pertenece el problema, corresponden a entidades. Al reconocer estas entidades, se pueden detectar errores de medicación con mayor facilidad.

Para lograr este objetivo, fue necesario cumplir con otros específicos. Primero, se tuvo que definir las entidades a emplear según los elementos que se necesita que aparezcan en una receta. Además, se construyeron modelos que recibiesen un texto correspondiente a una receta médica y que entregasen el mismo con cada token etiquetado con alguna de las entidades definidas. De ahí, a partir de los resultados, conocer cuál clasifica mejor. Esto último no necesariamente significa que el modelo sea mejor y más adecuado para lo que se necesita en el MINSAL, por lo que fue importante reunirse con profesionales para tener un mejor entendimiento del problema y de sus necesidades para así lograr una solución adecuada.

Es importante hacer énfasis en que, en este proyecto, lo que se está haciendo es reconocer la presencia de ciertos datos en recetas médicas. No se está revisando la correctitud de las recetas en cuanto a, por ejemplo, la dosis o la duración del tratamiento. También hay que considerar que se está trabajando con datos que poseen cierta estructura, todos provenientes del Hospital Clínico Dra. Eloísa Díaz I. de La Florida, cuyo formato puede no ser necesariamente igual al de otros hospitales. Por último, notar que dentro de los datos pueden existir errores de escritura. Aún con estos, el modelo debería ser capaz realizar la detección correctamente.

2. Metodología

2.1. Planificación

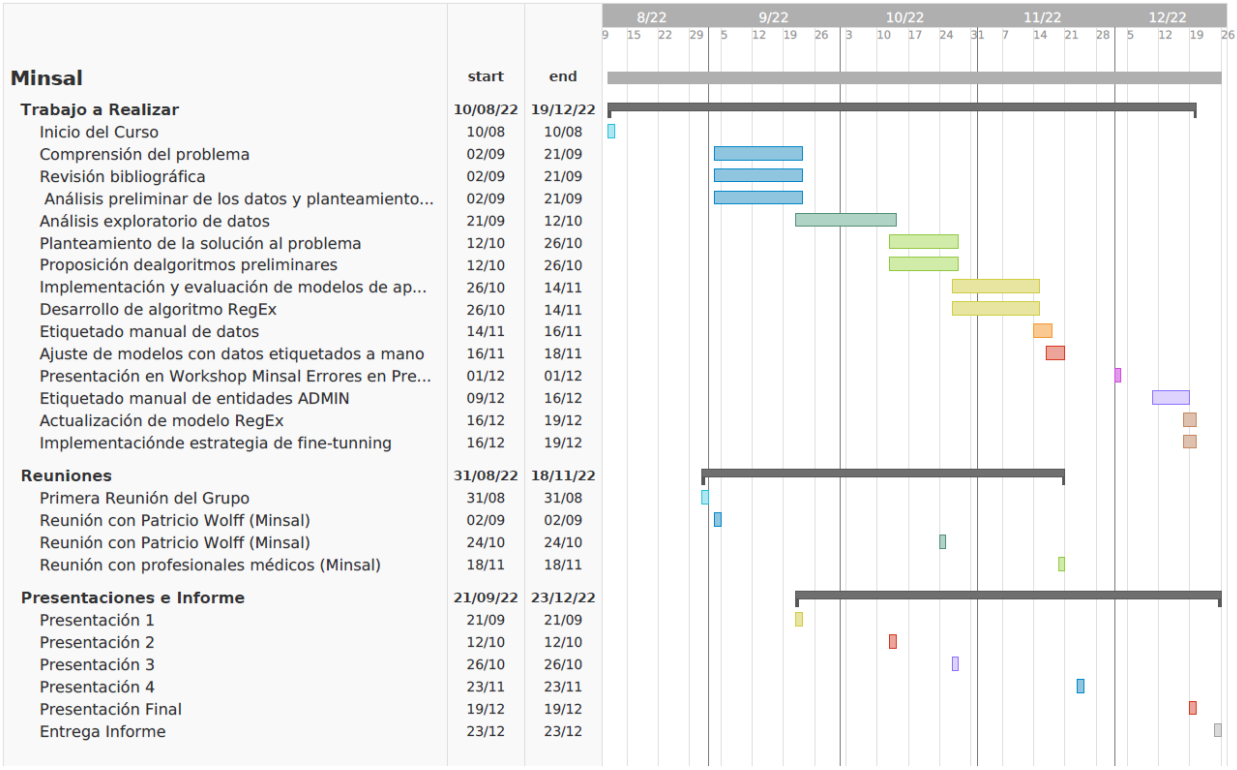
El reconocimiento de entidades en texto libre es una tarea de procesamiento de lenguaje natural ampliamente explorada. Es por esto que nuestra planificación de trabajo está fuertemente basada en la utilización de modelos existentes. La gran desventaja que se presenta en nuestro caso es el hecho de que no poseemos un dataset con el típico formato que presentan estas tareas. Dado lo anterior, las etapas más importantes del proyecto están ligadas a como llevaremos a cabo el tratamiento de datos, que a su vez estará ligada a los descubrimientos durante la fase de exploración de datos (ver sección 3 para esto). Otro hito importante es la implementación de modelos, entre los cuales estarán modelos conocidos en la resolución del problema. Se contempló la utilización de los siguientes frameworks:

- ***pytorch* [1]:** biblioteca para la creación y entrenamiento de redes neuronales en python.
- ***transformers* [2]:** esta biblioteca tiene por una parte un vasto repositorio de modelos open source. Por lo demás, provee herramientas para su ajuste de manera sencilla.
- ***seqeval* [3]:** consiste en un framework de evaluación, especializado en tareas de etiquetamiento secuencial (como es el caso de la detección de entidades).
- **Bibliotecas de visualización:** se emplearon diversas bibliotecas que permiten visualizar datos en *python* tales como: *plotly*, *matplotlib* y *seaborn*. Más aún, se utilizó *wordcloud* para obtener nubes visualizables de palabras, además de *umap* para visualizar en embeddings en dos dimensiones.
- **Otras dependencias:** se dispuso de *nltk* y *torchtext* para tareas menores de procesamiento de texto. Además, se usó bibliotecas clásicas en tratamiento de datos como *pandas* y *numpy*.

Estos han sido utilizados por los miembros del equipo en distintos contextos, lo cual facilita el trabajo a realizar. También se tomarán en cuenta métodos que se basen en reglas y se idearán maneras de mezclar ambos tipos de resolución de ser esto pertinente. Por último, las reuniones con expertos también marcarán los pasos a seguir en caso de necesitar hacer un giro en la toma de decisiones.

2.2. Plan de Trabajo

A continuación se muestran los puntos relevantes del desarrollo del proyecto, además de las presentaciones del curso y reuniones realizadas.



3. EDA y procesamiento

En el proceso de exploración de datos, se buscó analizar y conocer los datos a través de diversas visualizaciones, con el fin de familiarizarse con la data con la data que fue proporcionada, la cual consiste en entradas de recetas médicas electrónicas, donde el total de estas ascendía por sobre el millón y medio de entradas.

En un principio, se evaluó la composición, el número de entradas de datos, la cantidad de columnas, como así también la cantidad de valores nulos y cómo estos se concentraban en los diversos atributos. Un reporte de la constitución de los datos se puede ver a continuación. en la Figura 1.

Asimismo, es posible observar la distribución de los valores nulos dentro de los datos como se muestra en la Figura 2, donde los atributos con mayor cantidad de datos faltantes corresponden a las columnas IND_ADMINISTRACION 1 y 2. Sin embargo, el uso de estas columnas es de carácter especial, por lo que no significa un problema a la hora de reconocer entidades.

```
RangeIndex: 1526557 entries, 0 to 1526556
Data columns (total 21 columns):
 #   Column                Non-Null Count    Dtype
---  ------                --------------    -----
 0   index                 1526557 non-null  int64
 1   PRES_FECHA            1526556 non-null  object
 2   CODIGO                1526557 non-null  object
 3   PATIENT_ID            1526557 non-null  int64
 4   LINEA                 1526557 non-null  object
 5   EPISCODE              1526557 non-null  object
 6   CODESPECIALIDAD       1526556 non-null  object
 7   ESPECIALIDAD          1526556 non-null  object
 8   SALA                  1526508 non-null  object
 9   CAMA                  1526556 non-null  object
 10  AREA_KEY              1526556 non-null  object
 11  DIAGCODE              687787 non-null   object
 12  DIAGDESC              1526556 non-null  object
 13  CODIGO_MEDICAMENTO    1508178 non-null  object
 14  ESTADO_PRESCR         1526556 non-null  object
 15  PRES_DENOMINACION     1489517 non-null  object
 16  RESUMEN               1526556 non-null  object
 17  FORMA_FARMA           1426243 non-null  object
 18  IND_ADMINISTRACION_1  31525 non-null    object
 19  IND_ADMINISTRACION_2  363678 non-null   object
 20  MEDICO_ID             1485986 non-null  float64
dtypes: float64(1), int64(2), object(18)
```

Figura 1: Reporte de Atributos HLF dataframe.

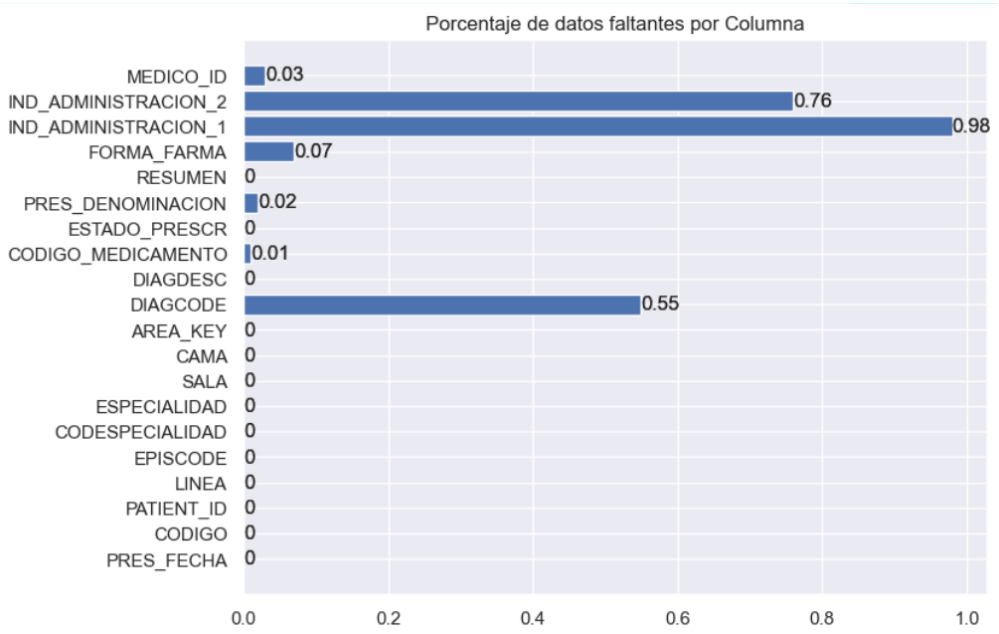


Figura 2: Porcentaje de atributos nulos por atributo.

Luego de este análisis general de la constitución del conjunto de datos, se determinaron los atributos relevantes, para lograr el objetivo de detección de entidades, los atributos principales utilizados para el trabajo de modelamiento posterior fueron las columnas de texto libre PRES_DENOMINACION

y RESUMEN, las cuales constaban de la mayor cantidad de información relevante sobre la prescripción de medicamentos, así mismo, otras columnas que fueron de interés, ya que contaban con información que hacía alusión a partes del texto libre, fueron CODIGO_MEDICAMENTO y FORMA_FARMA, las cuales referían a principios activos y formas farmacológicas respectivamente.

Un último preprocesamiento ejecutado anterior a la obtención de visualizaciones de los datos contenidos en los atributos previamente referidos, fue la obtención de nombre de principios activos asociados a los códigos de medicamentos. Así, se obtuvo un nuevo conjunto de datos, que refiere a principios activos junto al código asociado en el hospital la florida, así fue posible obtener una nueva columna a integrar en el conjunto de datos original, dando pie así a un nuevo atributo denominado PRINCIPIO_ACTIVO.

Luego de la adición de este último atributo y la eliminación de valores nulos en los atributos de interés, fue posible obtener visualizaciones de los datos, donde uno de los objetivos de esto fue observar el efecto de la pandemia de sars-cov-2 en los datos médicos registrados en la base de dato. Así, una primera visualización fue sobre las patologías asociadas para cada conjunto de atributos, la cual se expone a continuación.

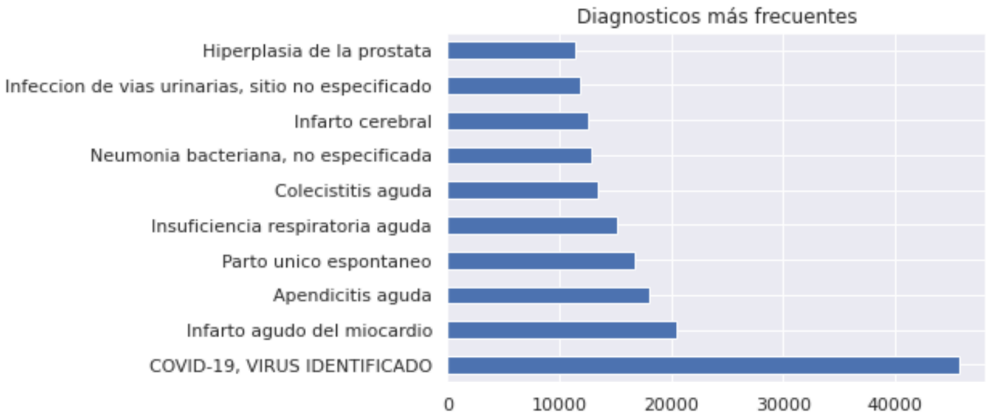
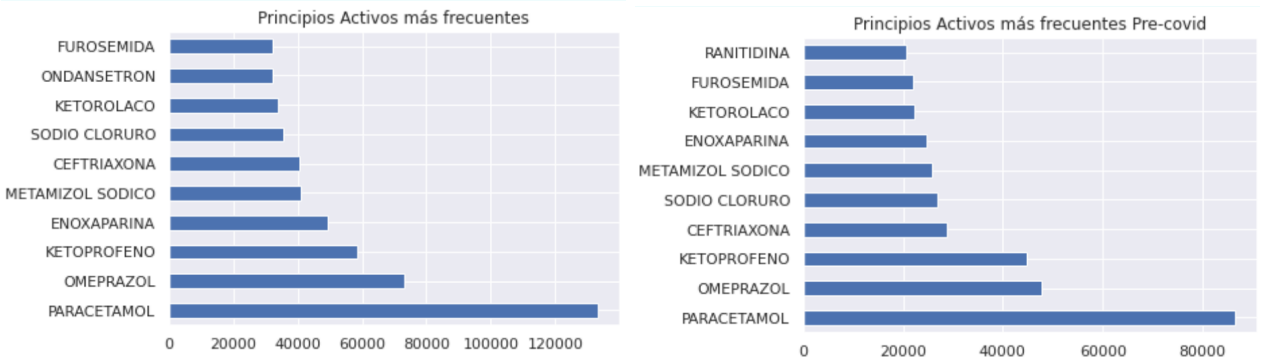


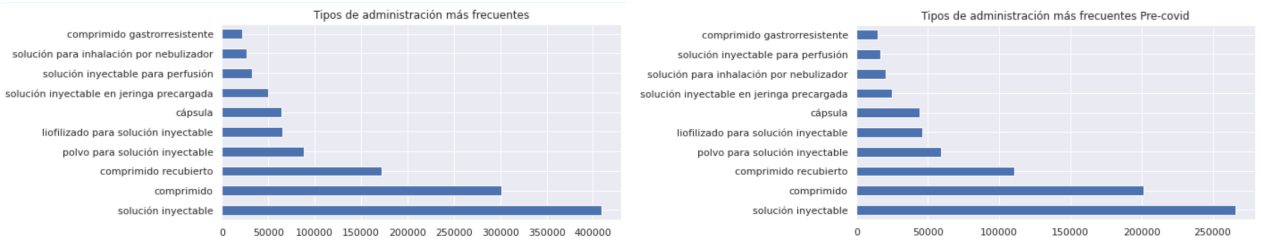
Figura 3: Diagnósticos mas frecuentes en el dataframe

Dado a que la gran mayoría de casos se deben a la patología antes mencionada, se quiso explorar si este fenómeno sesgaba los datos de alguna manera, lo que se procedió a ejecutar una visualización Pre Covid, considerando el periodo diciembre 2019 hacia atrás, y una con toda la data disponible para los conjuntos de Forma Farma y principios activos, estas visualizaciones se exponen a continuación.



(a) (b)

Figura 4: (a) Principios Activos mas frecuentes (b) Principios Activos mas frecuentes Pre-covid



(a) (b)

Figura 5: (a) Formas Farmacológicas mas frecuentes (b) Formas Farmacológicas mas frecuentes Pre-covid

En general, se puede observar que tanto los principios activos como las formas farmacológicas no varían de forma significativa que puedan llevar a afectar la ejecución de modelos, por lo que se decidió utilizar toda la data disponible para la implementación de modelos.

Posterior a esto, se buscó ejecutar un análisis cualitativo de la constitución de los atributos de lenguaje natural a utilizar en el modelamiento del proyecto, para esto se generaron wordclouds para el conjunto Resumen de los atributos del dataframe, a continuación se expone la visualización:

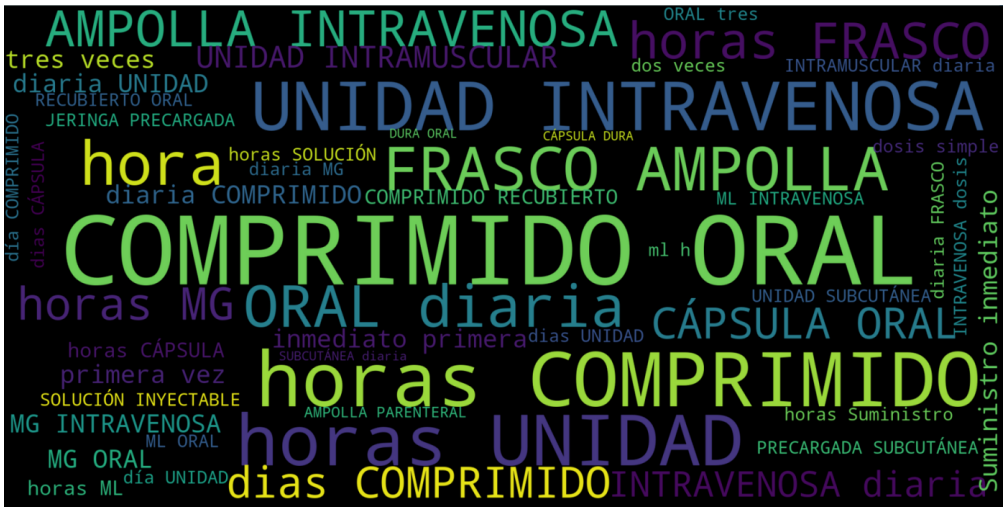


Figura 6: WordCloud de atributo Resumen

A primera vista, es posible ver que gran cantidad de las palabras expuestas hace referencia a unidades de medida, vías de administración y periodos de tiempo, lo que permite dimensionar que en general, es posible encontrar entidades que refieren a periodos, duraciones, además de cantidades y formas de administración. Mientras que para el caso de PRES_DENOMINACION, que no posee una word cloud asociada, posee estructuras del tipo PRINCIPIO ACTIVO + CANTIDAD + FORMA FARMACOLÓGICA.

En general, gracias al análisis exploratorio de datos, fue posible obtener las siguientes conclusiones que fueron consideradas para la definición de algoritmos y modelos en las secciones posteriores. El efecto covid no parece alterar de manera significativa los atributos de interés en el dataset, por lo que no se limitará el dataset utilizado. Además, en general los atributos de texto libre de resumen, poseen estructuras periódicas definidas y hacen referencias a vías de administración y determinan periodicidad de forma "cada x min/horas/días" y duración de forma "cada x horas/días", mientras que para en PRES_DENOMINACION se hacen referencias a las columnas de PRINCIPIO_ACTIVO y FORMA_FARMA, por lo que en general en cada una de las recetas se encuentran 5 entidades principales.

4. Resolución del problema

Con el objetivo de resolver el problema de "detección de errores de medicación" con un acercamiento con NLP, se plantea ejecutar algoritmos centrados en la tarea NER o named entity recognition, lo que refiere a determinar entidades y reconocerlas a través de modelos clasificadores.

Se definen el corpus o inputs de los modelos como una unión entre las ya mencionadas columnas PRES_DENOMINACION y RESUMEN, donde a través de la concatenación y posterior eliminación de elementos repetidos, es posible determinar 108.049 ejemplos únicos que siguen una estructura como la del siguiente ejemplo: "HIDRALAZINA 50 MG COMPRIMIDO 13 MG ORAL cada 12 horas durante 15 días".

Luego de haber sido determinado el objetivo y el corpus a utilizar en los modelos, es necesario definir cuáles son las clases de salida de cualquier modelo implementado, donde las entidades que representan de mejor manera cada ejemplo son ACTIVE_PRINCIPLE, correspondiente al principio activo de la receta; FORMA_FARMA, que corresponde a la forma farmacológica del medicamento; ADMIN corresponde a una combinación de cantidades, unidades de medida y vía de administración del medicamento de la receta; PERIODICITY, correspondiente al periodo con el que se administra el medicamento; DURATION, qué corresponde a la cantidad de tiempo en el cual se debe administrar el medicamento.

Por último, es importante señalar que para los modelos se utilizó el formato CoNLL para los conjuntos de entrada a los diversos modelos.

4.1. Etiquetado manual de datos

Para el etiquetado manual de datos, se utilizó Label Studio [4]. Esta es una herramienta que permite etiquetar distintos tipos de datos como texto, imágenes y audio. Para llevar a cabo el proceso, se creó un proyecto nuevo, se subió el archivo .txt con todos los ejemplos, se configuró el proyecto para la task de reconocimiento de entidades y se definieron estas últimas. De ahí, en cada ejemplo, se hizo click en alguna entidad y se seleccionó el texto correspondiente a ella. Este paso se repite para todas las entidades con que se correspondan con al menos un token de la secuencia. Una vez etiquetadas todas las recetas, se exportó el dataset como un archivo de formato CONLL2003.

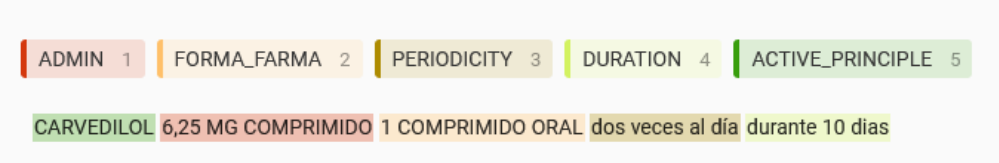


Figura 7: Un ejemplo de receta médica etiquetado en la interfaz gráfica de Label Studio

4.2. Modelos

A continuación, se exponen los modelos utilizados para la tarea de reconocimiento de entidades para el corpus de recetas definido.

4.2.1. Expresiones regulares (RegEx)

El primer modelo surge de la detección de patrones muy definidos en los atributos de texto libre presentes en los atributos de texto libre presentes en el dataframe, donde para las 5 etiquetas ACTIVE_PRINCIPLE, FORMA_FARMA, ADMIN, PERIODICITY y DURATION se determinaron una serie de reglas estructurales que, cuando están presentes en el texto, la entidad se encuentra en esa posición.

De manera más detallada, para las entidades ACTIVE_PRINCIPLE, FORMA_FARMA y ADMIN, se definieron conjuntos de palabras claves que denotan la existencia de la entidad donde, por ejemplo, se creó un diccionario de palabras desde la columna FORMA_FARMA del dataframe y se asoció la existencia de cualquiera de los conjuntos de palabras existentes en el diccionario con alguna

sección del ejemplo del corpus a etiquetar, definiendo así la ubicación de la entidad en el texto, esto se hizo recíprocamente para ACTIVE_PRINCIPLE con la columna PRINCIPIO_ACTIVO y para ADMIN con un conjunto de vías de administración que se pueden ver a más detalle en el código de RegEx en el repositorio.

Así mismo, para las entidades PERIODICITY y DURATION, se definieron patrones asociados a estas 2, para PERIODICITY la palabra clave de "cada" señala el inicio de la entidad y para DURATION la palabra "DURANTE" señala el inicio de la entidad periodo. Así mismo, se definieron algunas excepciones comunes, por ejemplo "Diariamente" también es una palabra clave de PERIODICITY, esto es visible más a fondo en el código de RegEx en el repositorio.

A continuación se expone el formato en el cual Regex etiqueta un input de receta:

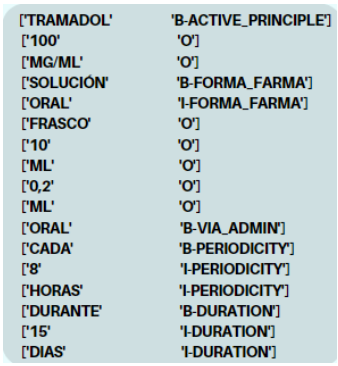


Figura 8: Ejemplo de la salida del modelo RegEX.

Finalmente, una segunda versión de RegEx fue implementada posterior a la obtención de los primeros resultados, donde para las entidades ACTIVE_PRINCIPLE y FORMA_FARMA se utilizó el atributos PRINCIPIO_ACTIVO y FORMA_FARMA asociados al atributo de lenguaje natural utilizado como entrada al modelo.

4.2.2. Redes recurrentes

Las redes recurrentes tienen como característica principal la capacidad de "recordar" estados previos mediante el uso de bucles en el diagrama de red, como se logra observar en la Figura 9 donde se tiene una red recurrente de una sola unidad, que al ser desplegada se logra observar que para una misma entrada, esta cuenta con múltiples salidas, donde la predicción o salida del próximo termino se encuentra determinada por la salida del estado oculto anterior h_t. Sin embargo, este método de recurrencia es útil para recordar secuencias cortas o a corto plazo, por lo que para recordar grandes secuencias de datos, o expresiones como este caso, es mejor utilizar un tipo de red recurrente más complejo, en este caso se utilizaron redes LSTM (Long Short Term Memory).

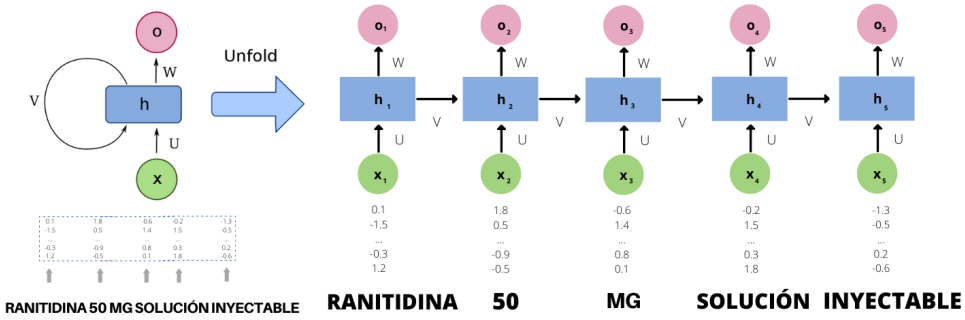


Figura 9: Idealización de una red recurrente procesando una prescripción.

Las redes recurrentes LSTM, además de contar con el estado oculto h_t, cuentan con un estado oculto adicional c_t (Figura 10), que representa la memoria a largo plazo, mientas que h_t la memoria corta. Dentro de la red, los estados ocultos pasan por 4 puertas, las cuales permiten controlar la información que se conserva y transmite a través del tiempo.

- **Forget Gate:** permite eliminar la información que ya no es considerada útil en el aprendizaje.
- **Store Gate:** almacena la información nueva.
- **Update Gate:** actualiza la información que vamos a dar a la RNN con el resultado de la forget gate y la store gate.
- **Output Gate:** entrega la entrada para el siguiente estado oculto, es decir y_t y h_t.

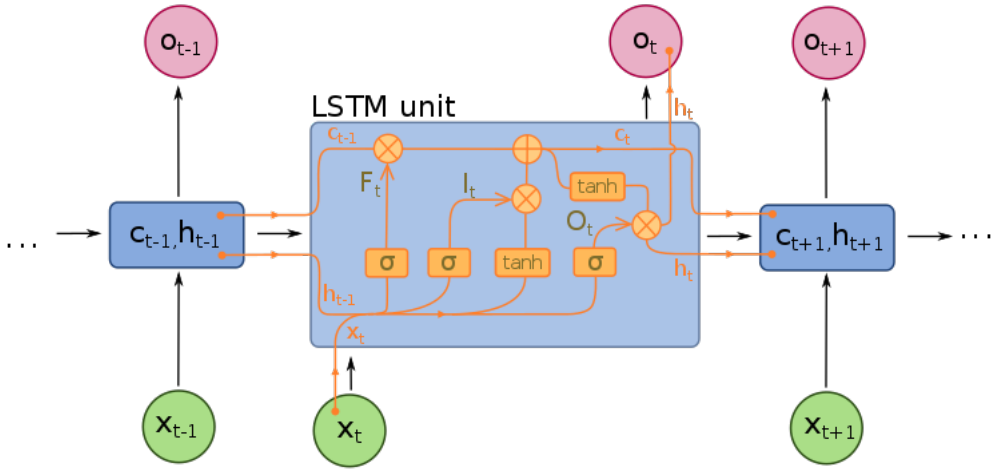


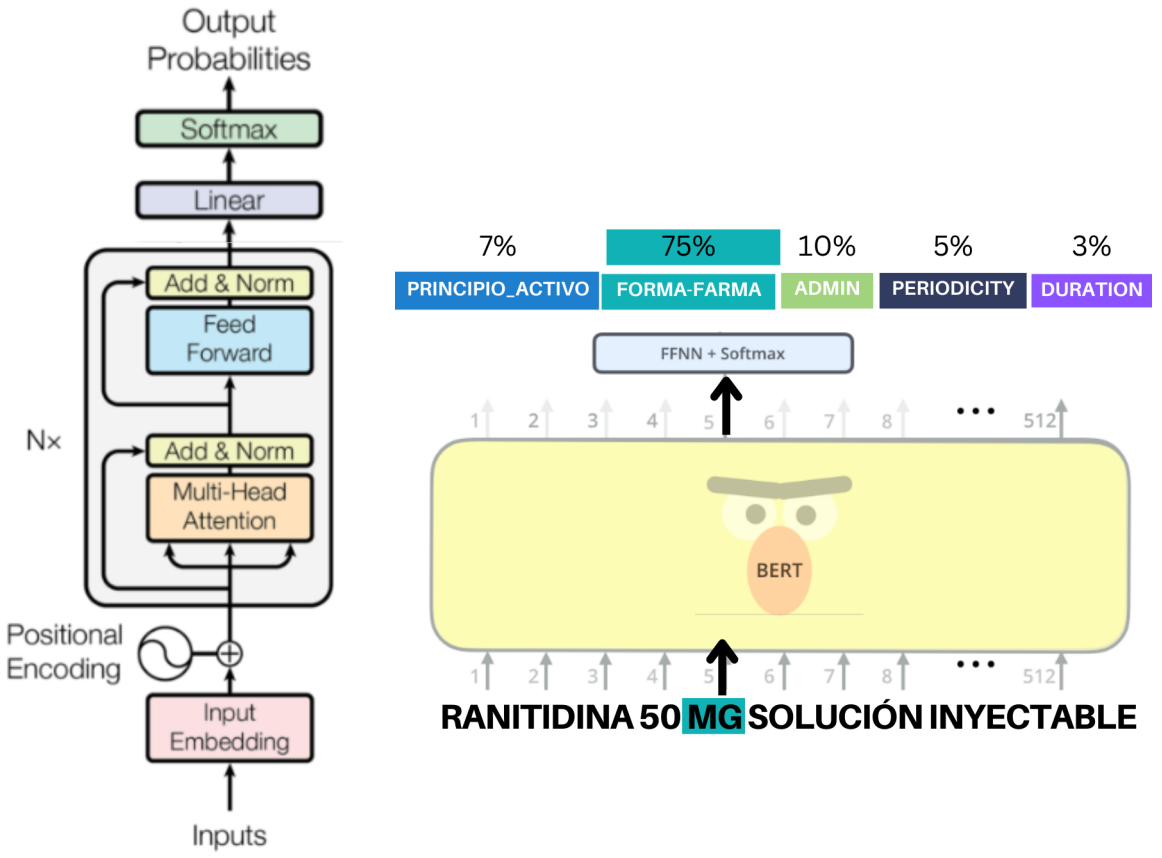
Figura 10: Arquitectura de red recurrente LSTM.

De esta manera, el segundo modelo utilizado corresponde a una Red Neuronal Recurrente, la cual cuenta con una capa de *embedding*, 3 capas *LSTM* de 256 unidades cada una, y finalmente una *capa Lineal* de dimensiones igual al numero de entidades que debe ser capaz de identificar.

4.2.3. BETO (transformers)

Como tercera arquitectura a considerar se utilizó un modelo basado en la arquitectura *transformer* [5], cuya parte relevante se muestra en figura 11. Más específicamente, se tomó el modelo BETO [6]. Este modelo de lenguaje está basado en BERT [7], pero pre-entrenado en un gran corpus en español. A su vez, a este modelo se le realizó fine-tunning en datos médicos [1], donde la tarea en cuestión tanto para esto como para el pre-entrenamiento fue la predicción de tokens faltantes en una secuencia (*masked-language modelling*). El dataset clínico en cuestión corresponde al corpus chileno de listas de espera [8], cuyo trabajo lo realizó el grupo de procesamiento de lenguaje natural del Centro de Modelamiento Matemático de la Universidad de Chile.

El modelo fue configurado para resolver la tarea de detección de entidades, ya que toma cada vector contextualizado (i.e., que depende del resto de los tokens de la secuencia), propio de cada token y lo transforma en probabilidades de pertenecer a cada clase (entidad posible).



(a) Encoder de una red transformer (b) Ejemplo de funcionamiento del modelo

Figura 11: Beto para prescripciones médicas.

[1] El modelo es accesible en el repositorio transformers de HuggingFace a través del enlace: https://huggingface.co/plncmm/bert-clinical-scratch-wl-es.

Estos modelos tienen la ventaja de codificar estructuras de sintaxis más finas a través de su arquitectura basada en módulos de atención. Además el pre-entrenamiento resulta en que este tipo de modelos sea, en general, más robusto a datos nuevos o de naturaleza nueva. Sin embargo, su complejidad es bastante grande, lo cual hay que tener en cuenta a la hora de escogerla sobre otras opciones, sobretodo cuando sus ganancias no justifiquen tal costo.

4.2.4. Primeros resultados

En figura 12 mostramos la comparación de los tres modelos mencionados anteriormente. Es interesante notar la alta precisión del modelo de RegEx, al poseer 94 % de *accuracy*. Por el contrario su sensibilidad es más bien baja (0.48 de *recall*). Esto nos sugiere que el modelo RegEx no suele equivocarse al etiquetar, sin embargo existen varios casos en los cuales no está identificando la entidad en cuestión. Esto es el comportamiento esperado para tal tipo de algoritmo. Como nosotros diseñamos las reglas usando expresiones regulares, es natural que se tenga confianza en ellas pero que estas no abarquen todos los casos posibles de cada entidad.

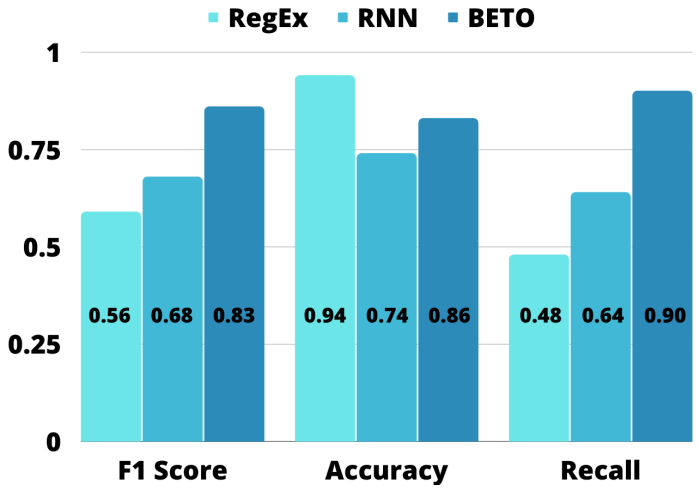


Figura 12: Resultados para los tres modelos ajustados en 1000 datos

Por otro lado, los modelos basados en aprendizaje de máquinas muestran una buena capacidad de generalización, por lo cual presentan una sensibilidad considerablemente más alta. No obstante, tienden a hacer "apuestas" erradas, con lo cual su precisión disminuye respecto al modelo RegEx. Esto es también parte del comportamiento lógico de estos modelos, en especial el modelo BETO, que cuenta con un proceso de pre-entrenamiento que se espera le agregue robustez. Nos gustaría sin embargo combinar los poderes predictivos de ambos tipos de modelos. A continuación hablaremos de como proponemos lograr aquello.

4.3. Estrategia de fine-tunning

Los resultados obtenidos en la sección 4.2.4 fueron obtenidos a partir de los 1000 datos etiquetados manualmente, alrededor del 1 % del total de datos únicos con los que cuenta la base de datos, por lo que se esperaba que al utilizar una mayor cantidad de datos en el entrenamiento de ambos modelos, los resultados obtenidos en estas métricas de evaluación, mejoraran considerablemente.

De esta manera, se decidió utilizar el modelo RegEx sobre los 100 [K] datos sin etiquetar, para generar un conjunto de entrenamiento y prueba (de proporciones 80 y 20 %). De esta manera, tanto el modelo Beto como el modelo RNN, fueron entrenados con una gran cantidad de datos que se encontraban relativamente bien etiquetados por el modelo RegEx, para posteriormente realizar un afinamiento de ambos modelos con los 1000 datos etiquetados manualmente.

Una vez entrenado y afinados los modelos, se evaluaron nuevamente con el conjunto de Prueba etiquetado manualmente, obteniendo los resultados de la Figura 13 para el caso de la RNN y la Figura 14 para el caso de Beto. En ambos casos se logra alcanzar porcentajes de precisión altos como al utilizar el modelo RegEx, pero conservando y en este caso mejorando la sensibilidad con que cuentan los modelos basados en aprendizaje de maquina en las métricas F1 y Recall.

Cabe destacar que para ambos modelos, el afinamiento de estos se realizó aproximadamente con un 1 % de los datos, por lo que se espera que al entrenar tanto el modelo Beto como la Red Neuronal con un 5 o 10 % de los datos etiquetados manualmente, los resultados anteriores mejorasen aún más con respecto a las métricas F1 score, Accuracy y Recall.

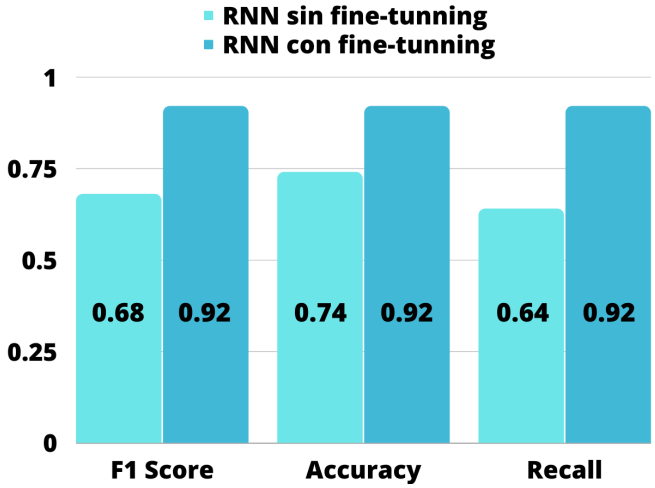


Figura 13: Resultados para modelo RNN luego de estrategia fine-tunning

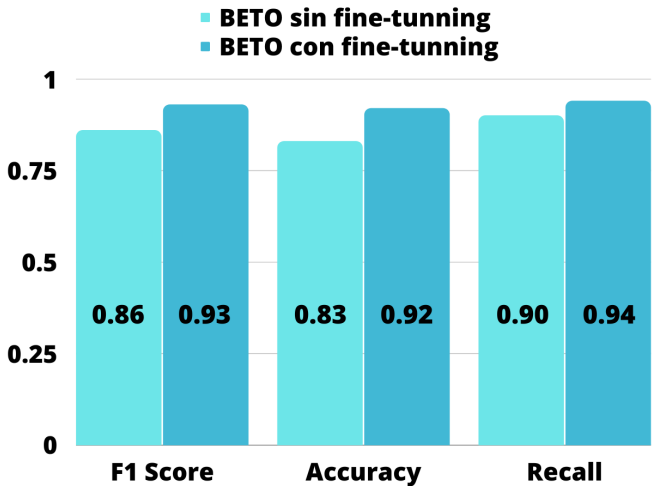


Figura 14: Resultados para modelo BETO luego de estrategia fine-tunning

4.4. Etiqueta de tokens ADMIN

Tras conversar con expertos del área de la salud, se llegó a la conclusión de que era necesario tener una mayor precisión en el reconocimiento de entidades dentro de una prescripción médica, por lo que se decidió dividir la entidad ADMIN en tres sub-entidades: CANT correspondiente a la cantidad o dosis del principio activo a prescribir, UND que hace referencia a la unidad utilizada para especificar la dosis y por último, VIA_ADMIN que corresponde a la vía de administración del principio activo. Sin embargo, agregar una mayor cantidad de entidades disminuiría el rendimiento general de los modelos ya entrenados, por lo que se decidió entrenar un nuevo modelo que recibiera los campos de textos identificados como la entidad ADMIN, para luego identificar las tres sub-entidades antes mencionadas, tal como se muestra en el ejemplo de la Figura 15.

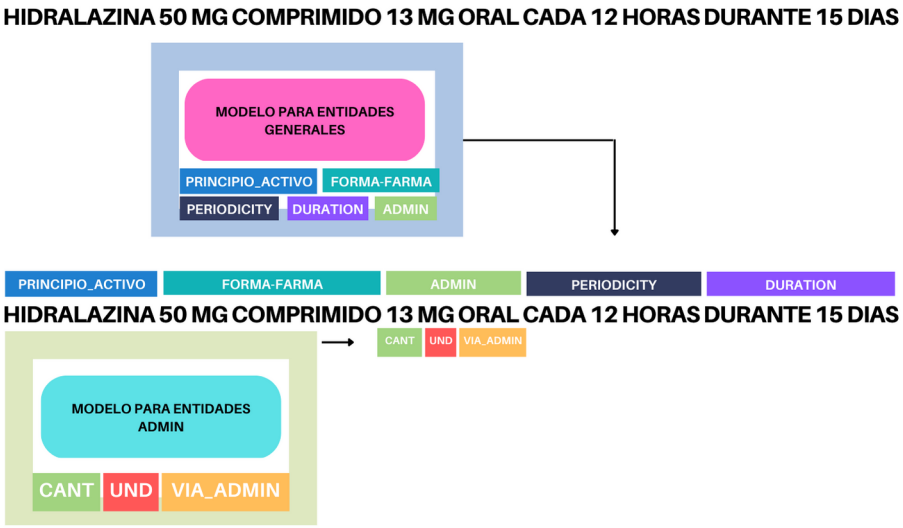


Figura 15: Ejemplo de funcionamiento de un modelo conjunto usando detección encadenada de entidades

Con el objetivo de entrenar nuevos modelos enfocados al reconocimiento de la entidades CANT, UND y VIA_ADMIN, se extrajo solo el texto relacionado a la entidad ADMIN del conjunto de 1000 datos etiquetados manualmente, para luego etiquetar nuevamente a mano estos datos con las nuevas etiquetas.

Los resultados obtenidos por ambos modelos se encuentran definidos en la Figura 16, donde se logra observar que ambos modelos obtuvieron resultados bastante similares a la hora de predecir las nuevas etiquetas.

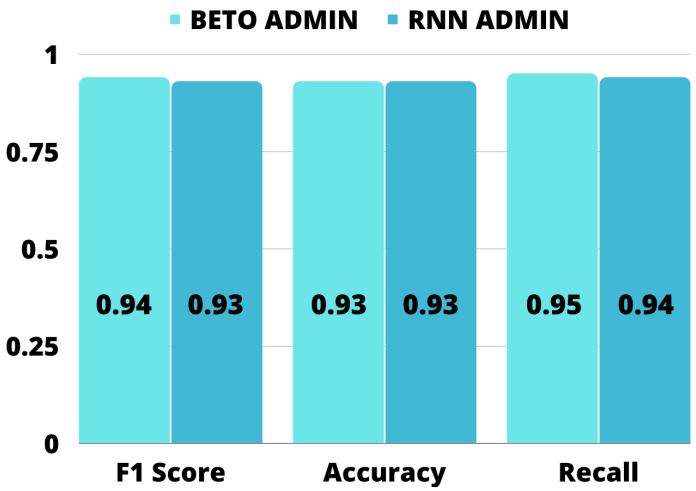


Figura 16: Resultados de reconocimiento de entidades finas para tokens ADMIN

4.5. Contribución

El objetivo general, y en el cual se enmarca el FONDEF del ministerio de Salud, es para la detección de errores de medicación, y desde el enfoque de procesamiento de lenguaje natural utilizado en el proyecto, los resultados obtenidos de este y los modelos entrenados poseen 2 grandes utilidades que pueden dar pie a implementaciones más profundas para detección de incompletitud, incongruencias lingüísticas o detección de outliers en medicación.

En el estado actual, es posible definir que los modelos con mejores métricas son capaces de detectar la mayoría de las 7 entidades determinadas para el proyecto, lo que da pie a que se pueden generar algoritmos de verificación de completitud, que podrían detectar errores de medicación asociados a la inexistencia de información fundamental para la medicación de un paciente.

Así mismo, el estado actual del proyecto, permite obtener datos que aportan una mayor cantidad de atributos significativos a cada uno de los conjuntos de atributos iniciales en el dataframe, es decir que es posible detectar atributos como, periodo, duración, vía de administración, cantidades y unidades de medida, que junto a datos fundamentales como principio activo, pueden dar pie a que se pueda obtener un sistema de alertas a tiempo real al crear una nueva receta, donde un sistema como el que se ve a continuación es una aplicación importante donde el proyecto puede ser un aporte significativo.

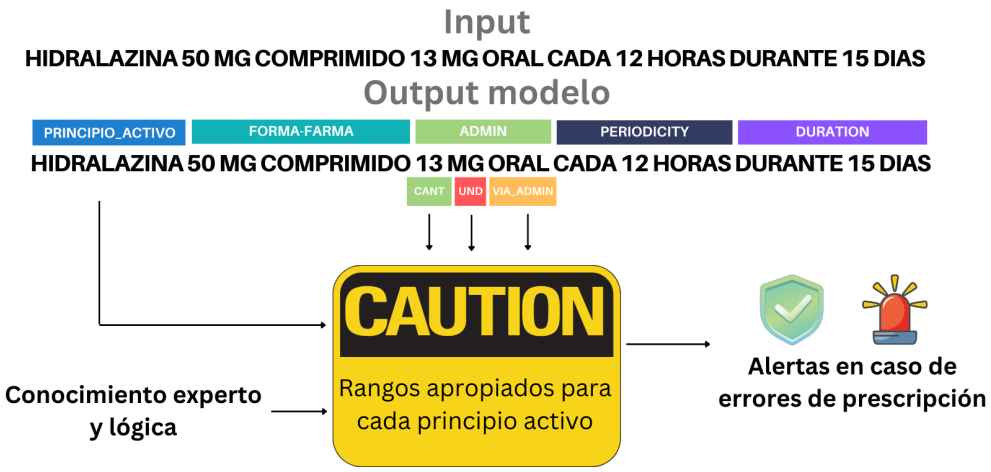


Figura 17: Ejemplo de detección de errores de dosis usando reconocimiento de entidades.

Además, una aplicación fuera del ámbito de detección de errores de medicación, es la implementación como algoritmo para estandarizar datos de hospitales, donde a través de recetas determinadas directamente desde lenguaje natural, sea posible transformar cualquier texto ingresado a un formato estándar que sería comparable entre distintos centros médicos, esto permitirá obtener métricas e información importante al comprar gran cantidad de datos desde orígenes diversos.

4.5.1. Demostración y Repositorio

El trabajo señalado se encuentra integramente en nuestro repositio Github, accesible a través del siguiente enlace: https://github.com/camilocarvajalreyes/entidades-minsal
Este repositorio cuenta con los scripts y notebooks con los cuales se entrenaron y testearon nuestros modelos. Además, incluimos en él los archivos de nuestras presentaciones y otros elementos relevantes. Tiene también notebooks de ejemplo, los cuales usan modelos entregables descargables.

Además hemos implementado una demostración en la plataforma HuggingFace Spaces. Este demo utiliza los modelos basados en *transformers* y etiqueta los tokens correspondientes. De este modo, se emula un ambiente de despliegue del modelo.
Enlace al demo: https://huggingface.co/spaces/ccarvajal/entidades-prescripciones

5. Conclusiones y trabajo futuro

Consideramos que nuestro proyecto es exitoso ya que logra el objetivo principal en lo respectivo a la detección de entidades. El trabajo además presenta una metodología interesante que aprovecha prácticas como el *transfer-learning* para combinar conocimiento experto con algoritmos de aprendizaje profundo. Los resultados de nuestro procedimiento son muy buenos, al punto de generar un interés del punto de vista técnico. Desde luego, la calidad de las métricas no es un éxito en si, pero si lo es el haberle entregado a la contraparte herramientas confiables con fuerte potencial de uso.

Sin embargo, es claro que pueden haber mejoras en varios aspectos. En primer lugar, hay diversos análisis que podrían llevarse a cabo dado nuestros modelos. Un tal caso es procesar grandes cantidades

de datos con nuestros ellos y hacer una exploración para ver que elementos relevantes se desprenden de esto, los cuales nos pueden guiar acerca del tipo de errores o malas prácticas se cometen al escribir recetas médicas. Además, podemos ver que tipo de errores comete nuestro algoritmo para así invertir tiempo en mejoras y advertir tipos de riesgo que presentaría su uso en un contexto de aplicación real.

En caso de que se requiera mayor seguridad en las predicciones, se puede conectar el modelo con un algoritmo basado en reglas similar al usado en el modelo de expresiones regulares. Lo que se realizó en este trabajo fue usar ese resultado como entrenamiento, pero se puede ampliar esa conexión entre los modelos de aprendizaje y el conocimiento experto confiable. Más aún, se pueden probar otros algoritmos que han dado buenos resultados para la predicción de entidades en contextos médicos, como lo es el caso de Clinical FLAIR [9], producido en el CMM para datos de el corpus chilieno de lista de espera [8].

Además de aquello, el procedimiento de refinamiento de etiquetas explicado en la sección 4.4 puede ser replicado para otras porciones de texto. Podemos, por ejemplo, extraer las cantidades temporales expresadas dentro de las entidades generales PERIODICITY y DURATION. Esto a su vez puede ser usado en un sistema como el de la figura 17. Como ciertas cantidades de mediación pueden o no ser apropiadas dependiendo de su frecuencia y duración de consumo, tal información haría que el sistema sea más robusto.

Los elementos mencionados anteriormente de todos modos muestran el éxito de nuestro proyecto. Globalmente se cumplió la misión a desarrollar y se hizo con soluciones suficientemente confiables como para proponer su uso en aplicaciones complementarias. Estas tienen la capacidad de ser útiles para los profesionales médicos que escriben las prescripciones, así como para el Ministerio de Salud en su rol de arbitrador y de resguardo de la salud del país. Este posee, como resultado de este proyecto, una herramienta más, la cual puede salvar vidas y tiempo si se utiliza de forma adecuada.

Referencias

[1] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., *et al.*, "Pytorch: An imperative style, high-performance deep learning library," Advances in neural information processing systems, vol. 32, 2019.

[2] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., y Rush, A. M., "Transformers: State-of-the-art natural language processing," en Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, (Online), pp. 38–45, Association for Computational Linguistics, 2020, https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[3] Nakayama, H., "seqeval: A python framework for sequence labeling evaluation," 2018, https://github.com/chakki-works/seqeval. Software available from https://github.com/chakki-works/seqeval.

[4] Tkachenko, M., Malyuk, M., Holmanyuk, A., y Liubimov, N., "Label Studio: Data labeling software," 2020-2022, https://github.com/heartexlabs/label-studio. Open source software available from https://github.com/heartexlabs/label-studio.

[5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, $\mathcal{L}$., y Polosukhin, I., "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[6] Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., y Pérez, J., "Spanish pre-trained bert model and evaluation data," en PML4DC at ICLR 2020, 2020.

[7] Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K., "Bert: Pre-training of deep bidirectional transformers for language understanding," en Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), (Minneapolis, Minnesota), p. 4171–4186, Association for Computational Linguistics, 2019, doi:10.18653/v1/N19-1423.

[8] Báez, P., Villena, F., Rojas, M., Durán, M., y Dunstan, J., "The chilean waiting list corpus: a new resource for clinical named entity recognition in spanish," en Proceedings of the 3rd clinical natural language processing workshop, pp. 291–300, 2020.

[9] Rojas, M., Dunstan, J., y Villena, F., "Clinical flair: A pre-trained language model for spanish clinical natural language processing," en Proceedings of the 4th Clinical Natural Language Processing Workshop, (Seattle, WA), p. 87–92, Association for Computational Linguistics, 2022, doi:10.18653/v1/2022.clinicalnlp-1.9.