Informe Técnico - Technical Report
DPTOIA-IT
junio, 2024

Sistema de detección de información mediante visión artificial

Javier Caballero Sandoval

Departamento de Informática y Automática
Universidad de Salamanca

Resumen

En la actualidad, las personas invidentes son capaces de leer, a partir del lenguaje braille incluido en la caja de los medicamentos, una pequeña información sobre los mismos. Además, el acceso al prospecto completo se ve limitado para personas que poseen visión reducida, debido a problemas visuales (miopía, vista cansada, entre otros) por el pequeño tamaño de su letra. A través de esta investigación, se trata de solucionar esta problemática, mediante un modelo de visión artificial que pueda identificar diferentes cajas de medicamentos desde el propio dispositivo móvil. Además, que utilizando Procesamiento del Lenguaje Natural (PLN) se pueda generar una respuesta a cualquier pregunta del usuario acerca del medicamento, teniendo en cuenta el contexto del prospecto.

Para lograr este objetivo, se ha creado un *dataset* de cajas de medicamentos españoles. Posteriormente, se han entrenado dos modelos pre-entrenados de clasificación (MobileNetV2 y ResNet50) y otros dos de detección para (EfficientDet-Lite0 y EfficientDet-Lite2) capaces de identificar la caja del medicamento a través de las imágenes proporcionadas por el dispositivo móvil del usuario en tiempo real. Por otro lado, se trata de obtener un modelo de PLN que tenga la capacidad de generar una respuesta a una pregunta de un usuario. Se han creado otros dos conjuntos de datos con la finalidad de responder a las preguntas del usuario acerca del medicamento. El primer *dataset* tiene como objetivo determinar en qué sección del prospecto se encuentra la respuesta a la pregunta. El segundo tiene la finalidad de generar la respuesta a la pregunta teniendo como contexto la sección elegida en el paso anterior. Se ha realizado *fine-tuning* con los dos conjuntos de datos de un modelo pre-entrenado con texto español basado en la arquitectura BART.

Para finalizar, se ha implementado una aplicación disponible para Android que permite utilizar ambos modelos de *machine learning* obtenidos.

Abstract

Nowadays, blind people are able to read, from the braille language included on the medicine box, a small amount of information about the medicine. In addition, access to the full package patient information is limited for people with reduced vision, due to visual problems (myopia, eyestrain, among others) because of the small size of the print. This research aims to solve this problem by means of an artificial vision model that can identify different boxes of medicines from the mobile device itself. Furthermore, using Natural Language Processing (NLP), it can generate an answer to any question from the user about the medicine, taking into account the context of the patient information leaflet.

To achieve this goal, a dataset of Spanish medicine boxes has been created. Subsequently, two pre-trained classification models (MobileNetV2 and ResNet50) and two detection models (EfficientDet-Lite0 and EfficientDet-Lite2) capable of identifying the drug box through the images provided by the user's mobile device in real time have been trained. On the other hand, the aim is to obtain a PLN model that has the ability to generate an answer to a user's question. Two other datasets have been created in order to answer the user's questions about the medicine. The first dataset is intended to determine in which section of the package leaflet the answer

to the question is found. The second one aims to generate the answer to the question in the context of the section chosen in the previous step. We have performed fine-tuning with the two datasets of a pre-trained model with Spanish text based on the BART architecture.

Finally, an application available for Android has been implemented that allows the use of both models of machine learning obtained.

Índice

Índice de figuras

Índice de cuadros

1. Introducción

En la actualidad, la sociedad española está sufriendo un envejecimiento de su población, como se muestra en la Figura 1 y de manera más acentuada en la comunidad autónoma de Castilla y León [1], debido al aumento de la esperanza de vida [2] y la reducción de natalidad [3]. Como consecuencia, cada vez más porcentaje de la población padece problemas de visión debido a que la vista de los humanos empeora, de manera general, a medida que avanza su edad [4].

Además, en España 1,5 cada 1000 habitantes padece de ceguera legal [5], equivalente alrededor de 71.130 de personas, por ello es importante crear soluciones accesibles a estas personas.

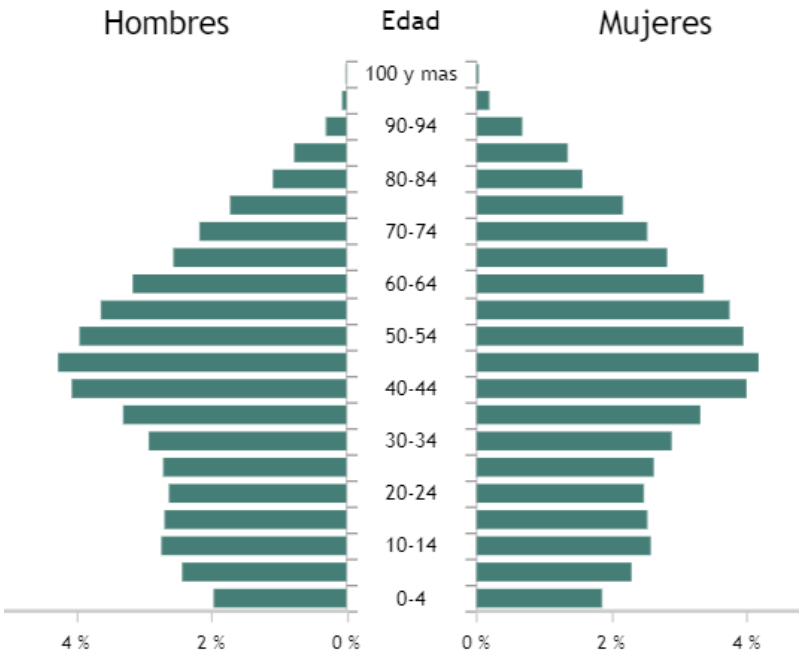


Figura 1: Pirámide Poblacional Española 2022
[6]

Por estos motivos, el primer objetivo de este estudio es desarrollar una solución que resuelve el problema, que poseen las personas con visión reducida, para leer los prospectos de los medicamentos a través del reconocimiento de las cajas de los medicamentos mediante visión artificial. Por otro lado, mediante procesamiento de lenguaje natural (PLN), el posterior tratamiento de las preguntas de los usuarios y la generación de una repuesta precisa y rápida. Por último, el desarrollo de una aplicación fácil de usar para aquellas personas que poseen problemas de visión. Existen diferentes acercamientos a esta solución realizados previamente.

En 2010, un grupo de la Universidad Pontificia de Salamanca desarrolló un sistema que permite obtener información acerca del medicamento a través de la tecnolo-

gía Near Field Communication (NFC) o la cámara del dispositivo. El farmacéutico incluye en la caja del dispositivo una etiqueta identificativa que permite ser personalizable para cada uno de los clientes. El dispositivo móvil lee la etiqueta a través de NFC o la cámara y se descarga un fichero de audio con la información incluida previamente [7].

Por otro lado, en 2014, el Consejo General de Colegios Oficiales de Farmacéuticos, Fundación ONCE y Fundación Vodafone España implementaron una aplicación llamada "Medicamento Accesible Plus"que permite acceder al prospecto del medicamento a través de la lectura del código de barras o *datamatrix* que contiene la caja [8].

Este proyecto trata de mejorar las soluciones anteriores en diferentes aspectos. El primero y más importante es la forma de reconocer el medicamento a través de visón artificial. En este caso se detecta la caja en su totalidad facilitando su uso y abaratando costes, en vez de utilizar el código de barras, que en España es recortado cuando se compra un medicamento en una farmacia, o a través de una etiqueta NFC lo que, debido al gran número de cajas de medicamentos que se producen no es viable económicamente.

Otra ventaja del sistema propuesto, es que no existe la necesidad de un servidor con gran capacidad de almacenamiento junto a una base de datos que contenga la información de todos los medicamentos. Esto es debido a que no se almacenan los prospectos de los medicamentos para proporcionar la información a los usuarios. A través de un modelo de visión artificial que es desplegado en el dispositivo móvil, se utiliza el paradigma *Edge Computing* que consiste en realizar el proceso de gran carga computacional en los dispositivos de borde, en este caso el dispositivo móvil del usuario.

Por otro lado, la generación de las respuestas a las preguntas del usuario se va a llevar a cabo utilizando, en esta primera versión del sistema, un modelo de PLN que ha sido entrenado con los prospectos de 3 medicamentos, aunque el objetivo es que tenga la capacidad de generalizar también para aquellos medicamentos que no han sido entrenados. Para probar el modelo final, se ha diseñado un caso de estudio que detectar un número limitado de medicamentos, en este caso 5, como un primer paso, para que se continúe la investigación, añadiendo más medicamentos.

2. Objetivos

El Trabajo de Fin de Máster realizado tiene diferentes objetivos, en los cuáles destacan los dos principales. El primero consiste en conseguir un modelo de visión artificial capaz de identificar diferentes cajas de medicamentos y este proceso se realice en el propio dispositivo móvil en vez de un servidor central. El segundo trata de obtener un modelo PLN que tenga la capacidad de generar una respuesta a una pregunta de un usuario acerca de un medicamento teniendo como contexto el prospecto del propio medicamento. Además de los objetivos principales, este estudio presenta otros objetivos secundarios:

- Realizar una revisión del estado del arte en el campo de la visión artificial para dispositivos móviles.
- Llevar a cabo una recopilación de los estudios previos acerca de la generación de respuestas a preguntas en el ámbito de la medicina y de los estudios del paradigma *text-to-text* en español.
- Crear un conjunto de datos válido para la identificación de diferentes cajas de medicamentos españoles.
- Crear un conjunto de datos válido para la generación de respuestas a preguntas acerca de medicamentos.
- Comparar los diferentes modelos obtenidos, tanto en el ámbito de reconocimiento de medicamentos como en la generación de las respuestas, y seleccionar el mejor para cada caso de estudio.

3. Estado del arte

3.1. Mapeo sistemático sobre la detección de objetos utilizando como framework TensorFlow Lite

En los últimos años se está produciendo un aumento en la investigación en el campo de la visión artificial, están surgiendo diferentes métodos para obtener información de una imagen como la clasificación, la detección de diferentes objetos o la segmentación de una imagen. En este mapeo se investigará acerca de las dos primeras tareas. La primera consiste en determinar que clase, del grupo de clases que componen el conjunto de datos, se muestra de manera predominante en la imagen. Por otro lado, la detección de objetos en una imagen consiste en localizar dichos objetos aportando las coordenadas, permitiendo realizar otras tareas como el seguimiendo de cada objeto en un vídeo.

Estos procesos son computacionalmente pesados por lo que al comienzo de esta línea de investigación era necesario tener equipos de altas prestaciones. A través del paso del tiempo se han mejorado las capacidades *hardware* y se han diseñado nuevas arquitecturas de *machine learning* que permiten ser utilizadas en dispositivos con menos recursos, como es el caso del framework TensorFlow Lite, facilitando incluir estos modelos de *machine learning* en dispositivos más utilizados por los usuarios.

Este mapeo sistemático busca recopilar y analizar los estudios más recientes, últimos 5 años, en el campo de la visión artificial utilizando el framework TensorFlow Lite en dispositivos móviles aportando una visión general del estado actual del campo de investigación.

3.1.1. Metodología utilizada

Para realizar el mapeo sistemático se sigue la metodología PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analysis). Esta metodología divide el proceso del mapeo en 4 fases. La primera es la fase de identifación (*identification*), en la que se realiza la búsqueda en las fuentes de datos y se eliminan los estudios duplicados. La segunda es la de eliminación (*screening*), en esta fase se aplican los criterios de inclusión y exclusión. La tercera fase se denomina *eligibility*, se recopilan estudios nuevos a partir de las citas de los estudios encontrados en las fuentes de datos, estos nuevos estudios deben cumplir con los criterios de inclusión y exclusión, y se aplican los criterios de calidad. En la última fase, de inclusión (*included*), se define el conjunto de estudios seleccionados que van a ser utilizados para recopilar información.

Para llevar a cabo este proceso se utiliza la herramienta web Parsifal que permite establecer las preguntas de búsqueda, los criterios de inclusión y exclusión, la cadena de búsqueda y los criterios de calidad de manera ordenada y facilita la tarea de filtrar los artículos correctamente.

Para comenzar el mapeo sistemático se han definido las siguientes preguntas que se buscan responder al finalizar el proceso:

- ¿Cuáles son los ámbitos en los que se utiliza la detección y/o clasificación de un objeto de una imagen en un dispositivo móvil?
- ¿Existen estudios acerca de la detección y/o clasificación en el ámbito médico?
- ¿Cuáles son las métricas más utilizadas?
- ¿Cuáles son las arquitecturas más utilizadas?

El siguiente paso es elegir que fuentes de datos se van a utilizar para realizar la búsqueda. En este caso se han considerado 3 de las bases de datos más completas y utilizadas en el campo de investigación que se lleva a cabo el mapeo. Las bases de datos elegidas son:

- Web of Science (www.webofknowledge.com).
- Scopus (https://www.scopus.com/)
- IEEE Digital Library (https://ieeexplore.ieee.org/).

Estas fuentes han sido elegidas porque contienen una gran cantidad de revistas, libros y congresos y se consideran las más fiables para encontrar información en este campo de investigación.

Tras elegir las fuentes donde realizar la búsqueda, se procede a establecer la cadena de búsqueda. Esta debe contener los términos más relevantes para obtener los trabajos que permitan resolver las preguntas de estudio. Los términos elegidos son: *TensorFlow Lite, image classification, object detection, real-time*. A partir de estos términos se construye la siguiente cadena de búsqueda, mostrada en el Cuadro 1.

Cadena de búsqueda
("TensorFlow Lite") **AND** ("image classification" **OR** "object detection") **AND** ("real-time")

Cuadro 1: Cadena de búsqueda general mapeo sistemático 1

Esta cadena de búsqueda es correcta pero es necesario adaptarla a cada una de las fuentes, obteniendo las siguientes cadenas de búsqueda, indicadas en el Cuadro 2, Cuadro 3 y Cuadro 4.

Cadena de búsqueda Web of Science
("TensorFlow Lite") **AND** ("image classification" **OR** "object detection") **AND** ("real-time")

Cuadro 2: Cadena de búsqueda Web of Science mapeo sistemático 1

Cadena de búsqueda Scopus
(TITLE-ABS-KEY("TensorFlow Lite") **AND** ("image classification" **OR** "object detection")) **AND** ("real-time") **AND** PUBYEAR >2019 **AND** PUBYEAR <2024

Cuadro 3: Cadena de búsqueda Scopus

Cadena de búsqueda IEEE Digital Library
(All Metadata:"TensorFlow Lite") **AND** (All Metadata:"image classification" **OR** All Metadata:"object detection") **AND** (All Metadata:real-time")

Cuadro 4: Cadena de búsqueda IEEE Digital Library mapeo sistemático 1

Tras definir la cadena de búsqueda y las fuentes, es necesario establecer criterios que permitan decidir si un estudio debe ser descartado o no, según su título, abstract y palabras clave. Los criterios de inclusión definidos son:

- El estudio está escrito en español o inglés.
- El estudio es gratuito con la licencia de la Universidad de Salamanca.
- El estudio pertenece a un libro, revista o congreso.
- El estudio presenta una implementación práctica.
- El estudio está orientado a un dispositivo móvil.

Los criterios de exclusión son:

- El estudio no está escrito en español o inglés.
- El estudio no es gratuito con la licencia de la Universidad de Salamanca.
- El estudio no pertenece a un libro, revista o congreso.
- El estudio no presenta una implementación práctica.

- El estudio no está orientado a un dispositivo móvil.

El siguiente paso tras descartar los estudios que no cumplen los requisitos de inclusión es analizar la calidad de los estudios que si los cumplen. Para ello, se establecen los criterios de calidad, en total 6. Cada pregunta se responde con "Sí", "Parcialmente" o "No" y cada respuesta tiene una puntuación asociada, 1, 0,5 y 0 respectivamente. Para que un estudio cumpla los requisitos de calidad debe sumar al menos 3 puntos. Los criterios de calidad definidos son:

- ¿Se detallan los objetivos del proyecto correctamente?
- ¿La metodología de trabajo utilizada es correcta para alcanzar los objetivos planteados?
- ¿Los resultados obtenidos son precisos y discutidos correctamente?
- ¿El análisis de la información es rigurosa y basada en la literatura?
- ¿El proyecto tiene una aplicación para realizar pruebas?

3.1.2. Artículos recogidos

Tras establecer los pasos de la metodología PRISMA, se procede a realizar las búsquedas en cada una de las fuentes de datos con sus respectiva cadena de búsqueda. Los resultados obtenidos se exportan en formato BibTex para ser utilizados en la herramienta Parsifal.

Al finalizar todas las búsquedas se han encontrado 86 estudios, 17 de Web of Science, 44 de Scopus y 25 de IEEE Digital Library. El primer paso es eliminar los estudios duplicados, en este caso son 33.

El siguiente paso es leer el título y el abstract de todos los estudios y verificar que cumplen todos los requisitos de inclusión y ninguno de los de exclusión. Tras realizar este paso, se han descartado 32 estudios.

Los 21 estudios restantes deben pasar los criterios de calidad planteados para considerarse como válidos, para ello, se lee cada uno de los estudios y se puntúan cada pregunta. Tras realizar este paso se han descartado 5.

Al finalizar el proceso de selección, filtrado y calidad se han obtenido 16 estudios. Este proceso se puede visualizar en la Figura 2.

3.1.3. Análisis sobre la detección de objetos utilizando como framework TensorFlow Lite

En el campo de la visión artificial, un ámbito en el que se realizan numerosos estudios es la conducción autónoma. En el contexto de TensorFlow Lite no se busca

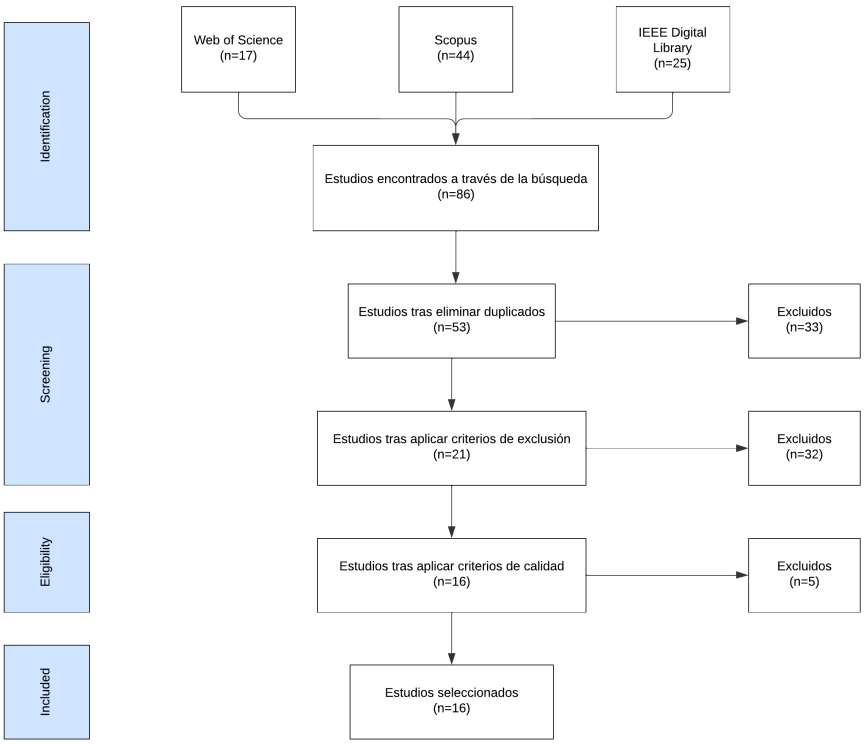


Figura 2: Diagrama de flujo PRISMA Mapeo 1

dicha conducción sino aportar herramientas que ayuden al manejo de vehículos de manera segura.

Por ello, el estudio [9] ha entrenado diferentes modelos para detectar si el conductor tiene la atención en la conducción o no. Para ello, ha utilizado un dataset de Kaggle en el que las imágenes están dividadas en 10 clases como conducción segura, escribiendo con la mano derecha, hablando por teléfono con la mano derecha, hablando con los acompañantes, entre otras. Han entrenado este dataset en diferentes arquitecturas siendo MobileNetV2 y ResNet50 las que mejores resultados obtuvieron.

El estudio [10] busca implementar una aplicación para detectar y clasificar las señales de tráfico. El modelo utilizado es MobileNetV2 y al procesar la imagen el usuario observa la señal diferenciada por un rectángulo creado por las coordenadas devueltas por el modelo. Los autores indican que el modelo funciona correctamente a pesar de que la imagen tenga diferentes ángulos o rotaciones de la señal de tráfico.

Por otro lado, para la conducción autónoma es importante reconocer los vehículos que se encuentran en el entorno. En este ámbito se han encontrado 3 estudios [11] [12] [13] que implementan en una aplicación móvil modelos para detectar vehículos. El primero clasifica los vehículos en 4 clases según el tipo de vehículo. El segundo además de localizar a los vehículos, detecta cuando se ha producido un choque y realiza una llamada a los servicios de emergencia. Para terminar, el tercero identifica el vehículo que aparece en la imagen y busca en un mapa lugares cercanos donde es posible comprar un vehículo similar al de la foto.

Otro ámbito en el que las investigaciones son numerosas es en la creación sistemas de ayuda para guiar a personas con discapacidad visual a través de la visión artificial. El estudio [14] ha creado una aplicación móvil que detecta el dinero y su valor, para facilitar su manejo, escaleras o personas para mejorar la movilidad de las personas con discapacidades visuales. El sistema realiza un *text to speech* para indicar al usuario lo que el sistema ha detectado. Este trabajo ha entrenado un modelo YOLO (You Only Look Once), posteriormente se ha convertido a modelo de TensorFlow y finalmente se ha convertido a TensorFlow Lite para ser utilizado correctamente en un dispositivo móvil.

Otro estudio es el presentado por Abhigyan Baruah [15], en él se detecta los objetos de la imagen y se calcula la distancia facilitando al usuario interactuar con su entorno. Para finalizar este ámbito, el estudio [16] presenta otra aplicación móvil para detectar objetos en una imagen para ayudar a las personas con discapacidad visual.

Además de estos campos, se han encontrado estudios para diferentes objetivos. El trabajo de Mayukha Thumiki [17] busca mejorar la eficiencia de la clasificación de objetos de la basura para su posterior reciclaje. Existen dos estudios que buscan la clasificación de animales. El primero [18] trata identificar que especie de serpiente se encuentra en la imagen que permite ayudar a encontrar el tratamiento en caso de envenenamiento. Se entrena el modelo YOLOv5 y posteriormente se transforma a TensorFlow Lite para su uso en la aplicación móvil. El segundo estudio [19] clasifica a 9 especies diferentes de mosca de la fruta que perjudican a la agricultura de Mauricio. Utiliza dos modelos, el primero es de detección, MobileNetV2, y el segundo es de clasificación pero se encuentra alojado en la nube, en este caso es Xception.

El estudio [20] ha implementado una aplicación móvil que posee un sistema de guía para la composición de una imagen para ayudar a los usuarios a sacar fotografías con mayor valor visual. Este sistema se consigue gracias a la utilización de un modelo MobileNet. El trabajo de Zsolt Domozi [21] ha desarrollado un sistema para encontrar y rescatar personas pérdidas a través de imágenes aportadas por un vehiculo aéreo no tripulado.

Se han encontrado dos trabajos que diseñan la arquitectura para dispositivos móviles [22] [23]. Para finalizar, el único estudio encontrado relacionado con el ámbito médico es [24]. El estudio busca clasificar 4 enfermedades del ganado bovino para ayudar a los ganaderos a detectar rápidamente las enfermedades de sus animales.

Tras realizar el análisis de los estudios se comprueba que no existe un número elevado que utilicen el framework TensorFlow Lite enfocado a dispositivos móviles para realizar la clasificación o detección de objetos en una imagen en tiempo real. A su vez, la mayoría de estos optan por el segundo proceso debido a la naturaleza de sus casos de estudio. El número de estudios analizados también ha disminuido debido a que numerosos trabajos que utilizan TensorFlow Lite se implementan en dispositivos Raspberry Pi en detrimento de los dispositivos móviles. También es importante destacar que solo se haya encontrado un estudio relacionado con el ámbito médico, esto es debido a que los modelos obtenidos para realizar la inferencia en los dispositivos móviles deben ser computacionalmente menos pesados y esto reduce el

acierto de las predicciones. El ámbito médico al ser, en algunas situaciones, crítico es necesario que las predicciones tengan la mayor veracidad posible sin tener en cuenta el tiempo que lleve a cabo.

3.1.4. Métricas utilizadas en la detección de objetos

Todos los artículos de la revisión del estado del arte se basan en unas determinadas métricas para comprobar los resultados de los modelos obtenidos. En la Figura 3 se muestran los indicadores estadísticos que se utilizan en dichas métricas.

Fact / Predict	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Figura 3: Indicadores estadísticos
[25]

- **TP:** El modelo predice correctamente una instancia como la clase positiva.
- **FP:** El modelo predice erróneamente una instancia como la clase positiva cuando no es así.
- **TN:** El modelo predice correctamente una instancia como la clase negativa.
- **FN:** El modelo predice erróneamente una instancia como la clase negativa cuando pertenece a la clase positiva.

Las métricas más utilizadas son *accuracy*, *precision*, *recall* y *F1 score*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Representa la proporción de muestras predichas correctamente respecto a todas las muestras.

$$Precision = \frac{TP}{TP + FP}$$

Indica el porcentaje de acierto de las predicciones respecto a las muestras positivas reales.

$$Recall = \frac{TP}{TP + FN}$$

Representa la proporción de muestras positivas predichas correctamente respecto a las muestras positivas reales.

$$F1 = \frac{2 * TP}{2 * TP + FP + FN}$$

Es la media armónica de *precision* y *recall*.

3.1.5. Arquitecturas de detección de objetos

Tras analizar los estudios anteriores se comprueba que para obtener los modelos deseados, utilizan dos técnicas, la primera mediante arquitecturas especializadas en dispositivos con poco recursos como MobileNet o ResNet y la segunda a través de una arquitectura como YOLO, no orientada a dispositivos *edge*, y posteriormente transformar el modelo obtenido tras el entrenamiento a una versión compatible para desplegarlo en un dispositivo móvil.

Previamente a explicar cada una de estas arquitecturas hablaremos de manera resumida de las Redes Neuronales Convolucionales (Convolutional Neural Networks, CNN) debido a que todas estas arquitecturas utilizan CNNs.

Una CNN es un tipo de red neuronal diseñada para procesar datos con estructura de cuadrícula, en este caso imágenes. Las CNNs están compuestas por varias capas con propósitos específicos. Las capas convolucionales aplican filtros, o *kernels*, a la entrada para crear mapas de características, detectando características como bordes, texturas y patrones básicos. Las capas de activación aplican funciones no lineales como ReLU a los mapas de características, permitiendo que la red aprenda funciones más complejas.

Las capas de *pooling* reducen la dimensionalidad de los mapas de características mediante operaciones de agrupamiento como *max pooling*, disminuyendo la carga computacional y ayudando a extraer características invariantes a pequeñas traslaciones.

Las capas completamente conectadas unen todas las neuronas de la capa anterior a cada neurona de la capa siguiente, permitiendo la clasificación o regresión final basada en las características extraídas.

El entrenamiento de las CNNs consiste en el ajuste de los valores de los filtros y los pesos de las capas completamente conectadas permitiendo que la red aprenda las características relevantes de las imágenes para la tarea específica, como clasificación de objetos, detección de objetos o segmentación de imágenes.

A continuación, se detallan las arquitecturas más utilizadas por los estudios analizados durante el estado del arte:

3.1.5.1. YOLO

Es una familia de arquitecturas, la primera versión se publicó en junio del año 2015 y la última versión ha sido publicada recientemente en febrero del año 2024. Cada versión ha modificado ligeramente la arquitectura inicial mejorando sus resultados. Las diferentes versiones han mejorado los resultados de las anteriores versiones tanto en precisión como en la velocidad de la identificación de los objetos. Esto se ha conseguido con la inclusión de *anchor boxes* para mejorar la detección de objetos pequeños, batch normalization para una mejor convergencia y precisión, la predicción de cajas delimitadoras en tres escalas diferentes para mejorar la detección de objetos de distintos tamaños, implementación de la técnica de focal loss para manejar el desequilibrio entre clases.

Estas mejoras pertenecen a las primeras versiones, las versiones YOLOv6, YOLOv7 y YOLOv8 han continuado optimizando la arquitectura de red, mejorando tanto la precisión como la velocidad, y enfocándose en la facilidad de implementación y soporte para diverso hardware. Además, integraron nuevas técnicas de aprendizaje profundo y aumentación de datos, adaptándose a diferentes necesidades y recursos computacionales.

En este apartado se explica la arquitectura original que sirve de base para las versiones posteriores. La red neuronal tiene 24 capas convolucionales seguidas de dos capas totalmente conectadas. Alternan capas convolucionales 1x1 se reduce el espacio de características de las capas anteriores. En la Figura 4 se observa su arquitectura.

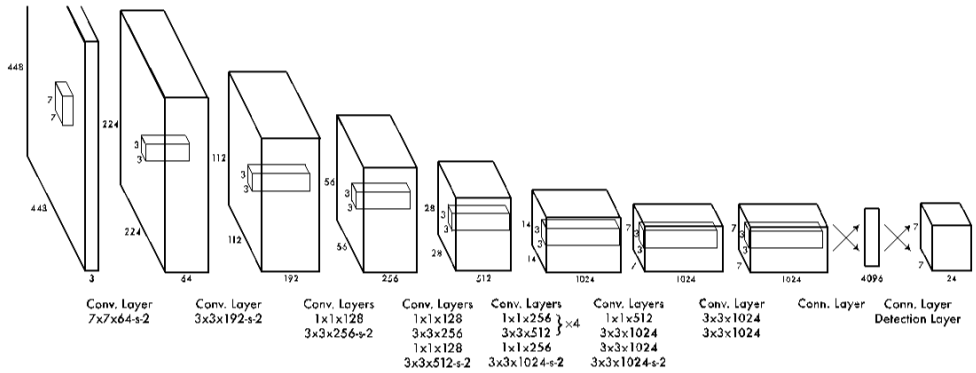


Figura 4: Arquitectura YOLO
[26]

La imagen de entrada es dividida en una cuadrícula originalmente 7x7, cada celda se encarga de localizar y predecir la clase que abarca junto a su confianza. Posteriormente, se determinan las cajas delimitadoras contienen los objetos de la imagen. Debido a que estas cajas pueden pertenecer a diferentes celdas, es necesario descartar las que no son relevantes, para ello se utiliza IoU (Intersection over Union), el área de la intersección dividida por el área de la unión debe ser mayor a un valor límite para considerar esa celda como representativa. [26]

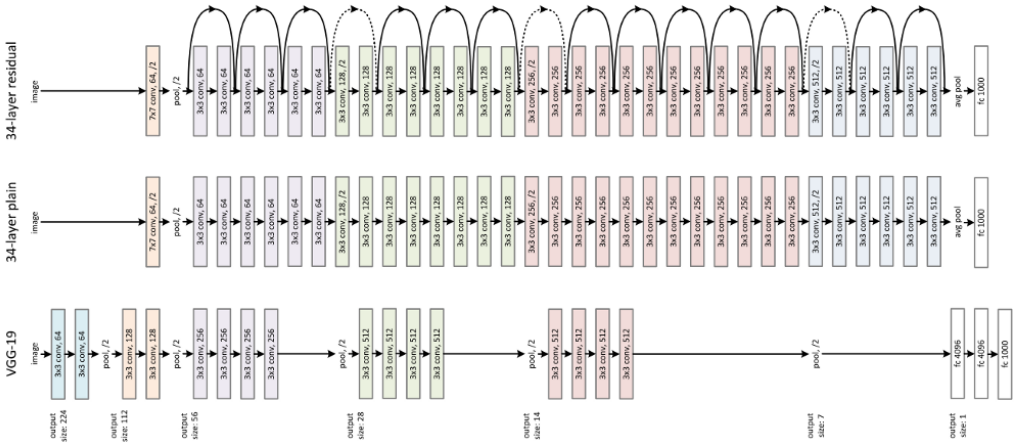


Figura 5: Arquitectura ResNet
[27]

3.1.5.2. ResNet

Presentada por Microsoft en 2015, tiene diferentes variantes según el número de las capas, 18, 34, 50, 101 y 152 pero todas ellas poseen la misma base, los bloques residuales. Cada bloque residual incluye convoluciones, dos capas para las versiones de 18 y 34 o tres para las superiores, con normalización por lotes y funciones de activación ReLU, junto con una conexión de salto que salta una o más capas y suma su salida a la salida de las capas convolucionales.

Las conexiones de salto permiten que el gradiente fluya más fácilmente a través de la red durante el entrenamiento, lo que reduce el problema de la desaparición del gradiente y facilita la optimización de redes muy profundas. ResNet se compone de una secuencia de bloques residuales agrupados en varias etapas, cada una con un número variable de bloques residuales y caracterizada por un cambio en la dimensionalidad de las características. La estructura general incluye una convolución inicial de 7x7 seguida de una operación de pooling, múltiples bloques residuales organizados en etapas, y una capa final con un promedio global seguido de una capa completamente conectada para la clasificación [27]. En la Figura 5 se observa la arquitectura de ResNet34.

3.1.5.3. MobileNetV2

Es una arquitectura desarrollada por Google que mejora a su primera versión, MobileNetV1. Se caracteriza por su uso de bloques de inversión residual con cuellos de botella lineales. Cada bloque incluye tres etapas: expansión (convolución 1x1 para aumentar la dimensionalidad), profundidad (convolución de profundidad separable 3x3), y proyección (convolución 1x1 para reducir la dimensionalidad). Este diseño permite un procesamiento eficiente en un espacio de alta dimensionalidad. Las conexiones de salto mejoran el flujo de gradientes durante el entrenamiento, mientras que las convoluciones de profundidad separables reducen la carga computacional sin perder capacidad de aprendizaje. La arquitectura incluye una convolución inicial de

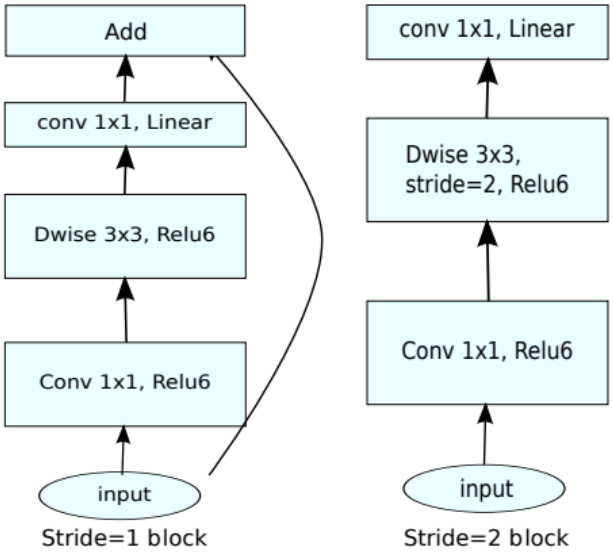


Figura 6: Arquitectura MobileNetV2
[28]

3x3, seguida por múltiples bloques de inversión residual, una convolución final de 1x1 y una capa de clasificación. [28]

Al finalizar este mapeo sistemático se ha conseguido tener una visión global acerca de los estudios anteriores que llevan a cabo clasificación o detección de objetos en una imagen utilizando TensorFlow Lite aplicado a dispositivos móviles. Además, se han conocido las métricas y arquitecturas más usadas en estos problemas y sus objetivos. Finalmente, a través de la revisión del estado del arte se ha podido comprobar la posibilidad y la eficacia de incluir un modelo de clasificación o de detección de objetos en un imagen en un dispositivo móvil gracias al framework TensorFLow Lite.

3.2. Revisión sistemática de la literatura sobre la generación *text-to-text* enfocada a la respuesta de preguntas

En los últimos años la inteligencia artificial generativa de texto está evolucionando rápidamente, mejorando los modelos así como los conjuntos de datos de entrenamiento y por otro lado ampliando las tareas que puede realizar.

En este caso se desea buscar estudios que traten la respuesta a preguntas del ámbito médico. Esta respuesta debe estar relacionada a un contexto aportado en la pregunta. De esta manera en esta revisión no se explora sobre los estudios que se basan en la generación de texto sin un contexto previo.

Por otro lado, en este campo el idioma más utilizado es el inglés, por lo que la mayoría de los modelos y conjuntos de datos se encuentran en este idioma, siendo reducido el número de los mismos en español. Por esto, se procede a investigar

acerca de los estudios que utilizan modelos o conjuntos de datos en español que estén enfocados al text to text aunque no sean necesariamente para la tarea de responder preguntas. En este análisis se busca únicamente los estudios publicados en los últimos 5 años debido a los grandes cambios con el paso del tiempo y se desea obtener los resultados más recientes.

3.2.1. Metodología utilizada

En este mapeo sistemático se va a seguir la misma metodología que en el anterior, PRISMA. De la misma manera, se utiliza la herramienta Parsifal para facilitar el proceso.

Para comenzar, se han definido las siguientes preguntas que se buscan responder al finalizar el proceso:

- ¿Existen proyectos especializados en el campo de la medicina o farmacología?
- ¿En qué tareas del procesamiento del lenguaje existen modelos o conjuntos de datos en español?
- ¿Cuáles son las métricas más utilizadas?
- ¿Cuáles son las arquitecturas más utilizadas?

Las fuentes a utilizar en este mapeo son las mismas que en el anterior debido a los motivos mencionados anteriormente.

Tras elegir las fuentes donde realizar la búsqueda, se procede a establecer la cadena de búsqueda. Esta debe contener los términos más relevantes para obtener los trabajos que permitan resolver las preguntas de estudio. Los términos elegidos son: *text to text, question answering, model, dataset, medicine, spanish*. A partir de estos términos se construye la siguiente cadena de búsqueda, se muestra en el Cuadro 5.

Cadena de búsqueda
(“text to text” **OR** “question answering”) **AND** (“model” **OR** “dataset”) **AND** (“medicine” **OR** “spanish”)

Cuadro 5: Cadena de búsqueda general mapeo sistemático 2

Es necesario que esta cadena de búsqueda se adapte a las fuentes de datos, mostradas en el Cuadro 6, Cuadro 7 y Cuadro 8.

Cadena de búsqueda Web of Science
("text to text" **OR** "question answering") **AND** ("model" **OR** "dataset") **AND** ("medicine" **OR** "spanish")

Cuadro 6: Cadena de búsqueda Web of Science mapeo sistemático 2

Cadena de búsqueda Scopus
(TITLE-ABS-KEY("text to text" **OR** "question answering") **AND** TITLE-ABS-KEY("model" **OR** "dataset") **AND** TITLE-ABS-KEY("medicine" **OR** "spanish")) **AND** PUBYEAR >2019 **AND** PUBYEAR <2024

Cuadro 7: Cadena de búsqueda Scopus mapeo sistemático 2

Cadena de búsqueda IEEE Digital Library
(All metadata:"text to text" **OR** All metadata:"question answering") **AND** (All metadata:"model" **OR** All metadata:"dataset") **AND** (All metadata:"medicine" **OR** All metadata:"spanish")

Cuadro 8: Cadena de búsqueda IEEE Digital Library mapeo sistemático 2

Los criterios de inclusión definidos son:

- El estudio está escrito en español o inglés.
- El estudio es gratuito con la licencia de la Universidad de Salamanca.
- El estudio pertenece a un libro, revista o congreso.
- El estudio realiza la respuesta a través de preguntas de texto.
- El estudio tiene corpus en español o está relacionado con el campo de la medicina.
- El estudio utiliza la arquitectura transformer.

Los criterios de exclusión son:

- El estudio no está escrito en español o inglés.
- El estudio no es gratuito con la licencia de la Universidad de Salamanca.
- El estudio no pertenece a un libro, revista o congreso.

- El estudio no realiza la respuesta a través de preguntas de texto.
- El estudio no tiene corpus en español y no está relacionado con el campo de la medicina.
- El estudio no utiliza la arquitectura transformer.

La puntuación de los criterios de calidad es la misma que en el primer mapeo. Los criterios de calidad definidos son:

- ¿Se detallan los objetivos del proyecto correctamente?
- ¿La metodología de trabajo utilizada es correcta para alcanzar los objetivos planteados?
- ¿Los resultados obtenidos son precisos y discutidos correctamente?
- ¿El análisis de la información es rigurosa y basada en la literatura?
- ¿El proyecto tiene una aplicación para realizar pruebas?

3.2.2. Artículos recogidos

Tras realizar el proceso siguiendo la metodología PRISMA, inicialmente se obtienen 286 estudios de las tres fuentes de datos. Se eliminan 63 duplicados, se excluyen 63 tras los criterios de inclusión. Cabe destacar que la mayoría de estos es debido a que su investigación se basa en responder a preguntas relativas a una imagen en vez de a un texto. Finalmente, se han excluido 38 tras aplicar los criterios de calidad, obteniendo 17 artículos para realizar el análisis. En la Figura 7 se observa este proceso.

3.2.3. Análisis sobre la generación *text-to-text* enfocada a la respuesta de preguntas

Para comenzar, se procede a conocer cuáles son los objetivos, los métodos y los resultados de los estudios que tratan la generación de respuestas a preguntas del ámbito médico. La mayoría de ellos está especializado en un campo específico. Por ejemplo, el estudio [29] presenta un chatbot especializado en el campo de la oftalmología. La arquitectura del sistema posee dos módulos, el primero únicamente responde a la pregunta del usuario mientras que la segunda además de generar una respuesta, aporta un contexto. El primer módulo consiste en un codificador y decodificador que utiliza Gated Recurrent Unit(GRU). El segundo módulo obtiene la respuesta y el contexto recuperando los 5 documentos de la base de datos que tengan más probabilidad de contener la respuesta a la pregunta y posteriormente se utiliza un modelo BERT para obtener el origen y final de la respuesta dentro del texto. Además, el estudio presenta un conjunto de datos, supervisado por un oftalmólogo, para cada uno de los módulos.

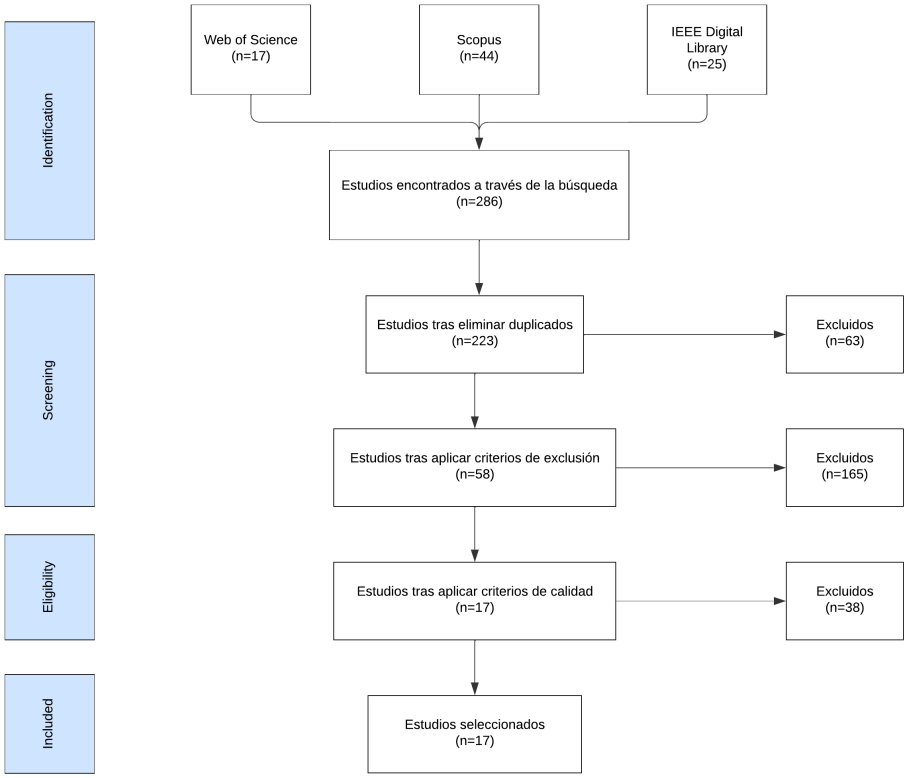


Figura 7: Diagrama de flujo PRISMA Mapeo 2

El estudio [30] presenta un modelo para la generación de respuestas relacionadas con la medicina china, basado en la arquitectura LLaMA que ha sido pre-entrenado a partir de un corpus de libros médicos y se ha realizado un fine-tuning con conjunto de datos de instrucciones médicas. Por otro lado, El estudio propuesto por Zini [31] presenta un modelo para responder preguntas de un examen clínico estructurado objetivo para reducir costes logísticos en exámenes de estudiantes de medicina.

Otro estudio encontrado está especializado en el covid-19 [32], en él se presenta un modelo para responder preguntas acerca de este virus. Para obtenerlo, se han utilizado dos modelos pre-entrenados, BERT y RoBERTa, se ha realizado un fine-tuning con el conjunto de datos SQUAQ para obtener modelos especializados en la tarea de requerida. Posteriormente, han utilizado un conjunto de datos que contiene preguntas y respuestas acerca del covid19 para especializarlo en este dominio.

Algunos estudios tratan de realizan comparaciones entre diferentes modelos o conjuntos de datos para obtener conclusiones de cuáles funcionan mejor para cada tarea del procesamiento de lenguaje natural. El primero [33] ha investigado acerca del rendimiento del modelo pre-entrenado BioBERT en la tarea de responder preguntas. Este modelo ha sido entrenado con un corpus específico para el ámbito biomédico. A través del entrenamiento, el modelo es capaz de responder preguntas cuya respuesta es un hecho, una lista o dicotómica, si o no. El segundo estudio [34] recopila 6 conjuntos de datos del ámbito médico, los compara y aporta uno nuevo, obteniendo un modelo, entrenado con el conjunto de datos creado, que mejora los anteriores.

Para finalizar, existen otros estudios como [35] [36] que presentan sistemas capaces de responder a preguntas genéricas del ámbito médico. Cada una utiliza diferentes técnicas y arquitecturas para conseguir sus resultados, en este caso la primera utiliza una arquitectura Bi-directional Long Short-Term Memory (Bi-LSTM) y la segunda el modelo pre-entrenado T5.

Tras analizar los estudios enfocados en la generación de una repuesta a una pregunta del ámbito médico se lleva a cabo otro análisis de los estudios acerca de los modelos o conjuntos de datos creados para el lenguaje español. El estudio [37] presenta una arquitectura para responder preguntas en español. El usuario realiza la pregunta, añadiendo o no un texto de contexto, en el caso de omitirlo se utiliza un modelo BI-LSTM para buscar en Wikipedia texto relacionado a la pregunta. Tras obtener el contexto, se responde a la pregunta a través del modelo BERT entrenado con el dataset XQuAD para la tarea de responder preguntas. La respuesta generada es un fragmento literal del contexto.

Por otro lado, hay estudios como el presentado por Óscar R. Navarrete-Parra[38] que buscan adaptar un modelo para diferentes ámbitos como un modelo GPT, DialoGPT, a un dominio específico en español. Para llevar a cabo este proceso se utiliza un modelo de recompensa a partir de las preferencias de los humanos para obtener un mejor resultado al realizar el fine-tuning del modelo. Este modelo obtenido no necesita un contexto para responder a la pregunta.

El Centro Nacional de Supercomputación ha presentado el estudio [39]. En él se presentan una familia de modelos y conjuntos de datos para el idioma español para diferentes tareas PLN como clasificación de texto, reconocimiento y clasificación de entidades, similitud semántica textual, respuesta a preguntas, entre otras.

Otro enfoque es el presentado en [40], que ha realizado un fine-tuning del modelo mT5 para generar respuestas incorrectas a una pregunta teniendo un contexto y la respuesta correcta. El objetivo del estudio es obtener un modelo que agilice el proceso de realizar un test cuyas preguntas presenten diferentes opciones y el usuario tenga que elegir la opción correcta.

Existen estudios enfocados a especializarse para responder preguntas de un ámbito concreto, [41], que presenta un sistema para responder preguntas en el entorno industrial de manufactura. Se ha creado un conjunto de datos utilizado para el entrenamiento basado en etiquetar los PDFs que contienen las instrucciones. El sistema tiene dos generadores de repuesta, uno general y otro específico, según la pregunta del usuario. El sistema contiene un clasificador que determina qué generador debe utilizarse para cada pregunta. Para llevar a cabo la generación de la repuesta se ha realizado fine-tuning de 6 modelos pre-entrenados, 5 de español y uno multilenguaje. Otros estudios, como [42], en cambio presentan un fine-tuning de modelos pre-entrenados para la generación de respuestas a preguntas de manera genérica.

Otros estudios no se especializan en el idioma del inglés o español sino que obtienen modelos que funcionan para diferentes lenguajes como [43] [44]. El primero presenta un conjunto de datos de preguntas y su respuesta en 9 idiomas, entre ellos el español. La fuente de los datos es Wikipedia. El segundo estudio presenta otro

conjunto de datos en inglés y otros 6 idiomas, uno de ellos es el español.

Para finalizar, se ha encontrado un trabajo relacionado con el ámbito médico en español [45]. Realiza una comparación entre diferentes variaciones del modelo BERT utilizando como conjunto de datos PharmaCoNER. Este dataset contiene numerosos casos clínicos en español, su objetivo es reconocer términos químicos y proteínas. El entendimiento de estos términos posibilita obtener un modelo con la capacidad de responder a preguntas relativas al ámbito médico o farmacológico.

Tras revisar todos los estudios, se comprueba que en los últimos años se está llevando a cabo un esfuerzo para aumentar el número de investigaciones de PLN en el ámbito médico, un campo importante en la sociedad pero en muchas ocasiones delicado. A pesar de esto, dichos estudios están orientados para el idioma inglés mientras que para el español únicamente se ha encontrado uno. Por otro lado, se detecta que el número de modelos o conjuntos de datos en español para cualquier tarea PLN es reducido.

3.2.4. Métricas utilizadas en la generación de respuestas

La mayoría de los estudios analizados utilizan u obtienen modelos que responden preguntas obtienen la respuesta de manera extractora, es decir, no generan la repuesta a partir del contexto sino que aportan la respuesta indicando la palabra de inicio y de fin del texto. Por este motivo, las métricas utilizadas en estos estudios son las mencionadas en el anterior mapeo sistemático debido a que los resultados son binarios es decir, una respuesta es correcta o no según si se ha elegido correctamente la primera y última palabra del texto como respuesta.

Para la generación de la respuesta u otras tareas como la traducción de un idioma a otro se utilizan otras métricas, entre las que destacan ROUGE y BLEU, la primera se centra en la métrica *recall* y la segunda en *precision*. Respecto a la métrica ROUGE existen variaciones como ROUGE-N o ROUGE-L. La primera compara los n-gramas entre el texto generado y la referencia, las más utilizadas son ROUGE-1 y ROUGE-2. ROUGE-L busca el máximo número de palabras en el mismo orden entre el texto generado y la referencia sin necesidad de que dichas palabras estén unas seguidas de las otras.

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{referencias}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{referencias}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)}$$

- **S:** Conjunto de referencias.
- **gram_n:** n-grama en el texto generado.
- **$\text{Count}_{\text{match}}(\text{gram}_n)$:** Número de n-gramas coincidentes entre el texto generado y el texto de referencia.
- **$\text{Count}(\text{gram}_n)$:** Número total de n-gramas en el texto de referencia.

$$\text{ROUGE-L} = F\text{-score} = \frac{(1+\beta^2)\cdot \text{R}_{\text{LCS}} \cdot \text{P}_{\text{LCS}}}{\text{R}_{\text{LCS}} + \beta^2 \cdot \text{P}_{\text{LCS}}}$$

$$\text{donde} \quad \text{R}_{\text{LCS}} = \frac{\text{LCS}(X,Y)}{\text{len}(Y)}, \quad \text{P}_{\text{LCS}} = \frac{\text{LCS}(X,Y)}{\text{len}(X)} [46]$$

- **$\mathbf{R_{LCS}}$:** *Recall* de la LCS.
- **$\mathbf{P_{LCS}}$:** *Precision* de la LCS.
- **$\boldsymbol{\beta}$:** Factor de ponderación.
- **X:** Texto generado.
- **Y:** Texto de referencia.
- **len(X):** Longitud del texto generado.
- **len(Y):** Longitud del texto de referencia.
- **LCS (X,Y):** Longitud de la subsecuencia común más larga entre X e Y.

Por otro lado, BLEU mide el solapamiento entre los n-gramas contiguos del texto generado y la referencia.

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

$$p_n = \frac{\sum_{\text{gram}_n \in \text{gen}} \text{mín}(\text{Count}(\text{gram}_n), \text{Count}(\text{gram}_n^{\text{ref}}))}{\sum_{\text{gram}_n \in \text{gen}} \text{Count}(\text{gram}_n)}$$

$$\text{BP} = \begin{cases} 1 & \text{si } c > r \\ \exp(1 - \frac{r}{c}) & \text{si } c \leq r \end{cases} [47]$$

- **BP:** Penalización por brevedad.
- **$\mathbf{w_n}$:** Peso para cada n-grama.
- **$\mathbf{p_n}$:** *Precision* para cada n-grama.
- **gen:** Conjunto de n-gramas en el texto generado.
- **ref:** Conjunto de n-gramas en el texto de referencia.
- **Count($\mathbf{gram_n}$):** Número total de ocurrencias del n-grama en el texto generado.

- **Count($gram_n^{ref}$):** Número total de ocurrencias del n-grama en común entre el texto generado y el texto de referencia.
- **c:** Longitud del texto generado.
- **r:** Longitud del texto de referencia más cercano.
- **len(X):** Longitud del texto generado.
- **len(Y):** Longitud del texto de referencia.
- **LCS (X,Y):** Longitud de la subsecuencia común más larga entre X e Y.

3.2.5. Arquitecturas utilizadas en la generación de respuestas

En relación a la arquitectura de los modelos, utilizan la arquitectura transformer [48]. Los transformers son una arquitectura de red neuronal diseñada para procesar y generar secuencias de texto. Introducidos por Vaswani et al. en 2017, los transformers se basan en un mecanismo de atención que permite a la red enfocarse en diferentes partes de la entrada de manera flexible y eficiente. La arquitectura del transformer consta de una serie de capas de codificador y decodificador. Cada capa del codificador incluye una subcapa de atención multi-cabezal y una subcapa de red neuronal feed-forward. La atención multi-cabezal permite que el modelo considere diferentes posiciones de la secuencia de entrada simultáneamente, capturando relaciones a largo plazo en el texto. La salida de la atención se pasa a través de la red feed-forward para una transformación adicional. El decodificador es similar al codificador, pero incluye una capa adicional de atención que se enfoca en la salida del codificador, permitiendo que el decodificador acceda a la información procesada por el codificador.

El mecanismo de atención calcula una representación ponderada de las entradas, donde los pesos se determinan en función de la relevancia de cada palabra en la secuencia, permitiendo al modelo enfocarse en las partes importantes del texto al procesar la información. Dado que los transformers no tienen una estructura secuencial inherente como las redes neuronales recurrentes (RNNs), se utilizan codificaciones posicionales para incorporar información sobre el orden de las palabras en la secuencia.

Las RNNs son un tipo de red neuronal diseñada para procesar secuencias de datos manteniendo información de las entradas anteriores mediante bucles internos. Cada neurona en una RNN recibe la entrada actual y la salida de la neurona en el paso anterior, lo que permite considerar el contexto temporal. Las RNNs pueden enfrentar problemas de gradientes desvanecidos y explosivos durante el entrenamiento, dificultando la propagación de información a largo plazo. Esta problemática es solucionada con los transformers ya que son altamente paralelizables y eficientes en el manejo de secuencias largas. En la Figura 8 se observa la arquitectura transformer.

La mayoría de los estudios analizados utilizan el modelo pre-entrenado BERT o variaciones de él. También numerosos estudios han utilizado el modelo T5, o su

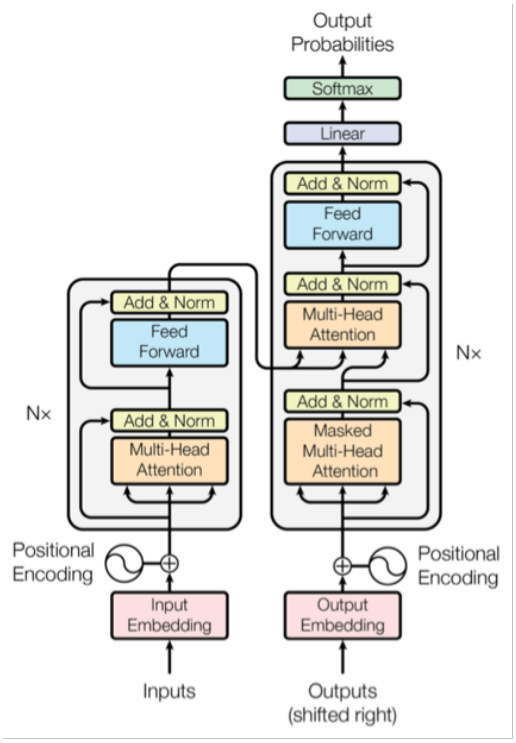


Figura 8: Arquitectura Transformer [48]

versión plurilingüe mT5, y BART.

3.2.5.1. BERT

Bidirectional Encoder Representations from Transformers, BERT, es un modelo de lenguaje desarrollado por Google en 2018, basado en la arquitectura de transformers. BERT se distingue por su capacidad para comprender el contexto de las palabras en una secuencia de texto de manera bidireccional, analizando tanto la parte anterior como la posterior de una palabra. Se pre-entrena en dos tareas principales: el modelado de lenguaje enmascarado (MLM), donde algunas palabras en una oración se enmascaran y el modelo debe predecirlas basándose en el contexto, y la predicción de la siguiente oración (NSP), que ayuda a entender las relaciones entre oraciones.

BERT utiliza únicamente el codificador del transformer, compuesto por varias capas de atención y redes neuronales feed-forward, que capturan relaciones complejas entre palabras. Después del pre-entrenamiento, es posible realizar fine-tuning se ajusta para tareas específicas de PLN adaptando los pesos pre-entrenados para especializar el modelo en esas tareas [49]. En la Figura 9 se observa la arquitectura de BERT.

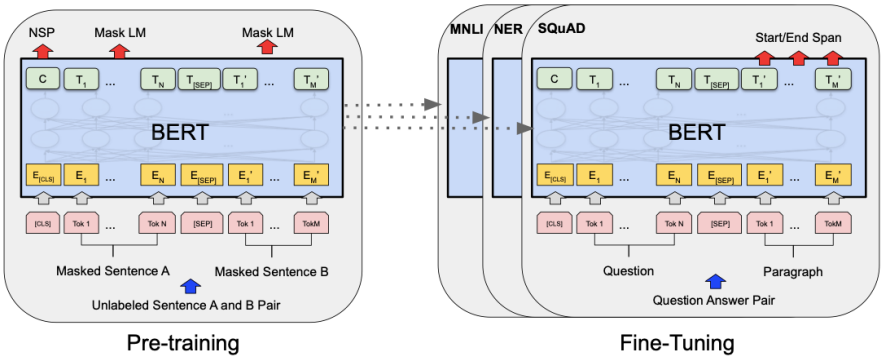


Figura 9: Arquitectura BERT
[49]

3.2.5.2. T5

Text-to-text Transfer Transformer, T5, es un modelo de lenguaje desarrollado por Google que aborda todas las tareas PLN como problemas de transformación de texto a texto. Introducido en 2019, T5 se basa en la arquitectura de transformers y unifica diversas tareas de PLN en un solo marco, donde la entrada y la salida son siempre texto. Por ejemplo, una tarea de traducción se formularía como "Traduce del inglés al español: X", siendo X texto en inglés, y la salida es el texto en alemán. T5 se pre-entrena en una gran cantidad de datos utilizando una tarea de -elleno de espacios", similar al modelado de lenguaje enmascarado (MLM) de BERT, pero con variaciones más complejas. Después del pre-entrenamiento, T5 se ajusta específicamente para tareas individuales utilizando conjuntos de datos etiquetados, permitiendo que el modelo adapte sus conocimientos generales a las particularidades de cada tarea. La arquitectura de T5 utiliza tanto el codificador como el decodificador del transformer [50]. En la Figura 10 se observa un esquema de diferentes entradas y sus correspondientes salidas de T5.

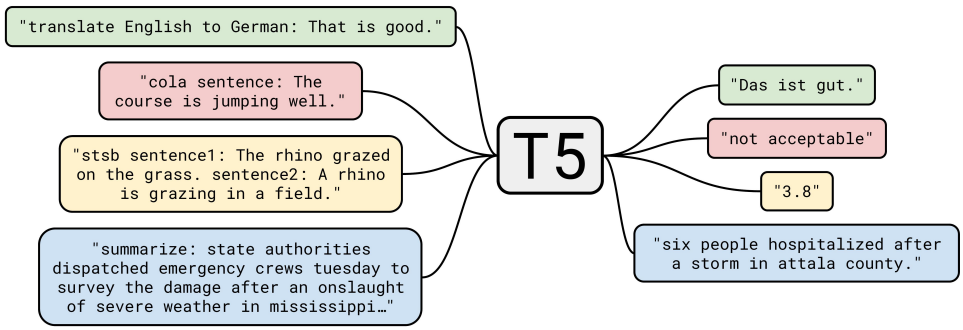


Figura 10: Diagrama T5
[50]

3.2.5.3. BART

Bidirectional and Auto-Regressive Transformers, BART, es un modelo de lenguaje desarrollado por Facebook AI en 2019. Combina características de modelos bidireccionales y autorregresivos mediante un pre-entrenamiento que aplica ruido a los textos de entrada como borrar, permutar y enmascarar tokens, y posteriormente entrena al modelo para reconstruir el texto original. La arquitectura de BART utiliza un codificador bidireccional, similar a BERT, y un decodificador autorregresivo, similar a GPT. Esta combinación permite que BART maneje tareas de comprensión y generación de texto de manera efectiva [51]. En la Figura 11 se observa un esquema de la arquitectura de BART.

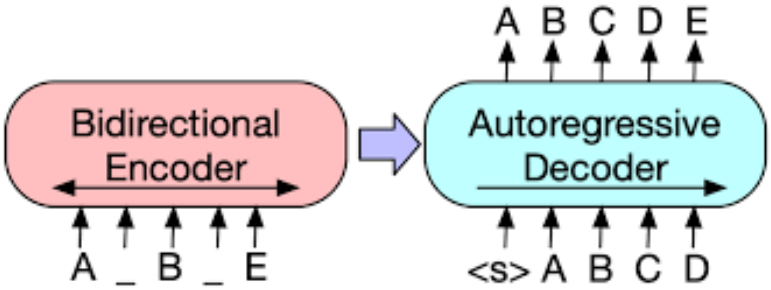


Figura 11: Arquitectura BART
[51]

4. Metodología

Tras realizar el estado del arte y conocer los estudios anteriores del campo, se procede a crear un sistema que permita "leer" el prospecto de los medicamentos a las personas con visión reducida detectando las cajas de los medicamentos de manera efectiva a través de reconocimiento de imágenes y después de realizar la identificación del medicamento, que el usuario pueda solicitar información del mismo y generar una respuesta a la pregunta del usuario, utilizando como contexto el prospecto del medicamento detectado.

Para llevar a cabo la metodología, se realizan dos fases, que se aplican al caso de estudio. La primera para la detección de medicamentos y la segunda para la generación de respuestas.

4.1. Detección de medicamentos

La identificación de diferentes objetos en una imagen se puede realizar utilizando diferentes técnicas de *machine learning* como la clasificación, detección de objetos y la segmentación de una imagen. En este proyecto se han utilizado diferentes modelos de clasificación y detección de objetos para realizar una comparación y elegir la mejor opción siguiendo los resultados obtenidos.

Tanto para la tarea de clasificación como para la detección de objetos, el proceso de obtención de un modelo posee las mismas fases: obtención de imágenes, preprocesamiento de las mismas, entrenamiento del modelo y finalmente la validación del modelo obtenido.

Para reconocer y clasificar diferentes cajas de medicamentos en una imagen es necesario utilizar un dataset suficientemente grande y variado. Además, tener en cuenta diferentes aspectos relacionados con el conjunto de imágenes como cambios en la luminosidad o diferentes fondos tras la caja del medicamento.

Al comienzo del estudio se realizó una exhaustiva búsqueda con el objetivo de encontrar un conjunto de datos ya etiquetado previamente. Tras realizar la búsqueda se concluyó que no existían datasets acerca de cajas de medicamentos españoles, los encontrados únicamente identifican pastillas de manera genérica.

Por este motivo, para este proyecto se ha creado un dataset propio, en el que se van a incluir los 5 medicamentos más vendidos en el año 2022 en España [52], "Nolotil", "Paracetamol Kern", "Adiro 100 EFG", "Enantyum" y "Paracetamol Cinfa".

Debido a que estos medicamentos, excepto el Adiro, tienen diferentes formatos es necesario elegir uno para medicamento. En este caso se han elegido los siguientes medicamentos que son las clases a identificar por el modelo:

- Nolotil 575 mg cápsulas duras (Clase 1)
- Paracetamol Kern Pharma 1 g comprimidos EFG (Clase 2)

- Adiro 100 mg comprimidos gastrorresistentes EFG (Clase 3)
- Enantyum 25 mg comprimidos recubiertos con película (Clase 4)
- Paracetamol Cinfa 650 mg comprimidos EFG (Clase 5)

Para entrenar un modelo de *machine learning* , el conjunto de imágenes se divide en tres subconjuntos:

- **Conjunto de entrenamiento:** Este conjunto de datos se utiliza para ajustar los pesos de la red neuronal, para enseñar a la red a reconocer patrones y aprender a hacer predicciones por lo que contiene la mayoría de los datos disponibles. Durante el entrenamiento, el algoritmo ajusta los parámetros internos, pesos, para minimizar el error en este conjunto. La red se entrena mediante múltiples iteraciones, epochs, pasando repetidamente por estos datos y ajustando los pesos en función del error calculado.
- **Conjunto de validación:** Se utiliza para ajustar los hiperparámetros del modelo y para prevenir el sobreajuste, overfitting. El sobreajuste ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien a datos nuevos. Este conjunto es independiente del conjunto de entrenamiento, pero se extrae del mismo conjunto de datos original. Se utiliza durante el proceso de entrenamiento para evaluar el rendimiento del modelo después de cada iteración sin afectar el entrenamiento de los pesos. Al evaluar el modelo con los datos de validación, se pueden tomar decisiones sobre cuándo detener el entrenamiento, ajustar la arquitectura del modelo, la tasa de aprendizaje y otros hiperparámetros.
- **Conjunto de prueba:** Este conjunto se usa para evaluar el rendimiento final del modelo después de haber sido entrenado y validado. Proporciona una estimación imparcial de cómo se desempeñará el modelo con datos no vistos. Es completamente independiente de los conjuntos de entrenamiento y validación. Se debe mantener aislado hasta que el modelo esté completamente entrenado y ajustado. El rendimiento en el conjunto de prueba se utiliza como una métrica final para determinar la efectividad y capacidad de generalización del modelo.

El dataset fue creado con 575 imágenes, 115 de cada medicamento, de esta manera no hay ninguna clase sobre o subrepresentada. En cada imagen únicamente aparece un medicamento y no existe ninguna imagen en la que no se muestre un medicamento. El conjunto de imágenes se ha divido en 400 para el entrenamiento, 75 para la validación y 100 para las pruebas. Además, se aplican técnicas de aumento de imágenes en el conjunto de entrenamiento como rotación, cambio de saturación y brillo o ruido en las imágenes, entre otros, con el objetivo de que el conjunto de entrenamiento sea más variado. Tras este aumento, se obtienen 1200 imágenes en el conjunto de entrenamiento, multiplicando por 3 el número de imágenes de este conjunto. En cada subconjunto hay el mismo número de imágenes de cada clase.

Debido a que se va a realizar clasificación y detección es necesario, que a partir del mismo conjunto se etiqueten las imágenes de diferente manera generando dos datasets. Para la tarea de clasificación únicamente es necesario indicar para cada imagen una clase. En cambio, para la detección se necesita crear un polígono indicando donde se encuentra la caja del medicamento en la imagen. Para la creación del polígono se ha utilizado en todas las imágenes un rectángulo. Se ha utilizado la herramienta web Roboflow para identificar las cajas de los medicamentos en las imágenes. En la Figura 12 se muestra un ejemplo de este proceso.

Figura 12: Etiquetado de imágenes

Para el entrenamiento se ha utilizado el framework TensorFlow Lite en el lenguaje de programación Python. Este framework permite transformar modelos de TensorFlow para ser utilizados en dispositivos de bajos recursos, como dispositivos móviles. Además de esta función, este framework permite entrenar directamente modelos con las características necesarias para implementarse directamente a los dispositivos. En este caso, se ha elegido esta última funcionalidad para entrenar los modelos de clasificación y detección de objetos.

En el proceso de entrenamiento, se han entrenado dos modelos pre-entrenados de clasificación y otros dos de detección de objetos con el objetivo de realizar una comparación de los resultados obtenidos con los diferentes modelos y elegir la mejor opción para nuestra aplicación. Los modelos elegidos para la clasificación han sido MobileNetV2 y ResNet50 debido a que en la revisión de la literatura numerosos artículos argumentan que dichos modelos fueron diseñados específicamente para la clasificación de imágenes en dispositivos con recursos limitados obtienen buenos resultados. En la tarea de la detección de los objetos se han elegido EfficientDet-Lite0 y EfficientDet-Lite2 debido a que el framework ofrece 5 modelos EffecientDet-Lite, desde el 0 hasta el 4, progresivamente cada uno de los modelos es de mayor tamaño, por lo tanto tiene una mejor precisión pero por otro lado tarda más en realizar la inferencia. Estos datos se pueden observar en el Cuadro 9.

Arquitectura del modelo	Tamaño(MB)	Latencia	Precisión media
EfficientDet-Lite0	4,4	37	25,69 %
EfficientDet-Lite1	5,8	49	30,55 %
EfficientDet-Lite2	7,2	69	33,97 %
EfficientDet-Lite3	11,4	116	37,70 %
EfficientDet-Lite4	19,9	260	41,96 %

Cuadro 9: Datos EfficientDet-Lite

El valor del tamaño hace referencia al tamaño de los modelos cuantificados a enteros, se transforma de float a entero para reducir el coste computacional en detrimento de la precisión. La latencia está medida en un Pixel 4 utilizando 4 hilos de su CPU. Por último, la precisión media es la mAP (mean Average Precision) obtenida en el conjunto de validación del conjunto de datos COCO 2017.

Para este caso de estudio se han elegido los modelos EfficientDet-Lite0 debido a que se desea obtener el resultado de la inferencia de la imagen con la menor latencia posible debido a que esta se realiza en tiempo real. Se ha elegido EfficientDet-Lite2 en vez de EfficientDet-Lite1 para comprobar si es posible una precisión mejor respecto a su versión 0 sin aumentar en exceso la latencia.

Para realizar el entrenamiento se han utilizado las clases *image_classifier* y *object_detector* del módulo *tflite-model-maker* para el entrenamiento del clasificador y el detector, respectivamente. La configuración del entrenamiento para los 4 modelos ha sido *batch_size=4*, *train_whole_model=True* y *epochs=100*.

4.2. Generación de respuestas

La generación de respuestas a una pregunta se lleva a cabo a través de diferentes técnicas PLN. Las respuestas generadas pueden estar orientadas a diferentes objetivos según el caso de estudio. Puede desearse que la generación se lleve a cabo sin ningún contexto (de dominio abierto o cerrado), sea de elección entre diferentes opciones, dado un contexto la respuesta se aporte indicando el carácter inicial y final, o como es el caso de este estudio, la respuesta se genera de manera automática dado un contexto.

El contexto que se aporta al modelo es el prospecto del medicamento previamente detectado con el objetivo de que el usuario obtenga la respuesta más fiable posible. El prospecto de los medicamentos incluye toda la información necesaria para resolver las posibles dudas de los consumidores. Todos los prospectos de los medicamentos españoles están divididos en 6 secciones:

- Qué es y para qué se utiliza (Sección 1)
- Qué necesita saber antes de empezar a tomar (Sección 2)
- Cómo tomar (Sección 3)

- Posibles efectos adversos (Sección 4)
- Conservación (Sección 5)
- Contenido del envase e información adicional (Sección 6)

Para generar la respuesta es necesario que el modelo procese la pregunta y su respectivo contexto, debido a esto cuanto más largo sea el contexto más tiempo tardará el modelo en proporcionar la respuesta y es posible que sea más difícil de encontrarla correctamente. Para resolver esta problemática, en vez de introducir el prospecto completo del medicamento como contexto, se introduce únicamente la sección del prospecto donde se encuentra la respuesta a la pregunta del usuario.

Para realizar este proceso, se ha decidido realizar dos conjuntos de datos diferentes. El primero está encargado en determinar en qué sección se encuentra la respuesta a la pregunta realizada. El segundo dataset está orientado a generar la respuesta a la pregunta teniendo en cuenta la sección del prospecto que ha determinado el paso anterior. Este proceso se puede llevar a cabo debido a que en cada sección de todos los prospectos se indica una información similar, aunque en diferentes formatos.

Para obtener resultados que se consideren válidos, es necesario que el conjunto de datos sea suficientemente grande y variado. Por ello, en el dataset que está orientado a determinar en qué sección del prospecto se encuentra la respuesta a la pregunta se han incluido 3202 preguntas junto a su respuesta correspondiente, en este caso la sección. Inicialmente, se enunciaron 29, 33, 39, 32, 18, 22 preguntas para cada sección respectivamente. Estas preguntas han sido elegidas para responder la totalidad o gran parte de las posibles preguntas en cada una de las secciones.

Debido a que este número de preguntas es muy reducido y era necesario aumentarlo de manera acentuada se decidió utilizar la herramienta ChatGPT3.5. Para realizar esta tarea se han utilizado dos prompts diferentes. El primero permite reescribir de manera rápida una pregunta de diferentes maneras, se introdujo el siguiente prompt: *"Escríbeme de 20 maneras diferentes, unas más simples y otras más complejas, como si fueran personas diferentes la siguiente pregunta: X"*, siendo X una de las preguntas del conjunto de datos. El segundo tipo de prompt permite modificar ligeramente la pregunta. Un ejemplo de este tipo es: *"Escríbeme de 20 maneras diferentes como si fueran personas diferentes la siguiente pregunta. Cambia el sueño por otras dolencias diferentes para tener más casos: La medicación me da mucho sueño, ¿es normal?"*. De esta manera se obtienen, a través de una única pregunta, numerosos casos diferentes.

Es necesario destacar que tras realizar cada una de las peticiones a ChatGPT3.5 se realizó una revisión de las preguntas devueltas y se eliminaron las consideradas como erróneas.

Tras realizar este proceso se aumenta el número de preguntas de 173 a las 3202 preguntas obtenidas finalmente. Cada sección tiene 577, 639, 712, 528, 345 y 401 preguntas respectivamente, ordenadas de primera a la última sección.

Para realizar el proceso de entrenamiento, se va a seguir la división del conjunto

de datos en los tres conjuntos mencionados en el apartado anterior. Estas preguntas pertenecen al conjunto de entrenamiento y validación. Con el objetivo de reducir el sesgo producido si la misma persona realiza tanto el conjunto de entrenamiento como el de prueba, para el conjunto de pruebas se ha solicitado a 5 usuarios que realicen diferentes preguntas que pueden tener acerca de un medicamento. Después de recopilar las preguntas de todos los participantes, se obtuvieron 130 preguntas dividas en 20, 17, 28, 24, 23 y 19 en cada sección.

Tras realizar el primer conjunto de datos, el siguiente paso es realizar el conjunto de datos relativo a responder a la pregunta del usuario. Para esta tarea se ha escrito la respuesta a las preguntas del primer conjunto de datos, eliminando ciertas preguntas por su similitud y repetición, dando un total de 8989 preguntas. Las respuestas han sido escritas para los medicamentos "Nolotil 575mg", "Paracetamol Kern Pharma 1 g comprimidos EFG" y "Adiro 100 mg comprimidos gastrorresistentes EFG", los tres medicamentos más vendidos en España en el año 2022. Las preguntas han sido respondidas buscando la máxima similitud respecto al prospecto original, en algunos casos en los cuáles la respuesta no se señala directamente en el prospecto se indica de esta manera en la respuesta. De la misma manera que en el anterior conjunto de datos, estas preguntas engloban el conjunto de entrenamiento y validación.

Respecto al conjunto de datos de prueba se han utilizado las mismas 130 preguntas que en el anterior dataset. Estas preguntas han sido respondidas para los tres medicamentos que se incluyen en el conjunto de entrenamiento y validación. Además, se ha escrito la respuesta para los medicamentos "Enantyum 25 mg comprimidos recubiertos con película" y "Eutirox 88 microgramos comprimidos" para comprobar si el modelo obtenido tras el entrenamiento generaliza para medicamentos que no han sido utilizados en el conjunto de datos. Estos medicamentos han sido elegidos debido a que "Enantyum" y "Eutirox" son el cuarto y sexto medicamento más vendidos. No se ha utilizado "Paracetamol Cinfa" debido a su similitud con "Paracetamol Kern Pharma 1 g comprimidos EFG" ya que el objetivo de incluir medicamentos no utilizados durante el entrenamiento es observar como genera respuestas ante medicamentos nuevos y diferentes.

Debido al alto coste computacional de un entrenamiento de procesamiento del lenguaje natural con numerosos datos, el entrenamiento se ha realizado en un *NVIDIA Jetson AGX Orin 64GB Developer Kit*, un dispositivo orientado la robótica y la inteligencia artificial.

El proceso de entrenamiento de un modelo PLN, normalmente, tiene dos fases. La primera se denomina aprendizaje no supervisado, que tiene como entrada una gran cantidad de texto sin etiquetar, es decir, no se aporta una salida a la entrada, para que el modelo identifique patrones en el texto. Por otro lado, la segunda fase se denomina aprendizaje supervisado, el texto es etiquetado y se aporta una salida a cada entrada. En esta fase el modelo se entrena en una tarea particular, en este caso, la generación de respuestas.

Debido a que no se posee acceso a una gran cantidad de datos para realizar el aprendizaje no supervisado se ha decidido utilizar diferentes modelos pre-entrenados encontrados en la plataforma Hugging Face[53], donde la comunidad publica sus

modelos y datasets para ser utilizados por el resto de usuarios. Tras realizar una búsqueda intensiva, se han encontrado dos modelos del mismo usuario publicados en [54].

El primer modelo se llama *vgaraujov/t5-base-spanish* y su acrónimo es T5S, es un modelo que intenta mejorar al modelo original T5. Es un modelo con un codificador bidireccional, como BERT, y un decodificador autoregresivo, como GPT. Este modelo necesita realizar un fine-tuning, entrenamiento supervisado, para una tarea determinada. Es posible utilizarlo para diferentes tareas como resumir, traducir, clasificar texto o responder preguntas, como en el caso de estudio.

El segundo modelo está publicado con el nombre *vgaraujov/led-base-16384-spanish* y su acrónimo es LEDO. Está basado en el modelo del mismo autor *vgaraujov/bart-base-spanish*, un modelo basado en el modelo BART. LEDO mejora a este modelo en la generación de texto para entradas largas, por este motivo se ha elegido esta versión debido a que el contexto de la pregunta, la sección del prospecto, contiene una gran cantidad de caracteres.

Ambos modelos han realizado el aprendizaje no supervisado con los mismos conjuntos datos, también disponibles en Hugging Face. Estos datasets están publicados con los nombres *oscarcorpus/OSCAR2109*, *bertin-project/mc4essampled* y *josecannete/large_spanish_corpus*.

Para realizar el fine-tuning de los modelos se ha utilizado el módulo transformers disponible en el lenguaje de programación Python. Para realizar el entrenamiento correctamente es necesario introducir: una instrucción, una entrada e indicar su salida.

La instrucción en el conjunto de datos para encontrar la sección es la siguiente: *"Elige textualmente el título que responde a la pregunta: X"*, donde X es cada una de las 3202 preguntas. La entrada es *""Qué es y para qué se utiliza", "Qué necesita saber antes de empezar a tomar", "Cómo tomar", "Posibles efectos adversos", "Conservación" y "Contenido del envase e información adicional""*. La salida indicada es la sección en la que se encuentra la respuesta para cada pregunta.

En el conjunto de datos encargado de generar la respuesta a la pregunta, la instrucción es: *"Responde a la siguiente pregunta: "X""*, donde X es cada una de las preguntas. La entradas es *"El contexto es el siguiente: Y"*, siendo Y la sección donde se ha determinado previamente que se encuentra la respuesta. Para finalizar, la salida indicada es la respuesta que se ha definido para cada pregunta.

Antes de introducir los datos al modelo para realizar el entrenamiento es necesario dividir el dataset en los conjuntos de datos de entrenamiento y validación. Para ello, se ha determinado de manera aleatoria, con el módulo Pandas, que el 80 % de los datos pertenezcan al entrenamiento y el 20 % restante a la validación.

Los parámetros de entrenamiento han sido los mismos para todos los casos. Se ha utilizado la clase *AutoModelForSeq2SeqLM* y la clase *AutoTokenizer* para cargar el modelo pre-entrenado y el tokenizador respectivamente. La clase *DataCollatorForSeq2Seq* la cual tiene como parámetros dicho modelo y tokenizador. Para indicar

los parámetros del entrenamiento se ha utilizado la clase *Seq2SeqTrainingArguments* cuyos parámetros son *evaluation_strategy = "epoch"*, *learning_rate=2e5*, *per_device_train_batch_size=2*, *per_device_eval_batch_size=2*, *gradient_accumulation_steps=2*, *num_train_epochs=8* y *push_to_hub=True*. Para finalizar se crea un objeto de la clase *Seq2SeqTrainer* para entrenar los datos.

5. Resultados

Tras explicar la metodología para el entrenamiento tanto para el modelo de identificación de los medicamentos como de la generación de la respuesta para las preguntas acerca de los mismos, en esta sección se detallan los resultados obtenidos tras los entrenamientos y se eligen los mejores modelos obtenidos argumentando su elección. Esta sección se va a dividir, de la misma manera que en la anterior, en las dos partes que componen el caso de estudio, la identificación del medicamento y por otra parte la generación de las respuestas.

5.1. Detección de medicamentos

Las imágenes usadas son las del conjunto de prueba, unas imágenes que no han sido utilizadas durante el entrenamiento para comprobar que los modelos obtenidos identifican los medicamentos de manera generalizada. Además, estas imágenes han sido seleccionadas debido a que tienen un fondo o iluminación diferentes a las imágenes de entrenamiento para aumentar la seguridad de comprobar si los modelos identifican o no correctamente los medicamentos en imágenes nuevas y en diferentes ambientes.

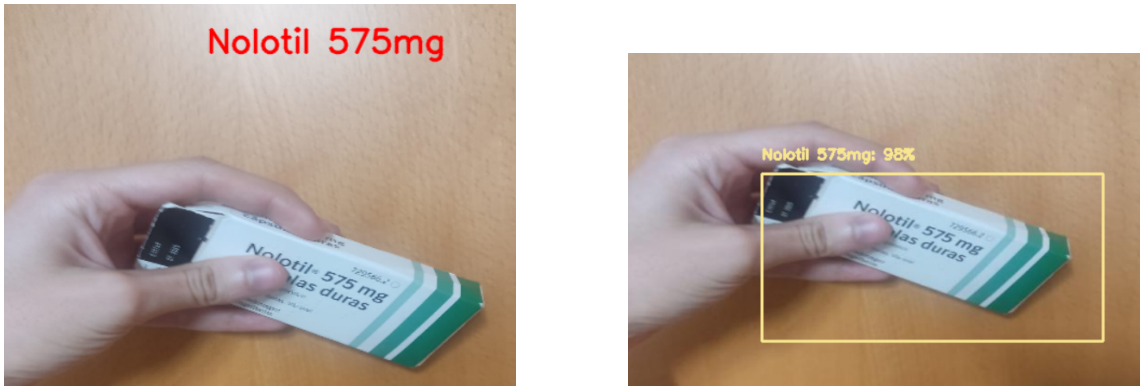
Para realizar esta tarea, la inferencia no se ha realizado en un dispositivo móvil sino en un ordenador debido a las bibliotecas existentes en lenguaje de programación Python que facilitan estos cálculos. Para ello, se han cargado los modelos .tflite obtenidos para el entrenamiento y se realiza la inferencia para todo el conjunto de imágenes de prueba.

Para la clasificación, la salida del modelo al introducir una imagen es un vector de dimensión 5, un valor para cada clase. La suma de los valores del vector es 255 y cada valor del vector representa la confianza que tiene cada clase de ser identificada en la imagen. Por ello, el valor más alto de este vector representa a la clase que se ha identificado en la imagen. En la Figura 13a se observa el resultado de la clasificación para una imagen del conjunto de prueba.

En el caso de la detección del medicamento, el modelo devuelve 4 valores, uno que indica el número de clases detectados, el segundo representa la puntuación de cada clase, el tercero indica las clases identificadas y el último las coordenadas, en forma de rectángulo, de cada objeto detectado. Debido a que en este caso de estudio solo se desea identificar a un medicamento en cada imagen, solo se obtiene la clase cuyo valor de puntuación es mayor. En la Figura 13b se muestra la misma imagen que en 13a pero en este caso se representa la localización de la caja del medicamento detectado en vez de la clasificación.

Para determinar la calidad de los modelos obtenidos, se van a utilizar las métricas descritas en la sección del estado del arte debido a que son las más utilizadas en este campo de la visión artificial. Estas métricas son *accuracy*, *precision*, *recall* y *F1 score*. La métrica *accuracy* se va a representar en forma de matriz de confusión, ya que permite observar si el modelo es tendencioso hacia alguna de las clases. En las

Figuras 14, 15, 16, 17 se muestran las diferentes matrices de confusión obtenidas tras introducir las imágenes del conjunto de prueba a través de cada uno de los modelos.



(a) Clasificación Medicamento (b) Detección Medicamento

Figura 13: Clasificación y detección de medicamentos

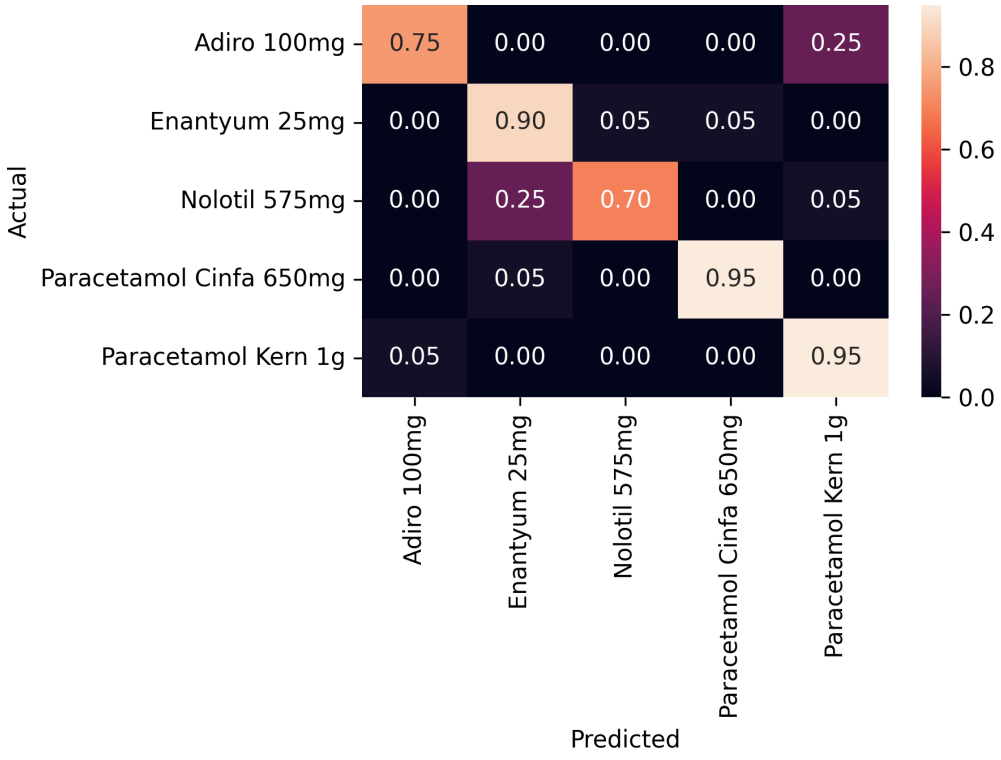


Figura 14: Matriz de confusión MobileNetV2

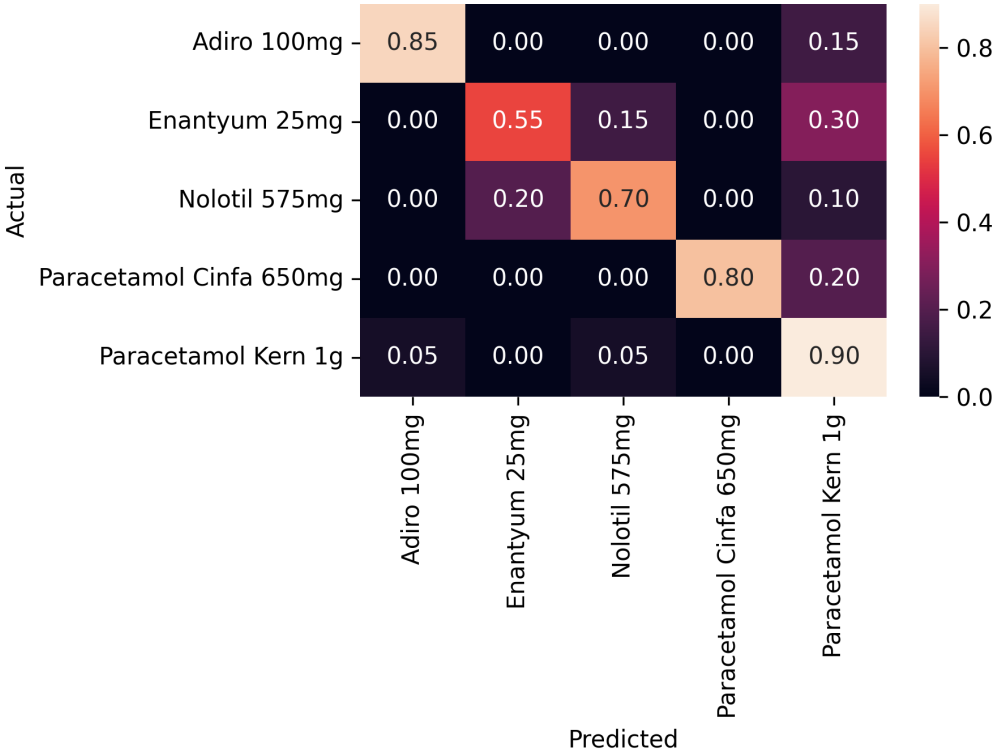


Figura 15: Matriz de confusión ResNet50

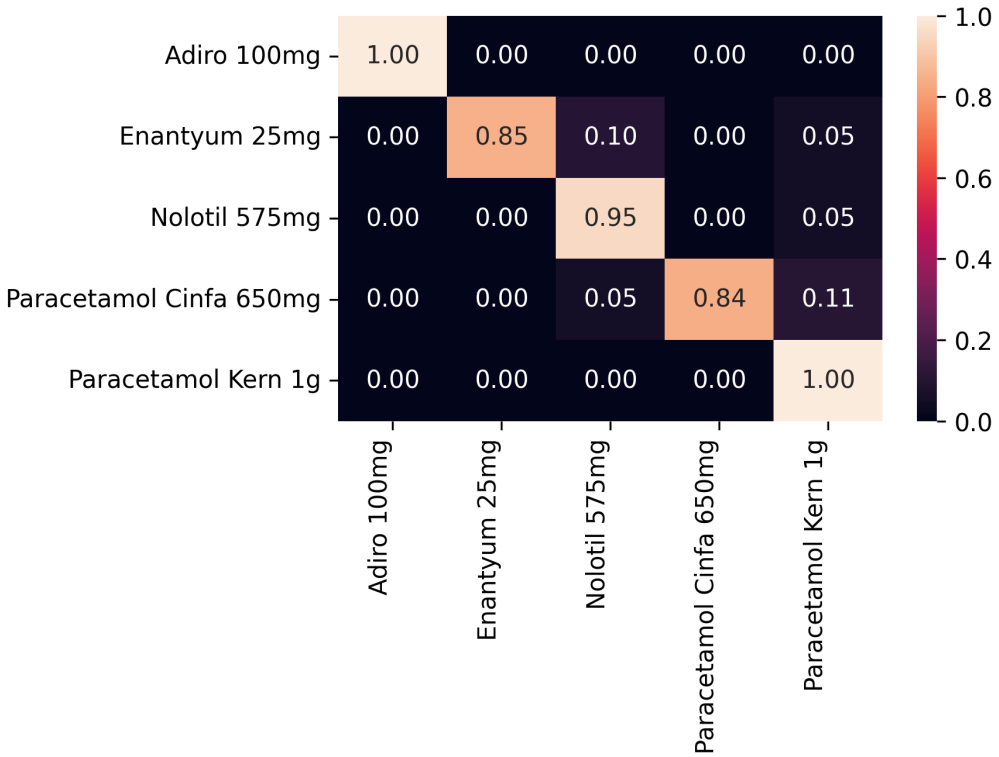


Figura 16: Matriz de confusión EfficientNet-Lite0

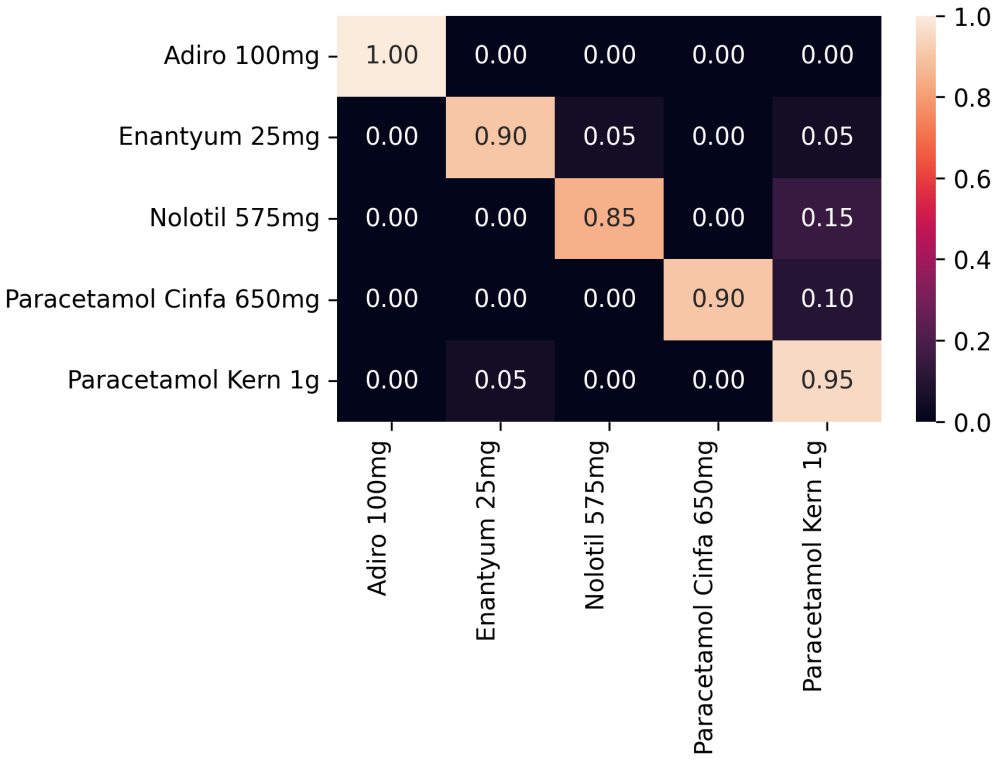


Figura 17: Matriz de confusión EfficientNet-Lite2

Además, un factor importante para la elección del modelo es la latencia y el espacio que ocupan. La latencia se ha medido en un Xiaomi Poco x3 cuyo procesador es Qualcomm Snapdragon 855 Plus. En el Cuadro 10 se observa la latencia media de inferencia y el espacio que ocupa cada modelo.

Modelo	Latencia ms	Espacio MB
MobilenetV2	23	2,72
Resnet50	87	23,81
EfficientDet-Lite0	65	4,35
EfficientDet-Lite2	150	7,22

Cuadro 10: Latencia y espacio modelos

Observando los resultados obtenidos, respecto a los modelos de clasificación, el modelo MobileNetV2 obtiene ligeramente mejores resultados que el modelo ResNet50 y además en un tiempo 3,78 veces menor. En cuanto a los modelos de detección, ambos obtienen resultados muy similares pero el modelo EfficientDet-Lite0 es 2,3 veces más rápido que el modelo EfficientDet-Lite2 por lo que en relación a estos modelos se elige el primero como mejor opción.

Además de las matrices de confusión, se han calculado los valores de *precision*, *recall* y *F1 score* para cada clase y modelo. Estos valores se muestran en el Cuadro 11, Cuadro 12 y Cuadro 13, respectivamente.

Modelo	Adiro	Enantyum	Nolotil	Paracetamol Cinfa	Paracetamol Kern
MobilenetV2	0,94	0,75	0,93	0,95	0,76
Resnet50	0,94	0,73	0,78	1	0,55
EfficientDet-Lite0	1	1	0,86	1	0,83
EfficientDet-Lite2	1	0,95	0,94	1	0,76

Cuadro 11: Precision

Modelo	Adiro	Enantyum	Nolotil	Paracetamol Cinfa	Paracetamol Kern
MobilenetV2	0,75	0,9	0,7	0,95	0,95
Resnet50	0,85	0,55	0,7	0,8	0,9
EfficientDet-Lite0	1	0,85	0,95	0,8	1
EfficientDet-Lite2	0,95	0,9	0,85	0,9	0,95

Cuadro 12: Recall

Modelo	Adiro	Enantyum	Nolotil	Paracetamol Cinfa	Paracetamol Kern
MobilenetV2	0,83	0,82	0,8	0,95	0,84
Resnet50	0,89	0,63	0,74	0,89	0,68
EfficientDet-Lite0	1	0,92	0,91	0,89	0,91
EfficientDet-Lite2	0,97	0,92	0,89	0,95	0,84

Cuadro 13: F1-score

Para finalizar la selección del modelo a utilizar en la aplicación, se comparan el mejor modelo de clasificación y de detección. La métrica más importante es *F1 score* y en esta, el modelo EfficientDet-Lite0 obtiene notablemente mejores resultados en todos los medicamentos excepto en "Paracetamol Cinfa". A pesar de que el modelo EfficientDet-Lite0 obtiene mejores resultados, se determina que el modelo a implementar en la aplicación es el modelo MobileNetV2 debido a que este es 2,83 veces más rápido y se busca que la aplicación pueda ser utilizada en el mayor número posible de dispositivos y podemos concluir que los resultados obtenidos por el modelo propuesto, pueden ser considerados aceptables.

5.2. Generación de respuestas

En la búsqueda de la mejor implementación del modelo se han llevado a cabo entrenamientos con los diferentes conjuntos de datos para observar sus resultados. Para cada uno de los dos modelos pre-entrenados, T5S y LEDO, se han realizado tres diferentes. El primero únicamente presenta el conjunto de datos de elección de la sección del prospecto. El segundo, contiene el conjunto de datos de la generación de la respuesta. Por último, el tercer modelo se ha entrenado con los dos conjuntos de datos simultáneamente.

El objetivo del estudio es conseguir un único modelo que tenga la capacidad de realizar las dos tareas, la determinación de la sección y la generación de la respuesta, de manera correcta pero se ha llevado a cabo este proceso para comprobar si se obtienen mejores resultados si se entrenan dos modelos especializados en cada una de las tareas en vez de un único modelo. A su vez, para comprobar las diferencias entre los 3 modelos obtenidos de cada modelo pre-entrenado.

Para obtener los resultados de los modelos obtenidos se ha utilizado el conjunto de datos de prueba enunciado en la sección anterior. Los modelos a entrenar son generativos y no deterministas, es decir, para una misma entrada el modelo puede producir resultados diferentes. Por este motivo se ha decidido, tanto para la elección de la sección como para la generación de la respuesta, que el modelo no genere una única salida para una entrada si no 3, es decir, para una única pregunta el modelo genera 3 respuestas, con el objetivo de obtener unos resultados más robustos.

Pese a que los modelos obtenidos devuelven las respuestas de manera generativa, en el caso de la elección de la sección es posible determinar las salidas como clases, cada sección representa una clase y se crea otra más si la salida del modelo no se refiere a ninguna sección. Por esto, los resultados de esta tarea se presentan como una matriz de confusión.

En la Figura 18, Figura 19, Figura 20 y Figura 21 se presentan las matrices de confusión obtenidas respecto a los modelos tanto especializados en la elección de la sección como generales de los dos modelos pre-entrenados. El orden de las secciones es el presentado en la anterior sección cuando se presentan las secciones que contienen todos los prospectos españoles.

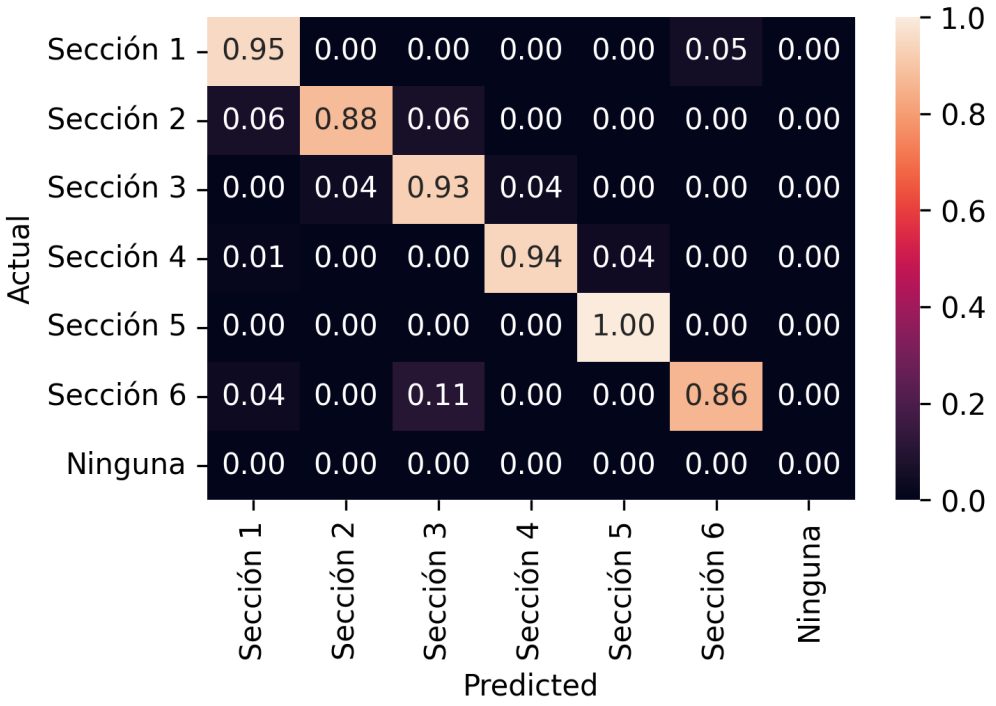


Figura 18: Matriz de confusión especializada en sección T5S

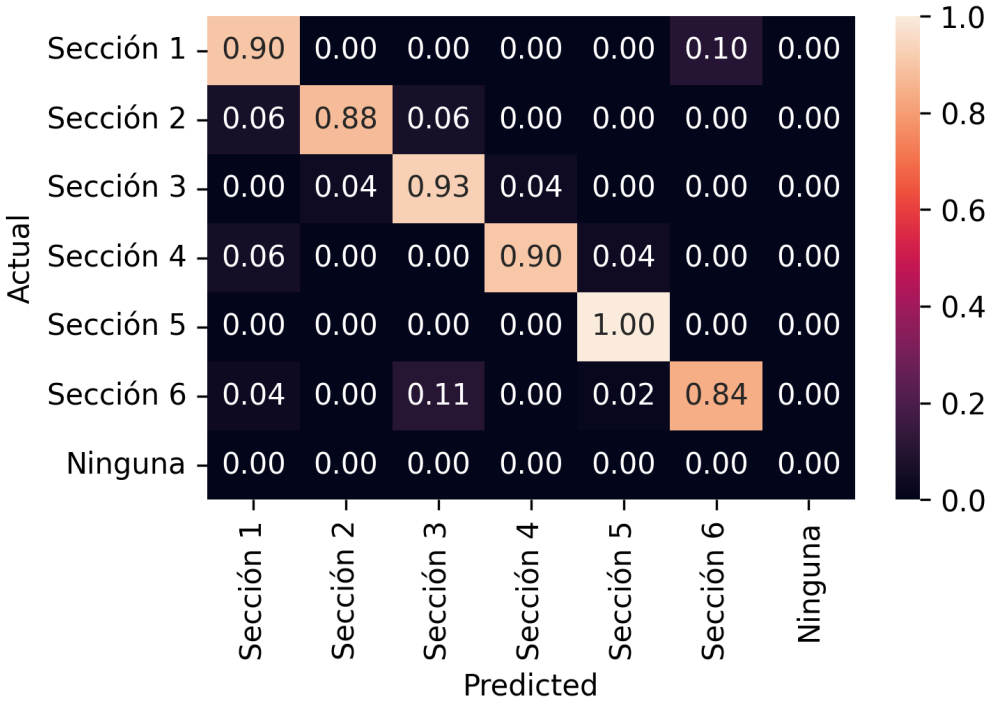


Figura 19: Matriz de confusión General T5S

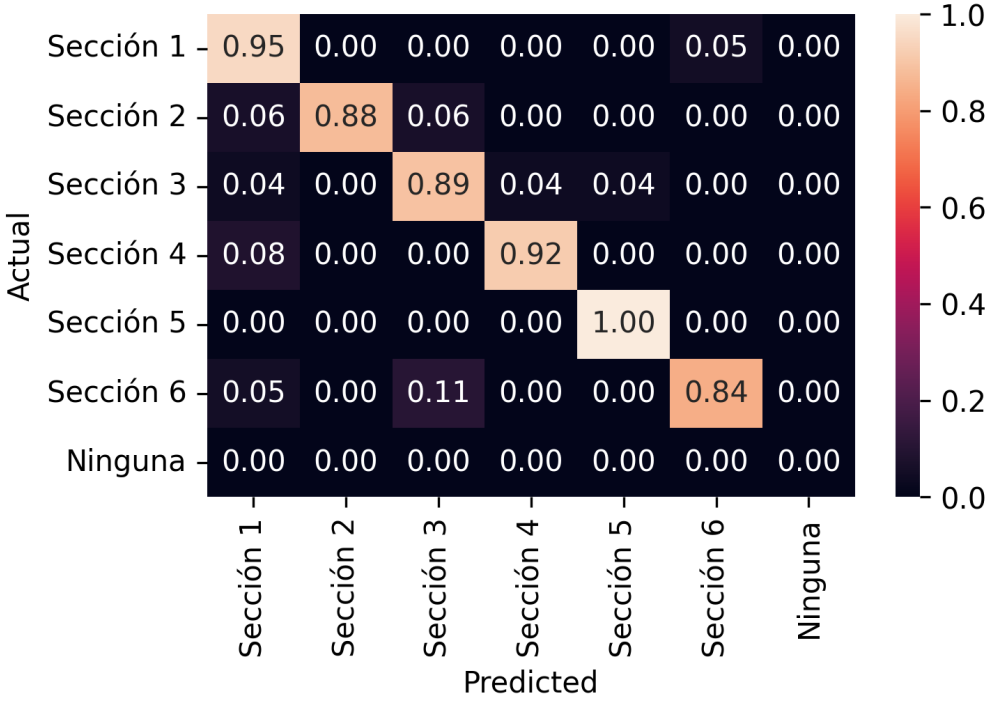


Figura 20: Matriz de confusión especializada en sección LEDO

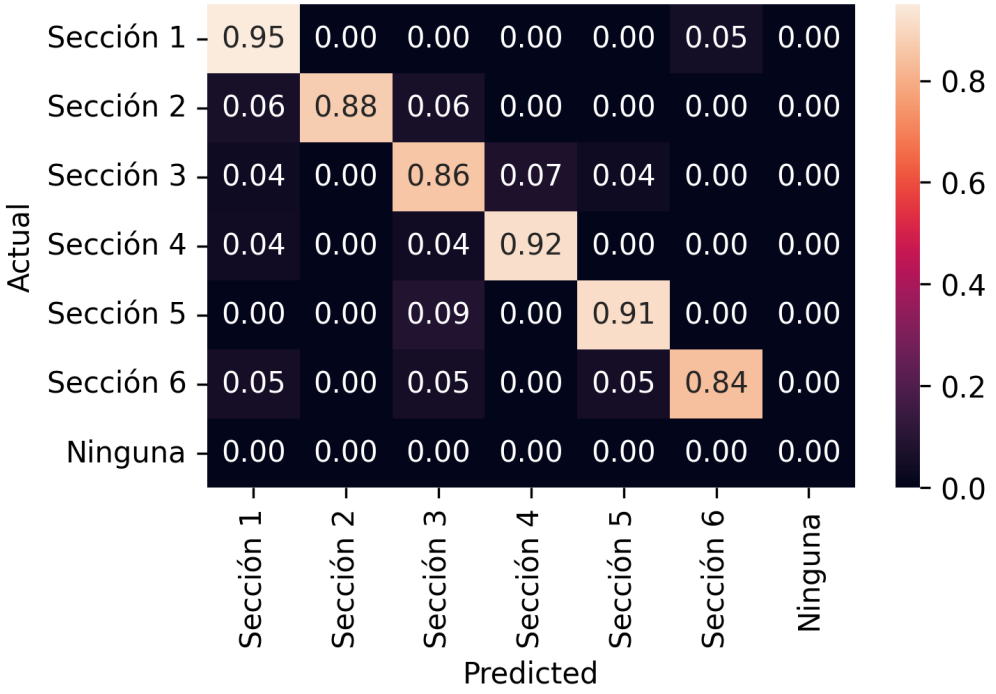


Figura 21: Matriz de confusión General LEDO

Tras analizar los resultados de las cuatro matrices de confusión, se puede concluir que todas proporcionan resultados similares en cada una de las secciones. Los cuatro modelos no generan ninguna respuesta que no se refiera a alguna sección. Además, las secciones con mejores resultados son: *"Qué es y para qué se utiliza"*, y *"Conservación"* mientras que la sección que peores resultados tiene *"Contenido del envase e información adicional"*.

Debido a que los cuatros modelos proporcionan resultados válidos y similares para todas las clases, es necesario analizar los resultados en la generación de las respuestas para decidir que modelo elegir.

La validez de los resultados acerca de las respuestas generadas se puede comprobar utilizando métricas como ROUGE o BLEU, métricas que muestran la similitud entre la respuesta generada y la objetivo, normalmente utilizadas en PLN en las tareas de traducir o resumir texto. Además, debido a que el caso de estudio es del ámbito médico, es necesario comprobar las salidas ya que el valor de estas métricas podría ser ciertamente alto pero el significado no corresponder al deseado. Por esto, además, de las métricas mencionadas, se utiliza la revisión humana como métrica, comprobando las respuestas generadas y determinando si la salida es válida o no.

En el cuadro 14, se presentan los resultados de la generación de las respuestas en porcentaje de acierto. La sintaxis en la tabla es "Valor1/Valor2", el Valor1 representa el porcentaje de acierto respecto a los medicamentos entrenados mientras que el Valor2 representa el porcentaje de acierto respecto a los medicamentos no entrenados. Las secciones se han numerado en el orden presentado en la sección anterior.

Modelo	Sección 1	Sección 2	Sección 3	Sección 4	Sección 5	Sección 6	Total
T5S Respuesta	98 %/23 %	83 %/41 %	86 %/57 %	58 %/17 %	94 %/17 %	81 %/66 %	83 %/37 %
T5S General	97 %/23 %	85 %/47 %	85 %/54 %	54 %/13 %	97 %/17 %	89 %/55 %	84 %/34 %
LEDO Respuesta	95 %/83 %	75 %/56 %	79 %/45 %	61 %/23 %	88 %/30 %	75 %/61 %	79 %/48 %
LEDO General	97 %/78 %	83 %/63 %	86 %/52 %	67 %/19 %	90 %/17 %	77 %/79 %	83 %/49 %

Cuadro 14: Porcentaje de acierto

En el Cuadro 15 se muestra los resultados de las métricas de ROUGE y BLEU, la sintaxis es la misma que en la anterior tabla.

Modelo	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
T5S Respuesta	0,74/0,41	0,7/0,32	0,73/0,39	0,67/0,29
T5S General	0,74/0,42	0,7/0,34	0,73/0,4	0,67/0,3
LEDO Respuesta	0,78/0,46	0,75/0,38	0,78/0,44	0,72/0,32
LEDO General	0,77/0,44	0,73/0,36	0,76/0,41	0,69/0,28

Cuadro 15: Métricas ROUGE y BLEU

Observando los resultados del Cuadro 14 que indica el porcentaje de acierto en cada sección se obtienen diferentes conclusiones. En cuanto a la diferencia entre las secciones, la sección 4, *"Posibles efectos adversos"*, es la que obtiene peores resultados. Esto es debido a que es la sección que más diferencias presenta entre los distintos prospectos tanto en contenido como en formato. La sección que obtiene mejores resultados es la sección 1, *"Qué es y para qué se utiliza"*, una de las posibles razones es porque es la sección con menos texto en todos los prospectos.

Se comprueba que los resultados para los 4 modelos, obtienen un alto porcentaje de acierto para los medicamentos que se han utilizado en el entrenamiento. Los resultados para estos medicamentos son muy similares, siendo ligeramente mejores en los modelos obtenidos de T5S. Respecto a los resultados obtenidos en los medicamentos no utilizados durante el entrenamiento, se produce un gran descenso de porcentaje de acierto. Este descenso es mayor en los modelos de T5S, que obtienen un 37 % y 34 % de acierto en total. En el caso de los modelos obtenidos de LEDO se obtienen unos resultados mejores, un acierto del 48 % y 49 % del total de las preguntas.

A pesar de que los modelos obtenidos tras realizar fine-tuning a T5S obtienen mejores resultados en los medicamentos que han sido utilizados para el entrenamiento, sus resultados para los que no han sido utilizados en el entrenamiento indican que no generaliza correctamente. Por este motivo, se elige los modelos obtenidos tras realizar fine-tuning al modelo LEDO ya que generalizan de una mejor manera. Entre los dos modelos obtenidos, el que proporciona mejores resultados es el que ha sido entrenado tanto para la elección de la sección como la generación de respuestas.

Tras analizar las diferentes métricas tanto en la elección de la sección como en la generación de respuestas, el modelo elegido para ser utilizado en la aplicación es el que sirve para ambas partes obtenido tras realizar el fine-tuning del modelo LEDO.

6. Sistema propuesto

En este apartado se detalla el proceso de implementación de la aplicación de dispositivos móviles que permite utilizar los modelos de *machine learning* obtenidos.

La aplicación ha sido implementada en Android Studio en el lenguaje de programación Kotlin. Esta aplicación únicamente está disponible para los dispositivos móviles con Android como sistema operativo.

La aplicación móvil implementada permite reconocer diferentes medicamentos a través de la identificación de las cajas mediante el procesamiento de imágenes en tiempo a real, proporcionadas por la cámara del dispositivo. El reconocimiento de las cajas de los medicamentos se realiza a través del modelo de *machine learning* propuesto utilizando el framework TensorFlow Lite, que permite aplicar el modelo en el propio dispositivo.

Tras llevar a cabo la identificación del medicamento, la aplicación permite realizar preguntas acerca del mismo. Estas preguntas son respondidas usando PLN mediante el modelo de generación de respuestas obtenido, que recibe como entrada la pregunta, el prospecto del medicamento como contexto y genera una respuesta teniendo en cuenta ambas partes. Se puede observar el esquema del funcionamiento de la aplicación en la Figura 22.

Figura 22: Esquema de la aplicación

La primera parte de la aplicación consiste en la búsqueda del medicamento del que se desea buscar información. En ella se muestran dos botones correspondientes a las dos opciones de búsquedas posibles: la primera utilizando el modelo de visión artificial que hemos creado, a través imágenes proporcionadas por la cámara del dispositivo. La segunda, introduciendo su nombre, utilizando una caja de texto o la voz, para buscar cualquier medicamento en español.

En la primer opción, se muestran en la pantalla las imágenes obtenidas por la cámara y se indica, a través de un texto, el proceso de identificación del medicamento se produce en tiempo real, debido a que en un determinado frame, el medicamento detectado puede no ser el correcto, se ha implementado un flujo de programación que aumenta el porcentaje de acierto del medicamento identificado. Se puede observar

en el Algoritmo 1.

Algoritmo 1: Detección Medicamento

Salida : Medicamento detectado

```
1  contador = 0;
2  medicamento = “” ;
3  while True do
4  |  imagen = obtenerImagenCamara();
5  |  imagenProcesada = procesarImagen(imagen);
6  |  medicamentoDetectado = inferenciaModelo(imagenProcesada);
7  |  if medicamentoDetectado == medicamento then
8  |  |  if contador == 10 then
9  |  |  |  return medicamento;
10 |  |  end
11 |  |  contador += 1;
12 |  end
13 |  else
14 |  |  contador = 0;
15 |  |  medicamento = medicamentoDetectado
16 |  end
17 end
```

Este algoritmo se basa en que es necesario detectar el mismo medicamento durante 10 inferencias del modelo de manera seguida. De esta forma, el medicamento detectado finalmente tendrá mucha más probabilidad de que sea el correcto en vez de utilizar una única imagen para realizar la identificación.

Tras devolver el medicamento, la aplicación muestra una alerta indicando el medicamento detectado y permitiendo al usuario elegir si es el medicamento deseado o no. Si el usuario pulsa la opción afirmativa, la aplicación cambia a la pestaña siguiente, mientras que si elige la opción negativa, comienza de nuevo la búsqueda del medicamento. Las personas con discapacidad visual pueden realizar correctamente esta tarea debido a que las cajas de los medicamentos contienen el nombre del mismo en lenguaje braille por lo que son capaces de conocer si la aplicación ha detectado correctamente o no la caja.

La segunda opción de la aplicación para la búsqueda del medicamento consiste en un buscador, tanto por texto como por voz. Se realiza cuando el usuario pulsa el botón de búsqueda una vez que escribe en la caja de texto o cuando ha terminado de hablar pulsando el botón del micrófono. La aplicación realiza una petición a CIMA, centro de información online de medicamentos de la Agencia Española de Medicamentos y Productos Sanitarios (AEMPS). La petición devuelve los medicamentos españoles que contienen la cadena introducida por el usuario. Por ejemplo, si el usuario escribe "paracetamol", la aplicación muestra la lista de medicamentos que contienen esta cadena. Cada uno de los elementos es un botón que, al ser pulsado por el usuario, la aplicación cambia de pestaña recogiendo la información del medicamento elegido.

En la segunda parte de la aplicación, se muestran las secciones que componen todos los prospectos de los medicamentos españoles. Cada una de las secciones es un botón, si el usuario pulsa uno de ellos se muestra en pantalla la sección seleccionada con el mismo formato que en el prospecto físico. Además, la aplicación permite realizar zoom para que los usuarios que no puedan leer las letras pequeñas, puedan realizar esta tarea más fácilmente.

En esta pantalla, además de mostrar el prospecto íntegro, es posible realizar preguntas. Estas preguntas se pueden realizar de la misma manera que en la búsqueda de medicamentos, tanto por una caja de texto como por el micrófono del dispositivo al pulsar el botón correspondiente. Al hacer la pregunta, se realiza una primera petición al servidor creado, donde se aloja el modelo PLN, que devuelve la sección en la que se encontrará la respuesta en el prospecto, a continuación se realiza una segunda petición en la que se incluye la sección del prospecto indicado por la anterior petición. Esta segunda petición devuelve la respuesta generada por el modelo de *machine learning* y es leída al usuario por un módulo de Text-To-Speech (TTS). Si en la primera petición no se asocia la pregunta a ninguna sección, la aplicación muestra una alerta indicando al usuario el problema.

Debido a que la aplicación está destinada tanto a las personas que no son capaces de ver las letras pequeñas de los prospectos como a las personas con discapacidad visual, es necesario desarrollar la aplicación accesible para ambos tipos de personas. En el sistema operativo Android existe una opción de configuración de accesibilidad para las personas con discapacidad llamada "TalkBack" que ayuda a estas personas a navegar por las diferentes aplicaciones e interactuar con cada uno de los elementos de las mismas. La aplicación desarrollada permite utilizar esta configuración de manera intuitiva y sencilla. En la pantalla en la que se muestra el prospecto, en vez de leer la sección completa, siendo posible que sea molesto para el usuario, la aplicación lee línea por línea cuando el usuario realiza el gesto de leer el siguiente elemento.

7. Conclusiones

Al finalizar el Trabajo de Fin de Máster se concluye que se han alcanzado los objetivos planteados. Se ha llevado a cabo una investigación sobre el estado del arte tanto en la tarea de reconocimiento de objetos en un dispositivo móvil utilizando el *framework TensorFlow Lite* así como en la generación de respuestas utilizando PLN a través del paradigma *text-to- text* que se aplica en el campo de la medicina en lenguaje en Español. Se han creado dos modelos, uno de visión artificial para identificar las diferentes cajas de medicamentos desde el propio dispositivo móvil y otro de PLN para generar una respuesta a una pregunta de un usuario acerca del medicamento.

Por otra parte, se ha creado un conjunto de datos para clasificar e identificar cajas de medicamentos. Al tener que elaborar nuestro propio *datasets* y en un periodo de tiempo limitado, no ha sido posible incluir una gran cantidad de medicamentos. Por esto, se han elegido los 5 medicamentos más vendidos en España para realizar las pruebas de la primera versión del sistema. Tras realizar este paso se han entrenado correctamente 4 modelos, dos de clasificación y dos de localización. Analizando los resultados obtenidos se elige el mejor modelo obtenido para probar el sistema con el caso de estudio.

En relación con la generación de respuestas, se han creado dos conjuntos de datos, el primero para identificar en que sección del prospecto se encuentra la pregunta del usuario. El segundo permite generar una respuesta a la pregunta teniendo como contexto la sección seleccionada en el paso anterior. Posteriormente, se ha realizado *fine-tuning* a dos modelos obtenidos de HuggingFace, uno basado en T5 y otro en BART, pre-entrenados en conjuntos de datos en Español. Tras realizar los entrenamientos, se decidió cuál es el mejor modelo obtenido teniendo en cuenta sus métricas.

Durante su desarrollo, han aparecido diferentes desafíos. El primero, al igual que en la detección de los medicamentos, no se ha encontrado un conjunto de datos especializado a la generación de respuestas acerca de medicamentos. De la misma manera, no existe un modelo ya entrenado para esta tarea. Se han encontrado diferentes modelos que responden a preguntas de manera generativa o extractiva, pero al realizar diferentes experimentos, los resultados no se consideraron válidos.

Por este motivo se ha decidido crear el conjunto de datos de manera autónoma. En el caso de la elección de la sección, los resultados obtenidos se consideran altamente buenos. Por el contrario, en el caso de la generación de la respuesta, debido a que solo ha sido posible el entrenamiento para tres medicamentos, los resultados para estos medicamentos son correctos, pero el modelo obtenido no generaliza para el resto de medicamentos de manera correcta según la sección.

Otro desafío surgido en esta parte del estudio, es la necesidad de un *hardware* de altas prestaciones. En la fase de reconocimiento del medicamento a través de las imágenes, ha sido posible realizar el entrenamiento utilizando Google Colab. En las fases iniciales del desarrollo del modelo que genera las respuestas fue posible utilizar

esta herramienta, pero al aumentar el conjunto de datos no era posible hacer uso de Google Colab debido al tiempo necesario para realizar el entrenamiento. Por este motivo, se tuvo que obtener un *NVDIA Jetson AGX Orin 64GB Developer Kit* para poder llevar a cabo los entrenamientos necesarios para obtener los resultados buscados.

Además, se ha implementado una aplicación disponible en dispositivos con sistema operativo Android que permite utilizar los modelos de *machine learning* obtenidos. Esta aplicación permite ser utilizada por personas con discapacidad visual.

Para finalizar, tras realizar el estudio se considera que se han solventado de manera satisfactoria los obstáculos surgidos, en esta primera versión, durante el desarrollo del proyecto y a pesar de las limitaciones que posee el sistema, es un primer paso para que se continúe la investigación en este campo.

8. Líneas de trabajo futuras

Tras realizar el estudio se concluye que se han obtenido unos resultados coherentes y cercanos a los previstos al inicio de la investigación. A pesar de esto, se considera que es posible una mejora en diferentes ámbitos.

En relación a la detección de medicamentos, el número de medicamentos capaces de reconocer es limitado por lo que el siguiente paso es aumentar dicho número, sin perder la calidad de los resultados obtenidos. Además, una posibilidad de mejora consiste en que los usuarios de la aplicación sean capaces a enviar imágenes de sus medicamentos, indicando su nombre, donde se almacenarán en un servidor en la nube y el modelo se re-entrene con dichas imágenes. De esta manera con la ayuda de los usuarios, el modelo de clasificación puede aumentar de forma sustancial el número de medicamentos a identificar.

Respecto al apartado de generar las respuestas a las preguntas de los usuarios, la mejora más importante consiste en aumentar el conjunto de datos para incluir a más medicamentos para obtener un modelo que tenga más capacidad de generalizar. Además, otra mejora es aumentar el conjunto de datos de prueba con preguntas realizadas por un número mayor de usuarios para que los resultados obtenidos sean más robustos y tener la capacidad de detectar alguna carencia en el conjunto de datos de entrenamiento respecto algún ámbito de preguntas.

Referencias

[1] INE. Pirámide de la población empadronada en Castilla y León, 2022. https://www.ine.es/covid/piramides.htm. [*Citado en pág. 1.*]

[2] INE. Esperanza de vida al nacimiento según sexo, 2021. https://www.ine.es/jaxiT3/Datos.htm?t=1414#!tabs-grafico. [*Citado en pág. 1.*]

[3] INE. Tasa bruta de natalidad, 2021. https://www.ine.es/jaxiT3/Datos.htm?t=1381#!tabs-grafico. [*Citado en pág. 1.*]

[4] Owsley. C. Vision and aging. *Annual review of vision science*, 2:255–271, 2016. [*Citado en pág. 1.*]

[5] ACCIÓN VISIÓN ESPAÑA. Informe sobre la ceguera en España, noviembre 2019. https://www.esvision.es/wp-content/uploads/2019/11/Informe_Ceguera.pdf. [*Citado en pág. 1.*]

[6] INE. Pirámide de la población empadronada en España. 2022. https://www.ine.es/covid/piramides.htm. [*Citado en pág. 1.*]

[7] El móvil que permite a los ciegos 'leer' los prospectos. *El Mundo*, mayo 2010. https://www.elmundo.es/elmundo/2010/05/24/castillayleon/1274689699.html. [*Citado en pág. 2.*]

[8] Una aplicación que permite leer prospectos de medicamentos. *El Correo*, marzo 2014. https://www.elcorreo.com/salud/vida-sana/20140310/medicamentos-discapacidad-mayores-herramientas-201403101935-rc.html. [*Citado en pág. 2.*]

[9] Rahman M. A. Islam M. M. Akhter A. Uddin M. A. Paul B. K. Hossain, M. U. Automatic driver distraction detection using deep convolutional neural networks. *Intelligent systems with applications*, 14:200075, 5 2022. [*Citado en pág. 8.*]

[10] Varkonyi-Koczy A. R. Kozlovszky M. Benhamida, A. Traffic Signs Recognition in a mobile-based application using TensorFlow and Transfer Learning technics. 6 2020. [*Citado en pág. 8.*]

[11] Kumar V. Lamba, A. A novel image model for vehicle classification in restricted areas using on-device machine learning. *International Journal of Information Technology*, 15(6):3037–3043, Aug 2023. [*Citado en pág. 8.*]

[12] Fatima N. S. Albeez S. A. Haris, K. M. Advanced vehicle detection heads-up display with tensorflow lite. In Subarna Shakya, Valentina Emilia Balas, and Wang Haoxiang, editors, *Proceedings of Third International Conference on Sustainable Expert Systems*, pages 631–647, Singapore, 2023. Springer Nature Singapore. [*Citado en pág. 8.*]

[13] G Mrázová, I Georgiev. Cnn-based classification of car images implemented for android devices. In Lucie Ciencialová, Martin Holena, Robert Jajcay, Tatiana Jajcayová, Frantisek Mráz, Dana Pardubská, and Martin Plátek, editors, *Proceedings of the 22nd Conference Information Technologies - Applications and Theory (ITAT 2022), Zuberec, Slovakia, September 23-27, 2022*, volume 3226 of *CEUR Workshop Proceedings*, pages 93–102. CEUR-WS.org, 2022. [*Citado en pág. 8.*]

[14] Alam K. M. R. Sadi M. S. Rahman, M. A. A smartphone based real-time object recognition system for visually impaired people. In Md. Shahriare Satu, Mohammad Ali Moni, M. Shamim Kaiser, and Mohammad Shamsul Arefin, editors, *Machine Intelligence and Emerging Technologies*, pages 524–538, Cham, 2023. Springer Nature Switzerland. [*Citado en pág. 9.*]

[15] Dev A. Das J. Misra S. K. Baruah, A. Android-based assistance system for visually impaired person using deep learning and augmented reality. In *Trends in Wireless Communication and Information Security*, pages 175–186, Singapore, 2021. Springer Singapore. [*Citado en pág. 9.*]

[16] Patulot M. J. Seguiro L. J. Tuazon A. N. Huyo-A S. L. Abisado M. Sampedro G. A. Eugenio, A. M. Eyeris: Visual image recognition using machine learning for the visually-impaired. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–5, 2023. [*Citado en pág. 9.*]

[17] Khandelwal A. Thumiki, M. Real-time mobile application for classifying solid waste material into recyclable and non-recyclable using image recognition and convolutional neural network. In *IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-6)*, pages 1–6, 2022. [*Citado en pág. 9.*]

[18] Samkeliso Suku Dube and Admire Bhuru. Snake identification system using convolutional neural networks. In *Dube, S. S., Bhuru, A.*, pages 1–5, 2022. [*Citado en pág. 9.*]

[19] Moloo R. K. Gosaye, K. A mobile application for fruit fly identification using deep transfer learning: A case study for mauritius. In *2022 International Conference for Advancement in Technology (ICONAT)*, pages 1–5, 2022. [*Citado en pág. 9.*]

[20] Han Z. Zhao X. Yuan, P. Integrating the edge intelligence technology into image composition: A case study. *Peer-to-Peer Networking and Applications*, 16(4):1641–1651, Aug 2023. [*Citado en pág. 9.*]

[21] Stojcsics D. Benhamida A. Kozlovszky M. Molnar A. Domozi, Z. Real time object detection for aerial search and rescue missions for missing persons. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, pages 000519–000524, 2020. [*Citado en pág. 9.*]

[22] Golcarenarenji G. Wang Q. Alcaraz-Calero J. M Martinez-Alpiste, I. Smartphone-based real-time object recognition architecture for portable and constrained systems. *Journal of Real-Time Image Processing*, 19(1):103–115, Feb 2022. [*Citado en pág. 9.*]

[23] J. Wang. Lightweight and real-time object detection model on edge devices with model quantization. *Journal of Physics: Conference Series*, 1748(3):032055, jan 2021. [*Citado en pág. 9.*]

[24] Lakshmidevi P. B. Josy J. T. Shivaanivarsha, N. A convnet based real-time detection and interpretation of bovine disorders. In *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, pages 1–6, 2022. [*Citado en pág. 9.*]

[25] Luo J. Zhang F. Tian Z. Chen, W. A review of object detection: Datasets, performance evaluation, architecture, applications and current trends. *Multimedia Tools and Applications*, Jan 2024. [*Citado en pág. 10.*]

[26] Divvala S. Girshick R. Farhadi A. Redmon, J. You only look once: Unified, real-time object detection, 2016. [*Citado en pág. 12.*]

[27] Zhang X. Ren S. Sun J. He, K. Deep residual learning for image recognition, 2015. [*Citado en pág. 13.*]

[28] Howard A. Zhu M. Zhmoginov A. Chen L. C. Sandler, M. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019. [*Citado en pág. 14.*]

[29] Jeong M. Cho J. Jeon H. Park J. Shin K.-Song S. Cheong Y. Lee, J. H. Developing a ophthalmic chatbot system. In *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pages 1–7, 2021. [*Citado en pág. 17.*]

[30] Tan Y. Li M. Pan F. Duan H. Huang Z. Deng-H. Yu Z. Yang C. Shen G. Qi P. Yue C. Liu Y. Hong L. Yu H. Fan G. Tang Y. Zhang, Z. Medchatzh: A tuning llm for traditional chinese medicine consultations. *Computers in Biology and Medicine*, 172:108290, 2024. [*Citado en pág. 18.*]

[31] Rizk Y. Awad M. Antoun J. Zini, J. E. Towards a deep learning question-answering specialized chatbot for objective structured clinical examinations. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2019. [*Citado en pág. 18.*]

[32] Rabia I. Aid A. Haddouche, A. Transformer-based question answering model for the biomedical domain. In *2023 5th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–6, 2023. [*Citado en pág. 18.*]

[33] Lee J. Kim D. Jeong M. Kang J Yoon, W. Pre-trained language model for biomedical question answering. In *PKDD/ECML Workshops*, 2019. [*Citado en pág. 18.*]

[34] Azizi S. Tu T. Mahdavi S. S. Wei J. Chung H. W. Scales N. Tanwani A. Cole-Lewis H. Pfohl S. Payne P. Seneviratne M. Gamble P. Kelly C. Babiker A. Schärli N. Chowdhery A. Mansfield P. Demner-Fushman D. . . . Natarajan V. Singhal, K. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, Aug 2023. [*Citado en pág. 18.*]

[35] Mrabet Y. Abacha A. B. Demner-Fushman, D. Consumer health information and question answering: helping consumers find answers to their health-related information needs. *Journal of the American Medical Informatics Association*, 27(2):194–201, 10 2019. [*Citado en pág. 19.*]

[36] Guo Y. Fu L. Chiu M. Chiu H. Lin H. Chang, Y. Interactive healthcare robot using attention-based question-answer retrieval and medical entity extraction models. *IEEE Journal of Biomedical and Health Informatics*, 27(12):6039–6050, 2023. [*Citado en pág. 19.*]

[37] Calderón-Vilca H. D. Cárdenas-Mariño F. C. Ramos, R. C. G. A bert-based question answering architecture for spanish language. *International Journal of Computer Information Systems and Industrial Management Applications*, 14:119–127, 2022. Publisher Copyright: © MIR Labs, www.mirlabs.net/ijcisim/index.html. [*Citado en pág. 19.*]

[38] Uc-Cetina V. Reyes-Magana-J. Navarrete-Parra, O. R. Aligning a medium-size gpt model in english to a small closed domain in spanish using reinforcement learning. *arXiv preprint arXiv:2303.17649*, 2023. [*Citado en pág. 19.*]

[39] Armengol-Estapé-J. Pàmies M. Llop-Palao-J. Silveira-Ocampo J. Carrino C. P. ... Villegas M Gutiérrez-Fandiño, A. Spanish language models. *CoRR*, abs/2107.07253, 2021. [*Citado en pág. 19.*]

[40] Garcia-Lopez E. Garcia-Cabot-A.-Del-Hoyo-Gabaldon J. Moreno-Cediel A. De-Fitero-Dominguez, D. Distractor generation through text-to-text transformer models. *IEEE Access*, 12:25580–25589, 2024. [*Citado en pág. 19.*]

[41] Torres M. I.- Del Pozo-A. Ruiz, E. Question answering models for human–machine interaction in the manufacturing industry. *Computers in Industry*, 151:103988, 2023. [*Citado en pág. 19.*]

[42] Montes-Rosales-Z. G. López-Monroy-A. P.-López-López A. García-Gorrostieta J. M. González-López, S. Short answer detection for open questions: A sequence labeling approach with deep learning models. *Mathematics*, 10(13), 2022. [*Citado en pág. 19.*]

[43] Aji A. F.- Saffari A. Sen, P. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering, 2022. [*Citado en pág. 19.*]

[44] Oğuz B. Rinott-R. Riedel S.- Schwenk H. Lewis, P. Mlqa: Evaluating cross-lingual extractive question answering. [*Citado en pág. 19.*]

[45] Yang Z. Wang-L. Zhang Y.-Lin H. Wang-J. Sun, C. Deep learning with language models improves named entity recognition for pharmaconer. *BMC Bioinformatics*, 22(1):602, Dec 2021. [*Citado en pág. 20.*]

[46] L Chin-Yew. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. [*Citado en pág. 21.*]

[47] Roukos S. Ward-T. Zhu-W. J. Papineni, K. Bleu: a method for automatic evaluation of machine translation. 10 2002. [*Citado en pág. 21.*]

[48] Shazeer N. Parmar-N. Uszkoreit J.-Jones L. Gomez A. N. ... Polosukhin-I. Vaswani, A. Attention is all you need. [*Citado en págs. 22 y 23.*]

[49] Chang M. W.-Lee K. Toutanova K. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. [*Citado en págs. 23 y 24.*]

[50] Shazeer N. Roberts-A. Lee K. Narang S. Matena M. ... Liu P. J. Raffel, C. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. [*Citado en pág. 24.*]

[51] Liu Y. Goyal-N. Ghazvininejad M. Mohamed A. Levy O. ... Zettlemoyer L. Lewis, M. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. [*Citado en pág. 25.*]

[52] Observatorio del medicamento marzo 2023, marzo 2023. https://fefe.com/wp-content/uploads/2023/05/Obs.-marzo23ok.pdf. [*Citado en pág. 26.*]

[53] Hugging face. https://huggingface.co/. [*Citado en pág. 31.*]

[54] Trusca M. M.-Tufiño R. Moens M. F. Araujo, V. Sequence-to-sequence spanish pre-trained language models, 2024. [*Citado en pág. 32.*]