

---

**Generación de descripciones de imágenes  
basadas en la experiencia**  
**Experience-based Image Description Generation**

---



**Trabajo de Fin de Grado  
Curso 2023–2024**

**Autor**

Adrián Pérez Peinador  
Adrián Sanjuan Espejo  
Rubén Gómez Blanco

**Director**

Antonio Alejandro Sánchez Ruiz-Granados  
María Belén Díaz Agudo

**Grado en Ingeniería Informática  
Facultad de Informática  
Universidad Complutense de Madrid**



# Generación de descripciones de imágenes basadas en la experiencia

## Experience-based Image Description Generation

Trabajo de Fin de Grado en Ingeniería Informática

### Autor

Adrián Pérez Peinador  
Adrián Sanjuan Espejo  
Rubén Gómez Blanco

### Director

Antonio Alejandro Sánchez Ruiz-Granados  
María Belén Díaz Agudo

Convocatoria: *Junio 2024*

Grado en Ingeniería Informática  
Facultad de Informática  
Universidad Complutense de Madrid

27 de Mayo de 2024



# Agradecimientos

Agradecer a nuestros tutores Belén Díaz Agudo y Antonio Alejandro Sánchez Ruiz-Granados por su guía y colaboración durante todo el proceso de realización de este trabajo.

También queremos expresar nuestro agradecimiento a todos aquellos que han contribuido al trabajo respondiendo los cuestionarios de evaluación.

Finalmente queremos agradecer a familiares y amigos por el apoyo incondicional que nos han brindado durante nuestra carrera universitaria.



# Resumen

## Generación de descripciones de imágenes basadas en la experiencia

El auge de la inteligencia artificial generativa ha sido notable en los últimos años, impulsado por avances significativos en algoritmos y tecnologías de aprendizaje automático. Estos modelos tienen la capacidad de generar contenido nuevo y realista, incluyendo imágenes, texto y audio, lo que ha generado un gran interés en una amplia gama de aplicaciones. Sin embargo, a pesar de los avances significativos, esta IA generativa aún enfrenta varias limitaciones importantes como la necesidad de grandes cantidades de datos de entrenamiento y recursos computacionales para producir resultados de alta calidad adaptados a lo que el usuario demanda, lo que puede limitar su aplicabilidad en entornos con recursos limitados.

En este trabajo proponemos una alternativa a estos grandes modelos mediante una arquitectura basada en el razonamiento basado en casos (CBR). La idea principal de la arquitectura CBR es generar descripciones basadas en experiencias con imágenes similares almacenadas en una base de casos. Este enfoque no solo evita el uso de modelos masivos, sino que también destaca por la posibilidad de utilizar experiencias concretas en el proceso de generación y aporta un mayor grado de explicabilidad.

Para llevarlo a cabo, hemos creado una base de casos de tamaño reducido con una recopilación de imágenes y preguntas asociadas a ellas. El proceso de generación de una descripción a partir de una imagen nueva se ha dividido en dos partes: la recuperación de imágenes y sus preguntas asociadas más relevantes para la imagen dada, y la utilización de estas preguntas junto con las respuestas dadas por un sistema VQA (Visual Question Answering) para generar una descripción utilizando un modelo de generación de texto pequeño.

Durante el desarrollo del trabajo se han evaluado diferentes maneras de obtener los casos o imágenes más relevantes para obtener el mejor rendimiento posible de nuestra aproximación. El resultado de este trabajo ha sido publicado en ICCBR, la conferencia internacional sobre razonamiento basado en casos. Finalmente, se ha hecho una evaluación de los resultados, demostrando la viabilidad y el potencial del concepto planteado en el marco de generación de descripciones.

## **Palabras clave**

Visual Question Answering (VQA), Case-Based Reasoning (CBR), Razonamiento basado en experiencia, Inteligencia Artificial (IA), Similitud entre imágenes, Deteción de objetos, COCO, Descripción de imágenes, Embeddings, IA Explicable (XAI).

# Abstract

## Experience-based Image Description Generation

The rise of generative artificial intelligence has been very remarkable in recent years, fueled by significant advances in machine learning techniques and algorithms. These models have the ability to generate new and realistic content, including images, text and audio, which has lead to a wide range of new applications. However, despite these advances, this generative AI still faces some limitations including the massive amount of data and the hardware resources needed for training these large models. Such limitations can reduce their usability when dealing with scarce resources.

In this research we propose an alternative to these generative models through a Case-Based Reasoning (CBR) architecture. The main idea of the CBR architecure is to generate captions based on experiences with similar images stored in a case base. This approach not only avoids the use of massive models but also stands out for the possibility of utilizing sepcific experiences in the generation process, contributing also to a greater exaplainability of the model.

To accomplish this goal, we have created a small case base with a compilation of images and associated questions. The description generation process for a new image has been divided into two parts: the retrieval of the most relevant images and their associated questions, and the utilization of those questions together with the answer provided by a VQA (Visual Question Answering) system to generate the description using a text-to-text model.

Throughout the development of the research we have evaluated several ways of retrieving the most relevant images in order to extract the best possible performance of our approach. The results of this research have been published in ICCBR, the International Conference on Case-Based Reasoning. Finally, we carried out an evaluation of the results, proving the viability and potential of our concept.

## **Keywords**

Visual Question Answering (VQA), Case-Based Reasoning (CBR), Experience based reasoning, Artificial Intelligence (AI), Image similarity, Object detection, COCO, Image description, Embeddings, Explainable AI (XAI).

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	4
1.3. Plan de trabajo . . . . .	5
1.4. Organización interna . . . . .	6
1.5. Estructura del documento . . . . .	7
<b>2. Estado de la cuestión</b>	<b>9</b>
2.1. Comparación de imágenes . . . . .	9
2.2. Recuperación de preguntas . . . . .	10
2.3. <i>Visual Question Answering (VQA)</i> . . . . .	10
2.4. Generación de descripciones . . . . .	12
<b>3. Base de casos</b>	<b>13</b>
<b>4. Recuperación de preguntas</b>	<b>19</b>
4.1. Similitud entre imágenes . . . . .	19
4.1.1. Similitud por píxeles . . . . .	20
4.1.2. Similitud por objetos . . . . .	22
4.1.3. Similitud por <i>embeddings</i> . . . . .	27
4.2. Recuperación de preguntas y adaptación . . . . .	32
4.2.1. Generación de <i>embeddings</i> (GTE-small) . . . . .	33
4.2.2. <i>Clustering</i> (DBSCAN) . . . . .	33
4.3. Evaluación de preguntas . . . . .	36
4.3.1. Hipótesis . . . . .	36
4.3.2. Configuración del experimento . . . . .	37
4.3.3. Resultados . . . . .	38
<b>5. Construcción de la descripción</b>	<b>41</b>
5.1. Respuesta a las preguntas . . . . .	41
5.2. Composición de la descripción . . . . .	44
5.2.1. Parámetros para la composición de la descripción . . . . .	44

5.3.	Evaluación de la descripción . . . . .	52
5.3.1.	Medidas objetivas . . . . .	53
5.3.2.	Resultados de la evaluación con medidas objetivas . . . . .	55
5.3.3.	Evaluación con usuarios . . . . .	58
5.3.4.	Resultados de la evaluación con usuarios . . . . .	59
<b>6.</b>	<b>Conclusiones y trabajo futuro</b>	<b>67</b>
6.1.	Cumplimiento de objetivos . . . . .	67
6.1.1.	Artículo ICCBR . . . . .	68
6.2.	Limitaciones y trabajo futuro . . . . .	69
<b>A.</b>	<b>Contribuciones Personales</b>	<b>71</b>
A.1.	Adrián Pérez Peinador . . . . .	71
A.2.	Adrián Sanjuán Espejo . . . . .	74
A.3.	Rubén Gómez Blanco . . . . .	76
<b>Introduction</b>		<b>79</b>
<b>Conclusions and future work</b>		<b>87</b>
<b>Bibliografía</b>		<b>91</b>

# Índice de figuras

1.1.	Flujo general de la generación de la descripción. . . . .	3
1.2.	Proceso de generación de descripciones con CBR . . . . .	5
3.1.	Información asociada a una imagen . . . . .	14
3.2.	Distribución por tipo de pregunta. . . . .	15
3.3.	Distribución de imágenes por número de pregunta. . . . .	16
3.4.	Tamaños de imágenes más repetidos en nuestra base de casos. . . . .	17
4.1.	Flujo de recuperación de preguntas. . . . .	20
4.2.	Imagen de referencia para la similitud . . . . .	20
4.3.	Imágenes comparadas con NRMSE . . . . .	22
4.4.	Clases de objetos definidas en COCO . . . . .	23
4.5.	Esquema de arquitectura de RetinaNet. . . . .	24
4.6.	Esquema de arquitectura de YOLO. . . . .	24
4.7.	Comparación de modelos de detección de objetos . . . . .	25
4.8.	Imágenes comparadas por etiquetas . . . . .	26
4.9.	Imágenes comparadas por <i>bounding boxes</i> . . . . .	27
4.10.	Arquitectura original de Resnet-18 . . . . .	28
4.11.	Arquitectura original de Alexnet . . . . .	28
4.12.	Arquitectura original de Vgg-16 . . . . .	29
4.13.	Arquitectura original de Densenet . . . . .	30
4.14.	Imágenes comparadas con similitud del coseno con <i>embeddings</i> . . . . .	31
4.15.	Ejemplo de una de las imágenes del experimento. . . . .	37
5.1.	Flujo de generación de la descripción desde la obtención de las preguntas. . . . .	42
5.2.	Comparación entre los modelos VLP tradicionales y ViLT . . . . .	42
5.3.	Comparación del rendimiento de <i>Stablelm-zephyr-3b</i> con otros dos LMs de código abierto más grandes en cada tarea evaluada en MT-Bench. . . . .	45
5.4.	Valores del parámetro temperatura y su efecto en las predicciones de un modelo de lenguaje. . . . .	47

5.5.	Imagen de referencia para ejemplificar las descripciones generadas con las distintas configuraciones. . . . .	49
5.6.	Distribución de probabilidades de las métricas Bleu_n en el conjunto de test para cada configuración. . . . .	56
5.7.	Distribución de probabilidades de la métrica ROUGE_L en el conjunto de test para cada configuración. . . . .	57
5.8.	Distribución de probabilidades de la métrica CIDEr en el conjunto de test para cada configuración. . . . .	58
5.9.	Distribución de probabilidades de la métrica <i>cosine_similarity</i> en el conjunto de test para cada configuración. . . . .	58
5.10.	Ejemplo real de la evaluación de descripciones con usuarios. . . . .	60
5.11.	Matriz de correlación entre las variables medidas. . . . .	62
5.12.	Imagen utilizada en la evaluación de las descripciones con usuarios. .	64
5.13.	Resultados de la evaluación con usuarios de las descripciones. . . . .	66
6.1.	Comparación del modelo propuesto con Gemini. . . . .	68

# Índice de tablas

4.1.	Número de veces que se seleccionó cada modelo y tipo de distancia. . . . .	30
4.2.	<i>Benchmark MTEB</i> para distintos modelos . . . . .	34
4.3.	Resultado de la evaluación de las técnicas de similitud entre imágenes . . . . .	38
4.4.	Resultado de la evaluación de las técnicas de similitud entre imágenes por tipo de imagen . . . . .	39
5.1.	Comparación de ViLT con otros modelos en tareas de clasificación . . . . .	43
5.2.	Evaluación del rendimiento de modelos LM de código abierto. . . . .	45
5.3.	Resumen del valor de los parámetros que caracteriza cada una de las configuraciones evaluadas. . . . .	50
5.4.	Valores medios de cada configuración para cada métrica. . . . .	57
5.5.	Resultados generales de la evaluación con usuarios. . . . .	61



# Capítulo 1

## Introducción

### 1.1. Motivación

Con la incipiente llegada de la inteligencia artificial (IA) a la vida de cada vez más personas y con el uso de ésta en muchos ámbitos profesionales como la enseñanza o la medicina, aparecen nuevos desafíos que abordar a la hora de entrenar y explicar modelos que hagan nuestra vida algo más fácil.

Es innegable la curiosidad y asombro de los usuarios por los resultados que hace menos de 2 años empezaban a obtener modelos como ChatGPT<sup>1</sup> o Dall-E<sup>2</sup> en su tarea como IAs generativas. Su rendimiento y accesibilidad al usuario promedio (sin grandes conocimientos de IA ni de informática en general) ha provocado una alta popularidad y un aumento exponencial de su uso. Tanto es así que muchas de estas herramientas se han convertido en una necesidad para millones de usuarios que a diario usan estos modelos de IA generativa. Esto provoca una reacción en cadena en la que cada vez más empresas confían en el futuro de estos modelos. Una investigación de Brainy Insights<sup>3</sup> estima que los ingresos generados por los servicios de IA generativa alcanzarán los 188 mil millones de dólares para 2032, impulsados por una mayor adopción de la IA en todas las industrias y el deseo de las empresas de aprovechar los datos para la toma de decisiones. En este sentido, el estudio de este tipo de sistemas ha provocado el desarrollo de modelos cada vez más ambiciosos que necesitan cada vez más datos de entrenamiento para ajustar aún más la corrección de sus resultados. Así, se continúa desarrollando nuevas versiones que mejoran a las actuales y que son capaces de ejecutar nuevas tareas cada vez más perfeccionadas y que tienen en cuenta más detalles.

Sin embargo, el entrenamiento a tan grande escala de modelos tan masivos de datos está limitado por los recursos tanto hardware como software que deben consumir. Además, la potencia de procesamiento y energía que exige lo hace menos accesible para organizaciones o individuos más pequeños. Esta intensidad de recursos también plantea preocupaciones ambientales, dada la huella de carbono asociada con los enormes centros de datos necesarios para entrenar y ejecutar estos modelos. Por

---

<sup>1</sup><https://chat.openai.com/>

<sup>2</sup><https://openai.com/dall-e-2/>

<sup>3</sup><https://www.thebrainyinsights.com/report/generative-ai-market-13297>

esta razón, también han aumentado los estudios que se centran en como minimizar estos recursos, por ejemplo, reduciendo de alguna manera el tamaño de los datos de entrenamiento [42].

Dentro de la IA generativa, las tareas de *Computer vision* son muy llamativas, y es que la combinación lenguaje-imagen es algo inherente al ser humano y es muy atractivo para las personas que la inteligencia artificial se aproxime cada vez más al rendimiento humano en tareas como la generación de imágenes o texto a través de ciertas instrucciones [15].

Así, se plantean sistemas como el *image captioning* capaces de generar descripciones automáticas en lenguaje natural de imágenes dadas. Este tipo de modelos tiene gran variedad de posibles usos en diferentes campos [58].

1. Accesibilidad para personas con discapacidad visual: Gracias a esta tecnología, las imágenes pueden ser traducidas a descripciones de texto, permitiendo a las personas con discapacidad visual entender y experimentar visualmente el mundo que les rodea.
2. Búsqueda y organización de imágenes: Las descripciones generadas pueden ayudar en la indexación y búsqueda de imágenes en bases de datos, permitiendo una recuperación más eficiente de imágenes relevantes.
3. Asistencia en aplicaciones médicas: En la medicina, puede ayudar a los profesionales de la salud a interpretar imágenes médicas, como radiografías y resonancias magnéticas, al proporcionar descripciones detalladas de las características observadas en las imágenes.
4. Creación de contenido para redes sociales y marketing: Las descripciones de imágenes generadas automáticamente pueden ayudar a los profesionales de marketing y a los creadores de contenido a etiquetar y describir imágenes de manera rápida y eficiente para su uso en redes sociales, blogs y campañas de marketing.

Sin embargo, la mayoría de sistemas actuales de generación de descripciones de imágenes son modelos a grande escala que entran con grandes conjuntos de datos consumiendo muchos recursos. La tarea de reducir estos recursos, como ya hemos adelantado, es crucial para la continuidad del desarrollo de este tipo de sistemas y motiva a explorar otras alternativas de desarrollo.

En este trabajo nos planteamos explorar problemas como el VQA (Visual Question Answering), consistente en ser capaz de responder preguntas sobre imágenes en lenguaje natural [2]. En particular, esta idea abre las puertas a desarrollar aplicaciones innovadoras que no se limitan a la respuesta de preguntas. Por ejemplo, estos sistemas VQA pueden ser utilizados para generar el contexto necesario, gracias a las preguntas y las respuestas que otorgan, para generar descripciones de imágenes. Nuestro enfoque se basa en tener en cuenta experiencias anteriores y usar el paradigma del razonamiento basado en casos (CBR) [6]. Así, basándose en preguntas que otras personas han hecho sobre otras imágenes similares, se podría obtener una descripción coherente de una imagen concreta de forma más eficiente. Es más, si el conjunto de casos del que se dispone cuenta con preguntas enfocadas en aspectos

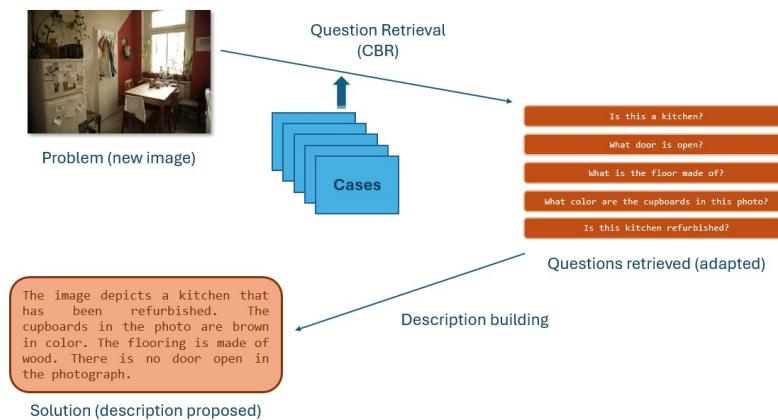


Figura 1.1: Flujo general de la generación de la descripción.

concretos, se puede conseguir que la descripción obtenida se centre también en estos aspectos. Esto puede ser útil a la hora de generar descripciones desde un punto de vista más técnico o específico.

Otro aspecto que consideramos en este trabajo es la explicabilidad. Es cierto que la capacidad de los sistemas para explicar las conclusiones a las que llegan toma cada vez más importancia, pues se tratan de alejar de la imagen de *caja negra*. No obstante, los sistemas basados en aprendizaje profundo mediante redes neuronales son limitados en este aspecto y aportar explicabilidad en ellos supone un reto.

Con este Trabajo de Fin de Grado queremos proponer un sistema de generación de descripciones para imágenes explicable y basado en la experiencia con razonamiento basado en casos (CBR). El objetivo final es conseguir un sistema que dado una imagen, sea capaz de generar una descripción suya basándose en las respuestas (dadas por un sistema VQA “ligero”) a las preguntas que se han recuperado de las imágenes más similares a esta (CBR) (ver figura 1.1). Para ello, trataremos de evitar los modelos de generación de descripciones que entran con un número de datos masivo para generar un contexto a la descripción, ya que nuestro contexto estará compuesto por la experiencia que aporta nuestro conjunto de casos. Además, esto debe conseguirse siguiendo determinados pasos intermedios que permitan aportar “explicabilidad” al sistema general, evitando que sea un modelo de *caja negra* y haciéndolo un poco más accesible a las personas, tengan o no conocimiento sobre estos modelos y sobre la Inteligencia Artificial.

## 1.2. Objetivos

En resumen, los principales objetivos que queremos cumplir durante la realización del trabajo son:

- (O<sub>1</sub>) **Estudiar el estado del arte.** Exploraremos el estado del arte de las principales subtareas de nuestro sistema; comparación de imágenes, recuperación de preguntas, sistemas VQA y generación de descripciones, con el fin de entender sus fortalezas y dificultades con las que nos encontraremos durante el desarrollo del trabajo.
- (O<sub>2</sub>) **Elegir una base de casos adecuada.** Debemos encontrar una base de casos etiquetada correctamente para realizar tareas VQA. Además, buscamos un conjunto de datos lo más representativo posible a la realidad tanto en las imágenes como en las preguntas asociadas a estas.
- (O<sub>3</sub>) **Desarrollo de un sistema de razonamiento basado en casos de obtención de preguntas.** Dado que nuestro objetivo es utilizar preguntas hechas a imágenes similares para componer una descripción, necesitamos desarrollar un sistema de razonamiento basado en casos (CBR) [6] capaz de recuperar, reusar y adaptar las preguntas formuladas a estas imágenes.
- (O<sub>4</sub>) **Desarrollo de distintas medidas de similitud entre imágenes.** Para la recuperación de preguntas que nos sean útiles para componer la descripción de la imagen objetivo necesitamos seleccionar las imágenes más parecidas a ella. Así, nos proponemos investigar varias medidas que permitan el cálculo de similitud entre imágenes basándose en distintos parámetros y características de la imagen.
- (O<sub>5</sub>) **Evaluación de las medidas de similitud.** El objetivo es evaluar con usuarios las medidas de similitud en términos de recuperación de preguntas (de las imágenes más similares) para una imagen dada. Tras un análisis de la evaluación se elegirá la más apropiada para nuestro propósito.
- (O<sub>6</sub>) **Construcción de las descripciones de imágenes.** Una vez obtenidas las preguntas más relevantes para nuestra imagen queremos ser capaces de responderlas y generar una descripción de la imagen basada en esas respuestas.
- (O<sub>7</sub>) **Evaluación de las descripciones.** Nos proponemos poner a prueba nuestro modelo evaluando las descripciones obtenidas con él, de forma que obtengamos una indicación de sus prestaciones.

En definitiva, nuestro objetivo es construir un modelo de generación de descripciones de imágenes basado en la experiencia almacenada en casos. Su funcionamiento resumido en la figura 1.2, bebe de la idea de que imágenes similares recibirán preguntas similares. El modelo recibe como entrada una imagen, a la que llamaremos imagen objetivo, y recupera las  $k$  más parecidas a ella. De entre las preguntas que se han formulado a estas imágenes, selecciona las más relevantes (ver sección 4.2) y con las respuestas de estas preguntas aplicadas a la imagen objetivo compone una descripción redactada de la imagen.

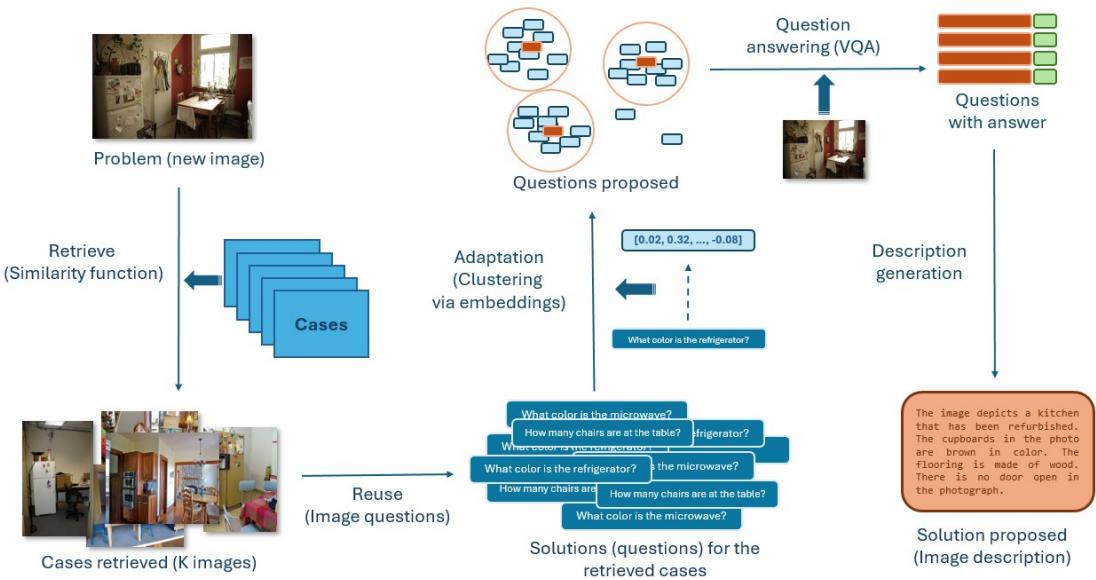


Figura 1.2: Flujo en detalle del proceso de generación de la descripción basado en el ciclo de razonamiento basado en casos.

### 1.3. Plan de trabajo

Con el fin de alcanzar los objetivos mencionados en la sección A.3, describimos el plan de trabajo que seguiremos durante la realización del proyecto.

En primer lugar, se deberá realizar una revisión bibliográfica exhaustiva sobre los conceptos más importantes acerca de la similitud entre imágenes, VQA y generación de descripciones mientras se desarrollan las ideas a tratar en el trabajo. Se deberá ir fijando objetivos a cumplir a medida que nuestro conocimiento sobre el tema vaya siendo mayor y tengamos una idea global del trabajo más formada.

Una vez asentadas las bases teóricas, se llevará a cabo una búsqueda más concreta sobre el estado del arte, las últimas novedades acerca de este campo, los papers más innovadores y los modelos más usados y eficientes hasta la fecha, tanto de VQA como de las fases que lo componen y de modelos generativos de texto para las descripciones.

Así, en primera instancia y después de escoger un dataset que tenga las características necesarias para realizar nuestra tarea, nos centraremos en los distintos métodos para medir la similitud entre imágenes. Escogeremos tres métodos distintos, encargándose cada miembro del grupo del estudio más a fondo de uno en concreto. Nos centraremos en el estudio de la detección de objetos, píxeles y *embeddings* en imágenes, donde basándonos en la información sacada de la investigación previa, testearemos varios modelos y usaremos distintas métricas para evaluar la efectividad de estos. También se deberá realizar una evaluación para tratar de elegir el mejor modelo para encontrar las imágenes más similares a una dada.

En segundo lugar, basándonos en el estudio previo acerca del estado del arte de los modelos VQA, elegiremos un modelo que sea capaz de responder las preguntas

de las imágenes más similares a la dada. Este es un paso importante ya que debemos encontrar un modelo que se adecúe al tipo de respuestas que queremos conseguir, así como que se adapte al tipo de datos de entrada y salida que necesitamos.

Por último, profundizaremos en el mundo de la IA generativa de texto buscando un modelo que tome como entrada las preguntas y sus respuestas, y sea capaz de generar una descripción basada exclusivamente en ello. Por esto, compararemos distintos modelos según su relación tamaño/rendimiento para conseguir uno que sin ser entrenado con un tamaño de datos masivo sea capaz de generar una buena descripción sin necesidad de más contexto. También, se realizará una evaluación acerca de los resultados finales de nuestro trabajo, evaluando la calidad de las descripciones generadas para poder sacar conclusiones objetivas acerca de nuestro trabajo global. Con esta evaluación valoraremos la viabilidad del concepto de este trabajo y reflexionaremos sobre causas de errores y posibles mejoras de cara a plantear un posible trabajo futuro.

## 1.4. Organización interna

El proyecto se desarrollará en un entorno interactivo *JupyterHub*, compartido tanto entre los componentes del grupo de trabajo como a los tutores, con el fin de poder llevar un seguimiento del trabajo realizado durante el curso. En este entorno se encuentra todo el código desarrollado desde el principio del proyecto, incluyendo también código que ha sido descartado en el modelo final, pero que ha formado parte de pruebas y del proceso de aprendizaje.

Además, se subirá el código final en un repositorio de Github<sup>4</sup>. El repositorio contiene instrucciones sobre cómo está estructurado y cómo se puede utilizar para reproducir nuestros resultados.

En cuanto a la gestión de la documentación que usaremos, tanto propios como ajenos, se usarán dos formas de coordinación. Por un lado, se hará uso de una carpeta compartida en Google Drive, donde se organizarán documentos interesantes para leer sobre el tema que nos ocupa, notas escritas por el equipo de trabajo y otros archivos relevantes (i.e. los cuestionarios de evaluación con usuarios). Por otro lado, para la redacción conjunta y coordinada de la memoria y otros documentos se hará uso de la herramienta Overleaf de edición de código L<sup>A</sup>T<sub>E</sub>Xen tiempo real.

En lo referente a organización y seguimiento de las tareas y desarrollo del trabajo, se decidió tener reuniones telemáticas cada dos semanas, siendo esta fecha orientativa. Así, si en cierto momento se considera oportuno por razones de festivos, mayor carga de trabajo o necesidad de tratar algún tema, las reuniones se pueden ajustar en día, hora, duración y frecuencia. En las reuniones se comentará el trabajo hecho desde el último encuentro, se propondrán las líneas de trabajo futuro y se aclararán las dudas que hayan podido surgir. Asimismo, si algún tema es de especial urgencia, se prevé el uso del correo electrónico institucional para la comunicación entre alumnos y tutores.

---

<sup>4</sup><https://github.com/TFG-UCM-VQA/VQA-TFG>

## 1.5. Estructura del documento

El documento ha sido organizado de manera que pueda ser seguido y comprendido por personas con una cierta base de conocimiento (sin ser expertos) acerca de las principales técnicas de aprendizaje profundo.

En el capítulo 2 se estudian las aproximaciones más actuales en las tareas de VQA, procesamiento del lenguaje natural y generación de descripciones de imágenes, describiendo sus principales características.

En el capítulo 3 se presenta la base de casos final en la que nos basaremos para el trabajo, explicando detalladamente sus características.

En el capítulo 4 se hace un recorrido por todo el proceso de recuperación de preguntas, desde el estudio de las similitudes entre imágenes, pasando por la selección de las preguntas recuperadas, hasta la evaluación de dichas preguntas.

En el capítulo 5 se explora el proceso de generación de la descripción de la imagen a partir de las preguntas recuperadas. Primero estudiando el modelo VQA encargado de responderlas, después estudiando los sistemas de generación de texto y sus parámetros para encontrar la mejor configuración, y por último evaluando la descripción generada.

En el capítulo 6 se exponen nuestras conclusiones y se evalúa el cumplimiento de nuestros objetivos. También se indica el posible trabajo futuro, prestando especial atención a los obstáculos que deben superarse antes de que un sistema como este se integre en la práctica.



# Capítulo 2

## Estado de la cuestión

Desde la aparición del aprendizaje profundo ha habido un avance significativo en la generación de descripciones de imágenes [3, 12, 20]. Muchos artículos de investigación siguen la estrategia de preentrenar inicialmente grandes modelos de *Computer Vision* y luego adaptarlos a tareas específicas como la generación de descripciones [27, 30, 67]. Sin embargo, nuestro concepto se basa en evitar este tipo de modelos pre-entrenados con grandes conjuntos de datos, por lo que nos centraremos en investigar el estado del arte de las subtareas que llevaremos a cabo en nuestro sistema: comparación de imágenes y recuperación de preguntas, VQA y generación de descripciones a partir de un *prompt* de entrada.

### 2.1. Comparación de imágenes

Tradicionalmente, los sistemas de comparación de imágenes han buscado identificar las características claves, a bajo nivel y sin abstracciones de las imágenes a comparar. Un ejemplo de esto son los modelos SIFT [37] y HOG [8], que requieren un conocimiento significativo para preprocessar y aprender características a bajo nivel de las imágenes. Otro ejemplo clásico es la comparación de imágenes mediante la comparación de sus píxeles utilizando medidas de similitud como *Normalized Root Mean Squared Error (NRMSE)*, *Normalized Mutual Information (NMI)* o *Peak Signal to Noise Ratio (PSNR)*. Además, existen otros métodos que tienen en cuenta información de ligeramente más alto nivel. Un ejemplo de esto son los modelos de detección de objetos en la imagen, entre los que destacan modelos como RetinaNet [35] y YOLO [51] que tienen buenos resultados reconociendo objetos de las clases definidas en COCO [36]. Sin embargo, actualmente se ha prestado mucha atención al uso de redes convolucionales (CNN) [43]. El enfoque actual busca extraer características de las imágenes con una red neuronal y compararlas para evaluar la similitud de dos imágenes. Al entrenar una red neuronal para una tarea específica, las capas completamente conectadas transforman los píxeles de la imagen en *embeddings* de características, que luego se pasan al clasificador (según la arquitectura) para su inferencia. Entre los distintos modelos de generación de *embeddings* de imágenes destaca Img2Vec de la librería PyTorch. Algunas de las redes utilizadas por estos modelos de generación de *embeddings* son: resnet-18 [48], alexnet [24] , vgg-16

[57], densenet [19]. Aunque a menudo son sistemas de “caja negra”, estos modelos generan *embeddings* de características que capturan suficiente información sobre las imágenes para compararlas de manera efectiva. Una vez obtenidos estos *embeddings* o vectores de características de las imágenes, existen distintas medidas de similitud como la distancia Euclídea, Manhattan, Minkowski o similitud del coseno.

## 2.2. Recuperación de preguntas

En cuanto a la recuperación de preguntas, nuestro enfoque se basa en utilizar *embeddings* para generar clusters o agrupaciones según el tipo de pregunta. Por tanto, es necesario para nuestra aproximación un modelo de generación de *embeddings*. Existen múltiples modelos de generación de *embeddings* a partir de texto, como Word2Vec [39], que utiliza un modelo de red neuronal para aprender asociaciones de palabras a partir de un gran corpus de texto. Sin embargo, la mayoría están basados en la arquitectura *Transformer* introducida en el artículo *Attention is all you need* [62], donde destaca el modelo *GTE-small*, perteneciente a la familia de modelos GTE introducidos en [32], los cuales se basan en el framework de BERT<sup>1</sup>[11] y se han entrenado para un propósito general utilizando *multi-stage contrastive learning*.

En cuanto a la selección de las preguntas mediante *clusters*, existen diversos algoritmos de *clustering* dependiendo la técnica en la que se basan. Algunos de los más conocidos son el *clustering* jerárquico, *clustering* basado en densidad o *k-means*. También destaca DBSCAN (*clustering* basado en densidad) [16] que es menos sensible a la presencia de ruido que otros algoritmos como *K-means* y no requiere de especificar el número de *clusters* de antemano.

## 2.3. *Visual Question Answering (VQA)*

La investigación actual sobre sistemas VQA se divide principalmente en dos aproximaciones: métodos de conocimiento ligero [70, 28] y métodos intensivos en conocimiento [66, 69]. Mientras los primeros se centran en el procesamiento de la información disponible directamente en las imágenes y las preguntas sin depender de grandes bases de conocimiento externas, los segundos se basan en el uso extensivo de fuentes de conocimiento externas, lo que les permite abordar preguntas más difíciles y ambiguas que requieren un entendimiento más profundo del contexto y el sentido común.

Sin embargo, ambos métodos siguen los siguientes pasos: 1) detección de objetos en una imagen con alta precisión y detalles, como el área aproximada del objeto o su localización en la imagen; 2) comprensión del lenguaje natural de la pregunta; y 3) compilación de la respuesta utilizando la información de los pasos 1 y 2, y fuentes de conocimiento externas (principalmente en sistemas intensivos en conocimiento) [5]. Es claro que el tipo de imagen y de pregunta determina la facilidad con la que

---

<sup>1</sup>Uno de los primeros modelos de procesamiento del lenguaje natural fue BERT (Bidirectional Encoder Representation from Transformers), y se basa en el artículo ya mencionado de *Attention is all you need*. BERT sentó las bases para muchos de modelos de procesamiento del lenguaje natural posteriores.

un sistema VQA es capaz de responder estas preguntas. Mientras que en ciertas imágenes hay objetos muy diferenciados y preguntas como “*¿Cuántos objetos hay en la imagen?*” son fáciles de responder correctamente, en otros casos es necesario inferir conocimientos de otras fuentes para encontrar las coincidencias entre nuestra pregunta y los objetos detectados en nuestra imagen [38]. Esto es especialmente relevante para preguntas abiertas o ambiguas, preguntas con múltiples respuestas o preguntas que requieren información adicional no disponible en la imagen [69]. Por ejemplo, en una imagen donde se ve un conjunto de personas conversando, preguntas como “*¿Cuántas personas están felices?*” pueden ser muy difíciles de responder correctamente.

Existen modelos actuales basados en datos que abarcan los tres pasos de la tarea de VQA. Los modelos de visión-lenguaje (VL) realizan VQA utilizando modelos de secuencia a secuencia que se entrena en conjuntos de datos extremadamente grandes [61]. Estos modelos utilizan aprendizaje no supervisado basado en *constrastive loss* [22] para aprender relaciones entre imagen y preguntas asociadas. Los modelos VL adquieren una dupla de entrada  $\langle \text{imagen}, \text{pregunta} \rangle$  y generan la respuesta mediante generación de secuencias auto-regresivas, es decir, el siguiente elemento de la secuencia se predice al medir las entradas anteriores de la secuencia.

Algunos modelos recientes incluyen el modelo ALIGN [21], que se entrena utilizando *constrastive loss*, el modelo ALBEF [28], que se entrena utilizando inter-modal alignment; el modelo VL [70], que se entrena utilizando *Triple Constrastive Loss*; y VinVL [72], que integra una detección de objetos mejorada con el modelo VL.

En general el sistema VQA es solo una parte de los modelos VL, que son entrenados en una segunda fase usando datasets formados por tuplas de la forma  $\langle \text{imagen}, \text{pregunta}, \text{respuesta} \rangle$ , donde la imagen y la pregunta son la entrada y la salida etiquetada es la respuesta en lenguaje natural. Teniendo esto en cuenta, se puede observar el VQA como una tarea de clasificación que empleará una capa softmax al final del codificador de fusión para predecir la respuesta a partir de un conjunto dado de respuestas [70]. Por otro lado, VQA puede ser modelado como una tarea generativa que empleará un generador de texto como GPT [47] para generar la respuesta.

Respecto al uso de conocimiento externo, esto puede resultar en una mejora notable en las tareas de VQA para preguntas complejas o de sentido común. Por ejemplo, en una pregunta como “*¿Cuántos tipos de frutas hay en la imagen?*”, el algoritmo debería reconocer no solo los alimentos que tenemos en la imagen, sino también identificar y entender cuáles alimentos son frutas y cuáles no.

Existen diversos sistemas que usan estos métodos intensivos de conocimiento [69, 38]. Uno de ellos es el modelo *Grounded Visual Question Answering mode (GV-QA)* [1] que utiliza diferentes algoritmos dependiendo del tipo de pregunta que nos encontramos (dependiendo si son de sí/no o no). También dividen el proceso de comportamiento del modelo para obtener la respuesta en diferentes pasos (obtener partes importantes en la imagen, recuperar conceptos de la pregunta, clasificar el tipo de preguntas o predecir la respuesta). En este sentido, se diferencian claramente de los métodos basados en datos, que tratan el VQA como una tarea “*end-to-end*” y no como un pipeline de subtareas como en este caso. Otro es VLC-BERT [50], un

modelo de VQA que utiliza COMET (“*Commonsense Transformer*”) [7], un modelo de generación de razonamiento de sentido común que dado un sujeto y una relación, predice un posible objeto. Un ejemplo de los autores es: si el sujeto es “tomar una siesta” y la relación es “causa”, un posible objeto es “tener energía”. Se entrena y se prueba en los grafos de conocimiento ATOMIC [54] y ConceptNet [59], ambos consistentes en conocimiento que socialmente es de sentido común. El razonamiento de sentido común extraído del COMET se utiliza en VLC-BERT para mejorar la generación de respuestas, lo que lo hace un sistema intensivo en conocimiento.

## 2.4. Generación de descripciones

Respecto a modelos de generación de texto más enfocados a la generación de una descripción, se buscan modelos capaces de interpretar mediante un *prompt* de entrada lo que se quiere describir. Esto es, que sea capaz de basarse en las preguntas y sus respuestas exclusivamente para hacer la descripción de la imagen. Para esta tarea, los modelos LLM (Large Language Models) como GPT [47] o Bert [10] que son pre-entrenados con grandes cantidades de texto son una muy buena solución. Sin embargo, en el contexto de este trabajo, nos centramos en modelos más “pequeños”. En el estado del arte, destacan los modelos desarrollados por Stability.ai<sup>2</sup>, de código abierto, y que tratan de minimizar el conjunto de datos de entrenamiento y parámetros necesitados por los modelos manteniendo buenos resultados. Entre sus modelos, destaca Stable LM Zephyr 3B<sup>3</sup>, que representa la última iteración en la serie de LLMs “ligeros” y ajustado preferentemente para seguir instrucciones y tareas tipo preguntas y respuestas. Como su modelo más “pequeño”, destaca Stable LM 2 Zephyr 1.6B<sup>4</sup>, que tiene una ventaja interesante en comparación a otros modelos LM como Microsoft Phi-1.5 [31] o TinyLlama 1.1B [73], y es que está entrenado en varios idiomas como en inglés, español, francés, italiano y alemán.

Una vez elegido el modelo de generación de descripciones, existen diversos métodos para realizar una evaluación objetiva de la corrección de dichas descripciones, por ejemplo BLEU [44], ROUGE [33], Meteor [4] o CIDEr [63]. Salvo CIDEr, que fue concebida específicamente para evaluar descripciones de imágenes, las medidas anteriores se diseñaron para la evaluación de traducciones o resúmenes generados por ordenador. Todas ellas funcionan calculando la similitud entre la salida del modelo y una o varias respuestas de referencia dadas por humanos, lo que ha sido bastante criticado y discutido [25, 14].

La literatura comparando todas estas medidas entre sí en términos de correlación con la evaluación humana es muy amplia y arroja resultados dispares en función de los datasets o el número de respuestas de referencia tomadas. En términos generales se puede decir que de un tiempo a esta parte parece que se tiende a usar datasets más grandes (MS COCO) y medidas con buena correlación con la evaluación humana en ellos (Meteor, CIDEr). No obstante, el uso de BLEU es aún extendido y las evaluaciones humanas siguen siendo la alternativa elegida en muchos casos.

---

<sup>2</sup><https://stability.ai/stable-lm>

<sup>3</sup><https://huggingface.co/stabilityai/stablelm-zephyr-3b>

<sup>4</sup>[https://huggingface.co/stabilityai/stablelm-2-zephyr-1\\_6b](https://huggingface.co/stabilityai/stablelm-2-zephyr-1_6b)

# Capítulo 3

## Base de casos

Es bien sabido que un modelo de aprendizaje automático supervisado necesita un conjunto de datos etiquetados con ejemplos de predicciones para poder ser entrenado. De esta manera, para nuestro propósito, necesitamos imágenes etiquetadas con preguntas y respuestas a las mismas para poder entrenar nuestro modelo. Existen varios tipos de datasets preparados para entrenar esta clase de modelos. Por un lado, existen conjuntos de imágenes con descripciones, y por otro lado también hay ejemplos de conjuntos de preguntas y respuestas asociados a imágenes. Evidentemente, necesitamos un dataset con una estructura como estos últimos ya que son respuestas a preguntas en lo que nos basaremos para describir imágenes. Estos tipo de conjuntos de datos se pueden estructurar de distintas maneras. Existen datasets en los que las anotaciones consisten en una pregunta con distintas respuestas, que serían la entrada del modelo, y la respuesta correcta, que sería la salida. En otros datasets tenemos exclusivamente preguntas como entrada y como salida un número variable de respuestas viables a esa pregunta.

En nuestro caso hemos optado por el dataset de COCO [36] ya que es uno de los más accesibles y utilizados. El dataset consta de 123.287 imágenes de entrenamiento que representan una gran diversidad de escenarios y situaciones como paisajes, interior de casas y habitaciones, personas realizando actividades deportivas, etc. De hecho, el conjunto de datos de COCO se recopiló para encontrar imágenes que contienen múltiples ítems y además den información contextual [2]. Además, cuanto más diversa sea nuestra colección de imágenes, más completo e interesante será el conjunto resultante de preguntas y sus respuestas para llevar a cabo nuestra tarea, por lo que el dataset de COCO será una buena opción para ser utilizado.

Estas imágenes vienen acompañadas de sus respectivas anotaciones, que proporcionan información tanto de cada imagen como de los ítems que están reconocidos en ella. Hay un total de 80 tipos de ítems o entidades etiquetados (persona, paraguas, perro, coche, cama, reloj...) [36]. Para cada imagen, el dataset de anotaciones nos proporciona la siguiente información:

- Segmentación de ítems categorizados.
- Área de la imagen.
- *Bounding boxes* de ítems categorizados.

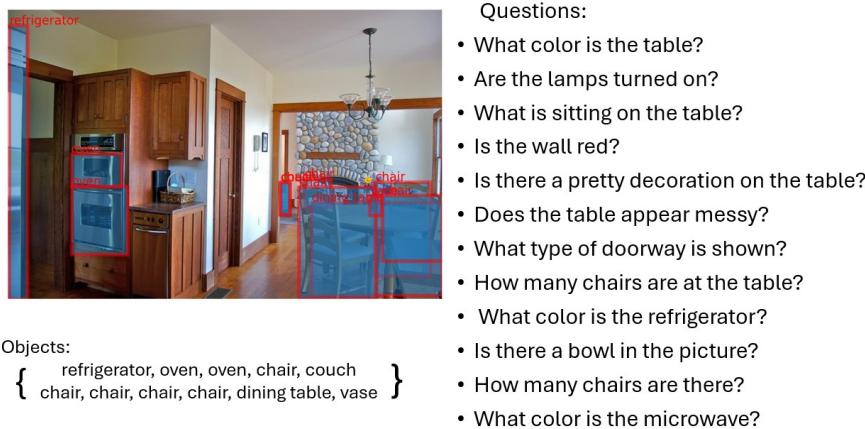


Figura 3.1: Información asociada a una imagen

- Atributo ‘iscrowd’
- Id de la imagen.
- Id de la categoría de la imagen.

El formato de segmentación depende de si la instancia representa un solo objeto (‘iscrowd’=0 en cuyo caso se utilizan polígonos) o una colección de objetos (‘iscrowd’=1 en cuyo caso se utiliza RLE<sup>1</sup>). Las *bounding boxes* son cuadros delimitadores que encierran cada objeto (las coordenadas del cuadro se miden desde la esquina superior izquierda de la imagen y están indexadas a 0). Para este trabajo se han utilizado las *bounding boxes* en la primera aproximación de detección de objetos (ver sección 4.1.2).

Otra de las razones por las que se ha elegido el dataset de COCO es por la existencia del dataset VQA 2.0<sup>2</sup> [17]. Se trata de un dataset estándar en la evaluación de sistemas VQA y es el que se va a utilizar en nuestro trabajo junto con las imágenes de COCO asociadas, para el entrenamiento del modelo. Este dataset esta formado a su vez por dos datasets: el primero es un conjunto de 614.163 preguntas asociadas a las imágenes; el segundo lo forman las anotaciones de estas preguntas, que contienen las posibles respuestas asociadas a ellas (hay un total de 7.984.119 respuestas). El dataset de preguntas contiene la siguiente información de cada pregunta:

- Id de la imagen asociada.
- Pregunta en lenguaje natural.
- Id de la pregunta

<sup>1</sup>RLE (Run Length Encoding) se basa en reemplazar cada símbolo repetido por un valor único y la cantidad de veces que este se repite.

<sup>2</sup><https://visualqa.org/>

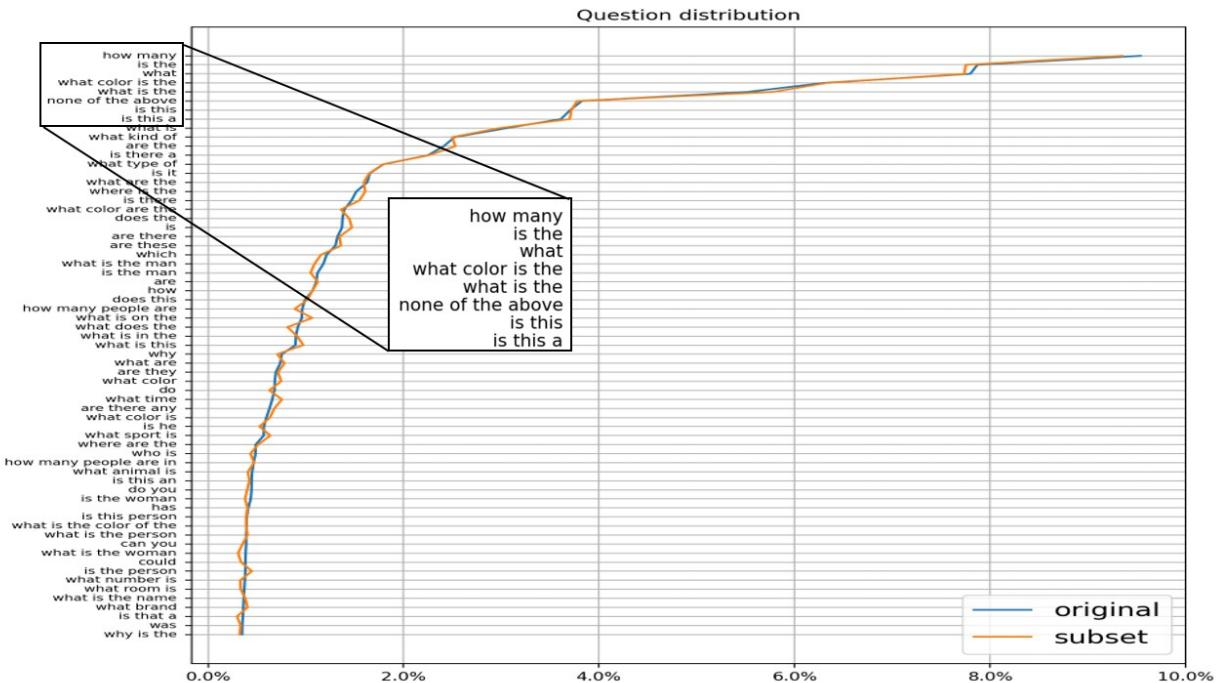


Figura 3.2: Distribución por tipo de pregunta.

El Id de la pregunta es único y tiene como prefijo el Id de la imagen asociada a la pregunta facilitando así el reconocimiento de un par pregunta-imagen asociadas. El dataset de anotaciones contiene la siguiente información de cada pregunta:

- Tipo de pregunta.
- Respuesta más común entre las existentes.
- Lista de respuestas posibles. Cada respuesta consta de:
  - Respuesta en lenguaje natural.
  - Verosimilitud de la respuesta.
  - Id de la respuesta.
- Id de la imagen asociada.
- Tipo de respuesta.
- Id de la pregunta asociada.

El tipo de pregunta viene determinado por las primeras palabras de la misma pregunta. Hay 65 tipos de pregunta [56], siendo las que más se repiten las del tipo *how many*, *is the*, *what*, *what color is the*, y *what is the*, como se puede observar en la figura 3.2. Cada imagen tiene al menos 3 preguntas asociadas, y cada pregunta tiene una lista de 10 respuestas [2].

Para cada respuesta se tiene un valor booleano de confianza de cada respuesta, siendo 1 si es una respuesta “confiable” y 0 en caso contrario. Este atributo es un valor

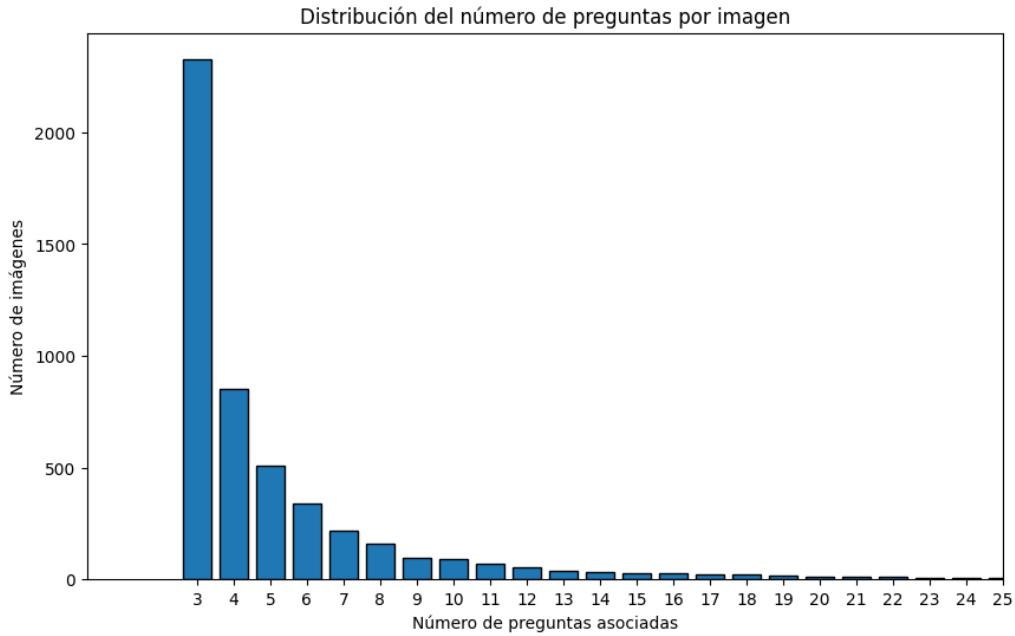


Figura 3.3: Distribución de imágenes por número de pregunta.

subjetivo de los usuarios que respondieron las preguntas del dataset. Además, cada pregunta tiene asociado un tipo de respuesta que puede ser “sí/no”, un número y otra categoría de “otro”. Este atributo puede ser útil para filtrar preguntas de distintos tipos y acotar el problema a preguntas con un tipo de respuestas concretas.

En este trabajo para nuestra base de casos hemos seleccionado de forma aleatoria un subconjunto del dataset que consta de 5000 imágenes, de tal forma que sus preguntas siguen una distribución similar a las del conjunto de datos original en términos de tipos de preguntas, como se muestra en la figura 3.2. La base de casos seleccionada tiene un total de 26987 preguntas asociadas (mínimo tres por imagen) y un total de 269870 respuestas (diez para cada pregunta). Como se puede observar en la figura 3.3, prácticamente la mitad de las imágenes de nuestra base de casos tiene asociada 3 preguntas (con sus respectivas 10 respuestas posibles).

Considerando la imagen de la cual queremos obtener una descripción como nuestra consulta, los casos están compuestos por imágenes y las soluciones a cada caso son, por tanto, las preguntas asociadas a dichas imágenes.

En la base de casos seleccionada, hay una gran variedad de tamaños de imágenes. En la figura 3.4 podemos observar que el tamaño que más se repite es (640,480), siendo la primera coordenada anchura y la segunda de altura. Este será el tamaño que usaremos de referencia si fuera necesario a la hora de comparar imágenes.

En este capítulo hemos descrito detalladamente la base de casos que usaremos durante nuestro trabajo. A continuación, veremos la fase de recuperación de preguntas explicando el proceso de extracción de imágenes similares y obtención de las preguntas más relevantes.

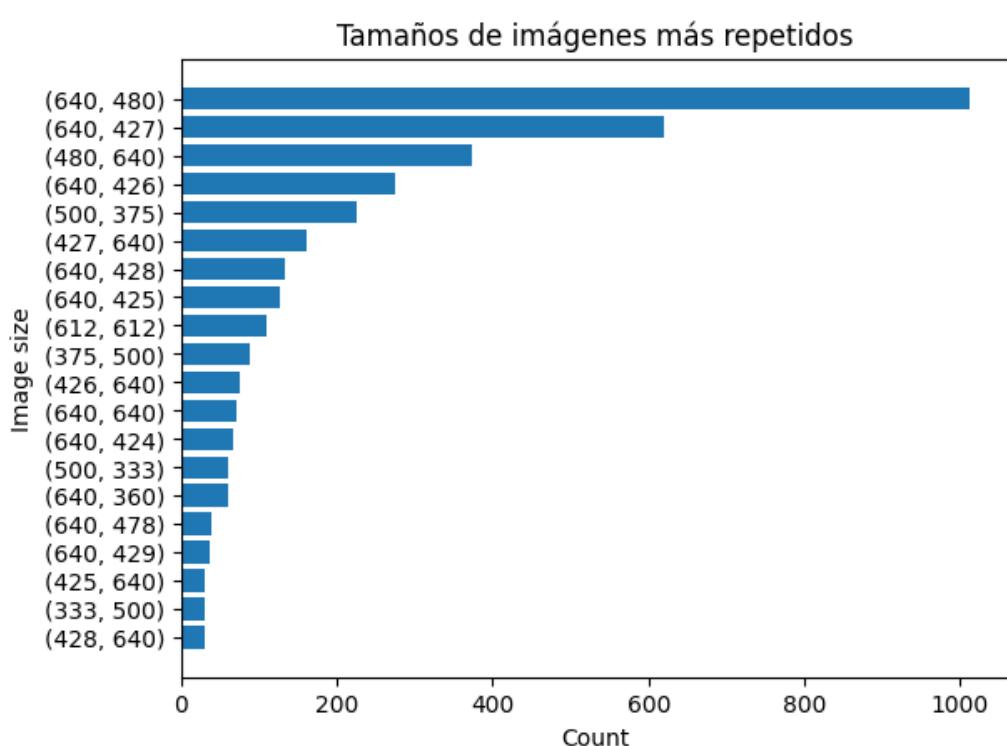


Figura 3.4: Tamaños de imágenes más repetidos en nuestra base de casos.



# Capítulo 4

## Recuperación de preguntas

La generación de texto a partir de imágenes es un problema complejo que proponemos abordar desde la perspectiva de un modelo CBR que sea capaz de generar descripciones de imágenes sin la necesidad de un entrenamiento masivo. La propuesta, como ya hemos adelantado, consiste en aprovechar el conocimiento basado en la experiencia de preguntas que se hayan hecho a imágenes parecidas a la que queremos describir, con el objetivo de generar una descripción a partir de las respuestas que de a estas preguntas un sistema VQA sencillo. En este capítulo planteamos un flujo para la recuperación de las preguntas.

En la figura 4.1 se muestra el modelo de recuperación de preguntas que proponemos y como se relaciona con las partes clave de un sistema CBR. Cuando un usuario quiere describir una imagen, buscamos los  $k$  casos más similares en nuestra base de casos, con el objetivo de reutilizar las preguntas asociadas a estos. Una vez recuperadas las preguntas, la fase final de adaptación debe tener en cuenta posibles solapamientos de preguntas. Se elimina este efecto agrupando las preguntas por tipo y eligiendo un representante de los grupos más numerosos.

Al implementar nuestro sistema hemos investigado distintos tipos de similitud entre imágenes, que detallaremos en la sección 4.1. Asimismo, en la sección 4.2 se describe la evolución de nuestro proceso de adaptación hasta lograr una estrategia eficaz para lograr la recuperación de preguntas relevantes evitando la duplicación de soluciones.

### 4.1. Similitud entre imágenes

En primera instancia es necesario comparar nuestra imagen objetivo<sup>1</sup> con el resto de imágenes de nuestra base de casos. Es importante señalar que nuestro propósito final no es recuperar imágenes similares, sino recuperar imágenes con preguntas asociadas que sean aplicables a nuestra imagen objetivo, por lo tanto estamos trabajando sobre la hipótesis de que imágenes similares tienen preguntas similares asociadas. Con esta idea en mente, hemos estudiado diferentes medidas de

<sup>1</sup>Llamaremos imagen objetivo o consulta al input de nuestro sistema, es decir, la imagen de la cual queremos generar una descripción.

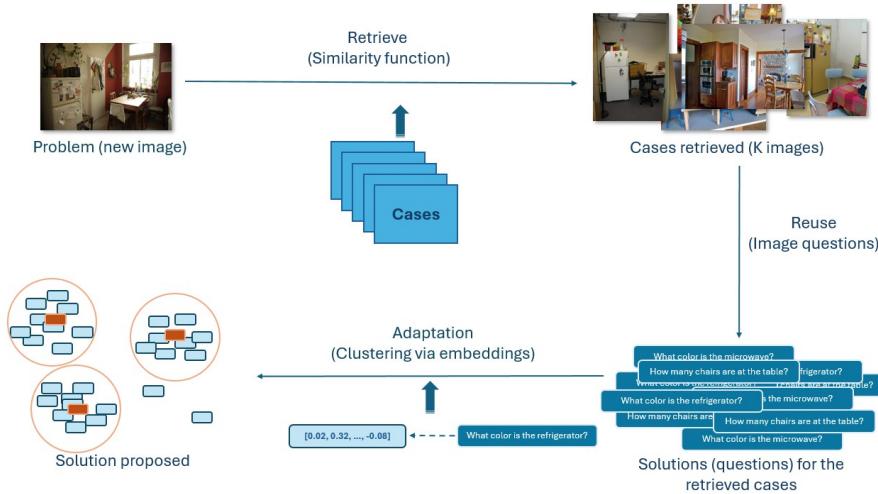


Figura 4.1: Flujo de recuperación de preguntas.



Figura 4.2: Imagen del dataset de COCO usada como referencia para ejemplificar las distintas métricas de similitud.

similitud entre imágenes con el fin de averiguar cuál es la más apta para lograr el objetivo planteado.

En el trabajo se han estudiado tres grupos de medidas de similitud: similitudes a nivel de píxel, similitudes a nivel de objetos detectados, y similitudes por *embeddings*. En las siguientes subsecciones se explicarán cada una de las métricas detalladamente, mostrando ejemplos de las imágenes más y menos similares a la imagen objetivo de ejemplo mostrada en la figura 4.2. Esta imagen es una de las que usaremos para la evaluación con usuarios.

#### 4.1.1. Similitud por píxeles

Hemos denominado similitud entre imágenes a nivel de píxel a aquellas medidas de similitud que consideran una comparación directa entre píxeles sin tener en cuenta otros aspectos. En este grupo hemos considerado dos métricas proporcionadas por la librería Scikit-image: *Normalized Root Mean Squared Error* y *Normalized Mutual Information*.

Existe un conflicto a la hora de calcular la similitud entre dos imágenes píxel por píxel cuando estas tienen un tamaño diferente, y este se puede resolver de varias maneras. Algunas alternativas incluyen extender la imagen con píxeles *dummy*, lo que podría provocar grandes diferencias debido a distintas dimensiones de las imágenes y no al contenido en sí, o tener en cuenta solo la intersección o coordenadas comunes, lo que implica no tener en cuenta gran parte de la imagen. Nosotros hemos optado por hacer un redimensionamiento de las imágenes a un tamaño común. Aunque este procedimiento pueda generar una distorsión en la imagen, nos permite comparar toda la imagen de manera equitativa. Se ha elegido un tamaño de 640 x 480, que es el tamaño modal de nuestro conjunto de imágenes representando un 20,26 % de las imágenes originales que componen la base de casos descrita en el capítulo 3. De esta manera, evitamos redimensionar el mayor número de imágenes posibles.

#### 4.1.1.1. Normalized Root Mean Squared Error (NRMSE)

Entendiendo las imágenes como una matriz de píxeles, esta medida consiste en calcular la diferencia entre los valores de los píxeles en cada coordenada, dando lugar al error en cada coordenada. Posteriormente se hace una media de los errores al cuadrado en todas las coordenadas de la imagen dando lugar al conocido Mean Squared Error o MSE. La raíz del MSE da lugar al RMSE o Root Mean Squared Error.

En la librería Scikit-image se describe el cálculo de NRMSE como<sup>2</sup>:

$$NRMSE(\mathbf{im}_A, \mathbf{im}_B) = \frac{\|\mathbf{im}_A - \mathbf{im}_B\|}{\|\mathbf{im}_A\|} \quad (4.1)$$

donde  $\mathbf{im}_A$ ,  $\mathbf{im}_B$  representan las dos imágenes comparadas como matrices de píxeles y  $\|\cdot\|$  es la norma de *Frobenius*.

Esta métrica indica la diferencia entre las imágenes, es decir, valores bajos indican una similitud alta mientras que valores bajos indican una similitud alta. Siguiendo un criterio uniforme, hemos establecido que para todas las métricas, valores altos indiquen una similitud alta y valores bajos indiquen una similitud baja. Siguiendo esta norma, para este caso se ha computado:

$$sim(\mathbf{im}_A, \mathbf{im}_B) = 1 - NRMSE(\mathbf{im}_A, \mathbf{im}_B) \quad (4.2)$$

En la figura 4.3 se muestra un ejemplo de la imagen más similar y disímil respecto a la imagen de referencia 4.2 siguiendo esta métrica.

#### 4.1.1.2. Normalized Mutual Information

En teoría de la probabilidad se conoce la información mutua entre dos variables aleatorias como una medida de la dependencia mutua entre las dos variables. En otras palabras, se describe como la cantidad de información conocida que se puede obtener de una variable observando la otra variable aleatoria.

---

<sup>2</sup>Aunque reciba el nombre de *Normalized*, la métrica no está siendo normalizada sino que se hace el cálculo indicado en la ecuación 4.1

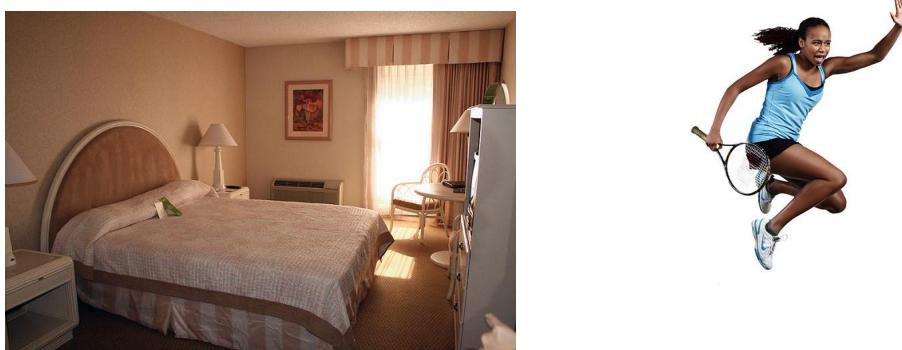


Figura 4.3: Ejemplo del cálculo de similitud con NRMSE respecto a la figura 4.2: La primera imagen tiene un valor de 0.4535 siendo la más similar en nuestra base de casos. La segunda imagen tiene un valor de -0.4771 siendo la menos similar en nuestra base de casos.

En [60] se plantea la utilidad de calcular la información mutua entre dos imágenes como una medida de comparación. Para ello, se trata la matriz de píxeles como una variable aleatoria.

En la librería Scikit-image se describe el cálculo como:

$$NMI(A, B) = \frac{H(A) + H(B)}{H(A, B)} \quad (4.3)$$

donde  $H(X) = -\sum_{x \in X} x \log x$  es la entropía de la variable X.

El valor del  $NMI$  devuelve un valor entre 1 (imágenes totalmente independientes) y 2 (imágenes totalmente relacionadas). Como vemos, este cálculo sigue el criterio establecido, sin embargo, para facilitar su comprensión y asemejar los valores a los de otras métricas, se ha computado:

$$sim(\mathbf{im}_A, \mathbf{im}_B) = NMI(\mathbf{im}_A, \mathbf{im}_B) - 1 \quad (4.4)$$

#### 4.1.2. Similitud por objetos

Para aplicar métricas de similitud que consideren los objetos que aparecen en la figura 4.4, lo primero que se necesita es un modelo para detectar dichos objetos. De entre la inmensa cantidad de modelos existentes, decidimos explorar RetinaNet [35] y YOLO [51] por su uso extendido, sus buenos resultados y su relativa eficiencia. Además, ambos modelos reconocen objetos de entre las clases definidas en COCO, lo cual es importante para poder comparar con las anotaciones de las imágenes. Las clases que definidas en el dataset de COCO son las representadas por iconos en la figura 4.4.

Para la detección de objetos hemos optado por el estándar que se utiliza en COCO de *bounding boxes*. Una *bounding boxes* es un área rectangular que delimita un

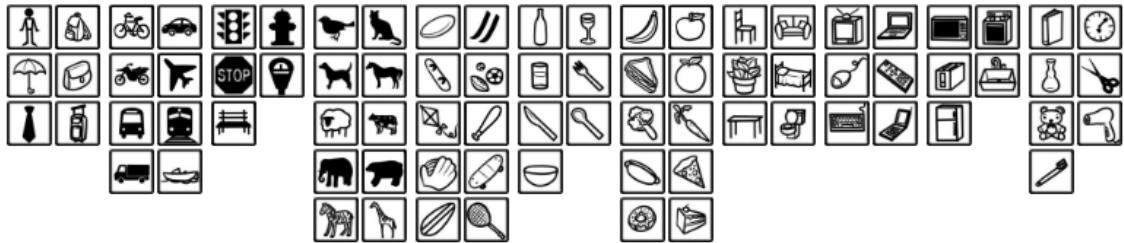


Figura 4.4: Clases de objetos definidas en COCO [65].

objeto en una imagen, esta caja viene definida por un par de coordenadas que indican la esquina superior izquierda de la caja, la anchura y la altura (en otros casos se define mediante dos vértices opuestos). Hemos elegido esta manera de detectar objetos por encima de una segmentación exacta debido a que se requieren considerablemente menos datos para representar una *bounding boxes*. En efecto, basta con dos pares de valores frente a la gran cantidad de puntos necesarios en la segmentación exacta de un objeto, lo que hace a la *bounding boxes* más rápida y manejable. Además, detectar objetos con precisión no es un objetivo principal del trabajo, por lo que consideramos las *bounding boxes* una aproximación suficientemente buena para nosotros.

Cada uno de estos modelos tiene sus características, que se exponen brevemente a continuación.

- **RetinaNet:** Se trata de un modelo de detección de objetos de una única etapa basado en ResNet50 [18]. Su éxito se debió principalmente a que consigue combinar una precisión propia de un modelo de dos etapas con la rapidez de un modelo de una etapa. Este hito lo alcanza esencialmente gracias a su función de pérdida focal, que introduce a la clásica pérdida de entropía cruzada un factor corrector que ayuda a disminuir el número de falsos negativos. El modelo lo componen un armazón (*backbone*) y una cabeza (*head*) que tomando como entrada la salida del armazón, se encarga de determinar la clase a la que pertenece el objeto detectado y sus coordenadas. El armazón a su vez se compone de una red residual (ResNet50) seguida de una Red Piramidal de Características (Feature Pyramid Net, FPN), que es la otra gran introducción de RetinaNet. La arquitectura del modelo se ilustra en la figura 4.5<sup>3</sup>.
- **YOLO:** El modelo You Only Look Once (YOLO) nace como un sistema de detección de objetos en tiempo real basado en DarkNet53 [51]. Por tanto su principal característica es su rapidez. Esta velocidad la consigue por su simplicidad, pues se trata de una única red neuronal de 24 etapas convolucionales intercaladas con capas de agrupación y dos últimas capas adicionales densamente conectadas. Al igual que el modelo anterior, YOLO tiene una función de pérdida basada en la entropía cruzada. Para su funcionamiento, la red neuronal divide la imagen en regiones, donde predice objetos y coordenadas con cierta probabilidad. Las capas iniciales de la red se encargan de la extracción de características de la imagen, mientras que las capas finales predicen la

<sup>3</sup><https://lamaquinadoraculo.com/deep-learning/deteccion-de-objetos-2-retinanet/>

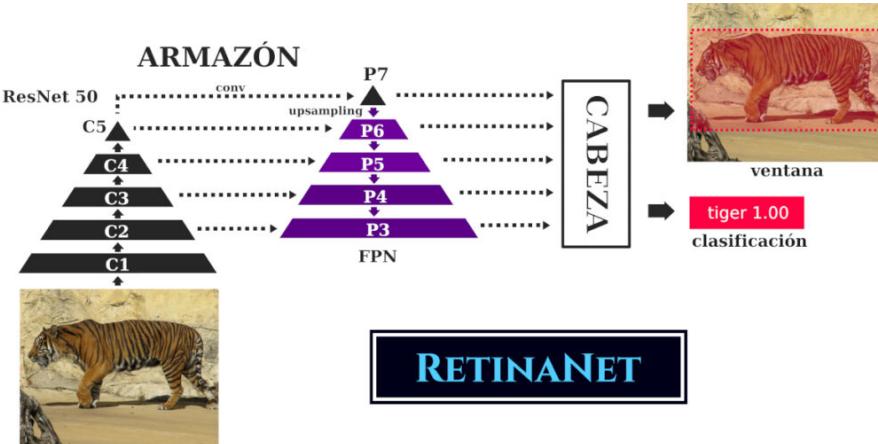


Figura 4.5: Esquema de arquitectura de RetinaNet.

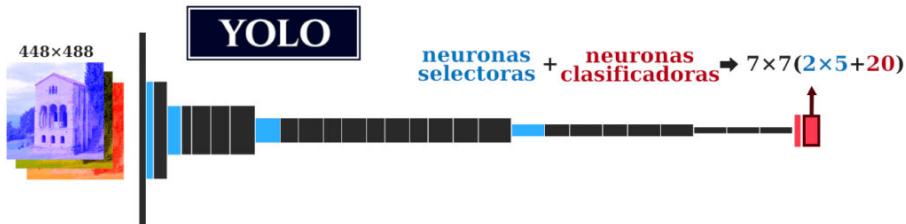


Figura 4.6: Esquema de arquitectura de YOLO.

probabilidad de salida y las coordenadas del objeto. La figura 4.6<sup>4</sup> ilustra la arquitectura del modelo.

Tanto RetinaNet como YOLO tienen como salida un nivel de confianza asociado al objeto detectado, que indica la seguridad del modelo sobre su detección. Por tanto, hemos de quedarnos con las detecciones que superen cierto umbral. Es importante notar que cuanto más bajo sea este umbral, más objetos lo superarán. Dado que las imágenes con las que vamos a comparar la imagen objetivo tienen anotaciones entre las que se encuentran los objetos, elegimos un umbral con el cual los objetos detectados con el modelo se asemejan lo más posible a los que aparecen en las anotaciones. De este modo las medidas de similitud no se ven tan afectadas por la diferencia en el número de objetos.

Los resultados obtenidos por nosotros en pruebas con diferentes tipos de imagen son muy parecidos, no apreciando un modelo claramente superior al otro. Por esta razón, acudimos a los resultados que han obtenido estos modelos sobre el dataset de COCO, representados en la figura 4.7. Estas evaluaciones arrojan que si bien YOLO obtiene buenos resultados con gran velocidad, RetinaNet tiene mayor precisión (*Average Precision*) a costa de ser algo más lento [45]. Para el uso que nosotros vamos a dar al modelo RetinaNet se ajusta mejor pues no requerimos detectar objetos en tiempo real, así que será el que usaremos.

Veamos ahora diferentes aproximaciones para definir la similitud de dos imágenes basándonos en los objetos que contienen.

<sup>4</sup><https://lamaquinoraculo.com/deep-learning/deteccion-de-objetos/>

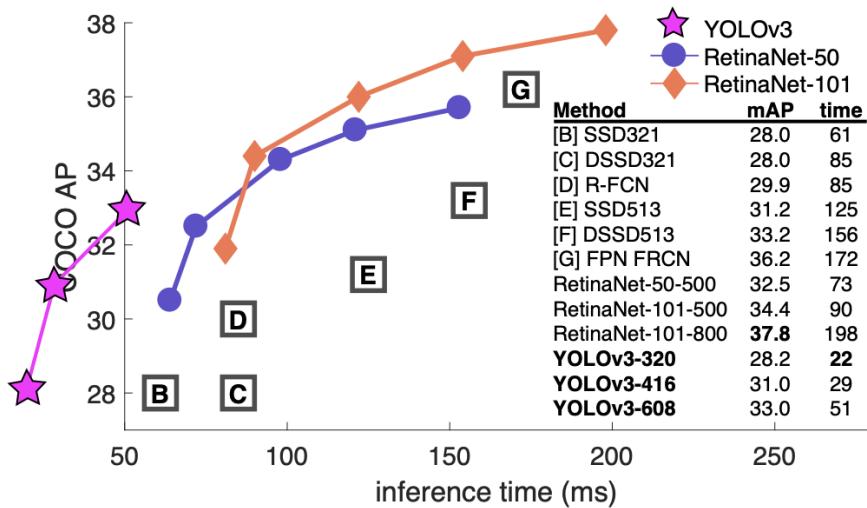


Figura 4.7: Comparación de modelos de detección de objetos en términos de *Average Precision* (AP) [51].

#### 4.1.2.1. Etiquetas como conjunto

Para el cálculo de la similitud entre imágenes basada en etiquetas como conjunto, se toman las etiquetas detectadas en la imagen objetivo y las de las anotaciones de la imagen del dataset. Estas etiquetas tomadas como conjuntos, es decir, sin relación de orden ni repeticiones, serán subconjuntos de *ClasesCOCO*, el conjunto de clases de objetos definido en COCO y representado en la figura 4.4. Definimos entonces esta medida de similitud como la intersección entre la unión de estos dos conjuntos.

$$sim(\mathbf{im}_A, \mathbf{im}_B) = \frac{|s_A \cap s_B|}{|s_A \cup s_B|} \quad (4.5)$$

donde  $s_A$  y  $s_B$  son los conjuntos de cada una de las imágenes y  $|\cdot|$  es el cardinal de un conjunto. Esta idea de la intersección entre la unión se basa en penalizar tanto los elementos que sobran como los que faltan en un conjunto para ser igual que el otro. Como observación, esta medida toma valores en  $[0, 1]$ , significando 0 conjuntos totalmente disímiles y 1 exactamente iguales.

En la figura 4.8 se muestra un ejemplo de la imagen más similar y disímil respecto a la imagen de referencia 4.2 siguiendo esta métrica.

#### 4.1.2.2. Etiquetas como multiconjunto

Siguiendo el planteamiento anterior, se tiene en cuenta ahora el número de veces que aparece cada etiqueta, es decir se toman como multiconjuntos o conjuntos con repeticiones. Se adapta entonces la intersección entre la unión, que se puede expresar ahora como sigue.

$$sim(\mathbf{im}_A, \mathbf{im}_B) = \frac{Int_{A,B}}{N_A + N_B - Int_{A,B}} \quad (4.6)$$



Figura 4.8: Ejemplo del cálculo de similitud con etiquetas como conjunto respecto a la figura 4.2: La primera imagen tiene un valor de 0.75 siendo la más similar en nuestra base de casos. La segunda imagen tiene un valor de 0 siendo la menos similar en nuestra base de casos.

donde  $N_A$  y  $N_B$  son el número de objetos en los multiconjuntos de  $A$  y  $B$ , y  $Int_{A,B}$  es la intersección de los multiconjuntos, que se puede escribir como

$$Int_{A,B} := \sum_{c \in C} \min(n_A^c, n_B^c) \quad (4.7)$$

siendo  $C := ClasesCOCO$  el conjunto de clases de objetos definido en COCO, y  $n_A^c$  (resp.  $n_B^c$ ) el número de objetos de la clase  $c$  en la imagen  $A$  (resp.  $B$ ). Nótese que se sigue interpretando el denominador como la unión, esta vez entre multiconjuntos y que del mismo modo que en la sección 4.1.2.1, la expresión 4.6 toma valores en  $[0, 1]$ .

#### 4.1.2.3. *Bounding boxes*

La idea de esta medida de similitud es capturar el tamaño de los objetos detectados, asumiendo que los objetos más grandes tendrán mayor importancia en la imagen. Primero se calcula cuánto ocupan los objetos de cada clase respecto del área total ocupada por objetos. Estos valores representan la importancia relativa de cada clase en la imagen. Después, se suman los valores mínimos para cada clase. Es fácil ver que al igual que en los casos anteriores el valor obtenido se encuentra en  $[0, 1]$  siendo 1 lo más similar.

$$sim(\mathbf{im}_A, \mathbf{im}_B) = \sum_{c \in C} \min\left(\frac{r_A^c}{r_A}, \frac{r_B^c}{r_B}\right) \quad (4.8)$$

donde  $C := ClasesCOCO$  es de nuevo el conjunto de clases de objetos definido por COCO,  $r_A^c$  (resp.  $r_B^c$ ) es la suma de las áreas de los objetos de la clase  $c$  en la imagen  $A$  (resp.  $B$ ), y  $r_A$  (resp.  $r_B$ ) es la suma de las áreas de todos los objetos en la imagen  $A$  (resp.  $B$ ).

En la figura 4.9 se muestra un ejemplo de la imagen más similar y disímil respecto a la imagen de referencia 4.2 siguiendo esta métrica.



Figura 4.9: Ejemplo del cálculo de similitud con *bounding boxes* respecto a la figura 4.2: La primera imagen tiene un valor de 0.82898 siendo la más similar en nuestra base de casos. La segunda imagen tiene un valor de 0 siendo la menos similar en nuestra base de casos.

#### 4.1.2.4. Extensión con superclases

Las clases que se definen en *ClasesCOCO* tienen asociada una superclase. Así, por ejemplo las clases *gato*, *oveja* o *perro* pertenecen a la superclase *animal*. Es razonable pensar que dos objetos pertenecientes a la misma superclase estarán más cerca semánticamente que dos de distinta superclase. Por tanto, se plantea una variante de las tres medidas anteriores en la cual se tiene esto en cuenta y se le da un valor a las coincidencias parciales, es decir a los casos en los que no coinciden las clases de los objetos pero sí sus superclases. El valor que se le da a las coincidencias parciales determina en gran medida estas variantes. Si éste es muy bajo será como no tomar en cuenta las superclases y si es muy alto será como equiparar una coincidencia total con una parcial.

#### 4.1.3. Similitud por *embeddings*

Los *embeddings* de imágenes son representaciones de baja dimensión que capturan, hasta cierto punto, las características semánticas de las imágenes. Se extraen de las capas internas de las redes neuronales ya entrenadas que usamos para vectorizar, que tienden a agrupar imágenes más cercanas con características comunes. Las similitudes basadas en *embeddings* consideran que dos imágenes son más similares cuanto más cerca estén sus *embeddings*.

Para obtener los *embeddings* de las imágenes, usamos el modelo Img2Vec de la librería PyTorch. Este modelo requiere de una red neuronal, por lo que hemos probado distintas arquitecturas de redes neuronales (resnet-18 [48], alexnet [24] , vgg-16 [57], densenet [19]) entrenadas con imágenes de ImageNet. La librería Img2Vec también permite elegir la capa de la red neuronal elegida de la que quieras extraer la salida como tu vector de *embeddings*, en este caso se usará la capa por defecto para cada modelo de red neuronal.

Cada uno de los cuatro modelos disponibles tienen sus propias características:

- **ResNet-18:** es una red CNN con una profundidad de 18 que a pesar de su simplicidad, entrenada con más de un millón de imágenes y que puede clasificar

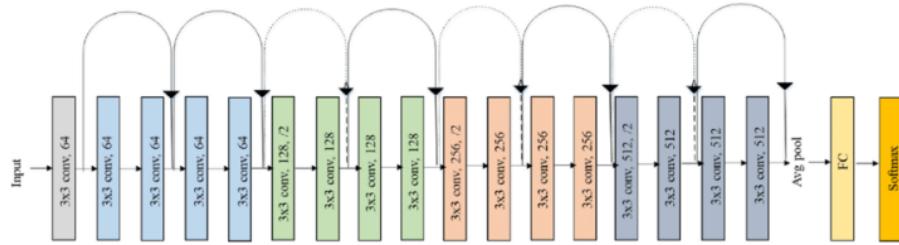


Figura 4.10: Arquitectura original de Resnet-18 [49].

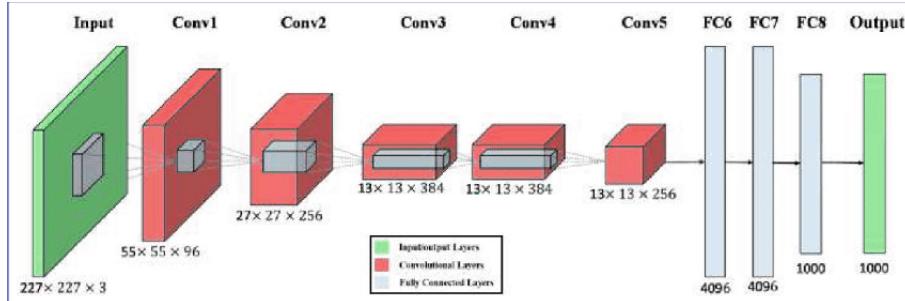


Figura 4.11: Arquitectura original de Alexnet [46].

objetos de las imágenes en más de 1000 categorías. En la figura 4.10 se muestra la arquitectura de esta red.

- **AlexNet:** es una red neuronal que tiene cinco capas convolucionales y tres capas densas, con un gran rendimiento en el reconocimiento visual de objetos en imágenes. Es importante mencionar que *AlexNet* fue la ganadora del desafío de reconocimiento visual a gran escala de ImageNet en 2012, marcando un hito en el campo del aprendizaje automático y el reconocimiento visual en imágenes. En la figura 4.11 se muestra la arquitectura de esta red.
- **Vgg-16:** es una arquitectura de CNN que consiste en 16 capas de convolución y *pooling*<sup>5</sup>, seguidas de tres capas totalmente conectadas. Es conocida por su simplicidad y efectividad en una variedad de tareas de clasificación de imágenes. Es usada como base para muchas otras arquitecturas de redes neuronales. En la figura 4.12 se muestra la arquitectura de esta red.
- **DenseNet:** es una arquitectura de CNN caracterizada por tener conexiones densas entre capas, lo que significa que cada capa está directamente conectada a todas las capas subsiguientes en el modelo [19]. Esto permite que el flujo de información a través de las diferentes capas sea más fluido y directo, lo que puede resultar en un mejor rendimiento y una mejor capacidad de generalización en comparación con otras arquitecturas. La arquitectura *DenseNet* logra un rendimiento muy cercano al estado del arte actual en varios conjuntos de

<sup>5</sup>El *pooling* (o agrupación) es una operación comúnmente utilizada en las redes neuronales convolucionales para reducir la dimensionalidad de las características mientras se preserva la información más importante.

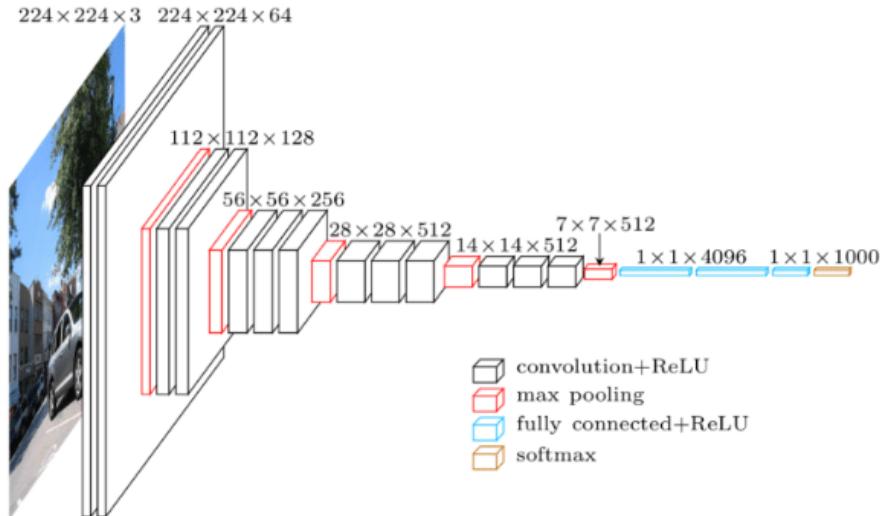


Figura 4.12: Arquitectura original de Vgg-16 [55].

datos de clasificación de imágenes usados de referencia. En la figura 4.13 se muestra la arquitectura de esta red.

No existen *benchmarks* generales de comparación de modelos de generación de *embeddings* de imágenes, pero si hay métricas para comparar estas arquitecturas en cuanto a tareas de clasificación. Para ayudarnos a decidir por uno de los modelos de generación de *embeddings* hemos comparado el rendimiento de las arquitecturas en la tarea de clasificación de ImageNet Classificaton<sup>6</sup>. A continuación se muestra la métrica *top 1 accuracy* para los cuatro modelos<sup>7</sup>:

- ResNet-18: 70.09 %
- AlexNet: 58.9 %
- VGG-16: 71.24 %
- DenseNet: 74.98 %

Aunque estos datos ayudan a hacerse una idea de que arquitecturas tienen un mejor rendimiento, es importante analizar qué modelo da mejores resultados para nuestro caso de uso particular. Para ello, realizamos nuestra propia comparativa de los cuatro modelos, pero esta vez midiendo la “calidad” de los *embeddings* generados de manera a través de la similitud entre imágenes. Esta comparativa consistió en calcular los *embeddings* de 30 imágenes aleatorias de nuestro conjunto de 5000, calculando las 3 imágenes más similares para cada modelo y utilizando los 3 tipos de distancia (utilizando 10 imágenes para cada tipo de similitud): distancia euclíadiana,

<sup>6</sup><https://paperswithcode.com/sota/image-classification-on-imagenet>

<sup>7</sup>Se han tomado los valores presentes en la tabla <https://paperswithcode.com/sota/image-classification-on-imagenet>. Debido a que hay varios modelos en la tabla que siguen cada una de estas arquitecturas, se ha tomado el criterio de elegir el valor más bajo.

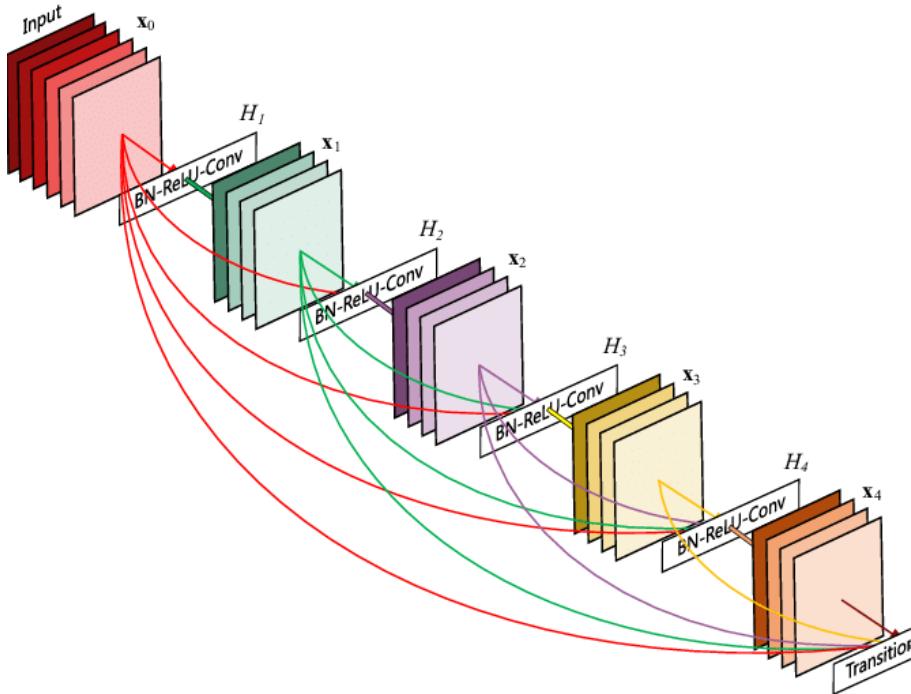


Figura 4.13: Arquitectura original de DenseNet [19].

distancia manhattan y similitud del coseno. En cada caso se eligió subjetivamente que modelo había recuperado las imágenes más similares para cada caso. Los resultados se muestran en la tabla 4.1.

	<b>ResNet-18</b>	<b>AlexNet</b>	<b>Vgg-16</b>	<b>DenseNet</b>
<b>Euclidean</b>	2	2	4	2
<b>Manhattan</b>	2	1	1	5
<b>Cosine</b>	2	1	2	4
<b>Total</b>	6	4	7	11

Tabla 4.1: Número de veces que se seleccionó cada modelo y tipo de distancia.

Como se muestra en la tabla 4.1 el modelo del que hemos seleccionado más imágenes similares fue *DenseNet*. Además tuvimos especialmente en cuenta el resultado obtenido para la similitud del coseno ya que tiene más peso en nuestro trabajo. Apoyándonos en este pequeño estudio y al rendimiento analizado anteriormente nos decantamos por el modelo *DenseNet*.

Una vez elegido el modelo de red neuronal y realizada la tarea de calcular los *embeddings* de las imágenes de la base de casos, definimos las similitudes basadas en estos *embeddings* para poder calcular la similitud entre dos imágenes. Sea  $\mathbf{v_A}$ ,  $\mathbf{v_B}$  los *embeddings* de las imágenes  $\mathbf{im_A}$  y  $\mathbf{im_B}$ ,  $(\cdot)$  representa el producto escalar entre dos vectores, y  $\|\cdot\|$  la norma euclídea de un vector, consideraremos la siguientes similitudes:



Figura 4.14: Ejemplo del cálculo de la similitud del coseno con *embeddings* respecto a la figura 4.2: La primera imagen tiene un valor de 0.6958 siendo la más similar en nuestra base de casos. La segunda imagen tiene un valor de -0.0939 siendo la menos similar en nuestra base de casos.

#### 4.1.3.1. Euclídea

Medida de similitud basada en la distancia euclídea entre dos vectores.

$$sim(\mathbf{im}_A, \mathbf{im}_B) = \sqrt{\sum_{i=1}^n (\mathbf{v}_{Ai} - \mathbf{v}_{Bi})^2} \quad (4.9)$$

#### 4.1.3.2. Manhattan

Es una medida de similitud basada en la distancia Manhattan entre dos vectores.

$$sim(\mathbf{im}_A, \mathbf{im}_B) = \sum_{i=1}^n |\mathbf{v}_{Ai} - \mathbf{v}_{Bi}| \quad (4.10)$$

#### 4.1.3.3. Coseno

Es una medida de similitud en la que se evalúa el valor del coseno del ángulo comprendido entre 2 vectores.

$$sim(\mathbf{im}_A, \mathbf{im}_B) = \frac{\mathbf{v}_A \cdot \mathbf{v}_B}{\|\mathbf{v}_A\| \|\mathbf{v}_B\|} \quad (4.11)$$

La similitud del coseno es particularmente útil cuando se trabaja con datos de alta dimensión, como los *embeddings* de imágenes, porque considera tanto la distancia como la dirección de cada vector. Esto la hace más robusta que las otras medidas euclídea y Manhattan que hemos considerado, que solo consideran la distancia.

En la figura 4.14 se muestra un ejemplo de la imagen más similar y disímil respecto a la imagen de referencia 4.2 siguiendo esta métrica.

## 4.2. Recuperación de preguntas y adaptación

Una vez recuperadas las imágenes más similares a la imagen que queremos describir, creamos una “bolsa” de preguntas candidatas a partir de las preguntas asociadas a estas imágenes, con el fin de conseguir una futura respuesta para generar la descripción. En esta sección se exploran diversas estrategias de adaptación para la recuperación de preguntas relevantes y variadas:

- **Selección de preguntas asociadas a las imágenes más similares.**

Este método consiste en recuperar todas las preguntas asociadas a las imágenes más similares. Sin embargo, en la práctica, si se recupera un número reducido de preguntas, generalmente todas suelen estar vinculadas a una sola imagen (la más similar). Esta estrategia suele ser poco efectiva ya que no es común que todas las preguntas asociadas a una imagen puedan ser aplicables en su totalidad a otra, por más similar que sea.

- **Selección de preguntas más cercanas al centroide de la bolsa de preguntas.**

Este enfoque se basa en la premisa de que las preguntas más frecuentes (o al menos el tipo de pregunta más común) de las asociadas a las imágenes más similares, pueden ser aplicables en nuestra imagen objetivo. En este segundo método, introducimos la generación de *embeddings* de las preguntas. Generando una representación vectorial de las preguntas, podemos calcular un centroide, y calcular las distancias de cada pregunta<sup>8</sup> a este centroide. Este cálculo nos permite obtener las preguntas “más cercanas” a la “pregunta media” de las asociadas al conjunto de imágenes similares. De esta manera obtenemos una mayor proporción de preguntas con correspondencia a nuestra imagen objetivo. Sin embargo, surge el problema de la variabilidad: las preguntas más cercanas al centroide tienden a ser preguntas muy similares entre sí, lo que no aporta mucha más información que tomar solo una de ellas.

- **Generación de *clusters* de preguntas y selección de representantes de cada *cluster*.**

Continuando con la idea anterior, podemos utilizar los *embeddings* para generar *clusters* o agrupaciones según el tipo de pregunta. De esta forma, las preguntas muy similares se agrupan en el mismo *cluster* y se selecciona un representante de cada grupo para evitar redundancias. Si bien este enfoque puede disminuir la precisión en la recuperación de preguntas con sentido para la imagen objetivo, añade una variabilidad necesaria para nuestro propósito final, que es obtener la mayor cantidad de información posible a través de las preguntas para generar una descripción completa.

---

<sup>8</sup>Cuando nos referimos a distancias en referencia a preguntas, nos referimos a su representación vectorial o *embedding*.

### 4.2.1. Generación de *embeddings* (GTE-small)

La generación de *embeddings* partir de texto se ha señalado como una parte fundamental de nuestro flujo de recuperación de preguntas. En esta sección nos adentramos en este concepto clave para comprender su papel y como se ha llevado a cabo su generación.

Un *embedding* es una representación vectorial de una frase<sup>9</sup>. Esta representación matemática está diseñada para capturar significado semántico y el contexto de las palabras que representa, lo que permite tratar el lenguaje natural de manera matemática.

Basándonos en la hipótesis de que las preguntas más recurrentes en imágenes similares son relevantes para la imagen objetivo, los *embeddings* son especialmente útiles. Esto se debe a que la representación vectorial una pregunta captura su valor semántico por lo que dos preguntas próximas semánticamente tendrán representaciones vectoriales cercanas aunque aunque las palabras que las compongan no sean las mismas. Por otro lado, la naturaleza vectorial de los *embeddings*, nos permite agrupar las preguntas por significado mediante *clustering*. De esta manera, obtenemos *clusters* que representan un aspecto sobre el que se pregunta con frecuencia en imágenes parecidas. La hipótesis que manejamos es que estos aspectos serán por extensión relevantes para nuestra imagen, así que se selecciona como representante de cada *cluster* la pregunta más cercana al centroide del grupo. De este modo, se consigue una selección final de preguntas relevantes, pues son sobre temas considerados relevantes en imágenes similares, y variadas, pues no se repiten preguntas de cada aspecto importante detectado.

De entre las opciones descritas en la sección 2, hemos empleado el modelo *GTE-small*. La elección de este modelo se debe al equilibrio entre el rendimiento en varios *benchmarks* generales, como se observa en la tabla 4.2, el tamaño de los *embeddings* generados, y el reducido número de parámetros del modelo. Es importante dar importancia a la dimensión de los *embeddings*, dado que tanto la comparación como el proceso de “clusterización” pueden verse afectados negativamente por una alta dimensionalidad de los datos.

### 4.2.2. *Clustering* (DBSCAN)

La generación de *clusters* es una técnica de aprendizaje automático que sirve para agrupar datos similares en conjuntos llamados *clusters*. Es un procedimiento especialmente útil cuando quieras agrupar o clasificar datos no etiquetados. En nuestro flujo, es el proceso responsable de añadir variabilidad a nuestra selección final de preguntas, ya que como hemos mencionado, agrupamos las preguntas por significado semántico, seleccionando un representante de cada tipo evitando así la redundancia semántica.

Existen diversos algoritmos de *clustering* dependiendo la técnica en la que se basan, algunos de los más conocidos son el *clustering* jerárquico, *clustering* basado en densidad o *k-means*. Nosotros hemos decidido optar por DBSCAN (*clustering*

---

<sup>9</sup>Un *embedding* puede representar desde una palabra hasta un texto con varios párrafos. En otro contexto también puede representar también una imagen o un audio.

# of datasets →	Params	Class.	Clust.	Pair.	Rerank	Retr.	STS	Summ.	Avg
		12	11	3	4	15	10	1	56
<i>Unsupervised models</i>									
Glove	120M	57.3	27.7	70.9	43.3	21.6	61.9	28.9	42.0
BERT	110M	61.7	30.1	56.3	43.4	10.6	54.4	29.8	38.3
SimCSE	110M	62.5	29.0	70.3	46.5	20.3	74.3	31.2	45.5
E5 <small>small</small>	30M	67.0	41.7	78.2	53.1	40.8	68.8	25.2	54.2
E5 <small>base</small>	110M	67.9	43.4	79.2	53.5	42.9	69.5	24.3	55.5
E5 <small>large</small>	330M	69.0	44.3	80.3	54.4	44.2	69.9	24.8	56.4
GTE <small>small</small>	30M	71.0	44.9	82.4	57.5	43.4	77.2	30.4	58.5
GTE <small>base</small>	110M	71.5	46.0	83.3	58.4	44.2	76.5	29.5	59.0
GTE <small>large</small>	330M	71.8	46.4	83.3	58.8	44.6	76.3	30.1	59.3
<i>Supervised models</i>									
SimCSE	110M	67.3	33.4	73.7	47.5	21.8	79.1	23.3	48.7
Contriever	110M	66.7	41.1	82.5	53.1	41.9	76.5	30.4	56.0
GTR <small>large</small>	330M	67.1	41.6	85.3	55.4	47.4	78.2	29.5	58.3
Sentence-T5 <small>large</small>	330M	72.3	41.7	85.0	54.0	36.7	81.8	29.6	57.1
E5 <small>small</small>	30M	71.7	39.5	85.1	54.5	46.0	80.9	31.4	58.9
E5 <small>base</small>	110M	72.6	42.1	85.1	55.7	48.7	81.0	31.0	60.4
E5 <small>large</small>	330M	73.1	43.3	85.9	56.5	50.0	82.1	31.0	61.4
InstructOR <small>base</small>	110M	72.6	42.1	85.1	55.7	48.8	81.0	31.0	60.4
InstructOR <small>large</small>	330M	73.9	45.3	85.9	57.5	47.6	83.2	31.8	61.6
OpenAI <small>ada-001</small>	n.a.	70.4	37.5	76.9	49.0	18.4	78.6	26.9	49.5
OpenAI <small>ada-002</small>	n.a.	70.9	45.9	84.9	56.3	49.3	81.0	30.8	61.0
GTE <small>small</small>	30M	72.3	44.9	83.5	57.7	49.5	82.1	30.4	61.4
GTE <small>base</small>	110M	73.0	46.1	84.3	58.6	51.2	82.3	30.7	62.4
GTE <small>large</small>	330M	73.3	46.8	85.0	59.1	52.2	83.4	31.7	63.1
<i>Larger models</i>									
InstructOR <small>xl</small>	1.5B	73.1	44.7	86.6	57.3	49.3	83.1	32.3	61.8
GTR <small>xxl</small>	4.5B	67.4	42.4	86.1	56.7	48.5	78.4	30.6	59.0
Sentence-T5 <small>xxl</small>	4.5B	73.4	43.7	85.1	56.4	42.2	82.6	30.1	59.5

Tabla 4.2: Comparación de resultados entre diversos modelos de generación de *embeddings* y las variantes del modelo GTE [32] en el *benchmark MTEB* [41] (evaluado en 56 datasets en inglés), destacando el balance entre el tamaño y el rendimiento del modelo *GTE-small*.

basado en densidad) [16], ya que es menos sensible a la presencia de ruido que otros algoritmos como *K-means* y no requiere de especificar el número de *clusters* de antemano. En nuestro caso, esto supone una ventaja, ya que las bolsas de preguntas recuperadas para cada caso pueden ser muy distintas entre sí y pueden tener diferentes estructuras de agrupamiento. Hemos elegido la implementación DBSCAN de la librería *sklearn.cluster* de Python.

Además del conjunto de datos el algoritmo DBSCAN necesita dos parámetros:  $\varepsilon$  y  $Min$ . En nuestro caso hemos optado por  $\varepsilon = 0.45$  y  $Min = 5$ . Estos parámetros

se han determinado a través de experimentación y generan entre 3 y 6 *clusters* en la mayoría de los casos. Si tenemos en cuenta que recuperamos 4 preguntas finales, esta cantidad de *clusters* nos permite obtener un representante de casi todos los *clusters* generados, incluso tomando dos representantes del *cluster* más poblado si fuera necesario.

## 4.3. Evaluación de preguntas

Con el fin de evaluar qué medida de similitud de las detalladas en la sección 4.1 encontraba las mejores imágenes, se realizó una evaluación mediante un experimento con usuarios.

El principal objetivo del experimento realizado fue la evaluación de las diferentes aproximaciones para medir la similitud en términos de eficacia recuperando imágenes con preguntas asociadas semánticamente relevantes. Se han expuesto un elevado número de medidas de similitud, sin embargo de cara al experimento se escogieron 4 para que hubiera suficientes datos sobre ellas como para tomar conclusiones fundamentadas. No obstante, no se descarta hacer pruebas en un futuro con las medidas de similitud descartadas, o incluso con diferentes combinaciones de medidas. En definitiva, se eligieron como representantes de las diferentes aproximaciones aquellas medidas que se consideraron mejores tras una evaluación subjetiva. En esta evaluación se tuvo en cuenta la explicabilidad, sencillez y comportamiento observado en pruebas con diferentes tipos de imagen. Así, para el experimento se usaron las siguientes medidas:

- Similitud basada en píxeles usando NRMSE (sección 4.1.1.1)
- Similitud basada en objetos usando etiquetas (sección 4.1.2.1)
- Similitud basada en objetos usando *bounding boxes* (sección 4.1.2.3)
- Similitud basada en *embeddings* usando distancia del coseno (sección 4.1.3.3)

La elección de las medidas de similitud se hizo atendiendo a razones de diversidad, eligiendo aquellas que creíamos mejores de cada grupo.

### 4.3.1. Hipótesis

Previamente a la realización del experimento, enunciamos las siguientes hipótesis sobre los resultados que se esperan obtener:

1. Debido a su falta de información semántica, conjeturamos que la similitud basada en píxeles mostrará un desempeño pobre en el test. Esperamos que recupere imágenes que si bien tendrán tonos parecidos, carecerán de significados semejantes, lo que inducirá a preguntas poco adecuadas.
2. Las medidas basadas en detección de objetos incorporan información semántica, sin embargo ésta se encuentra limitada por las categorías definidas en el dataset de COCO. Por tanto, creemos que encontrará dificultades en imágenes con pocos o ningún objeto perteneciente a estas categorías, como paisajes, e hipotetizamos que este tipo de imágenes mostrará una debilidad de estas medidas.
3. Las medidas de similitud basadas en *embeddings* también incorporan información semántica. De hecho, creemos que la elevada cantidad de variables

**IMAGEN 5**

Seleccione las preguntas cuya respuesta ayudaría a describir la imagen.

Imagen 5/10

What airline should you fly?  
 Is the table made of oak?  
 What animal is this?  
 Is there a refrigerator in this picture?  
 Is there a shark in the water?  
 Is it cold outside?  
 What type of room is this?  
 What is the floor made of?  
 What color are the curtains?  
 Is this kitchen refurbished?  
 Is this a kitchen?  
 What kind of flower is used in the vases?  
 What game is on the TV?  
 What door is open?  
 How many chairs?  
 Which room is this?  
 What color are the cupboards in this photo?

Figura 4.15: Ejemplo de una de las imágenes del experimento.

tomadas en cuenta en la vectorización conferirá a estas medidas mayor robustez recuperando imágenes parecidas. Esperamos por tanto que se comporte más consistentemente bajo un amplio espectro de imágenes.

### 4.3.2. Configuración del experimento

Para la realización del experimento<sup>10</sup> se seleccionaron imágenes con las que ninguno de los modelos utilizados se hubiera entrenado. Además se trató de tomar muestras de diferentes ámbitos (animales, paisajes, transporte, ciudad, tecnología, habitaciones, deportes y comida) de forma que se cubriera un espectro lo más amplio posible.

Se tomaron por tanto 20 imágenes de Pixaby<sup>11</sup> para el experimento y se usaron las 4 medidas de similitud mencionadas al comienzo de la sección para recuperar 4 preguntas con cada una. La recuperación se llevó a cabo usando las 50 imágenes más semejantes según cada medida de similitud y usando el método de “clusterización” descrito en la sección 4.2.2. Además, se introdujo un *baseline* compuesto por 4 preguntas, representantes de los 4 tipos de pregunta más comunes en la base de casos. Más concretamente, se usó la misma “clusterización” con todas las preguntas y se tomaron representantes de los 4 grupos más numerosos. Así, en el experimento se contó con 20 imágenes y para cada una de las cuales se presentaban 20 preguntas aleatoriamente ordenadas y sin ninguna indicación sobre el modelo del que procedían, tal y como se muestra en la figura 4.15.

En aras de incentivar la participación se decidió dividir las 20 imágenes en dos cuestionarios en línea con 10 imágenes cada uno, de forma que se pudieran completar

<sup>10</sup>A: <https://forms.gle/immm57KVk5c7uGyXz5>, B: <https://forms.gle/5bZ2GYkAei4wU14fA>

<sup>11</sup><https://pixabay.com/>

Similitud	Preguntas seleccionadas(%)
<i>Baseline</i>	6.43 %
Píxeles	17.78 %
Etiquetas	48.22 %
<i>Bounding boxes</i>	54.58 %
<i>Embeddings</i>	65.53 %

Tabla 4.3: Los porcentajes mostrados representan cuánto se escogió cada modelo, no una tasa de éxito general. Los valores indican la proporción de preguntas seleccionadas por los usuarios de cada modelo. Un 100 % significaría que todos los usuarios eligieron todas las preguntas presentadas del modelo.

en menos de 10 minutos. A los participantes del experimento se les mostraban las 10 preguntas junto con la imagen correspondiente y se les pedía marcar aquellas preguntas *cuya respuesta considerasen que ayudaría a construir una descripción de la imagen*. El cuestionario admitía varias respuestas, de forma que los participantes pudieran marcar una pregunta, varias, todas o ninguna.

#### 4.3.3. Resultados

Los dos cuestionarios en los que se dividieron las imágenes (A y B) se difundieron a personas sin conocimiento específico, pues no es necesario el dominio de ningún área para responderlos. Se llenaron 80 y 68 veces cada uno en un total de 8 días. El número medio de preguntas seleccionadas por imagen fue de 7,08 de 20.

La evaluación se llevó a cabo analizando los ratios de selección de las preguntas recuperadas usando cada medida de similitud. De este modo se consigue eliminar el efecto de la diferencia de respuestas entre los cuestionarios A y B. Los resultados generales se muestran en la tabla 4.3.

Tal y como se esperaba, la aproximación simplista de recuperar las preguntas más comunes de la base de casos no da buenos resultados. La similitud basada en píxeles es algo mejor pero no sustancialmente. Las medidas de similitud que consideran la información semántica de las imágenes parece una mejor opción pues obtienen resultados bastante mejores. En cuanto a las medidas basadas en detección de objetos es interesante destacar que parece que el tamaño de los objetos en la imagen tiene relevancia. A pesar de todo, de acuerdo a los datos recopilados la similitud basada en *embeddings* es la clara ganadora, obteniendo los mejores resultados con consistencia. Creemos que usando los *embeddings* de DenseNet para recuperar las imágenes se va más allá de la detección de objetos, teniendo en cuenta más variables que contribuyen a la similitud semántica entre imágenes y potencialmente obteniendo preguntas más relevantes.

Más allá de este análisis general, es interesante ver los resultados obtenidos dividiendo las imágenes distintas categorías. Para ello haremos una primera distinción entre imágenes tomadas al aire libre y en interiores. Dentro del primer grupo dividiremos también entre imágenes de entornos naturales y urbanos. En cuanto a imágenes de interiores separaremos entre aquellas que se enfocan en un elemento es-

Similitud	Preguntas Seleccionadas( %)					
	Interior			Exterior		
	Habitación	Elemento	Todas	Urbano	Natural	Todas
Baseline	16.54 %	5.37 %	10.34 %	2.48 %	5.27 %	3.34 %
Píxeles	17.28 %	4.49 %	10.18 %	21.32 %	31.13 %	24.00 %
Etiquetas	51.10 %	46.99 %	48.82 %	54.04 %	30.88 %	47.73 %
Bounding boxes	66.54 %	53.09 %	59.07 %	54.46 %	41.42 %	50.90 %
Embeddings	76.38 %	61.62 %	68.18 %	62.50 %	65.58 %	63.37 %

Tabla 4.4: Ratio de selección para cada modelo separado por tipo de imagen. Análogo a la tabla 4.3.

pecífico y aquellas que muestran planos más amplios como habitaciones. Siguiendo esta clasificación, analizamos los resultados obtenidos y recogidos en la tabla 4.4.

- **Imágenes de habitaciones (4 imágenes del test).** Las similitudes basadas en detección de objetos se comportaron razonablemente bien dado a que este tipo de imágenes suele contener objetos reconocibles como sillas, camas o mesas. Por otro lado, la similitud basada en píxeles no obtiene buenos resultados debido a la disparidad de colores y formas que aparecen en estas imágenes. Finalmente, la similitud basada en *embeddings* suele ser la más efectiva en este grupo, presentando los mejores valores para 3 de las 4 imágenes.
- **Imágenes enfocadas en un elemento específico (5 imágenes del test).** De nuevo la similitud basada en *embeddings* es la que mejor se comporta para este grupo. En cuanto a las medidas de similitud basadas en la detección de objetos, en este grupo acusan especialmente su limitación a la hora de detectar objetos fuera de las categorías de COCO pues si el elemento protagonista de la imagen no se detecta se prescinde de información semántica muy valiosa. Por último, la similitud basada en píxeles vuelve a tener un pobre desempeño, probablemente por la variedad de colores y formas que presentan los objetos del día a día.
- **Imágenes de ambientes urbanos (8 imágenes del test).** En este grupo la similitud basada en *embeddings* y las basadas en detección de objetos son las que mejor papel han tenido, aunque en casos diferentes. Por un lado, en imágenes plagadas de objetos la similitud basada en *embeddings* se comporta mejor, probablemente debido a la mejor abstracción que hace sobre la imagen. En aquellas imágenes que no estaban tan cargadas, sin embargo, la detección de objetos ha estado a la altura, siendo la mejor en ocasiones. La similitud basada en píxeles solo ha obtenido buenos resultados en imágenes donde la tonalidad jugaba un papel semántico (tonos grises en días lluviosos o tonos cálidos en días soleados).
- **Imágenes de ambientes naturales (3 imágenes del test).** En esta categoría las medidas de similitud basadas en detección de objetos tienen peor desempeño que en las anteriores porque en esta clase de imágenes no suelen

aparecer muchos objetos, sin embargo mejoran sustancialmente cuando hay animales presentes. La similitud basada en píxeles se comporta razonablemente bien en este grupo, dadas sus limitaciones. Especialmente obtiene resultados decentes en imágenes donde el color es uniforme y significativo como imágenes del mar, el cielo o un prado. Sin embargo, a pesar de obtener mejores resultados que en otras categorías, sigue estando por detrás de la detección de objetos. Finalmente, con los mejores datos se encuentra de nuevo la similitud basada en *embeddings*.

En resumen, la evaluación que hemos llevado acabo muestra que la similitud basada en *embeddings* (con la medida del coseno) ofrece los mejores resultados recuperando imágenes. Esta aproximación ha recuperado preguntas apropiadas para el usuario en más de un 60 % de las ocasiones, siendo con diferencia el mejor resultado. En cuanto a la similitud basada en detección de objetos, ha demostrado un desempeño aceptable cuando el modelo detectaba objetos relevantes en la imagen. No obstante, el modelo mostró carencias cuando se trataba de paisajes o imágenes sin objetos reconocibles. Por otro lado, es destacable que la incorporación del tamaño de los objetos mejora los resultados en hasta un 10 %. Finalmente, la similitud basada en píxeles ha sido la clara perdedora, llegando a ser superada por el baseline en ciertas ocasiones. Su único punto fuerte han sido las imágenes donde el tono juega un papel relevante o tiene valor semántico. Aun así, en estos casos apenas igualaba el peor resultado de la similitud basada en detección de objetos.

En este capítulo se ha explicado el proceso de recuperación de preguntas, pasando por el estudio y selección de una medida de similitud para calcular imágenes similares a la imagen objetivo y detallando el método de recuperación de las preguntas más relevantes. En el siguiente capítulo veremos el proceso de construcción de la descripción a partir de las preguntas recuperadas en el capítulo anterior, explorando sistemas VQA para responder las preguntas y estudiando los modelos de generación de descripciones.

# Capítulo 5

## Construcción de la descripción

Recordemos que nuestro objetivo final es generar una descripción textual de la imagen objetivo basada en la experiencia y conocimiento previos que nos aporta el conjunto de casos del que disponemos. Para ello, hemos recuperado preguntas variadas y relevantes para nuestra imagen a partir de imágenes semejantes en nuestro conjunto de casos. Ahora, para construir la descripción que queremos, hemos de responder esas preguntas para la imagen objetivo y redactar un texto descriptivo basado en las respuestas a estas preguntas.

Siguiendo el razonamiento anterior, se construye la descripción en dos pasos. En un primer paso, se utiliza un modelo para responder las preguntas recuperadas sobre la imagen objetivo y en un segundo paso, se hace uso de otro modelo que recoja estas respuestas y componga un texto coherente con ellas, a modo de descripción.

### 5.1. Respuesta a las preguntas

El primer paso una vez que se han extraído las preguntas es obtener una respuesta para nuestra imagen. Esta tarea la delegaremos en un modelo de VQA que tome como entrada la pregunta en lenguaje natural y la imagen objetivo y dé como salida una respuesta en lenguaje natural. Siguiendo la misma línea que hasta ahora, buscamos un modelo relativamente *pequeño*, esto es que no se haya entrenado con cantidades masivas de datos y no tenga un número de parámetros de entrada muy elevado.

Con estos requisitos no buscamos un modelo que sea capaz de dar una respuesta redactada a la pregunta sino más bien un clasificador que dada una imagen y una pregunta proponga como respuesta una palabra (o palabras). El modelo calculará la probabilidad de pertenencia a cada una de las clases, que representan las posibles respuestas, y devolverá aquella que resulte más probable.

Así, centramos nuestra atención en *Vision-and-Language Transformer* (ViLT) [23]. ViLT tiene una arquitectura muy simple para ser un modelo de visión y lenguaje, a la vez que un desempeño más que competente. Esta arquitectura se enmarca dentro de la categoría de modelos VLP (*Vision-and-Language Pre-training*) pero se diferencia bastante de los enfoques tradicionalmente empleados. Su principal ca-

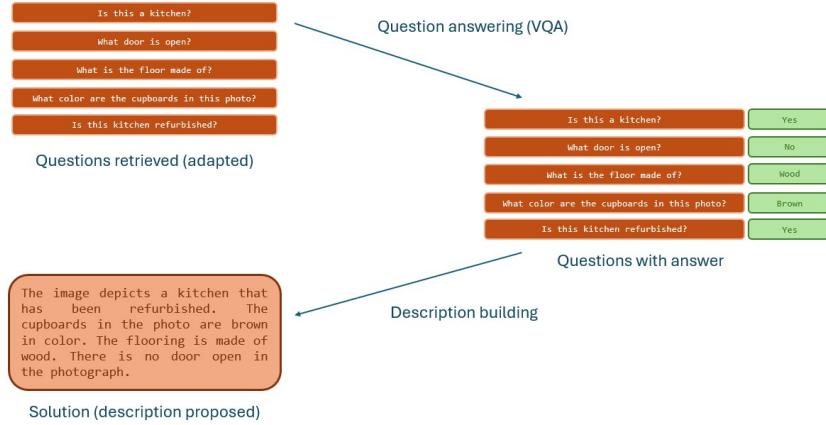


Figura 5.1: Flujo de generación de la descripción desde la obtención de las preguntas.

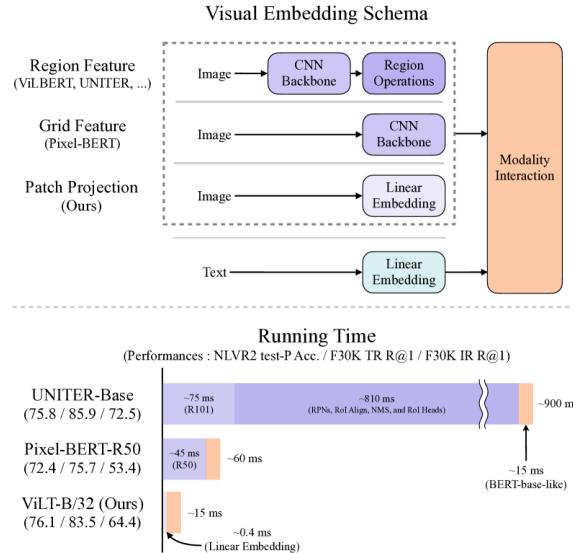


Figura 5.2: Comparación entre los modelos VLP tradicionales y ViLT [23].

Característica es que evita el uso de redes convolucionales profundas para generar los *embeddings* de las imágenes, lo que aligera bastante el modelo tanto en tamaño como en tiempo de ejecución.

Hasta que surgió este modelo, había dos formas de enfocar la generación de *embeddings* de la imagen, que suele ser el cuello de botella de los modelos VLP. Por un lado, existía un enfoque más centrado en características por regiones (*bottom-up*) normalmente obtenidas de detectores de objetos. Por otro, hay modelos que usan la salida de redes convolucionales como ResNet en forma de *grid* de características. ViLT aporta una tercera vía, proponiendo usar proyecciones lineales de área, que ha demostrado estar a la altura de las otras dos con un coste menor en tiempo y recursos, como se aprecia en la tabla 5.1 y en la figura 5.2.

Por lo expuesto anteriormente, decidimos usar este modelo para responder las preguntas obtenidas para la imagen objetivo. La salida de ViLT consiste normal-

Visual Embed	Modelo	Param (M)	FLOPs (G)	Tiempo (ms)	VQAv2 test-dev	NLVR2 dev	NLVR2 test
<i>Región</i>	ViLBERT	274.3	958.1	~920	70.55	-	-
	VisualBERT	170.3	425.0	~925	70.80	67.40	67.00
	LXMERT	239.8	952.0	~900	72.42	74.90	74.50
	UNITER-Base	154.7	949.9	~900	72.70	75.85	75.80
	OSCAR-Base	154.7	956.4	~900	73.16	78.07	78.36
<i>Grid</i>	VinVL-Base	157.3	1023.3	~650	75.95	82.05	83.08
	Pixel-BERT-X152	144.3	185.8	~160	74.45	76.50	77.20
<i>Lineal</i>	Pixel-BERT-R50	94.9	136.8	~60	71.35	71.70	72.40
	ViLT-B/32	87.4	55.9	~15	71.26	75.70	76.13

Tabla 5.1: Comparación de ViLT con otros modelos en tareas de clasificación *downstream* [23].

mente en una palabra (en ocasiones son varias palabras con un significado conjunto e.g. *laying down*) junto con la confianza (seguridad del modelo en su respuesta), por lo tanto necesitamos procesar estas respuestas junto con sus preguntas para obtener una descripción redactada de la imagen.

## 5.2. Composición de la descripción

Una vez que el modelo VQA responde a las preguntas para nuestra imagen, el paso siguiente es construir una descripción basada en estas preguntas y respuestas. Para ello, al igual que para el modelo VQA previo, hemos buscado un modelo relativamente *pequeño* dentro del estado del arte actual, ya que no necesitamos que el modelo genere contexto sobre las imágenes para generar las descripciones, lo que reduce considerablemente el número de parámetros necesarios para satisfacer nuestras necesidades.

De esta manera, estudiamos los modelos StableLM, una serie de modelos de lenguaje de código abierto desarrollados por Stability AI<sup>1</sup>. Dentro de toda la gama de modelos, principalmente estudiamos el rendimiento de dos: *Stablelm-2-zephyr-1\_6b* y *Stablelm-zephyr-3b*. Su principal diferencia es en tamaño del número de parámetros, 1.6 y 3 mil millones respectivamente.

Básandonos en la figura 5.2<sup>2</sup> que compara los principales modelos LM respecto a la métrica MT-Bench (Multi-turn Benchmark), como era de esperar, tiene mejor rendimiento *Stablelm-zephyr-3b*. MT-bench está diseñado para probar la capacidad de conversación de los modelos y su habilidad de seguir instrucciones, cubriendo casos de uso comunes y centrándose en preguntas desafiantes para diferenciar modelos [74]. Sin embargo, además de esto, llevamos a cabo pruebas con ambos modelos en distintas imágenes para juzgar si la relación tamaño/rendimiento era lo suficientemente bueno para escoger *Stablelm-zephyr-3b* por delante de *Stablelm-2-zephyr-1\_6b*. Tras múltiples pruebas, concluimos que las descripciones del modelo de 3 mil millones de parámetros se adecuaban mejor a las preguntas y sus respuestas ya que tenía una mejor comprensión del *prompt* de entrada que se le indicaba. Por último, creemos que la diferencia de tamaño entre ambos modelos no es tanta, y que un modelo de 3 mil millones de parámetros también es considerado lo suficientemente pequeño para ser usado en este contexto. En la figura 5.3<sup>3</sup> podemos observar un desglose del rendimiento de *Stablelm-zephyr-3b* para varias tareas medidas en MT-Bench. En ella se aprecia como el rendimiento es mejor en la mayoría de tareas que modelos entrenados con muchos más parámetros (como *Falcon-40B-Instruct* entrenado con 40 mil millones de parámetros) y es muy similar a modelos ampliamente conocidos y utilizados como *GPT-3.5-Turbo*. Por todo ello, decidimos usar el modelo *Stablelm-zephyr-3b* con 3 mil millones de parámetros para generar la descripción de la imagen a partir de las preguntas y respuestas obtenidas en la sección 5.1.

### 5.2.1. Parámetros para la composición de la descripción

En esta sección se van a explicar los diferentes parámetros configurables que nos pueden permitir ajustar el rendimiento generando descripciones de nuestro modelo.

#### Número de preguntas recuperadas

La composición de las descripciones depende en gran medida de la selección

---

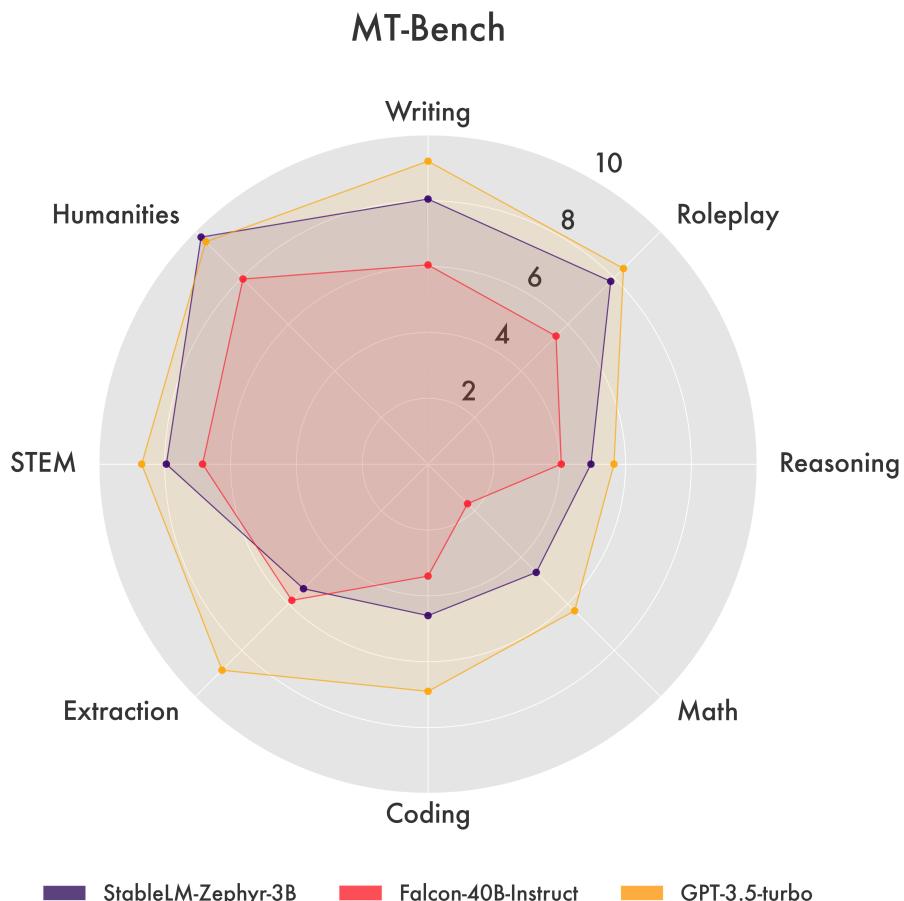
<sup>1</sup><https://stability.ai/stable-lm>

<sup>2</sup>[https://huggingface.co/stabilityai/stablelm-2-zephyr-1\\_6b](https://huggingface.co/stabilityai/stablelm-2-zephyr-1_6b)

<sup>3</sup><https://huggingface.co/stabilityai/stablelm-zephyr-3b>

Model	Size	MT-Bench
Mistral-7B-Instruct-v0.2	7B	7.61
Llama2-Chat	70B	6.86
stablelm-zephyr-3b	3B	6.64
MPT-30B-Chat	30B	6.39
<b>stablelm-2-zephyr-1.6b</b>	1.6B	5.42
Falcon-40B-Instruct	40B	5.17
Qwen-1.8B-Chat	1.8B	4.95
dolphin-2.6-phi-2	2.7B	4.93
phi-2	2.7B	4.29
TinyLlama-1.1B-Chat-v1.0	1.1B	3.46

Tabla 5.2: Evaluación del rendimiento de modelos LM de código abierto.

Figura 5.3: Comparación del rendimiento de *Stablelm-zephyr-3b* con otros dos LMs de código abierto más grandes en cada tarea evaluada en MT-Bench.

final de preguntas, dado que las respuestas a estas son empleadas por el modelo *Stablelm-zephyr-3b* para generar las descripciones. El número de preguntas recuperadas constituye, por tanto, un parámetro ajustable en nuestro modelo propuesto.

Se debe considerar, como se detalla en el capítulo 4, que las preguntas recuperadas pueden ser redundantes entre sí semánticamente, por lo que incrementar la

cantidad de preguntas recuperadas no garantiza un mayor cantidad de información y puede generar ruido. Es muy importante a la hora de trabajar con modelos grandes de lenguaje (*LLMs* o *LMs*) ser concisos y no aportar información innecesaria (ruido), pues puede provocar un comportamiento no deseado del LM.

### **Prompt** de entrada

Todo modelo grande de lenguaje, incluido *Stablelm-zephyr-3b*, necesita un *prompt* de entrada o input de usuario con las instrucciones necesarias para generar un texto a partir de él. En este input es donde indicamos a *Stablelm-zephyr-3b* la tarea de generar una descripción a partir de las preguntas y respuestas obtenidas en el paso anterior. El *prompt* de entrada va a constituir otro de los parámetros que podemos ajustar en nuestro flujo para modificar como genera las descripciones.

Al proceso de diseñar y ajustar las instrucciones de entrada para que un LM genere una respuesta adecuada a tus intereses, se le denomina ingeniería del *prompt*. Esta disciplina es relativamente nueva y depende en gran medida de cada modelo utilizado y el caso de uso que se le esté dando. Cada modelo ha sido entrenado de distinta forma y con distintos datos de entrenamiento, por lo que no todos van a responder igual al mismo *prompt* de entrada. Además de ajustar el *prompt* a la entrada y el formato que el modelo particular mejor entienda, también hay que modificarlo teniendo en cuenta lo que queremos que nos responda y el formato de la respuesta esperada.

Por las razones planteadas, no existen estándares en ingeniería del *prompt*, pero sí prácticas comunes. Atendiendo a lo planteado en una guía general de *prompting*<sup>4</sup>, vamos estructurar nuestro input de la siguiente manera:

- Rol: Se indica el rol que debe cumplir el LM, en nuestro caso, será un generador de descripciones.
- Instrucciones: Se redactan una serie de tareas o instrucciones que el modelo tiene que seguir para generar la respuesta. Nuestro caso se puede plantear de varias maneras, pero se resume en que se debe basar en una serie de preguntas y respuestas sobre un imagen para generar una descripción.
- Contexto: Se proporciona al modelo con información externa o adicional para ayudarle a generar mejores respuestas. En esta parte del *prompt* proporcionamos al modelo con las preguntas y respuestas obtenidas en el paso anterior.
- Formato de la respuesta: En esta parte final, se indica al modelo que tipo o formato debe tener la salida, por ejemplo si queremos que nos devuelva una lista o un formato JSON. En nuestro caso simplemente especificaremos que se limite a devolver una descripción.

En la sección 5.2.1.1 se exponen los tres *prompts* utilizados.

---

<sup>4</sup><https://www.promptingguide.ai/es>

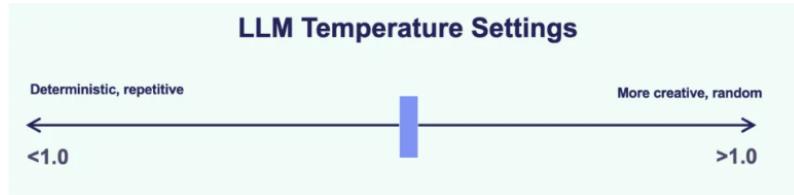


Figura 5.4: Valores del parámetro temperatura y su efecto en las predicciones de un modelo de lenguaje.

### Temperatura

Los grandes modelos de lenguaje tienen una serie de parámetros que permiten ajustar su comportamiento y rendimiento. Uno de ellos es la temperatura, que permite ajustar la “diversidad” o “creatividad” en las respuestas del modelo.

Una visión muy simplista del funcionamiento de estos modelos sería decir que predicen las palabras o *tokes* más probables que siguen a una secuencia de palabras o contexto dado (en la mayoría de casos una conversación) y van así componiendo la respuesta palabra a palabra. La temperatura es un parámetro que indica al modelo en qué medida escoger palabras o *tokes* menos probables para la secuencia dada.

En este contexto, una temperatura alta en la generación de texto provocará que éste sea más creativo o impredecible, aunque existe el riesgo de que sea incoherente si la temperatura es excesivamente elevada. Por otro lado, una temperatura baja conducirá a un texto más predecible o determinista, pero podría resultar repetitivo y poco informativo si el valor es demasiado bajo. Como se observa en la figura 5.4<sup>5</sup> se suele tomar como referencia el valor 1.

#### 5.2.1.1. Combinaciones de parámetros

Para ajustar los parámetros de la composición de la descripción hemos experimentado con diferentes valores de los parámetros configurables. Para los candidatos más viables, se ha hecho una comparación objetiva en la sección 5.3 tomando varias métricas que miden la calidad de la descripción generada.

### Número de preguntas recuperadas

El número de preguntas recuperadas se ha fijado en 8. Como se detalla en la sección 4, los parámetros fijados para el *clustering* de las preguntas candidatas a ser recuperadas genera típicamente entre 2 y 6 agrupaciones en la mayoría casos. Optar por un número mucho mayor que 6 podría resultar en la recuperación de un exceso de preguntas redundantes semánticamente, lo que podría generar mucho ruido para *Stablelm-zephyr-3b*. Sin embargo, no siempre tomar más de una pregunta de una agrupación resulta en tener dos preguntas muy similares semánticamente y en el peor de los casos se tendría información importante por duplicado. Además, es importante destacar que las agrupaciones de preguntas se repiten en orden de tamaño, es decir que cuando se tenga que repetir agrupación primero se hará con las más numerosas y, por tanto más relevantes para la imagen. Por esta razón, se ha tomado el valor de 8, valor para el que aunque exista riesgo de redundancia, el

<sup>5</sup><https://www.iguazio.com>

ruido añadido es mínimo, y las preguntas son generalmente variadas.

### **Prompt de entrada**

En cuanto al *prompt* de entrada se ha experimentado con varios valores siguiendo la estructura planteada en la sección anterior. En última instancia se escogieron los siguientes candidatos para hacer la evaluación objetiva:

#### *Candidato 1*

Write a caption of an image. These are questions that had been made about the image and their answers:

{pairs}

Limit the caption to the information available through the questions and answers. Just provide the caption.

#### *Candidato 2*

Act as an image caption generator.

Read pairs of questions and answer.

Generate a short description based on the pairs information.

Be brief limiting yourself to the information you have.

Here are the question-answer pairs:

{pairs}

Just answer with the generated caption

#### *Candidato 3*

Your task is to adapt a set of question and answers about an image to a caption format.

Limit yourself to the information provided in the pairs.

Here are the question-answer pairs:

{pairs}

Just provide the final caption.

Dónde el código de colores seguido para diferenciar las partes esenciales del *prompt* es el siguiente:

- Rol
- Instrucciones
- Contexto
- Formato de salida

y {pairs} hace referencia al conjunto de preguntas y respuestas siguiendo el formato del siguiente ejemplo (preguntas obtenidas para la imagen de la figura 5.5):

Question: Do the children ski with poles? Answer: yes

Question: Is this person moving or standing still? Answer: moving



Figura 5.5: Imagen de referencia para ejemplificar las descripciones generadas con las distintas configuraciones.

Question: What is on the ground? Answer: snow

Question: How many people are in this scene? Answer: 1

Question: Is this person wearing a helmet? Answer: yes

Question: What sport is shown? Answer: skiing

Question: What is covering the ground? Answer: snow

Question: How many girls in the picture? Answer: 1

El candidato 1 tiene los elementos básicos de un *prompt* pero no sigue el esquema en el mismo orden, a pesar de ello es un candidato prometedor por los resultados que ha dado en pruebas superficiales. El candidato 2 sigue el orden establecido en un esquema típico de un *prompt*. Tiene desarrollada en las instrucciones la cadena de pensamiento que el modelo debe seguir, técnica que se ha demostrado que puede ayudar a que el modelo produzca resultados esperados [68]. Los dos primeros candidatos se centran en tratar al modelo como un “descriptor de imágenes” o un “generador de descripciones”. El candidato 3 sigue el esquema en el mismo orden que

	Temperatura = 0.8	Temperatura = 1
<i>Prompt = Candidato 1</i>	Configuración 1	Configuración 2
<i>Prompt = Candidato 2</i>	Configuración 3	Configuración 4
<i>Prompt = Candidato 3</i>	Configuración 5	Configuración 6

Tabla 5.3: Resumen del valor de los parámetros que caracteriza cada una de las configuraciones evaluadas.

el segundo candidato pero trata el modelo como un adaptador de texto a texto y se le pide transformar una lista de preguntas y respuestas en una descripción o *caption*.

### Temperatura

En cuanto a la temperatura, uno de los valores que se va a utilizar para la evaluación va ser 1, ya que es el valor referencia para este parámetro. Se ha experimentado con valores alrededor de 1 y hemos encontrado que cuando la temperatura es mayor el modelo de lenguaje tiende a inventar información y añadirla a la descripción generada. Nuestro interés no radica en que el modelo sea “creativo”, sino en que se limite a la información que tiene y le de una estructura de descripción. Por lo tanto, optamos por valores inferiores a 1. Sin embargo, un valor demasiado pequeño provoca que el modelo sea demasiado conciso, limitándose a listar las respuestas en vez de construir una descripción completa. Por lo tanto hemos elegido 0.8 como otro de los valores para hacer la evaluación.

Las combinaciones de parámetros que se van a evaluar en la siguiente sección son por tanto las que se exponen en la tabla 5.3, donde en todos los casos el número de preguntas recuperadas es 8 y se omite por claridad.

Así, tomando estas configuraciones para las preguntas del ejemplo anterior, se obtienen los siguientes *captions*<sup>6</sup> para la imagen 5.5.

- **Configuración 1**

*A single girl, wearing a helmet, is seen skiing with ski poles on a snow-covered ground, moving steadily.*

- **Configuración 2**

*A single girl, wearing a helmet and carrying ski poles, is in motion on a snowy slope, embodying the sport of skiing.*

- **Configuración 3**

*The young girl is skiing with poles in the snowy scene, moving at an unknown pace with a helmet on her head. There is only one person in the picture, and it is representing one of the many scenes showcasing skiing sports. The ground is fully covered in snow, creating a serene and picturesque atmosphere.*

- **Configuración 4**

---

<sup>6</sup>En ocasiones se genera texto justificando el *caption* generado, repitiendo las preguntas y respuestas, o explicando decisiones. Este texto ajeno al *caption* se omite por claridad.

*Children are skiing with poles in a scene with a moving person on the ground covered in snow, indicative of skiing. Only one person is present in the image, and they are wearing a helmet and participating in skiing. The ground is primarily composed of snow.*

■ **Configuración 5**

*A single girl, wearing a helmet, is seen skiing on the snow-covered ground with ski poles in her hands, indicating that she is engaging in the sport of skiing.*

■ **Configuración 6**

*A single girl, wearing a helmet, can be seen moving through the snow in this skiing scene.*

### 5.3. Evaluación de la descripción

Determinar la calidad de un modelo que tiene una salida en lenguaje natural (Natural Language Generator, NLG) es una tarea complicada [9, 52]. Lo más común cuando se quiere evaluar un modelo que genera texto es llevar a cabo una evaluación subjetiva con expertos. Normalmente, estos expertos juzgan el texto en términos de gramática y contenido, valorando la corrección sintáctica y la relevancia de lo que se expresa. En ocasiones también se puntúan otros aspectos como la fluidez del texto o la creatividad. Para evaluar descripciones de imágenes podemos seguir las mismas técnicas y aplicar lo que se usa con NLG generales, presentando también la imagen al evaluador. Normalmente, las evaluaciones de descripciones de imágenes con humanos se guían con preguntas que siguen la escala Likert<sup>7</sup>. Las siguientes son algunas que se han usado con datasets y poblaciones de distintos tamaños.

- La descripción describe la imagen con precisión [29, 25, 40, 26, 13]
- La descripción es gramaticalmente correcta [40, 26, 71, 13]
- La descripción no contiene información incorrecta [40]
- La descripción es relevante dada la imagen [29, 71]
- La descripción está construida de forma creativa [29]
- La descripción podría ser la que daría una persona [40]

Otra aproximación a la evaluación de descripciones es usar medidas definidas que no dependen de evaluaciones subjetivas llevadas a cabo por personas, como por ejemplo BLEU [44], ROUGE [33], Meteor [4] o CIDEr [63]. Estas dependen de las coincidencias entre palabras pero no capturan el valor semántico de la frase. Por ese motivo cabe añadir la opción de comparar las descripciones/*captions* mediante *embeddings*, utilizando por ejemplo el modelo *GTE-small* explicado en la sección 4.2.1. Esta medida de comparación, propuesta en [53] para capturar significado, depende en gran medida del modelo utilizado y la información semántica que éste capture. No obstante, es un método de comparación que permite comparar las descripciones/*captions* generadas en términos semánticos, coincidan o no las palabras usadas para expresar las ideas y por ello debe ser tenido en cuenta.

Nosotros no nos restringiremos a una sola evaluación, sino que en vista de la falta de consenso en la mejor forma de evaluar una descripción optaremos por diferentes enfoques. Por un lado usaremos las evaluaciones sin usuarios para refinar parámetros de nuestro flujo de generación de descripciones. De este modo sacaremos partido de la velocidad y bajo coste que ofrece esta alternativa para evaluar diferentes configuraciones. Por otro lado, usaremos la evaluación con usuarios una vez tengamos una versión ajustada. De esta forma, verificaremos si la configuración elegida obtiene buenos resultados a los ojos de evaluadores humanos.

---

<sup>7</sup>Esto es, se presenta una afirmación y el evaluador indica su grado de acuerdo o desacuerdo con ella de entre las siguientes 5 opciones: totalmente de acuerdo, parcialmente de acuerdo, neutral, parcialmente en desacuerdo o totalmente en desacuerdo

### 5.3.1. Medidas objetivas

Para la evaluación con métricas objetivas se han elegido 50 imágenes aleatorias del dataset de validación de COCO 2017<sup>8</sup>. Cada imagen del dataset tiene asociadas 5 *captions* de referencia con las que comparar nuestra descripción generada para evaluar el rendimiento de nuestro modelo.

En las siguientes subsecciones se explican las métricas utilizadas y los resultados de la evaluación para las distintas combinaciones de parámetros configurables elegidas. Se ha utilizado un repositorio de Github<sup>9</sup> para evaluar nuestro modelo, excepto para la comparación por *embeddings*. Estos resultados nos permitirán elegir una configuración final con la que realizar la evaluación con usuarios.

#### 5.3.1.1. BLEU

Este método compara una frase candidata con varias frases de referencia y cuenta cuántas palabras o grupos de palabras (n-gramas) coinciden entre ellas. La idea intuitiva es que una palabra de referencia debe considerarse “agotada” después de identificar una palabra candidata que coincida.

En BLEU se calcula lo que en [44] denominan *modified n-gram precision*:

$$\text{modified } n\text{-gram precision} = \sum_{n\_gram} \frac{\max(\text{count}_{ref}(n\_gram))}{\min(\text{count}_{ref}(n\_gram), \text{count}_{candidate}(n\_gram))} \quad (5.1)$$

donde:

- $\max(\text{count}_{ref}(n\_gram))$  calcula el máximo del número de ocurrencias del n-grama en una referencia.
- $\text{count}_{candidate}(n\_gram)$  calcula el número de ocurrencias del n-grama en el candidato.
- $\sum_{n\_gram}$  sumatorio que recorre todos los n-gramas posibles.

Con esta fórmula, se obtiene un valor de 1 si hay una coincidencia exacta entre los textos, y un valor de 0 si no hay n-gramas en común.

A modo de ejemplo, a continuación se listan los valores de las métricas Bleu para la descripción generada por la configuración 3 de la imagen 5.5.

- Bleu 1: 0.23453
- Bleu 2: 0.24762
- Bleu 3: 0.22142
- Bleu 4: 0.18667

---

<sup>8</sup>Tomando imágenes de una versión diferente del dataset evitamos que haya intersección con nuestra base de casos de 5000 imágenes.

<sup>9</sup><https://github.com/tylin/coco-caption>

Estos valores se obtienen comparando con las 5 descripciones de referencia, que se listan a continuación.

*A person walking through the snow near a fence.*

*A person cross country skiing on a path.*

*A person riding skis across a snow covered road.*

*A cross country skier travels a worn snow path.*

*A man cross country skiing in the country.*

### 5.3.1.2. ROUGE-L

Esta métrica compara la frase candidata con varios textos de referencia mediante el cálculo de la longitud de la subsecuencia común más larga (LCS), entendiendo como subsecuencia conjuntos de palabras que no tienen por que ser seguidas en el texto.

En ROUGE-L [34], el valor de la métrica es calculado mediante las siguientes fórmulas:

$$R_{lcs} = \frac{LCS(X, Y)}{m} \quad (5.2)$$

$$P_{lcs} = \frac{LCS(X, Y)}{n} \quad (5.3)$$

$$ROUGE - L = \frac{(1 + \beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2P_{lcs}} \quad (5.4)$$

donde  $X$  e  $Y$  son los textos a comparar,  $n$  es el tamaño del texto candidato,  $m$  es el tamaño del texto de referencia y  $\beta = \frac{P_{lcs}}{R_{lcs}}$ . Con estos valores, se consigue un valor de 1 si  $X = Y$  y un valor de 0 si  $LCS(X, Y) = 0$ .

Teniendo en cuenta el ejemplo mencionado en la sección anterior. La descripción generada por la configuración 3 ha obtenido un valor de 0.28938 en la métrica ROUGE\_L.

### 5.3.1.3. CIDEr

Este método se centra en medir la similitud entre la descripción candidata y un conjunto de descripciones de referencia, añadiendo un cierto peso semántico al proceso. Para conseguir añadirle ese peso semántico que no tienen las dos métricas explicadas anteriormente, primero filtra todas las palabras para quedarse con su lexema o raíz, así palabras como “fishing”, “fishes” y “fished” serían reducidas a “fish”. Después, se calcula la similitud del coseno entre las representaciones TF-IDF (Frecuencia de Término - Frecuencia Inversa de Documento) de las descripciones candidatas y de referencia [64].

Sea  $I_i$  la imagen cuya descripción es la candidata,  $c_i$  la descripción candidata y sea  $S = \{s_{i1}, \dots, s_{im}\}$  el conjunto de descripciones de referencia. Denominamos  $h_k(s_{ij})$  al número de veces que se repite un n-grama  $w_k$  de la descripción candidata en una de referencia  $s_{ij}$ . Se calcula el valor TF-IDF de una descripción  $s_{ij}$  mediante la fórmula:

$$g_k(s_{ij}) = \frac{h_k(s_{ij})}{\sum_{w_l \in \Omega} h_l(s_{ij})} \log \frac{|I|}{\sum_{I_p \in I} \min(1, \sum_q h_k(s_{pq}))} \quad (5.5)$$

donde  $\Omega$  es el vocabulario de todos los n-gramas de la descripción candidata. Intuitivamente, el primer término de la ecuación calcula el TF de cada n-grama  $w_k$  y el segundo término calcula el IDF. Con esto, se consigue que la parte de TF aumenta el peso a los n-gramas que aparecen más frecuentemente en las descripciones de referencia, mientras que la parte IDF reduce el peso de aquellas que aparecen más comúnmente en todas las imágenes del dataset.

Con este valor, se computa la similitud del coseno entre la sentencia candidata y las de referencia:

$$CIDEr_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{\mathbf{g}^n(c_i) \mathbf{g}^n(s_{ij})}{\|\mathbf{g}^n(c_i)\| \|\mathbf{g}^n(s_{ij})\|} \quad (5.6)$$

donde  $\mathbf{g}^n(c_i)$  es un vector formado por los valores  $g_k(c_i)$  correspondiente a todos los n-gramas de longitud  $n$  y  $\|\cdot\|$  es la norma euclídea del vector. Análogo para  $s_{ij}$ .

Finalmente, se combinan los resultados para distintas longitudes de n-gramas tal que:

$$CIDEr(c_i, S_i) = \sum_{n=1}^N \frac{1}{N} CIDEr_n(c_i, S_i) \quad (5.7)$$

con  $N=4$ .

En el análisis final veremos que la métrica CIDEr no ha otorgado resultados interpretables para la mayoría de casos. En el ejemplo planteado en las secciones anteriores, la métrica CIDEr ha adquirido un valor de 0.

#### 5.3.1.4. Distancia del coseno con *embeddings*

La distancia del coseno es una métrica que ya se ha explicado anteriormente para comparación de *embeddings* de imágenes. Aunque el origen de los vectores o *embeddings* es diferente (en este caso se generan *embeddings* a partir de texto) el formato de estos es el mismo, por tanto la distancia se calcula igual que en la sección 4.1.3.3.

A diferencia de las métricas anteriores, esta métrica nos va ayudar a medir la similitud semántica entre nuestra *caption* generada y las cinco que tenemos como referencia para cada imagen. Como valor final de la comparación para cada imagen, se ha elegido el valor máximo de las similitudes (tomando como el mínimo de las distancias del coseno) de nuestra *caption* generada con cada una de las *captions* de referencia.

Para el ejemplo mencionado en las secciones anteriores, la descripción generada ha obtenido una similitud del coseno de 0.880.

#### 5.3.2. Resultados de la evaluación con medidas objetivas

En esta sección se muestran los resultados de evaluar las 6 configuraciones planteadas en la tabla 5.3 en el conjunto de test de 50 imágenes. Las tablas 5.6, 5.7, 5.8 y 5.9 muestran histogramas de frecuencia de valores para cada métrica respectivamente. Para facilitar la interpretación de la gráfica se han representado funciones de densidad de probabilidad de cada configuración.

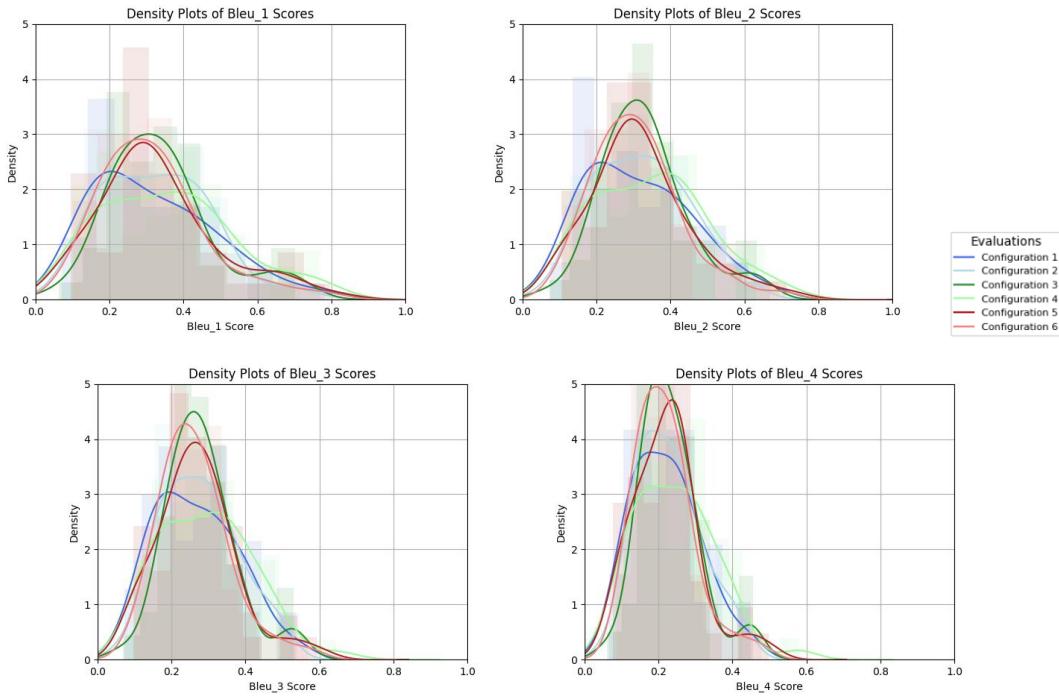


Figura 5.6: Distribución de probabilidades de las métricas Bleu\_n en el conjunto de test para cada configuración.

En la figura 5.6 no se aprecian diferencias especialmente significativas, sin embargo sí que se puede apreciar una mayor densidad de valores mayores a 0.4 en las métricas BLEU para las configuraciones 1, 2 y 4. Por otro lado, podemos observar que la configuración 3 es la que menor densidad de valores por debajo de 0.1 tiene y, aunque no haya dado tantos resultados por encima de 0.4 como las configuraciones mencionadas, parece consistente en media. Lo que es evidente es que las configuraciones 5 y 6 correspondientes al *prompt candidato 3* son las que peor resultados han dado en la métricas BLEU. Cabe recordar que este *prompt* estaba enfocado de distinta manera en cuanto al rol que se le da al modelo, lo que podría estar generando un comportamiento peor.

En la figura 5.7 de la métrica ROUGE\_L tampoco podemos apreciar grandes diferencias. Las tendencias seguidas para esta métrica son similares a las mostradas en la figura 5.6, donde la configuración 3 destaca por la baja densidad de valores bajos.

La métrica CIDEr no ha mostrado resultados buenos para ninguna configuración.

La métrica *cosine\_similarity* mostrada en la figura 5.9 se debe interpretar de manera opuesta a las anteriores. Una distancia menor implica una similitud mayor entre las descripciones generadas y alguna de las descripciones de referencia. En este caso es la configuración 4 la que muestra un mejor rendimiento. Observamos también que las configuraciones pares suelen tener una densidad centrada ligeramente más a la izquierda que sus análogas con el mismo *prompt*. Esto implica que las configuraciones con temperatura igual a 1 (configuraciones pares) rinden mejor que sus análogas con temperatura igual a 0.8 (configuraciones impares) para esta

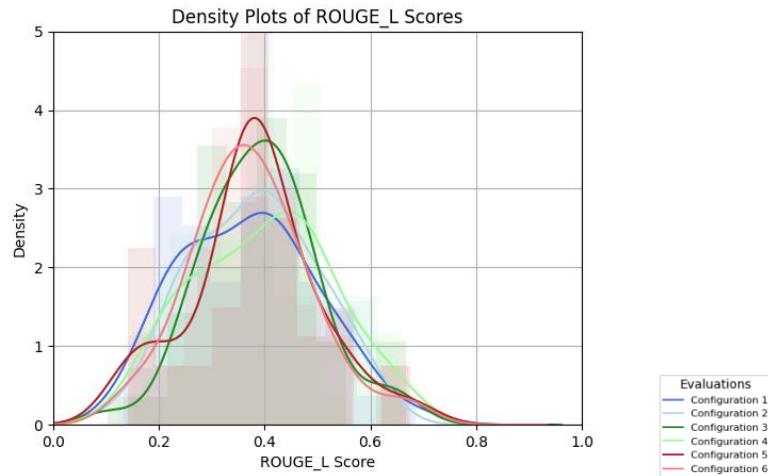


Figura 5.7: Distribución de probabilidades de la métrica ROUGE\_L en el conjunto de test para cada configuración.

	Bleu 1	Bleu 2	Bleu 3	Bleu 4	ROUGE L	CIDEr	cos sim
<b>config. 1</b>	0.240	0.245	0.214	0.181	0.370	0.000	0.872
<b>config. 2</b>	0.258	0.262	0.230	0.192	0.381	0.000	0.874
<b>config. 3</b>	0.277	0.278	0.240	0.200	0.391	0.000	0.865
<b>config. 4</b>	0.271	0.272	0.235	0.196	0.398	0.001	0.863
<b>config. 5</b>	0.257	0.257	0.223	0.187	0.379	0.000	0.866
<b>config. 6</b>	0.257	0.260	0.227	0.190	0.373	0.000	0.872

Tabla 5.4: Valores medios de cada configuración para cada métrica.

métrica que es puramente semántica. Esto se puede deber a que una configuración del modelo más “creativa”, que ya observamos que añadía información no presente en las preguntas y respuestas, añada términos basándose en el contexto dado en el *prompt* y esto provoque que se encuentren más términos semánticamente similares.

En la tabla 5.4 se puede analizar las medias de las métricas para las distintas configuraciones. En esta tabla se aprecia claramente como las configuraciones 3 y 4 correspondientes al *prompt candidato 2* obtienen mejores resultados.

Se observa una tendencia clara durante todo el análisis hacia el mejor rendimiento de las configuraciones 3 y 4, correspondientes al *prompt candidato 2*. La configuración 4 ha obtenido una mayor densidad de valores altos, mientras que la configuración 3 ha registrado menos resultados negativos. La temperatura puede haber jugado un papel importante en estos resultados. Una temperatura alta podría haber generado descripciones más creativas, añadiendo información que en ocasiones fuera correcta (lo que explica la mayor probabilidad de resultados positivos), pero en otras ocasiones pudo ser incorrecta (lo que explica también una mayor probabilidad de resultados negativos). En cambio, con una temperatura baja, donde el modelo tiende a ceñirse más a la información disponible, se generaron con menos frecuencia resultados negativos y se mantuvo una consistencia, aunque sin sobresalir en la mayoría de los casos. Nosotros buscamos una configuración que sea consistente y

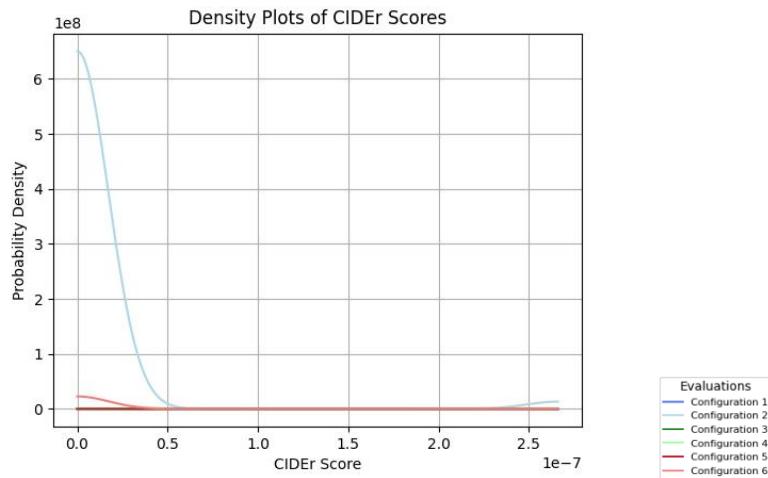


Figura 5.8: Distribución de probabilidades de la métrica CIDEr en el conjunto de test para cada configuración.

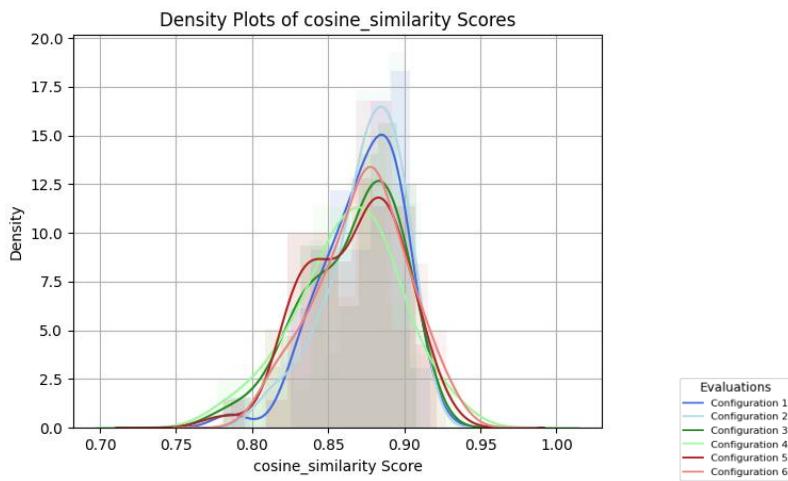


Figura 5.9: Distribución de probabilidades de la métrica *cosine\_similarity* en el conjunto de test para cada configuración.

que proporcione resultados decentes en la mayoría de los casos, razón por la cual hemos elegido la configuración 3.

### 5.3.3. Evaluación con usuarios

La evaluación con usuarios trae consigo ventajas e inconvenientes, pero creemos que teniendo en cuenta sus limitaciones puede ser útil para dar perspectiva diferente a la evaluación global del modelo. Por un lado es cierto que introduce un factor subjetivo de forma inevitable, pero no es menos cierto que hay determinados aspectos en el texto escrito que son inherentemente subjetivos. Por ejemplo, no se puede decidir de manera objetiva y numérica la calidad de una obra literaria, pues para cada persona, incluso si es experta en la materia, merecerá una calificación diferente. Es por

este motivo que la evaluación con usuarios no pretende dar puntuaciones exactas, sino más bien dar una idea de cómo funciona el modelo en diferentes aspectos.

Para llevar a cabo esta evaluación usaremos las 6 afirmaciones mencionadas al principio del capítulo, pues como allí se expone son las más repetidas en la literatura. Por la misma razón usaremos la escala Likert para ofrecer al usuario 5 grados de acuerdo o desacuerdo con cada afirmación sobre cada descripción. Recordemos que dichas afirmaciones son las siguientes, donde entre paréntesis se sintetiza la magnitud que pretenden medir.

- La descripción describe la imagen con precisión (*Precisión*)
- La descripción es gramaticalmente correcta (*Corrección gramatical*)
- La descripción no contiene información incorrecta (*Corrección semántica*)
- La descripción es relevante dada la imagen (*Relevancia*)
- La descripción está construida de forma creativa (*Creatividad*)
- La descripción podría ser la que daría una persona (*Naturalidad*)

En cuanto a las imágenes, podríamos usar las mismas 20 que se usaron para evaluar las preguntas, pues los usuarios que realizaron tal evaluación ya están familiarizados con ellas y les resultará menos pesado realizar la evaluación. Sin embargo, esa evaluación se utilizó para elegir la medida de similitud con la que se eligieron las preguntas a partir de las cuales se compuso la descripción de la imagen que ahora se quiere evaluar. Por tanto, para evitar cualquier clase de sesgo en los resultados obtenidos usaremos imágenes nuevas, obtenidas de la misma fuente que las anteriores<sup>10</sup> y tratando de mantener la misma diversidad en cuanto a tipo de imágenes (animales, comida, deporte, habitaciones...). En la anterior evaluación con usuarios decidimos separar las 20 imágenes en dos cuestionarios para hacer el tiempo de evaluación más corto con el objetivo de incentivar la participación. En este caso también queremos mantener lo más bajo posible el tiempo necesario para responder el cuestionario, pero lo haremos de otra forma. Esta vez usaremos solo 10 imágenes en un solo cuestionario<sup>11</sup> pues consideramos que para este caso no es tan importante el número de imágenes. Además, elegimos un formato desplegable para mejorar la visualización del cuestionario desde dispositivos móviles, que es desde donde la mayoría de usuarios responderán. La figura 5.10 muestra cómo aparece una de las imágenes con su correspondiente descripción en el test llevado a cabo.

#### 5.3.4. Resultados de la evaluación con usuarios

El cuestionario se difundió a personas sin conocimiento específico, pues no se requiere el dominio de ningún área en particular para responderlo. Se obtuvieron 53 respuestas durante un período de 5 días y en todas ellas se llenó la totalidad del cuestionario.

---

<sup>10</sup>Pixaby (<https://pixabay.com/>)

<sup>11</sup><https://forms.gle/deS1h3BL3met3yKK7>

**Imagen 5/10**

Indique su grado de acuerdo o desacuerdo con las afirmaciones siguientes sobre la descripción de esta imagen.

**Descripción:** A cow with a tag on its ear is relaxing near mountains with trees in the background.

Es gramaticalmente correcta \*

Elige

Parcialmente de acuerdo

Ni de acuerdo ni en desacuerdo

Parcialmente en desacuerdo

Totalmente en desacuerdo

Está construida con creatividad \*

Totalmente de acuerdo

Podría ser la que un humano daría \*

Ni de acuerdo ni en desacuerdo

Atrás Siguiente Página 6 de 11 Borrar formulario

Figura 5.10: Ejemplo real de la evaluación de descripciones con usuarios.

En la figura 5.13 se pueden ver los resultados que obtuvo de media cada imagen en cada uno de los seis aspectos que se evaluaban (precisión, corrección gramatical, corrección semántica, relevancia y naturalidad). Además, se añade un gráfico con los resultados superpuestos de todas las imágenes y otro con la media de todas ellas, en cada una las magnitudes medidas. Para promediar, se han asignado los siguientes valores numéricos a las respuestas tipo Likert.

- Totalmente de acuerdo → 4
- Parcialmente de acuerdo → 3
- Ni de acuerdo ni en desacuerdo → 2
- Parcialmente en desacuerdo → 1
- Totalmente en desacuerdo → 0

Así, un valor mayor en cierta magnitud representa mayor calidad en la descripción en términos de esa magnitud. Con esto en cuenta, procedemos a interpretar los datos obtenidos, apoyándonos también en los datos numéricos recogidos en la tabla 5.5.

En un primer vistazo se observa bastante disparidad de resultados de unas imágenes a otras, con casos como *Imagen 5* que rozan el 4 (Totalmente de acuerdo) y casos como *Imagen 10* que en corrección semántica apenas llega al 1 (Parcialmente en desacuerdo). Asimismo, en el último gráfico se observa que hay magnitudes en las que se obtienen mejores resultados y otras en las que el desempeño es más pobre.

	<b>Media</b>	<b>Desviación</b>	<b>Mínimo</b>	<b>Máximo</b>
<i>Precisión</i>	2.094	1.098	0.868	3.830
<i>Corrección gramatical</i>	2.781	1.108	1.981	3.755
<i>Corrección semántica</i>	1.815	1.297	0.604	3.679
<i>Relevancia</i>	2.447	1.127	1.377	3.755
<i>Creatividad</i>	2.492	1.122	2.132	2.962
<i>Naturalidad</i>	1.745	1.260	0.962	3.547

Tabla 5.5: Resultados generales de la evaluación con usuarios. La desviación es la media de las desviaciones en las respuestas de cada imagen y el máximo (resp. mínimo) es el mayor (resp. menor) resultado medio obtenido por una imagen.

En particular, se obtienen malos resultados (por debajo de 1) si hablamos de naturalidad y corrección semántica, mientras que la magnitud mejor valorada ha sido la corrección gramatical. Hagamos una análisis de los resultados medios obtenidos para cada una de las magnitudes.

#### ■ Precisión

Aunque podemos decir que en media la precisión no es del todo mala, hay casos en los que ni siquiera se alcanza el 1 y desde luego no podemos decir que el modelo sea preciso. Posiblemente esto se deba a las preguntas recuperadas de las imágenes similares en las que se puede nombrar ciertos ítems inexistentes en la imagen objetivo. Así, el sistema de generación de texto puede usar estos ítems para construir la descripción proporcionando datos erróneos que perjudiquen la precisión.

#### ■ Corrección gramatical

Los buenos resultados en términos de corrección gramatical (2.78 en media) son mérito del sistema generativo de texto. En este caso, la obtención de una descripción gramaticalmente correcta es responsabilidad suya y parece que lleva a cabo un buen trabajo.

#### ■ Corrección semántica

Los malos resultados de la corrección semántica, que en algunos casos bajan del 1, pueden tener un origen diverso. Es muy probable que éstos vengan de datos erróneos de partida, anteriores a la construcción de la descripción por el modelo texto-texto. Sin embargo, pueden tener causas diversas. Por un lado, pueden deberse a respuestas incorrectas del sistema VQA. En este caso la responsabilidad del fallo recae en este sistema. Por otro lado, pueden venir de preguntas que no eran aplicables a la imagen objetivo y que han derivado en respuestas sin sentido tras el modelo VQA.

#### ■ Relevancia

En cuanto a la relevancia de las descripciones generadas los resultados no son demasiado buenos ni demasiado malos, situándose generalmente cerca de la nota intermedia en la escala usada. Esto puede deberse a que si bien la

	Precision	Cor_gram	Cor_sem	Relevancia	Creatividad	Naturalidad
Precision	1.000000	0.488980	0.540058	0.651848	0.344019	0.582099
Cor_gram	0.488980	1.000000	0.377514	0.385992	0.438953	0.411095
Cor_sem	0.540058	0.377514	1.000000	0.408135	0.243525	0.362002
Relevancia	0.651848	0.385992	0.408135	1.000000	0.382861	0.562405
Creatividad	0.344019	0.438953	0.243525	0.382861	1.000000	0.337503
Naturalidad	0.582099	0.411095	0.362002	0.562405	0.337503	1.000000

Figura 5.11: Matriz de correlación entre las variables medidas.

mayoría de las descripciones tienen en cuenta o nombran ciertos ítems muy significativos o protagonistas de las imágenes, en algunas ocasiones dan información incorrecta sobre ellos, lo que hace que la evaluación baje. Asimismo, las preguntas sobre elementos no presentes en la imagen derivan muchas veces información negativa. Esto es, información sobre cosas que no son, como por ejemplo “no hay surfistas”. Esta información aunque correcta, es poco relevante y penaliza mucho en este aspecto.

#### ■ Creatividad

Los resultados obtenidos en la evaluación de la creatividad del modelo son notablemente altos, estando generalmente por encima de la nota intermedia. Además, son bastante homogéneos, situándose entre 2 y 3 en todas las imágenes. Analizando las causas en base a las descripciones generales es claro que estos resultados vienen determinados por los ítems inexistentes en la imagen objetivo que las descripciones nombran y por las oraciones “negativas” que descartan la existencia de algún ítem en la imagen. Por ejemplo, en una imagen donde no haya un sombrero, la descripción puede contener la frase “no hay un sombrero”. Esto se debe a que en las imágenes similares de las que se recuperan las preguntas puede existir este ítem habiendo alguna pregunta sobre ello, que el sistema recupera y el generador de texto introduce en la descripción. Los usuarios pueden haber interpretado esta información poco relevante como creativa, aumentando el valor obtenido para esta variable.

#### ■ Naturalidad

Como ya hemos dicho, en general no se obtienen buenos resultados para la naturalidad. Una posible explicación es que la naturalidad puede verse afectada por información incoherente. Si la descripción no es coherente a causa de información de partida que no concuerda, no va a resultar natural. Además, en ocasiones el modelo genera texto que si bien puede ser correcto, tiene una redacción poco fluida. En definitiva, creemos que los malos resultados en cuanto a naturalidad se deben en parte al modelo generador de texto y en parte a información incoherente proveniente de los pasos anteriores.

Observando los resultados obtenidos en la evaluación y tras hablar con algunos de los usuarios que llenaron la encuesta, tenemos la sensación de que en ocasiones no

se entendió bien lo que se debía evaluar con cada magnitud. En particular, creemos que la corrección semántica opacaba el resto de aspectos evaluados y que cuando había información incorrecta los usuarios penalizaban también aspectos ajenos a este hecho, como la relevancia o la corrección gramatical.

Para arrojar un poco de luz sobre si los usuarios comprendieron bien su tarea, empecemos observando la desviación obtenida en cada magnitud. Un valor alto de ésta significaría una alta disparidad en las respuestas, fruto de una interpretación diferente por distintos usuarios, según nuestra hipótesis. En la tabla 5.5 vemos que la desviación supera la unidad en todas las magnitudes, alcanzando el máximo valor de 1.297 para la corrección semántica. Teniendo en cuenta que los valores que manejamos están entre 0 y 4, y que una distribución uniforme en nuestro problema (mismo número de respuestas en cada una de las 5 opciones) tendría una desviación de 1.41, creemos que son unos valores elevados. Estas desviaciones anormalmente altas, sobre todo en la corrección semántica y en naturalidad nos hacen pensar que en efecto los usuarios no comprendieron del todo lo que debían evaluar en cada caso.

Creemos que es muy probable que incluso hayan mezclado conceptos, siendo influidos por magnitudes ajena a la que evaluaban en cada caso. Si observamos en la figura 5.11 la matriz de correlación entre las variables, basada en las respuestas de los usuarios, vemos que hay variables muy correlacionadas. Por supuesto, es posible que la alta correlación no se deba a una confusión de las variables por los usuarios, sin embargo, hay que señalar que algunos valores de correlación son bastante altos. En particular, la relevancia, naturalidad y precisión destacan por estar íntimamente correlacionadas. Además, la precisión y la corrección semántica también están bastante correlacionadas. Estos resultados cuadran con nuestra hipótesis de la influencia entre variables, pues además creemos que éstas variables pueden tener efecto unas sobre otras, como se explica cuando se habla de ellas individualmente.

En conclusión, creemos que es posible que los usuarios no hayan terminado de comprender los aspectos de la descripción que se trataban de medir con cada pregunta, dándose una influencia entre las diferentes magnitudes a medir. El análisis de los resultados lo respalda pero no es suficiente para asegurarlo. En cualquier caso, creemos que este efecto podría haberse mitigado explicando más extensamente en el cuestionario lo que se quería evaluar en cada momento.

### **Explicabilidad**

Durante el trabajo hemos asegurado que nuestra propuesta aportaba un grado de explicabilidad del que otros modelos carecen. Veámoslo en la práctica para tener una idea de por qué se ha obtenido una descripción tan mala para la *Imagen 10* (ver figura 5.12) según los usuarios. Al mismo tiempo ilustraremos diferentes casos que se pueden dar en nuestra arquitectura y que suponen la causa de información incorrecta.

La descripción generada por nuestra arquitectura para la *Imagen 10* (ver figura 5.12) es “*A brown dog is walking on a sidewalk while a man in black luggage carries a black suitcase. It is sunny and the woman has a backpack on her back. The dog is not traveling.*” y se genera a partir de las siguientes preguntas y respuestas.

1. *Question: What color is the dog?*



Figura 5.12: Imagen utilizada en la evaluación de las descripciones con usuarios.

*Answer: brown*

2. *Question: What is the man doing?*

*Answer: walking*

3. *Question: What color is his luggage?*

*Answer: black*

4. *Question: Is it sunny?*

*Answer: yes*

5. *Question: What does the woman have on her back?*

*Answer: backpack*

6. *Question: Where is the dog?*

*Answer: sidewalk*

7. *Question: Is the man traveling?*

*Answer: no*

8. *Question: What color is the suitcase on the right?*

*Answer: black*

Como se puede apreciar, pueden darse diferentes casos con las preguntas. En primer lugar, hay preguntas como la 1, que son apropiadas y cuya respuesta proporciona información valiosa. En segundo lugar, también hay preguntas como la 8 que preguntan por algo inexistente. Este tipo de preguntas son especialmente problemáticas porque dan a entender que existe tal cosa, en este caso una maleta. Esta información errónea se filtra a la descripción reduciendo sustancialmente su calidad. Existe otro tercer caso, en el que la pregunta es apropiada pero se da una respuesta incorrecta. En este caso la información errónea también se filtra a la descripción final.

pero la responsabilidad recae en el sistema VQA. Un ejemplo de esto es la pregunta 5, apropiada pero erróneamente contestada.

Por último, cabe destacar otra fuente de errores. Es posible que el sistema generador de texto encargado de redactar la descripción no transfiera correctamente la información recogida en las preguntas y respuestas al texto generado. Esto ocurre con la pregunta 7, que pregunta si el hombre está viajando. Podemos ver en la descripción generada que esta información se plasma en la última frase, en la que ahora es el perro el que viaja.

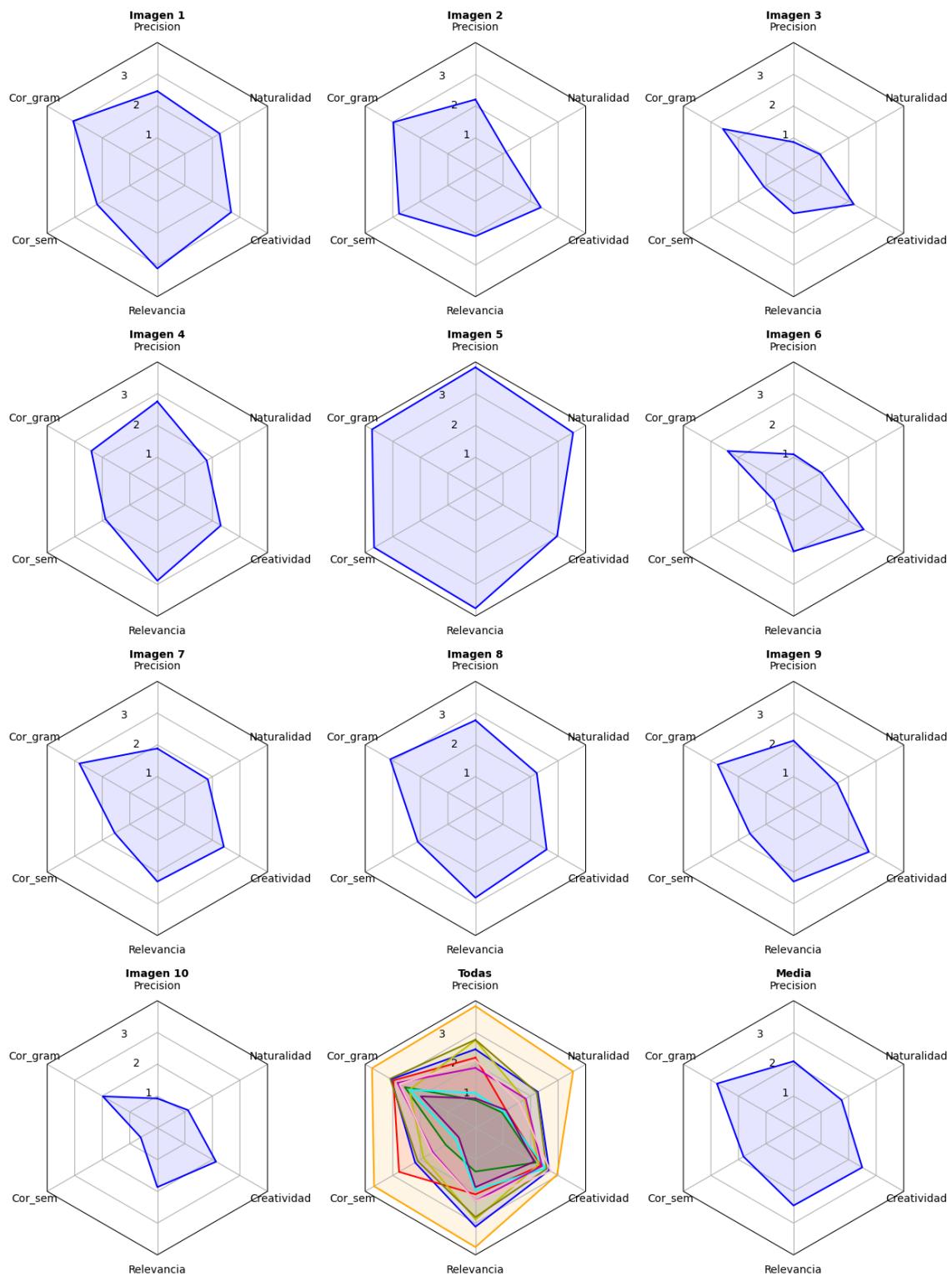


Figura 5.13: Resultados de la evaluación con usuarios de las descripciones.

# Capítulo 6

## Conclusiones y trabajo futuro

Mantenerse al día con los avances en inteligencia artificial, y en concreto, con los modelos de texto e imágenes, se ha vuelto una tarea desafiante. Empresas líderes en tecnología lanzan mensualmente modelos multimodales capaces de mantener una conversación, describir una imagen y analizar documentos, entre muchas otras habilidades. Las capacidades de estos modelos sobrepasan con creces el rendimiento que ofrece nuestra aproximación siendo capaces de analizar hasta el último detalle de imágenes, incluso leyendo texto en ellas. En la figura 6.1 podemos apreciar una comparativa entre las descripciones breves generadas por nuestro modelo y las generadas utilizando un modelo abierto al público, Gemini. En ella vemos como Gemini tiene un rendimiento decente en ambas imágenes, mientras que nuestro modelo puede llegar a estar a la altura en algunos casos (segundo ejemplo) pero en otros (primer ejemplo) no cumple del todo las expectativas.

A pesar de estos rápidos avances, creemos que nuestro modelo CBR para descripción de imágenes ofrece una perspectiva diferente con la que abordar el problema de generación de texto a partir de imágenes. En él planteamos el tratamiento de imágenes y texto asociado a ellas como experiencia en la que buscar casos similares cuando queremos tratar una imagen nueva. Todo ello se hace sin la necesidad de recurrir a modelos multimodales, lo que puede suponer una ventaja al no ser necesarios tantos recursos tanto de software como de hardware. Además, el sistema de flujo de información permite recuperar resultados intermedios que aportan un buen grado de explicabilidad, de la que estos modelos multimodales carecen. Aunque tenga limitaciones, basándonos en la evaluación final del trabajo, creemos que la aproximación planteada tiene un rendimiento decente y fácilmente mejorable.

### 6.1. Cumplimiento de objetivos

En cuanto al cumplimiento de los objetivos propuestos al inicio y durante el transcurso del trabajo, consideramos que todos se han cumplido de manera satisfactoria. Además, se ha respetado el plan de trabajo establecido cumpliendo dichos objetivos en un tiempo razonable para desarrollar todas las ideas previstas.

En una fase inicial exploramos las alternativas disponibles en modelos de detección de objetos y sistemas VQA. Probamos distintos enfoques de los modelos

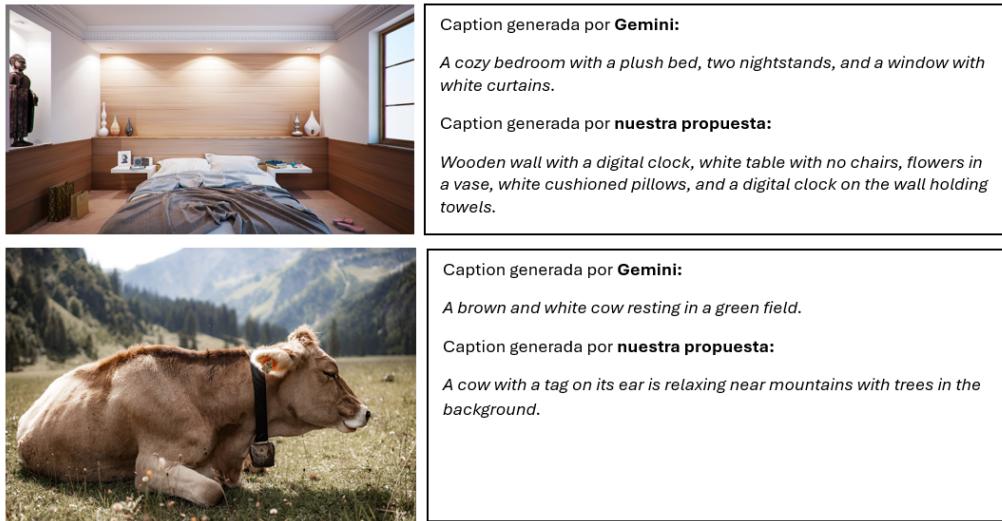


Figura 6.1: Comparación entre Gemini y nuestro modelo propuesto en la generación de una descripción breve o caption para dos imágenes del conjunto de evaluación con usuarios. El *prompt* utilizado para generar el caption en Gemini ha sido: “Generate a single caption for the following image {image}” (donde {image} representa el lugar en el que se ha insertado la imagen).

mencionados y se investigaron los datasets más utilizados para entrenar este tipo de modelos, eligiendo finalmente COCO. Cumplimos así con los objetivos  $O_1$  y  $O_2$  establecidos en la introducción. Los objetivos  $O_3$ ,  $O_4$  y  $O_5$  completan la primera parte de nuestra propuesta, la recuperación de preguntas relevantes para la imagen mediante un sistema de razonamiento basado en casos. Se completaron satisfactoriamente, realizando una comparación y evaluación más exhaustiva de 4 medidas de similitud concretas. Posteriormente se completó el trabajo implementando una manera de generar descripciones utilizando un modelo VQA y otro de generación de texto ya existentes, alcanzando el objetivo  $O_6$ . Finalmente se evaluó la aproximación completa logrando el último objetivo planteado al comienzo del trabajo, el objetivo  $O_7$ . Bien es cierto que algunos objetivos han debido ser adaptados para el correcto cumplimiento del resto. Por ejemplo, para el objetivo  $O_6$  la idea previa era implementar un modelo VQA propio, pero finalmente se descartó la idea por falta de tiempo. Respecto al objetivo  $O_7$  creemos que la evaluación presenta dudas en cuanto a la validez de los resultados, por lo que quedaría como trabajo futuro revisar y mejorar el método de evaluación.

### 6.1.1. Artículo ICCBR

Queremos destacar también que el trabajo realizado en cuanto a la recuperación de preguntas sirvió para la redacción de un artículo que fue publicado en la conferencia internacional ICCBR 2024 (*International Conference on Case-Based Reasoning*). En este artículo se plantea la posibilidad de utilizar un arquitectura CBR para la generación de descripciones utilizando preguntas previamente formuladas a imágenes similares, aprovechando así la experiencia almacenada en la case de casos. Para

la publicación del artículo no se implementa el modelo completo, sino que se estudia la manera más adecuada de recuperar preguntas para el propósito mencionado mediante un análisis de medidas de similitud entre imágenes.

## 6.2. Limitaciones y trabajo futuro

La propuesta planteada en el trabajo cuenta con algunas limitaciones que tienen que ver tanto con la cantidad de datos disponible como con algunos de los procesos elegidos.

La base de casos es uno de los elementos más limitantes y determinantes de un modelo CBR. Con el objetivo de agilizar los procesos de pruebas y la obtención de resultados, decidimos crear una base de casos de 5000 imágenes. Esto limita significativamente la capacidad de obtener casos muy similares a la imagen objetivo. En este sentido, una base de casos más amplia podría dar mejores resultados. No solo el tamaño de la base de casos, sino también el contenido y la temática de esta, podrían mejorar el rendimiento de este proceso. Aunque hayamos creado una base de datos con casos genéricos, este proceso resulta realmente útil cuando se tienen casos que atienden a experiencias concretas. Por ejemplo, un usuario fotógrafo profesional probablemente no haga las mismas preguntas sobre una fotografía o imagen que un usuario con dificultades visuales. Tener una base de casos atendiendo a experiencias concretas podría hacer este proceso realmente útil a la hora de generar descripciones enfocadas en un ámbito específico.

Otra de las limitaciones actuales, y área de mejora del modelo CBR planteado, es la forma en la que se utilizan las preguntas recuperadas de los casos más relevantes o similares. Se ha realizado un análisis exhaustivo para obtener la medida de similitud que recuperarse los casos más relevantes y, posteriormente, se ha hecho un análisis iterativo de como obtener las soluciones o preguntas finales de las candidatas asociadas a estos casos. Sin embargo, existe margen de mejora en la utilización de estas preguntas y sus respuestas para la generación de la descripción. El caso más llamativo es la inclusión de información negativa (debido a respuestas negativas a preguntas) en la descripción, destacando en la descripción final la utilización de frases como “no hay x” o “no se puede encontrar x” (como se puede observar en el primer ejemplo de la figura 6.1), lo cual suele ser irrelevante para la construcción de descripciones breves. Este efecto se puede ver mitigado por el tamaño de la base de casos, pero también se puede ver mejorado mejorando la selección final de preguntas relevantes o realizando una ingeniería de *prompt* más exhaustiva. Asimismo, se puede mejorar el caso en el que las preguntas recuperadas no son apropiadas para la imagen, normalmente porque preguntan sobre elementos que no están presentes en ella. Este caso es más dañino que el anterior pues en lugar de añadir información irrelevante pero correcta, añade información incorrecta. Si no hay en la imagen cierto objeto y se pregunta por su color, el sistema VQA dará una respuesta y el generador de texto asumirá que sí está presente dicho objeto. En estos casos, que son muy perjudiciales para la calidad de la descripción final, se proponen dos soluciones. Por un lado, se pueden descartar las preguntas para las que el sistema VQA da una respuesta con baja confianza, entendiendo que son preguntas que no tienen

sentido para la imagen. Por otro lado, se plantea la posibilidad de adaptar la pregunta cambiando palabras de forma que tenga sentido para la imagen. Por ejemplo, con una imagen objetivo de un perro, si alguna de las preguntas asociadas es sobre un gato, sustituir la palabra “gato” por la palabra “perro”. Esta alternativa supone un reto en cuanto a la identificación de los elementos clave de las preguntas y la decisión de si deben ser sustituidos o no en función de la imagen objetivo, sin embargo, creemos que puede ofrecer un salto de calidad considerable en nuestro modelo.

Teniendo en cuenta los objetivos propuestos, los resultados obtenidos y las limitaciones de nuestro modelo, existe un potencial de mejora amplio en nuestro concepto. En primer lugar, parte del trabajo futuro se basa en seguir realizando pruebas sobre el modelo actual mejorando distintos aspectos. Por ejemplo, un aumento del tamaño de la base de casos, como ya se ha comentado, podría mejorar ciertos resultados y abrir nuevas líneas de investigación sobre posibles nuevos problemas o imprecisiones del sistema. Además, todavía se pueden explorar más medidas de similitud entre imágenes (y combinaciones de ellas) y más sistemas de extracción de características para generar los *embeddings* de las imágenes a comparar. Por otra parte, se podrían realizar pruebas para subconjuntos de la base de casos estudiando los resultados obtenidos para focalizar mejor para qué tipo de imágenes es realmente útil nuestra aproximación. En segundo lugar, el desarrollo de un modelo VQA propio con un previo estudio y ajuste de los parámetros de las redes neuronales convolucionales usadas podría proporcionar mejoras a nuestro sistema, que usa modelos VQA ya existentes. De este modo y basándonos en tareas más específicas (como escoger solo imágenes de una temática en concreto), la tarea de ajuste de las redes que generan las respuestas aportaría más valor que un modelo genérico como el que usamos.

Por último y teniendo en cuenta que es una tarea más ambiciosa, se podría desarrollar un sistema propio de generación de descripciones a partir de un *prompt* de entrada. Creemos que proporcionaría una mejora considerable al ajustar este modelo propio a las necesidades de este problema específico y evitaría usar modelos que en cierto modo son más generalistas y tienden a no ajustarse a lo que se pide exclusivamente. Por ejemplo, para ciertas imágenes y aún siendo el *prompt* de entrada muy claro y sin ambigüedades, los resultados no siempre se ajustan del todo a lo que se demanda en el input.

En definitiva, creemos que nuestro modelo puede ser una buena aproximación para la generación de descripciones de imágenes ofreciendo un enfoque diferente al del estado de la cuestión actual.

# Apéndice A

## Contribuciones Personales

En este apéndice se detallan las contribuciones y aportaciones de cada miembro del grupo al trabajo. En multitud de ocasiones las aportaciones solapan pues lo común durante la realización de este trabajo de fin de grado ha sido la colaboración. Así, ha sido la norma la participación de varios de nosotros en cada una de las partes, tanto en el desarrollo del código como en la redacción de la memoria. No obstante, en ocasiones ha sido un miembro en concreto el que se ha encargado de alguna parte en particular. A continuación cada miembro detalla sus contribuciones al trabajo.

### A.1. Adrián Pérez Peinador

Quiero empezar destacando que el trabajo realizado con Rubén y Adrián ha sido agradable y equitativo. En ocasiones se ha repartido el trabajo para organizar los avances pero por lo general hemos avanzado por iniciativa propia, compartiendo después con nuestros compañeros el trabajo realizado. Asimismo, quiero destacar que si bien cada parte del código o de la memoria la escribe una persona, los otros dos miembros la han revisado y frecuentemente han sugerido o añadido detalles. Dicho esto, expondré los aspectos del trabajo en los que he tenido un papel principal, siguiendo el orden en el que aparecen en esta memoria, si bien en la práctica no se haya seguido ese orden para el desarrollo.

Empezando por el resumen, en su redacción tuve un papel secundario, redactando algunas partes mas no el grueso. En cuanto a la introducción, me encargué de la redacción de los apartados 1.2 y 1.4, que versan sobre los objetivos del trabajo y la organización interna que hemos seguido. Asimismo, me encargué de la confección de esquemas que ilustrasen nuestro modelo, entre ellos los que se muestran en las figuras 1.1 y 1.2. Siguiendo con el estado de la cuestión, los tres integrantes del grupo nos encargamos por igual de la búsqueda de información sobre el tema, la investigación de las técnicas más punteras y la elaboración de pequeños resúmenes que sirvieran al resto de compañeros para tener la información esencial sobre los campos que habíamos investigado cada uno. Es cierto que en un principio nos centramos, y yo en particular, en la detección de objetos y finalmente no se le ha dado tanto uso a esta rama, pero considero que el conocimiento adquirido ha ayudado en algunos aspectos del trabajo. En cuanto al estudio del dataset finalmente utilizado, el grueso

del trabajo lo llevaron a cabo mis compañeros pero puedo destacar la confección de la figura 3.1 y aportaciones menores en la redacción.

El capítulo 4, donde se trata la recuperación de las preguntas siguiendo un esquema CBR, es uno de los más importantes y extensos del trabajo, así que lo expondré por secciones.

La sección 4.1, sobre las formas de medir la similitud entre imágenes, es una de las más importantes del trabajo. En ella se estudian tres aproximaciones principales, y cada uno de nosotros se centró sobre todo en una de ellas, aunque aportara también al resto. En mi caso, tomé la responsabilidad de la similitud basada en objetos detectados, encargándome del desarrollo del código y la redacción de la memoria. En particular, definí e implementé 6 medidas de similitud entre imágenes basadas en los objetos que contienen, sus *bounding boxes* y sus superclases, e implementé el código necesario para obtenerlas dadas dos imágenes. Además, desarrollé el código común a todas las similitudes que obtiene las  $k$  imágenes más similares de entre la base de casos. En cuanto a la memoria, redacté la subsección 4.1.2, en la que se trata la similitud basada en objetos y confeccioné el esquema de la figura 4.1.

La sección 4.3 trata sobre la evaluación con usuarios realizada para comparar las diferentes medidas de similitud. En cuanto a la redacción en la memoria de esta sección, me encargué tanto de la introducción de la sección como de la subsección 4.3.2, donde se explica cómo se llevó a cabo el experimento. Asimismo, tuve también importantes aportaciones en la confección de los cuestionarios utilizados.

El capítulo 5, que trata la segunda parte del modelo, es el otro capítulo principal del trabajo. En él, se expone la construcción de la descripción a partir de las preguntas recuperadas mediante el sistema CBR. Este capítulo se divide en tres secciones, tras una introducción que corrió a mi cuenta, así como la confección del esquema presentado en la figura 5.1.

La primera sección, la 5.1, sobre el sistema VQA usado para responder a las preguntas recuperadas, la investigación sobre este tipo de modelos y la elaboración de la tabla 5.1, corrieron también a mi cuenta.

La sección 5.2 trata sobre la composición de la descripción desde las preguntas recuperadas y sus respuestas. En la redacción de esta sección tuve un papel secundario, escribiendo algunas partes, modificando algunas cosas de mis compañeros y elaborando la tabla 5.3.

La última sección de este capítulo, la 5.3, trata sobre la evaluación de la descripción generada por nuestro modelo. En su introducción, redactada por mí, se explica la dificultad de evaluar la salida un sistema generador de texto y se proponen varias alternativas para hacerlo. Para escribir esta introducción llevé a cabo una investigación sobre la forma en la que se evalúan las descripciones y las redacciones en general, las medidas objetivas que existen y las mejores formas de evaluación subjetiva con usuarios. Las subsecciones 5.3.1 y 5.3.2 se centran en las medidas objetivas y eligen una configuración de parámetros para nuestro modelo en base a los resultados obtenidos con ellas. La siguiente subsección (5.3.3), escrita por mí, explica la evaluación llevada a cabo con usuarios. Para esta evaluación, confeccioné un cuestionario con imágenes no usadas hasta el momento que me encargué de seleccionar, siguiendo las pautas extraídas de la investigación que llevé a cabo. Los resultados de esta evaluación se tratan en la subsección 5.3.4, escrita también por mí. Para el aná-

lisis de los resultados, que corrió a mi cargo, elaboré tablas y figuras que ilustrasen los aspectos que se querían tratar. La tabla 5.5 y las figuras 5.11 y 5.13 ilustran los resultados obtenidos y sirven de apoyo a lo que desarrollo en esta subsección acerca de los resultados, su validez y las posibles razones de su obtención.

El último capítulo del trabajo es el 6, “Conclusiones y trabajo futuro”. En él, se exponen las conclusiones sobre el trabajo realizado, el cumplimiento de los objetivos propuestos al principio del mismo y las líneas de trabajo que se abren en un futuro. La redacción de este capítulo corrió a cargo de mis compañeros en gran medida, pero hice contribuciones escribiendo algunas partes. En cuanto a la redacción de la versión traducida de éste, sí que corrió a mi cuenta, escribiendo la mayor parte de la traducción.

Por último, fui el encargado de configurar la plantilla de L<sup>A</sup>T<sub>E</sub>Xutilizada para la elaboración de esta memoria, cambiando lo necesario para adaptarla a nuestras necesidades y preferencias de presentación.

## A.2. Adrián Sanjuán Espejo

Para comenzar la exposición de mis contribuciones personales quiero resaltar que el trabajo realizado junto con mis compañeros Rubén Gómez Blanco y Adrián Pérez Peinador ha sido equitativo desde el principio. Desde el inicio del trabajo, hemos ajustando la distribución de tareas atendiendo a la disponibilidad de cada uno, lo que ha podido provocar desequilibrios en secciones específicas, pero ha asegurado una colaboración global balanceada.

Antes de comenzar con mis aportaciones al contenido del trabajo, me gustaría resaltar mi rol en relación a la organización del grupo y a la comunicación con los tutores. Durante el transcurso del trabajo he sido el encargado de compartir los avances bisemanales (y semanales) con los tutores y, junto a Adrián Pérez Peinador, de compartir dudas y discutir futuras líneas de trabajo con ellos.

En cuanto al desarrollo de código y posterior redacción de la memoria, pocas han sido las partes o secciones en las que no hayamos contribuido todos los miembros del equipo de algún modo. Sin embargo, a continuación detallaré las fases y elementos del trabajo en los que he tenido un papel más destacado. Voy a exponer los aportes ordenados por secciones, siguiendo la estructura del documento principal, aunque en la práctica no se haya seguido ese orden a la hora de desarrollar el trabajo.

Respecto a los preámbulos, he sido el encargado de redactar el abstract tanto en su versión en español como en inglés. Aunque no haya tenido un papel principal en la redacción de la introducción, si que me he encargado principalmente de su traducción, abarcando las secciones 1.1, 1.3 y 1.4, además de revisar las traducciones realizadas por mis compañeros.

Al comenzar el trabajo, participé de forma equitativa a mis compañeros en la investigación del estado de la cuestión. En mi caso, además de realizar de una revisión de la literatura, también realicé un curso de detección de objetos que finalmente no se incluyó en el trabajo. Asimismo, exploré entornos para llevar a cabo demostraciones de modelos de generación de textos y VQA. El objetivo de estas demostraciones fue mostrar a los demás integrantes del grupo las capacidades de estos modelos.

En lo que respecta al capítulo 3, referente a la base de casos, hicimos una investigación conjunta de los datasets más apropiados para nuestro trabajo. Mi contribución principal en esta fase consistió en, una vez elegido el dataset de COCO, plantear distintas maneras de construir nuestra base de casos y la programación que permitió generarla. Debido a mi papel en la creación de la base de casos, también contribuí en gran medida al análisis de esta, creando la figura 3.2 entre otras que no se llegaron a añadir a la memoria. En este sentido, aunque sin un rol principal, he tenido pequeñas contribuciones en la redacción de este capítulo.

Las contribuciones de los capítulos 4 y 5 las voy a exponer por secciones, ya que son los capítulos con más contenido del trabajo. En la sección 4.1 cada miembro se encargó de estudiar un tipo de medida de similitud entre imágenes, y por tanto, de programar lo necesario y posteriormente redactarlo. En mi caso, hice un estudio de las similitudes a nivel de píxel en el capítulo 4.1.1. También tuve una breve contribución en la sección 4.1.3 evaluando el rendimiento de los 4 modelos de *embeddings* sopesados en esta sección, y redactando las conclusiones de esta evaluación. A parte de desarrollar el código correspondiente a la similitud por píxeles, también participé

en el desarrollo de partes del código comunes, aunque con un papel más secundario.

La sección 4.2, en la que se estudia la manera de recuperar las preguntas finales de las candidatas asociadas a imágenes similares, ha sido tanto redactada como programada principalmente por mí. A pesar de ello, como en el resto del trabajo, tanto el proceso inicial de discusión de ideas y desarrollo de pruebas, como la fase final de revisión y corrección de la redacción, fueron realizados de manera colaborativa por todos los miembros.

La siguiente sección (4.3) habla del experimento realizado para la redacción del artículo que va a ser publicado en el congreso ICCBR 2024, en el que también se incluyó información del capítulo 3 y el resto del capítulo 4. En esta sección me encargué de redactar las hipótesis, y en conjunto con mis compañeros, de crear el cuestionario para la realización del estudio. En el análisis de los resultados finales de la subsección 4.3.3 me ocupé del análisis exploratorio con Python, y contribuí en gran medida a la redacción final de esta subsección, incluyendo la creación de las tablas 4.3 y 4.4 para exponer los resultados.

En relación al capítulo 5, y más específicamente la sección 5.1, mi papel principal fue el de programar todo el flujo completo de generación de la descripción. Esta tarea incluyó la integración de la obtención de imágenes similares y la selección de preguntas asociadas, planteada en el capítulo 4, con un sistema VQA capaz de responder las preguntas para la imagen objetivo, y la posterior utilización de un modelo de generación de texto para crear la descripción a partir de los pares pregunta-respuesta obtenidos.

Dado mi rol principal en la programación del flujo completo de obtención de la descripción, fui el encargado de explicar en la sección 5.2.1 los parámetros configurables de nuestro sistema.

Referente a la evaluación final de los resultados generados por nuestra propuesta, sección 5.3, me encargué de llevar a cabo el desarrollo en código de la evaluación semiautomática con métricas objetivas (explicadas en la subsección 5.3.1), cuyo análisis se utilizaría para elegir una configuración final con la que realizar la evaluación con usuarios. Fui también el encargado de realizar el análisis de los resultados ofrecidos por estas métricas, incluyendo la creación de las figuras 5.6, 5.7, 5.8 y 5.9 y la tabla 5.4 presentes en esta subsección.

Finalmente, redacté gran parte del capítulo 6, que abarca las conclusiones y trabajo futuro de nuestra propuesta. Sin embargo, dado que se trata del cierre del proyecto, todos tuvimos algún papel en su desarrollo. En cuanto a su traducción, desempeñé un papel secundario y me encargué principalmente de revisar el contenido.

En los párrafos anteriores se han plasmado mis contribuciones principales y algunas secundarias que he considerado relevantes. A pesar de ello, también me gustaría mencionar que en todas las fases y secciones a las que no he hecho referencia, he tenido, como mínimo, un papel de revisor. Como ya he mencionado, aunque el proceso de programación y/o redacción de cada una de las partes del trabajo haya sido llevado a cabo por una persona en particular, todos los miembros del grupo participamos equitativamente en la discusión de ideas y desarrollo de pruebas para testear estas.

### A.3. Rubén Gómez Blanco

En esta sección redactaré detalladamente mi contribución individual al trabajo de fin de grado. No sin antes destacar que desde el comienzo del desarrollo del trabajo de fin de grado, la organización y repartición de tareas se realizó de forma equitativa, adaptándonos a la vida personal y laboral de cada integrante del equipo. Cabe destacar que a pesar de posibles desbalances de participación de unos u otros en ciertas partes del trabajo debidas a distintas situaciones, creemos que a nivel general hay un buen equilibrio de trabajo entre los integrantes del equipo.

Además, me gustaría destacar que las siguientes contribuciones redactadas serán solo al trabajo final, no se tendrá en cuenta todo el desarrollo descartado y no usado en el trabajo final, tanto en lo referente a la redacción como al código. Si bien esta parte no es visible para

A continuación se detallan todas las contribuciones en las que he tenido o bien un papel principal, o uno más secundario, tanto en la redacción de la memoria del trabajo como en el desarrollo del código utilizado, ordenadas por secciones siguiendo la estructura del documento principal.

En cuanto a los preámbulos, he tenido un papel secundario en la redacción del resumen aportando la redacción del primer párrafo o párrafo introductorio. En cuanto al capítulo 1, he sido el encargado de la redacción correspondiente a la sección 1.1 referente a la motivación del trabajo, incluyendo la búsqueda de información que se encuentra en esta. Siguiendo con la 1.2, he contribuido en una segunda iteración en la corrección de la redacción de esta sección reestructurando la parte de objetivos para que fueran más específicos, siendo en la primera versión más generalistas. He sido también el responsable de la redacción de la sección 1.3 referente al plan de trabajo, redactando a nivel general el proceso a seguir durante el desarrollo de este. Finalmente, redacté también la sección 1.5 referente a la estructura del documento final.

En cuanto al capítulo 2, los tres miembros del grupo hemos participado de forma equitativa en el estudio e investigación del estado del arte de las distintas subtareas en las que basamos nuestro proyecto. En mi caso, fui el responsable de la puesta en común de todo el estudio realizado siendo el encargado de la redacción de esta sección.

En lo referente al capítulo 3 donde se expone la base de casos que usaremos en nuestro trabajo, he sido el encargado del análisis de los datasets de COCO de los que más adelante extrajimos nuestra base de casos, así como de la redacción de este análisis en el capítulo. Además, he contribuido en la parte del código referente al análisis de los distintos datasets que forman nuestra base de casos final, por ejemplo y entre otras aportaciones, creando las figuras 3.3 y 3.4.

Analizaré por secciones mis contribuciones personales del capítulo 4 al ser este capítulo más extenso y estar más subdividido en carga de trabajo. En cuanto a la sección 4.1, cada integrante del equipo se encargó de estudiar un tipo de medida de similitud entre imágenes, y por tanto, de la programación necesaria y posterior redacción. En mi caso me encargué del estudio de la similitud por *embeddings* en la sección 4.1.3, investigando acerca de los distintos tipos de redes convolucionales capaces de realizar una correcta extracción de características de imágenes y reco-

pilando distintas medidas de similitud entre estos *embeddings*. Además, todos los integrantes del equipo participamos en el desarrollo de código común referente a la comparación entre imágenes, adaptando el estudio y el código que cada uno desarrolló para abstraer en código común todo lo necesario. En cuanto a la sección 4.2, mantuve un rol secundario de revisión de redacción y puesta en común de ideas iniciales. En la sección 4.3 referente a la evaluación con usuarios de la recuperación de preguntas sobre las imágenes para comparar las diferentes medidas de similitud, fui el encargado de buena parte de la redacción y análisis de resultados por tipos de imágenes en la sección 4.3.3. Además, en cuanto a la confección del cuestionario, si bien todos los miembros del grupo tuvimos aportaciones, yo me encargué de la clasificación de imágenes en los tipos seleccionados y recuperación de preguntas para su posterior inclusión a los dos cuestionarios desarrollados.

En el capítulo 5 referente a la construcción de las descripciones de las imágenes, fui el responsable de la redacción e investigación respectiva a la introducción de la sección 5.2. Además, fui el responsable del estudio y búsqueda de información, así como de la redacción de las medidas objetivas para evaluar las descripciones generadas, en la sección 5.3.1. Por otra parte, en lo referente a la evaluación con usuarios de las descripciones generadas, contribuí al análisis de los resultados con la redacción de cierta parte de la sección 5.3.4, aunque del grueso de ella se encargó Adrián Pérez Peinador.

Finalmente, en el capítulo 6 en la que se exponen las conclusiones y el trabajo futuro de nuestro proyecto, fui responsable de buena parte de la redacción, aunque todos participamos en mayor o menor medida en la búsqueda de conclusiones y la reflexión acerca de las limitaciones de nuestra propuesta y posibles mejoras para un trabajo en el futuro.

En cuanto a la traducción de las partes de introducción y conclusiones, fui el encargado de las secciones 1.2 y 1.5.



# Introduction

## Motivation

With the arrival Artificial Intelligence (AI) in everyday life and its use in many professional fields such as medicine and education, new challenges arise when it comes to training and explaining the models that make our lives easier.

It is undeniable that users showed a significant interest and astonishment at the results achieved by models such as ChatGPT<sup>1</sup> or Dall-E<sup>2</sup> in generative tasks, a trend that started two years ago. Their performance and accessibility, even to users with little to no AI or Computer Science knowledge, has led to a widespread popularity and an exponential increase in their utilization. So much so that many of these generative AI tools have become an indispensable for millions of daily users around the globe. This has triggered a chain reaction in which an increasing number of companies are placing more trust in the future of these models. According to a report by Brainy Insights<sup>3</sup>, the income generated by generative AI services will reach 188 billion dollars by 2032. This growth is fuelled by the increased adoption of AI in industries and the companies' desire to leverage their data for decision-making. In this sense, research in this field has led to the development of very ambitious models which need a large amount of data for training for the correctness of their results and predictions. This leads to the development of newer versions of these models which are capable of executing new tasks with better attention to detail.

However, the training of these large models requires massive amounts of data which can be a problem when dealing with limited hardware and software resources. Additionally, the processing capability and energy required by these models makes them less accessible to small organizations and individuals. The resources needed raise significant environmental concerns too, due to the Carbon footprint associated with the large data centres needed to train and execute these models. For this reason, research on how to reduce the resources needed to train these models is also on the rise. A line of research to achieve this is to explore ways of reducing the amount of data needed for training [42].

Computer vision tasks are one of the most eye-catching branches of Artificial Intelligence. The combination of language and images is inherent to human beings, and it is very appealing to see AI get closer and closer to human performance in image and text generation based on specific instructions [15].

---

<sup>1</sup><https://chat.openai.com/>

<sup>2</sup><https://openai.com/dall-e-2/>

<sup>3</sup><https://www.thebrainyinsights.com/report/generative-ai-market-13297>

In this context, systems capable of dealing with these types of tasks emerge to help users in many different ways. The specific task of natural language image captioning, which is the one we devote our research to, can be helpful in many ways:

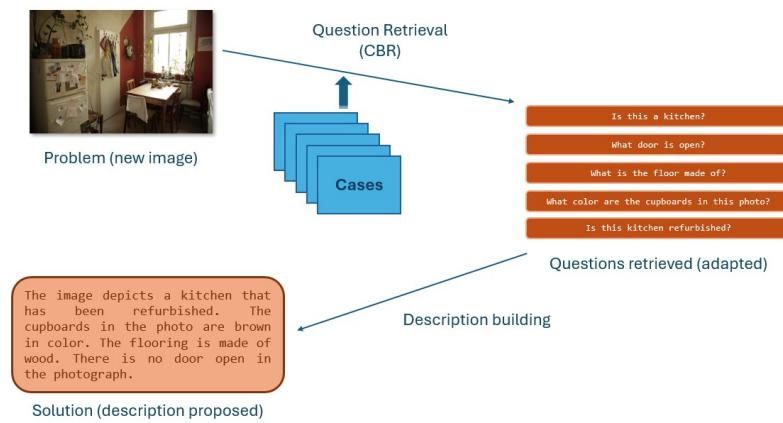
- Accessibility to visually impaired people: Thanks to this technology images can be translated to text, allowing visually impaired people to understand and experience firsthand and visually what surrounds them
- Search and categorization of images: Generated descriptions can help index images in databases, allowing a more efficient retrieval of relevant images.
- Medical applications: It can help professionals in the interpretation of medical images, such as radiographies or MRIs, providing detailed analysis.
- Content creation for marketing and social media: Image description generation can help marketing professionals and social media content creators to rapidly tag and caption images for their usage in blogs, social media and marketing campaigns.

However, most current description generation systems are large scale models trained with massive amount of data and resources. The task of reducing the resources needed, as it was anticipated in this section, is crucial for the development of this kind of models and encourages exploring other development alternatives.

In this work we explore areas as Visual Question Answering (VQA), which consists in being able to answer questions about images in natural language using AI techniques [2]. This idea in particular opens up opportunities for innovative applications that don't limit to question answering. An example of this is that VQA systems can be utilized to generate text based on the answers these models provide to generate descriptions. Our proposal is based on previous experience using the Case-based reasoning (CBR) paradigm. In this sense, we intend to be able to generate a coherent description of a given image by grounding the process in previously asked questions about similar images. Furthermore, if the case base used is built upon a specific experience (i.e. questions made by visually impaired people), the description generation can be directed towards that experience. This could be useful when building descriptions in a more technical or specific context.

Another aspect to be considered is explainability. It is becoming increasingly important to be able to explain how these models reach their outputs and conclusions, moving away from the idea of black-box models. However, deep learning systems are very limited in this aspect and explainability presents a big challenge.

With this Bachelor's thesis we propose an explainable description generator system which is based on a Case-based Reasoning architecture. The ultimate objective is to create a system that given an image, is able to make a description based on answers (provided by a small VQA model) to previously asked questions about similar images (CBR) (see figure 1.1). To be able to do that, we will try to avoid large multimodal models which are trained with large amount of data by giving our system previous experiences from our base case as context. On top of that, the intention is to achieve this goal being able to explain intermediate steps in order to contribute to the explainability of the system. This way we avoid a black-box system and try to



General schema of the description generation.

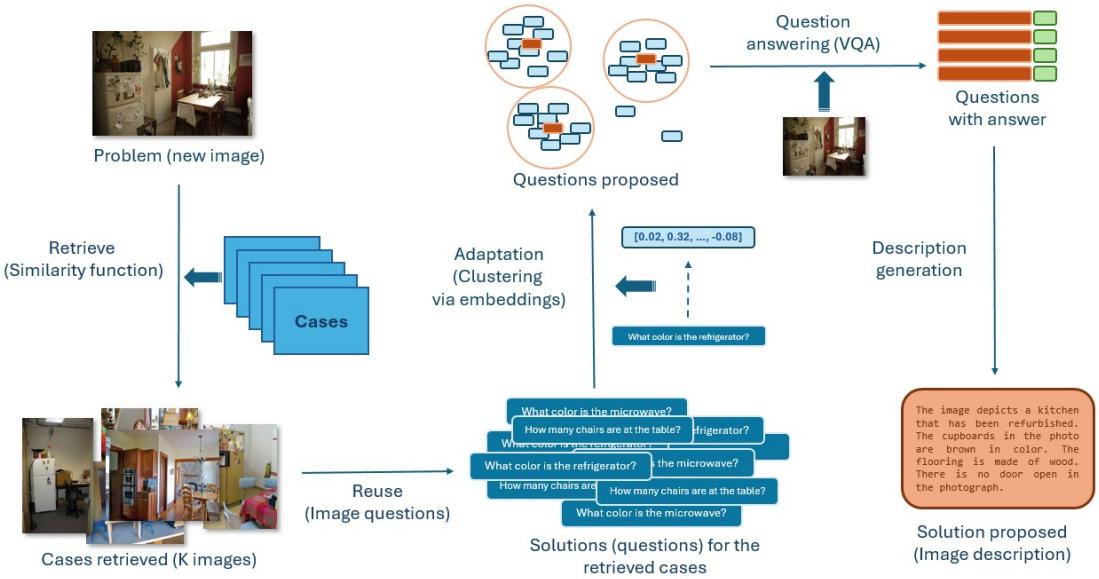
make it more accessible to people that don't have technical knowledge about these models.

## Objectives

In summary, the main objectives that we want to achieve during the realization of the project are:

- (O<sub>1</sub>) **Study the state of the art.** We will explore the state of the art of the main subtasks of our system; comparison of images, question recovery, VQA systems and generation of descriptions, in order to understand their strengths and difficulties that we will encounter during the development of the project.
- (O<sub>2</sub>) **Choose an appropriate case base.** We must find a properly labelled case base to perform VQA tasks. In addition, we seek a dataset that represents reality, both in the images and in the associated questions.
- (O<sub>3</sub>) **Development of a case-based reasoning system for obtaining questions.** Since our goal is to use questions asked to similar images to compose a description, we need to develop a case-based reasoning (CBR) [6] system capable of retrieving, reusing and adapting the questions asked to these images.
- (O<sub>4</sub>) **Development of different similarity measures between images.** For the retrieval of relevant questions to compose the description of the target image, we need to select the most similar images. Thus, we propose to investigate several similarity measures between images based on different parameters and characteristics of the image.
- (O<sub>5</sub>) **Similarity measures evaluation.** The objective is to make a user evaluation of the similarity measures in terms of question retrieval (of the most similar images) for a given image. After an analysis of the evaluation, the most appropriate one for our purpose will be chosen.
- (O<sub>6</sub>) **Construction of image descriptions.** Once we have obtained the most relevant questions for our image, we want to be able to answer them and generate a description of the image based on those answers.
- (O<sub>7</sub>) **Descriptions evaluation.** We propose to test our model by evaluating the descriptions generated by it, so that we obtain an indication of its performance.

Ultimately, our goal is to build a description generation model based on experience stored in a case base. The process we propose to achieve our goal is described in figure 1.2, it is built upon the idea that questions made about similar images will be similar between themselves. The model receives an image as input and it retrieves the  $k$  most similar images. Among the questions associated to these images, the most relevant are selected. These questions are used on the input image, and with the answers provided the system makes a natural language description about the input image.



Detailed schema of the process of description generation based on the CBR structure.

## Work plan

With the aim of achieving the mentioned objectives, we describe working plan to follow during the execution of this project.

First of all, an exhaustive literature review about the most important concepts of image similarity, VQA, text generation and description generation must be done. This is carried out while we develop the main ideas of our proposal. As we expand our knowledge in these areas, new objectives will emerge and will be taken into account till we reach the final idea of our proposal.

With the foundations of our work laid, more specific research about the state of the art will be carried out, investigating recent updates in the field, the most innovative scientific papers and the most widely used VQA and generative models.

Initially, and after choosing an appropriate dataset, we will focus on different methods to measure similarity between images. Three methods will be chosen, where each member of the project will make deep research on each of the methods. The three methods chosen will be object detection, pixel-level similarities and embedding-based similarities. Different metrics and tests will be used to evaluate these methods and choose the best one according to our interests.

In second place, based on the previous study about the state of the art in VQA, we need to explore models capable of answering the questions retrieved from similar images. This is an important step because we need to find a model with inputs and outputs in our desired format.

Lastly, we will delve into the world of generative AI by researching text generation models capable of providing a description based solely on input question-answer pairs. Consequently, we will compare diverse models, taking keeping in mind their size/performance ratio, in order to ultimately select a model capable of achieving

our objectives with the fewest parameters possible. Additionally, a final evaluation of our work will be conducted, rating the quality of the generated descriptions, to be able to draw conclusions about our proposal. With this evaluation, we will assess the viability of our concept and reflect on causes of errors and potential improvements with a view to proposing a possible future work.

## Internal organization

The project will be developed in an interactive environment, *JupyterHub*, shared among both the members of the group and the tutors, in order to track the progress of the project. This environment hosts all the code developed since the beginning of the project, including testing code, pieces that have been finally discarded and developments made as part of our learning process.

Additionally, final code will be uploaded in a Github repository<sup>4</sup>. This repository includes a *Readme* with the structure of the project and reproducibility instructions.

In terms of the management of documentation we will use two ways of coordination. On one hand, a shared Google Drive directory will be used to organize documents of interest, meeting notes and other relevant files such as questionaries. On the other hand, the Overleaf tool will be employed to draft the project's memory jointly in real time using L<sup>A</sup>T<sub>E</sub>Xcode.

Concerning the organization and monitoring of objective achievements, it was decided from the beginning to maintain biweekly online meetings. If at any point it is considered necessary due to holidays, increased workload or the need to address a particular issue, the meetings can be adjusted in terms of duration and frequency. The content of the meetings will consist in a review of the work done since the last checkpoint, discussion of future lines of work and clarification of doubts. Similarly, the use of the institutional email will be utilized for communication between students and tutors if any topic is of particular urgency.

## Document structure

The document has been organized in a way that it can be followed and understood by people with a certain knowledge base (without being experts) about the main deep learning techniques.

In chapter 2, the most current approaches in the tasks of VQA, natural language processing and generation of image descriptions are studied, describing their main characteristics.

In chapter 3 we present the final case base on which we will base the work, explaining its characteristics in detail.

In chapter 4 we detail the entire question recovery process, from the study of similarities between images, through the selection of the recovered questions, to the evaluation of said questions.

---

<sup>4</sup><https://github.com/TFG-UCM-VQA/VQA-TFG>

In chapter 5 we explore the process of generating the image description based on the recovered questions. First, we study the VQA model, whose task is to answer questions. Then, we examine text generation systems and their parameters to find the optimal configuration. Finally, we will evaluate the generated description.

In chapter 6 our conclusions are presented, and the accomplishment of our objectives is evaluated. Possible future work is also indicated, paying particular attention to obstacles that must be overcome before integrating a system like ours into practical use.



# Conclusions and future work

Keeping up to date with the progresses in artificial intelligence has become a challenging task, specially talking about text or image models. Top companies are launching new products monthly, multimodal models able to have conversations, describe images or analyse documents, between many others. The capabilities of these models outperform our proposal by far, being able to process images with such a detail that allows to even read text on them. The figure below shows a comparison between descriptions, one of them is given by our model while the other is provided by Gemini, an open generative model. It is observed that Gemini provides quite good captions for both images, while our model is capable of making the grade in some cases (second example) but underperform in others (first example).

Despite these quick progresses, we believe that our CBR based model offers a new perspective with which the image captioning problem can be addressed. We introduce the treatment of images and their associated questions as experience where we can look for similar cases when a new image is given. All of it is made avoiding the use of large multimodal models, what can mean an advantage in terms of software and hardware resources, as our model require a significantly lower amount of them. Furthermore, our information flow allows the retrieval of intermediate data which can be of significant use when explaining the obtained results, giving our model a degree of explainability of which multimodals lack. In spite of its limitations, and based on the final evaluation, we believe that our proposal has a sufficient performance, easily improvable.

## Objective accomplishment

About the accomplishment of the objectives proposed at the beginning and during the project, we consider that all of them were successfully fulfilled. In addition, the work plan was followed in time, completing the objectives in a reasonable time so that all of them could be developed.

At an initial phase we explored the available alternatives of object detection models and VQA systems. We tested different approaches of the models and investigated the datasets which are most utilized for training this sort of models, deciding finally to use COCO. Hence, we fulfilled objectives  $O_1$  and  $O_2$ , established in the introduction. Objectives  $O_3$ ,  $O_4$  and  $O_5$  complete the first part of our proposal, the retrieving of relevant questions for the image by a case-based reasoning system. These objectives were completed successfully, conducting an exhaustive comparison and evaluation of 4 similarity measures. Then, the work was completed with the



Caption generada por **Gemini**:

*A cozy bedroom with a plush bed, two nightstands, and a window with white curtains.*

Caption generada por **nuestra propuesta**:

*Wooden wall with a digital clock, white table with no chairs, flowers in a vase, white cushioned pillows, and a digital clock on the wall holding towels.*



Caption generada por **Gemini**:

*A brown and white cow resting in a green field.*

Caption generada por **nuestra propuesta**:

*A cow with a tag on its ear is relaxing near mountains with trees in the background.*

Comparison between Gemini and our model in the generation of captions for two images used in the evaluation. The prompt given to Gemini was “Generate a single caption for the following image {image}” (where {image} represents the place where the image was inserted).

implementation of a way to generate descriptions using existing models, a VQA one and a text generator, accomplishing the objective  $O_6$ . Finally, an evaluation of the complete proposal was carried out, achieving the objective  $O_7$ . However, it is true that some objectives had had to be adapted in order to reach the completion of the rest of them. For instance, the initial idea for objective  $O_6$  was the implementation of our own VQA model, but it had to be dismissed due to a lack of time. About objective  $O_7$ , we cast doubts on the validity of the evaluation results, so it is future work to revise and improve the evaluation method.

## ICCBR paper

We would also like to highlight that our work in question retrieval was included in a paper which was published at the International Conference on Case-Based Reasoning ICCBR 2024. In this paper we introduced the idea of using a Case-Based Reasoning architecture for image caption generation, by the retrieval of questions made to similar images previously and taking advantage of the acquired experience. The paper focuses on the study of the most suitable way of retrieving questions for the explained purpose, conducting an analysis of image similarity measures but not building the complete description generator system.

## Limitations and future work

Our proposal has some limitations regarding the amount of data available in our case base and some of the methods chosen to carry out the different steps of our model.

The case base is one of the most limiting and determining elements of a CBR model. With the goal of speeding up tests and results acquisition, we decided to create a case base composed of 5000 images. This considerably limits the system's capacity to retrieve similar images to the input one. In this aspect, the system could benefit substantially from an expansion of the case base. Not only the case base size, but its content too, can improve the model's performance. Although our case base is built with generic images, our proposal becomes truly useful when dealing with cases that address specific experiences. For example, there is a high probability that professional photographer user will not make the same questions about the same images than a visually impaired user. Having a case base addressing specific experiences could make this process highly effective in generating descriptions focused on a specific domain.

Another current limitation, and improvement area of our CBR model, is the way in which the questions retrieved from the most relevant cases are used. A detailed analysis of similarity metrics between images was carried out in order to be able to retrieve the most relevant cases of our base. A way of retrieving the most relevant questions from the candidates associated to retrieved images was also achieved through an iterative study of it. However, there is a significant improvement margin in the way these questions and their answers are utilized to generate a description. One of the cases that stands out the most is when the final description includes negative information (due to negative answers to questions) such as "There is no x" or "x can't be found in the image" (as it can be observed in figure 6.1), which is usually irrelevant in short descriptions. This effect might be mitigated by a larger case base, but also can be improved with a better retrieval of relevant questions or a refinement of the input prompt. As well, the case in which the retrieved questions are not suitable for the input image can be improved. This problem is usually caused because of questions asking about elements present in similar images but not in the input image. This issue is more damaging for the model than the previous one because instead of adding irrelevant but correct information about the image, it incorporates false information to the description. This is due to the test generation mode using question-answer pairs provided by the VQA system about objects not present in the image. Due to the adverse effect this issue has on the final description, we propose two ways of addressing it. One way to solve the problem is to discard questions that the VQA model has answered with low confidence, assuming these types of questions do not make sense to be asked about the input image. The other proposal is to adapt the question replacing key elements of the question with others that make sense for the input images. For example, taking into account an input image about a dog, if any of the retrieved questions asks about a cat, replace the word "cat" with the word "dog". This alternative can be challenging to implement but we believe it can offer a significant improvement in our model's performance.

Considering the proposed objectives, the obtained results and the limitations our model shows, there is a wide potential improvement in our concept.

Firstly, a part of future work is the improvement of the actual model, keeping testing it. For instance, the increase of case base could help some results and open further investigation lines about possible new problems of our system. In addition, we can investigate new image similarity measures (and combination among them)

and more feature extracting systems to build the image embeddings. Additionally, we could conduct tests for subsets of the case base studying the obtained results in order to determine if our model is especially useful for a certain type of image.

Secondly, the development of our own VQA model with the previous study and fit of the parameters of the convolutional neural network could provide improvements to our system. In this way, and through more specific tasks (such as picking images belonging to some particular topic), the fitting of the neural networks which generate the answers would contribute to give value to our model.

Finally, and considering that it is an ambitious goal, we could develop our own description generation system, able to provide a caption from a prompt. We believe this would mean a significant improvement as the model would fit our specific purpose and necessities, avoiding the use of more general models which are not trained exclusively for this task. For instance, even with a clear and concise prompt, our model sometimes gave more information than demanded.

In conclusion, we believe that our model can be a good approach for image caption generation, offering a distinct perspective to the actual state of art.

# Bibliografía

- [1] AGRAWAL, A., BATRA, D., PARIKH, D. y KEMBHAVI, A. Don't just assume; look and answer: Overcoming priors for visual question answering. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 4971–4980. 2018.
- [2] AGRAWAL, A., LU, J., ANTOL, S., MITCHELL, M., ZITNICK, C. L., BATRA, D. y PARIKH, D. Vqa: Visual question answering. 2016.
- [3] ANDERSON, P., HE, X., BUEHLER, C., TENNEY, D., JOHNSON, M., GOULD, S. y ZHANG, L. Bottom-up and top-down attention for image captioning and visual question answering. En *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 6077–6086. 2018.
- [4] BANERJEE, S. y LAVIE, A. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. En *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, páginas 65–72. 2005.
- [5] BASU, K., SHAKERIN, F. y GUPTA, G. Aqua: Asp-based visual question answering. En *Practical Aspects of Declarative Languages: 22nd International Symposium, PADL 2020, New Orleans, LA, USA, January 20–21, 2020, Proceedings 22*, páginas 57–72. Springer, 2020.
- [6] BERGMANN, R., ALTHOFF, K.-D., MINOR, M., REICHLE, M. y BACH, K. Case-based reasoning. *KI*, vol. 23, páginas 5–11, 2009.
- [7] BOSSELUT, A., RASHKIN, H., SAP, M., MALAVIYA, C., CELIKYILMAZ, A. y CHOI, Y. Comet: Commonsense transformers for automatic knowledge graph construction. 2019.
- [8] DALAL, N. y TRIGGS, B. Histograms of oriented gradients for human detection. En *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, páginas 886–893 vol. 1. 2005.
- [9] DALE, R. y WHITE, M. Shared tasks and comparative evaluation in natural language generation. En *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*, páginas 1–6. 2007.

- [10] DEVLIN, J., CHANG, M., LEE, K. y TOUTANOVA, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, vol. abs/1810.04805, 2018.
- [11] DEVLIN, J., CHANG, M.-W., LEE, K. y TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.
- [12] DONAHUE, J., HENDRICKS, L. A., ROHRBACH, M., VENUGOPALAN, S., GUADARRAMA, S., SAENKO, K. y DARRELL, T. Long-term recurrent convolutional networks for visual recognition and description. 2016.
- [13] ELLIOTT, D. y KELLER, F. Image description using visual dependency representations. En *Proceedings of the 2013 conference on empirical methods in natural language processing*, páginas 1292–1302. 2013.
- [14] ELLIOTT, D. y KELLER, F. Comparing automatic evaluation measures for image description. En *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, páginas 452–457. 2014.
- [15] EPSTEIN, Z., HERTZMANN, A., AKTEN, M., FARID, H., FJELD, J., FRANK, M. R., GROH, M., HERMAN, L., LEACH, N., MAHARI, R., PENTLAND, A., RUSSAKOVSKY, O., SCHROEDER, H. y SMITH, A. Art and the science of generative ai. *Science*, vol. 380(6650), página 1110–1111, 2023. ISSN 1095-9203.
- [16] ESTER, M., KRIEGEL, H.-P., SANDER, J., XU, X. ET AL. A density-based algorithm for discovering clusters in large spatial databases with noise. En *kdd*, vol. 96, páginas 226–231. 1996.
- [17] GOYAL, Y., KHOT, T., SUMMERS-STAY, D., BATRA, D. y PARIKH, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 6904–6913. 2017.
- [18] HE, K., ZHANG, X., REN, S. y SUN, J. Deep residual learning for image recognition. 2015.
- [19] HUANG, G., LIU, Z., VAN DER MAATEN, L. y WEINBERGER, K. Q. Densely connected convolutional networks. 2018.
- [20] HUANG, L., WANG, W., CHEN, J. y WEI, X.-Y. Attention on attention for image captioning. En *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, páginas 4633–4642. 2019.
- [21] JIA, C., YANG, Y., XIA, Y., CHEN, Y.-T., PAREKH, Z., PHAM, H., LE, Q., SUNG, Y.-H., LI, Z. y DUERIG, T. Scaling up visual and vision-language representation learning with noisy text supervision. En *International Conference on Machine Learning*, páginas 4904–4916. PMLR, 2021.

- [22] KHOSLA, P., TETERWAK, P., WANG, C., SARNA, A., TIAN, Y., ISOLA, P., MASCHINOT, A., LIU, C. y KRISHNAN, D. Supervised contrastive learning. 2021.
- [23] KIM, W., SON, B. y KIM, I. Vilt: Vision-and-language transformer without convolution or region supervision. 2021.
- [24] KRIZHEVSKY, A., SUTSKEVER, I. y HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, vol. 60(6), página 84–90, 2017. ISSN 0001-0782.
- [25] KULKARNI, G., PREMRAJ, V., ORDONEZ, V., DHAR, S., LI, S., CHOI, Y., BERG, A. C. y BERG, T. L. Babytalk: Understanding and generating simple image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, vol. 35(12), páginas 2891–2903, 2013.
- [26] KUZNETSOVA, P., ORDONEZ, V., BERG, A., BERG, T. y CHOI, Y. Collective generation of natural image descriptions. En *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, páginas 359–368. 2012.
- [27] LI, J., LI, D., SAVARESE, S. y HOI, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. 2023.
- [28] LI, J., SEIVARAJU, R., GOTMARE, A., JOTY, S., XIONG, C. y HOI, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, vol. 34, páginas 9694–9705, 2021.
- [29] LI, S., KULKARNI, G., BERG, T., BERG, A. y CHOI, Y. Composing simple image descriptions using web-scale n-grams. En *Proceedings of the fifteenth conference on computational natural language learning*, páginas 220–228. 2011.
- [30] LI, W., GAO, C., NIU, G., XIAO, X., LIU, H., LIU, J., WU, H. y WANG, H. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. 2022.
- [31] LI, Y., BUBECK, S., ELDAN, R., GIORNO, A. D., GUNASEKAR, S. y LEE, Y. T. Textbooks are all you need ii: phi-1.5 technical report, 2023.
- [32] LI, Z., ZHANG, X., ZHANG, Y., LONG, D., XIE, P. y ZHANG, M. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.
- [33] LIN, C.-Y. Rouge: A package for automatic evaluation of summaries. En *Text summarization branches out*, páginas 74–81. 2004.
- [34] LIN, C.-Y. ROUGE: A package for automatic evaluation of summaries. En *Text Summarization Branches Out*, páginas 74–81. Association for Computational Linguistics, Barcelona, Spain, 2004.

- [35] LIN, T., GOYAL, P., GIRSHICK, R. B., HE, K. y DOLLÁR, P. Focal loss for dense object detection. *CoRR*, vol. abs/1708.02002, 2017.
- [36] LIN, T.-Y., MAIRE, M., BELONGIE, S., BOURDEV, L., GIRSHICK, R., HAYS, J., PERONA, P., RAMANAN, D., ZITNICK, C. L. y DOLLÁR, P. Microsoft coco: Common objects in context. 2015.
- [37] LINDEBERG, T. *Scale Invariant Feature Transform*, vol. 7. 2012.
- [38] MARINO, K., RASTEGARI, M., FARHADI, A. y MOTTAGHI, R. Ok-vqa: A visual question answering benchmark requiring external knowledge. En *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, páginas 3195–3204. 2019.
- [39] MIKOLOV, T., CHEN, K., CORRADO, G. y DEAN, J. Efficient estimation of word representations in vector space. 2013.
- [40] MITCHELL, M., DODGE, J., GOYAL, A., YAMAGUCHI, K., STRATOS, K., HAN, X., MENSCH, A., BERG, A., BERG, T. y DAUMÉ III, H. Midge: Generating image descriptions from computer vision detections. En *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, páginas 747–756. 2012.
- [41] MUENNIGHOFF, N., TAZI, N., MAGNE, L. y REIMERS, N. MTEB: Massive text embedding benchmark. En *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, páginas 2014–2037. Association for Computational Linguistics, Dubrovnik, Croatia, 2023.
- [42] NUHA, F. U. y AFIAHAYATI. Training dataset reduction on generative adversarial network. *Procedia Computer Science*, vol. 144, páginas 133–139, 2018. ISSN 1877-0509. INNS Conference on Big Data and Deep Learning.
- [43] O'SHEA, K. y NASH, R. An introduction to convolutional neural networks. 2015.
- [44] PAPINENI, K., ROUKOS, S., WARD, T. y ZHU, W.-J. Bleu: a method for automatic evaluation of machine translation. En *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, páginas 311–318. 2002.
- [45] PRUSTY, M., TRIPATHI, V. y DUBEY, A. A novel data augmentation approach for mask detection using deep transfer learning. *Intelligence-Based Medicine*, vol. 5, página 100037, 2021.
- [46] PUTZU, L., PIRAS, L. y GIACINTO, G. Convolutional neural networks for relevance feedback in content based image retrieval: A content based image retrieval system that exploits convolutional neural networks both for feature extraction and for relevance feedback. *Multimedia Tools and Applications*, vol. 79(37), páginas 26995–27021, 2020.

- [47] RADFORD, A., WU, J., CHILD, R., LUAN, D., AMODEI, D., SUTSKEVER, I. ET AL. Language models are unsupervised multitask learners. *OpenAI blog*, vol. 1(8), página 9, 2019.
- [48] RAMZAN, F., KHAN, M. U., REHMAT, A., IQBAL, S., SABA, T., REHMAN, A. y MEHMOOD, Z. A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks. *Journal of Medical Systems*, vol. 44, 2019.
- [49] RAMZAN, F., KHAN, M. U. G., REHMAT, A., IQBAL, S., SABA, T., REHMAN, A. y MEHMOOD, Z. A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks. *Journal of medical systems*, vol. 44, páginas 1–16, 2020.
- [50] RAVI, S., CHINCHURE, A., SIGAL, L., LIAO, R. y SHWARTZ, V. Vlc-bert: Visual question answering with contextualized commonsense knowledge. En *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, páginas 1155–1165. 2023.
- [51] REDMON, J. y FARHADI, A. Yolov3: An incremental improvement. 2018.
- [52] REITER, E. y BELZ, A. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, vol. 35(4), páginas 529–558, 2009.
- [53] SAGGION, H., RADEV, D., TEUFEL, S. y LAM, W. Meta-evaluation of summaries in a cross-lingual environment using content-based metrics. En *COLING 2002: The 19th International Conference on Computational Linguistics*. 2002.
- [54] SAP, M., LE BRAS, R., ALLAWAY, E., BHAGAVATULA, C., LOURIE, N., RASHKIN, H., ROOF, B., SMITH, N. A. y CHOI, Y. Atomic: An atlas of machine commonsense for if-then reasoning. En *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, páginas 3027–3035. 2019.
- [55] SEIJAS, C., MONTILLA, G. y FRASSATO, L. Identification of rodent species using deep learning. *Computación y Sistemas*, vol. 23(1), páginas 257–266, 2019.
- [56] SHI, Y., FURLANELLO, T., ZHA, S. y ANANDKUMAR, A. Question type guided attention in visual question answering. 2018.
- [57] SIMONYAN, K. y ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. 2015.
- [58] SINGH, Y. P., AHMED, S. A. L. E., SINGH, P., KUMAR, N. y DIWAKAR, M. Image captioning using artificial intelligence. *Journal of Physics: Conference Series*, vol. 1854(1), página 012048, 2021.
- [59] SPEER, R., CHIN, J. y HAVASI, C. Conceptnet 5.5: An open multilingual graph of general knowledge. 2018.

- [60] STUDHOLME, C., HILL, D. L. y HAWKES, D. J. An overlap invariant entropy measure of 3d medical image alignment. *Pattern recognition*, vol. 32(1), páginas 71–86, 1999.
- [61] SUTSKEVER, I., VINYALS, O. y LE, Q. V. Sequence to sequence learning with neural networks. 2014.
- [62] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GO-MEZ, A. N., KAISER, Ł. y POLOSUKHIN, I. Attention is all you need. *Advances in neural information processing systems*, vol. 30, 2017.
- [63] VEDANTAM, R., LAWRENCE ZITNICK, C. y PARikh, D. Cider: Consensus-based image description evaluation. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 4566–4575. 2015.
- [64] VEDANTAM, R., ZITNICK, C. L. y PARikh, D. Cider: Consensus-based image description evaluation. *CoRR*, vol. abs/1411.5726, 2014.
- [65] VEIT, A., MATERA, T., NEUMANN, L., MATAS, J. y BELONGIE, S. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.
- [66] WANG, P., WU, Q., SHEN, C., DICK, A. y VAN DEN HENGE, A. Explicit knowledge-based reasoning for visual question answering. En *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, páginas 1290–1296. 2017.
- [67] WANG, Z., YU, J., YU, A. W., DAI, Z., TSVETKOV, Y. y CAO, Y. Simvlm: Simple visual language model pretraining with weak supervision. 2022.
- [68] WEI, J., WANG, X., SCHUURMANS, D., BOSMA, M., XIA, F., CHI, E., LE, Q. V., ZHOU, D. ET AL. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, vol. 35, páginas 24824–24837, 2022.
- [69] WU, Q. ET AL. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, vol. 163, páginas 21–40, 2017.
- [70] YANG, J., DUAN, J., TRAN, S., XU, Y., CHANDA, S., CHEN, L., ZENG, B., CHILIMBI, T. y HUANG, J. Vision-language pre-training with triple contrastive learning. En *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 15671–15680. 2022.
- [71] YANG, Y., TEO, C., DAUMÉ III, H. y ALOIMONOS, Y. Corpus-guided sentence generation of natural images. En *Proceedings of the 2011 conference on empirical methods in natural language processing*, páginas 444–454. 2011.
- [72] ZHANG, P., LI, X., HU, X., YANG, J., ZHANG, L., WANG, L., CHOI, Y. y GAO, J. Vinvl: Revisiting visual representations in vision-language models. En *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 5579–5588. 2021.

- [73] ZHANG, P., ZENG, G., WANG, T. y LU, W. Tinyllama: An open-source small language model. 2024.
- [74] ZHENG, L., CHIANG, W.-L., SHENG, Y., ZHUANG, S., WU, Z., ZHUANG, Y., LIN, Z., LI, Z., LI, D., XING, E. P., ZHANG, H., GONZALEZ, J. E. y STOICA, I. Judging llm-as-a-judge with mt-bench and chatbot arena. 2023.