



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

**Validación de explicabilidad post-hoc de
modelos de tipo caja-negra usando
grandes modelos de lenguaje**

Autor(a): Alejandro Sánchez González
Tutor(a): Esteban García Cuesta

Madrid, Junio 2024

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial

Título: Validación de explicabilidad post-hoc de modelos de tipo caja-negra usando grandes modelos de lenguaje

Junio 2024

Autor(a): Alejandro Sánchez González
Tutor(a): Esteban García Cuesta
Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

Este trabajo de fin de máster, titulado “Validación de explicabilidad post-hoc de modelos de tipo caja-negra usando grandes modelos de lenguaje”, aborda los desafíos que presenta la naturaleza opaca de los modelos de inteligencia artificial (IA) de tipo caja negra. Estos modelos, a pesar de su alta precisión, carecen de transparencia e interpretabilidad, lo cual es un problema significativo en aplicaciones donde los procesos de toma de decisiones necesitan ser comprensibles y justificables.

El documento comienza exponiendo el problema de los modelos de caja negra en la IA, particularmente la dificultad de interpretar cómo modelos como las redes neuronales profundas y los *transformers* llegan a sus predicciones. Se introduce el concepto de Inteligencia Artificial Explicable (IAX) y la necesidad de técnicas de explicabilidad post-hoc para hacer estos modelos más transparentes.

La sección del estado del arte profundiza en las técnicas existentes de explicabilidad post-hoc como LIME, SHAP y Grad-CAM, que buscan proporcionar explicaciones locales y globales de las predicciones de los modelos. También se presenta la evolución de los grandes modelos de lenguaje (LLMs) como GPT-3 y GPT-4, que han demostrado capacidades notables en la comprensión y generación de texto similar al humano.

La metodología describe la integración de técnicas de explicabilidad con grandes modelos de lenguaje para mejorar la interpretabilidad de los modelos de caja negra. Se desarrollan métricas para medir la consistencia y la veracidad de las explicaciones generadas. El desarrollo experimental incluye el uso de varios conjuntos de datos y LLMs para validar explicaciones.

Los resultados presentan los hallazgos de los experimentos, destacando la efectividad de los LLMs en proporcionar explicaciones coherentes y contextualmente relevantes para las predicciones de los modelos. También se analiza la robustez y fiabilidad de estas explicaciones en diferentes contextos.

Finalmente, el documento concluye resumiendo las contribuciones de la integración técnicas de explicabilidad con LLMs para mejorar la transparencia de los modelos de IA, al validar las explicaciones obtenidas. Se sugiere una futura línea de investigación, que incluye la expansión de la aplicación de estas técnicas en un contexto diferente para evaluar cómo pueden cambiar las respuestas generadas de los modelos de lenguaje.

Abstract

This master's thesis, titled "Validación de explicabilidad post-hoc de modelos de tipo caja-negra usando grandes modelos de lenguaje" ("Validation of post-hoc explainability for black-box models using large language models"), addresses the challenges posed by the opaque nature of black-box artificial intelligence (AI) models. Despite their high accuracy, these models lack transparency and interpretability, which is a significant issue in applications where decision-making processes need to be understandable and justifiable.

The document begins by outlining the problem of black-box models in AI, particularly the difficulty in interpreting how models such as deep neural networks and transformers arrive at their predictions. The concept of Explainable Artificial Intelligence (XAI) is introduced, along with the need for post-hoc explainability techniques to make these models more transparent.

The state-of-the-art section delves into existing post-hoc explainability techniques like LIME, SHAP, and Grad-CAM, which aim to provide local and global explanations for model predictions. It also presents the evolution of large language models (LLMs) like GPT-3 and GPT-4, which have demonstrated remarkable capabilities in understanding and generating human-like text.

The methodology describes the integration of explainability techniques with large language models to improve the interpretability of black-box models. Metrics are developed to measure the consistency and truthfulness of the generated explanations. The experimental development includes using various datasets and LLMs to validate the explanations.

The results present the findings of the experiments, highlighting the effectiveness of LLMs in providing coherent and contextually relevant explanations for model predictions. The robustness and reliability of these explanations in different contexts are also analyzed.

Finally, the document concludes by summarizing the contributions of integrating explainability techniques with LLMs to enhance the transparency of AI models by validating the obtained explanations. A future line of research is suggested, which includes expanding the application of these techniques in different contexts to evaluate how the responses generated by language models might change.

Tabla de contenidos

1. Introducción	1
1.1. Objetivos	2
1.2. Estructura del documento	2
2. Estado del arte	3
2.1. Inteligencia Artificial Explicable	3
2.1.1. Técnicas de explicabilidad post-hoc	4
2.1.1.1. LIME	5
2.1.1.2. SHAP	7
2.1.1.3. Grad-CAM	9
2.1.1.4. <i>Counterfactual Explanations</i>	10
2.2. Modelos grandes de lenguaje	12
2.2.1. Evolución de los modelos de lenguaje	12
2.2.2. <i>Transformers</i>	14
2.2.3. Pre-entrenamiento y <i>fine-tuning</i>	15
2.3. Trabajo relacionado	16
3. Metodología y desarrollo experimental	19
3.1. Modelos empleados	19
3.1.1. GPT	19
3.1.2. Gemini	20
3.1.3. Llama2	20
3.1.4. Mistral	20
3.2. Conjuntos de datos	20
3.2.1. <i>Dataset</i> para el análisis consistencia	21
3.2.2. <i>Dataset</i> para el análisis de veracidad	21
3.2.3. Prompts	22
3.3. Métricas	23
3.3.1. Similitud semántica	23
3.3.2. Solapamiento	24
3.3.3. ROUGE	24
3.3.4. BLEU	25
3.3.5. Ejemplo de métricas	25
3.4. Implementación	27
3.4.1. Experimento de análisis de la consistencia	27
3.4.2. Experimento de análisis de la veracidad	28
3.4.2.1. Extensión del experimento	28

4. Análisis de resultados	31
4.1. Resultados del experimento de análisis de consistencia	31
4.1.1. Métricas	32
4.1.2. Resultados adicionales de consistencia	34
4.1.3. Enriquecimiento	36
4.2. Resultados del experimento de análisis de veracidad	37
4.2.1. Ejemplos de veracidad	37
4.2.2. Resultados de la extensión del experimento	38
5. Conclusiones y futuras líneas de investigación	43
5.1. Conclusiones	43
5.2. Trabajo futuro	44
Bibliografía	51
Anexo	52

Capítulo 1

Introducción

La inteligencia artificial (IA) ha avanzado notablemente en las últimas décadas, revolucionando múltiples áreas como la medicina, la industria, el transporte y los servicios financieros. Desde algoritmos simples que pueden clasificar correos electrónicos como spam hasta complejas redes de neuronas que potencian vehículos autónomos, la IA se ha convertido en una herramienta esencial para resolver problemas difíciles y optimizar procesos. Sin embargo, a medida que estos sistemas se vuelven más sofisticados, también se vuelven más opacos, es decir, su funcionamiento interno no es fácilmente interpretable por los seres humanos. Este tipo de modelos se denominan modelos de “caja-negra”.

Uno de los principales desafíos asociados con estos modelos es su falta de transparencia y explicabilidad. Los modelos de caja negra, como las redes neuronales profundas y los *transformers*, generan predicciones precisas, pero no proporcionan una comprensión clara de cómo llegan a esas predicciones. Esta falta de interpretabilidad puede ser un obstáculo significativo en aplicaciones críticas donde la toma de decisiones debe ser explicable y justificable.

En respuesta a esta necesidad, ha surgido un área de investigación centrada en la explicabilidad y la interpretabilidad de los modelos de IA, conocida como inteligencia artificial explicable (IAX). Este campo de estudio se dedica a desarrollar métodos y técnicas que permitan entender y justificar las decisiones y predicciones de los modelos de IA, especialmente aquellos considerados como cajas-negras debido a su complejidad y opacidad. La importancia de la IAX radica en la necesidad de generar confianza en la IA, facilitar la identificación de posibles sesgos y errores, y cumplir con regulaciones y normativas que exigen transparencia en el uso de estas tecnologías.

Dentro de esta vertiente existen técnicas de explicabilidad post-hoc, que buscan proporcionar interpretaciones a posteriori de las predicciones realizadas por modelos de caja negra. Estas técnicas incluyen métodos como LIME, SHAP, y los mapas de calor basados en Grad-CAM, entre otros. Aunque estas técnicas han avanzado significativamente, todavía se enfrentan a desafíos importantes, especialmente en términos de su validación y robustez.

En este contexto, los grandes modelos de lenguaje (LLMs, por sus siglas en inglés) como GPT-4, han surgido como herramientas poderosas capaces de procesar y generar texto con un alto grado de coherencia y comprensión contextual. Estos modelos, entrenados en grandes volúmenes de datos, han desarrollado un entendimiento del

lenguaje natural, lo cual puede ser aprovechado para mejorar la explicabilidad de los modelos de caja negra. La capacidad de los LLMs para sintetizar información, generar explicaciones coherentes y contextualizar datos complejos los convierte en posibles candidatos para validar las técnicas de explicabilidad post-hoc.

En conclusión, este trabajo de fin de máster se centra en la validación de técnicas de explicabilidad post-hoc utilizando grandes modelos de lenguaje. Por lo tanto, se propone que los LLMs pueden proporcionar una capa adicional de interpretación y validación a las técnicas existentes, mejorando así la transparencia y la confianza en los modelos de caja negra.

1.1. Objetivos

El objetivo principal de este trabajo es desarrollar un sistema que integre las explicaciones proporcionadas por los modelos de *machine learning* con los modelos de lenguaje. Para alcanzar este objetivo, se deben seguir los siguientes pasos:

- Desarrollar un conjunto de métricas para medir la consistencia y la veracidad de las respuestas generadas por los modelos de lenguaje.
- Evaluar la consistencia y la veracidad de las respuestas generadas por los LLMs.
- Analizar la variación en las respuestas al modificar el número de opciones en las respuestas.
- Enriquecer y validar las explicaciones obtenidas mediante los métodos de inteligencia artificial explicable.
- Determinar si los modelos pueden considerarse expertos en un dominio específico a partir de los resultados obtenidos con las métricas desarrolladas.

1.2. Estructura del documento

Para concluir la introducción, se propone la estructura del presente trabajo de fin de máster, dividido en varios capítulos:

1. **Estado del arte:** se realizará un análisis de varias técnicas actuales de explicabilidad post-hoc y se explicarán en detalle las propiedades y el funcionamiento de los grandes modelos de lenguaje. Esta revisión proporcionará una base teórica sólida y ayudará a comprender los componentes clave de este trabajo.
2. **Metodologías y desarrollo experimental:** se propondrán varios experimentos para integrar LLMs en el proceso de explicabilidad post-hoc. Estos experimentos incluirán la selección de técnicas de explicabilidad, la implementación de LLMs para la generación de explicaciones y la definición de métricas para evaluar la robustez y confianza de las explicaciones generadas.
3. **Análisis de resultados:** los resultados obtenidos se analizarán para determinar la consistencia y la veracidad de los LLMs.
4. **Conclusiones y futuras líneas de investigación:** finalmente, se discutirán las conclusiones del estudio, sus implicaciones para la inteligencia artificial y las posibles direcciones futuras para la investigación.

Capítulo 2

Estado del arte

El objetivo principal de este capítulo, el estado del arte, consiste en detallar los principales conceptos que serán necesarios para entender el posterior desarrollo. Esto incluye la definición de inteligencia artificial explicable con sus respectivas técnicas de explicabilidad y las propiedades de los grandes modelos de lenguaje. Finalmente, el capítulo concluirá con una sección que incluye trabajos relacionados con el alcance de este documento.

2.1. Inteligencia Artificial Explicable

La inteligencia artificial explicable es una rama emergente dentro del campo de la inteligencia artificial que se centra en desarrollar métodos y técnicas que permiten entender y confiar en los resultados generados por modelos de IA. A medida que los sistemas de IA se vuelven más complejos y difíciles de descifrar, especialmente con el uso de técnicas avanzadas como el aprendizaje profundo (*deep learning*), la necesidad de explicabilidad se vuelve esencial.

En la figura 2.1 podemos observar un gráfico que representa la relación entre la precisión de los modelos y su interpretabilidad. En este gráfico se destaca el *deep learning* como uno de los modelos con mayor precisión, pero con menor interpretabilidad. Es relevante mencionar el objetivo a alcanzar en los próximos años, representado en la figura mediante un triángulo, donde se espera que los futuros modelos sean completamente interpretables y alcancen la mayor precisión posible.

En primer lugar, se procederá a definir algunos conceptos fundamentales, como explicación e interpretación. No obstante, es importante señalar que no existe una formalización general ni un consenso universal sobre estas definiciones. Por un lado, una **interpretación** se refiere a la asignación de un concepto abstracto, como una clase prevista por un modelo, a un dominio que el ser humano pueda entender, es decir, se trata de traducir los resultados del modelo (que pueden ser complejos y abstractos) a términos y conceptos que sean claros para las personas. Por otro lado, una **explicación** consiste en identificar y comunicar el conjunto de características específicas del dominio interpretable que han contribuido a una decisión específica del modelo [2, 3], dicho de otro modo, se trata de identificar cuáles características o factores influyeron en la decisión del modelo para una predicción concreta. También existen términos como la **transparencia**, la cual implica que, si un modelo es

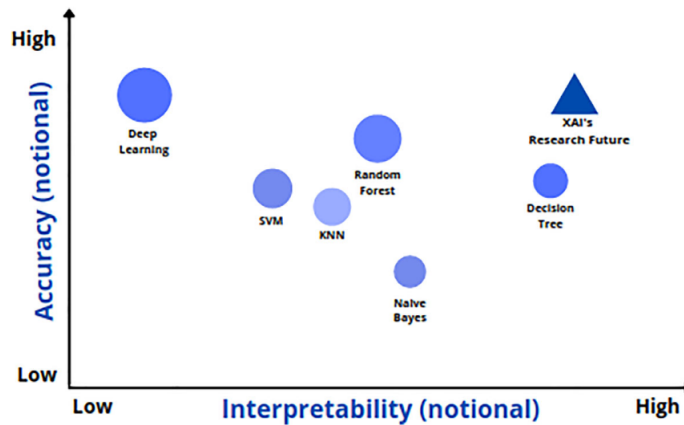


Figura 2.1: Precisión frente a interpretabilidad para distintos modelos [1].

transparente, este será inherentemente comprensible e interpretable [4].

En este contexto, ambos términos, explicabilidad e interpretabilidad, son muy utilizados. La **explicabilidad** se refiere al nivel en el que un sistema puede aclarar la causa de sus decisiones o resultados. La **interpretabilidad**, por otro lado, se centra en el nivel en el que un agente obtiene, y puede utilizar, tanto la información incluida en las explicaciones dadas por el sistema como la información proporcionada por el nivel de transparencia del sistema [5].

Otros conceptos importantes son los modelos de caja-blanca y los modelos de caja-negra. Un modelo de **caja-blanca** se caracteriza por su transparencia y explicabilidad. A diferencia de los modelos de **caja-negra**, cuyos procesos internos son opacos y difíciles de interpretar. Los modelos de caja-blanca permiten a los usuarios comprender cómo se toman las decisiones y cómo se generan las predicciones. Esto es posible gracias a que los parámetros y el funcionamiento interno del modelo son accesibles y entendibles por un ser humano. Del mismo modo, en el ámbito del *deep learning*, la distinción entre un modelo de caja-blanca y uno de caja-negra se basa en si los parámetros y la arquitectura del modelo son conocidos por el usuario o no [6].

A pesar de que se han publicado estudios que buscan concienciar a los investigadores sobre la importancia de utilizar modelos interpretables desde el principio, evitando así la necesidad de explicar los modelos de caja negra [7], sigue siendo inevitable el empleo de estos últimos debido a sus buenos resultados. Por lo tanto, han surgido metodologías que facilitan la explicación de estos modelos complejos.

2.1.1. Técnicas de explicabilidad post-hoc

Una vez definidos los conceptos de la IAX, se explicarán las técnicas de explicabilidad post-hoc. Este tipo de metodologías se aplican después de que un modelo ha sido entrenado y ha realizado sus predicciones. Las técnicas de explicabilidad post-hoc buscan proporcionar una comprensión de cómo y por qué el modelo toma ciertas decisiones, sin alterar el funcionamiento interno del mismo.

En primer lugar se debe diferenciar el alcance que tiene cada técnica, implicando explicaciones locales y globales. La interpretabilidad local explica las características del modelo y el impacto de las características de entrada para una predicción especí-

fica, mientras que la interpretabilidad global explica toda la lógica de un modelo y el razonamiento que subyace a todos los resultados posibles [8].

Además de diferenciar entre el alcance de las técnicas y la accesibilidad del modelo (caja-negra y caja-blanca), las técnicas se pueden clasificar por la búsqueda empleada (basada en perturbaciones frente a basada en gradientes) y la unidad básica de explicación (importancia de las características frente a basada en reglas) [9].

Por un lado, las técnicas basadas en perturbaciones implican la modificación de las entradas del modelo para observar cómo cambian las salidas, en contraste a las técnicas basadas en gradientes, que utilizan la información derivada de los gradientes del modelo respecto a sus entradas para proporcionar explicaciones.

Por otro lado, las técnicas de importancia de las características se centran en evaluar cuánto contribuye cada característica (o variable) a las predicciones del modelo. Esto ayuda a entender qué características son más influyentes en el comportamiento del modelo, mientras que las técnicas basadas en reglas buscan generar explicaciones en forma de reglas lógicas o condicionales que describen cómo se toman las decisiones. Estas reglas son fáciles de entender e interpretar por los seres humanos.

También existen otras subdivisiones de las técnicas de explicabilidad, donde se incluyen las explicaciones textuales de los resultados del sistema, las visualizaciones de las representaciones o modelos aprendidos y las explicaciones mediante ejemplos [10, 11]. Como podemos observar, no hay un consenso claro entorno a la clasificación de estas metodologías. No obstante, es posible abordar todas las técnicas detalladas en este trabajo utilizando los conceptos mencionados. A continuación, se expondrán algunas de estas técnicas post-hoc, incluyendo SHAP, que se emplea en el desarrollo de este trabajo.

2.1.1.1. LIME

LIME (*Local Interpretable Model-Agnostic Explanations*) es una técnica que explica las predicciones de cualquier clasificador de forma interpretable, mediante el aprendizaje de un modelo interpretable localmente en torno a la predicción [12].

Siguiendo el propio nombre de LIME, la terminología *Local Interpretable* implica que es una técnica de interpretación local, lo que significa que su enfoque está en la explicación de predicciones específicas en lugar de ofrecer una comprensión global del modelo. En la figura 2.2 observamos cómo funcionan las predicciones individuales. A través de unos datos y un modelo de tipo caja-negra se predice que un paciente tiene un resfriado. LIME destaca los síntomas en el historial del paciente que llevan a esa predicción. Por un lado, los síntomas “estornudar” y “dolor de cabeza” contribuyen positivamente a la predicción de resfriado, mientras que el síntoma de “no fatiga” contribuye negativamente a la predicción. Gracias a estos resultados, un doctor puede decidir con conocimiento de causa si confiar en el resultado del modelo o no.

Por otro lado, *Model-Agnostic* significa que LIME puede aplicarse a cualquier tipo de modelo de aprendizaje automático sin importar su naturaleza interna. Esto quiere decir que LIME no está limitado a un tipo específico de modelo y que puede generar explicaciones para cualquier modelo independientemente de su complejidad o estructura interna. Finalmente, *Explanations* o explicaciones, denotan la salida producida.

Una vez explicada la finalidad y las predicciones de una forma general, se va a pro-

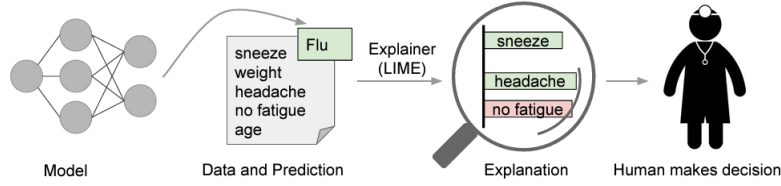


Figura 2.2: Explicación de una predicción individual [12].

ceder a explicar el funcionamiento interno de LIME. En primer lugar, tenemos que la explicación producida se obtiene mediante la ecuación 1, donde $\xi(x)$ denota la explicación generada para la instancia x , G es el conjunto de modelos explicativos posibles, $L(f, g, \pi_x)$ es una función de pérdida que mide qué tan bien el modelo explicativo g aproxima al modelo complejo f en la vecindad π_x alrededor de la instancia x , y $\Omega(g)$ es una regularización que impone restricciones en la complejidad del modelo explicativo g .

$$\xi(x) = \arg \min_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (1)$$

En resumen, la expresión anterior refleja el proceso de encontrar una explicación simple y precisa para una instancia x mediante la minimización de una función de pérdida que compara el modelo explicativo g con el modelo complejo f en una vecindad local, penalizando al mismo tiempo la complejidad del modelo g .

El funcionamiento de LIME se puede desglosar en los siguientes pasos:

- **Selección de la instancia de interés:** se elige una instancia específica x del conjunto de datos para la cual se desea obtener una explicación.
- **Generación de datos perturbados:** se crean nuevas instancias perturbando los valores de las características de la instancia x . Esto se realiza generando muestras alrededor de la instancia original utilizando una distribución apropiada.
- **Predicciones del modelo complejo:** las instancias perturbadas se pasan al modelo complejo f para obtener sus predicciones.
- **Pesos de las instancias:** las instancias perturbadas se ponderan según su similitud con la instancia original x .
- **Entrenamiento del modelo interpretable:** se entrena un modelo interpretable g (e.g. una regresión lineal) utilizando las instancias perturbadas y sus predicciones obtenidas del modelo complejo f , ponderando las instancias según su similitud con la instancia original.
- **Generación de la explicación:** el modelo interpretable g proporciona una explicación local de cómo el modelo complejo f toma decisiones en torno a la instancia x . Las características y sus coeficientes en el modelo interpretable indican la importancia local de cada característica.

Este funcionamiento se puede observar de manera simplificada en la figura 2.3. Esta figura es un ejemplo para presentar de manera sencilla el funcionamiento de LIME. El fondo azul y rosa representa las regiones donde el modelo complejo predice una de

las dos clases. Los círculos azules representan las instancias que pertenecen a la clase azul y las cruces rojas representan las instancias que pertenecen a la clase rosa. El tamaño de ambos indica la proximidad a la instancia que se está explicando, representada por la gran cruz roja. LIME generará una explicación para esta instancia. La línea discontinua representa el límite de decisión del modelo explicativo g generado por LIME. Este modelo es una aproximación lineal local del modelo complejo f en la vecindad de la instancia de interés. Esta línea muestra cómo el modelo explicativo divide el espacio de características para predecir las dos clases.

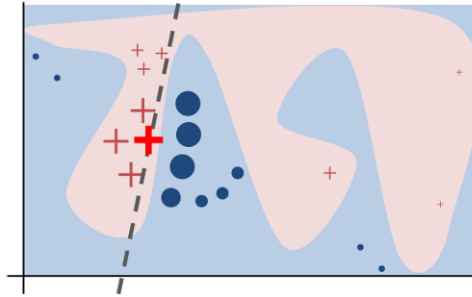


Figura 2.3: Ejemplo para presentar la intuición de LIME [12].

No obstante, LIME tiene ciertas limitaciones, algunos estudios [13] concluyen que, aunque LIME ayuda a aumentar la interpretabilidad de los modelos, se necesitan análisis de usabilidad para mejorar la experiencia del usuario, y herramientas y técnicas para facilitar las comparaciones globales. Otros estudios [14] sugieren que una propiedad importante de los métodos de explicación de modelos es la estabilidad: las ejecuciones repetidas del algoritmo sobre el mismo objeto deben lanzar resultados coherentes. Sin embargo, se demuestra que LIME no es estable debido a la aleatoriedad inducida en la generación de muestras sintéticas.

2.1.1.2. SHAP

SHAP (*SHapley Additive exPlanations*) es otra técnica de explicabilidad post-hoc, que está basada en la teoría de juegos [15]. De manera específica, SHAP se basa en los valores de Shapley para asignar de manera justa la importancia de cada característica en una predicción. El valor Shapley es un concepto en teoría de juegos cooperativos propuesto en 1953, que asigna un valor a cada jugador en un juego cooperativo, reflejando su contribución marginal al juego [16]. Diferentes coaliciones de jugadores consiguen diferentes recompensas. En el contexto de la IA, los “jugadores” son las características del modelo y la “recompensa” es la predicción del modelo.

En la siguiente ecuación, podemos ver cómo se calculan los valores de Shapley. Es importante explicar cada término para entender la ecuación, por lo que d es el conjunto de todas las características, S es un subconjunto de d que no incluye la característica i , $v(S)$ es la predicción del modelo cuando solo las características en S están presentes y ϕ_i es el valor de Shapley para la característica i .

$$\phi_i(V) = \sum_{S \subseteq D \setminus i} \frac{|S|!(|d| - 1 - |S|)!}{|d|!} [v(S \cup \{i\}) - v(S)] \quad (2)$$

En resumen, SHAP utiliza los valores de Shapley para distribuir de manera justa la

contribución de cada característica a la predicción del modelo. Después, descompone la predicción de un modelo complejo en una suma de contribuciones individuales de las características, más una constante. Esta descomposición permite entender cómo cada característica afecta a la predicción total. Finalmente, se tiene que los valores de Shapley calculados son consistentes: si la importancia de una característica aumenta en un modelo, su valor de Shapley no disminuye. A continuación, se presentan dos gráficos obtenidos utilizando esta técnica, los cuales proporcionan una visión completa de cómo se pueden interpretar y visualizar los resultados obtenidos con SHAP.

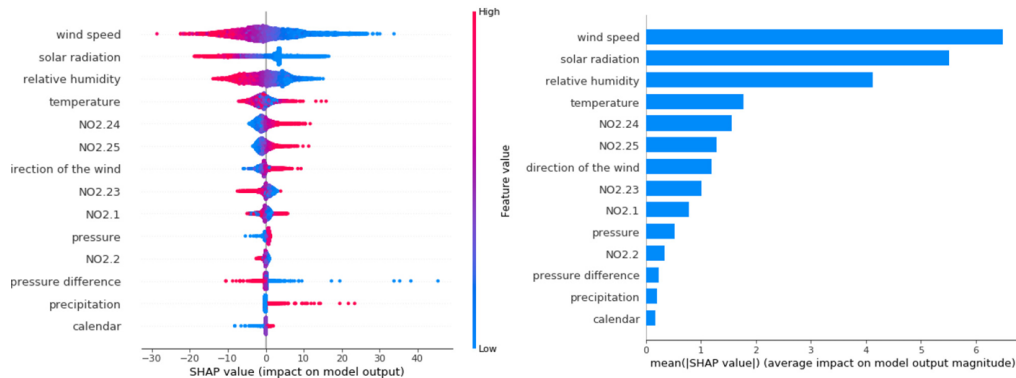


Figura 2.4: Ejemplo de gráficos obtenidos mediante SHAP [17].

En la figura 2.4 observamos dos gráficos obtenidos con SHAP en un estudio realizado sobre el NO_2 [17]. A la izquierda, tenemos un gráfico resumen que muestra que cuanto mayor es el valor SHAP de una característica, mayor será el valor de NO_2 . A la derecha, se presenta el valor relativo de la importancia de cada característica del estudio, obtenido tomando el valor absoluto medio de los valores SHAP.

SHAP tiene alguna limitación, como el elevado coste computacional. Por ello, en el artículo original donde se enuncia [15], se proponen varios métodos adicionales. En primer lugar, **Kernel SHAP** es una aproximación de la técnica original para modelos agnósticos, por lo que sirve para cualquier tipo de modelo, combina Linear LIME y los valores de Shapley, es un método más eficiente que requiere menos evaluaciones del modelo para lograr una precisión similar en la estimación de los valores de SHAP.

El artículo también propone otras aproximaciones para modelos específicos, como **Linear SHAP**, para modelos lineales, que permite que los valores SHAP pueden aproximarse directamente a partir de los coeficientes de peso del modelo. Asimismo, otra aproximación es **Low-Order SHAP**, para regresión lineal, que realiza aproximaciones eficientes para modelos con pocas interacciones entre características y sirve para simplificar cálculos. También, se propone **Max SHAP**, que maximiza la relevancia de las características más influyentes, optimizando las explicaciones para interpretaciones clave. Por último, **Deep SHAP**, que está adaptado para redes neuronales profundas y combina los valores Shapley con DeepLIFT [18].

Para finalizar la descripción de esta técnica, es importante subrayar que las características relevantes obtenidas de un estudio mediante SHAP se utilizarán en la sección de desarrollo posterior. Por lo tanto, el enfoque principal de este trabajo se centra en esta técnica de explicabilidad post-hoc, destacando su aplicabilidad y utilidad en la interpretación de modelos complejos.

2.1.1.3. Grad-CAM

Grad-CAM (*Gradient-weighted Class Activation Mapping*) es una técnica de explicabilidad post-hoc que se utiliza para generar mapas de calor ponderados por gradientes [19], los cuales resaltan las regiones de una imagen que son más importantes para la predicción de una clase específica en una red neuronal convolucional (CNN).

En la figura 2.5 se muestran varias imágenes de la visualización de contribuciones de características para la clasificación de “gato” y “perro” usando diferentes técnicas. En lo referente a Grad-CAM, se tienen las imágenes (c) e (i) que muestran las regiones importantes para las predicciones de las clases “gato” y “perro”, respectivamente. Las imágenes (f) y (l) utilizan ResNet [20] para visualizar las regiones discriminatorias relevantes de las mismas clases. Grad-CAM localiza estas regiones clave resaltando las áreas con alta puntuación para cada clase, proporcionando así interpretaciones claras y detalladas de las predicciones del modelo.

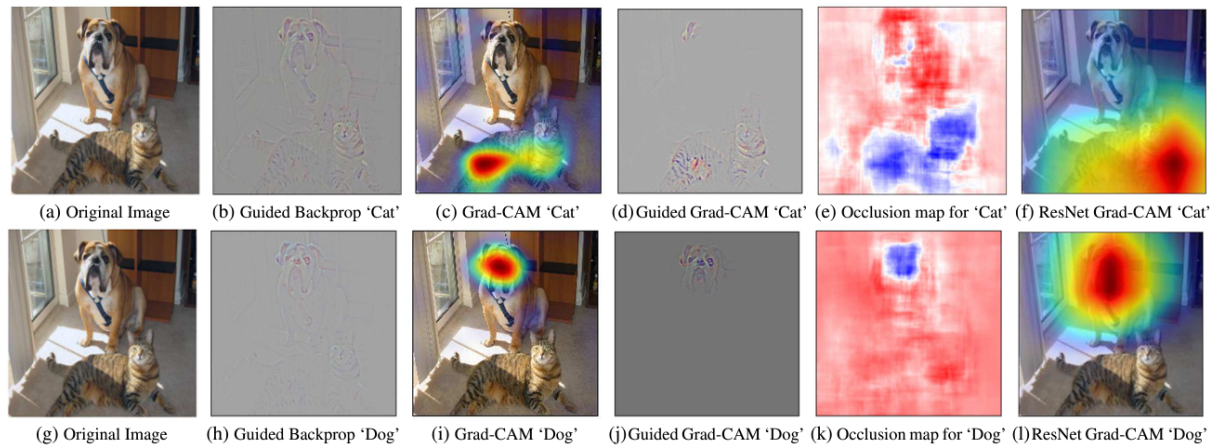


Figura 2.5: Visualización de contribuciones de características para “gato” y “perro” usando Guided Backpropagation, Grad-CAM y Occlusion Maps. [19].

Los fundamentos de Grad-CAM son los siguientes:

- **Activaciones de convolución:** las activaciones de una capa convolucional específica A^k capturan características espaciales importantes de la imagen de entrada. Aquí, A^k representa la activación del k-ésimo mapa de características en la capa convolucional.
- **Gradientes:** los gradientes de la puntuación de la clase objetivo y^c respecto a las activaciones de la capa convolucional A^k indican la importancia de cada neurona en A^k para la predicción de la clase c . Estos gradientes se denotan como $\frac{\partial y^c}{\partial A^k}$.
- **Promedio global de gradientes:** se calcula el promedio global de los gradientes en todas las ubicaciones espaciales (i, j) del mapa de características A^k para obtener un peso α_k^c :

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (3)$$

donde Z es el número total de ubicaciones espaciales en el mapa de características A^k .

- **Mapas de calor:** se obtiene un mapa de calor $L_{\text{Grad-CAM}}^c$ como una combinación lineal ponderada de las activaciones de la capa convolucional, usando los pesos α_k^c :

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\sum_k \alpha_k^c A^k \right) \quad (4)$$

La función ReLU (*Rectified Linear Unit*) se aplica para mantener solo las características que tienen una influencia positiva en la clase objetivo.

A partir de estos fundamentos, el procedimiento de Grad-CAM comienza realizando una propagación hacia adelante de la imagen de entrada a través de la red, para obtener las activaciones de la capa convolucional de interés y la puntuación de la clase objetivo. Después, se calcula el gradiente de la puntuación de la clase objetivo respecto a las activaciones de la capa convolucional seleccionada y también se calcula el promedio global de estos gradientes para obtener los pesos α_k^c . Posteriormente, se calcula el mapa de calor ponderando las activaciones de la capa convolucional con los pesos α_k^c y se aplica la función ReLU. Finalmente, el mapa de calor resultante se interpola a la resolución de la imagen de entrada y se superpone a la imagen original para visualizar las regiones más importantes para la predicción. La figura 2.6 muestra de manera visual este procedimiento.

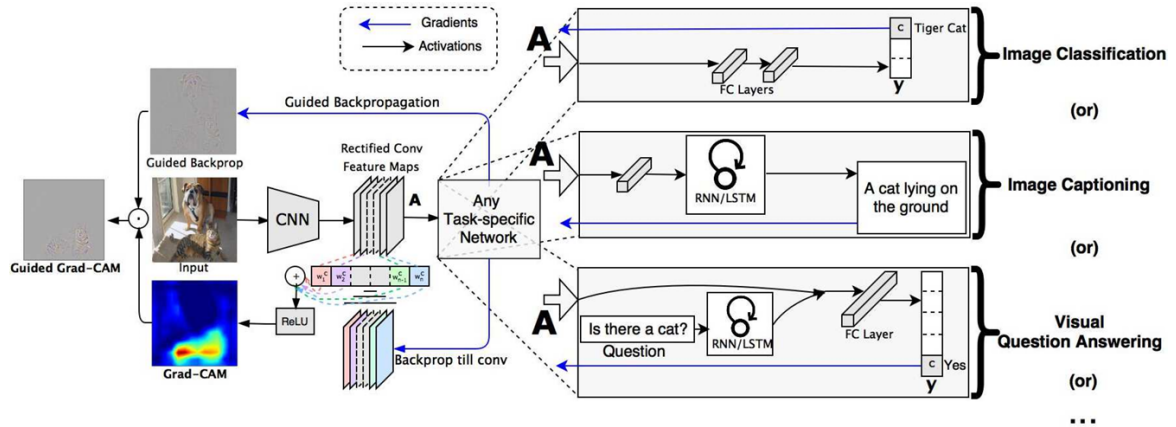


Figura 2.6: Ejemplo del funcionamiento de Grad-CAM [19].

2.1.1.4. Counterfactual Explanations

Las explicaciones contrafácticas (*Counterfactual Explanations*) son una técnica de explicabilidad post-hoc que identifica cambios mínimos necesarios en las características de una entrada para alterar la predicción del modelo. Funcionan al encontrar una instancia cercana x' a la entrada original x que produce una predicción diferente y' .

La definición matemática del problema consiste en, dada una instancia de entrada x con una predicción del modelo $f(x) = y$, buscar una instancia x' tal que $f(x') = y'$, donde $y' \neq y$. El objetivo se basa en resolver un problema de optimización para

encontrar la nueva instancia. Para ello, la fórmula de Wachter [21] proporciona un marco formal, expresado como:

$$\min_{x'} \lambda \cdot d(x, x') + L(f(x'), y') \quad (5)$$

donde:

- $d(x, x')$ es una medida de distancia que cuantifica la similitud entre la instancia original x y la instancia modificada x' .
- $L(f(x'), y')$ es una función de pérdida que penaliza las diferencias entre la predicción $f(x')$ y el valor objetivo deseado y' .
- λ es un parámetro de regularización que controla el equilibrio entre la proximidad de x y x' y la necesidad de cambiar la predicción.

El resultado de esta optimización es una instancia x' que representa el cambio mínimo necesario en las características de x para obtener una predicción y' diferente. Esto proporciona una explicación contrafáctica clara y precisa, mostrando exactamente cómo y por qué la predicción del modelo cambiaría si se modificaran ciertas características de la entrada.

La figura 2.7 muestra visualmente el resultado de aplicar las explicaciones contrafácticas. El ejemplo original (círculo azul) se clasifica en la clase negativa (área naranja), esto lo podríamos asociar a una persona a la que se le deniega un préstamo.

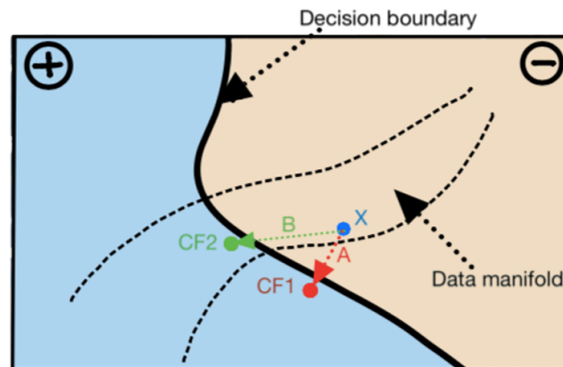


Figura 2.7: Ejemplo de explicación contrafactual [22].

A continuación, existen dos posibles caminos, obtenidos resolviendo el problema de optimización, para que la instancia cruce el límite de decisión. Los puntos finales de ambas trayectorias (círculos azules y verdes) son contrafactuales válidos para el punto original. Esto se podría asociar a que la persona a la que se le deniega el préstamo, obtiene una mejora (perturbación) en los datos iniciales que fueron denegados, como una subida de sueldo o un nivel de educación superior, de manera que se clasifique en la clase positiva (área azul) y se apruebe el préstamo. Se observa que la trayectoria roja es la más corta (implica un cambio menor en los datos de entrada), mientras que la verde es más larga (implica un cambio superior).

2.2. Modelos grandes de lenguaje

El siguiente concepto a tratar son los modelos grandes de lenguajes o LLMs (*Large Language Models*). Los LLMs son sistemas entrenados con parámetros masivos diseñados para comprender, generar y manipular texto en lenguaje natural. Estos modelos de lenguaje han experimentado avances significativos en los últimos años, impulsando la capacidad de las máquinas para interactuar de manera más efectiva con los seres humanos. Un LLM capaz debe tener cuatro características clave: (i) comprensión profunda del contexto del lenguaje natural; (ii) capacidad de generar texto similar al humano; (iii) conciencia contextual, especialmente en dominios de conocimiento intensivo; (iv) fuerte capacidad de seguimiento de instrucciones, útil para la resolución de problemas y la toma de decisiones [23].

2.2.1. Evolución de los modelos de lenguaje

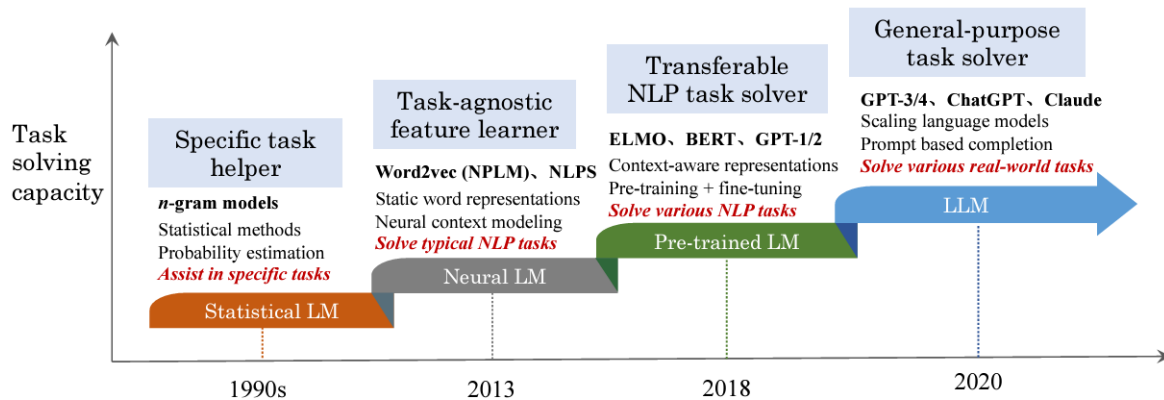


Figura 2.8: Evolución de las diferentes etapas de los modelos de lenguaje [24].

La evolución de los modelos de lenguaje se puede dividir en cuatro etapas, representadas gráficamente en la figura 2.8. En primer lugar, surgen los **modelos de lenguaje estadísticos (SLM)**, que se basan en la probabilidad de una secuencia de palabras dada la secuencia previa. Los SLM con una longitud de contexto fija n , se denominan modelos de lenguaje de n -gramas. Estos modelos presentan limitaciones para capturar dependencias a largo plazo y son difíciles de escalar.

En segundo lugar, aparecen los **modelos de lenguaje neuronales (NLM)**, que se caracterizan por modelar la probabilidad de secuencias de palabras mediante redes neuronales, como las redes neuronales recurrentes (RNN) [25]. Estos modelos son capaces de manejar secuencias de longitud variable y capturar dependencias a corto plazo; sin embargo, los NLM presentan limitaciones con las secuencias largas. Además, en este periodo, se propuso Word2Vec [26, 27], una técnica eficiente para representar palabras en un espacio vectorial, facilitando el análisis posterior de sus relaciones semánticas y sintácticas.

En esta etapa, el concepto de **embeddings** se popularizó gracias a modelos como Word2Vec, GloVe [28], y FastText [29], que expandieron esta idea mejorando la calidad y eficiencia de los *embeddings*. Los *embeddings* son representaciones vectoriales densas de palabras o frases, donde palabras con significados similares tienen representaciones similares en el espacio vectorial. La figura 2.9 ilustra de manera simplificada cómo los *embeddings* representan palabras en un espacio de menor dimensión,

enormemente a los modelos de la etapa anterior. Estos modelos no solo mejoran en la comprensión y generación de texto, sino que también pueden realizar tareas complejas con poca o ninguna supervisión adicional, como la resolución de problemas matemáticos, la traducción automática y la generación de código. La habilidad de GPT-3 para resolver tareas con pocos ejemplos de aprendizaje contextual destaca su avanzada capacidad en comparación con sus predecesores.

2.2.2. Transformers

Los **transformers** son una arquitectura de red neuronal introducida por Vaswani et al. en el artículo “Attention is All You Need” en 2017 [33]. Este modelo ha revolucionado el campo del procesamiento del lenguaje natural y se ha convertido en la base de muchos de los modelos de lenguaje más avanzados, como BERT y GPT. La principal innovación de los *transformers* es el mecanismo de atención, que permite a la red enfocarse en diferentes partes de la entrada para mejorar la comprensión y generación de texto. La estructura descrita en el artículo original se muestra visualmente en la figura 2.10.

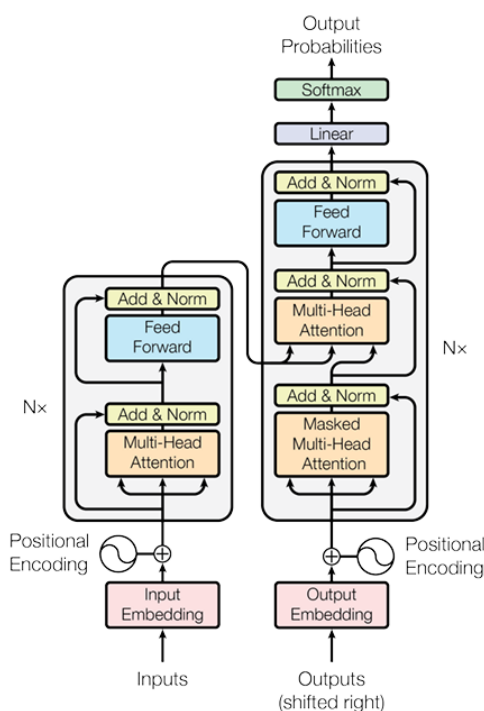


Figura 2.10: Arquitectura del *transformer* [33].

La arquitectura del *transformer* consta de dos componentes principales: el codificador (*encoder*) y el decodificador (*decoder*). Cada uno de estos componentes está formado por una serie de capas que utilizan mecanismos de atención y redes neuronales *feed-forward*. El **mecanismo de atención** es el núcleo del modelo *transformer*. Permite que cada elemento de la secuencia de entrada preste atención a otros elementos cuando se está procesando. Esto se logra mediante los siguientes componentes:

- **Scaled dot-product attention**: calcula una ponderación entre elementos de la secuencia de entrada para determinar su relevancia mutua.
- **Multi-head attention**: permite al modelo enfocarse en diferentes partes de la

secuencia simultáneamente. Se utilizan múltiples cabezas de atención paralelas que luego se combinan.

El **codificador** consta de dos subcapas principales:

- **Multi-head self-attention:** cada palabra en la secuencia de entrada puede atender a todas las demás palabras en la misma secuencia.
- **Feed-forward neural network:** una red neuronal completamente conectada que se aplica a cada posición de la secuencia de forma independiente.

Además, se utilizan conexiones residuales y normalización de capas para estabilizar y acelerar el entrenamiento.

El **decodificador** es similar al codificador, se utilizan conexiones residuales y normalización de capas. Además, cada capa también tiene dos subcapas principales, pero con una diferencia:

- **Masked multi-head self-attention:** se enmascaran las posiciones futuras en la secuencia de entrada para evitar que la red “vea” las palabras que aún no se han generado.
- **Encoder-decoder attention:** permite que cada posición en el decodificador atienda a todas las posiciones en la representación generada por el codificador.
- **Feed-forward neural network:** similar a la subcapa del codificador.

Finalmente, el último elemento importante es el **positional encoding**. Debido a que los *transformers* no tienen una estructura recurrente, necesitan una forma de incorporar información sobre el orden de las palabras en la secuencia. Esto se logra mediante la codificación posicional, que agrega información de posición a los *embeddings* de las palabras.

2.2.3. Pre-entrenamiento y *fine-tuning*

Otros conceptos importantes sobre los LLMs son el pre-entrenamiento y el ajuste fino o *fine-tuning*. Por un lado, el pre-entrenamiento es un proceso en el cual un modelo de lenguaje se entrena inicialmente en un gran corpus de datos no etiquetados para aprender representaciones útiles del lenguaje. La idea de esta técnica se basa en que un modelo puede adquirir conocimientos generales del lenguaje y luego ser ajustado para tareas específicas con menos datos etiquetados. El pre-entrenamiento ha demostrado ser efectivo en situaciones donde los datos etiquetados son escasos o costosos de obtener e incluso muestra buenos resultados al usarse a gran escala [39].

Por otro lado, el *fine-tuning* es el proceso mediante el cual un modelo pre-entrenado se ajusta para una tarea específica utilizando un conjunto de datos más pequeño, pero etiquetado. Esto se logra entrenando el modelo en la tarea específica mientras se preservan las representaciones generales aprendidas durante el pre-entrenamiento [40]. Un ejemplo sencillo se puede ver en una red neuronal, donde se podrían re-entrenar únicamente las últimas capas, de manera que se ajuste el modelo sin alterar las propiedades obtenidas por la red en las primeras capas.

2.3. Trabajo relacionado

Una vez sentadas las bases sobre las técnicas de explicabilidad post-hoc y los LLMs en las secciones anteriores, es importante mostrar cómo se han integrado estas metodologías en la investigación actual. Esta sección se centra en estudios que han desarrollado, aplicado o evaluado técnicas post-hoc en el contexto de LLMs, proporcionando una visión general de cómo estas metodologías se integran en el estado del arte actual para mejorar la interpretabilidad y confiabilidad de los modelos de lenguaje modernos.

En primera instancia, existen varios métodos que facilitan la comprensión de algunos modelos de IAX al proporcionar nuevas herramientas a los usuarios finales. En [41], se propone un modelo de lenguaje personalizado denominado “x-[plAIn]”, el cual genera resúmenes claros de métodos como SHAP o LIME. Estos resúmenes están adaptados a diferentes públicos, incluyendo tanto a expertos como a no expertos. En este artículo, se transforman las explicaciones de salida proporcionadas por las técnicas de IAX a lenguaje natural, incorporando descripciones técnicas de los resultados.

Del mismo modo, en [42] se desarrolla “HuntGPT”, un sistema de detección de intrusos que incorpora métodos como SHAP y LIME. Este sistema utiliza un clasificador *Random Forest* [43] entrenado con un conjunto de datos para clasificar y detectar anomalías en el tráfico de red. HuntGPT combina LLMs, con técnicas de IAX para proporcionar explicaciones comprensibles de las amenazas detectadas. La integración de SHAP y LIME permite ofrecer interpretaciones claras sobre las características que contribuyen a la clasificación de una anomalía, facilitando así la comprensión por parte de los analistas de seguridad.

Este tipo de métodos mejoran la usabilidad y la intuitividad del modelo de *machine learning*, y al combinarse con LLMs, los resultados obtenidos se presentan en un formato fácilmente explicable, mejorando la comprensión del usuario. Además, esta unión permite que tanto expertos como usuarios sin formación técnica puedan interpretar y actuar sobre las predicciones del modelo de manera efectiva.

En el estudio [44], utilizan redes neuronales profundas y mapas de calor para la detección de objetos, empleando Grad-CAM. A partir de los atributos de la imagen (localización del objeto, tamaño, color, etc), se genera una leyenda que explica la imagen utilizando una plantilla predefinida. Posteriormente, se formulan preguntas de control a modelos de lenguaje basadas en la leyenda generada. Si la leyenda es correcta, el modelo de lenguaje produce un reporte bien escrito e informativo de explicabilidad.

En concreto, este trabajo propone un sistema que incluye dos módulos principales: contexto y razonamiento. El primer módulo utiliza una técnica basada en plantillas para crear información contextual en texto a partir del mapa de calor y los datos de entrada. El segundo módulo aprovecha un LLM para proporcionar explicaciones en combinación con conocimientos especializados. Este proceso no solo hace que la interpretación de los mapas de calor sea automática y escalable, sino que también la hace interactiva al permitir retroalimentación del usuario. El diagrama del funcionamiento se muestra en la figura 2.11.

Por otro lado, en [45], se investiga la potencia de la IAX en el apoyo a la toma de

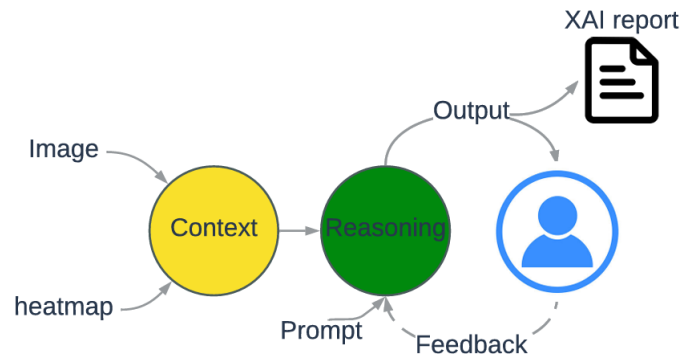


Figura 2.11: Diagrama de ejecución del *framework* [44].

decisiones mediante el uso de LLMs. Este trabajo se centra en el contexto sanitario y emplea modelos de *machine learning* junto con métodos interpretables para identificar características importantes derivadas de los modelos. Además, se utilizan modelos de lenguaje para abordar problemas de clasificación binaria, alimentándolos con las características importantes y atributos de los modelos de *machine learning*, de manera que todo el proceso derive en clasificaciones precisas.

El estudio demuestra cómo ChatGPT <https://chatgpt.com/> puede ser una herramienta poderosa en la toma de decisiones clínicas al utilizar *prompts* contextuales y conocimientos específicos del dominio para mejorar la precisión de las predicciones. Los modelos empleados se compararon con modelos de *machine learning* tradicionales como *Random forest*, regresión logística y XGBoost [46], mostrando mejoras significativas cuando se integraron conocimientos del dominio en el diseño de los *prompts*. Este enfoque permite a ChatGPT igualar e incluso superar el rendimiento de los modelos supervisados tradicionales en escenarios con datos limitados. La ilustración con el flujo de trabajo de este artículo se presenta en la figura 2.12

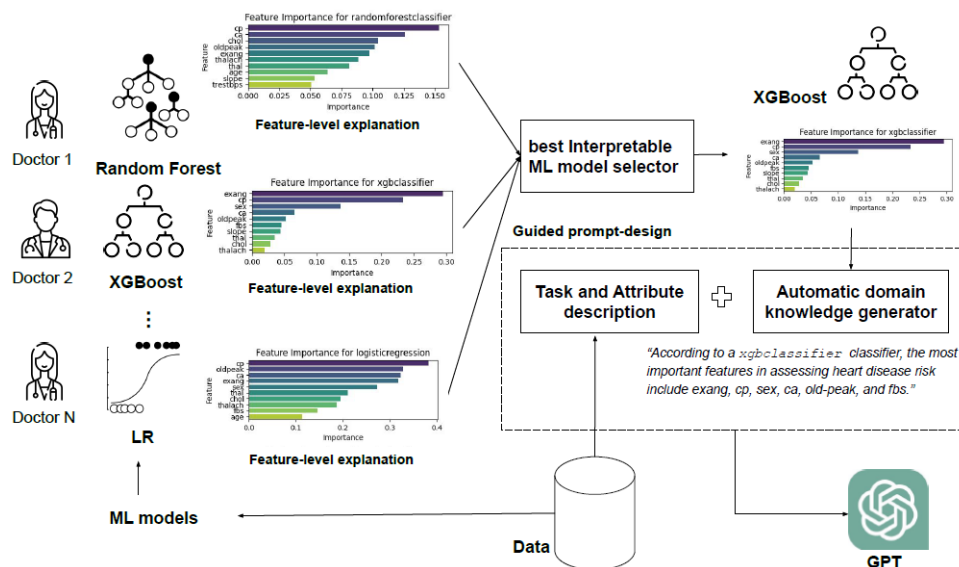


Figura 2.12: Ilustración del flujo de trabajo del artículo [45].

Finalmente, en [47], se destacan las limitaciones de los métodos de XAI, que a menudo son desarrollados por usuarios con un perfil técnico. Para abordar estas limita-

ciones, se propone un marco denominado “XpertAI”, que integra los métodos de XAI con LLMs que acceden a literatura científica, ampliando su base de conocimiento. Esto permite la generación de explicaciones específicas, científicas, interpretables y accesibles en lenguaje natural.

Este marco tiene la capacidad de identificar características importantes dentro de los conjuntos de datos y proporcionar explicaciones enriquecidas y detalladas, basadas en evidencia científica, lo que no solo mejora la interpretabilidad, sino que también hace que estas explicaciones sean accesibles para usuarios no técnicos. Un resumen de los atributos que tienen las explicaciones de XpertAI se observa en la figura 2.13

	Interpretable	Targeted explanations	Literature evidence	Accessible to non-technical users
XAI	✓	✓	✗	✗
LLMs	✓	✗	✗	✓
LLMs + Literature	✓	✗	✓	✓
XpertAI	✓	✓	✓	✓

Figura 2.13: Atributos de las explicaciones de XpertAI [47].

Este trabajo de fin de máster no sólo busca enriquecer o generar explicaciones en lenguaje natural derivadas de las técnicas de inteligencia artificial explicable, sino que también pretende determinar si los grandes modelos de lenguaje pueden considerarse expertos y validar los métodos de XAI. Para evaluar esta hipótesis, es necesario medir la consistencia y veracidad de las respuestas generadas. Con este propósito, se desarrollará un conjunto de métricas, las cuales se exponen en el siguiente capítulo, junto con la metodología y el desarrollo experimental del trabajo.

Capítulo 3

Metodología y desarrollo experimental

En este capítulo se presenta la metodología y el desarrollo experimental del trabajo. En primer lugar, se describirán los modelos de lenguaje empleados, junto con sus versiones y configuraciones específicas, los cuales se encargarán de generar respuestas basándose en técnicas de explicabilidad post-hoc. En segundo lugar, se detallan los *datasets* utilizados junto con sus características. A continuación, para medir las respuestas, se han desarrollado cuatro métricas que miden su consistencia y veracidad. Estas respuestas enriquecerán las explicaciones obtenidas. Además, se evaluará la variación de las respuestas al modificar el número de opciones permitidas. Por último, se presentarán los experimentos completos para los procesos de consistencia y veracidad.

3.1. Modelos empleados

Esta sección presenta los grandes modelos de lenguaje utilizados, comenzando con un ajuste común para todos los modelos: la temperatura. El parámetro de la temperatura controla la aleatoriedad de la salida generada, con un valor más bajo que conduce a salidas más deterministas al favorecer las predicciones más probables. Por lo tanto, en la mayoría de los modelos, el valor de la temperatura se fijó en 0 para minimizar la aleatoriedad. Sin embargo, es importante señalar que para el modelo de Llama2, detallado más adelante, el valor mínimo de temperatura admitido es 0.01. A pesar de esta ligera desviación de 0, el objetivo sigue siendo el mismo: lograr el nivel más bajo posible de aleatoriedad en la salida y conseguir resultados más deterministas.

3.1.1. GPT

OpenAI es una organización de investigación en inteligencia artificial que ha desarrollado varios modelos de lenguaje, donde se incluye la serie GPT (*Generative Pre-trained Transformer*), con modelos como GPT-2 [35] y GPT-3 [36], mencionados anteriormente. Para el desarrollo de este documento, se experimentó con GPT-3.5 y GPT-4 [48]. No obstante, se acabó haciendo uso de **GPT-4**, debido a que esta versión supone un avance significativo con respecto a sus predecesores. GPT-4 amplía el número de pa-

rámetros y la diversidad de sus datos de entrenamiento, lo que mejora su capacidad para generar textos coherentes y contextualmente relevantes.

3.1.2. Gemini

Gemini de Google DeepMind es un competidor clave de OpenAI. Gemini es una familia de modelos construidos sobre *transformers* que emplean mecanismos de atención, similares a GPT-4. Dentro de esta familia, Gemini Pro es el segundo modelo en términos de tamaño y ha sido optimizado para reducir tanto los costos como la latencia, ofreciendo mejoras significativas en rendimiento en diversas tareas [49]. Gemini está diseñado para comprender, razonar y generar resultados a través de múltiples tipos de datos, incluyendo texto. La versión aplicada en el desarrollo ha sido **gemini-1.0-pro-001**.

3.1.3. Llama2

De forma similar, Llama2 constituye otra colección de LLMs pre-entrenados y que hacen uso de *fine-tuning* que se distingue por su naturaleza de código abierto [50]. Este grupo de modelos desarrollados por Meta AI incluye dos variantes principales: Llama2 y Llama2-Chat, disponibles en versiones entrenadas con 7B, 13B y 70B parámetros. Llama2 está diseñado para tareas generales de procesamiento de lenguaje natural, mientras que Llama2-Chat se optimiza para aplicaciones conversacionales. Cada versión se adapta a diferentes necesidades de rendimiento y recursos, con Llama 2-7B siendo eficiente y Llama2-70B ofreciendo una capacidad de procesamiento superior. Para el desarrollo actual se ha hecho uso de **llama-2-70b-chat**.

3.1.4. Mistral

Por último, Mistral AI es una empresa emergente centrada en hacer que la tecnología de inteligencia artificial avanzada sea más accesible y útil, contribuyendo al crecimiento y desarrollo de la IA. Además, incorpora otra importante colección de LLMs, caracterizada por sus avanzadas capacidades de razonamiento y rendimiento. Su modelo de mayor tamaño, “Mistral Large”, obtiene resultados prometedores en el estado del arte actual, incluyendo áreas como el sentido común, inferencia y tareas basadas en el conocimiento [51]. La familia Mistral también incluye modelos de código abierto que superan a variantes de Llama2 en varios *benchmarks* [52]. El modelo específico utilizado en este trabajo ha sido **mistral-large-2402**.

3.2. Conjuntos de datos

En esta sección se detallan los *datasets* utilizados para los experimentos. Es importante mencionar que se parte de estudios previamente realizados. Por un lado, el conjunto de datos empleado para el análisis de la consistencia se ha obtenido mediante técnicas de *machine learning* y explicabilidad, consiguiendo ciertas características importantes. Por otro lado, para el análisis de la veracidad, se ha hecho uso de un *dataset* reconocido en la literatura científica.

3.2.1. Dataset para el análisis consistencia

En un estudio realizado en España, uno de los países europeos más afectados por la pandemia de COVID-19, se describen las características epidemiológicas de pacientes hospitalizados con COVID-19. Este estudio observacional y retrospectivo incluyó a pacientes adultos ingresados en ocho hospitales privados entre el 1 de marzo y el 5 de abril de 2020. Se analizaron un total de 1331 pacientes con COVID-19, de los cuales 841 eran hombres, con una edad media de 66.9 años y una estancia hospitalaria media de 8 días. La mortalidad general observada fue del 17.5 %, aumentando al 36.5 % entre aquellos ingresados en la unidad de cuidados intensivos (UCI). Factores como la edad, el género masculino, los niveles de leucocitos y plaquetas, la saturación de oxígeno, la terapia crónica con esteroides y el tratamiento con hidroxiclороquina/azitromicina se identificaron como características asociadas con la mortalidad [53].

Mediante este estudio que empleó técnicas como el análisis univariante y la regresión de Cox [54], y un estudio que complementa los resultados obtenidos haciendo uso de SHAP para obtener las características importantes [55], se obtuvieron una serie de hipótesis que implicaban un mayor riesgo de muerte por COVID-19: hipertensión, bajo número de plaquetas, recuento alto de leucocitos en situación de emergencia, edad avanzada, género masculino, terapia crónica previa con esteroides, no tratamiento con hidroxiclороquina, saturación de oxígeno en situación de emergencia, no prescripción temprana de lopinavir/ritonavir y no tratamiento con bolo de esteroides.

En consecuencia, se han desarrollado una serie de preguntas siguiendo una estructura coherente para mejorar el desempeño de los modelos de lenguaje. Específicamente, cada pregunta se formula como “Does **hypothesis** mean an increased risk of death from COVID19?”. Esta uniformidad garantiza que las respuestas de los LLMs sean directamente comparables y minimiza la variabilidad que podría surgir al utilizar *prompts* distintos como se menciona en [56]. También permite poner a prueba las hipótesis obtenidas por los modelos de explicabilidad, evaluando las respuestas recibidas por los LLMs.

En la tabla 3.1 se puede observar la batería de preguntas o conjunto de datos utilizado para evaluar la consistencia de las respuestas de los modelos de lenguaje. Es importante mencionar que basándose en el conocimiento del dominio y las explicaciones obtenidas en los estudios se catalogan como respuestas verdaderas Q1, Q2, Q3, Q4, Q5 Q6 y Q8. Por otro lado, Q7, Q9 y Q10 son consideradas como respuestas con controversia, al no estar claramente definido el efecto de aplicar estas hipótesis.

3.2.2. Dataset para el análisis de veracidad

En relación con el *dataset* utilizado para el análisis de la veracidad, se ha recogido un subconjunto de preguntas del *Stanford Question Answering Dataset* (SQuAD) [57]. El SQuAD es un conjunto de datos a gran escala, diseñado para entrenar y evaluar sistemas de comprensión de texto. Es ampliamente utilizado para medir las habilidades de los modelos de lenguaje en este contexto. Este *dataset* consiste en fragmentos de texto extraídos de artículos de la Wikipedia, acompañados de preguntas formuladas por seres humanos. Cada pregunta está asociada con un fragmento de texto y la respuesta es un segmento (subcadena) del fragmento. Este formato permite evaluar la capacidad de un modelo para comprender el contenido del texto y localizar la información relevante.

	Consistency questions
q1	Does hypertension mean an increased risk of death from COVID-19?
q2	Does a low platelet count mean an increased risk of death from COVID-19?
q3	Does a high leukocyte count at emergency mean an increased risk of death from COVID-19?
q4	Does older age mean an increased risk of death from COVID-19?
q5	Does male gender mean an increased risk of death from COVID-19?
q6	Does previous chronic therapy with steroids mean an increased risk of death from COVID-19?
q7	Does not treating with hydroxychloroquine mean an increased risk of death from COVID-19?
q8	Does oxygen saturation at emergency mean an increased risk of death from COVID-19?
q9	Does no early prescription of lopinavir/ritonavir mean an increased risk of death from COVID-19?
q10	Does no treatment with steroid bolus mean an increased risk of death from COVID-19?

Cuadro 3.1: Conjunto de preguntas para la consistencia

En este trabajo, se utiliza el subconjunto de preguntas para validar el conocimiento y la información transmitida por los modelos de lenguaje. De esta manera, es posible determinar la precisión y fiabilidad de las respuestas que devuelven los LLMs en comparación con las respuestas de referencia, las cuales han sido elaboradas por seres humanos.

3.2.3. Prompts

Otro concepto importante son los *prompts*, este término ha sido mencionado en los apartados anteriores y hace referencia a la entrada de texto que se proporciona a un modelo para generar una respuesta. Un *prompt* actúa como un estímulo o una indicación para el modelo, guiando su producción de texto basada en el contenido y la estructura del *prompt* proporcionado.

En esta parte del documento se van a detallar las entradas de texto proporcionadas a los modelos de lenguaje para obtener las salidas deseadas. En primer lugar, como se menciona en los objetivos, se pretende analizar la variación en las respuestas al modificar el número de opciones en las repuestas. De este modo, se diseña el **Contexto 1**, que incorpora una opción binaria con “yes” y “no”, y el **Contexto 2**, que incluye una tercera opción, “controversy”, que representa el debate o la no certeza en la respuesta. Con esta diferencia, se pretende evaluar el desempeño del modelo al modificar la estructura de las preguntas, viendo si las salidas obtenidas son consistentes y veraces a pesar de alterar el número de opciones.

Además, mediante el *prompt* también se guía al modelo de lenguaje a ser preciso y responder como un experto en el COVID-19 o la medicina, con el objetivo de me-

jorar la relevancia y trascendencia de las respuestas. Finalmente, se incorpora una secuencia para que el modelo devuelva una explicación de máximo dos frases que valide las respuestas, de forma que se pueda completar otro objetivo presentado en la introducción: enriquecer las explicaciones obtenidas por los LLMs. Ambos *prompts*, denominados como contextos 1 y 2, junto con la entrada utilizada para evaluar la veracidad, nombrada como **Contexto 3**, son presentados en la tabla 3.2.

	Prompt
Context 1	You are an expert on COVID-19 and your duty is to answer questions related to the topic only with yes or no followed by the explanation that validates the answer in a maximum of 2 sentences.
Context 2	You are an expert on COVID-19 and your duty is to answer questions related to the topic only with yes, no or controversy followed by the explanation that validates the answer in a maximum of 2 sentences.
Context 3	You are a medical expert and your duty is to answer medical questions in a single sentence in a precise and brief manner.

Cuadro 3.2: Contextos utilizados en la evaluación de la consistencia y la veracidad

Es importante resaltar que, los modelos de lenguaje tienen un parámetro denominado “max_tokens” que permite modificar la cantidad máxima de *tokens* devueltos en las respuestas. Sin embargo, tras experimentar con este parámetro, se obtuvo que los modelos cortaban las respuestas repentinamente. Debido a esto, se optó por introducir en el *prompt* la longitud que debían tener las respuestas.

3.3. Métricas

El siguiente paso consiste en aplicar una colección de métricas para evaluar la consistencia y la veracidad de los modelos de lenguaje. Esta colección incluye la similitud semántica y el solapamiento entre textos, además de las medidas ROUGE y BLEU.

3.3.1. Similitud semántica

En primer lugar, se tiene la similitud semántica. Esta métrica considera cómo de parecidos son dos fragmentos de texto en lo referente a su significado o contenido, más allá de las palabras exactas que utilizan. Es una forma de determinar si dos textos expresan ideas similares, aunque no necesariamente usen las mismas palabras o estructura gramatical. Para esto, los textos se representan como vectores en un espacio multidimensional (los *embeddings* mencionados en el capítulo anterior). Sin embargo, los *embeddings* son representaciones de las palabras, por lo que en este trabajo se hace uso de *sentence embeddings* para representar frases y textos. De manera análoga, las representaciones de estos vectores que se encuentren próximas entre sí en el espacio multidimensional corresponden a textos con una alta similitud semántica.

Existen estudios [58] que analizan gran variedad de métodos para calcular la similitud semántica. Por ello, en este estudio se va a hacer uso de *sentence embeddings*

junto con la similitud del coseno para calcular la distancia entre vectores. Concretamente, se hará uso de Sentence-BERT [59], una variación de BERT [34] optimizada para representaciones vectoriales de textos. Sentence-BERT contiene una amplia colección de modelos que se diferencian en la precisión y la eficiencia que aportan. Este estudio ha empleado el modelo “**all-MiniLM-L6-v2**”, un modelo que, a pesar de ser pequeño en términos de tamaño, en comparación con otras versiones, destaca por su rendimiento y velocidad, dando buenos resultados.

3.3.2. Solapamiento

La segunda métrica es el solapamiento, que en este contexto hace referencia a la medida en que dos textos comparten elementos comunes como símbolos, palabras o frases (*tokens*). En este trabajo, se mide el solapamiento a partir de los *tokens* que aparecen en dos frases. De esta manera, se determina cuánto contenido compartido existe entre ambas, lo que complementa la métrica de similitud semántica.

La implementación consiste en: dadas dos frases, contar el número de tokens comunes entre ambas y, seguidamente, dividir este valor por el menor número total de palabras en las dos respuestas comparadas. Este procedimiento se basa en la idea de que dividir por el menor número de palabras ajusta la métrica para que refleje más adecuadamente la proporción de solapamiento en relación con la respuesta más corta.

Como ejemplo, se muestra una justificación en la tabla 3.3. Las palabras en negrita son coincidentes en ambas frases y, al dividir por el número de tokens de la primera frase, se obtiene un solapamiento del 100 % (5 tokens comunes entre 5 tokens de la frase más corta), lo cual es más representativo al cubrirse toda la frase. Si se dividiese por el número de tokens de la frase más larga se obtendría un 62.5 %, lo cual puede subestimar la cantidad de solapamiento significativo en contextos donde las frases tienen diferentes longitudes, como en este ejemplo.

El	gato	negro	duerme	tranquilo			
El	gato	negro	por	la	noche	duerme	tranquilo

Cuadro 3.3: Ejemplo de la métrica de solapamiento

3.3.3. ROUGE

La siguiente métrica es ROUGE, (*Recall-Oriented Understudy for Gisting Evaluation*) [60]. Esta métrica es utilizada para evaluar la calidad de resúmenes automáticos comparándolos con resúmenes de referencia (normalmente creados por humanos). Esta métrica también se emplea en la evaluación de la generación de texto y traducción automática. ROUGE mide la coincidencia entre las subsecuencias y las palabras de los resúmenes generados con los de referencia.

Existen diversas variantes de ROUGE. En este trabajo, se implementa ROUGE-L, una variante que mide la longitud de la subsecuencia común más larga (*Longest Common Subsequence*, LCS) entre dos textos. Esta métrica complementa el solapamiento al capturar la coherencia a nivel de frase y la estructura de la oración, ya que la anterior métrica no tenía en cuenta el orden de las palabras. A continuación, se muestran las fórmulas de ROUGE-L para el recall, la precisión, y F_1 score, siendo m el número de

tokens del texto M , n el número de tokens del texto N , y $LCS(M, N)$ la longitud de la subsecuencia más larga entre M y N :

$$\text{ROUGE-L}_{\text{Recall}} = \frac{LCS(M, N)}{m} \quad (1)$$

$$\text{ROUGE-L}_{\text{Precisión}} = \frac{LCS(M, N)}{n} \quad (2)$$

$$\text{ROUGE-L}_{F_1} = \frac{2 \cdot \text{ROUGE-L}_{\text{Recall}} \cdot \text{ROUGE-L}_{\text{Precisión}}}{\text{ROUGE-L}_{\text{Recall}} + \text{ROUGE-L}_{\text{Precisión}}} \quad (3)$$

3.3.4. BLEU

La última métrica que se implementa en este trabajo es BLEU (*Bilingual Evaluation Understudy*) [61]. BLEU se utiliza para evaluar la calidad de texto generado automáticamente, particularmente en tareas de traducción automática. Esta métrica calcula la precisión de n -gramas (secuencias contiguas de n elementos de un texto) para diferentes niveles de n y los combina con una media geométrica ponderada para obtener una medida general de similitud. Además, incluye una penalización por brevedad para evitar favorecer textos más cortos que otros. La fórmula general de BLEU es:

$$\text{BLEU} = BP \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right) \quad (4)$$

donde BP es la penalización por brevedad, w_n son los pesos asignados a cada nivel de n -gramas (generalmente iguales), y p_n es la precisión modificada para los n -gramas de longitud n .

La fórmula de la penalización por brevedad es:

$$BP = \begin{cases} 1 & \text{si } c > r \\ e^{(\frac{1-r}{c})} & \text{si } c \leq r \end{cases} \quad (5)$$

donde c es la longitud de un texto y r es la longitud del texto del otro texto comparado.

En resumen, BLEU complementa al resto de métricas al utilizar una media geométrica ponderada de precisiones de n -gramas de diferentes longitudes, lo que le permite capturar mejor la fluidez y coherencia general del texto generado. Además, incorpora la penalización por brevedad, que asegura no solo que los textos sean precisos a nivel de palabras coincidentes, sino también completos en cuanto a su longitud.

3.3.5. Ejemplo de métricas

De manera complementaria, se van a exponer varios ejemplos de métricas para comparar la diferencia que suponen valores de métricas más altos o más bajos que otros. Este apartado ofrece una visión más específica de las métricas y su implicación al comparar textos. Por otro lado, en el próximo capítulo, se podrán observar ejemplos reales de los textos generados y el análisis de las métricas obtenidas.

En primer lugar, se tiene el caso ideal u óptimo, un modelo de lenguaje que al generar respuestas retorna absolutamente todas las salidas iguales, obteniendo el máximo valor posible y una consistencia máxima.

Semantic similarity	Overlap	ROUGE	BLEU
1,000	1,000	1,000	1,000

Cuadro 3.4: Ejemplo de métricas elevadas

Un ejemplo de similitud semántica menor respecto al resto de métricas, puede implicar el uso de un vocabulario y organización del texto similar, pero un contexto distinto. En las frases “El gato come sentado en la rama de un árbol” y “El perro come sentado en la rama de un árbol”, tienen unos valores de solapamiento, ROUGE y BLEU superiores respecto a la similitud semántica:

Semantic similarity	Overlap	ROUGE	BLEU
0,751	0,900	0,900	0,850

Cuadro 3.5: Ejemplo de métricas 2: similitud semántica menor al resto

Por otro lado, cuando se tiene un bajo valor de solapamiento, esto implica que las métricas de ROUGE y BLEU también serán bajas. Dicho valor se debe al funcionamiento interno de estas métricas, basado en la coincidencia de tokens. Por tanto, se muestra una comparativa de lo que implican los valores de ROUGE y BLEU bajos respecto al resto.

Primero, dos frases como “El gato come sentado en la rama de un árbol” y “En la rama de un árbol el gato come sentado” tienen un bajo valor de ROUGE, de 0,5, debido a que la secuencia más larga de tokens es alrededor de la mitad de la frase. Sin embargo, se tiene un valor elevado de BLEU ya que se comparan n -gramas.

Semantic similarity	Overlap	ROUGE	BLEU
0,986	1,000	0,500	0,912

Cuadro 3.6: Ejemplo de métricas 3: ROUGE menor al resto

Después, dos frases como “El gato come sentado en la rama de un árbol” y “El gato come sentado en la rama de un árbol. Luego duerme tranquilo” obtienen un valor de BLEU bajo, principalmente porque se penaliza la diferencia de longitud de textos. Por el contrario, ROUGE obtiene un 0,833 al tener una secuencia de caracteres común y relativamente larga en ambos textos.

Semantic similarity	Overlap	ROUGE	BLEU
0,756	1,000	0,833	0,535

Cuadro 3.7: Ejemplo de métricas 4: BLEU menor al resto

Por último, dos frases que no tienen nada que ver obtendrán valores muy bajos en las métricas. Como ejemplo, se tienen las frases “El gato come sentado en la rama de

un arbol” y “Ocho más seis son catorce ” con los siguientes valores:

Semantic similarity	Overlap	ROUGE	BLEU
0,31	0,000	0,000	0,000

Cuadro 3.8: Ejemplo de métricas 5: todas las métricas con valores reducidos

3.4. Implementación

Para concluir este capítulo, se presenta la sección de implementación. En esta sección se van a detallar los experimentos que se aplican para generar las respuestas de los LLMs y evaluarlas. En ellos, se tiene en cuenta cada modelo, los conjuntos de datos adecuados y las métricas correspondientes.

La **idea principal** de este trabajo de fin de máster es validar las características importantes obtenidas mediante técnicas de explicabilidad post-hoc como SHAP con la ayuda de un experto. Se parte de la propuesta de que los grandes modelos de lenguaje se podrían considerar expertos, lo que plantea problemas relacionados con la robustez y la fiabilidad de los LLMs en sí mismos. Estos atributos pueden depender de los parámetros (temperatura, el *prompt*, etc.) de los modelos de manera independiente o incluso de las diferentes respuestas que provienen de distintos modelos para una misma pregunta.

En primer lugar, para que un modelo de lenguaje puede considerarse experto debe ser consistente y veraz. La **consistencia** es la capacidad del modelo para retornar respuestas coherentes y uniformes cuando se le presentan preguntas similares o repetidas. Por otro lado, la **veracidad** se refiere a la exactitud y fidelidad de la información proporcionada por el modelo en sus respuestas. Esto implica que las respuestas generadas deben ser correctas y basadas en hechos verificables.

En consecuencia, se plantean dos experimentos para medir estos atributos. Los dos métodos pueden distinguirse por la disponibilidad del *ground truth*. En el primer método, el experimento de análisis de la consistencia, la respuesta verdadera no está disponible; por ello, esta métrica se evalúa comparando las respuestas de un modelo entre sí. En el segundo método, el experimento de análisis de la veracidad, la respuesta verdadera sí existe, por lo que el modelo se compara con este *ground truth* para medir la precisión y fiabilidad en devolver respuestas verídicas.

Además, mediante el uso de preguntas que incluyen las características importantes (se recuerda la estructura de la tabla 3.1 “Does **hypothesis** mean an increased risk of death from COVID19?”) y los contextos adecuados, se obtienen respuestas que enriquecen las técnicas de explicabilidad y se reduce el espacio entre la inteligencia artificial explicable y las aplicaciones prácticas en el mundo real. Asimismo, al configurar el *prompt* para que los modelos devuelvan “yes”, “no” o “controversy”, se logra el objetivo inicial de validar las respuestas.

3.4.1. Experimento de análisis de la consistencia

El experimento de análisis de la consistencia es independiente de cada modelo de lenguaje, es decir, se aplica para cada uno de los modelos. Este experimento ejecuta

un algoritmo, representado en 1, que comienza con el conjunto de preguntas de la tabla 3.1 explicado en la sección de conjuntos de datos (*dataset1* en la representación). Para cada una de estas preguntas (q_i) se envía como entrada la pregunta y el contexto correspondiente (Contexto 1 o Contexto 2 de la tabla 3.2) a uno de los LLMs. Este procedimiento se realiza 10 veces para posteriormente medir la consistencia de las respuestas devueltas por la misma pregunta. Una vez acaba la ejecución y se poseen las 10 respuestas de una misma pregunta, se calculan las 4 métricas: similitud semántica, solapamiento, ROUGE y BLEU. Estas métricas son calculadas para cada par de respuestas, por lo que finalmente se tendrán 100 valores, al calcular la métrica de cada una de las 10 preguntas con el resto de ellas. Por último, se calcula la media de los 100 valores y se devuelve un valor único. Este último proceso es interno de cada métrica.

Algorithm 1 Evaluate the consistency of a single LLM

```

1: for each question  $q_i$  in dataset1 do
2:   Initialize Responses to an empty list
3:   for  $i \leftarrow 1$  to 10 do
4:     response  $r \leftarrow \text{AskLLM}(q_i, \text{context1})$ 
5:     Append  $r$  to Responses
6:   end for
7:   SemanticSimilarity  $\leftarrow \text{CalculateSemanticSimilarity}(\text{Responses})$ 
8:   Overlap  $\leftarrow \text{CalculateOverlap}(\text{Responses})$ 
9:   ROUGE  $\leftarrow \text{CalculateROUGE}(\text{Responses})$ 
10:  BLEU  $\leftarrow \text{CalculateBLEU}(\text{Responses})$ 
11:  Store metrics for further analysis
12: end for

```

3.4.2. Experimento de análisis de la veracidad

Por otro lado, el experimento de análisis de la veracidad también es independiente de cada modelo de lenguaje. Este experimento ejecuta otro algoritmo, representado en 2, que sigue un procedimiento similar al algoritmo de consistencia, pero con algunas variaciones. En primer lugar, el conjunto de datos utilizados es un subconjunto de 10 preguntas aleatorias del SQuAD (*dataset2* en la representación) y para cada pregunta se envía al LLM como entrada la pregunta y en este caso, el Contexto 3 de la tabla 3.2. El cambio principal en este algoritmo, es que a diferencia del anterior, que compara las respuestas devueltas por el modelo con ellas mismas, el algoritmo de veracidad compara las respuestas devueltas con el *ground truth* (representado como q_i *answer*) del *dataset* SQuAD, evaluando la capacidad de los modelos para generar respuestas correctas.

3.4.2.1. Extensión del experimento

Adicionalmente, se ha realizado un experimento para ampliar la evaluación de la veracidad en un LLM específico: GPT-4. Esta extensión se ha desarrollado mediante una aplicación de OpenAI, ChatGPT (<https://chatgpt.com/>). La razón de no utilizar la API, como se hizo con el resto de la implementación, es que la funcionalidad requerida no está disponible vía API. Dentro de ChatGPT, se ha incorporado una funcionalidad que permite a los usuarios crear GPTs personalizados [62], los cua-

Algorithm 2 Evaluate the veracity of a single LLM

```
1: for each question  $q_i$  in dataset2 do
2:   Initialize Responses to an empty list
3:   for  $i \leftarrow 1$  to 10 do
4:      $response\ r_i \leftarrow \text{AskLLM}(q_i, \text{context3})$ 
5:     Append  $r_i$  to Responses
6:   end for
7:    $\text{SemanticSimilarity} \leftarrow \text{CalculateSemanticSimilarity}(\text{Responses}, q_i \text{ answer})$ 
8:    $\text{Overlap} \leftarrow \text{CalculateOverlap}(\text{Responses}, q_i \text{ answer})$ 
9:    $\text{ROUGE} \leftarrow \text{CalculateROUGE}(\text{Responses}, q_i \text{ answer})$ 
10:   $\text{BLEU} \leftarrow \text{CalculateBLEU}(\text{Responses}, q_i \text{ answer})$ 
11:  Store metrics for further analysis
12: end for
```

les mejoran su desempeño en tareas específicas. En particular, existen GPTs como Consensus [63] que integran GPT-4 con millones de artículos científicos, obteniendo respuestas con un *ground truth* verídico.

Esta extensión del experimento utiliza el GPT Consensus debido a la baja precisión de ChatGPT (aplicando GPT-4) en la provisión de referencias precisas y acertadas [64]. Además, una limitación de esta investigación es que no se puede medir de forma eficiente las respuestas, ya que el proceso debe realizarse a través de la interfaz gráfica de ChatGPT, lo que impide modificar la temperatura y evaluar muchas respuestas de manera eficaz.

Como se mencionaba anteriormente, la veracidad se refiere a la exactitud y fidelidad de la información proporcionada por un modelo en sus respuestas, y es importante comprobar que están basadas en hechos verídicos. La idea principal de esta extensión es verificar que la información devuelta en las respuestas de un modelo esté respaldada por un conocimiento ya establecido. En la figura 3.1 se muestra un diagrama del funcionamiento de este experimento:

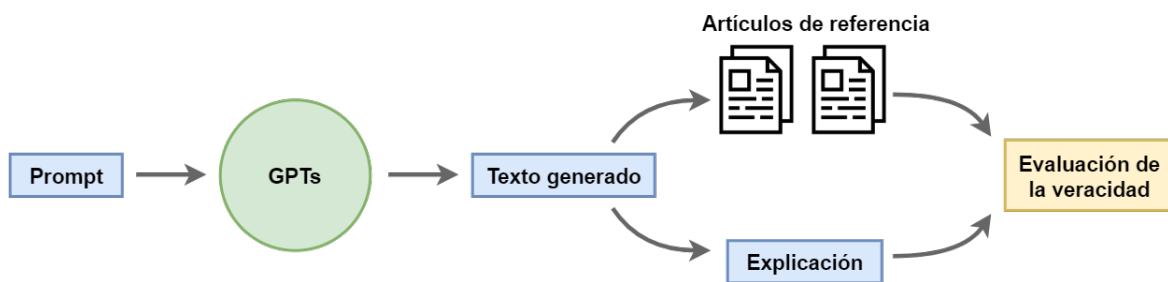


Figura 3.1: Diagrama de ejecución de la extensión del experimento.

En primer lugar, se ha diseñado un *prompt* que incorpora una pregunta de ámbito general y una especificación para obtener una salida relevante. El *prompt* utilizado ha sido “How does photosynthesis work in plants? Give well-founded answers and the references that give the evidence of the response”.

A continuación, el modelo devuelve un texto que explica la pregunta realizada; esta respuesta concreta se divide en cuatro párrafos, cada uno incluyendo una referencia a un artículo. El siguiente paso es evaluar si la información en estos textos gene-

rados por el modelo de lenguaje es correcta, comparando cada explicación con el texto completo del artículo para encontrar las frases o párrafos más coincidentes. Esta comparación se realiza utilizando las métricas mencionadas. Una vez dividido el artículo en párrafos, se obtiene el párrafo con mayor valor de las métricas. Finalmente, se analiza este párrafo y se concreta si la información generada en un primer momento era verídica y acertada.

Capítulo 4

Análisis de resultados

En este capítulo, se exponen y analizan los distintos resultados obtenidos tras la experimentación y el desarrollo, descritos en el capítulo anterior. Esta parte del documento se ha dividido en dos secciones, cada una correspondiente a un experimento específico para analizar la consistencia y la veracidad de los modelos de lenguaje.

4.1. Resultados del experimento de análisis de consistencia

En primer lugar, tras realizar el experimento de análisis de consistencia con todos los modelos, se obtiene una lista de 10 respuestas para cada pregunta del *dataset* de consistencia. Además, el experimento se ejecuta tanto con el Contexto 1 como con el Contexto 2, para analizar la variación en las respuestas al modificar el número de opciones posibles (“yes” y “no” frente a “yes”, “no” y “controversy”). Esta variación se puede observar en la tabla 4.1, donde se muestra la respuesta esperada para cada pregunta y la variación de las respuestas para cada modelo, al permitir la opción binaria y la opción que incluye controversia.

	Expected	GPT		Mistral		Gemini		Llama2	
		Context 1	Context 2	Context 1	Context 2	Context 1	Context 2	Context 1	Context 2
g1	yes	yes	yes	yes	yes	yes	controversy	yes	yes
g2	yes	yes	yes	yes	yes	yes	controversy	no	controversy
g3	yes	yes	yes	yes	yes	yes	controversy	yes	controversy
g4	yes	yes	yes	yes	yes	yes	yes	yes	yes
g5	yes	yes	yes	yes	yes	yes	yes	no	controversy
g6	yes	yes	controversy	yes	controversy	no	controversy	yes	controversy
g7	controversy	no	controversy	no	controversy	no	controversy	no	controversy
g8	yes	yes	yes	yes	yes	yes	yes	yes	controversy
g9	controversy	no	controversy	yes	controversy	no	controversy	yes	controversy
g10	controversy	yes	controversy	yes	controversy	no	controversy	yes	controversy

Cuadro 4.1: Variación en las respuestas al utilizar los contextos 1 y 2

En la tabla se observa que cada celda presenta un valor único. Esto se debe a que las respuestas siguen el formato descrito en el *prompt*, obteniendo una salida del tipo **[validación, explicación]**, donde la validación puede ser “yes”, “no” o “controversy”, y la explicación es el texto que acompaña la validación y la enriquece con información adicional. En las celdas de la tabla se muestra únicamente la validación para analizar claramente la variación por el número de opciones en el *prompt*. Cabe destacar que, aunque se generan 10 respuestas para cada pregunta, el valor de la validación no

4.1. Resultados del experimento de análisis de consistencia

varía entre iteraciones de la misma pregunta, lo que explica que cada celda contenga un único valor y no 10.

De forma óptima, los LLMs deben realizar tres variaciones justificadas (Q7, Q9 y Q10) en las salidas generadas, ya que con el primer contexto solo tienen la opción de sí o no y deben elegir. Sin embargo, algunos modelos de lenguaje realizan variaciones no justificadas (e.g. Gemini Q1, Llama2 Q2) al generar respuestas con la opción de controversia. Específicamente se tiene que:

- **GPT:** produce 3 variaciones precisas (Q7, Q9 y Q10), pero se clasifica la pregunta Q6 como controversia. Este cambio es erróneo y resulta en 9/10 respuestas correctas al modificar el contexto.
- **Mistral:** de manera similar a GPT, realiza 3 variaciones precisas (Q7, Q9 y Q10), pero genera una variación injustificada, Q6. Se diferencia del anterior modelo al validar la pregunta Q9 como “yes” a diferencia de GPT con “no”. Esta validación no es significativa al permitir únicamente la opción binaria con el contexto 1, dejando al modelo que elija entre sí o no para una respuesta con controversia.
- **Gemini:** genera 3 variaciones correctamente (Q7, Q9 y Q10), pero destacan 4 variaciones injustificadas (Q1, Q2, Q3 y Q6). Además, este modelo de lenguaje difiere en el resultado de la pregunta Q6, donde se espera “yes” pero se obtiene “no”.
- **Llama2:** efectúa las 3 variaciones necesarias (Q7, Q9 y Q10), pero se equivoca con creces en las preguntas Q2, Q3, Q5, Q6, y Q8 con variaciones incorrectas. Similarmente a Gemini, este LLM produce discrepancias en las respuestas. Las preguntas Q2 y Q5 deben clasificarse como “yes”, sin embargo, la salida obtenida es “no”.

Estos resultados sugieren que la incorporación en el *prompt* de una opción adicional en las opciones de las respuestas tiene un impacto significativo en el comportamiento de los modelos de lenguaje. Esta modificación causa cambios notables entre las salidas generadas, mostrando alteraciones en las respuestas de GPT y Mistral en 4 de 10 casos, en Gemini en 7 de 10 casos y en Llama2 en 8 de 10 casos, sugiriendo una preferencia importante al elegir controversia frente a un sí o no definitivo.

No obstante, para GPT y Mistral, el 75 % (3 de 4 cambios justificados) de las alteraciones son justificadas, mejorando la precisión de las respuestas. Este número desciende al 42,8 % (3 de 7 cambios justificados) para Gemini y 37,5 % (3 de 8 cambios justificados) para Llama2, indicando una variabilidad en la calidad de las justificaciones de los modelos al enfrentarse a opciones adicionales. Estos datos sugieren que mientras GPT y Mistral tienden a mantener una mayor coherencia y precisión en sus respuestas modificadas, Gemini y Llama2 muestran una mayor susceptibilidad a la confusión o a interpretaciones menos precisas cuando se les presentan alternativas más complejas. Este hallazgo resalta la importancia de considerar la estructura de los *prompts* y las opciones de respuesta.

4.1.1. Métricas

A continuación, se presentan los resultados numéricos de la evaluación de la consistencia de las respuestas, basados en las métricas. Se recuerda que la estructura de una respuesta es del tipo **[validación, explicación]**, donde la validación puede

Análisis de resultados

ser “yes”, “no” o “controversy”, y la explicación es el texto que acompaña la validación y la enriquece con información adicional. En consecuencia, esta subsección del documento evalúa las métricas (similitud semántica, solapamiento, ROUGE y BLEU) para cada explicación. De esta forma, es posible analizar la consistencia entre las respuestas de los modelos de lenguaje.

En la tabla 4.2 se muestran los resultados obtenidos de las métricas para cada LLM con cada contexto. Esta tabla es un resumen de todos los valores generados, al presentar únicamente la media de la consistencia para todas las respuestas de un modelo de lenguaje en ambos contextos. En esta tabla se recogen los resultados medios de las tablas 1 y 2 del anexo.

		Semantic similarity	Overlap	ROUGE	BLEU
GPT	Context 1	0,983	0,888	0,844	0,770
	Context 2	0,980	0,897	0,853	0,763
Mistral	Context 1	1,000	1,000	1,000	1,000
	Context 2	0,999	0,995	0,992	0,988
Gemini	Context 1	1,000	0,996	0,996	0,992
	Context 2	0,999	0,996	0,993	0,986
Llama2	Context 1	1,000	0,998	0,997	0,996
	Context 2	0,999	0,991	0,989	0,983

Cuadro 4.2: Evaluación media de la consistencia

Los valores presentados en la tabla anterior sugieren una diferencia mínima al comparar la consistencia entre las respuestas de un mismo modelo de lenguaje, en los contextos 1 y 2 (e.g. Diferencia mínima en Gemini; Contexto 1 [1,000 0,996 0,996 0,992] frente a Contexto 2 [0,999 0,996 0,993 0,986]).

Por un lado, se tienen los valores de las métricas para los modelos Mistral, Gemini y Llama2, donde todos los valores son 1 o muy cercanos a 1. Esto sugiere que las respuestas son altamente similares y, por lo tanto, consistentes. Específicamente, estos valores tan cercanos a 1 son debido a la modificación de palabras sueltas en las respuestas, utilizando sinónimos, entre otros. Como ejemplo, en la tabla 4.3, aparece un ligero cambio en dos respuestas que ocasiona que las métricas no alcancen el valor óptimo de consistencia: 1. Estas respuestas son para la pregunta Q10 de Mistral con el contexto 2, y se puede ver cómo afecta a los valores medios de las métricas si se analiza la tabla 2 del anexo. En negrita se muestra la diferencia que contribuye a ocasionar la leve disminución en las métricas de la tabla 4.2.

Por otro lado, los valores de las métricas de GPT se distinguen frente al resto de modelos. A pesar de obtener valores de similitud semántica altos (0,983 y 0,980) el resto de métricas son inferiores. Esto se puede atribuir a que no todas las respuestas utilizan exactamente las mismas palabras o tienen variaciones al hacer uso de sinónimos o frases alternativas, lo que reduce el solapamiento. ROUGE (0,844 y 0,853) sugiere que hay diferencias leves en fragmentos de texto entre las respuestas, ya que la variación en la estructura de las oraciones puede reducir la puntuación ROUGE. BLEU (0,770 y 0,763) es la métrica con la puntuación más baja, lo que sugiere que hay más variaciones en las secuencias de palabras. La longitud de las respuestas y la elección de palabras específicas pueden variar.

4.1. Resultados del experimento de análisis de consistencia

<i>Ejemplos de respuesta de Mistral en la Q10 con el Contexto 2</i>
Controversy: The impact of steroid bolus treatment on COVID-19 outcomes is still under investigation. Some studies suggest that steroids can reduce mortality in severely ill patients, while others indicate potential harm, such as delayed viral clearance or secondary infections. The optimal timing, dosage, and patient selection for steroid treatment in COVID-19 remain unclear.
Controversy: The impact of steroid bolus treatment on COVID-19 outcomes is still under investigation. Some studies suggest that steroids may reduce mortality in severely ill patients, while others indicate potential harm, such as delayed viral clearance or secondary infections. The optimal timing, dosage, and patient selection for steroid treatment in COVID-19 remain unclear.

Cuadro 4.3: Ejemplos de respuestas consistentes en Mistral

En la tabla 4.4 se muestran ejemplos de respuestas para la pregunta Q5 utilizando GPT y el contexto 1. Se recuerda que estas son solo dos de las diez respuestas generadas para Q5. En la tabla se puede observar la diferencia que existe entre ambas. Los valores exactos de consistencia para las 10 respuestas de Q5 con GPT en el contexto 1 se pueden ver en la tabla 1 del anexo (Valores de las métricas : [0,981 0,895 0,841 0,738]).

<i>Ejemplos de respuesta en la Q5 de GPT para el Contexto 1</i>
Yes, studies have shown that men are at a slightly higher risk of severe illness and death from COVID-19 than women. This could be due to a variety of factors, including differences in immune response and lifestyle factors such as higher rates of smoking and certain underlying health conditions in men.
Yes, studies have shown that men are at a slightly higher risk of severe illness and death from COVID-19 than women. This could be due to a variety of factors, including biological differences and lifestyle factors.

Cuadro 4.4: Ejemplos de respuestas poco consistentes en GPT

Las dos preguntas mostradas en la tabla 4.4 presentan un solapamiento de casi el 100%. Sin embargo, la estructura de la última frase cambia, siendo más largo el primer texto, lo que resulta en un valor inferior de ROUGE debido a la diferencia en la estructura y de BLEU debido a la penalización por la diferencia de longitud. En negrita se muestran las palabras de texto que se solapan entre ambos textos

En conclusión, los valores de las métricas sugieren que los modelos Mistral, Gemini y Llama2 son altamente consistentes de forma interna, al producir respuestas prácticamente iguales en cada pregunta. Por otra parte, GPT, produce respuestas menos consistentes que el resto de modelos. A pesar de esto, las métricas de GPT sugieren que las respuestas son relativamente similares entre ellas (Similitud semántica alta: 0,983 y 0,980), a pesar de utilizar palabras distintas, estructuras de frases diferentes y longitudes de respuestas no iguales (Solapamiento, ROUGE y BLEU inferiores).

4.1.2. Resultados adicionales de consistencia

Adicionalmente, se han analizado más resultados a partir de los datos anteriores. En primer lugar, se ha realizado una comparativa entre las respuestas generadas con el

Análisis de resultados

contexto 1 frente a las respuestas generadas con el contexto 2. De esta manera, se puede estudiar la diferencia en la consistencia al modificar número de opciones en las respuestas.

En la tabla 4.5 se presentan los resultados mencionados. Destacan los bajos valores obtenidos con las métricas, que sugieren que las respuestas generadas por el contexto 1 y el contexto 2, a pesar de ser consistentes en sí mismos, no son muy consistentes cuando se comparan entre ellos. Las métricas indican que, para todos los modelos, apenas algo más de la mitad de las palabras se solapan entre frases (Solapamientos: 0,659 0,673 0,600 0,552). Además, los bajos valores de las métricas ROUGE y BLEU señalan una gran diferencia de estructura en las respuestas. Por otro lado, aunque se obtengan valores entre 0,905 y 0,931 de similitud semántica, esto solo indica que los textos están altamente relacionados semánticamente y que generan respuestas sobre el mismo tema, pero no implica que sean iguales. Esta conclusión se obtiene al analizar el conjunto de todas las métricas.

	Semantic similarity	Overlap	ROUGE	BLEU
GPT	0,916	0,659	0,541	0,375
Mistral	0,931	0,673	0,570	0,389
Gemini	0,915	0,600	0,442	0,247
Llama2	0,905	0,552	0,411	0,219

Cuadro 4.5: Consistencia media al enfrentar el contexto 1 y 2

De manera complementaria, se ha desarrollado una tabla (Tabla 4.6) que muestra la diferencia en la consistencia al cambiar entre el contexto 1 y el contexto 2. En otras palabras, esta tabla indica la pérdida de consistencia entre contextos. Los valores se obtienen calculando la media de las métricas (similitud semántica, solapamiento, ROUGE y BLEU) entre contextos de la tabla 4.2. Después, a este valor se le resta el valor de consistencia media al enfrentar contextos (tabla 4.5) y se divide entre el primer valor mencionado. Es decir, el primer valor que se muestra en la primera celda de la tabla, ha sido obtenido mediante la fórmula $(0,981 - 0,916)/0,981$ que es igual a una pérdida del 6,63 %

	Semantic similarity loss	Overlap loss	ROUGE loss	BLEU loss
GPT	6,63 %	26,21 %	36,31 %	51,10 %
Mistral	6,90 %	32,50 %	42,79 %	60,84 %
Gemini	8,47 %	39,75 %	55,52 %	75,03 %
Llama2	9,46 %	44,52 %	58,65 %	77,85 %

Cuadro 4.6: Pérdida en las métricas al comparar respuestas con contexto 1 y 2

Con estos resultados se concluye que GPT es el modelo que menos modifica sus respuestas al comparar los resultados obtenidos con distintos *prompts*, utilizando la opción de respuesta binaria y la opción que incluye controversia. GPT obtiene la menor pérdida de consistencia, mientras que Llama2 muestra discrepancias entre respuestas con contextos distintos al mostrar la mayor pérdida de consistencia (9,46 % 44,52 % 58,65 % 77,85 %).

Por último, se expone el número medio de palabras por explicación, tanto en el texto completo como en las frases que lo componen. Estos resultados se presentan en

4.1. Resultados del experimento de análisis de consistencia

la tabla 4.7. Llama2 genera de media las explicaciones más largas (≈ 54 palabras por texto) y complejas sintácticamente (≈ 25 palabras por frase). Mistral genera las explicaciones más largas después de Llama2 (≈ 37 palabras por texto), sin embargo, genera explicaciones sintácticamente menos complejas que GPT, al tener de media frases de 20,91 palabras frente a las 21,17 palabras por frase de GPT. Gemini produce las respuestas más cortas (≈ 37 palabras por texto) y las menos complejas (≈ 20 palabras por frase) de todos los modelos de lenguaje.

		Avg words per text	Avg words per sentence
GPT	Context 1	39,38	19,69
	Context 2	37,46	22,65
	Average	38,42	21,17
Mistral	Context 1	53,30	21,62
	Context 2	45,70	20,20
	Average	49,5	20,91
Gemini	Context 1	37,70	18,85
	Context 2	35,74	20,98
	Average	36,72	19,91
Llama2	Context 1	57,70	23,20
	Context 2	50,76	27,69
	Average	54,23	25,44

Cuadro 4.7: Número medio de palabras por explicación

4.1.3. Enriquecimiento

Ejemplos de respuesta con Gemini y contexto 1	
Q1	Yes, hypertension is associated with an increased risk of death from COVID-19. People with hypertension have a higher risk of developing severe complications from COVID-19, such as pneumonia, acute respiratory distress syndrome (ARDS), and death.
Q2	Yes, a low platelet count is associated with an increased risk of death from COVID-19. Platelets play a crucial role in blood clotting, and their depletion can lead to excessive bleeding and organ damage, which can be fatal in severe COVID-19 cases.
Q3	Yes, a high leukocyte count at emergency is associated with an increased risk of death from COVID-19. A high leukocyte count indicates inflammation and immune system activation, which can be a sign of severe COVID-19 infection.

Cuadro 4.8: Ejemplos de respuestas con Gemini

El último apartado de esta sección de resultados trata sobre el enriquecimiento. En la tabla 4.8 se muestran ejemplos de respuestas para las preguntas Q1, Q2 y Q3 con Gemini y el contexto 1. Es importante mencionar el objetivo de la introducción, que consistía en enriquecer las explicaciones obtenidas por las técnicas de explicabilidad. Mediante la sentencia en el **prompt** ("You are an expert on COVID-19 ... followed

by the explanation that validates the answer”) que guía a los modelos a explicar la razón de elegir una respuesta, se obtiene información relevante que enriquece las explicaciones.

Por un lado, Q1 enriquece la respuesta añadiendo información sobre el mayor riesgo de muerte asociado a padecer hipertensión y su relación con el desarrollo de complicaciones relacionadas con el virus. Q2 enriquece la respuesta al añadir razones por las que la característica importante (bajo nivel de plaquetas) desempeña un papel vital en la coagulación de la sangre, lo que puede resultar en un alto riesgo de muerte. Por último, Q3 enriquece la respuesta al indicar los síntomas de tener un nivel alto de leucocitos, que resultan en inflamación y la activación del sistema inmune, siendo una señal de infección grave.

4.2. Resultados del experimento de análisis de veracidad

En la sección actual, se presentan los resultados obtenidos del experimento de análisis de veracidad con todos los modelos. Del mismo modo que con el experimento anterior, se obtiene una lista de 10 respuestas por cada pregunta del *dataset* de veracidad. En este caso, en la tabla 4.9 se muestran los valores medios de las respuestas en el contexto de la veracidad. Es importante mencionar la diferencia con el experimento anterior. En este caso, las respuestas generadas por los modelos se comparan con el *ground truth* del *dataset* SQuAD, que incorpora las respuestas de cada pregunta. Adicionalmente, la tabla 3 del anexo recoge todos los valores de las métricas para cada pregunta, proporcionando una perspectiva más detallada.

	Semantic similarity	Overlap	ROUGE	BLEU
GPT	0,740	0,464	0,301	0,408
Mistral	0,727	0,403	0,222	0,380
Gemini	0,676	0,415	0,273	0,328
Llama2	0,734	0,466	0,239	0,393

Cuadro 4.9: Evaluación media de la veracidad

Los resultados sugieren que GPT genera las respuestas más correctas y precisas, lo que significa que, en promedio, sus respuestas se acercan más al *ground truth* que las de otros modelos. Del mismo modo, Llama2 obtiene resultados similares, con una leve reducción en varias métricas (Similitud semántica, ROUGE y BLEU). Gemini es el modelo de lenguaje menos veraz, ya que sus métricas sugieren que genera las respuestas menos correctas e imprecisas.

Teniendo en cuenta los resultados de la sección de consistencia, estos resultados sugieren que, aunque GPT puede tener problemas de consistencia en comparación con otros LLMs, destaca en la generación de respuestas que se ajustan mejor a los hechos verificables (*ground truth*). Esto revela un equilibrio sutil entre consistencia y veracidad.

4.2.1. Ejemplos de veracidad

De forma complementaria, se exponen ejemplos de las preguntas utilizadas del *dataset* SQuAD, junto con la respuesta de referencia y una de las respuestas generadas

4.2. Resultados del experimento de análisis de veracidad

por los modelos de lenguaje. Estos ejemplos permiten tener una mejor visión y comprensión de las métricas presentadas en la tabla 4.9 mencionada anteriormente.

A continuación, la tabla 4.10 presenta un ejemplo de la veracidad con GPT, esta respuesta generada obtiene unas métricas elevadas (Similitud semántica 0,889, solapamiento 0,679, ROUGE 0,543, BLEU 0,769), sugiriendo un elevado nivel de veracidad, comparado con la media. Estos resultados se pueden comprobar en la tabla 3 del anexo (pregunta Q8).

Pregunta	What is the percentage of Mother to Child Transmission of HIV-1, when there is no intervention?
Ground truth	Without specific interventions, the rate of HIV-1 mother-to-child transmission (MTCT) is approximately 15-45 %
Respuesta generada	Without any intervention, the rate of mother-to-child transmission of HIV-1 is approximately 15-45 %.

Cuadro 4.10: Ejemplo de veracidad con GPT para una pregunta específica

Continuando con otro ejemplo; un caso de respuesta poco veraz se muestra en la tabla 4.11. En este caso (pregunta Q5 de la tabla 3 del anexo), se obtiene una respuesta menos veraz al compararse la respuesta generada con el *ground truth*, con unos valores de métricas inferiores a la media (Similitud semántica 0,634, solapamiento 0,297, ROUGE 0,101, BLEU 0,271).

Pregunta	What is DC-GENR and where is it expressed?
Ground truth	Dendritic cell-specific ICAM-grabbing non-integrin-related (DC-SIGNR, also known as CD209L or liver/lymph node-specific ICAM-grabbing non-integrin (L-SIGN)) can interact with a plethora of pathogens including HIV-1 and is expressed in placental capillary endothelial cells
Respuesta generada	DC-GENR (Dendritic Cell Glycoprotein-1) is a protein expressed on the surface of dendritic cells, which are a type of immune cell that plays a crucial role in the initiation and regulation of immune responses.

Cuadro 4.11: Ejemplo de veracidad con Llama2 para una pregunta específica

4.2.2. Resultados de la extensión del experimento

Este apartado, resultados de la extensión del experimento de veracidad, concluye el capítulo actual. Dicha extensión, consiste en la utilización del GPT Consensus [63] para generar respuestas a una pregunta de ámbito general, que incluyan información exacta y basada en hechos verificables.

Este estudio se ha centrado únicamente en la veracidad y no en la consistencia, debido a que el desarrollo a través de ChatGPT no permite la modificación de la temperatura (valor predefinido a 1 y no 0). Este valor de la temperatura implica la generación de respuestas aleatorias y no deterministas; en otras palabras, a pesar de utilizar la misma entrada, se obtienen salidas no iguales.

Siguiendo el esquema 3.1 mencionado en el capítulo anterior, se muestra el proceso completo para evaluar la veracidad:

Análisis de resultados

- **Paso 1:** se utiliza como entrada para el GPT Consensus, un *prompt* que incluye tanto la pregunta de ámbito general aplicada en este estudio como las indicaciones de la salida esperada.
- **Paso 2:** se obtiene una salida con la explicación a la pregunta y una serie de párrafos con información que valida la respuesta y que incluye artículos de referencia.

A continuación, se expone el *prompt* específico y la respuesta completa, descritos en los pasos 1 y 2:

ENTRADA: *“How does photosynthesis work in plants? Give well-founded answers and the references that give the evidence of the response.”*

SALIDA: *“Photosynthesis in plants is a process that converts light energy into chemical energy, primarily involving water and carbon dioxide to produce biomass and oxygen. Key mechanisms involved in this process include:*

- 1. The absorption of light energy by chlorophyll and the conversion of this energy into chemical forms, which involves managing highly reactive intermediates without causing damage to plant cells. This requires an efficient, robust, and flexible photosynthetic apparatus that can adapt to changing environmental conditions (Kramer, Avenson, & Edwards, 2004).*
- 2. The regulation of the balance between the two photosystems (PSI and PSII) to optimize photosynthetic performance. Plants are capable of redistributing excitation towards the light-limited photosystem through mechanisms such as state transitions, which involve the detachment and attachment of the light-harvesting complex (LHCII) from PSII to PSI in response to changes in illumination (Lunde et al., 2000).*
- 3. Photoprotection mechanisms to prevent photo-oxidative damage under excess light, such as the quenching of excited chlorophyll states, synthesis of antioxidant molecules, and repair processes for damaged photosystems. These mechanisms are crucial for avoiding photoinhibition and maintaining photosynthetic efficiency (Pinola & Bassi, 2018).*
- 4. The dynamic flexibility in the light reactions of photosynthesis, governed by both electron and proton transfer reactions, is essential for the adaptation to short-term environmental changes. This flexibility involves mechanisms for modulating the output ratio of ATP:NADPH and changing the regulatory sensitivity of the light-harvesting antenna to electron and proton flow (Kramer, Avenson, & Edwards, 2004).*

In conclusion, photosynthesis in plants involves a complex interplay of light absorption, energy conversion, and protective mechanisms to ensure efficient and safe energy production necessary for plant growth and oxygen generation. These processes highlight the remarkable adaptability of the photosynthetic apparatus to varying environmental conditions.”

- **Paso 3:** se extraen los 4 párrafos de la salida generada, así como las referencias dadas. Por un lado se almacenan los párrafos como textos independientes (párrafos de referencia) y por otro se buscan los artículos.

4.2. Resultados del experimento de análisis de veracidad

- **Paso 4:** se analizan los artículos y se dividen todos los párrafos de los artículos en una lista de párrafos. Todos los párrafos de la lista se cruzan con el párrafo correspondiente de referencia. Este proceso consiste en medir la veracidad entre el párrafo de referencia y cada párrafo del artículo para obtener el párrafo con las métricas más altas (similitud semántica, solapamiento, ROUGE y BLEU).
- **Paso 5:** se analiza el párrafo con las métricas más elevadas y se evalúa la veracidad de la respuesta dada por el GPT Consensus (que utiliza GPT-4).

Para los párrafos de la salida generada se han obtenido las métricas presentadas en la tabla 4.12. Esta tabla muestra las métricas para cada uno de los párrafos (P1, P2, P3 y P4). Los resultados obtenidos nos indican que las respuestas de este modelo son relativamente veraces. La similitud semántica de todos los párrafos es alta, teniendo en cuenta la estructura de las respuestas de los artículos (ver ejemplo en tabla 4.13). Además, el solapamiento de los textos generados con los de referencia varía entre el 73 % y el 80 %, lo que sugiere una gran relación entre las palabras utilizadas en ambos textos.

	Semantic similarity	Overlap	ROUGE	BLEU
P1	0,798	0,735	0,372	0,321
P2	0,878	0,763	0,333	0,307
P3	0,842	0,738	0,299	0,350
P4	0,867	0,804	0,331	0,377

Cuadro 4.12: Métricas de veracidad para los párrafos generados

Por otro lado, se tienen unas métricas de ROUGE y BLEU bajas y muy inferiores al resto de métricas. Esto se debe a que los párrafos de los artículos tienen una estructura distinta a los párrafos de referencia (P1, P2, P3 y P4). Los párrafos de referencia son breves y precisos en cuanto a la información que exponen, mientras que los párrafos de los artículos son mucho más extensos. Esto resulta en discrepancias en la longitud y en la organización del texto, lo cual resulta en las bajas métricas de ROUGE y BLEU. Un ejemplo de lo mencionado se evidencia en la tabla 4.13.

En este ejemplo, se observa el párrafo con las métricas de similitud semántica y solapamiento más bajas (0,798 0,735). Sin embargo, al analizar este párrafo P1, se aprecian similitudes entre las frases de P1 y el párrafo del artículo. Estas similitudes se han marcado en negrita, y muestran como el modelo de lenguaje incluye toda la información en el texto generado a través de la información del artículo científico. Si bien es cierto que no utiliza exactamente las mismas palabras y las mismas estructuras (lo que implica bajos valores de las métricas ROUGE y BLEU), el texto generado emplea información correcta y verificable que resulta en una respuesta veraz y precisa.

Adicionalmente, se incluye en el anexo el resto de párrafos de referencia y sus correspondientes párrafos coincidentes de los artículos. Las tablas (4, 5, 6) muestran los párrafos P2, P3 y P4 junto al párrafo con mayor similitud dentro de cada artículo. Es importante mencionar que los párrafos P1 y P4 se generan del mismo artículo y del mismo párrafo, por lo que la referencia científica es la misma.

En conclusión, este mecanismo extiende el experimento de veracidad para evaluar la exactitud y fidelidad de las respuestas dadas por un LLM específico: GPT-4. Se ha

Análisis de resultados

empleado el uso del GPT Consensus, que junta el desempeño del modelo de lenguaje con una gran base de datos de artículos científicos, lo que resulta en la generación de respuestas basadas en hechos verificables. Además, se ha demostrado como esta ampliación del experimento obtiene resultados prometedores, con respuestas respaldadas en conocimiento establecido y en información veraz. Por lo tanto, se puede confirmar que el **GPT Consensus es veraz**.

Párrafo de referencia P1	The absorption of light energy by chlorophyll and the conversion of this energy into chemical forms, which involves managing highly reactive intermediates without causing damage to plant cells. This requires an efficient, robust, and flexible photosynthetic apparatus that can adapt to changing environmental conditions
Párrafo del artículo (Kramer, Avenson, & Edwards, 2004)	Plant photosynthesis performs the remarkable feat of converting light energy into usable chemical forms, which involves taming highly reactive intermediates without harming plant cells. This requires an apparatus that is not only efficient and robust but also flexible in its responses to changing environmental conditions. It also requires that the output of the energy-storing reactions be matched with the demands of metabolism. This article addresses the mechanisms by which this flexibility is achieved for short-term environmental changes. We argue that chloroplasts need two types of flexible mechanisms: one for modulating the output ratio of ATP:NADPH, which involves cyclic electron flux around photosystem I; and another for changing the regulatory sensitivity of the light-harvesting antenna to electron (and proton) flow.

Cuadro 4.13: Párrafo de referencia **P1** y párrafo del artículo con mayor similitud.

Capítulo 5

Conclusiones y futuras líneas de investigación

5.1. Conclusiones

La inteligencia artificial explicable es un área de investigación dentro de la IA que busca aportar interpretabilidad y explicabilidad a modelos complejos, cuyo funcionamiento interno es difícil de entender y analizar por los seres humanos. Las técnicas de explicabilidad post-hoc permiten comprender cómo y por qué los modelos toman ciertas decisiones. Estas metodologías se aplican después de que un modelo ha realizado sus predicciones y no alteran el funcionamiento interno.

Dentro de estas técnicas se incluye SHAP, que identifica las características más importantes que influyen en un modelo para tomar ciertas decisiones. No obstante, es esencial validar estas características, lo cual requiere la ayuda de un experto en el dominio donde se aplique la técnica de explicabilidad. En consecuencia, se plantea la posibilidad de utilizar modelos de lenguaje como expertos en el dominio, lo que genera problemas relacionados con la robustez y exactitud de las respuestas que estos modelos producen.

Los grandes modelos de lenguaje han demostrado capacidades notables en la comprensión, generación y manipulación de texto en lenguaje natural. El desempeño de estos modelos está avanzando rápidamente y su uso es cada vez más extendido. Concretamente, en este estudio se han analizado cuatro grandes modelos de lenguaje (GPT, Mistral, Gemini y Llama2) del estado del arte actual, en relación con su consistencia y su veracidad.

Se ha utilizado un conjunto de métricas (similitud semántica, solapamiento, ROUGE y BLEU) para medir la consistencia y la veracidad de las respuestas generadas por los modelos de lenguaje. Las preguntas introducidas a los modelos se han formulado relacionando las salidas de las técnicas de explicabilidad con el lenguaje natural, creando preguntas concretas a partir de características importantes.

La evaluación de los LLMs ha demostrado que, en general, son consistentes y veraces, ya que proporcionan respuestas coherentes y correctas. Específicamente, Mistral sobresale como el LLM más consistente internamente. Sin embargo, su consistencia se reduce al modificar el número de opciones posibles en el *prompt*. GPT resulta ser el

modelo que menos modifica sus respuestas al cambiar dicho número de opciones posibles. Además, este modelo ha demostrado ser el más veraz, no solo por obtener los mejores resultados en comparación con los otros modelos, sino también porque incluye la posibilidad de utilizar GPTs que proporcionan referencias en las respuestas, resultando en respuestas con información basada en hechos verificables.

Además, al analizar la variación de las respuestas al modificar el número de opciones posibles (respuestas binarias frente a respuestas que incluyen una tercera opción de controversia), se ha deducido que la consistencia no depende del número de opciones, pero sí de la coherencia de las explicaciones. Esto es muy relevante porque significa que un número diferente de opciones no solo puede cambiar la respuesta a la pregunta, sino que también puede justificarla de manera distinta.

Por otro lado, se han enriquecido las explicaciones mediante un *prompt* que guía a los modelos de lenguaje a incluir información adicional relevante, relacionada con las características importantes obtenidas de las técnicas de explicabilidad.

En lo que respecta a considerar estos modelos de lenguaje como expertos en un dominio específico, se concluye que son una tecnología muy potente y prometedora que avanza a pasos agigantados. Personalmente, creo que es pronto para emplearlos como fuente de verdad absoluta, ya que tienden a equivocarse en ocasiones. Sin embargo, considero que son una herramienta de apoyo valiosa que puede ayudar tanto a expertos como a no expertos en la toma de decisiones. Además, son capaces de enriquecer las técnicas de explicabilidad y proporcionar información útil y relevante.

5.2. Trabajo futuro

Finalmente, para terminar este trabajo de fin de máster, se indican las futuras líneas de investigación y el trabajo pendiente. La idea principal para continuar con el desarrollo es modificar las respuestas obtenidas de los modelos de lenguaje al alterar el *prompt*.

Se ha demostrado cómo pueden variar las respuestas de los LLMs al modificar únicamente el número de opciones posibles en las respuestas. Por ello, la futura línea de investigación consistirá en introducir diferentes tipos de *prompts* a los modelos de lenguaje para obtener respuestas falsas. Es decir, se tratará de inducir distintos roles y/o inyectar información falsa para lograr que las respuestas de los LLMs cambien y no se obtengan las respuestas de un *ground truth*.

Los diferentes tipos de *prompts* para alterar las respuestas de los modelos de lenguaje son:

- **Agresivo:** se guía al modelo de lenguaje a ser ignorante o con pocos conocimientos, además de proporcionar en el *prompt* información falsa (noticias falsas, menciones de científicos importantes falsas, etc).
- **Sutil:** no se orienta al modelo de lenguaje con ningún rol. Únicamente se aporta información falsa.
- **Neutral:** no se aplica ningún cambio de rol ni se proporciona información adicional.
- **Experto:** se indica al LLM que es un experto en el dominio.

Conclusiones y futuras líneas de investigación

Mediante esta metodología se debería conseguir que el *prompt* agresivo haga que el modelo de lenguaje se equivoque o genere información falsa. Por ejemplo, al hacer una pregunta con una respuesta de conocimiento establecido como “Is water composed of two hydrogen atoms and one oxygen atom?”, la salida deseada al emplear un *prompt* agresivo debería ser “no”, cuando en realidad es “yes”. Por otro lado, el *prompt* sutil, debería lograr en ocasiones que el modelo de lenguaje se equivoque, al inyectar *fake news*. El *prompt* neutral proporciona las respuestas habituales de un modelo de lenguaje, y por último, el *prompt* experto debería ser resistente a cambiar su respuesta aunque se inyectaran noticias falsas.

Bibliografía

- [1] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson, “Explainable artificial intelligence: an analytical review,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 5, p. e1424, 2021.
- [2] G. Montavon, W. Samek, and K.-R. Müller, “Methods for interpreting and understanding deep neural networks,” *Digital signal processing*, vol. 73, pp. 1–15, 2018.
- [3] F. K. Došilović, M. Brčić, and N. Hlupić, “Explainable artificial intelligence: A survey,” in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 0210–0215, 2018.
- [4] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (xai),” *IEEE Access*, vol. 6, pp. 52138–52160, 2018.
- [5] R. Tomsett, D. Braines, D. Harborne, A. Preece, and S. Chakraborty, “Interpretable to whom? a role-based model for analyzing interpretable machine learning systems,” *arXiv preprint arXiv:1806.07552*, 2018.
- [6] A. Das and P. Rad, “Opportunities and challenges in explainable artificial intelligence (xai): A survey,” 2020.
- [7] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” 2019.
- [8] D. Vale, A. El-Sharif, and M. Ali, “Explainable artificial intelligence (xai) post-hoc explainability methods: risks and limitations in non-discrimination law,” *AI and Ethics*, vol. 2, pp. 1–12, 03 2022.
- [9] T. Han, S. Srinivas, and H. Lakkaraju, “Which explanation should i choose? a function approximation perspective to characterizing post hoc explanations,” in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 5256–5268, Curran Associates, Inc., 2022.
- [10] E. M. Kenny, C. Ford, M. Quinn, and M. T. Keane, “Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies,” *Artificial Intelligence*, vol. 294, p. 103459, 2021.
- [11] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barabado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, “Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities

- and challenges toward responsible ai,” *Information Fusion*, vol. 58, pp. 82–115, 2020.
- [12] M. T. Ribeiro, S. Singh, and C. Guestrin, ““why should i trust you?”: Explaining the predictions of any classifier,” 2016.
 - [13] J. Dieber and S. Kirrane, “Why model why? assessing the strengths and limitations of lime,” 2020.
 - [14] Z. Zhou, G. Hooker, and F. Wang, “S-lime: Stabilized-lime for model explanation,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD ’21, (New York, NY, USA), p. 2429–2438, Association for Computing Machinery, 2021.
 - [15] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
 - [16] L. S. Shapley *et al.*, “A value for n-person games,” 1953.
 - [17] M. Vega García and J. L. Aznarte, “Shapley additive explanations for no2 forecasting,” *Ecological Informatics*, vol. 56, p. 101039, 2020.
 - [18] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153, PMLR, 06–11 Aug 2017.
 - [19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
 - [20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
 - [21] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated decisions and the gdpr,” 2018.
 - [22] S. Verma, J. Dickerson, and K. Hines, “Counterfactual explanations for machine learning: A review,” *arXiv preprint arXiv:2010.10596*, vol. 2, 2020.
 - [23] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, “A survey on large language model (llm) security and privacy: The good, the bad, and the ugly,” *High-Confidence Computing*, p. 100211, 2024.
 - [24] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, “A survey of large language models,” 2023.
 - [25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
 - [26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013.

- [27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems* (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.
- [28] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.
- [29] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.
- [30] “Tensorflow embedding projector.” https://www.tensorflow.org/tensorboard/embedding_projector, 2017. Accessed: 2024-06-07.
- [31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (M. Walker, H. Ji, and A. Stent, eds.), (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.
- [32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
- [34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019.
- [35] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [36] T. Brown and e. a. Mann, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.
- [37] A. C. et al, “Palm: Scaling language modeling with pathways,” *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.
- [38] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, “Emergent abilities of large language models,” 2022.
- [39] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, “Why does unsupervised pre-training help deep learning?,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterton, eds.), 2009.

- eds.), vol. 9 of *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy), pp. 201–208, PMLR, 13–15 May 2010.
- [40] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” *Advances in neural information processing systems*, vol. 19, 2006.
 - [41] P. Mavrepis, G. Makridis, G. Fatouros, V. Koukos, M. M. Separdani, and D. Kyriazis, “Xai for all: Can large language models simplify explainable ai?,” *ArXiv*, vol. abs/2401.13110, 2024.
 - [42] T. Ali and P. Kostakos, “Huntgpt: Integrating machine learning-based anomaly detection and explainable ai with large language models (llms),” 2023.
 - [43] L. Breiman, “Random forests,” vol. 45, p. 5–32, oct 2001.
 - [44] O. Tursun, S. Denman, S. Sridharan, and C. Fookes, “Towards self-explainability of deep neural networks with heatmap captioning and large-language models,” 2023.
 - [45] F. Nazary, Y. Deldjoo, and T. D. Noia, “Chatgpt-healthprompt. harnessing the power of xai in prompt-based healthcare decision support using chatgpt,” 2023.
 - [46] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, ACM, Aug. 2016.
 - [47] G. P. Wellawatte and P. Schwaller, “Extracting human interpretable structure-property relationships in chemistry using xai and large language models,” 2023.
 - [48] O. J. A. et al., “Gpt-4 technical report,” 2023.
 - [49] G. Team and R. A. et al., “Gemini: A family of highly capable multimodal models,” 2024.
 - [50] H. T. et al., “Llama 2: Open foundation and fine-tuned chat models,” *ArXiv*, vol. abs/2307.09288, 2023.
 - [51] Mistral, “Mistral large, our new flagship model.” URL <https://mistral.ai/news/mistral-large/>, 2024.
 - [52] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, “Mistral 7b,” *ArXiv*, vol. abs/2310.06825, 2023.
 - [53] P. Cardinal-Fernandez, E. Garcia-Cuesta, J. Barberan, J. F. Varona, A. Estirado, A. Moreno, J. Villanueva, M. Villareal, O. Baez-Pravia, J. Menendez, *et al.*, “Clinical characteristics and outcomes of 1,331 patients with covid-19: Hm spanish cohort,” *Revista Española de Quimioterapia*, vol. 34, no. 4, p. 342, 2021.
 - [54] D. R. Cox, “Regression Models and Life-Tables,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, pp. 187–202, 12 2018.
 - [55] E. García-Cuesta, J. Huertas-Tato, P. Cardinal-Fernandez, and J. Barberán, “Explainable risk prediction on streaming data: a lstms application to sars-cov-2 patients,” in *IEEE International Conference on Evolving and Adaptive Intelligent*

- Systems 2024 (IEEE EAIS 2024)*, (Universidad Carlos III de Madrid: Campus de Madrid - Puerta de Toledo), 2024.
- [56] H. Raj, V. Gupta, D. Rosati, and S. Majumdar, “Semantic consistency for assuring reliability of large language models,” 2023.
- [57] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (J. Su, K. Duh, and X. Carreras, eds.), (Austin, Texas), pp. 2383–2392, Association for Computational Linguistics, Nov. 2016.
- [58] D. Chandrasekaran and V. Mago, “Evolution of semantic similarity—a survey,” *ACM Comput. Surv.*, vol. 54, feb 2021.
- [59] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” 2019.
- [60] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.
- [61] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (P. Isabelle, E. Charniak, and D. Lin, eds.), (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002.
- [62] OpenAI, “Introducing gpts,” 2023. Accessed: 2024-06-23.
- [63] Consensus, “What is consensus?,” 2024. Accessed: 2024-06-23.
- [64] P. Sebo, “How accurate are the references generated by chatgpt in internal medicine?,” *Internal and Emergency Medicine*, vol. 19, no. 1, pp. 247–249, 2024.

Anexo

	GPT				Mistral				Gemini				Llama2			
	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU
q1	0.949	0.760	0.666	0.462	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q2	0.989	0.941	0.935	0.880	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999	0.983	0.983	0.977
q3	0.975	0.868	0.789	0.731	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q4	0.984	0.823	0.757	0.653	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.992	0.990	0.985
q5	0.981	0.895	0.841	0.738	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q6	0.996	0.972	0.948	0.925	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q7	0.980	0.864	0.833	0.800	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q8	0.993	0.899	0.874	0.797	1.000	1.000	1.000	1.000	0.998	0.964	0.961	0.922	1.000	1.000	1.000	1.000
q9	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q10	0.987	0.858	0.800	0.710	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Average	0.983	0.888	0.844	0.770	1.000	1.000	1.000	1.000	1.000	0.996	0.996	0.992	1.000	0.998	0.997	0.996

Cuadro 1: Evaluación de la consistencia por pregunta para el *Contexto 1*

	GPT				Mistral				Gemini				Llama2			
	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU
q1	0.976	0.969	0.924	0.825	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q2	0.997	0.990	0.988	0.976	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q3	0.972	0.869	0.832	0.663	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q4	0.998	0.926	0.920	0.875	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q5	0.997	1.000	0.963	0.904	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q6	0.939	0.706	0.556	0.383	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.991	0.969	0.957	0.925
q7	0.949	0.736	0.649	0.475	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q8	0.996	0.932	0.896	0.867	0.991	0.954	0.927	0.903	0.995	0.960	0.957	0.925	0.995	0.944	0.932	0.903
q9	0.990	0.927	0.906	0.862	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
q10	0.982	0.918	0.895	0.796	1.000	0.991	0.989	0.977	0.998	1.000	0.975	0.938	1.000	1.000	1.000	1.000
Average	0.980	0.897	0.853	0.763	0.999	0.995	0.992	0.988	0.999	0.996	0.993	0.986	0.999	0.991	0.989	0.983

Cuadro 2: Evaluación de la consistencia por pregunta para el *Contexto 2*

	GPT				Mistral				Gemini				Llama2			
	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU	Semantic similarity	Overlap	ROUGE	BLEU
q1	0.777	0.750	0.466	0.612	0.770	0.750	0.424	0.537	0.412	0.250	0.181	0.368	0.777	0.750	0.466	0.612
q2	0.647	0.554	0.331	0.347	0.677	0.550	0.244	0.298	0.596	0.599	0.461	0.496	0.668	0.700	0.363	0.294
q3	0.687	0.285	0.178	0.289	0.681	0.380	0.173	0.349	0.698	0.500	0.266	0.358	0.680	0.380	0.210	0.420
q4	0.764	0.448	0.207	0.409	0.805	0.466	0.239	0.507	0.714	0.157	0.093	0.218	0.648	0.290	0.163	0.304
q5	0.665	0.428	0.255	0.184	0.619	0.312	0.222	0.259	0.622	0.350	0.260	0.137	0.634	0.297	0.101	0.271
q6	0.778	0.458	0.368	0.479	0.737	0.333	0.170	0.384	0.774	0.533	0.294	0.242	0.787	0.500	0.307	0.436
q7	0.758	0.350	0.191	0.373	0.733	0.256	0.161	0.361	0.631	0.413	0.254	0.306	0.821	0.435	0.242	0.454
q8	0.889	0.679	0.543	0.769	0.791	0.411	0.235	0.440	0.858	0.529	0.461	0.673	0.926	0.647	0.156	0.405
q9	0.705	0.269	0.202	0.196	0.725	0.242	0.107	0.259	0.746	0.230	0.163	0.220	0.691	0.222	0.175	0.300
q10	0.731	0.420	0.264	0.423	0.728	0.333	0.244	0.402	0.708	0.588	0.299	0.266	0.706	0.441	0.210	0.437
Average	0.740	0.464	0.301	0.408	0.727	0.403	0.222	0.380	0.676	0.415	0.273	0.328	0.734	0.466	0.239	0.393

Cuadro 3: Evaluación de la veracidad para cada pregunta con el *Contexto 3*

Párrafo de referencia P2	The regulation of the balance between the two photosystems (PSI and PSII) to optimize photosynthetic performance. Plants are capable of redistributing excitation towards the light-limited photosystem through mechanisms such as state transitions, which involve the detachment and attachment of the light-harvesting complex (LHCII) from PSII to PSI in response to changes in illumination
Párrafo del artículo (Lunde et al., 2000)	Photosynthesis in plants involves two photosystems responsible for converting light energy into redox processes. The photosystems, PSI and PSII, operate largely in series, and therefore their excitation must be balanced in order to optimize photosynthetic performance¹. When plants are exposed to illumination favouring either PSII or PSI they can redistribute excitation towards the light-limited photosystem. Long-term changes in illumination lead to changes in photosystem stoichiometry^{2,3}. In contrast, state transition is a dynamic mechanism that enables plants to respond rapidly to changes in illumination. When PSII is favoured (state 2), the redox conditions in the thylakoids change and result in activation of a protein kinase^{4±6}. The kinase phosphorylates the main light-harvesting complex (LHCII) and the mobile antenna complex is detached from PSII. It has not been clear if attachment of LHCII to PSI in state 2 is important in state transitions. Here we show that in the absence of a specific PSI subunit, PSI-H, LHCII cannot transfer energy to PSI, and state transitions are impaired.

Cuadro 4: Párrafo de referencia **P2** y párrafo del artículo con mayor similitud.

Párrafo de referencia P3	Photoprotection mechanisms to prevent photo-oxidative damage under excess light, such as the quenching of excited chlorophyll states, synthesis of antioxidant molecules, and repair processes for damaged photosystems. These mechanisms are crucial for avoiding photoinhibition and maintaining photosynthetic efficiency
Párrafo del artículo Pinnola & Bassi, 2018)	Photosynthesis uses sunlight to convert water and carbon dioxide into biomass and oxygen. When in excess, light can be dangerous for the photosynthetic apparatus because it can cause photo-oxidative damage and decreases the efficiency of photosynthesis because of photoinhibition. Plants have evolved many photoprotective mechanisms in order to face reactive oxygen species production and thus avoid photoinhibition. These mechanisms include quenching of singlet and triplet excited states of chlorophyll, synthesis of antioxidant molecules and enzymes and repair processes for damaged photosystem II and photosystem I reaction centers. This review focuses on the mechanisms involved in photoprotection of chloroplasts through dissipation of energy absorbed in excess.

Cuadro 5: Párrafo de referencia **P3** y párrafo del artículo con mayor similitud.

Párrafo de referencia P4	The dynamic flexibility in the light reactions of photosynthesis, governed by both electron and proton transfer reactions, is essential for the adaptation to short-term environmental changes. This flexibility involves mechanisms for modulating the output ratio of ATP:NADPH and changing the regulatory sensitivity of the light-harvesting antenna to electron and proton flow
Párrafo del artículo (Kramer, Avenson, & Edwards, 2004)	Plant photosynthesis performs the remarkable feat of converting light energy into usable chemical forms, which involves taming highly reactive intermediates without harming plant cells. This requires an apparatus that is not only efficient and robust but also flexible in its responses to changing environmental conditions. It also requires that the output of the energy-storing reactions be matched with the demands of metabolism. This article addresses the mechanisms by which this flexibility is achieved for short-term environmental changes. We argue that chloroplasts need two types of flexible mechanisms: one for modulating the output ratio of ATP:NADPH, which involves cyclic electron flux around photosystem I; and another for changing the regulatory sensitivity of the light-harvesting antenna to electron (and proton) flow.

Cuadro 6: Párrafo de referencia **P4** y párrafo del artículo con mayor similitud.