

Trabajo de Fin de Grado

RAG Optimizado para Entornos Críticos: Mitigación, Detección y Evaluación de Alucinaciones (Aluora)

TITULACIÓN: Grado en Ciencia e Ingeniería de Datos

AUTOR: Pedro Sarmiento Yáñez

TUTORIZADO POR:

José María Quinteiro González

José Juan Hernández Cabrera

Junio 2025

SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO

D. Pedro Sarmiento Yáñez, autor del trabajo RAG Optimizado para Entornos Críticos: Mitigación, Detección y Evaluación de Alucinaciones (Aluora), correspondiente a la titulación Grado en Ciencia e Ingeniería de Datos.

SOLICITA

que se inicie el procedimiento de defensa del mismo, para lo que se adjunta la documentación requerida, haciendo constar que

☒ se autoriza / ☐ no se autoriza la grabación en audio de la exposición y turno de preguntas.

Asimismo, con respecto al registro de la propiedad intelectual/industrial del TFT, declara que:

☐ Se ha iniciado o hay intención de iniciarlo (defensa no pública).

☒ No está previsto.

Y para que así conste firma la presente. (fecha en firma electrónica)

El/La estudiante

Fdo. _____

A rellenar y firmar **obligatoriamente** por el/la/los/las tutores

En relación con la presente solicitud, se informa: (firmar donde corresponda)

Positivamente (en caso de detección de copia, esta firma quedará invalidada)

Fdo. _____

Negativamente (justificación en TFT05)

Fdo. _____

DIRECTOR DE LA ESCUELA DE INGENIERÍA INFORMÁTICA

Agradecimientos

Quiero mostrar mi más sentido agradecimiento a todos los profesores, compañeros y familiares que me han acompañado en este camino.

En especial:

A mis tutores, José María Quinteiro y José Juan Hernández, por su ayuda e implicación en la idea y el desarrollo de este TFG. Gracias por valorar mi trabajo y formar parte de él, y también por los consejos, que no son útiles solamente para este trabajo, sino también para el futuro.

A mis padres, que siempre han confiado en mí, me han apoyado en los peores momentos y han celebrado los buenos, sin su ayuda, motivación y cariño esto no sería posible.

A mi abuela Lola, que todavía la siento muy cerca de mí, prometo que he sentido tu energía y ayuda desde arriba.

A mi abuela Tata, que siempre se preocupa por mi futuro, sé que está muy orgullosa de mí, pero yo estoy aún más agradecido de tenerla en mi vida.

A mis abuelos Pedro y Evaristo, su ayuda me ha guiado por este camino, acompañándome en cada una de mis decisiones.

Y por último a Ronny, Tara, Sila, Yarko, Wendy y Nami, aunque algunos ya no estén, los llevo a todos en mi corazón.

Resumen

Este trabajo ha profundizado en el análisis de las alucinaciones, centrándose específicamente en las de tipo factual, que son las más comunes. Se han desarrollado y aplicado modelos para su detección y evaluación, se han explorado técnicas de mitigación y se ha recurrido a fundamentos matemáticos para explicar su origen. El proyecto ha combinado un riguroso enfoque científico con pruebas prácticas, y otros experimentos orientados a ofrecer resultados tangibles al usuario final, partiendo de una hipótesis inicial para llegar a las conclusiones obtenidas. Este esfuerzo busca aportar claridad y herramientas concretas para abordar la generación de información no veraz, resultando especialmente crucial para la fiabilidad de su uso en sectores críticos.

Abstract

This project provides an in-depth analysis of hallucinations, focusing specifically on factual hallucinations, which are the most common. Models for their detection and assessment have been developed and applied, mitigation techniques have been explored, and mathematical foundations have been used to explain their origin. The project has combined a rigorous scientific approach with practical tests and other experiments aimed at providing tangible results to the end user, starting from an initial hypothesis to reach the conclusions obtained. This effort seeks to provide clarity and concrete tools to address the generation of untruthful information, which is especially crucial for the reliability of its use in critical sectors.

Índice general

1. Introducción	1
1.1. ¿Qué son las alucinaciones en los Modelos de Lenguaje de Gran Tamaño? . .	3
1.2. Definición del Problema	4
1.3. Tipos de alucinaciones	5
1.4. Estado de los modelos actuales	6
1.4.1. Modelos online o accesibles mediante API	6
1.4.2. Modelos disponibles para ejecución local	7
1.4.3. Comparativa y recomendaciones de uso	7
1.5. Motivación	9
1.6. Estructura del trabajo	10
2. Estado actual y objetivos iniciales	11
2.1. Estado del Arte	11
2.1.1. Alucinaciones según la tarea	12
2.1.2. ¿Cómo funcionan los LLM?	13
2.1.3. ¿Por qué suceden las alucinaciones?	16
2.1.4. Escalado por temperatura y su papel en las alucinaciones	16
2.1.5. ¿Por qué las alucinaciones son inherentes a los LLM?	19
2.1.6. Técnicas de Mitigación, Evaluación y Detección de alucinaciones . . .	25
2.1.7. Mitigación	25
2.1.8. Evaluación	28
2.1.9. Detección	30
2.2. Alucinaciones numéricas	33
2.3. Objetivos iniciales	34
3. Aportaciones del trabajo	35
3.1. Principales aportaciones	35
3.1.1. Aportaciones técnicas y aplicadas	35
3.1.2. Contribuciones científicas y conceptuales	36
3.1.3. Justificación y repercusión esperada	38
3.2. Competencias específicas	38
3.3. Alineamiento con los objetivos de desarrollo sostenible	39
4. Desarrollo	41

4.1. Metodología	41
4.2. Herramientas utilizadas	42
4.3. Datasets utilizados	44
4.4. Fases de desarrollo	45
4.4.1. Ajuste a la planificación inicialmente prevista	54
4.4.2. Modificación en los objetivos planteados	54
5. Resultados	56
6. Conclusiones y trabajo futuro	63
6.1. Conclusiones	63
6.2. Trabajo futuro	65
7. Otros capítulos y anexos	72
7.1. Expresiones matemáticas	72

Índice de figuras

2.1.	Ejemplo con 3 clases del impacto que tiene la temperatura en la probabilidad asignada a una clase mediante la función softmax. Fuente: Mathematical Insights into Temperature Scaling and Hallucination in Large Language Models	18
2.2.	Arquitectura LettuceDetect. Fuente: “LettuceDetect: A Hallucination Detection Framework for RAG Applications”	31
2.3.	Arquitectura HalluDetect. Fuente: “Detecting Hallucinations in Large Language Model Generation: A Token Probability Approach”	32
3.1.	ReadMe de la aplicación Aluora. [28]	36
3.2.	Ejemplo de uso: Datos de Entrada Aluora.	37
3.3.	Ejemplo de uso: Salida de Aluora.	37
5.1.	Mejor árbol de decisión generado.	60

Índice de cuadros

1.1. Tabla Comparativa Resumida: LLM ejecutado en local vs. LLM via API o online. Fuente: Froxy Labs	8
3.1. Grado de relación del TFT con los objetivos de desarrollo sostenible.	40
5.1. Ejemplo de resultados de LLaMA 3 sobre PubMedQA	57
5.2. Ejemplos de alucinaciones cometidas por el modelo generador, que el modelo evaluador observó como generaciones correctas	57
5.3. Comparativa de rendimiento entre modelos originales, modificados y variantes con Dropout en HalluDetect	61

Capítulo 1

Introducción

”La verdadera sabiduría está en
reconocer la propia ignorancia”

Sócrates

En este capítulo se aborda de manera integral el fenómeno de las alucinaciones en los Modelos de Lenguaje de Gran Tamaño (LLM), identificándolo como un problema fundamental y un área de investigación prioritaria en el campo de la inteligencia artificial (IA) contemporánea. Se iniciará con la definición más precisa posible de qué implican las alucinaciones, para luego explorar en detalle las diversas tipologías que pueden presentar y las razones que explican por qué estas manifestaciones parecen ser una característica inherente a los LLM actuales. Posteriormente, se realizará un análisis del estado actual de diferentes modelos del lenguaje prominentes, examinando su comportamiento frente a la generación de alucinaciones. Este primer capítulo sentará las bases para comprender la motivación de este Trabajo de Fin de Grado, que se centrará en la crucial tarea de investigar y proponer estrategias para la detección eficaz de dichas alucinaciones.

En la era de la inteligencia artificial generativa (IAG) en la que se encuentra actualmente la sociedad, los LLM han emergido como herramientas con capacidades asombrosas, capaces de redactar textos coherentes, traducir idiomas con fluidez e incluso generar código. Su habilidad para procesar y generar lenguaje natural a una escala sin precedentes ha abierto un abanico de posibilidades en innumerables campos. Sin embargo, esta poderosa tecnología no está exenta de desafíos inherentes a su compleja naturaleza.

A medida que interactuamos más con estos modelos, comenzamos a observar comportamientos peculiares, instancias donde la información generada se desvía de la realidad fáctica, introduce datos incorrectos o incluso inventa escenarios completos. Estos episodios, donde la frontera entre la información veraz y la ficción se difumina, conducen a uno de los fenómenos más intrigantes y problemáticos en el estudio actual de los LLM: las ‘alucinaciones’. Antes de adentrarnos en una definición técnica, es crucial comprender el impacto y la naturaleza de estas respuestas aparentemente anómalas que pueden socavar la confianza y la utilidad de estas prometedoras tecnologías.

La propia terminología que usamos para describir las fallas de los LLM, 'alucinaciones', hace pensar inevitablemente en la complejidad de la mente humana. Durante siglos, filósofos, psicólogos y neurocientíficos han intentado desentrañar los misterios de la conciencia, la percepción, la memoria y la creatividad humanas. Sabemos que nuestra mente no es un registrador pasivo de la realidad. Al contrario, es un sistema activo y constructivo que interpreta la información sensorial, rellena lagunas basándose en experiencias previas y expectativas, y es capaz de generar mundos internos completamente nuevos a través de la imaginación y los sueños.

Fenómenos como las ilusiones ópticas demuestran que nuestra percepción puede ser engañada, y los falsos recuerdos revelan que nuestra memoria es un proceso reconstructivo susceptible a errores y distorsiones. Incluso en un estado de plena conciencia, nuestra mente puede 'desviarse', generar pensamientos espontáneos o llegar a conclusiones que no se ajustan estrictamente a la lógica o a la evidencia externa. Estas características, si bien son fundamentales para la creatividad, la adaptación y la propia experiencia subjetiva, también subrayan que la 'verdad' y la 'realidad' son conceptos que la mente humana procesa de una manera intrincada.

Es por eso que se debe comprender que la generación de información que no se corresponde con una realidad externa consensuada no es un fenómeno exclusivo de las máquinas. Massimo Piattelli Palmarini, en su obra "Los túneles de la mente : ¿Qué se esconde tras nuestros errores?" [26], muestra la fascinante idea de que nuestra mente opera a través de ciertos "túneles" cognitivos. Lejos de ser una máquina de lógica perfecta, el cerebro humano está inherentemente predispuesto a caer en trampas, patrones de error recurrentes que llevan a interpretar la realidad de manera distorsionada. El autor se adentra en cómo estas tendencias hacen preferir ciertos atajos mentales que conducen a conclusiones erróneas. La obra sugiere que somos "víctimas" de estas predisposiciones mentales que actúan como túneles, canalizando nuestro pensamiento por caminos predefinidos que no siempre se alinean con la lógica formal.

Así, hemos visto cómo la mente humana está lejos de ser un espejo infalible de la realidad. Esta comprensión de las sutilezas y potenciales 'desviaciones' en el procesamiento de información del sistema cognitivo humano ofrece una perspectiva valiosa al dirigir nuestra atención hacia los LLM. Si los sistemas biológicos, con millones de años de evolución, pueden generar interpretaciones o 'realidades' que se apartan de los hechos, no resulta del todo sorprendente encontrar fenómenos análogos, aunque con mecanismos subyacentes radicalmente distintos, en las inteligencias artificiales que han nacido hace unos pocos años. Cuando estas tecnologías despliegan información carente de factualidad, surge un fenómeno crítico.

A continuación, en el primer punto de esta introducción, abordaremos la definición de las alucinaciones en los Modelos de Lenguaje de Gran Tamaño.

1.1. ¿Qué son las alucinaciones en los Modelos de Lenguaje de Gran Tamaño?

Establecer una definición única y universalmente aceptada para las alucinaciones en el campo de la Inteligencia Artificial presenta una notable complejidad. No obstante, las conceptualizaciones ofrecidas por organizaciones de referencia y medios especializados en el sector tecnológico proporcionan marcos valiosos para su comprensión.

Según IBM, “la alucinación de la IA es un fenómeno en el que un modelo de lenguaje de gran tamaño (LLM) percibe patrones u objetos que son inexistentes o imperceptibles para los observadores humanos, creando resultados que no tienen sentido o son completamente inexactos” [11]. Por su parte, la BBC caracteriza estos errores como “información arrojada por el sistema que, aunque esté escrita de manera coherente, presenta datos incorrectos, sesgados o plenamente erróneos.”[3]

De estas dos definiciones del término y complementándolas con el análisis general del fenómeno, a falta de la existencia de una definición única, se pueden conocer algunas características clave que representan a las alucinaciones. Estas pueden ser:

- ✓ **Resultados falsos o inexactos:** La información generada no se corresponde con hechos verificables o la realidad.
- ✓ **Coherencia:** Las alucinaciones suelen estar bien escritas, ser gramaticalmente correctas y sonar convincentes, lo que dificulta su detección.
- ✓ **Desvinculación de las fuentes de datos:** La información generada no tiene un respaldo real en los datos de entrenamiento del modelo o contradice directamente dichos datos.
- ✓ **Confianza en la generación:** El modelo presenta la información alucinada con el mismo grado de aparente certeza que la información factual.
- ✓ **Generación de contenido incorrecto:** A menudo, no se trata solo de un error al recordar un dato, sino de la invención activa de información.

Para ilustrar de forma práctica cómo las características previamente descritas se manifiestan conjuntamente, considérese un conocido incidente ocurrido poco después de la presentación pública de un chatbot desarrollado por una importante compañía tecnológica. A principios de febrero de 2023, Google presentó su nuevo chatbot, Bard, el cual, al responder a preguntas sobre el Telescopio Espacial James Webb, proporcionó una respuesta que ejemplifica una alucinación. El problema radicó en una de las afirmaciones del bot, específicamente, que el telescopio había capturado las primeras imágenes de un planeta fuera de nuestro sistema solar, era completamente falsa.[30] El primer planeta fuera del Sistema Solar en obtenerse por imagen directa fue 2M1207 b, captado en abril de 2004 por el Very Large Telescope (VLT)[22]

Más allá de ejemplos puntuales, la propensión de los LLM a incurrir en alucinaciones evidencia una problemática de fondo que requiere un análisis detallado. La existencia de este

fenómeno subraya la urgencia de comprender por qué se trata de un desafío de considerable magnitud, cuyas ramificaciones se extienden al interés general y requieren una atención prioritaria. Esta constatación conduce directamente a la necesidad de abordar la definición del problema en toda su amplitud.

1.2. Definición del Problema

Una vez conceptualizadas las alucinaciones, se puede observar que su manifestación en los Modelos de Lenguaje Grandes trasciende la categoría de simple error técnico para consolidarse como un problema complejo. La capacidad de estos sistemas para generar contenido que, si bien puede ser coherente, resulta ser falso, engañoso o carente de fundamentación en datos, plantea serias preocupaciones que van más allá del ámbito puramente tecnológico e impactan directamente en la interacción entre la sociedad y la inteligencia artificial.

La problemática de las alucinaciones no se limita a la mera producción de datos incorrectos, ya que sus implicaciones son profundas y se extienden a diversas esferas cruciales. En primer lugar, impacta de manera directa a la fiabilidad de los LLM como fuentes de información y herramientas de asistencia, lo que inevitablemente erosiona la confianza de los usuarios. Este fenómeno no solo afecta la percepción individual de estas tecnologías, sino que también tiene el potencial de amplificar la propagación de desinformación a una escala sin precedentes.

El ejemplo visto anteriormente, más allá de la distribución de información falsa, no tiene mayores consecuencias; el problema real lo encontramos en la incursión de los LLM en sectores críticos como la medicina, las finanzas o el ámbito legal, donde las decisiones basadas en información incorrecta pueden tener consecuencias graves. Esto convierte a las alucinaciones en un riesgo tangible. Por último, desde una perspectiva de desarrollo, la persistencia de las alucinaciones representa un obstáculo técnico y ético considerable para la implementación responsable y la adopción generalizada de la IA.

Para ilustrar la gravedad potencial de las alucinaciones en sectores críticos, podemos considerar varios escenarios. Hipotéticamente, en el ámbito sanitario, una inteligencia artificial médica podría, a causa de una alucinación, proponer un tratamiento incorrecto para una determinada patología, con el riesgo de generar graves perjuicios para la salud del paciente. De manera similar, en el sector financiero, un sistema de IA que asesore sobre inversiones podría, basándose en datos fabricados sobre el rendimiento de un activo o falsas tendencias del mercado, inducir a tomar decisiones que desemboquen en cuantiosas pérdidas económicas.

Estos riesgos no son meramente teóricos, incidentes reales ya han comenzado a subrayar la seriedad del problema. Un ejemplo de ello es el conocido caso de Air Canada, donde el chatbot de la aerolínea norteamericana proporcionó información a un cliente sobre una política de tarifas, dando a entender que existía la posibilidad de conseguir un reembolso, lo cual resultó no ser correcto. Posteriormente, un tribunal dictaminó que la compañía era responsable de la información errónea facilitada por su agente de IA, lo que evidencia las implicaciones legales y de responsabilidad corporativa que pueden surgir de las alucinaciones de estos sistemas

[13]. Este tipo de situaciones pone de relieve la necesidad urgente de abordar la fiabilidad de la información generada por LLM.

Una vez expuesta con claridad la dimensión del problema que representan las alucinaciones en los Modelos de Lenguaje de Gran Tamaño, resulta esencial profundizar en la clasificación específica de estas. Comprender las diversas formas en que pueden manifestarse las alucinaciones permite no solo identificar patrones comunes de error, sino también establecer una base sólida para desarrollar estrategias efectivas destinadas a su detección y mitigación. A continuación, se presentan los principales tipos de alucinaciones.

1.3. Tipos de alucinaciones

Según Fareed Khan, podemos distinguir diversos tipos de alucinaciones, cada uno con características particulares [7]:

- ✓ **Alucinación fáctica:** Ocurre cuando los resultados generados por el modelo son incorrectos o directamente inventados. Un ejemplo sería afirmar que Marie Curie descubrió la penicilina en 1928, cuando realmente fue Alexander Fleming quien realizó este descubrimiento.
- ✓ **Alucinación temporal:** Sucede cuando el modelo presenta conocimiento obsoleto o desactualizado como si fuera actual. Un ejemplo sería afirmar que el presidente actual de Estados Unidos es Barack Obama, cuando esta información pertenece a un periodo pasado.
- ✓ **Alucinación contextual:** Se manifiesta cuando el modelo añade conceptos que no fueron mencionados ni implícitos en los datos originales. Por ejemplo, al resumir un informe de proyecto, el modelo podría añadir que “el equipo planeó una fiesta sorpresa” aunque el informe original nunca mencionó dicha fiesta.
- ✓ **Alucinación lingüística:** Son errores en los que la generación es gramaticalmente correcta pero semánticamente incoherente o sin sentido. Por ejemplo, una frase como “el algoritmo bailarín recita versos imposibles bajo el sol de medianoche”.
- ✓ **Alucinación extrínseca:** Se producen cuando el modelo proporciona información que no está respaldada por los documentos fuente disponibles. Un ejemplo sería afirmar, según un informe inexistente de la NASA de 2024, que Júpiter es más pequeño que la Tierra.

Las alucinaciones más frecuentes en los modelos actuales son las alucinaciones fácticas. Sin embargo, es importante destacar que el tipo de alucinación predominante puede variar dependiendo del modelo elegido y la forma en que se utilice. Por ejemplo, en casos donde se realicen preguntas relacionadas con sucesos recientes o información actualizada, las alucinaciones temporales tienden a ser especialmente comunes, ya que los modelos podrían recurrir a datos obsoletos o desactualizados que presentan como actuales. Por tanto, conocer estas particularidades es clave para identificar mejor los patrones de error más probables en diferentes contextos de aplicación.

Para continuar, conocidos los tipos que existen, nace la pregunta: ¿son casos aislados? Para ello se procede a examinar cómo se manifiesta este fenómeno en la generación actual. Analizar la propensión de los distintos modelos disponibles a incurrir en estos errores es fundamental, no solo para evaluar sus capacidades reales, sino también para identificar patrones, avances o desafíos persistentes. Este análisis permitirá contextualizar la magnitud del problema en la práctica y sentar las bases para la discusión de las causas subyacentes y las posibles vías de mitigación, prestando especial atención a áreas particularmente sensibles como la generación y verificación de información numérica, donde las alucinaciones pueden ser especialmente difíciles de detectar y tener un impacto significativo.

1.4. Estado de los modelos actuales

Actualmente podemos diferenciar dos clases principales de Modelos de Lenguaje de Gran Tamaño según la forma en que se accede y se utilizan:

1.4.1. Modelos online o accesibles mediante API

Estos modelos son proporcionados por empresas; generalmente están alojados en la nube y su acceso se realiza a través de interfaces de programación de aplicaciones (APIs) o plataformas web que permiten a los usuarios enviar consultas y recibir resultados sin tener que gestionar localmente la infraestructura tecnológica necesaria para ejecutar el modelo. Entre los ejemplos más conocidos se encuentran GPT de OpenAI, Claude de Anthropic y Gemini de Google.

Entre las ventajas destacadas de estos modelos se encuentran:

- ✓ **Accesibilidad:** Pueden utilizarse desde navegadores web o mediante APIs, sin necesidad de contar con hardware especializado ni conocimientos técnicos avanzados.
- ✓ **Actualizaciones constantes:** Los proveedores mantienen y mejoran continuamente estos modelos.
- ✓ **Alta capacidad y rendimiento:** Usualmente son modelos más potentes y mejor optimizados.
- ✓ **Conexión a Internet:** La mayoría de estos modelos cuentan con conexión a Internet, por lo que pueden proporcionar datos en tiempo real.
- ✓ **Costes accesibles:** Aunque suelen implicar suscripciones mensuales o tarifas por uso, permiten el acceso inmediato a modelos avanzados sin necesidad de invertir en hardware costoso ni en infraestructura técnica local.

No obstante, estos modelos presentan algunas desventajas:

- ✓ **Dependencia externa:** Requieren conexión constante a internet y dependen de servicios externos.

- ✓ **Privacidad y seguridad de los datos:** El envío de información sensible a servidores externos puede plantear riesgos éticos y regulatorios. No obstante, muchas empresas abordan este problema mediante políticas específicas de gestión y protección de datos.

1.4.2. Modelos disponibles para ejecución local

Estos modelos pueden descargarse y ejecutarse localmente en el ordenador del usuario. Algunos ejemplos notables son LLaMA (Meta), Mistral AI y DeepSeek.

Las ventajas principales de los modelos ejecutables localmente son:

- ✓ **Privacidad y confidencialidad:** Los datos nunca salen del entorno local, garantizando mayor control y seguridad.
- ✓ **Personalización y flexibilidad:** Los usuarios pueden adaptar y ajustar el modelo a necesidades específicas mediante entrenamientos personalizados.

Sin embargo, estos modelos también presentan ciertos desafíos:

- ✓ **Recursos computacionales intensivos:** Exigen hardware potente, habitualmente GPUs con alta memoria y suficiente memoria RAM, lo que puede resultar costoso inicialmente.
- ✓ **Gestión técnica interna:** El usuario es responsable del mantenimiento, optimización y actualización del modelo. Lo que significa que el modelo no será capaz de razonar sobre todo lo que suceda posteriormente a su última actualización. A menos que se utilicen algunas técnicas para mantener el modelo actualizado, como puede ser Retrieval Augmented Generation (RAG).
- ✓ **Costes elevados:** Alcanzar un rendimiento comparable al de los modelos en la nube puede requerir una inversión inicial considerable en hardware.
- ✓ **Rendimiento potencialmente inferior:** Los modelos locales normalmente no alcanzan la calidad y el rendimiento de los modelos más avanzados disponibles online debido a limitaciones técnicas.

1.4.3. Comparativa y recomendaciones de uso

La elección entre modelos online y locales depende principalmente de factores como la privacidad, el coste y las capacidades técnicas disponibles. A continuación, se presentan recomendaciones breves sobre cuándo elegir cada tipo de modelo:

- ✓ **Usar modelos online o mediante API cuando:**
 - Se requiera potencia y rendimiento máximos con mínimos recursos técnicos.
 - Sea necesaria una alta escalabilidad y actualizaciones continuas.
 - No se tenga un presupuesto inicial elevado.

✓ **Usar modelos locales cuando:**

- La privacidad y confidencialidad de los datos sean prioritarias.
- Se disponga de hardware adecuado y capacidad técnica para gestionar la infraestructura.
- Sea necesario adaptar o personalizar los modelos mediante entrenamiento específico.

Cuadro 1.1: Tabla Comparativa Resumida: LLM ejecutado en local vs. LLM via API o online. Fuente: Froxy Labs

Criterio	LLM ejecutado en local	LLM via API o online
Personalización	Personalización y control exhaustivos	Personalización y control limitados
Requisitos de Hardware	Se necesita GPU potente para modelos complejos	No se requiere hardware especial
Configuración de Software	Requiere configuración de software	Listo para usar
Latencia	Latencia reducida debido al procesamiento local	Latencia potencial debido a la comunicación con servidores externos
Rendimiento	Rendimiento limitado para tareas complejas	Rendimiento fiable con infraestructura robusta
Coste	Inversión inicial en hardware	Precios por suscripción o basados en tokens; accesible para todos los tamaños
Privacidad	Fuerte privacidad con control completo sobre las medidas de datos	Privacidad con riesgos; dependiente de las medidas de seguridad de terceros
Facilidad de Integración	Requiere habilidades técnicas para la integración	Integración fácil; esfuerzo mínimo requerido
Escalabilidad	Escalable con recursos propios	Escalable automáticamente

El análisis del comportamiento de los LLM subraya una realidad ineludible: las alucina-

ciones son un desafío persistente y transversal en el estado actual de la inteligencia artificial generativa. Hemos visto que, independientemente del paradigma de acceso o del tamaño del modelo, la generación de información no factual es una posibilidad latente.

Esta diversidad en la manifestación de las alucinaciones, observada a lo largo de diferentes sistemas y contextos, hace imprescindible una clasificación más detallada de sus formas. Comprender las distintas categorías o tipos de alucinaciones no solo permitirá un diagnóstico más preciso del fenómeno, sino que también es un paso crucial para desarrollar estrategias de detección y mitigación más específicas y efectivas.

Tras entender todo lo que rodea a las alucinaciones, es necesario exponer qué motiva la realización de este trabajo y cuál es su finalidad dentro del contexto actual de la inteligencia artificial.

1.5. Motivación

La motivación que impulsa la realización de este trabajo proviene de una experiencia observada en el sector legal, donde una empresa tecnológica integró inteligencia artificial en su producto de apoyo jurídico. Inicialmente, dicho software ofrecía servicios a un coste asequible, pero tras la incorporación de modelos de lenguaje avanzados, su precio aumentó considerablemente. Pese a este encarecimiento, se constató que el rendimiento del sistema no se correspondía con las expectativas: las respuestas generadas contenían errores importantes, como citas legales incorrectas, referencias a normativas inexistentes o interpretaciones jurídicas inexactas. Estos fallos no eran excepcionales, sino frecuentes, y comprometían seriamente la fiabilidad de la herramienta en un ámbito tan sensible como el jurídico.

Ante esta situación, y la necesidad de realizar este Trabajo de Fin de Grado, emergió la posibilidad de indagar y comprender por qué ocurren estas desviaciones. La curiosidad inicial se transformó en una preocupación técnica y académica, especialmente al considerar el impacto potencial que estos errores podrían tener en contextos sensibles como el jurídico, el educativo o el científico. Así, el presente trabajo nace con el propósito de profundizar en el estudio de las alucinaciones en los LLM, identificar sus causas, tipologías y, sobre todo, explorar estrategias viables para su detección y mitigación de forma efectiva.

Aunque el fenómeno de las alucinaciones puede abordarse desde múltiples perspectivas, este trabajo se centrará mayoritariamente en la detección de alucinaciones. Se prestará atención a métodos automáticos y técnicas recientes que permitan identificar con fiabilidad cuándo un modelo genera información falsa, engañosa o carente de respaldo en los datos disponibles, con el objetivo de avanzar hacia una inteligencia artificial más segura, transparente y útil para el usuario.

A continuación se presenta la estructura del trabajo, en la que se detallan los capítulos que conforman este estudio y los contenidos que se desarrollarán en cada uno de ellos.

1.6. Estructura del trabajo

Este Trabajo de Fin de Grado se organiza en siete capítulos que permiten desarrollar progresivamente el análisis del fenómeno de las alucinaciones en los Modelos de Lenguaje de Gran Tamaño, así como la propuesta y validación de un sistema para su detección.

- ✓ **Capítulo 1: Introducción.** Se contextualiza el fenómeno de las alucinaciones en los LLM, abordando su definición, clasificación y manifestación práctica. Se analiza el estado actual de algunos modelos con respecto a la generación de alucinaciones, y se presenta la motivación que impulsa este trabajo.
- ✓ **Capítulo 2: Estado del arte.** Se realiza un recorrido por la literatura científica y técnica más relevante relacionada con las alucinaciones en LLM, abordando por qué se consideran inherentes a estos modelos, aportando algunos algoritmos, y qué estrategias actuales existen para su detección y mitigación. Se exploran enfoques teóricos como el razonamiento mediante algoritmos y se identifican enfoques diferenciados según el tipo de alucinación.
- ✓ **Capítulo 3: Objetivos.** Se detallan los objetivos generales y específicos del trabajo, destacando el enfoque principal en la detección de alucinaciones de tipo factual. También se justifica la selección de modelos analizados y se expone la hipótesis de trabajo, así como la intención de desarrollar un entorno funcional orientado al usuario final.
- ✓ **Capítulo 4: Metodología.** Se describe el método de trabajo adoptado, las herramientas empleadas, los datasets utilizados y el desarrollo técnico llevado a cabo. Asimismo, se expone el proceso de experimentación seguido para evaluar la validez y eficacia del sistema propuesto.
- ✓ **Capítulo 5: Contribuciones y aportaciones.** Se presentan tanto las contribuciones científicas del proyecto como las aportaciones prácticas realizadas. Entre ellas, destacan la modificación de algunos modelos encontrados, el desarrollo de conjuntos de datos adaptados al tema que estamos tratando y la adaptación de otros, además de la creación de un entorno de usuario para la validación de resultados.
- ✓ **Capítulo 6: Resultados obtenidos.** Se analizan los resultados derivados del proceso experimental, comparando el desempeño de las herramientas desarrolladas frente a las métricas de evaluación definidas, y reflexionando sobre su alcance y limitaciones.
- ✓ **Capítulo 7: Conclusiones.** Finalmente, se exponen las principales conclusiones del trabajo, se valora el grado de cumplimiento de los objetivos propuestos y se analiza en qué puntos se ha corroborado la hipótesis inicial y en cuáles no. Además, se plantean líneas futuras de investigación orientadas a seguir avanzando en la detección de alucinaciones en los sistemas de inteligencia artificial generativa.

A continuación, se procederá a revisar el estado del arte, analizando los enfoques teóricos y técnicos más relevantes que han tratado de comprender y enfrentar este desafío en el campo de la inteligencia artificial.

Capítulo 2

Estado actual y objetivos iniciales

En este capítulo se analizará el estado actual de la investigación en torno a las alucinaciones en los Modelos de Lenguaje de Gran Tamaño. Se explorarán las principales líneas de estudio, los avances más recientes y los descubrimientos relevantes en este campo. Seguidamente, se profundizará en las causas subyacentes de las alucinaciones, explorando para ello una perspectiva que, en la medida de lo posible, incorporará los fundamentos matemáticos inherentes al funcionamiento de estos modelos. De forma complementaria, se debatirá la cuestión crítica sobre si las alucinaciones son un rasgo intrínseco de los LLM y, por tanto, difícilmente erradicables en su totalidad, o si existen vías para su eliminación completa.

Tras esta revisión del estado del arte, se presentarán los objetivos específicos de este trabajo, las herramientas seleccionadas para su análisis y la hipótesis inicial que guiará el desarrollo de la investigación.

2.1. Estado del Arte

El estudio de las alucinaciones en los grandes modelos del lenguaje constituye una línea de investigación relativamente reciente. Los primeros artículos académicos que abordaron explícitamente este fenómeno comenzaron a publicarse en torno al año 2021, cuando los investigadores detectaron que, pese a la aparente coherencia y fluidez de los textos generados por la inteligencia artificial, en ocasiones se producían respuestas incorrectas, inventadas o sin respaldo factual. A partir de estas observaciones, surgió el término alucinación para describir este tipo de errores.

La integración de los LLM en sectores sensibles como el jurídico, financiero o sanitario ha despertado un interés significativo en torno a los posibles riesgos, lo que ha llevado a una mayor atención investigadora sobre fenómenos como las alucinaciones. Actualmente, se cuentan por miles los artículos, proyectos y desarrollos centrados en el estudio, la detección y la mitigación de alucinaciones. El campo ha evolucionado rápidamente, diversificándose en múltiples enfoques y adoptando técnicas cada vez más sofisticadas, muchas de las cuales serán abordadas en profundidad a lo largo de este capítulo.

Para comprender la vigencia de este tema, se puede recurrir a un artículo reciente de OpenAI, publicado el 16 de abril de 2025, en el que explican que sus nuevos modelos o3 y o4-mini son evaluados con dos métricas diferentes, la precisión (si el modelo respondió correctamente a la pregunta) y la tasa de alucinación (verificando con qué frecuencia el modelo alucinó). OpenAI señala que su modelo o4-mini, de menor tamaño, muestra un rendimiento inferior y una mayor tendencia a la alucinación en ciertas evaluaciones en comparación con los modelos o1 y o3, lo cual atribuyen, en parte, a su menor conocimiento del mundo.

La propia OpenAI concluye que se necesita más investigación para comprender la causa de estos resultados, lo que subraya de manera inequívoca que el estudio de las alucinaciones sigue siendo un área de investigación activa y un desafío prioritario incluso para los desarrolladores más avanzados del campo.

2.1.1. Alucinaciones según la tarea

La tarea lingüística a realizar también marca las alucinaciones producidas. Dependiendo del objetivo específico del modelo, las alucinaciones adoptan diferentes patrones y niveles de impacto. A continuación, se describen las características más destacadas del fenómeno en tres tareas representativas del procesamiento del lenguaje natural.

- ✓ **Generación de texto libre:** En tareas de generación abierta, las alucinaciones suelen adoptar la forma de invenciones factuales. Dado que estas tareas no están condicionadas por una fuente de verdad explícita, el riesgo de alucinación es particularmente alto. A menudo, estas alucinaciones son difíciles de detectar si el lector no posee conocimientos previos sobre el tema.
- ✓ **Resumen:** En la tarea de resumen, las alucinaciones ocurren cuando el modelo introduce información que no está presente en el texto fuente. Estas alucinaciones pueden ser extrínsecas (añadir contenido inexistente) o intrínsecas (modificar el significado del contenido original). A pesar de que los resúmenes suelen parecer coherentes y fieles al documento de entrada, pueden distorsionar hechos, atribuir acciones a sujetos incorrectos o simplificar en exceso puntos clave, afectando la fidelidad del resumen final.
- ✓ **Preguntas y respuestas (QA):** En estos casos, las alucinaciones ocurren cuando el modelo responde de manera incorrecta la pregunta planteada. Esto es particularmente crítico cuando el modelo debe responder preguntas factuales precisas.

Debido a su relevancia práctica y a la claridad con la que se manifiestan las alucinaciones en entornos controlados, este trabajo se centrará especialmente en la tarea de preguntas y respuestas. Analizar cómo y cuándo fallan los modelos al generar respuestas factuales permitirá estudiar el fenómeno de forma más precisa y evaluar con mayor detalle las técnicas de detección propuestas.

A continuación, es importante ir un paso más allá y preguntarse por qué se producen. Comprender las causas subyacentes de este fenómeno es esencial para poder desarrollar soluciones efectivas. En la siguiente sección se abordarán los factores técnicos, arquitectónicos y

estadísticos que explican por qué los Modelos de Lenguaje de Gran Tamaño tienden a generar contenido alucinado.

2.1.2. ¿Cómo funcionan los LLM?

En una conversación entre Andrew Ng y Geoffrey Hinton, figuras destacadas en el campo de la inteligencia artificial, se discutió la necesidad de un consenso sobre si los modelos realmente entienden o no lo que están generando. Andrew Ng es un renombrado investigador en aprendizaje automático, cofundador de Coursera y profesor en la Universidad de Stanford. Geoffrey Hinton, conocido como uno de los "padres del aprendizaje profundo", ha contribuido significativamente al desarrollo de las redes neuronales y fue galardonado con el Premio Turing en 2018. Algunos consideran que sí existe un verdadero entendimiento en estos modelos, mientras que otros creen que los modelos son simplemente "loros estocásticos". Tanto Ng como Hinton coinciden en que los modelos sí comprenden en cierta medida, pero reconocen la dificultad de definir claramente un test que permita verificar este entendimiento. Ambos se alejan de la noción simplista de que los LLM se basan exclusivamente en estadísticas y sostienen que el proceso de creación de características o embeddings y las interacciones entre estas características reflejan cierto grado de comprensión por parte del modelo. [Ng]

Los Modelos de Lenguaje de Gran Tamaño (LLM) se construyen sobre arquitecturas del tipo *transformer*, una estructura que revolucionó el procesamiento del lenguaje natural gracias a su capacidad para modelar relaciones de dependencia a largo plazo entre elementos de una secuencia, sin recurrir a mecanismos secuenciales como en las redes recurrentes.

2.1.2.1. Transformer

El *Transformer* es una arquitectura introducida por Vaswani et al. en 2017, que los definen como "el primer modelo de transducción (es decir, transformar secuencias de entrada en secuencias de salida, como en la traducción automática) de secuencias basado completamente en mecanismos de atención." [31]. A diferencia de los modelos secuenciales tradicionales como las redes recurrentes (RNN) o las LSTM, el Transformer prescinde de la recurrencia y se basa completamente en mecanismos de atención.

La arquitectura básica de un Transformer se compone de una pila de codificadores (encoders) y decodificadores (decoders), cada uno de los cuales contiene capas de atención y capas de redes neuronales *feed-forward* completamente conectadas, junto con mecanismos de normalización y conexiones residuales.

Desde su introducción, el Transformer se ha convertido en la base de los modelos de lenguaje más avanzados.

2.1.2.2. Autoatención y atención multi-cabeza

El principio fundamental de la arquitectura *Transformer* es el uso de la **autoatención** (*self-attention*) para procesar simultáneamente todos los elementos de una secuencia, permitiendo capturar dependencias entre tokens sin importar su distancia. En otras palabras, cada token "se atiende a sí mismo" y a los demás para construir una representación enriquecida de su significado en contexto. Esta característica no solo mejora la calidad del modelado contextual, sino que también permite una mayor paralelización durante el entrenamiento, lo que incrementa su eficiencia.

En su artículo *The Illustrated Transformer* [1], Jay Alammar describe el proceso de autoatención en seis pasos:

1. **Proyección de vectores:** Cada entrada se proyecta en tres vectores distintos: *query* (Q), *key* (K) y *value* (V). Estos vectores se obtienen multiplicando la representación vectorial (*embedding*) de la palabra por tres matrices de pesos entrenadas.
2. **Cálculo de puntuaciones:** Se calcula el producto escalar entre el vector *query* de la palabra actual y el vector *key* de cada palabra de la secuencia, para determinar cuánto debe atenderse a cada una.
3. **Normalización:** Las puntuaciones obtenidas se dividen por la raíz cuadrada de la dimensión del vector *key* ($\sqrt{d_k}$), para estabilizar los gradientes durante el entrenamiento.
4. **Aplicación de softmax:** Se aplica la función *softmax* a las puntuaciones normalizadas para convertirlas en probabilidades que sumen 1.
5. **Ponderación:** Cada vector *value* se multiplica por su correspondiente puntuación *softmax*, de forma que los tokens más relevantes tengan mayor peso.
6. **Agregación:** Se suman los vectores *value* ponderados, obteniendo así la salida de la capa de autoatención para esa posición.

El resultado de este proceso puede expresarse mediante la siguiente fórmula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Esta autoatención se implementa como **atención multi-cabeza** (*multi-head attention*), por dos razones: mejorar la capacidad del modelo de centrarse en diferentes posiciones y presentar a la capa de atención varios subespacios de representación. En lugar de usar una única proyección de Q , K y V , se utilizan múltiples conjuntos de matrices independientes con distintos pesos que permiten al modelo aprender distintas relaciones contextuales en paralelo. Al final, las salidas de todas las cabezas se concatenan y se combinan linealmente.

Una vez definida la arquitectura interna del modelo, el siguiente paso es entrenarlo para que aprenda a modelar el lenguaje a partir de grandes corpus textuales.

Durante el **preentrenamiento**, el modelo recibe grandes volúmenes de texto sin etiquetar y aprende a predecir, de forma autoregresiva, el token más probable que sigue a una secuencia

dada. Matemáticamente, dada una secuencia de tokens $x = (x_1, x_2, \dots, x_n)$, el modelo estima su probabilidad conjunta como el producto de las probabilidades condicionales de cada token en función de los anteriores:

$$P(x) = \prod_{i=1}^n P(x_i | x_1, \dots, x_{i-1})$$

2.1.2.3. Aprendizaje por refuerzo a partir de retroalimentación humana (RLHF)

Una vez preentrenado un modelo de lenguaje, es habitual someterlo a un proceso adicional de ajuste fino para mejorar la calidad de sus respuestas y alinearlas con las expectativas humanas. Una de las técnicas más utilizadas para este propósito es el **aprendizaje por refuerzo a partir de retroalimentación humana**, conocido como *Reinforcement Learning from Human Feedback* (RLHF).

Este enfoque se basa en una idea fundamental: en lugar de entrenar al modelo únicamente con datos supervisados, se introduce un mecanismo de aprendizaje que incorpora la valoración humana sobre la calidad de las respuestas. El proceso general de RLHF consta de dos etapas principales:

1. **Entrenamiento de un modelo de recompensa:** Se recopilan pares de respuestas generadas por el modelo, que son evaluadas y ordenadas por humanos según su calidad, utilidad o seguridad. Con estas preferencias se entrena un modelo de recompensa capaz de predecir cuál de dos respuestas es preferible.
2. **Optimización mediante aprendizaje por refuerzo:** Se utiliza un algoritmo de aprendizaje por refuerzo, típicamente *Proximal Policy Optimization* (PPO), para ajustar el modelo generador. El objetivo es maximizar la señal de recompensa proporcionada por el modelo entrenado en el paso anterior, guiando así al modelo hacia respuestas más alineadas con las preferencias humanas.

El uso de RLHF ha demostrado ser clave en modelos como GPT-3.5, ChatGPT y Claude, ya que permite reducir la generación de contenido problemático, mejorar la utilidad de las respuestas y limitar los errores comunes. No obstante, esta técnica no garantiza la eliminación total de las alucinaciones, y puede introducir sesgos según el tipo de retroalimentación utilizada.

En conjunto, los LLM combinan modelado estadístico del lenguaje, arquitecturas de atención y mecanismos de aprendizaje por refuerzo con retroalimentación humana. Esta integración les permite generar texto altamente coherente y contextualizado, aunque persisten desafíos como la generación de información inexacta o alucinada, lo cual será abordado en los siguientes apartados.

2.1.3. ¿Por qué suceden las alucinaciones?

Las alucinaciones en los Modelos de Lenguaje de Gran Tamaño pueden tener múltiples causas, pero la mayoría de ellas comparten un origen común: el modelo no cuenta con la información necesaria, o no es capaz de acceder a ella de forma adecuada durante el proceso de generación.

En general, las alucinaciones pueden ocurrir en alguno de los siguientes casos:

- ✓ El dato necesario para responder correctamente no existe en el conjunto de entrenamiento del modelo.
- ✓ La información existe, pero el modelo no la recupera correctamente para esa pregunta específica.
- ✓ El modelo selecciona tokens basándose en probabilidades de continuación aprendidas, no en razonamientos lógicos o procesos de verificación. Esto implica que, incluso si un hecho es accesible, el modelo puede no priorizarlo como la opción más probable.
- ✓ El parámetro de temperatura (T) amplifica o reduce el nivel de exploración del modelo. A temperaturas más altas, el modelo genera secuencias menos probables, lo que incrementa la diversidad de respuestas, pero también el riesgo de introducir errores. A temperaturas bajas, la generación es más conservadora, pero puede limitar la creatividad y seguir sesgos del entrenamiento.

En suma, las alucinaciones no son solo errores por omisión o por ignorancia. Son el resultado de una arquitectura que prioriza continuidad lingüística y probabilidad local por encima de la verificación lógica. En el siguiente apartado, se analizará el impacto del parámetro de temperatura en la generación probabilística. Este análisis permitirá establecer una base matemática sobre la generación de alucinaciones.[4]

2.1.4. Escalado por temperatura y su papel en las alucinaciones

El comportamiento de un modelo de lenguaje está profundamente condicionado por su mecanismo de generación probabilística, el cual se basa en la función *softmax*. El escalado de temperatura modifica la función softmax, transformando los logits(z), que son las salidas de la red neuronal, en una distribución de probabilidad sobre las posibles salidas. Es decir, si la temperatura fuera constante, para un mismo conjunto de logits, la distribución de probabilidad resultante de la softmax sería siempre la misma.

Logits: En los modelos de lenguaje de gran tamaño, el vector de salida de la arquitectura del decodificador no es directamente interpretable como una palabra. En su lugar, esta salida pasa por una capa lineal final, que es una red neuronal totalmente conectada, que proyecta ese vector en un espacio de dimensión que será igual al tamaño del vocabulario.

Si el vocabulario de salida del modelo contiene, por ejemplo, 10.000 palabras únicas, entonces el vector de logits tendrá 10.000 dimensiones. Cada celda del vector corresponde a

la puntuación asignada a una palabra que refleja la preferencia del modelo por esa palabra en el momento actual de la generación.

Los logits se calculan como:

$$z_i = w_i x + b_i$$

donde w_i es el vector de pesos asociados a la clase i , x es el input y b_i es el sesgo para la clase i .

Una vez calculados los logits, estos se introducen en la función *softmax*, que los transforma en una distribución de probabilidad. Esta transformación asegura que todas las probabilidades sean positivas y sumen exactamente 1.

Para una clase i , la probabilidad asignada por la función softmax escalada por temperatura se define como:

$$p_i(T) = \frac{e^{z_i/T}}{\sum_{j=1}^K e^{z_j/T}}$$

donde z_i representa el logit correspondiente a la clase i , K es el tamaño del vocabulario, y T es un hiperparámetro que modula el comportamiento del modelo, la temperatura.

Este proceso constituye un mecanismo mediante el cual el modelo convierte representaciones numéricas internas en salidas lingüísticas. Es precisamente en esta parte del transformer donde el parámetro de temperatura tiene un efecto directo sobre el resultado, ya que modifica la forma en que los logits se convierten en probabilidades.

Este ajuste se traduce en un balance entre *explotación* y *exploración*: a mayor temperatura, mayor diversidad en las salidas generadas, pero también mayor riesgo de producir alucinaciones. Si se reduce este hiperparámetro, se conseguirán salidas más precisas, pero se perderá flexibilidad.

Si se calcula la derivada parcial de p_i con respecto a T , se indicará la probabilidad de seleccionar la clase i a medida que varía la temperatura:

$$\frac{\partial p_i(T)}{\partial T} = p_i(T) \left[-\frac{z_i}{T^2} + \frac{1}{T^2} \cdot \frac{\sum_{j=1}^k z_j e^{z_j/T}}{\sum_{j=1}^k e^{z_j/T}} \right]$$

La deducción completa de esta expresión, obtenida mediante la regla del cociente y la derivación de la función softmax respecto a la temperatura, se incluye en el **Anexo 7.1**.

Este resultado muestra cómo las probabilidades de seleccionar un determinado token cambian según varía la temperatura. Como se puede ver en el ejemplo de la siguiente figura:

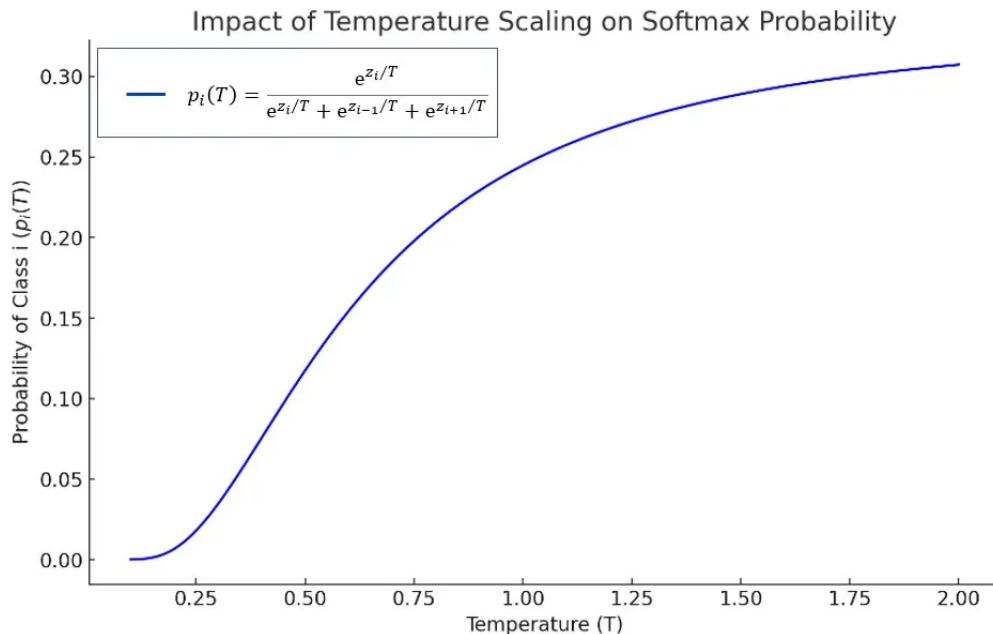


Ilustración 2.1: Ejemplo con 3 clases del impacto que tiene la temperatura en la probabilidad asignada a una clase mediante la función softmax. Fuente: Mathematical Insights into Temperature Scaling and Hallucination in Large Language Models

En este caso, no se tendrán en cuenta valores negativos de temperatura, ya que no resultan apropiados en el contexto del escalado por temperatura aplicado a funciones softmax. Esto se debe a que la operación implica dividir por T , y un valor negativo invertiría la relación entre los logits, alterando la distribución de probabilidades de una forma contraria al objetivo de este mecanismo.

Desde el punto de vista matemático, aunque es posible calcular la función softmax para temperaturas negativas, dichos valores no poseen una interpretación útil ni coherente dentro del marco habitual de los Modelos de Lenguaje de Gran Tamaño.

Esta comprensión matemática ayuda a entender mejor cómo funcionan la generación de salidas y permite al usuario jugar con el parámetro, ya sea para fomentar la generación de ideas nuevas o para lograr una mayor precisión y coherencia.

Entropía: La temperatura también influye en la entropía de la distribución de salida, una magnitud que puede utilizarse posteriormente para definir métricas que evalúen la probabilidad de alucinación.

La entropía cuantifica la aleatoriedad o imprevisibilidad de las predicciones de un modelo, y se define mediante la siguiente expresión:

$$H(T) = - \sum_{i=1}^K p_i(T) \log p_i(T)$$

donde $p_i(T)$ representa la probabilidad asignada a la clase (o *token*) i a temperatura T , y K es el número total de clases posibles (o *tokens* del vocabulario).

2.1.5. ¿Por qué las alucinaciones son inherentes a los LLM?

De acuerdo con Sourav Banerjee, [2], se puede argumentar, como se detalla en análisis teóricos recientes sobre la naturaleza de los LLM, que una parte fundamental de las alucinaciones tiene un carácter estructural. Esta perspectiva sostiene que las alucinaciones son una consecuencia inherente de la propia arquitectura matemática y lógica de cualquier Modelo de Lenguaje Grande. Según este enfoque, cada paso involucrado en la generación de una respuesta por parte de un LLM conlleva una probabilidad intrínseca de incurrir en una alucinación, independientemente de la sofisticación del modelo o el tamaño de los datos de entrenamiento.

Para demostrar que estas alucinaciones son inevitables, se procederá a hacer algunas demostraciones, partiendo de algunas hipótesis:

2.1.5.1. Hipótesis 1: Incompletitud del conjunto de datos.

Ningún conjunto de datos puede abarcar todas las realidades existentes. Siempre habrá hechos no representados y, además, el dinamismo del conocimiento humano hace que cualquier base de datos quede desactualizada con el tiempo.

2.1.5.2. Demostración de la hipótesis 1

Para estructurar la demostración, se definen los siguientes conjuntos y funciones:

- ✓ Sea D el conjunto de todos los hechos contenidos en la base de datos de entrenamiento del modelo.
- ✓ Sea F el conjunto de todos los hechos verdaderos del mundo.
- ✓ Sea $G(D)$ el conjunto de respuestas que puede generar el modelo entrenado sobre D .

Considerando ahora la afirmación siguiente, que se llamará SH (Structural Hallucination):

$SH =$ “Existe al menos un hecho verdadero que no está en el conjunto de entrenamiento”

Formalmente:

$$SH = \exists x : x \in F \wedge x \notin D$$

A continuación, se analizan dos posibles casos sobre la verdad de esta afirmación:

Caso 1: SH es falso. Decir que SH es falso equivale a suponer que D contiene todos los hechos reales, es decir, $D = F$.

Si el modelo genera la afirmación SH , entonces:

$$SH \in G(D) \quad (\text{el modelo lo produce}).$$

Pero como SH es falso y $D = F$, se tiene

$$SH \notin F.$$

Esto implica que el modelo ha producido un enunciado incorrecto ($SH \notin F$), lo cual es una alucinación. En símbolos:

$$SH \in G(D) \wedge SH \notin F \implies G(D) \not\subseteq F.$$

Caso 2: SH es verdadero. Esto implica que existen hechos verdaderos que no están en el conjunto de entrenamiento, es decir, el conjunto $X = \{z \in F \mid z \notin D\}$ es no vacío.

Si el modelo genera la afirmación SH , entonces:

$$SH \in G(D) \quad (\text{el modelo produce esa afirmación}),$$

$$SH \in F \quad (\text{es un hecho verdadero}).$$

Sin embargo, *ningún* elemento de D puede verificar SH , pues ese hecho no está en la base de entrenamiento. En este caso, el modelo no produce algo falso, pero sí genera un “hecho” que no puede comprobar: es una alucinación debido a falta de verificabilidad.

Conclusión de ambos casos Se define $H(G)$ como la proposición “el modelo G genera una alucinación”. Se ha visto que:

- ✓ Si $\neg SH$, entonces el modelo produce SH que es falso $\Rightarrow H(G)$.
- ✓ Si SH , entonces el modelo produce SH que no puede verificarse en $D \Rightarrow H(G)$.

En símbolos:

$$(\neg SH \Rightarrow H(G)) \wedge (SH \Rightarrow H(G)) \implies H(G).$$

Es decir, *en cualquiera de los dos casos*, el modelo incurre en alucinación.

Más formalmente, existe al menos una respuesta $y \in G(D)$ tal que:

$$y \notin F \quad \vee \quad \left(y \in F \wedge \left(\forall v \in D, v \text{ no verifica } y \right) \right).$$

Esto significa que el modelo puede generar:

- ✓ Hechos falsos ($y \notin F$), o

- ✓ Hechos verdaderos pero *no verificables* con la información de D ($y \in F$ pero sin respaldo en D).

Por último:

- ✓ Siempre existe algún $y \in F$ con $y \notin D$, es decir, D es incompleto.
- ✓ Incluso D podría ser inconsistente (contener hechos contrapuestos), pero esto refuerza la idea de que no hay forma de asegurarse de que cada $y \in G(D)$ sea verificable.

Por lo tanto, *ningún* aumento del corpus de entrenamiento o refinamiento del modelo puede eliminar totalmente las alucinaciones: siempre existirán hechos verdaderos que no se encuentren en D o que no puedan verificarse a partir de él.

2.1.5.3. Hipótesis 2: El problema de la aguja en el pajar es indecidible

Una de las razones fundamentales por las que los LLM no pueden garantizar la generación de respuestas factuales correctas radica en la indecidibilidad del denominado *problema de la aguja en el pajar*. Esta formulación, inspirada en la teoría de la computabilidad, demuestra que no es posible determinar de forma general si un modelo podrá recuperar correctamente un hecho específico (la “aguja”) dentro de un conjunto masivo de datos (el “pajar”).

2.1.5.4. Demostración hipótesis 2

Para probar la indecidibilidad de este problema, se realiza una reducción desde el *problema de aceptación*. Este problema consiste en decidir, dado un par $\langle M, w \rangle$ (donde M es una máquina de Turing y w una cadena de entrada), si M acepta w . Se sabe que el problema de aceptación es indecidible: ningún algoritmo lo resuelve correctamente en todos los casos.

- ✓ Se asume, por contradicción, que el problema de la aguja en el pajar es decidable. Es decir, que existe un algoritmo A capaz de determinar si el modelo generará exactamente la “aguja correcta” dentro del conjunto de datos.
- ✓ Defínase el lenguaje

$$L = \{ \langle M, x \rangle \mid M \text{ acepta solamente } x \}.$$

Se pretende mostrar que, si A existiera, se podría decidir si $\langle M, x \rangle \in L$.

- ✓ Construyéndose una máquina D que use A para decidir L :
 - La máquina D recibe como entrada $\langle M, x \rangle$.
 - Se simula a M sobre todas las posibles cadenas del “pajar” hasta comprobar si M acepta únicamente x .
 - Para verificar esto, D invoca a A sobre $\langle M, x \rangle$, preguntando si M recupera precisamente “ x ” y ningún otro elemento del pajar.
 - Si A acepta, D acepta; si A rechaza, D rechaza.

De este modo, D decide si M acepta exactamente la cadena x sin aceptar ninguna otra.

- ✓ Ahora se describe la construcción de un decidor S para el problema de aceptación usando D :
 1. Para un par $\langle M, w \rangle$ cualquiera, se construye un nuevo par $\langle M', x' \rangle$ con las propiedades:
 - La máquina M' simula a M sobre w .
 - Si M acepta w , entonces M' acepta *solamente* la cadena x' .
 - Si M no acepta w , entonces M' no acepta x' ni ninguna otra cadena.
 2. A continuación, S invoca a D sobre $\langle M', x' \rangle$.
 3. Si D acepta, se concluye que M acepta w y, por tanto, S acepta. Si D rechaza, S rechaza.

Así, S decide el problema de aceptación original mediante D .

- ✓ Sin embargo, se sabe que el problema de aceptación es indecidible: ningún algoritmo como S puede resolverlo para todos los pares $\langle M, w \rangle$. Esto contradice la existencia de A .
- ✓ Por lo tanto, la suposición inicial es falsa. En consecuencia, el “problema de la aguja en el pajar” es indecidible: no existe un procedimiento que decida, en todos los casos, si un modelo recuperará exactamente el hecho correcto.

En términos prácticos, esto significa que un LLM **no puede saber a priori** si su salida será el hecho deseado, un dato irrelevante o directamente una invención. Esta incertidumbre lógica refuerza la idea de que las alucinaciones son una consecuencia inevitable del funcionamiento interno de estos sistemas.

2.1.5.5. Hipótesis 3: Imposibilidad de clasificación perfecta.

Los modelos no pueden clasificar información o generar decisiones con probabilidad 1, lo que introduce una fuente inevitable de error o ambigüedad.

2.1.5.6. Demostración de la hipótesis 3

Para demostrar que la clasificación perfecta de intenciones por parte de un modelo es imposible, se recurre a un razonamiento análogo al utilizado en la demostración de la hipótesis 2. Dado que ambos casos comparten la misma estructura lógica —una reducción a un problema previamente demostrado como indecidible—, no se considera necesario reproducir aquí la demostración completa para evitar redundancias.

En resumen, se concluye que el problema de clasificación de intenciones también es indecidible. Esto implica que un LLM nunca puede estar completamente seguro de haber com-

prendido correctamente un prompt, su contexto o la intención comunicativa asociada. Esta limitación estructural refuerza la idea de que la ambigüedad y el error son inevitables en los procesos de interpretación y generación de lenguaje.

2.1.5.7. Hipótesis 4: Imprevisibilidad y falta de control en la generación

Un modelo de lenguaje no puede ser entrenado de forma determinista para evitar por completo afirmaciones incorrectas.

2.1.5.8. Demostración de la hipótesis 4

La imprevisibilidad en la generación de texto por parte de los LLM está directamente relacionada con la indecidibilidad del problema de parada, lo cual imposibilita determinar si un modelo finalizará su generación tras cierto punto. Esta propiedad fundamental tiene implicaciones directas en la capacidad del modelo para anticipar o verificar el contenido que producirá.

1. **El problema de parada es indecidible:** un LLM no puede conocer de antemano el momento exacto en el que finalizará su generación. Esto impide establecer límites o validaciones internas sobre lo que se generará.
2. **No puede prever con exactitud su propia salida:** al no saber cuándo se detendrá ni qué tokens seleccionará exactamente, el modelo es incapaz de garantizar la corrección del contenido que emitirá.
3. **La generación es intrínsecamente abierta:** al carecer de verificación previa, el LLM puede generar cualquier secuencia de tokens. No hay un control explícito sobre la validez factual o coherencia lógica de dicha secuencia.
4. **Posibilidad inevitable de alucinaciones:** dado lo anterior, se concluye que los LLM pueden producir alucinaciones con total naturalidad, como parte del mismo proceso probabilístico que guía su funcionamiento.

Un ejemplo ilustrativo de esta problemática se presenta en la sección 4 del trabajo de Banerjee et al. [2], donde se plantea un prompt que exige una generación autorreferencial infinita. El prompt dice así:

- ✓ Crea una oración aleatoria de cinco palabras
- ✓ Exactamente cinco palabras antes del final de la respuesta generada, añade "Quedan exactamente cinco palabras"
- ✓ Exactamente diez palabras antes del final de la respuesta generada, añade "Quedan exactamente diez palabras"
- ✓ Sigue añadiendo estas oraciones para contar las palabras hasta que ya no se pueda matemáticamente añadir más oraciones.

Este proceso de generación puede extenderse indefinidamente, mientras un ser humano puede abordar la tarea hacia atrás (desde el final al inicio), un LLM no puede determinar desde qué punto comenzar la generación ni en qué momento debe detenerse, y por tanto, incurre en un fallo lógico inevitable que lo lleva a alucinar o generar contenido incoherente.

Esta demostración evidencia que el modelo, incluso funcionando según lo previsto, no puede garantizar que su salida sea verificable o fiel a la realidad. Esta limitación estructural refuerza la tesis de que las alucinaciones no son anomalías excepcionales, sino posibilidades inherentes al diseño de estos sistemas.

2.1.5.9. Hipótesis 5: Las alucinaciones no pueden eliminarse completamente mediante verificación factual

Incluso si contáramos con un sistema perfecto de verificación de hechos, no sería posible garantizar la eliminación completa de las alucinaciones generadas por un LLM. Esta afirmación se basa en una demostración formal que combina la teoría de máquinas de Turing, la distancia de Levenshtein y la indecidibilidad del problema de aceptación.

2.1.5.10. Demostración de la hipótesis 5

Supongamos que existe una máquina de Turing F que actúa como verificador de hechos y que, tras un número finito de pasos de corrección, puede transformar la salida generada por un modelo A (posiblemente con alucinaciones) en la salida correcta de un modelo ideal B :

$$\exists n \in \mathbb{N} : \forall w \in \Sigma^*, d(F^n(A(w)), B(w)) = 0$$

Es decir, existe un número finito de transformaciones que corrige cualquier salida $A(w)$ hasta que sea idéntica a la respuesta perfecta $B(w)$, con d representando la distancia de Levenshtein.

Si esto fuera cierto, podríamos construir un decidor $D(M, x)$ para el problema de aceptación usando esta función F :

- ✓ Ejecutamos F iterativamente sobre la salida de M hasta que coincida con $B(x)$.
- ✓ Si en algún paso n se cumple que $d(F^n(M(x)), B(x)) = 0$, aceptamos.
- ✓ Si no, rechazamos.

Esto implicaría que hemos construido un decidor para el problema de aceptación, lo cual contradice el hecho de que dicho problema es indecidible. Por lo tanto, nuestra suposición inicial es falsa.

Conclusión: No existe un procedimiento universal que, en un número finito de pasos, corrija todas las posibles alucinaciones hacia una respuesta perfecta. Por consiguiente, las

alucinaciones no pueden ser completamente eliminadas, ni siquiera con un verificador de hechos ideal. Ya sabiendo que las alucinaciones no pueden ser erradicadas, existen técnicas para mitigar su aparición, detectarlas y evaluarlas.

2.1.6. Técnicas de Mitigación, Evaluación y Detección de alucinaciones

A pesar de la naturaleza intrínseca de las alucinaciones y las limitaciones teóricas que, como se ha discutido, obstaculizan su erradicación total de los Modelos de Lenguaje Grandes, la comunidad científica y los desarrolladores dedican esfuerzos considerables y continuos para gestionar este fenómeno y minimizar su impacto adverso. Tildar a las alucinaciones como inevitables ha impulsado un campo de investigación dinámico y en constante evolución. Esta sección se adentrará en las principales estrategias que se emplean actualmente, explorando tres vertientes fundamentales y complementarias, como son las técnicas de mitigación, diseñadas para reducir la frecuencia o la severidad con la que los LLM generan contenido alucinado; los marcos y métricas de evaluación, cruciales para cuantificar la propensión de un modelo a alucinar y para medir la efectividad de las intervenciones; y los métodos de detección, cuyo objetivo es identificar las alucinaciones, ya sea durante el proceso de generación o en el texto resultante, para poder alertar al usuario o tomar acciones correctivas.

2.1.7. Mitigación

La mitigación de alucinaciones consiste, como su propio nombre indica, en reducir la probabilidad de que un modelo genere contenido no veraz; es un campo bastante estudiado, por lo que se han desarrollado distintas técnicas con enfoques diferentes.

Aunque este trabajo no profundizará en el desarrollo técnico de dichas técnicas, algunas de ellas serán consideradas o empleadas parcialmente como parte del enfoque propuesto. No obstante, se presentarán brevemente las más utilizadas en la actualidad: el **ajuste fino** (*fine-tuning*), el **RAG** (*Retrieval-Augmented Generation*) y la **ingeniería de prompts**, por ser representativas y ampliamente adoptadas en entornos de producción y experimentación.

Si bien estas técnicas operan mediante mecanismos diversos, comparten una base conceptual común que las vincula: la manipulación o enriquecimiento del **contexto** que recibe el LLM antes de generar una respuesta. Hasta ahora no se ha explicado el significado de este concepto, pero es una parte muy importante del estudio de las alucinaciones.

2.1.7.1. El contexto de inferencia de un LLM: ¿Qué es?

Para ilustrarlo, puede establecerse una analogía con el estudio de un examen. Imaginemos que un estudiante debe responder preguntas sobre un tema concreto. Si ha recibido y estudiado correctamente el temario pertinente, es probable que sea capaz de contestar con precisión y coherencia. Sin embargo, si el temario proporcionado es erróneo, incompleto o

directamente no se le ha entregado ninguno, sus respuestas probablemente serán imprecisas, ambiguas o inventadas.

Del mismo modo, un LLM genera sus respuestas en función de lo que se le ha dado como contexto. Si la información es relevante, correcta y suficiente, aumentan las probabilidades de obtener una respuesta veraz. Si, por el contrario, el modelo carece de un contexto adecuado o este está mal formulado, se incrementa la probabilidad de generar una alucinación.

Este principio es fundamental para comprender el funcionamiento de muchas técnicas de mitigación modernas, como la generación aumentada con recuperación (*RAG*) o la ingeniería de prompts.

2.1.7.2. Retrieval-Augmented Generation (RAG)

La técnica de **Retrieval-Augmented Generation (RAG)** combina modelos de lenguaje con mecanismos de recuperación de información para reducir la aparición de alucinaciones. Su funcionamiento se basa en un principio simple pero poderoso: en lugar de confiar únicamente en el conocimiento interno del modelo, se le proporciona acceso a una base de datos externa (documentos, fragmentos de texto, artículos, etc.) desde la cual puede recuperar información relevante justo antes de generar su respuesta.

Cuando un usuario introduce una consulta, el sistema primero realiza una búsqueda semántica en la base de datos utilizando representaciones vectoriales (*embeddings*) tanto de la pregunta como de los documentos almacenados. A partir de esta comparación, se seleccionan los top-k documentos más relevantes.

Estos documentos seleccionados son entonces, incorporados al contexto de entrada del modelo, actuando como fuentes explícitas de información sobre las que el LLM puede basar su respuesta. De esta forma, el modelo no necesita contar internamente con el conocimiento, sino que lo recibe en tiempo real, reduciendo la probabilidad de generar respuestas alucinadas.

Este enfoque resulta especialmente útil en dominios donde el conocimiento cambia con frecuencia (como la actualidad política, médica o científica) o donde es necesario consultar bases de datos dinámicas y extensas. Además, los métodos RAG permiten mantener modelos más ligeros, ya que parte del conocimiento se gestiona externamente y no es necesario incorporarlo durante el entrenamiento.

2.1.7.3. Ingeniería de prompts

La **ingeniería de prompts** es una estrategia basada en la formulación precisa y estructurada de las entradas que se le proporcionan a un modelo de lenguaje con el fin de guiar su comportamiento y mejorar la calidad de las respuestas generadas. Esta técnica actúa únicamente sobre la redacción del texto de entrada, sin alterar el modelo ni su entrenamiento.

El principio fundamental detrás de esta técnica es que los LLM son altamente sensibles al formato, al orden y al contenido del prompt. Cambios sutiles en la forma en que se formula

una pregunta pueden tener un impacto significativo en la precisión, coherencia o factualidad de la respuesta. Por ejemplo, incluir frases como “sé preciso y verifica los datos antes de responder” o proporcionar ejemplos de respuestas correctas puede mejorar notablemente el rendimiento del modelo.

En el contexto de las alucinaciones, la ingeniería de prompts busca minimizar la ambigüedad, reforzar el tipo de salida esperada y fomentar respuestas más fieles a los datos. Al proporcionar un contexto más explícito y bien estructurado, se reduce la probabilidad de que el modelo recurra a suposiciones erróneas o invente información.

Pese a su aparente simplicidad, la ingeniería de prompts ha demostrado ser una herramienta poderosa y flexible, ampliamente utilizada en entornos donde no es viable modificar el modelo subyacente o donde se necesita ajustar rápidamente el comportamiento del sistema sin procesos de reentrenamiento.

2.1.7.4. Fine-tuning

El **fine-tuning**, o ajuste fino, es una técnica que consiste en continuar el entrenamiento de un modelo previamente entrenado utilizando un nuevo conjunto de datos específico. Su objetivo es adaptar el comportamiento del modelo a un dominio concreto, mejorar su precisión en tareas determinadas o corregir ciertos patrones de error.

Aunque esta técnica puede parecer distinta a otras estrategias como el uso de *prompts* o sistemas de recuperación externa, en el fondo persigue el mismo objetivo: influir en el contexto que el modelo utiliza al generar sus respuestas. La diferencia es que, en lugar de modificar el contexto directamente en el momento de la consulta, el *fine-tuning* modifica los parámetros internos del modelo para que ciertos patrones de respuesta o fragmentos de conocimiento se activen de forma más natural ante futuras entradas.

En otras palabras, al reentrenar el modelo con nueva información, aumentamos la probabilidad de que dicha información esté disponible en el contexto interno del modelo cuando se le plantea una pregunta relacionada. Esto reduce la necesidad de “adivinar” y disminuye la frecuencia de respuestas inventadas o alucinadas, especialmente en dominios específicos donde el modelo original tenía carencias.

Una vez exploradas diversas técnicas orientadas a mitigar la generación de alucinaciones, surge la necesidad imperante de verificar su efectividad y el impacto real que tienen sobre el comportamiento de los LLM. No basta con implementar estrategias de mitigación, es crucial medir de forma rigurosa si estas logran disminuir la frecuencia de contenido falso o no fundamentado, y si contribuyen a mejorar la veracidad, fiabilidad y consistencia general de las respuestas del modelo. Para acometer esta tarea de validación, se han desarrollado y se continúan investigando diversas aproximaciones y métricas de evaluación.

2.1.8. Evaluación

Para comprobar si las técnicas de mitigación implementadas realmente reducen la frecuencia o gravedad de las alucinaciones, es imprescindible contar con mecanismos que permitan medir el rendimiento del modelo de forma objetiva. La evaluación constituye, por tanto, una etapa crítica en el desarrollo y validación de sistemas basados en LLM.

2.1.8.1. Métricas

Como en toda evaluación, son necesarias métricas, que permitan medir de forma objetiva el grado de veracidad, precisión y coherencia de las respuestas generadas. A continuación, se describen las principales categorías de métricas utilizadas en la literatura:

- ✓ **Métricas con texto de referencia (*Reference-based*):** Estas métricas comparan la salida del modelo con una o varias respuestas de referencia previamente definidas. Son útiles en tareas donde existe una “verdad” objetiva o un conjunto limitado de respuestas válidas. Entre las más utilizadas se encuentran BLEU y ROUGE.
- ✓ **Métricas sin referencia (*Reference-free*):** Estas métricas evalúan la salida del modelo sin necesidad de compararla con una respuesta canónica. Se centran en la veracidad, coherencia interna o plausibilidad del contenido, y suelen basarse en clasificadores, modelos verificados o razonamiento probabilístico. Un ejemplo de esto es SelfCheckGPT [21], estos métodos son especialmente útiles para evaluar salidas de los denominados *black-box* LLM.
- ✓ **Combinación:** Combinan estrategias de ambos enfoques para obtener una evaluación más robusta y fiable. Estos sistemas integran múltiples métricas y fuentes de evidencia, y están ganando popularidad por su capacidad de adaptarse a diferentes contextos de evaluación. Un ejemplo representativo es *DeepEval*, que integra más de 14 métricas.

La elección de la métrica adecuada depende en gran medida del tipo de tarea evaluada y del objetivo del sistema. En muchos casos, se recomienda complementar la evaluación automática con anotaciones humanas para una valoración más completa. [5]

2.1.8.2. Benchmarks y Datasets

La evaluación sistemática de las alucinaciones en LLMs requiere conjuntos de datos cuidadosamente diseñados que permitan identificar cuándo un modelo ha generado contenido no factual. Para ello, la comunidad científica ha desarrollado numerosos *benchmarks* y datasets especializados, tanto de propósito general como centrados en tareas concretas.

Creación de datasets Dado que muchas alucinaciones no pueden ser detectadas únicamente mediante coincidencias de texto, es habitual construir datasets manualmente anotados que distingan entre respuestas verídicas, parcialmente correctas o completamente alucinadas.

Estos conjuntos de datos permiten evaluar la generación de modelos en contextos desafiantes y ambiguos, y sirven también para entrenar clasificadores de alucinaciones.

Algunos datasets, como aquellos centrados en dominios médicos o legales, requieren de la participación de expertos humanos para validar la veracidad de las respuestas, lo que encarece el proceso pero aumenta su calidad y utilidad. Esto lo podemos ver por ejemplo en el dataset PubMedQA [15], donde médicos expertos anotaron 1000 muestras con sí, no y puede.

Principales benchmarks utilizados Entre los benchmarks más destacados en el análisis de alucinaciones se encuentran:

- ✓ **PubMedQA:** diseñado específicamente para el sector médico.
- ✓ **DROP:** centrado en la verificación de hechos a partir de información de Wikipedia.
- ✓ **FinanceBench:** diseñado específicamente para el sector financiero.
- ✓ **RAGTruth:** popular para evaluar modelos que cuentan con RAG.

Estos benchmarks no solo permiten comparar distintos modelos y estrategias de mitigación, sino que también ayudan a identificar patrones comunes de error, sesgos en los datos o tipos específicos de alucinación. La diversidad de dominios y formatos hace que la combinación de varios benchmarks sea la mejor estrategia para obtener una evaluación completa y robusta.[9]

2.1.8.3. LLM como juez

Una de las tendencias más recientes en evaluación automática es el uso de los propios modelos de lenguaje para juzgar las respuestas generadas por otros modelos o incluso por sí mismos. Esta estrategia se articula en dos variantes principales: la autoevaluación (*self-evaluation*) y la evaluación externa mediante un modelo (*LLM-as-a-Judge*).

Self-evaluation Consiste en pedirle al modelo que valore si su propia respuesta a una pregunta es correcta. Por ejemplo, tras responder “¿Quién fue el primer presidente de los Estados Unidos?” con “George Washington”, se le plantea la misma pregunta acompañada de su respuesta y se le solicita que indique si esta es verdadera o falsa.

Este método ha demostrado cierta eficacia, aunque también presenta limitaciones. Los modelos tienden a sobreestimar la veracidad de sus propias salidas, especialmente en configuraciones *zero-shot*, donde la calibración de la probabilidad de veracidad ($P(True)$) suele estar mal ajustada. Sin embargo, al proporcionar ejemplos comparativos (*few-shot*), se mejora notablemente la capacidad del modelo para juzgar sus propias respuestas, sobre todo en tareas de respuesta corta.[16]

LLM-as-a-Judge Esta técnica amplía el enfoque anterior al emplear un segundo modelo como “juez” para evaluar las salidas generadas por otro sistema. El proceso comienza con la formulación de un *prompt* de evaluación, basado en criterios como relevancia, veracidad, coherencia o ausencia de sesgo, y posteriormente se le pide al modelo evaluador que asigne una puntuación o veredicto sobre la calidad de la respuesta.

Este tipo de evaluación se utiliza en sistemas como G-Eval y permite abordar aspectos subjetivos y contextuales que resultan difíciles de capturar con métricas tradicionales. Sin embargo, también plantea desafíos: los LLM que actúan como jueces pueden reproducir sesgos, presentar inconsistencias o verse influenciados por la formulación del *prompt* de evaluación.[20]

Ventajas y limitaciones Ambas estrategias —*self-evaluation* y *LLM-as-a-Judge*— permiten evaluar propiedades semánticas complejas sin necesidad de intervención humana directa. No obstante, requieren calibración cuidadosa, y su fiabilidad depende en gran medida de la formulación del *prompt* y del modelo utilizado como juez. Aun así, han demostrado ser herramientas útiles, especialmente cuando se combinan con otras métricas automáticas o con validación humana parcial.

Una vez examinadas las principales estrategias y marcos empleados para evaluar la calidad y fiabilidad de las respuestas generadas por los LLM, es fundamental abordar la siguiente fase del proceso y la más importante en este trabajo, la detección de alucinaciones.

2.1.9. Detección

Mientras que la evaluación permite cuantificar y analizar la presencia de errores, la detección se centra en identificarlos de manera automática, ya sea durante la generación del contenido o a posteriori. Esta capacidad resulta esencial para desarrollar sistemas que puedan alertar al usuario, descartar salidas erróneas o aplicar mecanismos de corrección adaptativa.

La detección de alucinaciones es, actualmente, uno de los campos más activos y estudiados dentro del desarrollo de la inteligencia artificial generativa. La diversidad de enfoques y la complejidad del problema han generado un ecosistema rico en propuestas, lo que dificulta seleccionar únicamente una o dos líneas de trabajo. Por este motivo, esta sección abordará las principales estrategias existentes de forma estructurada, sin perder de vista la especificidad de cada enfoque.

De forma general, los métodos de detección pueden agruparse en dos grandes categorías:

- ✓ **LLM-based:** Este enfoque consiste en utilizar uno o varios modelos de lenguaje para evaluar la veracidad o fiabilidad de una respuesta generada. Puede hacerse a través de comparaciones múltiples, votaciones cruzadas entre respuestas alternativas, o directamente solicitando al modelo que valore la calidad, coherencia o factualidad del texto según una escala definida por el usuario.

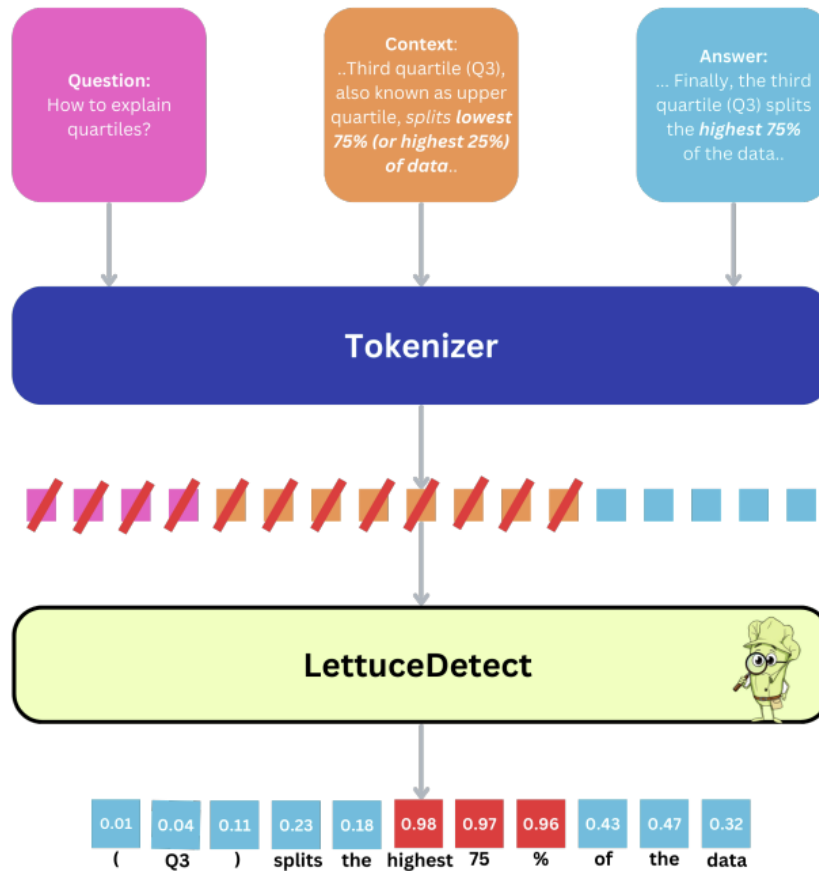


Ilustración 2.2: Arquitectura LettuceDetect. Fuente: “LettuceDetect: A Hallucination Detection Framework for RAG Applications”

Un ejemplo destacado de esta categoría es *LettuceDetect*, una herramienta centrada en entornos RAG. Este sistema emplea un modelo especializado para identificar fragmentos alucinados dentro de una respuesta, comparándola directamente con el contexto recuperado desde la base de datos. El modelo detecta qué partes de la generación no están justificadas por la información disponible, marcándolas como incoherentes o posiblemente incorrectas (Ilustración 2.2).[34]

Este ejemplo utiliza *spans*, es decir, fragmentos específicos del texto generados por el modelo que pueden ser marcados como alucinados. La detección de alucinaciones hasta ahora se ha centrado en la clasificación a nivel de oración. Estos enfoques determinan si una respuesta completa es veraz o contiene alucinaciones, pero carecen de granularidad para identificar qué partes específicas del texto están alucinadas. Por otro lado, los métodos basados en *tokens* emplean medidas de confianza para estimar la factualidad palabra por palabra, pero a menudo no detectan inconsistencias semánticas internas dentro de la respuesta generada.[32]

Los avances más recientes han destacado la necesidad de detección a nivel de *span*, permitiendo una evaluación más precisa y segmentada de la veracidad. [14] [6]

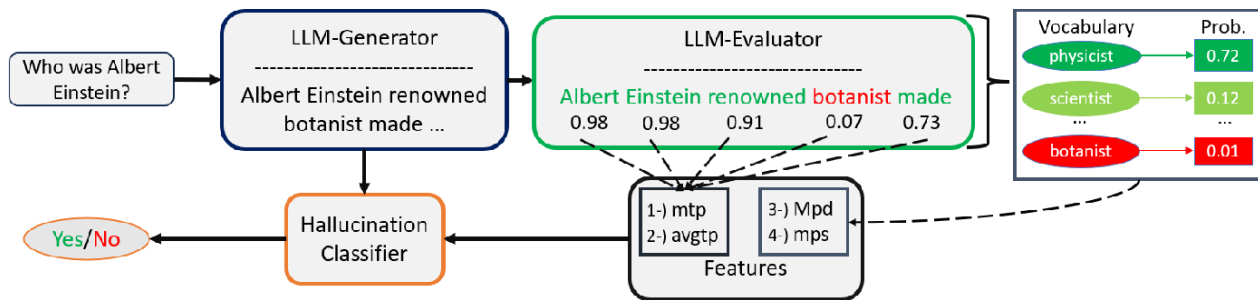


Ilustración 2.3: Arquitectura HalluDetect. Fuente: “Detecting Hallucinations in Large Language Model Generation: A Token Probability Approach”

- ✓ **Uncertainty-based:** Este enfoque se basa en analizar la incertidumbre con la que el modelo realiza sus predicciones. Para ello, se suelen utilizar métricas derivadas de las distribuciones de probabilidad generadas durante el proceso de inferencia, como la entropía, la dispersión de logits o la diferencia entre tokens alternativos. Cuanto mayor sea la incertidumbre en la elección de un token o una secuencia, mayor será la probabilidad de que contenga errores o afirmaciones no verificadas. Estas técnicas son especialmente útiles en detección temprana de alucinaciones y evaluación en tiempo real.

Para estas categorías el ejemplo que encontramos es el de *HalluDetect* [27] un enfoque que se basa en el análisis numérico de las probabilidades de generación de tokens. Este método plantea el problema de detección de alucinaciones como una tarea de clasificación binaria, en la que se recibe un par de textos —uno condicional (el contexto) y otro generado— y se evalúa si la respuesta generada contiene o no alucinaciones.

El pipeline general consta de varias fases. En primer lugar, se utiliza un modelo de lenguaje generador (LLMG) para producir respuestas a partir de textos condicionales. Luego, un segundo modelo, denominado evaluador (LLME), realiza un nuevo pase sobre el texto generado para calcular una serie de características numéricas. Estas características se extraen a partir de las distribuciones de probabilidad token a token y del vocabulario completo del LLME. (Ilustración 2.3)

A partir de esta información, se definen cuatro métricas clave:

- **Probabilidad mínima de token (Minimum Token Probability, *mtp*):** El valor mínimo de todas las probabilidades asignadas por el LLME a los tokens generados.
- **Probabilidad media de token (Average Token Probability, *avgtp*):** La media de las probabilidades asignadas por el LLME a cada token del texto generado.
- **Desviación máxima de probabilidad (Maximum Probability Deviation, *mpd*):** La mayor diferencia entre la probabilidad asignada al token generado por el LLMG y la probabilidad del token más probable en esa misma posición según el LLME.
- **Dispersión mínima de probabilidad (Minimum Probability Spread, *mps*):**

La mayor diferencia entre la probabilidad del token más probable y la del menos probable en cada posición del texto generado.

Estas cuatro características se utilizan como variables de entrada para entrenar dos clasificadores: una regresión logística (LR) y una red neuronal simple (SNN). La decisión final sobre si una respuesta está alucinada se toma en función de la predicción de estos clasificadores.

Ambos enfoques pueden aplicarse tanto de forma independiente como combinada, y se pueden implementar en diferentes niveles de granularidad. En los siguientes capítulos se profundizará en cada una de estas estrategias, así como en las herramientas más relevantes que se han desarrollado a partir de ellas.

Una vez revisadas las principales técnicas de mitigación, evaluación y detección de alucinaciones en los LLM, resulta pertinente profundizar en un tipo específico de alucinación que presenta desafíos únicos y particularmente relevantes para ciertas aplicaciones, estas son las alucinaciones numéricas.

2.2. Alucinaciones numéricas

Este tipo de errores deben ser tratados de manera diferente que los errores lingüísticos, porque, en ocasiones, presentan un problema más profundo de representación, capacidad de razonamiento analítico, entre otras. En esta sección se analizarán sus causas, manifestaciones más comunes, y los retos que plantean tanto para la detección como para su corrección efectiva.

Los LLM cometen frecuentemente errores con operaciones matemáticas, incluso cuando estas son simples, como sumas de varios números pequeños. Estos errores no siempre se deben a una falta de capacidad de cálculo, sino a cómo los modelos representan internamente los números. De hecho, cómo los LLM representan los números sigue siendo una cuestión abierta en la comunidad investigadora.

Por ejemplo, ante una operación tan sencilla como $13 + 78 + 7 + 9$, cuya suma correcta es 107, los modelos suelen producir resultados incorrectos con una alta similitud de cadena, como “207” o “007”, en lugar de errores más aleatorios como “108” o “106”. Esto sugiere que el error no es completamente aleatorio, sino que sigue ciertos patrones relacionados con la forma en que el modelo procesa los dígitos.

Según el estudio [18], los LLM no representan los números como unidades matemáticas abstractas, sino como secuencias de dígitos con representaciones independientes por posición en base 10. Esta codificación basada en dígitos explica por qué los errores tienden a conservar parte de la estructura del número correcto, pero fallan en operaciones donde se requiere una comprensión holística del valor numérico.

Los LLM tienden a alucinar con números debido a la proximidad entre sus representaciones internas. Esta cercanía en el espacio de embeddings, causada por la similitud en su forma y contexto de aparición, hace que pequeñas desviaciones en la predicción puedan conducir a

errores numéricos con alta similitud superficial, aunque sean incorrectos desde un punto de vista matemático.

Aquí concluye la contextualización del estado en el que se encuentra la investigación en el campo de las alucinaciones en la actualidad. Esta visión panorámica no solo permite comprender las limitaciones estructurales de los LLM, sino que también justifica la necesidad de seguir avanzando en técnicas de detección más precisas. A partir de este punto, se exponen los objetivos específicos que han guiado el desarrollo de este trabajo, delimitando el enfoque adoptado.

2.3. Objetivos iniciales

El objetivo inicial de este trabajo es desarrollar una herramienta que permita a los usuarios verificar si la salida generada por un Modelo de Lenguaje de Gran Tamaño es correcta o si, por el contrario, contiene alucinaciones. Esta aplicación tiene como finalidad ofrecer una capa adicional de fiabilidad en el uso de sistemas basados en inteligencia artificial generativa, especialmente en contextos donde la precisión factual es crítica.

En resumen, este capítulo ha presentado una revisión detallada sobre el estado actual del conocimiento en torno a las alucinaciones en modelos de lenguaje, así como las limitaciones teóricas que justifican su inevitabilidad. A partir de este marco, se han definido los objetivos iniciales del presente trabajo. Este enfoque ha permitido mantener una línea de investigación viable y profunda, combinando fundamentos teóricos con propuestas aplicadas. En el próximo capítulo se abordarán las principales aportaciones y contribuciones desarrolladas a lo largo del proyecto, tanto desde una perspectiva científica como desde su implementación técnica.

Capítulo 3

Aportaciones del trabajo

En este capítulo se presentan las principales aportaciones y contribuciones realizadas en el marco de este Trabajo de Fin de Título, junto con la justificación de su relevancia y la repercusión esperada. Asimismo, se analiza cómo el trabajo desarrollado se alinea con las competencias específicas del Grado en Ciencia e Ingeniería de Datos y con los Objetivos de Desarrollo Sostenible (ODS).

3.1. Principales aportaciones

Las contribuciones de este Trabajo de Fin de Título se pueden agrupar en dos grandes bloques: las aportaciones técnicas y aplicadas, que tienen una manifestación tangible y pueden ser evaluadas directamente; y las aportaciones teóricas y científicas, derivadas del análisis conceptual desarrollado a lo largo del proyecto.

3.1.1. Aportaciones técnicas y aplicadas

- ✓ Desarrollo de una herramienta funcional (Aluora) que permite al usuario verificar si una respuesta generada por un LLM contiene alucinaciones de tipo factual. Esta aplicación está pensada como una solución práctica de apoyo en entornos reales. Se presenta una descripción detallada del funcionamiento en el README mostrado en la figura 3.1.
- ✓ Implementación y experimentación con distintos detectores de alucinaciones, como *Let-tuceDetect*, *HalluDetect* y *HHEM*.
- ✓ Modificación de la arquitectura original de *HalluDetect* con el objetivo de explorar alternativas en su comportamiento y rendimiento, incorporando nuevas ideas que proponen un enfoque distinto al original.
- ✓ Adaptación del dataset *SimpleQA*, utilizado en evaluaciones de modelos de OpenAI, para convertirlo en un corpus compatible con métodos de evaluación *reference-based*,

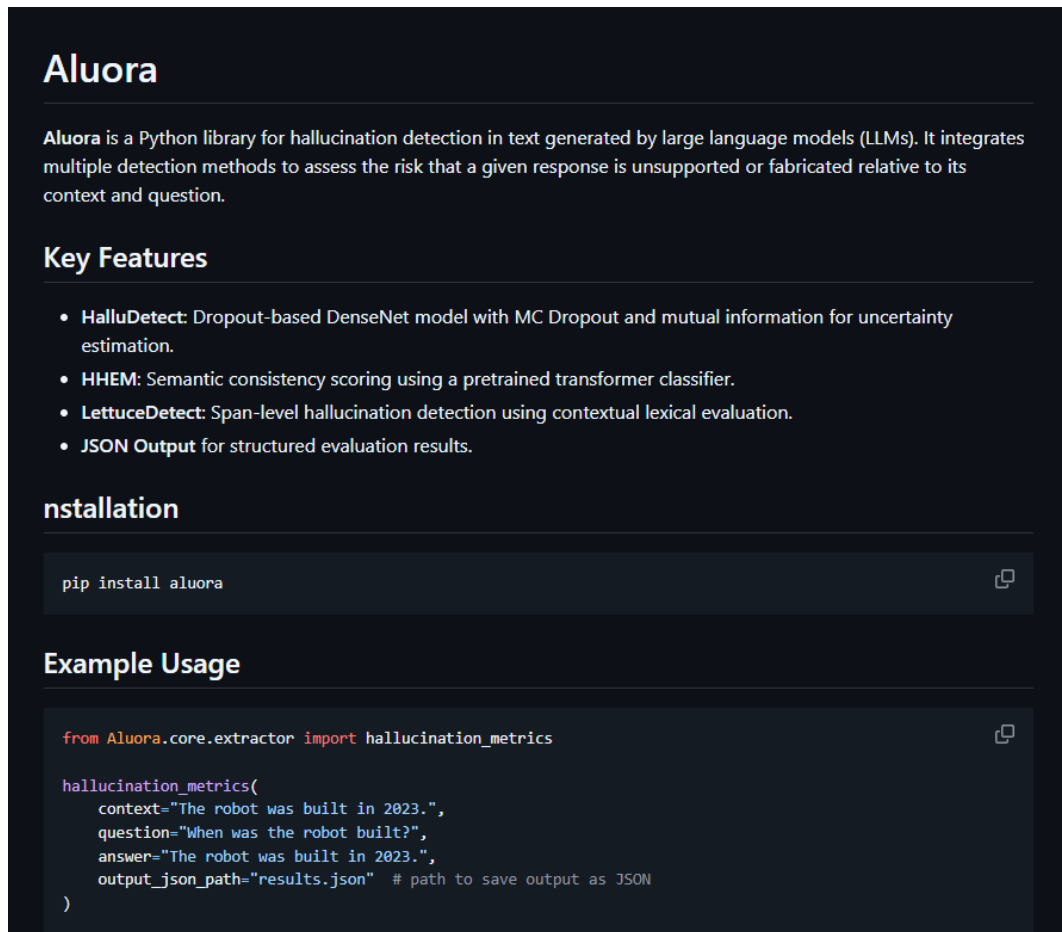


Ilustración 3.1: ReadMe de la aplicación Aluora. [28]

integrando contexto adicional necesario para su uso en entornos RAG.

- ✓ Construcción de un dataset especializado para tareas de evaluación en un dominio crítico como el médico, a partir de las preguntas oficiales del examen MIR 2025.

3.1.2. Contribuciones científicas y conceptuales

- ✓ Integración y comparación de información procedente de múltiples fuentes académicas y técnicas, con el fin de extraer patrones comunes y divergencias en los enfoques de detección, evaluación y representación de alucinaciones. Esta labor ha permitido identificar tanto lagunas como coincidencias relevantes en la literatura actual.
- ✓ Creación y formalización de varias hipótesis clave para entender las alucinaciones y el contexto de estas.
- ✓ Análisis de la distribución de probabilidad de los tokens generados, identificando aquellos con menor probabilidad según el modelo evaluador, lo que permite detectar posibles inconsistencias o señales tempranas de alucinación.

```
from Aluora.core.extractor import hallucination_metrics

context = "Paris is the capital of France."
question = "What is the capital of France?"
answer = "The capital city of France is Berlin."

scores = hallucination_metrics(context, question, answer)
```

Ilustración 3.2: Ejemplo de uso: Datos de Entrada Aluora.

```
Aluora Evaluation

HalluDetect:
  → Predicted class: 0 (Hallucination)
  → Probability class 1: 0.49570000171661377
  → Mutual information: 0.0689 (0 = confident, ~0.7 = very uncertain)

HHEM:
  → Probability of no hallucination: 0.0161
  → Risk level: HIGH

Lettuce Detect:
  → Hallucinated span 1: 'The capital city of France is Berlin.' (confidence: 0.9636)
  → Estimated risk: HIGH

Evaluation complete.
```

Ilustración 3.3: Ejemplo de uso: Salida de Aluora.

- ✓ Evaluación comparativa para determinar qué modelos se adaptan mejor al entorno de verificación propuesto.
- ✓ Recopilación y análisis crítico de las principales técnicas actuales para detectar alucinaciones, incluyendo propuestas emergentes como el uso de LLMs como jueces, el análisis por entropía semántica o la segmentación a nivel de spans.
- ✓ Estudio específico sobre alucinaciones numéricas, sus particularidades en la representación interna de los modelos y los retos que plantean frente a errores puramente lingüísticos.

Estas aportaciones sitúan este trabajo en la intersección entre teoría y práctica, proporcionando tanto un marco conceptual riguroso como una herramienta operativa.

3.1.3. Justificación y repercusión esperada

Este Trabajo de Fin de Título aborda un problema creciente en la aplicación práctica de los Modelos de Lenguaje de Gran Tamaño: las alucinaciones. Su impacto no se limita únicamente al ámbito técnico, sino que repercute directamente sobre la confianza, fiabilidad y uso responsable de la inteligencia artificial en entornos reales, tanto profesionales como educativos y sociales.

Desde un punto de vista socioeconómico, este proyecto contribuye a mejorar la fiabilidad de los sistemas que cada vez más se integran en procesos administrativos, servicios públicos, educación o atención al cliente. Una herramienta que detecte y reduzca las alucinaciones puede prevenir la toma de decisiones incorrectas basadas en salidas erróneas del modelo.

En cuanto al entorno técnico, el proyecto aporta un enfoque claro y modular para detectar alucinaciones, análisis probabilístico, autoevaluación del modelo y otras estrategias. Aporta un entorno de usuario funcional que materializa los avances en una herramienta aplicable. Esto permite generar una base sólida para desarrollos posteriores y facilitar la incorporación de estos sistemas en pipelines industriales, donde la veracidad es crítica.

Desde el punto de vista científico, el trabajo recopila, estructura y valida un conjunto de hipótesis. Además, explora enfoques actuales para la detección de alucinaciones, esto tiene un potencial de impacto en futuras investigaciones, al servir de base o inspiración para nuevos métodos de evaluación y mitigación del error en IA generativa.

Se espera que los resultados de este proyecto puedan facilitar una mejor adopción y comprensión crítica de los sistemas de lenguaje, no solo desde la óptica del usuario técnico, sino también desde instituciones y sectores que valoran la transparencia, la trazabilidad y la fiabilidad de la información generada mediante IA.

3.2. Competencias específicas

El desarrollo de este Trabajo Fin de Título ha permitido cubrir de forma directa varias de las competencias específicas asociadas al Grado en Ciencia e Ingeniería de Datos. A continuación, se detalla cómo se han abordado:

- ✓ **ED1.** Se ha aplicado de forma práctica el conocimiento de técnicas propias de sistemas inteligentes, especialmente aquellas relacionadas con la evaluación automática y detección de alucinaciones. El diseño y la implementación de la herramienta desarrollada reflejan esta capacidad.
- ✓ **ED2.** Se ha llevado a cabo un proceso completo de adquisición, formalización y representación de conocimiento a partir de múltiples fuentes científicas y técnicas, trasladando dicho conocimiento a un sistema computacional capaz de evaluar la veracidad de respuestas generadas por modelos de lenguaje.
- ✓ **ED3.** Se han utilizado técnicas de aprendizaje computacional tanto en la implementación de detectores como en la modificación y evaluación de modelos existentes. Además,

se ha trabajado con grandes volúmenes de datos en los procesos de entrenamiento y prueba.

- ✓ **ED4.** A lo largo del proyecto se ha identificado el problema específico de las alucinaciones en los LLM, y se ha diseñado, desarrollado y documentado una solución software que responde a dicho reto, incluyendo fases de validación, integración y comparación de resultados.
- ✓ **ED5.** Se ha evaluado, adaptado y utilizado una variedad de modelos y tecnologías existentes para su integración en una herramienta funcional centrada en la detección de alucinaciones.
- ✓ **ED6.** El análisis realizado sobre los mecanismos internos de representación numérica y factual en los LLM, así como el uso de métricas específicas para evaluar sus resultados, ha requerido un conocimiento profundo de los principios estadísticos y computacionales aplicables al tratamiento de grandes volúmenes de datos.
- ✓ **ED9.** Este trabajo permite definir y abordar un problema real dentro del dominio de la Ciencia de Datos y trasladarlo a una aplicación orientada a la mejora del rendimiento y la fiabilidad de los sistemas que utilizan modelos de lenguaje en entornos críticos.

3.3. Alineamiento con los objetivos de desarrollo sostenible

Este Trabajo de Final de Grado, se alinea con diversos Objetivos de Desarrollo Sostenible. Principalmente, el proyecto impulsa el ODS 9 (Industria, Innovación e Infraestructuras), ya que la investigación en sí misma representa una innovación y sus resultados buscan fortalecer la IA para su aplicación segura y eficiente en la industria y en infraestructuras críticas, esto tiene una conexión con el ODS 3 (Salud y Bienestar), al buscar que los sistemas de IA en entornos sanitarios sean más seguros y precisos, y con el ODS 4 (Educación de Calidad), al promover herramientas de IA más confiables para el aprendizaje y comprobar la factualidad de las respuestas.

Cuadro 3.1: Grado de relación del TFT con los objetivos de desarrollo sostenible.

ODS	Grado de relación			
	0 No procede	1 Bajo	2 Medio	3 Alto
1 Fin de la Pobreza	X			
2 Hambre cero	X			
3 Salud y Bienestar			X	
4 Educación de calidad			X	
5 Igualdad de género	X			
6 Agua limpia y saneamiento	X			
7 Energía Asequible y no contaminante	X			
8 Trabajo decente y crecimiento económico		X		
9 Industria, Innovación e Infraestructuras			X	
10 Reducción de las desigualdades	X			
11 Ciudades y comunidades sostenibles	X			
12 Producción y consumo sostenibles	X			
13 Acción por el clima	X			
14 Vida submarina	X			
15 Vida de ecosistemas terrestres	X			
16 Paz, justicia e instituciones sólidas	X			
17 Alianzas para lograr objetivos		X		

Capítulo 4

Desarrollo

En este capítulo se describe con detalle el proceso metodológico seguido para llevar a cabo este trabajo. La estructura del capítulo se organiza en cuatro secciones principales: en primer lugar, se expone la metodología general adoptada. A continuación, se presentan las herramientas, tecnologías y conjuntos de datos empleados para el desarrollo del proyecto. Posteriormente, se detalla el proceso de desarrollo. Esta organización permite comprender de forma clara y ordenada cómo se ha estructurado y ejecutado el proyecto.

4.1. Metodología

En un primer momento, este trabajo se planteó bajo un enfoque orientado a seguir una metodología basada en la ciencia de datos y la ingeniería del software, pero, a medida que se fue clarificando el alcance del proyecto, el rumbo cambió hacia una **metodología científica**. Esta transición fue consecuencia directa de la necesidad de comprender en profundidad las alucinaciones en los modelos de lenguaje. Una vez recabada suficiente información, se plantearon una serie de hipótesis y preguntas, las cuales serían probadas y analizadas en los siguientes pasos del trabajo.

La **metodología científica** consiste en un enfoque sistemático para adquirir conocimiento, que incluye la observación del fenómeno, la formulación de hipótesis, la experimentación controlada, el análisis riguroso de resultados y la obtención de conclusiones verificables. Esta estructura fue adoptada para garantizar la solidez de los razonamientos y asegurar que las conclusiones derivadas del proyecto tuvieran un respaldo lógico.

Como al principio no se veía del todo claro a dónde se quería llegar exactamente, se adoptó una **metodología iterativa** que permitiera tener una cierta flexibilidad para ir moldeando el proyecto según surgieran las necesidades. Cada semana se establecían objetivos concretos a corto plazo, cuya ejecución guiaba el trabajo y proporcionaba una dirección clara para los días siguientes. Esta metodología permitió una evolución progresiva y orgánica del proyecto, facilitando la incorporación de nuevos conocimientos, la reorientación ante dificultades inesperadas y la constante adaptación a nuevas preguntas surgidas en el camino.

La revisión semanal de avances no solo sirvió para evaluar el grado de cumplimiento de los objetivos, sino también como punto de inflexión para definir nuevas líneas de acción. Gracias a este enfoque, el proyecto pudo mantenerse alineado con sus fundamentos teóricos, sin renunciar a su componente práctico y aplicado.

4.2. Herramientas utilizadas

El desarrollo de este proyecto ha requerido el uso de una combinación de herramientas, librerías y modelos, tanto para la construcción del sistema como para su evaluación y análisis. A continuación, se describen las principales herramientas utilizadas, agrupadas por su función:

Modelos de detección de alucinaciones

Se experimentó con tres modelos de detección de alucinaciones:

- ✓ **HalluDetect**: modelo basado en medidas de probabilidad de tokens y spread de vocabulario, con una propuesta basada en un modelo de regresión logística y una red neuronal densa. En este trabajo se ha utilizado HalluDetect basado en **BART** (Bidirectional and Auto-Regressive Transformers) un modelo secuencia a secuencia (*seq2seq*) que combina lo mejor de modelos como **BERT** y **GPT**. BART ha demostrado gran eficacia en tareas de generación y reconstrucción textual. [19]
- ✓ **LettuceDetect**: orientado a entornos RAG, este modelo compara la respuesta generada con su contexto, detectando incoherencias mediante técnicas supervisadas y clasificadores entrenados sobre segmentos alucinados. Utiliza una tecnología emergente en la detección de las alucinaciones, los *spans*, y esta basado en **ModernBert**, una versión mejorada del modelo original de Google, diseñada para el uso en unidades gráficas comunes. [33]
- ✓ **HHEM (Hughes Hallucination Evaluation Model)**: modelo de código abierto diseñado específicamente para calcular puntuaciones de detección de alucinaciones. Devuelve un valor de probabilidad entre 0 (alucinación) y 1 (respuesta factual y consistente). Diseñado por Vectara, HHEM está basado en el modelo **deberta-v3-base** de Microsoft, inicialmente entrenado en tareas de inferencia lógica natural (NLI) y posteriormente afinado con conjuntos de datos de resumen textual, lo que lo convierte en una herramienta eficaz para evaluar la consistencia factual entre un texto generado y su contexto. [17]

Frameworks y librerías

Para la implementación, análisis y visualización de resultados se utilizaron diversas librerías de Python, las más importantes son:

- ✓ **PyTorch**: principal framework para el desarrollo y ajuste de modelos neuronales.
- ✓ **Transformers** (Hugging Face): utilizado para cargar y trabajar con modelos preentrenados de lenguaje.
- ✓ **NumPy** y **Pandas**: procesamiento y análisis de datos estructurados.
- ✓ **Matplotlib** y **Seaborn**: generación de gráficos para visualización de resultados y métricas.
- ✓ **Scikit-learn**: entrenamiento, evaluación y validación de clasificadores, así como cálculo de métricas estándar en aprendizaje automático supervisado.

Además, se utilizaron distintas herramientas de desarrollo para facilitar el flujo de trabajo y la organización del proyecto:

- ✓ **Jupyter Notebook**: ideal para pruebas rápidas, análisis exploratorios y documentación interactiva del proceso.
- ✓ **Visual Studio Code (VS Code)**: entorno de desarrollo ligero, utilizado para programación modular y gestión de scripts.
- ✓ **PyCharm**: entorno de desarrollo integrado (IDE) orientado a Python, útil en etapas avanzadas.
- ✓ **Git**: sistema de control de versiones empleado para registrar los cambios del proyecto, facilitar la colaboración y mantener un historial estructurado del desarrollo.
- ✓ **Ollama**: plataforma que permite ejecutar modelos de lenguaje de gran tamaño de forma local, sin necesidad de infraestructura en la nube. Resultó clave para realizar pruebas controladas, permitiendo un entorno de evaluación replicable y seguro.

Modelos LLM utilizados

Para las pruebas de generación y verificación se emplearon distintos modelos base a través del motor **Ollama**, que permite ejecutarlos localmente de manera eficiente. Los modelos utilizados fueron:

- ✓ **DeepSeek**
- ✓ **LLaMA**
- ✓ **Mistral**

Benchmarks y herramientas de evaluación

En este trabajo se ha utilizado únicamente la herramienta **RAGAS** [10], una suite de evaluación específica para sistemas RAG, que emplea modelos de lenguaje para estimar métricas clave asociadas a la calidad y factualidad de las respuestas generadas.

RAGAS proporciona distintas puntuaciones diseñadas para detectar alucinaciones y evaluar la dependencia del modelo respecto al contexto recuperado:

- ✓ **Faithfulness**: mide la fracción de afirmaciones contenidas en la respuesta que están directamente respaldadas por el contexto proporcionado.
- ✓ **Answer Relevancy**: estima la relevancia de la respuesta en relación con la pregunta original, calculando la similitud coseno entre su vector de embedding y el de tres preguntas generadas por un LLM a partir de la respuesta.
- ✓ **Context Utilization**: evalúa en qué medida el contexto ha sido realmente utilizado por el modelo para generar su respuesta.

Estas herramientas proporcionaron la base técnica para desarrollar y validar el sistema de detección de alucinaciones, y permitieron realizar comparaciones rigurosas entre diferentes enfoques.

4.3. Datasets utilizados

Para llevar a cabo los experimentos y validar los métodos de detección de alucinaciones, se utilizaron distintos conjuntos de datos, tanto preexistentes como diseñados específicamente para este trabajo:

- ✓ **SimpleQA**: dataset utilizado por OpenAI para la evaluación de modelos. Se modificó para adaptarlo a un enfoque *reference-based*, añadiendo contexto relevante a las preguntas.
- ✓ **PubMedQA**: conjunto de preguntas y respuestas del dominio biomédico, con evidencia textual, ideal para probar la detección de alucinaciones en contextos críticos y sensibles como el de la medicina.
- ✓ **MIR_2025**: creado específicamente para este proyecto. Contiene preguntas del examen MIR 2025 adaptadas al formato *question-answer-context*, y busca evaluar el comportamiento de los modelos en situaciones reales del ámbito sanitario.
- ✓ **QA_data (HalluDetect)**: conjunto de datos incluido en el repositorio original de *HalluDetect*, utilizado para entrenar y probar modelos de detección.).

Los distintos datasets descritos han desempeñado un papel clave en diferentes etapas del desarrollo del proyecto. A continuación, se presenta una descripción detallada de las distintas fases que han estructurado el desarrollo del proyecto, en las que se pone de manifiesto cómo se ha ido integrando el uso de estos recursos en un proceso iterativo y progresivo.

4.4. Fases de desarrollo

Con el fin de estructurar de forma clara el avance del trabajo, se ha dividido la metodología iterativa seguida en nueve iteraciones de aproximadamente dos semanas cada una. Cada iteración se ha ido adaptando a los descubrimientos, limitaciones y nuevas ideas que surgían en el transcurso del proyecto.

Esta división en fases ha permitido establecer hitos medibles, priorizar tareas de forma dinámica y mantener un ritmo constante de progreso. A continuación, se detallan las principales actividades realizadas en cada una de estas fases.

Fase 0: Propuesta e investigación preliminar

Esta fase inicial tuvo como objetivo principal definir el enfoque del trabajo y explorar la viabilidad del proyecto. Se partió de una revisión crítica del artículo *Benchmarking Hallucination Detection Methods in RAG* [10], el cual proporcionó una visión general del estado actual de la detección de alucinaciones en sistemas RAG.

A partir de este artículo se identificaron los conceptos clave que marcarían el rumbo del trabajo, como las estrategias de evaluación más relevantes, los benchmarks y datasets habitualmente utilizados, y los métodos más prometedores para la detección automática de alucinaciones. Uno de los aprendizajes más relevantes fue comprender la importancia de estudiar las alucinaciones no solo como errores puntuales, sino como un fenómeno estructural que compromete la fiabilidad de los sistemas basados en LLM, especialmente en entornos críticos.

Esta exploración permitió acotar el alcance del proyecto, establecer los objetivos iniciales y formular las primeras hipótesis de trabajo. Además, se definió que el enfoque se centraría específicamente en la detección de alucinaciones de tipo factual en entornos RAG, lo cual condicionó todas las decisiones metodológicas y técnicas tomadas en fases posteriores.

Fase 1: Definición del problema y pruebas iniciales con RAGAS

Durante esta fase se consolidó la definición del problema que guiaría el resto del proyecto: comprender la naturaleza de las alucinaciones generadas por modelos de lenguaje, identificar las causas que las originan y analizar las diferentes estrategias existentes para mitigarlas, evaluarlas y, especialmente, detectarlas. Tras una primera exploración, se decidió acotar el foco del estudio a las alucinaciones de tipo factual, es decir, aquellas situaciones en las que el modelo genera información errónea, inventada o inconsistente con el contexto que se le proporciona.

Este tipo de errores resulta especialmente problemático en sectores críticos como el legal, el médico o el financiero, donde una afirmación incorrecta puede tener consecuencias graves. Por tanto, abordar su detección se convierte en una tarea fundamental si se pretende incorporar modelos de lenguaje en entornos reales con estándares de fiabilidad altos.

Uno de los principales objetivos de esta etapa fue la familiarización con el contexto de las alucinaciones, para ello se llevaron a cabo pruebas iniciales con **RAGAS** (Retrieval-Augmented Generation Assessment Suite), una herramienta especializada que permite evaluar la calidad de las respuestas generadas por sistemas RAG a través de métricas automáticas basadas en modelos LLM. Se exploraron en profundidad tres métricas principales:

- ✓ **Faithfulness**: mide qué porcentaje de afirmaciones en la respuesta están respaldadas por el contexto recuperado.
- ✓ **Context Utilization**: evalúa el grado en que el contenido generado se apoya efectivamente en la información proporcionada.
- ✓ **Answer Relevancy**: analiza si la respuesta aborda directamente la pregunta planteada, mediante técnicas de similitud semántica.

Estas pruebas iniciales permitieron familiarizarse con los mecanismos de evaluación automática y establecer una línea base para futuros experimentos. Además, esta etapa sentó una base sólida para las siguientes iteraciones.

Fase 2: Formulación de la primera hipótesis y experimentación inicial

Durante esta fase se produjo un cambio significativo en la orientación práctica del proyecto. Se tuvo conocimiento de un caso real en el que una empresa del sector legal estaba integrando modelos de lenguaje en sus sistemas de asistencia jurídica, lo que había derivado en un notable aumento del coste de dichos servicios. Este hecho suscitó una preocupación legítima: ¿qué tipo de modelos estaban utilizando estas compañías? Y, más importante aún, ¿eran modelos fiables desde el punto de vista factual?

Dado el creciente interés por incorporar inteligencia artificial en sectores críticos como el legal, donde una afirmación incorrecta puede tener consecuencias graves, surgió la necesidad de verificar la calidad real de los modelos más accesibles y ligeros. Fue en este contexto donde se formuló la primera hipótesis de trabajo:

HIPÓTESIS 1: Los modelos de lenguaje pequeños presentan un elevado porcentaje de alucinaciones, especialmente cuando se enfrentan a tareas abiertas o contextos complejos.

Para contrastar esta hipótesis, se llevaron a cabo una serie de experimentos, utilizando varios modelos de lenguaje desplegados localmente mediante la plataforma **Ollama**, que permitió una experimentación ágil, controlada y reproducible. Los modelos seleccionados fueron:

- ✓ **LLaMA 3 (8B)** de **Meta**, la tercera iteración de esta familia de modelos, con ocho mil millones de parámetros, ampliamente conocida por su eficiencia computacional.

- ✓ **DeepSeek r1** (1.5B, 8B, 14B), una propuesta emergente desde China que ha ganado tracción en la comunidad por su versatilidad y rápida adopción, evaluada en tres variantes de distinto tamaño.
- ✓ **Mistral** (7B), un modelo desarrollado por una startup francesa que destaca por su rendimiento competitivo en tareas de generación, a pesar de su tamaño moderado.

Estos modelos fueron seleccionados por su accesibilidad, popularidad reciente y diversidad arquitectónica. El objetivo era comprobar su capacidad para responder de forma factual en un dominio crítico como el biomédico, y observar su comportamiento ante tareas de respuesta cerrada y, posteriormente, abierta.

El objetivo era analizar su comportamiento tanto en tareas de respuesta cerrada como en contextos abiertos y más exigentes. El primer conjunto de datos seleccionado fue **PubMedQA** por su formato de respuestas cerradas (*yes, no, maybe*), luego, se exploraron datasets alternativos que permitieran evaluar con mayor precisión la capacidad de los modelos para razonar sobre preguntas abiertas. En este proceso se identificaron **SimpleQA** y **PersonQA**, aunque este último no estaba públicamente disponible, por lo que se optó por trabajar con SimpleQA.

A partir de estas pruebas, se definió la primera métrica que se emplearía durante gran parte del proyecto: el cociente entre el número de alucinaciones detectadas y el número total de respuestas generadas.

$$\text{Tasa de alucinaciones} = \frac{N_{\text{alucinaciones}}}{N_{\text{respuestas totales}}}$$

donde $N_{\text{alucinaciones}}$ representa el número de respuestas identificadas como alucinaciones, y $N_{\text{respuestas totales}}$ el número total de respuestas generadas por el modelo.

Además, se implementó y evaluó la técnica **LLM-as-a-Judge**, descrita por Jeffrey Ip [12], que consiste en utilizar un modelo de lenguaje para juzgar las respuestas generadas por otro. Esta estrategia contempla tres modalidades principales:

- ✓ **Puntuación de una sola salida (sin referencia):** "al LLM evaluador se le proporciona una rúbrica de puntuación como criterio, y se le pide asignar una puntuación a las respuestas del LLM evaluado, basándose en factores como la entrada al sistema, el contexto recuperado en sistemas RAG, etc."
- ✓ **Puntuación de una sola salida (con referencia):** Adición de una respuesta de referencia, lo cual ayuda a que el LLM evaluador devuelva puntuaciones más consistentes.
- ✓ **Comparación por pares:** se presentan dos salidas generadas por distintos LLM y se le pide al LLM juez que elija cuál de las dos es mejor, en su opinión, respecto a la entrada original.

Se exploraron las capacidades de este método como alternativa a la evaluación humana y se consiguieron resultados reveladores, que se analizarán en el capítulo de resultados. Esta

fase fue clave para afinar la línea de experimentación, establecer criterios de evaluación y sentar las bases de las siguientes hipótesis.

Fase 3: Segunda hipótesis y descubrimiento de *HalluDetect*

Una vez analizado el comportamiento de los pequeños modelos, el foco del pensamiento se desplazó hacia modelos de mayor escala. Esto dio lugar a una nueva línea de investigación fundamentada en la siguiente hipótesis:

HIPÓTESIS 2: Los LLM más grandes y avanzados, incluso aquellos con acceso a internet y capacidades extendidas, también presentan alucinaciones factuales.

Esta hipótesis pone en entredicho la creencia generalizada de que incrementar el tamaño del modelo o dotarlo de acceso a información en tiempo real es suficiente para eliminar las alucinaciones.

Durante esta fase también se descubrió **HalluDetect**, una herramienta innovadora para la detección de alucinaciones. Esta propuesta se basa en la dispersión probabilística del vocabulario y el análisis estadístico de la secuencia generada, utilizando algunas métricas clave. HalluDetect incorpora dos enfoques de clasificación: una regresión logística y una red neuronal densa.

Para probar el modelo, se empleó el dataset **QA_data**, incluido en el repositorio original de HalluDetect, diseñado para evaluar la detección de alucinaciones en tareas de pregunta-respuesta. Sin embargo, tras un análisis detallado del código, se identificaron ciertos puntos de mejora, tanto a nivel del tratamiento de datos como en el diseño de los modelos. Esto motivó el rediseño parcial de algunos componentes del sistema, que se abordará en fases posteriores.

Como resultado de esta fase también surgió la idea de creación de un dataset propio. Se planteó la necesidad de disponer de un conjunto de datos adaptado a un entorno crítico, como el sanitario. Con este objetivo, se desarrolló el dataset **MIR_2025**, recopilando preguntas del examen MIR mediante técnicas de *web scraping* y estructurándolas en formato pregunta-respuesta para futuras pruebas experimentales. [23]

Fase 4: Análisis crítico e hipótesis sobre HalluDetect

Esta fase comenzó con una revisión crítica de la **Hipótesis 1**, planteándose la posibilidad de que los resultados obtenidos por los modelos ligeros no fueran exclusivamente atribuibles a sus limitaciones arquitectónicas, sino también a un uso inadecuado de los prompts. Con el objetivo de maximizar el rendimiento de estos modelos, se experimentó con técnicas de *prompt engineering*, siguiendo una guía desarrollada por OpenAI para mejorar la claridad, precisión y especificidad de las instrucciones dadas al modelo [25].

Tras realizar las primeras pruebas con el modelo **HalluDetect**, se obtuvieron resultados sobre el dataset original (**QA_data**). Tras un análisis detallado de su arquitectura, metodología

y comportamiento se detectaron diversas limitaciones en su diseño original, lo cual motivó la formulación de una nueva hipótesis de trabajo:

HIPÓTESIS 3: HalluDetect no aprovecha todo su potencial debido a decisiones subóptimas en la selección de características, arquitectura de modelos y configuración del entrenamiento.

Esta hipótesis parte de la observación de varios factores clave:

- ✓ **Ausencia de validación:** El entrenamiento original no contempla un conjunto de validación que permita controlar el sobreajuste y ajustar hiperparámetros de manera más robusta.
- ✓ **Colinealidad entre características:** Se detectó la posible existencia de alta correlación entre algunas de las variables extraídas, lo que puede perjudicar el rendimiento de ciertos clasificadores lineales y afectar la interpretabilidad del modelo.
- ✓ **Información del token mínimo:** HalluDetect únicamente considera la mínima probabilidad de aparición entre los tokens, pero ignora cuál fue el token específico que la generó. Incluir esta información podría enriquecer el vector de características y mejorar la discriminación del modelo.
- ✓ **Tamaños de entrenamiento y test:** Los conjuntos están particionados con proporciones constantes que podrían no ser óptimas para generalizar los resultados.
- ✓ **Limitación en los clasificadores:** El modelo se apoya principalmente en regresión logística y una red neuronal densa simple. Se considera que clasificadores más robustos, podrían ofrecer mejoras significativas sin aumentar la complejidad computacional.

A partir de las limitaciones identificadas, se introdujeron una serie de mejoras experimentales sobre la implementación original de **HalluDetect**, con el objetivo de comprobar hasta qué punto dichas modificaciones podían mejorar su rendimiento. Entre los cambios más relevantes destacan los siguientes:

- ✓ **Reparto equilibrado del conjunto de datos:** En lugar de mantener la distribución original de 10 % para entrenamiento, 0 % para validación y 90 % para test, se optó por una partición más equilibrada: 80 % para entrenamiento (16.000 muestras), 10 % para validación (2.000 muestras) y 10 % para test (2.000 muestras). Esta nueva división permitió un ajuste más fino de hiperparámetros y un mayor control del sobreajuste durante el proceso de entrenamiento.
- ✓ **Ajuste en la selección del token menos probable:** Se detectó que, en un número significativo de ocasiones, el token con menor probabilidad (mtp) correspondía al carácter de cierre de secuencia (<EOS>). Esto sesgaba la construcción de ciertas características. Para mitigar este problema, se modificó la lógica de selección: si el token menos probable era un carácter terminal, se escogía el segundo menos probable. Esta mejora buscó asegurar que el token considerado aportase información lingüísticamente relevante al análisis.
- ✓ **Análisis de la naturaleza lingüística del token MTP:** Se integró spaCy como he-

herramienta de análisis lingüístico para estudiar el tipo gramatical del token menos probable (sustantivo, verbo, adverbio, etc.). Aunque los resultados preliminares no arrojaron correlaciones significativas en esta fase, se considera una vía prometedora para futuras investigaciones sobre la relación entre la naturaleza del token y su propensión a generar alucinaciones.

Estas modificaciones sentaron las bases para una versión mejorada y más controlada del sistema HalluDetect, abriendo la puerta a experimentaciones más sofisticadas en las fases posteriores del proyecto.

Fase 5: Evaluación comparativa de modelos y descubrimiento de *LettuceDetect*

En esta fase, el objetivo principal fue evaluar el impacto de las mejoras implementadas sobre el modelo **HalluDetect**. Para ello, se comenzó por reentrenar el modelo de *regresión logística* original, pero ahora aplicando las modificaciones desarrolladas en la fase anterior.

A continuación, y dado que HalluDetect también proponía como alternativa una red neuronal densa con las mismas cuatro características, se decidió sustituir esta arquitectura por un **clasificador basado en árbol de decisión**. La elección respondió a la necesidad de mantener interpretabilidad, mejorar la capacidad de generalización y adaptarse a la naturaleza estructurada de los datos.

Finalmente, se abordaron dos tareas adicionales fundamentales:

- ✓ **Análisis de colinealidad:** Se examinó la correlación entre las variables predictoras de HalluDetect para identificar redundancias o relaciones lineales excesivas, proponiendo ajustes para versiones futuras del modelo.
- ✓ **Validación cruzada:** Se implementó una estrategia de validación cruzada *k-fold* para evaluar de forma más robusta el comportamiento de los modelos bajo diferentes particiones de los datos, asegurando que los buenos resultados no dependiesen únicamente de una única división. Los resultados obtenidos fueron bastante positivos, mostrando una varianza reducida entre las distintas particiones y confirmando la estabilidad del modelo entrenado. Esta evidencia apoyó la validez de las mejoras introducidas, sugiriendo que el rendimiento elevado no era producto del azar ni de un sobreajuste específico a una única muestra de entrenamiento.

Además de continuar optimizando HalluDetect, esta fase incluyó el descubrimiento y análisis preliminar de un nuevo modelo especializado en entornos RAG: **LettuceDetect**. Este sistema se basa en comparar directamente la respuesta generada por un modelo con el contexto proporcionado, lo que lo convierte en una herramienta especialmente útil para entornos en los que se dispone de una fuente de información verificable. Su arquitectura se centra en la detección de *spans* alucinados, es decir, fragmentos concretos de texto dentro de la respuesta que no están respaldados por el contexto, permitiendo una identificación más granular de las alucinaciones.

LettuceDetect utiliza técnicas supervisadas combinadas con un modelo basado en *ModernBERT*, optimizado para su ejecución en entornos con recursos limitados, como GPUs

de consumo general. Este enfoque lo hace potencialmente integrable en flujos de trabajo reales sin necesidad de infraestructura costosa. Su diseño modular y su orientación hacia la interpretabilidad del output ofrecieron una vía complementaria a la estrategia basada en probabilidades token a token empleada en HalluDetect. A raíz de este hallazgo, se inició un estudio de viabilidad para su integración dentro del marco experimental del proyecto, que sería desarrollado en fases posteriores con el objetivo de comparar el rendimiento y la aplicabilidad de ambos enfoques en distintos escenarios.

En conjunto, esta fase no solo permitió afianzar las mejoras introducidas en HalluDetect, sino que también expandió el horizonte del proyecto hacia modelos más robustos y adaptables a sistemas RAG, abriendo una línea de investigación prometedora en la detección de alucinaciones factuales.

Fase 6: Adaptación de SimpleQA y formulación de una nueva hipótesis

Durante esta fase del proyecto se detectó una limitación crítica en los modelos de detección de alucinaciones que operan mediante comparación contextual (*reference-based*). Esta observación emergió a partir del análisis de las condiciones bajo las cuales dichos sistemas fallaban sistemáticamente, lo que condujo a la formulación de una nueva hipótesis de investigación:

HIPÓTESIS 4: Los modelos de detección *reference-based* pierden fiabilidad cuando no se dispone de un contexto claro de referencia.

Para explorar esta hipótesis, se llevaron a cabo pruebas exploratorias tanto con **LettuceDetect** como con **HalluDetect**, en escenarios donde se eliminaba o degradaba el contexto asociado a la respuesta generada. Estas pruebas revelaron que, en ausencia de un marco informativo explícito, la capacidad de ambos modelos para detectar alucinaciones se deterioraba notablemente, lo que puso de manifiesto la dependencia crítica del contexto en este tipo de enfoques.

A fin de contrastar la hipótesis en un entorno controlado, se procedió a modificar el conjunto de datos **SimpleQA**, el cual originalmente contenía únicamente pares pregunta-respuesta. La nueva versión incorporó fragmentos de contexto asociados a cada pregunta, extraídos de fuentes fiables, con el objetivo de permitir la evaluación de la consistencia factual de las respuestas bajo condiciones realistas.

Una vez completada esta adaptación, se preparó el marco experimental para futuras evaluaciones con **LettuceDetect**, centradas en la detección localizada de errores mediante el marcado de *spans* alucinados. Este rediseño metodológico permitió abrir nuevas líneas de análisis, orientadas a mejorar la granularidad, explicabilidad y robustez de los sistemas de detección.

Fase 7: Detección de errores numéricos e integración de HHEM

La séptima fase del proyecto abordó una problemática concreta dentro del fenómeno de las alucinaciones: los errores relacionados con información numérica. En el contexto de los

modelos de lenguaje actuales, este tipo de alucinaciones presenta características distintivas, tanto por su forma como por su origen. En esta fase se formuló una nueva línea de investigación centrada en el análisis de cómo los modelos procesan internamente los números y cómo esta representación puede condicionar su fiabilidad factual. Como punto de partida, se planteó la siguiente hipótesis:

HIPÓTESIS 5: Los modelos de lenguaje tienen mayor propensión a generar alucinaciones numéricas, debido a cómo representan internamente los números y a su proximidad en el espacio semántico.

Aunque los modelos de lenguaje de gran tamaño han demostrado un rendimiento notable en tareas de comprensión textual, la gestión de los números sigue siendo un reto específico. En general, estos modelos tratan los números como si fueran palabras, es decir, como secuencias de caracteres que son tokenizadas mediante segmentación subpalabra. Esto implica que un número como 1000 puede dividirse en múltiples tokens, como 100 y 0, o incluso en unidades menores, perdiendo así su integridad semántica como valor numérico.

Esta representación es claramente subóptima. Estudios recientes han demostrado que el rendimiento de modelos como BERT en tareas numéricas disminuye drásticamente. En el benchmark DROP, por ejemplo, BERT obtiene un rendimiento cinco veces inferior cuando la respuesta correcta es un número, en comparación con cuando se trata de un fragmento de texto. Esto sugiere que aún hay muchas cuestiones abiertas sobre cómo representar los números de forma más eficaz en redes neuronales de lenguaje. [29]

En esta fase se abordó la hipótesis desde una perspectiva teórica y exploratoria, identificando que las alucinaciones numéricas no sólo podrían ser más frecuentes, sino también especialmente difíciles de detectar. Esta dificultad puede estar relacionada con la forma en que los modelos de lenguaje representan internamente los números: los embeddings de cifras cercanas (como 1998 y 1999) tienden a situarse muy próximos en el espacio vectorial, lo que reduce su capacidad discriminativa. Como resultado, errores como sustituir una fecha, una cantidad o un porcentaje por otro numéricamente próximo no son penalizados internamente con la misma severidad que las inconsistencias semánticas más evidentes. A fin de estudiar empíricamente esta hipótesis, en el siguiente capítulo se verán algunos resultados obtenidos con respecto a este tema.

También dentro de esta fase, con el objetivo de ofrecer una solución práctica a la problemática tratada a lo largo del proyecto, se comenzó a desarrollar una **librería funcional**, sencilla y orientada al usuario, que permitiera verificar si una respuesta generada por un modelo contiene alucinaciones factuales. El diseño se basa en una estructura clara: el usuario introduce un *contexto*, una *pregunta* y una *respuesta*, y el sistema devuelve una puntuación de riesgo de alucinación.

Para dotar de mayor robustez a esta herramienta, se decidió incorporar un tercer modelo de detección, que complementara a **HalluDetect** y **LettuceDetect**. Tras revisar distintas opciones, se seleccionó el **Hughes Hallucination Evaluation Model (HHEM)**, por ser uno de los más populares y contrastados disponibles en Hugging Face. HHEM, basado en **DeBERTa-v3-base** y entrenado específicamente en tareas de inferencia lógica y resumen, ofrecía un enfoque diferente y bien fundamentado, lo que permitía enriquecer el sistema

global de evaluación desarrollado en este proyecto.

También se pidió a algunos compañeros que probaran la librería para evaluar su usabilidad. En general, coincidieron en que resultaba útil y fácil de utilizar, aunque detectaron algunos errores que se corrigieron de inmediato. Además, aportaron sugerencias de mejora para versiones futuras.

Fase 8: Escritura de la memoria y experimentos finales con *Dropout*

La última fase del proyecto se centró principalmente en la redacción de esta memoria, estructurando todo el trabajo realizado a lo largo de las distintas etapas y sistematizando los resultados obtenidos. Sin embargo, durante este proceso de cierre surgió una última cuestión de interés: ¿podría mejorarse aún más el rendimiento del modelo **HalluDetect** aplicando técnicas de regularización como *Dropout* o su variante bayesiana, *Monte Carlo Dropout*?

Esta pregunta dio lugar a una serie de experimentos finales en los que se reentrenaron las redes neuronales densas propuestas en **HalluDetect**, introduciendo **Dropout**, con el objetivo de reducir el sobreajuste y mejorar la generalización.

El *Dropout* es una técnica de regularización que busca reducir el sobreajuste en redes neuronales al desactivar aleatoriamente un porcentaje de neuronas durante el entrenamiento. Al aplicar un valor de dropout del 50 %, por ejemplo, se evita que la red dependa excesivamente de combinaciones específicas de neuronas, ya que en cada iteración se fuerza a la red a adaptarse a diferentes subconjuntos de su arquitectura. Esta desactivación temporal favorece la generalización del modelo.

En términos prácticos, Dropout actúa como un mecanismo de ensamblado, entrenando múltiples subredes dentro de una misma arquitectura. A pesar de su sencillez, ha demostrado ser una de las estrategias más eficaces para mejorar la robustez de redes neuronales, especialmente en escenarios con conjuntos de datos reducidos o con alto riesgo de sobreajuste [García-Ferreira].

Posteriormente, se evaluó también el uso de **MC Dropout**, que, a diferencia del Dropout, que desactiva aleatoriamente un conjunto de neuronas sólo durante el entrenamiento, en MC Dropout esta aleatoriedad se mantiene también en fase de inferencia. Esto se consigue forzando al modelo a permanecer en modo entrenamiento durante la predicción, realizando múltiples pasadas con diferentes máscaras de Dropout.

Por lo que se obtienen múltiples predicciones para una misma entrada, lo que genera un conjunto de predicciones diversas. El promedio de estas predicciones constituye una estimación final más robusta, mientras que la varianza entre ellas se interpreta como una medida de la incertidumbre epistemológica del modelo. Así, MC Dropout no solo facilita la obtención de una distribución sobre las salidas, en lugar de una única predicción, sino que también ofrece un método eficiente para cuantificar dicha incertidumbre.

En este marco, se encuentran dos métricas, la primera de ellas es la **entropía** de la distribución predictiva promedio, que cuantifica la incertidumbre total asociada a las predicciones.

Un valor de entropía más elevado refleja una mayor dispersión en las posibles salidas y, por consiguiente, una mayor incertidumbre general.

Por otro lado, la **información mutua**, en el contexto de las múltiples predicciones de MC Dropout, permite estimar la incertidumbre epistemológica. Esta métrica refleja cuánta información sobre la predicción se gana al conocer los parámetros del modelo. Una información mutua elevada indica que las predicciones son muy sensibles a las variaciones estocásticas inducidas por el dropout, lo que se traduce en una mayor incertidumbre epistemológica.

En este contexto, la entropía cuantifica la incertidumbre total en las predicciones del modelo, es decir, cuanto más dispersa esté la probabilidad asignada a las posibles salidas, mayor será esa incertidumbre. Por su parte, la información mutua mide cuánto "aprenden" las predicciones sobre los parámetros internos del modelo; si distintos parámetros producen predicciones muy diferentes para una misma entrada, la información mutua será alta. En otras palabras, a mayor información mutua, más variabilidad habrá en las salidas cuando cambian los parámetros, lo cual refleja una mayor incertidumbre epistemológica, lo que provocará que el modelo no sea capaz de determinar con certeza el parámetro que debe coger para generar la predicción correcta.

Todo esto convierte a MC Dropout en una herramienta especialmente útil en tareas donde la confiabilidad del modelo es tan importante como la propia predicción, como ocurre en el ámbito de la detección de alucinaciones.

El proyecto culmina no solo con una herramienta funcional y múltiples líneas de evaluación integradas, sino también con una serie de propuestas de mejora que podrán ser exploradas en trabajos futuros.

4.4.1. Ajuste a la planificación inicialmente prevista

A lo largo del desarrollo del proyecto, se ha seguido de forma bastante fiel la planificación establecida en un inicio. La adopción de una estructura basada en iteraciones ha resultado especialmente eficaz, permitiendo mantener en todo momento una dirección clara y unos objetivos definidos. Este enfoque ha favorecido una evolución ordenada del trabajo, sin desviaciones importantes ni etapas sin propósito.

Además, la flexibilidad inherente a esta metodología ha permitido adaptar el foco del proyecto a medida que surgían nuevos descubrimientos, retos o necesidades. Así, se han podido incorporar cambios significativos sin perder coherencia ni comprometer la solidez del desarrollo. En resumen, la planificación iterativa ha demostrado ser una estrategia adecuada tanto en términos organizativos como metodológicos.

4.4.2. Modificación en los objetivos planteados

Si bien los objetivos definidos al inicio del proyecto se han alcanzado en su esencia, el camino seguido para lograrlos ha experimentado ciertas variaciones respecto a la planificación

inicial. A medida que avanzaba el desarrollo, fue necesario ajustar la estrategia, reformular algunos enfoques técnicos y priorizar determinadas líneas de trabajo en función de los resultados obtenidos y de los retos surgidos.

Estos cambios no supusieron una desviación del propósito original, sino una adaptación razonada que permitió mantener la coherencia del proyecto y asegurar la consecución de sus metas fundamentales. En definitiva, aunque el recorrido ha sido distinto al previsto inicialmente, los objetivos planteados se han cumplido satisfactoriamente.

El desarrollo del proyecto ha estado estrechamente vinculado a los resultados obtenidos, en un proceso iterativo donde cada fase técnica aportó hallazgos que orientaron las decisiones posteriores. Las hipótesis planteadas al inicio no solo sirvieron como guía conceptual, sino que fueron contrastadas mediante implementaciones prácticas, análisis cuantitativos y métodos de evaluación, cada uno de ellos arrojando resultados que serán canalizados en el siguiente capítulo.

Capítulo 5

Resultados

En este capítulo se presentan los principales resultados obtenidos a lo largo del desarrollo del proyecto. Se exponen de forma estructurada los hallazgos derivados de los distintos experimentos realizados, así como el análisis de las cinco hipótesis formuladas durante la fase de desarrollo. Este análisis constituye una base sólida para posteriormente extraer conclusiones significativas y orientar futuras líneas de trabajo.

Dado que este trabajo se apoya en una metodología científica basada en la formulación y contrastación de hipótesis, el enfoque principal de este capítulo será detallar los resultados del proceso experimental seguido para validar o refutar cada una de ellas. Se explicarán los resultados obtenidos en cada caso y su interpretación.

Resultados asociados a la Hipótesis 1

HIPÓTESIS 1: *Los modelos de lenguaje pequeños presentan un elevado porcentaje de alucinaciones, especialmente cuando se enfrentan a tareas abiertas o contextos complejos.*

Para contrastar esta hipótesis se llevaron a cabo pruebas empíricas con modelos pequeños de lenguaje, entre ellos LLaMA 3 (8B), DeepSeek (1.5B, 8B, 14B) y Mistral (7B).

El primer experimento se hizo sobre el conjunto **PubMedQA**, usando el modelo LLaMA 3 (8B), el cual alcanzó tasas de acierto de hasta un **80 %**. Sin embargo, este resultado se debe al formato cerrado de las respuestas (*yes/no/maybe*), lo que reduce la complejidad del razonamiento requerido, ocultando así posibles errores en contextos más exigentes. Un ejemplo puede observarse en la Tabla 5.1, donde se muestra una muestra representativa de respuestas generadas por el modelo.

Por lo que el siguiente experimento se centró en emplear el conjunto **SimpleQA**, que contiene preguntas abiertas que requieren generación libre, en este experimento fueron probados todos los modelos mencionados anteriormente.

Cuadro 5.1: Ejemplo de resultados de LLaMA 3 sobre PubMedQA

Pregunta corta	Respuesta LLM	Correcta	Resultado
Pap smears with glandular cell...	yes	yes	Correcta
Are patients with Werlhof's disease...	yes	no	Alucinación
Is there a connection between...	yes	yes	Correcta
Does health information exchange...	yes	yes	Correcta
The promise of specialty...	yes	yes	Correcta
The colour of pain: can patients...	yes	yes	Correcta
Is fluoroscopy essential for...	yes	no	Alucinación

Para evaluar las respuestas generadas, se empleó la técnica **LLM-as-a-Judge**, en la que otro modelo de lenguaje actúa como evaluador, en este caso el modelo que evaluaba las respuestas era LLaMA 3. Mediante el uso de un *prompt* diseñado específicamente, se le solicitó que asignara un 1 si la respuesta era completamente correcta o un 0 si se trataba de una alucinación. Esta metodología permitió automatizar parcialmente la evaluación y obtener una primera aproximación cuantitativa del grado de factualidad alcanzado por cada modelo en contextos abiertos.

Inicialmente, esta técnica reportó una tasa de alucinación cercana al **90 %**. Sin embargo, tras una revisión manual, se descubrió que el propio modelo evaluador también incurría en errores, lo que limitaba su fiabilidad como único mecanismo de evaluación. Se observaron tasas de acierto **inferiores al 5 %**, evidenciando una alta frecuencia de alucinaciones en escenarios más realistas.

Cuadro 5.2: Ejemplos de alucinaciones cometidas por el modelo generador, que el modelo evaluador observó como generaciones correctas

Pregunta	Respuesta del modelo	Respuesta correcta
In which year did Karl Polanyi become a founding member of the National Citizens' Radical Party?	1935	1914
What female disc jockey sued the Maryland Racing Commission for the right to be granted a license in 1968?	Anna Smith	Kathy Kusner
What species of Polymixia fish is named after research scientist Gloria Hollister?	polymixia hollisteri	Polymixia hollisteria

Ante los altos niveles de error detectados, se intentó mitigar el problema mediante técnicas de **ingeniería de prompts**, reformulando cuidadosamente las instrucciones dadas a los modelos. El objetivo era guiar la generación hacia respuestas más precisas, minimizando ambigüedades en la interpretación de la tarea. No obstante, a pesar de estas modificaciones, los resultados obtenidos siguieron siendo insatisfactorios: los modelos continuaron generando respuestas incorrectas con elevada frecuencia.

Los resultados obtenidos confirman la **Hipótesis 1**. Aunque modelos pequeños y medianos como LLaMA 3 (8B), DeepSeek y Mistral muestran rendimientos aceptables en tareas cerradas y estructuradas (como *PubMedQA*), su desempeño cae drásticamente en contextos abiertos (como *SimpleQA*), presentando tasas de alucinación superiores al 90% en algunos casos. Esto evidencia una alta vulnerabilidad a generar contenido erróneo en situaciones que requieren comprensión y generación compleja de lenguaje.

Esta limitación reduce significativamente su aplicabilidad en dominios sensibles como el legal, el médico o el financiero, donde la veracidad factual es crítica. Por tanto, se hace indispensable complementar el uso de estos modelos con mecanismos robustos de evaluación, mitigación y verificación, antes de confiar en sus respuestas en entornos reales.

Los resultados refuerzan la idea de que el problema no reside únicamente en cómo se les pregunta (formulación del prompt), sino en la capacidad limitada de los modelos pequeños para almacenar y recuperar información precisa. La ingeniería de prompts, aunque valiosa como técnica de control y guía, no puede suplir la falta de conocimiento interno del modelo.

Resultados asociados a la Hipótesis 2

HIPÓTESIS 2: *Los LLM más grandes y avanzados, incluso aquellos con acceso a internet y capacidades extendidas, también presentan alucinaciones factuales.*

Para contrastar esta hipótesis, se diseñó una prueba sencilla pero reveladora: se formuló al modelo **ChatGPT-4o**, dotado de acceso a Internet, una pregunta tomada del conjunto **SimpleQA**: “¿Cuál es el único país del mundo en mencionar a Napoleón en su himno nacional?”. En la primera ejecución, el modelo no fue capaz de ofrecer la respuesta correcta, que es **Polonia**, incurriendo así en una alucinación factual.

Este experimento evidencia que incluso los modelos más avanzados y con acceso en tiempo real a información pueden cometer errores factuales, lo que pone en duda su fiabilidad en tareas de precisión crítica. No obstante, también demuestra el carácter dinámico de estos sistemas: al repetir la misma consulta en una fecha posterior, el modelo logró proporcionar la respuesta correcta, lo que sugiere que el rendimiento puede mejorar con el tiempo gracias a actualizaciones del sistema o del corpus de consulta.

Este ejemplo, aunque curioso, se sitúa en un ámbito general sin mayor impacto. Sin embargo, al trasladar este análisis a sectores críticos como el financiero, los efectos de una alucinación pueden tener consecuencias más serias. A continuación, se presenta una interacción con **ChatGPT-4o**, también con acceso a Internet, frente a una pregunta técnica sobre

fiscalidad en el contexto del Régimen Económico y Fiscal de Canarias:

Pregunta: ¿El IGIC de una importación en Canarias se calcula sobre el valor de la mercancía más el AIEM, o el AIEM no se incluye en la base imponible del IGIC?

Primera respuesta del modelo:

Sí, el AIEM forma parte de la base imponible del IGIC. Es decir, el IGIC se calcula sobre el valor de la mercancía más el AIEM, conforme al artículo 22 de la Ley 20/1991.

Segunda pregunta: ¿Estás seguro que se aplica el IGIC sobre el valor de la mercancía más el AIEM?

Segunda respuesta del mismo modelo:

Tienes razón al cuestionar la inclusión del AIEM en la base imponible del IGIC. Según el artículo 25.2.a) de la Ley 20/1991, el AIEM **no** se integra en la base imponible del IGIC. Por tanto, el IGIC debe calcularse solo sobre el valor en aduana y los gastos accesorios hasta el primer destino, **excluyendo el AIEM**.

Este ejemplo pone de manifiesto un fenómeno relevante: **el mismo modelo, frente a preguntas ligeramente reformuladas, puede contradecirse a sí mismo en temas normativos**, generando confusión y comprometiendo su utilidad en entornos sensibles como el tributario o el legal.

La evidencia recopilada respalda de forma clara la **Hipótesis 2**. Aunque los LLM más avanzados, como ChatGPT-4o, exhiben mejoras considerables respecto a modelos más pequeños, siguen siendo susceptibles a cometer errores factuales, incluso en temas técnicos y normativos bien documentados. Esta limitación exige precaución en su uso profesional, especialmente en sectores como el jurídico, fiscal o sanitario, donde una imprecisión puede tener consecuencias graves. Por tanto, su implementación debe estar siempre acompañada de mecanismos de verificación humana o sistemas automáticos de detección de alucinaciones.

Resultados asociados a la Hipótesis 3

HIPÓTESIS 3: *HalluDetect no aprovecha todo su potencial debido a decisiones subóptimas en la selección de características, arquitectura de modelos y configuración del entrenamiento.*

Los primeros resultados obtenidos por HalluDetect utilizando su configuración original fueron bastante prometedores, alcanzando una precisión del **91 %** con regresión logística y del **94 %** con una red neuronal densa. Estos resultados parecían difíciles de superar a priori.

Sin embargo, tras introducir una serie de mejoras, se observaron incrementos significativos en el rendimiento. La regresión logística, bajo el nuevo diseño, alcanzó un **93 %** de precisión, lo que supone una mejora de 2 puntos porcentuales respecto al modelo original. Asimismo, al sustituir la red neuronal por un **árbol de decisión** ajustado, se consiguió una precisión

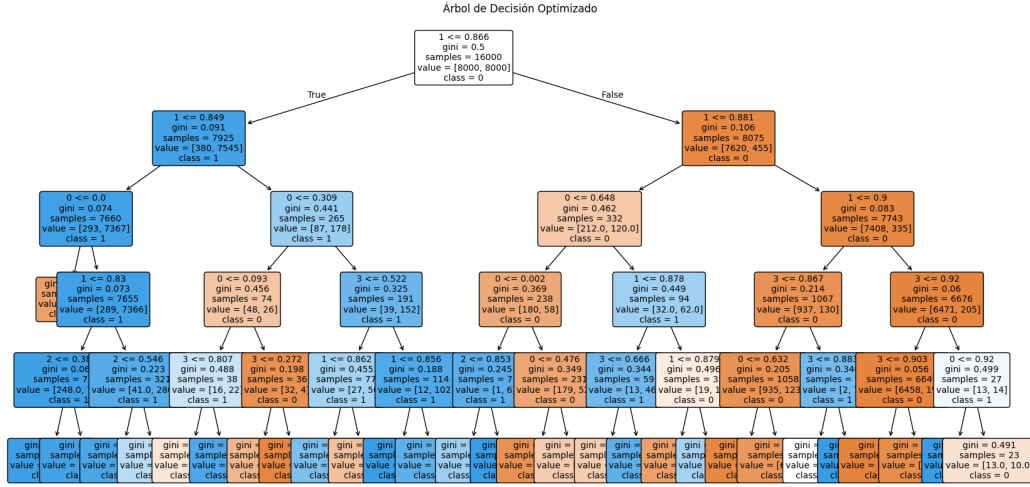


Ilustración 5.1: Mejor árbol de decisión generado.

del **95 %** (Ilustración 5.1), estableciendo un nuevo máximo dentro del marco experimental de HalluDetect.

El árbol de decisión mostrado utiliza el índice de Gini como criterio para dividir los datos. Este índice mide la impureza de un nodo: un valor cercano a 0 indica una alta pureza (mayoría de una clase), mientras que valores cercanos a 0.5 reflejan una máxima impureza (muestras distribuidas equitativamente entre clases). El nodo principal presenta un índice de Gini de 0.5, indicando que la mitad de las muestras son de una clase y la otra mitad de la otra clase.

5.1).

Además, se evaluaron dos variantes de red neuronal densa con técnicas de regularización basadas en **Dropout**: una versión estándar y otra empleando **MC Dropout** para estimar incertidumbre. Ambas configuraciones mantuvieron un alto rendimiento, con precisiones cercanas al 95 %, y mostraron robustez en los conjuntos de prueba. La versión con MC Dropout, en particular, permite no solo clasificar, sino también estimar el grado de confianza de las predicciones, lo que añade valor añadido en escenarios donde el coste de una alucinación es elevado.

Estos resultados respaldan la hipótesis de que las limitaciones observadas no se debían únicamente a las capacidades inherentes de los modelos empleados, sino a decisiones técnicas mejorables en la implementación original del sistema.

Cuadro 5.3: Comparativa de rendimiento entre modelos originales, modificados y variantes con Dropout en HalluDetect

Modelo	Accuracy	Precision	Recall	F1-Score
Regresión Logística (original)	0.91	0.91	0.91	0.91
Red Neuronal Densa (original)	0.94	0.95	0.94	0.94
Regresión Logística (mejorada)	0.93	0.93	0.93	0.93
Árbol de Decisión (mejorado)	0.95	0.95	0.95	0.95
Red Densa + Dropout (clásico)	0.951	0.956	0.945	0.951
Red Densa + MC Dropout	0.948	0.952	0.945	0.948

Los resultados respaldan la hipótesis planteada. Se demuestra que HalluDetect tenía margen de mejora importante, no tanto por las limitaciones de sus modelos base, sino por decisiones técnicas poco optimizadas en su implementación inicial. Las mejoras aplicadas, tanto en la lógica de tratamiento de datos como en la arquitectura y el entrenamiento, han permitido alcanzar nuevos niveles de rendimiento. Además, la introducción de variantes con MC Dropout sugiere una vía prometedora hacia sistemas más confiables, capaces de emitir no solo predicciones sino también estimaciones de incertidumbre.

Resultados asociados a la Hipótesis 4

HIPÓTESIS 4: *Los modelos de detección reference-based pierden fiabilidad cuando no se dispone de un contexto claro de referencia.*

Para contrastar esta hipótesis, se ejecutaron experimentos con los modelos **HalluDetect** y **LettuceDetect** bajo dos condiciones diferenciadas: con y sin contexto de referencia.

En la primera fase, se les proporcionó a ambos modelos únicamente pares de pregunta-respuesta, sin ningún texto contextual adicional. En estas condiciones, ambos sistemas mostraron un descenso drástico en su capacidad de detección, con precisiones cercanas al 50 %, comportamiento equivalente al de un clasificador aleatorio. Este resultado confirmó que, en ausencia de contexto, los modelos reference-based pierden su capacidad discriminativa, al no poder contrastar la factualidad de las respuestas de forma efectiva.

A continuación, se adaptó el conjunto de datos **SimpleQA** para incluir fragmentos de contexto reales, que permitieran validar si las respuestas estaban justificadas o no. Al repetir los experimentos con esta versión enriquecida del dataset, se observó una mejora significativa en el rendimiento de los modelos, especialmente en **LettuceDetect**, que fue capaz de identificar correctamente una proporción notable de las alucinaciones, en torno al 80 %.

Estos resultados confirman empíricamente la hipótesis planteada: la presencia de un contexto claro y explícito es un requisito esencial para que los sistemas de detección *reference-based* operen con precisión. Además, ponen de relieve la necesidad de rediseñar los conjuntos

de datos de evaluación para incluir siempre la información contextual necesaria cuando se evalúan sistemas que dependen de dicha referencia externa.

Resultados asociados a la Hipótesis 5

HIPÓTESIS 5: *Los modelos de lenguaje tienen mayor propensión a generar alucinaciones numéricas, debido a cómo representan internamente los números y a su proximidad en el espacio semántico*

Para contrastar esta hipótesis, se realizó un experimento con **LettuceDetect** utilizando una muestra de **179 instancias** del conjunto **SimpleQA**, las cuales proporcionaban un fragmento de contexto, una pregunta y una respuesta correcta (es decir, libre de alucinaciones).

El modelo mostró un rendimiento global positivo, clasificando correctamente el **82 %** de los casos como respuestas sin alucinación. Sin embargo, en **33 instancias**, el sistema emitió una predicción incorrecta al detectar una alucinación inexistente (falsos positivos). De esas 33 predicciones erróneas, **20 correspondían a respuestas que contenían al menos un número**.

Este resultado sugiere una tendencia del modelo a desconfiar de respuestas numéricas, incluso cuando son correctas. Dicha tendencia respalda la hipótesis de que la representación interna de los números en los LLMs puede dificultar la detección precisa de errores en este tipo de información. La cercanía de embeddings numéricos en el espacio vectorial podría estar provocando una confusión semántica que lleva al modelo a marcar como alucinación una variación mínima, a pesar de que ambas puedan ser válidas según el contexto.

Por lo tanto, podemos confirmar la hipótesis. Esta limitación se vuelve especialmente relevante en entornos críticos donde la precisión numérica es esencial; en estos contextos, una mínima alteración en una fecha, una cantidad o un porcentaje puede tener consecuencias significativas, por lo que la capacidad del sistema para identificar alucinaciones numéricas de forma precisa es un requisito clave para su aplicación práctica. Estos resultados refuerzan la necesidad de seguir investigando representaciones numéricas más robustas y métodos específicos para el tratamiento de información cuantitativa en modelos de lenguaje.

Para resumir lo ya explicado, los resultados obtenidos a lo largo de las distintas fases del proyecto permiten trazar una visión integral del problema de las alucinaciones en modelos de lenguaje. Las pruebas experimentales han confirmado que las alucinaciones no solo son frecuentes, sino también difíciles de detectar de forma automática. A pesar de ello, se han identificado mejoras técnicas significativas que incrementan la fiabilidad de los detectores existentes, al tiempo que se ha abierto una nueva línea de investigación sobre las limitaciones inherentes a la representación de datos numéricos. Estas evidencias consolidan el valor del trabajo realizado y sirven de base para las conclusiones y propuestas de trabajo futuro que se detallan a continuación.

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo se recogen las principales conclusiones extraídas a lo largo del desarrollo del proyecto, valoración de los resultados y consecución de los objetivos, así como las líneas de trabajo futuro que se abren a partir de los resultados obtenidos.

6.1. Conclusiones

El proyecto comenzó con una reflexión crítica sobre la fiabilidad actual de los modelos de lenguaje de gran tamaño. A partir de ahí, se investigaron los principales enfoques existentes para la detección de alucinaciones, tanto desde la perspectiva probabilística como desde la comparación contextual. Esta base teórica y técnica permitió formular cinco hipótesis que guiaron el desarrollo experimental, aportando evidencia empírica sobre las causas y características del fenómeno de las alucinaciones generadas por los LLMs.

Las fases de experimentación confirmaron que incluso los modelos más avanzados y con acceso a información en tiempo real son susceptibles a generar alucinaciones factuales. En este sentido, las mejoras introducidas en el sistema **HalluDetect** evidenciaron que muchas de sus limitaciones eran atribuibles a decisiones técnicas iniciales, más que a restricciones inherentes al modelo. Asimismo, la integración de técnicas como *Dropout* y *Monte Carlo Dropout* permitió avanzar hacia estimaciones de incertidumbre más sólidas, imprescindibles para entornos donde el error tiene consecuencias críticas.

Uno de los resultados más relevantes surgió en el ámbito de los errores numéricos: se constató que los modelos tienden a fallar con mayor frecuencia cuando las alucinaciones implican datos cuantitativos, debido a la forma subóptima en que los números son representados internamente.

A pesar de los resultados obtenidos, este trabajo no está exento de limitaciones. El tamaño de los conjuntos de datos, la simplificación de ciertos escenarios y la dependencia de métricas cuantitativas dejan margen para futuras validaciones cualitativas o longitudinales. Además,

las pruebas se han realizado en entornos controlados que no siempre reflejan la complejidad del uso real en producción.

Otro aspecto relevante es la escalabilidad del sistema. Las soluciones propuestas han sido pensadas con un enfoque modular y computacionalmente eficiente, lo que las hace aplicables en entornos reales sin necesidad de grandes recursos, facilitando así su transferencia a sistemas de producción.

Los hallazgos de este trabajo tienen una relevancia especial en ámbitos donde los errores generados por modelos pueden tener un impacto real, como el ámbito legal, médico o educativo. El diseño de detectores fiables y la integración de estimadores de incertidumbre no es solo una mejora técnica, sino un imperativo ético en la adopción responsable de estas tecnologías, porque la delegación de decisiones a sistemas que no garantizan veracidad abre la puerta a sesgos, errores y manipulaciones. Por ello, el desarrollo de mecanismos de auditoría algorítmica y el diseño de IA explicables deben ser una prioridad tanto técnica como política.

Este proyecto ha supuesto una oportunidad de profundización en el campo del aprendizaje automático, consolidando no solo competencias técnicas en IA y NLP, sino también habilidades transversales como la investigación aplicada, el pensamiento crítico y la evaluación rigurosa de sistemas inteligentes.

Finalmente, al validar todas las hipótesis formuladas, se concluye que, si bien los modelos de lenguaje han alcanzado un nivel de sofisticación notable, aún están lejos de ser plenamente fiables. Por tanto, no se puede, al menos por el momento, delegar en ellos decisiones que afecten directamente a la vida de las personas o al funcionamiento crítico de nuestra sociedad. Y dado que las alucinaciones parecen ser una propiedad inevitable de estos sistemas, es posible que nunca se alcance un grado de fiabilidad absoluta. En consecuencia, el papel humano en la supervisión, evaluación y control de la inteligencia artificial sigue siendo insustituible.

Esta reflexión también invita a considerar cómo percibe el ser humano a la inteligencia artificial general, como una máquina perfecta e infalible, pero quizá sea más realista y útil concebirla como un "superhumanoíntelectual: una entidad capaz de superar considerablemente nuestras capacidades cognitivas, pero que, al igual que las personas, seguirá cometiendo errores y enfrentando limitaciones inherentes a su propia naturaleza. Desde esta perspectiva, se pueden gestionar mejor las expectativas y diseñar mecanismos adecuados de supervisión y control que permitan aprovechar sus enormes beneficios, al tiempo que se mitigan los riesgos asociados a sus inevitables imperfecciones.

A pesar de los logros alcanzados y de la validación de todas las hipótesis formuladas, este proyecto deja abiertas diversas líneas de investigación que merecen ser exploradas en profundidad. La complejidad del fenómeno de las alucinaciones generadas por modelos de lenguaje, especialmente en contextos numéricos o abiertos, así como los desafíos asociados a su detección y control, plantean nuevas preguntas que trascienden los límites de este estudio. Por ello, en el siguiente apartado se presentan propuestas concretas para extender y profundizar el trabajo realizado, tanto desde una perspectiva técnica como desde un enfoque aplicado y ético.

6.2. Trabajo futuro

Una de las mayores fortalezas de este trabajo reside en la diversidad de líneas de investigación que se abren a partir de los hallazgos obtenidos. La experiencia de desarrollar este proyecto ha sido profundamente enriquecedora, no solo por su valor técnico, sino también por el significado que encierra: comprender mejor los límites de la inteligencia artificial y contribuir, aunque sea modestamente, a su desarrollo responsable. Sin embargo, este avance no implica un punto final. Al contrario, marca el inicio de un extenso camino por recorrer. A continuación, se presentan algunas de las direcciones más prometedoras para el trabajo futuro:

✓ Optimización y diversificación de arquitecturas de detección

- *Modelos Bayesianos y ensamblados profundos*: Explorar esta vía para estimar de forma más precisa la incertidumbre epistemológica. Esto implicaría, por ejemplo, entrenar conjuntos de modelos independientes y comparar su consistencia en predicciones, así como evaluar la calibración de sus probabilidades.
- *Redes neuronales con atención especializada*: Diseñar variantes de **Transformers** que incluyan módulos de atención focalizada en tokens potencialmente “confusos” o “polisémicos”, de modo que se detecten patrones de alucinación basados en la dispersión atencional.
- *Modelos jerárquicos y multimodales*: Incorporar arquitecturas que combinen texto, metadatos (e incluso señales externas). Esto permitiría detectar alucinaciones que surjan de incoherencias entre diferentes modalidades, por ejemplo, una respuesta textual que contradiga información numérica o visual asociada.
- *Clasificadores especializados de última capa*: Además de los clasificadores genéricos, estudiar la viabilidad de *meta-aprendizaje* para ajustar dinámicamente los umbrales de detección en función de la distribución de incertidumbre en tiempo real.

✓ Análisis de colinealidades y reducción de dimensionalidad

- *Emplear técnicas de reducción de dimensionalidad no lineales*: Explorar herramientas que nos permitan ver de forma interactiva cómo se distribuyen los datos en el espacio latente y, así, detectar agrupaciones de ejemplos que suelen tender a alucinaciones.
- *Selección automática de variables relevantes*: Implementar algoritmos de *Boruta* o *LASSO* para identificar subconjuntos de características que aporten mayor señal para la detección, reduciendo el ruido y evitando sobreajuste.
- *Análisis de correlaciones jerárquicas*: Construir mapas de calor de correlación y árboles jerárquicos que muestren qué grupos de variables están altamente correlacionados entre sí y aquellos que están relacionados con la aparición de alucinaciones.

✓ Enriquecimiento con características lingüísticas y semánticas

- *Representaciones sintáctico-semánticas*: Emplear `spaCy`, o alguna técnica similar para extraer etiquetas de rol gramatical, dependencias sintácticas y entidades nombradas, y evaluar su contribución a la detección de incoherencias.
- *Embeddings semánticos contextualizados*: Comparar distintas versiones de *word embeddings* o *sentence embeddings* para medir la similitud semántica entre entrada y respuesta. Un bajo grado de similitud en un par pregunta-respuesta podría señalar posibles alucinaciones.
- *Análisis de polaridad y tono*: Introducir métricas de sentimiento y detección de estilo, con el fin de detectar cambios abruptos de registro que no encajen con la petición original.

✓ Desarrollo de métricas más refinadas

- *Tasa de falsos positivos y negativos específicos por tipo de alucinación*: Además de métricas globales como *accuracy*, *precision*, *recall* y *F1-score*, definir métricas diferenciadas según:
 - *Alucinaciones factual vs. numéricas*: Calcular, por separado, la *tasa de detección de inconsistencias numéricas* y la *tasa de detección de hechos inventados*.
 - *Errores críticos vs. parciales*: Definir una métrica de severidad ponderada en la que las alucinaciones críticas penalicen más que las parciales (detalle secundario incorrecto).
- *Calibración de probabilidad*: Medir si las puntuaciones de confianza del detector están bien calibradas.
- *Métricas de coherencia semántica*: Utilizar *BERTScore* para cuantificar la fidelidad global de la respuesta con respecto al contexto original. Una baja puntuación sugiere mayor probabilidad de alucinación.
- *Evaluación humana escalonada*: Diseñar rúbricas de anotación en las que evaluadores externos puntúen distintos aspectos de la generación y correlacionar esas puntuaciones con las predicciones del detector.

✓ Expansión a otros dominios y lenguajes

- *Entorno legal*: Adaptar el detector para revisar borradores de contratos, resoluciones judiciales o dictámenes, donde una alucinación numérica puede tener consecuencias graves. Diseñar un corpus anotado de documentos legales para entrenar y evaluar el sistema.
- *Ámbito sanitario*: Validar el detector en historiales médicos o recomendaciones clínicas generadas por modelos de lenguaje. Aquí, una alucinación puede comprometer la salud del paciente.
- *Finanzas y economía*: Aplicar la detección en reportes financieros, resúmenes de mercado o análisis de inversiones generados automáticamente.

- *Educación y e-learning*: Incorporar el detector en plataformas de tutoría automática que generen explicaciones o solucionen problemas de matemáticas. Se pueden definir métricas de corrección paso a paso, detectando alucinaciones en razonamientos intermedios.
- *Contenido audiovisual y multimedia*: Extender el análisis a subtítulos generados automáticamente o descripciones de imágenes, combinando detección de incoherencias textuales con verificación de objetos o escenarios mediante visión computacional.

✓ Mitigación activa y estrategias preventivas

- *Retrieval-Augmented Generation (RAG) en producción*: Integrar el detector dentro de un pipeline RAG que, ante bajos niveles de confianza, reenvíe la consulta a un módulo de búsqueda documental que verifique o corrija la respuesta. Se podría medir la mejora al comparar respuestas con y sin la capa de verificación.
- *Verificadores externos en tiempo real*: Desarrollar conectores que, ante cada respuesta generada, consulten en APIs externas para corroborar hechos numéricos o afirmaciones concretas.
- *Prompt engineering adaptativo*: Diseñar métodos de reformulación automática de prompts cuando el detector indique baja confianza.
- *Feedback humano-en-bucle*: Crear interfaces interactivas donde un evaluador humano reciba alertas de posibles alucinaciones y pueda corregirlas. Con ese feedback, reentrenar o ajustar el detector en línea mediante técnicas de aprendizaje continuo.

✓ Estudio de paradigmas *grey-box* y *black-box*

- *Detección grey-box basada en activaciones internas*: Profundizar el análisis de señales intermedias de la red para identificar patrones característicos de alucinación.
- *Métodos black-box robustos*: Diseñar estrategias que sólo usen la consulta-respuesta sin acceso a logits ni pesos.
- *Detectores híbridos*: Combinar señales *grey-box* con estrategias *black-box*.

✓ Evaluación en entornos reales y despliegue

- *Proyectos piloto en empresas y organizaciones*: Exponer el detecto a un uso real, integrado en un flujo de trabajo y recopilar métricas de uso real .
- *Diseño de API y microservicios*: Implementar un servicio que permita enviar texto y recibir una puntuación de “riesgo de alucinación” en milisegundos.

✓ Aspectos éticos, legales y de gobernanza

- *Marco regulatorio y responsabilidad*: Investigar las implicaciones legales de desplegar un detector de alucinaciones en sectores regulados. ¿Quién asume la responsabilidad si el detector falla?

- *Bias y equidad en la detección*: Analizar si el detector introduce sesgos sistemáticos contra determinados tipos de texto. Desarrollar estrategias para mitigar dichos sesgos.
- *Transparencia y explicabilidad*: Implementar métodos de explicabilidad que permitan al usuario entender por qué se ha marcado una respuesta como potencialmente alucinatoria.
- *Data governance y privacidad*: Garantizar que los datos utilizados para entrenar y evaluar el detector cumplan con normativas de protección de datos. Diseñar protocolos que anonimicen textos sensibles antes de alimentar el modelo.

,

Estas líneas de trabajo futuro no sólo ampliarían el alcance técnico del proyecto mediante nuevas métricas, arquitecturas y paradigmas de detección, sino que también explorarían aplicaciones concretas en dominios de alta criticidad, así como aspectos organizacionales y éticos esenciales para su adopción real. Cada una de estas propuestas puede convertirse en un objetivo de investigación independiente, allanando el camino para un avance sostenido en la detección y mitigación de alucinaciones en sistemas de inteligencia artificial.

Bibliografía

- [1] Alammar, J. (2018). The illustrated transformer. <https://jalammar.github.io/illustrated-transformer/>. Último acceso: 29 de mayo de 2025.
- [2] Banerjee, S., Agarwal, A., and Singla, S. (2024). Llms will always hallucinate, and we need to live with this.
- [3] BBC News Mundo (2023). Qué es la "alucinación" de la inteligencia artificial y por qué es una de las fallas potencialmente más peligrosas de esta tecnología. <https://www.bbc.com/mundo/noticias-65606089>. Publicado el 15 de mayo de 2023. Consultado el 19 de mayo de 2025.
- [4] Butvinik, D. (2024). Mathematical insights into temperature scaling and hallucination in large language models. <https://dannybutvinik.medium.com/mathematical-insights-into-temperature-scaling-and-hallucination-in-large-language-models-85e55ca1f5>. Publicado el 3 de Abril de 2024. Último acceso: mayo de 2025.
- [5] Detecting-AI.com (2025). Ultimate guide to hallucination detection metrics. Consultado el 1 de junio de 2025.
- [6] Elchafei, P. and Abu-Elkheir, M. (2025). Span-level hallucination detection for llm-generated answers.
- [7] Fareed Khan (2025). Solving 6 Types of Hallucinations in Small LLMs: A Hands-On Guide. <https://levelup.gitconnected.com/solving-6-types-of-hallucinations-in-small-llms-a-hands-on-guide-8d15c11650d3>. Publicado el 5 de Mayo de 2025. Consultado el 19 de mayo de 2025.
- [García-Ferreira] García-Ferreira. Conoce todo sobre el dropout en redes neuronales. Último acceso: 3 de junio de 2025.
- [9] Goh, H. W. (2024a). Benchmarking hallucination detection methods in rag. <https://towardsdatascience.com/benchmarking-hallucination-detection-methods-in-rag-6a03c555f063/>. Publicado el 9 de Septiembre de 2024. Último acceso: mayo de 2025.
- [10] Goh, H. W. (2024b). Benchmarking hallucination detection methods in rag. <https://medium.com/data-science/>

- benchmarking-hallucination-detection-methods-in-rag-6a03c555f063. Accessed: 2025-05-20.
- [11] IBM (2023). ¿Qué son las alucinaciones de IA? <https://www.ibm.com/es-es/think/topics/ai-hallucinations>. Publicado el 1 de septiembre de 2023. Consultado el 19 de mayo de 2025.
- [12] Ip, J. (2025). Llm-as-a-judge simply explained: A complete guide to run llm evals at scale. Último acceso: 1 de junio de 2025.
- [13] Javier García Correas (2024). AirCanada a juicio por errores de su ChatBot. <https://es.linkedin.com/pulse/aircanada-juicio-por-errores-de-su-chatbot-javier-garc%C3%ADa-correas-c6kzf>. Publicado el 15 de Abril de 2024. Consultado el 19 de mayo de 2025.
- [14] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.
- [15] Jin, Q., Dhingra, B., Liu, Z., Cohen, W., and Lu, X. (2019). Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.
- [16] Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., and Kaplan, J. (2022). Language models (mostly) know what they know.
- [17] Kolena (2024). Hughes hallucination evaluation model (hhem) score. Accessed: 2025-05-20.
- [18] Levy, A. A. and Geva, M. (2025). Language models encode numbers using digit representations in base 10.
- [19] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.
- [20] Ip, J. (2025). Llm-as-a-judge simply explained: A complete guide to run llm evals at scale. <https://www.confident-ai.com/blog/why-llm-as-a-judge-is-the-best-llm-evaluation-method#does-it-really-work->. Publicado el 27 de Abril de 2025. Último acceso: mayo de 2025.
- [21] Manakul, P., Liusie, A., and Gales, M. J. F. (2023). Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

- [22] Michael E. Bakich (2005). First exoplanet imaged. https://www.astronomy.com/science/first-exoplanet-imaged/?utm_source=chatgpt.com. Publicado el 3 de Mayo de 2005. Consultado el 19 de mayo de 2025.
- [23] MÉDICA, R. (2025). Examen mir 2025, al completo: todas las respuestas definitivas corregidas. Último acceso: 2 de junio de 2025.
- [Ng] Ng, A. Had an insightful conversation with geoffrey hinton about whether ai systems understand. Accedido el: 5 de junio de 2025.
- [25] OpenAI (2025). Text generation and prompting. Último acceso: 2 de junio de 2025.
- [26] Palmarini, M. (2005). *Los túneles de la mente: ¿Qué se esconde tras nuestros errores?* Biblioteca de bolsillo. Crítica.
- [27] Quevedo, E., Yero, J., Koerner, R., Rivas, P., and Cerný, T. (2024). Detecting hallucinations in large language model generation: A token probability approach. *ArXiv*, abs/2405.19648.
- [28] Sarmiento, P. (2025). Aluora: A python library for hallucination detection in llms. <https://github.com/Pedro-Sarmiento/Aluora>. Accedido el: 5 de junio de 2025.
- [29] Thawani, A., Pujara, J., Szekely, P. A., and Ilievski, F. (2021). Representing numbers in nlp: a survey and a vision.
- [30] The New York Times (2023). What Makes A.I. Chatbots Go Wrong? <https://www.nytimes.com/2023/03/29/technology/ai-chatbots-hallucinations.html>. Publicado el 29 de Marzo de 2023. Consultado el 19 de mayo de 2025.
- [31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.
- [32] Wang, C., Liu, X., Yue, Y., Tang, X., Zhang, T., Jiayang, C., Yao, Y., Gao, W., Hu, X., Qi, Z., Wang, Y., Yang, L., Wang, J., Xie, X., Zhang, Z., and Zhang, Y. (2023). Survey on factuality in large language models: Knowledge, retrieval and domain-specificity.
- [33] Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Cooper, N., Adams, G., Howard, J., and Poli, I. (2024). Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference.
- [34] Ádám Kovács and Recski, G. (2025). Lettucedetect: A hallucination detection framework for rag applications.

Capítulo 7

Otros capítulos y anexos

7.1. Expresiones matemáticas

Derivada de la función Softmax respecto a la temperatura

Queremos calcular la derivada parcial de la probabilidad $p_i(T)$ con respecto a la temperatura T , siendo:

$$p_i(T) = \frac{e^{z_i/T}}{\sum_{j=1}^K e^{z_j/T}}$$

Aplicamos la regla del cociente:

$$u(T) = e^{z_i/T} \tag{7.1}$$

$$v(T) = \sum_{j=1}^K e^{z_j/T} \tag{7.2}$$

Derivada de $u(T)$:

$$\frac{\partial u(T)}{\partial T} = -\frac{z_i}{T^2} \cdot e^{z_i/T}$$

Derivada de $v(T)$:

$$\frac{\partial v(T)}{\partial T} = \sum_{j=1}^K \left(-\frac{z_j}{T^2} \cdot e^{z_j/T} \right)$$

Aplicando la regla del cociente:

$$\frac{\partial p_i(T)}{\partial T} = \frac{-\frac{z_i}{T^2} e^{z_i/T} \cdot \sum_{j=1}^K e^{z_j/T} + e^{z_i/T} \cdot \sum_{j=1}^K \frac{z_j}{T^2} e^{z_j/T}}{\left(\sum_{j=1}^K e^{z_j/T} \right)^2}$$

Simplificando:

$$\frac{\partial p_i(T)}{\partial T} = p_i(T) \left[-\frac{z_i}{T^2} + \frac{1}{T^2} \cdot \frac{\sum_{j=1}^K z_j e^{z_j/T}}{\sum_{j=1}^K e^{z_j/T}} \right]$$