UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,INFORMÁTICA Y MECÁNICA

ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS

TESIS

EVALUACIÓN DE LARGE LANGUAGE MODEL EN LA GENERACIÓN DE RESPUESTAS PARA PREGUNTAS SOBRE LAS LEYES DEPORTIVAS DE AJEDREZ

PRESENTADO POR:
Br. ROGER YUPANQUI HUAMAN

PARA OPTAR AL TÍTULO PROFESIONAL DE INGENIERO INFORMÁTICO Y DE SISTEMAS

ASESOR:
Dr. EDWIN CARRASCO POBLETE

CUSCO - PERÚ

2026

Universidad Nacional de San Antonio Abad del Cusco

INFORME DE SIMILITUD

(Aprobado por Resolución Nro.CU-321-2025-UNSAAC)

El que suscribe, el **Asesor** EDWIN CARRASCO POBLETE ..... quien aplica el software de detección de similitud al trabajo de investigación/tesistitulada: EVALUACIÓN DE LARGE LANGUAGE MODEL EN LA GENERACIÓN DE RESPUESTAS PARA PREGUNTAS SOBRE LAS LEYES DEPORTIVAS DE AJEDREZ .....

Presentado por: ROGER YUPANQUI HUAMAN ..... DNI N° 44764030 .....;
presentado por: ..... DNI N°: .....
Para optar el título Profesional/Grado Académico de INGENIERO INFORMÁTICO Y DE SISTEMAS .....

Informo que el trabajo de investigación ha sido sometido a revisión por ....5.... veces, mediante el Software de Similitud, conforme al Art. 6° del ***Reglamento para Uso del Sistema Detección de Similitud en la UNSAAC*** y de la evaluación de originalidad se tiene un porcentaje de .....07..... **%.**

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No sobrepasa el porcentaje aceptado de similitud.	X
Del 11 al 30 %	Devolver al usuario para las subsanaciones.	
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, conforme al reglamento, quien a su vez eleva el informe al Vicerrectorado de Investigación para que tome las acciones correspondientes; Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y **adjunto** las primeras páginas del reporte del Sistema de Detección de Similitud.

Cusco, .06.. de .....JULIO..... de 20..26.....

.....
Firma

Post firma..EDWIN CARRASCO POBLETE.....

Nro. de DNI....24001157.....

ORCID del Asesor....0000-0003-4437-3960.....

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema de Detección de Similitud: **oid:** 27259:605126279.....

ROGER YUPANQUI HUAMAN

EVALUACIÓN DE LARGE LANGUAGE MODEL EN LA GENERACIÓN DE RESPUESTAS PARA PREGUNTAS SOBRE LAS...

Universidad Nacional San Antonio Abad del Cusco

Detalles del documento

Identificador de la entrega
trn:oid:::27259:605126279

Fecha de entrega
6 jul 2026, 11:11 a.m. GMT-5

Fecha de descarga
6 jul 2026, 12:08 p.m. GMT-5

Nombre del archivo
EVALUACIÓN DE LARGE LANGUAGE MODEL EN LA GENERACIÓN DE RESPUESTAS PARA PREGUNTA....pdf

Tamaño del archivo
1.8 MB

143 páginas

30.355 palabras

162.081 caracteres

7% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 10 palabras)

Fuentes principales

6% Fuentes de Internet
1% Publicaciones
5% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Dedicatoria

A Dios por ser mi guía y fortaleza en todo tiempo.

A mis padres, por su inmenso amor y apoyo incondicional durante todo este proceso.

A mis hermanos, por ser un ejemplo en mi vida.

A Carmen Rosa, por ser mi mejor motivación para llegar hasta aquí.

Agradecimientos

Agradezco a Dios que siempre ilumina ni vida, a mis padres por todo lo que hicieron por mi, a mi hermano Oscar por su gran ayuda durante mis años de estudiante, al Dr. Jean Franco Escobedo, por su amistad y apoyo en este proyecto, a mi asesor por su paciencia y dedicación durante la elaboración de esta investigación, finalmente a todos los que hicieron posible la culminación de esta etapa: docentes, compañeros y amigos, gracias.

Resumen

Los Large Language Model (LLM) vienen demostrando un desempeño muy sobresaliente en tareas del procesamiento del lenguaje natural (NLP). Sin embargo, su rendimiento en tareas de preguntas y respuestas en dominios cerrados necesita un análisis más profundo. El manual deportivo de ajedrez representa un dominio cerrado y estructurado lo que hace pertinente la evaluación de los LLMs en este escenario. Este estudio plantea la evaluación de los LLMs considerando las arquitecturas encoder, encoder-decoder y decoder aplicando sobre ellos las métricas Strict Accuracy, Lenient Accuracy, Similitud Semántica y que fueron complementadas con evaluación humana que es indispensable en este tipo de tareas. Para ello se elabora un dataset basado en le manual deportivo de ajedrez considerando el contexto, la pregunta y la respuesta asociada a la misma. Los resultados muestran que los LLMs tienen un buen desempeño en responder preguntas donde la información se halla directamente en el contexto. Por otro lado, se observan limitaciones en las preguntas que necesitan un razonamiento adicional o la aplicación de varias reglas al mismo tiempo. Se concluye que los LLMs son de gran utilidad en tareas de preguntas y respuestas, su aplicación en dominios cerrados requiere de evaluaciones específicas, además que la elaboración de un dataset es un proceso complejo y necesario para una buena evaluación de los modelos.

Palabras clave: Large Language Models, Encoder, Decoder, Strict Accuracy, Lenient Accuracy.

Abstract

Large Language Models (LLMs) have been demonstrating outstanding performance in natural language processing (NLP) tasks. However, their performance in closed-domain question-answering tasks requires further analysis. The chess handbook represents a closed and structured domain, making the evaluation of LLMs in this scenario pertinent. This study proposes evaluating LLMs—considering encoder, encoder-decoder, and decoder architectures—by applying Strict Accuracy, Lenient Accuracy, and Semantic Similarity metrics, complemented by human evaluation, which is indispensable for tasks of this nature. To this end, a dataset is created based on a chess handbook, taking into account the context, the question, and the corresponding answer. The results show that LLMs perform well in answering questions where the information is found directly within the context. On the other hand, limitations are observed in questions requiring additional reasoning or the simultaneous application of multiple rules. It is concluded that LLMs are highly useful for question-answering tasks; however, their application in closed domains requires specific evaluations, and the creation of a dataset is a complex yet necessary process for effectively evaluating the models.

Keywords: Large Language Models, Encoder, Decoder, Strict Accuracy, Lenient Accuracy.

Listado de Abreviaturas

BERT	Bidirectional Encoder Representations from Transformers
FDEA	Federación Deportiva Española de Ajedrez
FIDE	Federación Internacional de Ajedrez
GPT	Generative Pre-trained Transformer
IA	Inteligencia Artificial
KAI	Knowledge Acquisition and Inferencing
LLM	Large Language Model
NLG	Natural Language Generation
NLP	Procesamiento del Lenguaje Natural
NLU	Natural Language Understanding
QA	Preguntas y Respuestas
SQAC	Spanish Question Answering Corpus
T5	Text-to-Text Transfer Transformer

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Generalidades

En los últimos años el Procesamiento del Lenguaje Natural tuvo un gran repunte dentro de la inteligencia artificial, esto motivado en gran medida por el desarrollo de los modelos de lenguaje largo (Large Language Model) que han permitido grandes avances en tareas como: traducción automática, resumen de textos y preguntas y respuestas (QA por sus siglas en ingles) (Alammar and Grootndorst, 2024). Estos modelos que se basan en la arquitectura Transformers han mostrados notables desempeños de tareas de preguntas y respuestas, donde son capaces de generar respuestas coherentes en lenguaje natural. Este progreso a motivo su adopción en muchos dominios como son: educación, toma de decisiones, atención al cliente, etc.

A pesar de estos avances, la evaluación de los LLMs en tareas de preguntas y respuestas sigue siendo un desafío abierto sobre todo en dominios cerrados (Rajpurkar et al., 2016). Los modelos presentan configuraciones, métricas y conjuntos de datos que dificultan realizar comparaciones entre los mismos, además la mayor parte de estudios se centrar en el rendimiento cuantitativo sin analizar los aspectos cualitativos como la coherencia, veracidad

o capacidad de razonamiento, esto limita la comprensión de la capacidad real que tienen los modelos(Vaswani et al., 2023). La falta de un marco de evaluación claro para los LLMs en tareas de preguntas y respuestas es un problema cuando se quiere hacer uso de un modelo. Esta carencia nos genera la incertidumbre de que modelo es el más adecuado en un escenario específico, especialmente en aplicaciones donde la precisión y la confiabilidad de la información son vitales.

Existen diversos trabajos en cuanto a evaluación de LLMs en QA los cuales usan dataset como SQuAD sin embargo estos estudios presentan limitaciones relacionadas con el tipo de preguntas y la dependencia de las respuestas extractivas. Estudios más recientes incorporan métricas automáticas y evaluaciones humanas, aun y con todo eso no existe un consenso de cual es la mejor ruta en la evaluación en tareas de preguntas y respuestas.

Por lo anteriormente mencionado la presente investigación tiene como objetivo evaluar el desempeño de los LLMs en la generación de respuestas a preguntas basadas en el manual de ajedrez. Para ello se propone un enfoque metodológico que incluye la selección de modelos, la definición de un conjunto de diversas preguntas y la aplicación de métricas como Strict Accuracy, Lenient Accuracy y Similitud Semántica, además de la evaluación humana que continúa siendo indispensable en este tipo de dominios.

La importancia de este estudio radica en brindar una base solidad en la comparación de los LLMs específicamente en tareas de preguntas y respuestas, lo cual es imprescindible en la implementación de aplicaciones reales, al hacer uso de un dominio cerrado estos modelos podrían emplearse en entornos de ligas y clubes, especialmente durante torneos donde los jugadores y árbitros necesitan respuestas claras y precisas respecto a las leyes del ajedrez y a las bases de los torneos. De manera más amplia, una evaluación rigurosa contribuye a mejorar la confiabilidad de estos sistemas en dominios críticos.

Finalmente, el presente trabajo esta organizado de la siguiente manera: En el capítulo

1 se presenta las generalidades, la justificación, el planteamiento del problema, alcances y limitaciones, los objetivos y los antecedentes, en el capítulo 2 se presentan las bases teóricas, en el capítulo 3 se presentan metodología de la investigación, el capítulo 4 se centra en el desarrollo del proyecto, el capítulo 5 muestra los resultados y discusiones. Finalmente se presentan las conclusiones obtenidas en relación con los objetivos planteados, también mostramos recomendaciones sobre trabajos futuros en tareas de preguntas y respuestas usando Large Language Model.

1.2. Justificación

Esta investigación propone realizar una evaluación de los LLMs en tareas de QA en un dominio cerrado, propiamente dicho en el manual deportivo de ajedrez. En el procesamiento de la data se propone la fragmentación del texto en contextos para la extracción de preguntas y respuestas además de consultas con árbitros nacionales, dado que las formas automáticas de generar preguntas y respuestas en contextos suelen presentar inconsistencias. Se propone el uso de diccionarios Python para el almacenamiento de la data y por ultimo se plantea hacer uso de las métricas Strict Accuracy, Lenient Accuracy (Hu and Zhou, 2024), sumado a estas dos métricas se propone hacer uso de evaluación Humana para el desarrollo de nuestro proyecto.

1.2.1. Impacto Social

El impacto social de este trabajo se centra dentro de la comunidad ajedrecística por su potencial para mejorar la toma de decisiones durante los torneos. Árbitros y jugadores se enfrentan con frecuencia a situaciones donde es importante tener una interpretación clara de las reglas del juego, en tal sentido un sistema basado en LLMs, correctamente evaluado respecto al manual deportivo de ajedrez, sería una herramienta de apoyo que proporcione

respuestas rápidas y coherentes, aumentando así la transparencia en la aplicación de las normas (Brown et al., 2020). Por otra parte, este enfoque contribuye a una estandarización del conocimiento normativo, dentro de la comunidad arbitral ya que en muchos casos la interpretación del reglamento puede variar según la experiencia del árbitro o del contexto del torneo, utilizando un modelo evaluado específicamente en el manual deportivo de ajedrez, se promueve una interpretación más uniforme del reglamento, lo cual beneficia tanto a competidores como a organizadores. Este aspecto es particularmente relevante en torneos de diferentes niveles, donde no siempre se dispone de árbitros altamente experimentados.

1.2.2. Implicancias Prácticas

Las implicaciones prácticas con lleva a sentar las bases para el desarrollo de asistentes inteligentes especializados que puedan integrarse en plataformas digitales de torneos, aplicaciones móviles o sistemas de arbitraje electrónico (Bommasani et al., 2021). Estas herramientas serian de gran ayuda no solo durante una competición, sino también en procesos de capacitación y formación de árbitros y jugadores, brindando así herramientas que ayuden en el acceso al conocimiento normativo.

1.2.3. Valor Teórico

El valor teórico radica en la evaluación de Large Language Models (LLMs) en contextos normativos especializados, específicamente en tareas de preguntas y respuestas basadas en documentos oficiales (Chen et al., 2019). Este trabajo aporta un enfoque orientado a dominios donde la exactitud semántica y la fidelidad al texto fuente son críticas, lo que permite ampliar la comprensión teórica sobre las capacidades y limitaciones de los LLMs en escenarios de alta precisión (Ji et al., 2022). Las métricas de evaluación nos permiten conocer el comportamiento de los LLMs con nuestros datos y también ver como se deben ajustar los

contextos y las preguntas para obtener mejores respuestas (Hu and Zhou, 2024).

1.3. Planteamiento del problema

Los grandes avances en el campo de la inteligencia artificial generativa sobre todo en el Procesamiento del Lenguaje Natural, lograron que los LLMs se popularizaran rápidamente en los últimos años, su aplicación en tareas como: asistentes virtuales, generación de resúmenes, análisis de texto y respuestas automáticas han puesto la mira en estos modelos sobre su eficacia al realizar estas tareas (Brown et al., 2020). Una tarea que desarrollan los LLMs son las llamadas preguntas y respuestas (QA por sus siglas en ingles) que se caracterizan por buscar brindar respuestas acertadas en diferentes áreas. Sin embargo, al ser entrenados con grandes corpus de datos y generalmente en idioma inglés, pueden brindar respuestas diferentes, deficientes, incompletas o incluso presentar alucinaciones propias en los modelos (Ji et al., 2022). El desempeño de los LLMs en dominios cerrados, como normas o reglamentos, en muchas ocasiones no fueron verificados lo que ocasiona, en muchos casos, fallas en las tareas de QA.

Las leyes deportivas del ajedrez representan un dominio cerrado y especializado por el uso de un lenguaje preciso que además contiene reglas, la Federación Internacional de Ajedrez (FIDE) es la encargada de elaborar este documento que cubre todas las situaciones propias del juego de ajedrez, desde los movimientos hasta situaciones arbitrales (Arbiters, 2024). Las respuestas dentro de este dominio deben de ser precisas y coherentes.

Aunque muchos de los LLMs fueron evaluados a tareas de QA, hay poca información para el manejo de los mismos en contextos especializados. El desafío que enfrentan estos modelos radica en ser capaces de comprender el contexto antes de emitir su respuesta, además de aplicar reglas si fuera necesario.

Este estudio planea evaluar los LLMs en la generación de respuestas en el dominio

especializado de las leyes del ajedrez, además de comprender la capacidad de los LLMs en este dominio en el idioma español. A lo largo de esta investigación se analizarán los errores de los modelos, los recursos en español con que cuentan y la evaluación de las respuestas en diferentes contextos como son: estructurales, situacionales y ambiguos, con el fin de ver como los LLMs actúan en ese tipo de escenarios.

1.3.1. Formulación del Problema

Problema general

¿Qué tan efectivos son los LLM en la generación de respuestas correctas basadas en contextos del manual de ajedrez, considerando los diferentes niveles de complejidad?

Problemas específicos

- ¿Qué tipos de errores cometen los LLMs al generar respuestas a preguntas directas, condicionales y situacionales basadas en el manual de ajedrez?
- ¿Cómo se ve afectado el rendimiento de los diferentes modelos en preguntas de razonamiento?
- ¿Qué métricas permiten evaluar de manera confiable las respuestas generadas por los modelos?

1.4. Alcances y Limitaciones

1.4.1. Alcance

- Nuestra data de contexto se regirá exclusivamente al manual de la Federación Internacional de Ajedrez

- Para la generación de las preguntas y respuestas se tomará en cuenta evaluaciones de árbitros de ajedrez, casuisticas de torneos y preguntas simples de usuarios de ajedrez.
- Se elegirán 3 modelos a ser evaluados, considerando los parámetros, el corpus de entrenamiento y la arquitectura del modelo.
- La evaluación humana se realizará con apoyo de un árbitro internacional, un árbitro FIDE y un árbitro Nacional de ajedrez, todos avalados por la FIDE.
- La recuperación de la data se realizará sobre asuntos de competiciones para ajedrez estándar, rápido y relámpago.

1.4.2. Limitaciones

- Potencia de computo, GPU, se hace uso de la herramienta google colab en su versión gratuita, por la potencia de computo que exige el uso de los LLMs.
- Disponibilidad de dataset, no existe un dataset de preguntas y respuestas del manual deportivo de ajedrez, por ello se elabora un conjunto de datos que incluyen contextos con su pregunta y respuesta asociadas respectivamente.
- No se consideran imágenes de posiciones del manual para la elaboración de los contextos.

1.5. Objetivos

1.5.1. Objetivo general

Evaluar el rendimiento de los LLMs en la generación de respuestas a partir de contextos del manual de ajedrez, considerando los tipos de preguntas y contextos

1.5.2. Objetivos específicos

- Analizar los errores cometidos por los LLMs en la generación de respuestas a preguntas basadas en el manual de ajedrez
- Comparar el rendimiento de los LLMs al responder preguntas de razonamiento sobre el manual de ajedrez.
- Determinar las métricas adecuadas para evaluar la precisión de los modelos en las respuestas generadas
- Elaborar un dataset QA con contextos: estructurales, situacionales y multievidencia del manual deportivo de ajedrez.

1.6. Antecedentes

1. **Luis Martín de Pablo (2024). Evaluación y Comparativa de Modelos de Lenguaje a Gran Escala (LLM) en Local para Arquitecturas Retrieval Augmented Generation (RAG), Universidad Oberta de Catalunya, Cataluña - España**

   El objetivo de este trabajo de investigación consiste en evaluar la efectividad de los Modelos de Lenguage a Gran escala dentro de entornos locales para arquitecturas de Retrieval Augmented Generation (RAG). Entre los modelos estudiados destacan modelos pre entrenados como: GPT y LLaMA.

   Para la evaluación de los LLM se hace uso de librería RAGAs, muy útil para aplicaciones de tipo RAG, se evalúan específicamente la precisión y la coherencia.

   Ademas este estudio busca determinar si la ejecución de los LLMs en local puede ofrecer resultados similares en precisión y eficacia a los generados en la nube. (Martín de Pablo,

2024).

Comentario: El Trabajo ejecuta los modelos en forma local mostrando un cuidado en la privacidad de la información, para evaluación de la arquitectura RAG se tuvo que descomponer la misma en sus cuatro componentes principales: modelo de embedding, base de datos vectorial, modelo de lenguaje y procesador de PDF. Se considera al motor de recuperación como el principal responsable de la eficiencia del sistema RAG, por otro lado los LLMs de gran tamaño muestran desempeños similares a los obtenidos con los LLMs mas pequeños.

2. **Liang, Percy (2023). Holistic Evaluation of Language Models, Stanford University.**

En este trabajo se propone la Evaluación Holística de Modelos Lingüísticos (HELM) con el fin de mejorar los LLMs, HELM utiliza 16 escenarios para la evaluación de los LLMs donde se usan 7 categorías métricas, en los escenarios abarcan tareas como: respuesta a preguntas, recuperación de información, resumen, detección de toxicidad, estas evaluaciones se realizaron exclusivamente en el idioma ingles, ademas las 7 métricas utilizadas fueron: precisión, calibración, robustez, imparcialidad, sesgo, toxicidad y eficiencia. Los modelos evaluados son 30 de los cuales 10 son de codigo abierto, 17 de acceso limitado y 3 cerrados, entre los cuales destacan: Anthropic-LM v4-s3 (52B), davinci (175B), YaLM (100B), T5 de Google (que será usado en nuestro proyecto) entre otros.

Para evaluar tareas de Preguntas y respuesta esta investigación uso conjuntos de datos enormes como: NaturalQuestions, NarrativeQA y QuAC.

Se realiza evaluación humana en 6 modelos: AnthropicLM v4-s3 (52B), Cohere xlarge v20220609 (52.4B), OPT (175B), davinci (175B), text-davinci-002 y GLM (130B). (Liang et al., 2023)

Comentario: Este extenso trabajo pone de manifiesto la necesidad de evaluar un LLM antes de su uso aplicativo, busca mostrar la transparencia de los LLM en diversas tareas, se observa que la precisión y la calibración depende del escenario donde actua el LLM, esto pone de manifiesto que un LLM puede ser bueno en una determinada tarea y no ser en otra. Es importante considera un dataset en la evaluación de tareas de preguntas y respuestas

3. **Cheng-Han Chiang y Hung-yi Lee (2023). Can Large Language Models Be an Alternative to Human Evaluations?**

   Este estudio pone en el tapete si un LLM puede reemplazar la evaluación humana, para esto se realiza una serie de preguntas tanto al LLM como al evaluador humano, esta evaluación se realiza en tareas de generación de texto, como modelo generador se uso GPT-2 y como modelos evaluadores se usaron a: T0, Curie, Davinci y ChatGPT. Además se pone de manifiesto la debilidad de la evaluación humana (la reproducibilidad) ya que es complicado reclutar los mismos evaluadores para repetir una evaluación y aún y si se reclutasen a los mismo evaluadores es posible que sus resultados difieran en la próxima evaluación. (Cheng-Han Chiang, 2024)

   Comentario: Es interesante la propuesta de evaluar el rendimiento de un LLM con otro LLM, este es un tema que se viene investigando mucho en la actualidad, una de las razones de que la evaluación con LLMs sea algo muy llamativo es el hecho de que la evaluación humana es muy costosa en tiempo y dinero, por tal motivo usar un LLM para realizar las evaluaciones parece una solución adecuada a estos problemas, aún así la evaluación humana sigue siendo considerada el standar de oro en evaluación de LLMs.

4. **Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke y Davood Rafiei (2023). Evaluating Open-Domain Question Answering in the Era of Large Language Models**

   Este estudio propone una evaluación de modelos de Question and Answer de dominio abierto que incluyen los LLM, sobre sale la eficacia del modelo InstructGPT en Zero-Shot, es importante reconocer que los modelos automatizados tienen dificultades en la deteccion de respuestas razón por la cual la evaluación de respuestas puede verse afectada, al igual que otras investigaciones se concluye que no hay sustituto para la evaluación humana. Entre los modelos evaluados se tiene: InstructGPT (zero-shot), InstructGPT (few-shot), DPR, FiD, ANCE + & FiD, RocketQAv2 & FiD, Contriever & FiD, FiD-KD, GAR+ & FiD, EviGen, EMDR, R2-D2. Las métricas de evaluación usadas fueron: Exact-Match accuracy y F1 score y el dataset seleccionado fue NQ-OPEN que consta con más de 3000 preguntas para pruebas.(Ehsan Kamalloo, 2023)

   Comentario: El emparejamiento léxico como método de evaluación en preguntas y respuestas resulta ser muy rígido esto por la probabilidad de que respuestas candidatas o generadas por los modelos no aparezcan en la lista de respuestas de referencia, la evaluación manual aplicada en este estudio muestra otra vez la necesidad de expertos en un área para someter a evaluación las respuestas generadas por los modelos.

5. **Satyadhar Josh (2025). Evaluation of Large Language Models: Review of Metrics, Applications, and Methodologies**

   Este estudio proporciona un amplio entorno de métricas y metodologías empleadas en la evaluación de los LLM, entre las métricas usadas destacan: precisión, coherencia, relevancia y seguridad; los dominios en los que se aplico las evaluaciones fueron tres: medicina, finanzas y leyes. Las comparaciones fueron hechas usando herramientas y plataformas de evaluación lo cual ayudó a evaluar más de 50 métricas en los dominios

mencionados. Entre los modelos evaluados destacan: GPT-4, Claude, y otros modelos abiertos. La evaluación humana no es ajena en esta investigación, lo que pone una vez más la necesidad de incluir esta evaluación en nuestra investigación. (Joshi, 2025)

Comentario: La importancia de contar con métricas para evaluar los LLMs queda de manifiesto, entre las conclusiones resaltantes de esta investigación se considera que los LLM tienen mucho potencial en tareas de dominio especializados, sin embargo requieren ser sometidos a evaluación antes de su uso aplicativo, la adaptación del dominio y la supervisión humana siguen siendo vitales en este tipo de evaluaciones. Consideramos importante las consideraciones en cuanto al costo computacional mencionadas en esta investigación sobre todo si se desea implementar aplicaciones con LLMs.

6. **Taojun Hu, Xiao-Hua Zhou (2024). Unveiling LLM Evaluation Focused on Metrics: Challenges and Solutions**

   Este estudio se centra principalmente en las métricas que se usan en la evaluación de LLMs, la aplicación de esta métricas fueron realizadas en el campo de la biomedicina en diferentes tareas entre las que se considera preguntas y respuestas. Este trabajo realiza un estudio profundo respecto a las métricas de evaluación de LLMs en la actualidad, junto a sus formulas matemáticas y explicaciones estadísticas haciendo uso de bibliotecas de código abierto. La propuesta de este trabajo es evaluar a los LLM en: Precisión, ética, imparcialidad, generalización, robustez, razonamiento; sobre datos biomédicos y haciendo uso de bibliotecas de código abierto. Para tareas de Question and answer este estudio propone las métricas: Strict Accuracy (SaCC) y Lenient Accuracy (LaCC). Entre los modelos evaluados destacan: BioBERT, BioGPT, BioLinkBERT, BioMedGPT, BioMegatron, ClinicalBERT, DoT5, NEM, CheXbert, ELECTRAMed, GatorTronGPTetc, todos ellos evaluados en el campo de la biomedicina. (Hu and Zhou, 2024)

Comentarios: La importancia de las métricas al momento de evaluar un LLM son de suma importancia, recordemos que no es lo mismo evaluar un modelo en tareas de análisis de sentimientos que un modelo usado en tareas de preguntas y respuestas, esto hace que nuestras métricas cambien dependiendo de la tarea y del dominio en el que planeamos usar nuestro LLM, una característica de las tareas de preguntas y respuestas es su alta precisión aun más cuando se trata de dominios reglamentarios como en nuestra propuesta de tesis, por tal motivo nosotros usaremos Strict Accuracy (SaCC) y Lenient Accuracy (LaCC) como métricas principales de evaluación en nuestro proyecto.

7. **Latif Atif, Kim Jihie (2024). Evaluation and Analysis of Large Language Models for Clinical Text Augmentation and Generation**

   Este estudio investiga la capacidad de los LLMs en su capacidad de generar texto clínico. Los modelos utilizados son: T5, BART (Modelo de Transformer Bidireccional y Auto-Regresivo para tareas de generación de texto), además de utilizar el conjunto de datos CHARDAT que se centra en condiciones de la salud. Las preguntas formuladas en el dataset son en idioma ingles y las respuestas son estructuradas según las dimensiones clínicas. Las métricas que se utilizaron en este estudio son ROUGE-1 para medir la coincidencia de unigramas, ROUGE-2 para medir la coincidencia de bigramas y ROUGE-L para medir la coincidencia de la longitud de la subsecuencia común más larga (Latif and Kim, 2024)

   Comentarios: T5 y BART muestran una gran efectividad en tareas de generación y aumento de texto, el uso de una dataset especializado también se pone de manifiesto al hacer uso de CHARDAT, cabe resaltar que la técnica de aumento de datos mejora la calidad de los datos, sin embargo se observaron limitaciones en el manejo de terminologías especializada en salud, lo pone en evidencia la necesidad de adaptar los modelos

a dominios específicos.

8. **Kevin Fischer, Darren Fürst, Sebastian Steindl, Jakob Lindner, Ulrich Schäfer (2024). Question: How do Large Language Models perform on the Question Answering tasks? Answer**

Este estudio muestra la comparacion de modelos de uso general con modelos más pequeños y ajustado en tareas de preguntas y respuestas; entre los modelos evaluados tenemos LLaMa, GPRT-4 Turbo, Flan-T5 y RoBERTa, el conjunto de datos utilizado fue el Stanford Question Answering Dataset 2.0 (SQuAD2.0) y otros dataset sin realizar fine-tune. Las métricas usadas fueron: Exact Match y F1 Score (Fischer et al., 2024). Los modelos pequeños ajustados finamente fueron superiores o los modelos generales en el conjunto de datos de SQuAD2.0, pero en preguntas y respuestas fuera de distribución los modelos grandes mostraron un rendimiento superior a los pequeños.

Tabla 1: Modelos Evaluados

Modelo	**Tipo**	**Características**
LLaMA2 7B	Código abierto	Arquitectura Transformer
LLaMA3.1 8B	Código abierto	Mejoras en comprensión
LLaMA3.1 70B	Código abierto	Mejoras en tareas complejas
LLaMA3.2 1B	Código abierto	Variante ligera
LLaMA3.2 3B	Código abierto	Buen rendimiento y eficiencia
GPT-4 Turbo	LLM propietario	Modelo de OpenAI
Flan-T5	Modelo afinado	Ideal para QA
DistilBERT	Modelo afinado	Versión ligera de BERT
RoBERTa	Modelo afinado	Variante de BERT mejorado

Fuente: Elaborado a partir de los resultados del trabajo citado.

Comentarios: Una parte importante de este estudio es que el dataset SQuAD2.0 contiene preguntas sin responder, por lo tanto si nuestro dataset es muy grande se debe buscar maneras de mitigar estos huecos en las respuestas usando alguna técnica como "single-inference prompting"(que es la que se uso en el estudio), que busca responder a preguntas sin respuesta, si bien es cierto que los modelos afinados para tareas de preguntas y respuestas mostraron un mejor rendimiento que los LLM generales en el

dataset SQuAD2.0, no ocurrió lo mismo con otros dataset, entonces caemos en cuenta que una evaluación de los LLMs antes de llevarlo al desarrollo aplicativo es más que necesaria.

9. **Amer Farea, Zhen Yang, Kien Duong, Nadeesha Perera, Frank Emmert-Streib (2022). Evaluation of Question Answering Systems: Complexity of judging a natural language**

   Este estudio nos muestra la complejidad que existe en la evaluación de sistemas de preguntas y respuestas como parte del PNL. Muestra un enfoque completo sobre los sistemas de preguntas y respuestas, los conjuntos de datos a utilizar, la evaluación cuantitativa y la evaluación humana como métrica. Se muestra un sistema QA como: una pregunta, una fuente de conocimiento y una respuesta. Entre las métricas usadas se tienen a: Exact Match (EM), F1-score, Precisión, Recall. Para la evaluación cualitativa se considera la evaluación humana. El modelo usado fue DistilBERT y el dataset SQuAD ya usados en otras investigaciones .(Farea et al., 2022)

   Comentario: A pesar de los avances en los sistemas de QA, evaluarlos sigue siendo complejo, las métricas aun no se ajustan al juicio humano sobre todo en tareas y dominios específicos, los dataset deben de contener preguntas y respuestas correctas, con la finalidad de poder medir el rendimiento de los modelos en tareas especificas de forma más precisa, como observamos también en esta investigación la evaluación humana del modelo es de gran importancia sobre todo considerando a especialista sobre un dominio.

Capítulo 2

Marco Teórico

2.1. Procesamiento del Lenguaje Natural (NLP)

Dar una definición exacta de procesamiento del lenguaje natural puede ser una tarea complicada, por los diferentes conceptos que tiene la misma dentro de la inteligencia artificial que en los últimos años creció en gran medida, por ello mostramos algunas definiciones para tener un panorama más amplio.

- El procesamiento del lenguaje natural es el conjunto de métodos que hacen accesible el lenguaje humano a las computadoras. (Eisenstein, 2018)
- El procesamiento del lenguaje natural es un conjunto de técnicas computacionales, con base teórica, para analizar y representar textos naturales en uno o más niveles de análisis lingüístico, con el fin de lograr un procesamiento del lenguaje similar al humano para diversas tareas o aplicaciones. (Liddy, 2001)
- El procesamiento del lenguaje natural es el campo del diseño de métodos y algoritmos que toman como entrada datos en lenguaje natural no estructurados o producen como salida de datos en lenguaje natural. (Goldberg, 2017)

Con el paso de los años la inteligencia artificial generativa y el Procesamiento del Lenguaje Natural mostraron avances transformadores en la tecnología, logrando revolucionar los sistemas de preguntas y respuestas. (Kumar Pandey and Sekhar Roy, 2024).

El NLP cuenta con componentes propios como son : Comprensión del Lenguaje Natural (NLU), Adquisición de conocimiento (KAI) y Generación del lenguaje Natural (NLG) (Raymond Lee, 2025).

2.1.1. NLU

Es el método encargado de la comprensión de significados de los lenguajes hablados por humanos por medio de análisis sintácticos, semánticos y pragmáticos. (Raymond Lee, 2025). Es quien nos permite encontrar similitudes en oraciones o procesar palabras con significados diferentes.

2.1.2. KAI

Es el encargado de la generación de respuestas adecuadas luego que el NLU es plenamente reconocido, es la forma de adquirir conocimiento que no puede ser resuelto por el aprendizaje automático debido a la complejidad del lenguaje natural (Raymond Lee, 2025).

2.1.3. NLG

Es la forma de generar respuestas, réplicas y retroalimentación entre humanos y maquinas, este proceso convierte las respuestas en texto con el fin de sintetizarlas para producir respuestas casi humanas (Raymond Lee, 2025). Este proceso permite a los sistemas de IA generar lenguaje escrito o hablado al igual que los humanos, copiando los estilos y patrones.

2.1.4. Tokenización

El problema al que nos enfrentamos al analizar texto por primera vez es a la división del texto en una secuencia de tokens discretos. (Eisenstein, 2018)

La tokenización es el proceso por el cual se divide un texto en tokens (también llamados palabras en algunos casos) considerando los espacios en blanco y las puntuaciones, este proceso de división puede ser sencillo o complejos dependiendo del idioma con el que estemos trabajando. Los token son el resultado de la tokenización, un token puede estar compuesto por varias palabras, una sola palabra puede estar compuesto por varios tokens y en algunos casos diferentes tokens pueden hacer referencia a la misma palabra (Goldberg, 2017).

Existen diferentes algoritmos para realizar la tokenización, sin embargo la más utilizada por los LLMs es la tokenización ascendente.

Byte-Pair Encoding un algoritmo de tokenización ascendente

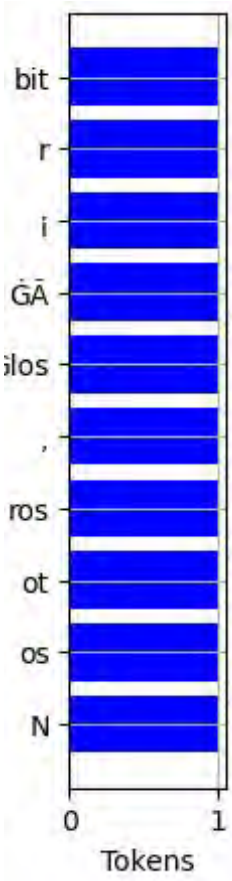
En este proceso no se definen los tokens por palabras o caracteres, sino que usa los datos para de forma automática determinar como deben ser los tokens. Esto es de gran utilidad cuando los modelos se enfrentan a palabras desconocidas, lo cual es común en el procesamiento del lenguaje (Jurafsky and Martin, 2025).

Los algoritmos de NLP suelen aprender datos del corpus con el que son entrenados y luego los utilizan para tomar decisiones sobre corpus de prueba diferentes y su lenguaje. Por tal motivo si el corpus de entrenamiento no contiene palabras del corpus de prueba el modelo no sabrá que hacer con la palabra desconocida. Con el fin de lidiar con este problema los tokenizadores modernos incluyen un conjunto de tokens más pequeños que las palabras que son llamados subpalabras (Jurafsky and Martin, 2025).

Por ejemplo si tokenizamos la siguiente frase: “Nosotros, los árbitros, no solo tenemos que supervisar las partidas para garantizar que se cumplan las Leyes del Ajedrez”(Arbiters,

2024) y usamos el tokenizador de GPT2 y vemos primeros 10 tokens obtendríamos lo siguientes:

Figura 1: Ejemplo de tokenización usando GPT2



Fuente: Elaborado en colab

2.2. Transformers

Antes de la aparición de la arquitectura transformers los modelos se basaban en redes neuronales recurrentes o convolucionales, que incluían un codificador y un decodificador, adicionalmente a esto algunos modelos conectaban el codificador y el decodificador a un mecanismo de atención. (Vaswani et al., 2023) Las redes neuronales recurrentes tienen dificultades al procesar secuencias de texto largos, ademas de presentar con frecuencia problemas de gradientes de desaparición en la capturar las dependencias a largo plazo del lenguaje (Edward Choi, 2017)

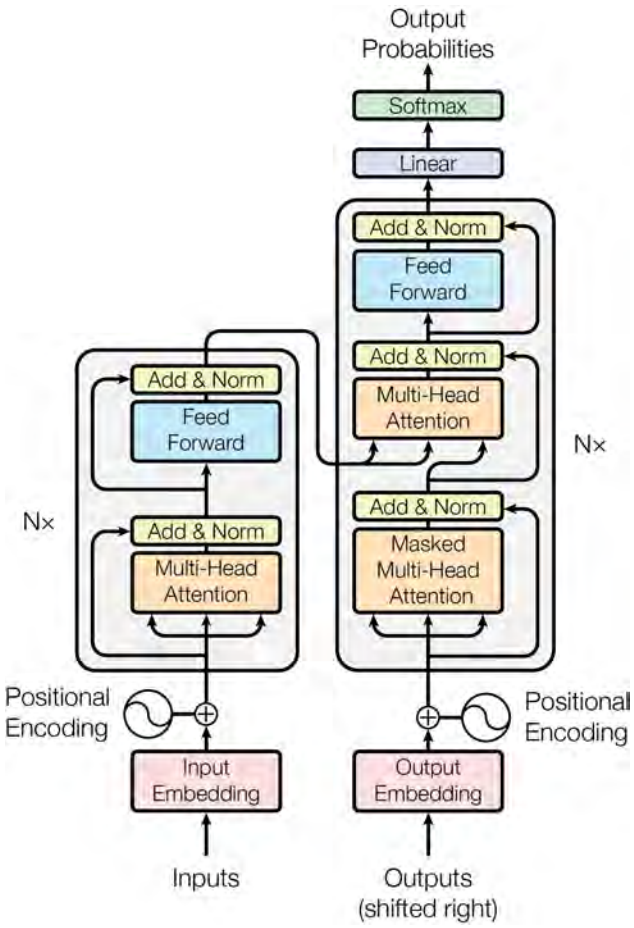
La arquitectura transformers es denominada la arquitectura de los modelos de aprendizaje automático en el NLP. (Xiao and Zhu, 2023) Inicialmente mostraron su excelente rendimiento en tareas de traducción automática y actualmente son indispensables para la construcción de sistemas de aprendizaje supervisado. (Brown et al., 2020)

El gran impacto de los transformers a sido palpados en los últimos años y medida que se hacen mejoras en esta arquitectura su utilidad en la inteligencia artificial es innegable, a diferencia de las redes neuronales recurrentes o convolucionales tradicionales, los transformers solo usan mecanismos de atención y redes neuronales hacia adelante para modelar secuencia de palabras. (Xiao and Zhu, 2023)

La característica del método de atención propio de transformer es que relaciona las posiciones de una secuencia para lograr su representación con lo cual se busca descomponer el problema en subproblemas para resolverlos por separado (Parikh et al., 2016). La atención puede considerarse una forma de representar contextualmente el significado de un token prestando atención e integrando esa información con la de los tokens circundantes, esto ayuda al modelo a comprender la relación de los tokens a lo largo de amplios intervalos (Jurafsky and Martin, 2025).

Esta arquitectura basada en métodos de atención a sido verificada en múltiples tareas (Lin et al., 2017), en tareas de preguntas y respuestas, transformer también muestra su eficacia (Nassiri, 2023)

Figura 2: Modelo de la Arquitectura Transformer.



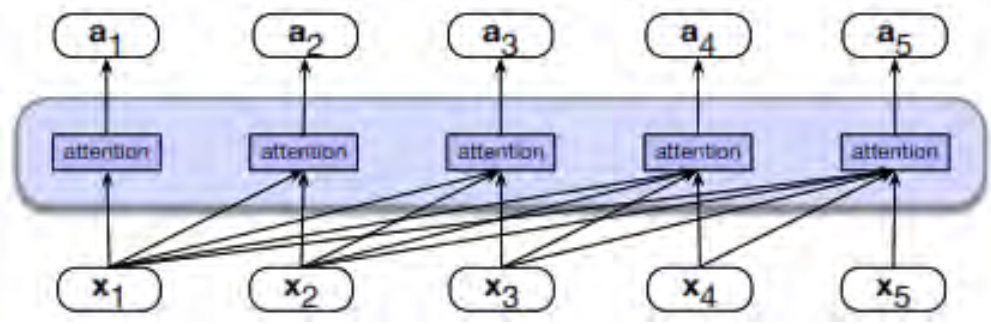
Fuente: Attention Is All You Need 2023

2.2.1. Mecanismo de atención

La atención es el mecanismo del transformer donde se pondera y combina las representaciones de los otros tokens para poder capturar el contexto de la capa *k-1* lo que permite construir la representación de los tokens en la capa k (Jurafsky and Martin, 2025). Lo que quiere decir que la atención es una representación vectorial de un token en una determinada capa del transformer, tomando atención e integrando información de los tokens anteriores en la capa previa.

La atencion tomo como entrada la representación x_i que corresponde al token de entrada en la posición i y una ventana de contexto de entradas previas x_i ... x_{x-1}, para producir la salida a_i (Jurafsky and Martin, 2025).

Figura 3: Capa de auto atención



Fuente: Speech and Language Processing

Esencialmente la atención es la suma pondera de los vectores de contexto, con añadiduras de cálculo de pesos y lo que se suma, se describe que la salida de atención a_i en la posición i del token es la suma ponderada de las representaciones x_j, para todo $j \leqslant i$: usaremos a_{ij} con el fin de indicar cuando x_j contribuye a a_i

$$\textit{versión simplificada: } a_i = \sum_j a_{ij} x_j$$

2.2.2. Encoder:

El codificador está compuesto de 6 capas idénticas, cada capa con dos subcapas que son: Un mecanismo de autoatención multicabezal, y una red de alimentación anticipada simple, conectada posicionalmente (Vaswani et al., 2023).

2.2.3. Decoder:

Al igual que el codificador, el decodificador también se compone de 6 capas idénticas y adicional a las dos subcapas que tiene el codificador en cada capa, el decodificador añade una tercera subcapa para la atención multicabezal sobre la salida del codificador (Vaswani et al., 2023)

2.3. Tareas de Preguntas y Respuestas (QA)

Una de las tareas del NLP es preguntas y respuestas (QA) por sus siglas en ingles (Question and Answers). Los sistemas de QA fueron diseñados para satisfacer las necesidades de información requeridas por las personas.

La tarea de QA consiste en un tipo de recuperacion de información, partiendo de una consulta (escrita o verbal) realizada en lenguaje natural con la finalidad de devolver la respuesta a esa consulta (Martínez Barco et al., 2007)

Los sistemas modernos de QA responde a preguntas utilizando LLMs, la forma más común es usar un modelo preentrenado en dicha tarea y solicitar una respuesta, sin embargo esto trae un gran problema y es que los LLMs aún brindan respuestas erróneas, esto por las alucinaciones, un ejemplo se descubrió que al plantearle a un LLM preguntas de aspecto jurídico los modelos presentaban alucinaciones de hasta más de un 80 % (Dahl et al., 2024).

Muchas veces no es posible determinar cuando el modelo esta alucinando esto por la calibración del mismo, esto genera que los modelos brinden respuestas erroneas con total certeza (Zhou et al., 2024).

Un problema de los sistemas de QA son las preguntas sobre datos confidenciales, y también existen problemas con información cambiante, esto porque los LLMs son entrenados con los datos de ese momento, por ello la forma más común de responder preguntas es por recuperación (Jurafsky and Martin, 2025).

Los sistemas de QA son de gran importancia en la aplicación de actividades cotidianas como son: la interacción con los ordenadores, la automatización, etc (Raymond Lee, 2025)

Existen muchos tipos de QA, siendo el más utilizado el QA extractivo, donde las respuestas a las preguntas son extraídas de un fragmento de texto en un documento ya sea una web, un artículo, un documento legal etc (Lewis et al., 2022).

$$\text{QA} : Q \times K \longrightarrow A$$

donde:

- Q = pregunta en lenguaje natural
- K = fuente de conocimiento (documentos, base de datos, web)
- A = respuesta generada

Es decir se reciben preguntas en lenguaje natural y se devuelven respuestas precisas que son tomadas de una base de datos de conocimiento (Jurafsky and Martin, 2025).

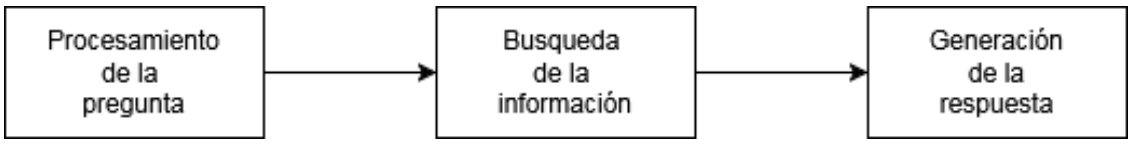
Estos son los 3 pasos que usan la mayoría de los sistemas de QA.

Procesamiento de Preguntas: Analiza la pregunta, identifica su tipo, palabras clave, entidades y estructura gramatical.

Recuperación de Conocimiento: Busca información relevante en corpus o bases de datos.

Generación de Respuesta: Entrega la respuesta en lenguaje natural.

Figura 4: Flujo de un sistema QA



Fuente: Elaboración Propia

2.3.1. Tipos de contextos

En las tareas de QA, el contexto juega un papel importante, pues es el texto en que se basa el modelo para la generación de respuestas. Dependiendo del modelo los contextos pueden mostrar significativas diferencias, esto afecta directamente tanto a la pregunta como a la respuesta.

Contexto Estructural

Son fragmentos de texto que representan definiciones directas y que no necesitan interpretación adicional, se caracteriza por ser directo y proporcionar información precisa, es común en dominios técnicos donde existen reglas bien definidas (Rajpurkar et al., 2016)

Ejemplo: Se dice que un jugador "está en juego" cuando se ha realizado el movimiento de su adversario.

Una característica de estos contextos es que permite a los modelos generar respuestas extractivas, por el hecho que la respuesta se encuentra de forma explícita en el contexto.

Contexto Situacional

Este tipo de contextos ponen escenarios que deben ser interpretados por los modelos para generar respuestas adecuadas. Los modelos deben de ser capaces de aplicar reglas, lo que lleva a realizar un razonamiento adicional (Rajpurkar et al., 2016).

Ejemplo: *"De acuerdo con esta regla, si un jugador no ha dicho "compongo" antes de tocar una pieza y el hecho de tocar la pieza no es accidental, la pieza tocada debe ser movida" (Arbiters, 2024)*

Estos contextos buscan que el modelo no solo genere la respuesta, sino que realice inferencias para aplicar reglas, esto implicar realizar un razonamiento lógico.

Contexto Multievidencia

Este tipo de contextos contienen múltiples reglas o interpretaciones y el modelo debe de ser capaz de determinar cuál es la más relevante para responder la pregunta (Rajpurkar et al., 2016).

Ejemplo: *Un movimiento ilegal se completa una vez que el jugador ha pulsado su reloj. Si se comprueba que se ha completado un movimiento ilegal, se restablecerá la posición*

inmediatamente anterior a la irregularidad. Si eso no es posible, la partida continua a partir de la última posición identificable antes de producirse la irregularidad..

Estos contextos son los más desafiantes para los modelos porque no solo evalúa cada regla, sino que busca cuál de ellas es la más pertinente en la situación descrita.

Contexto conversacional

Este tipo de contextos buscan generar un intercambio de preguntas y respuestas de forma secuencial. El modelo no solo toma en cuenta el contexto actual, sino también los usados previamente (Hermann et al., 2015). Estos contextos son muy usados en los chatbot donde se busca una interacción entre el modelo y el usuario.

2.3.2. Tipos de preguntas

Las preguntas en tareas de QA son demasiado importantes y se clasifican según el tipo de información de los contextos, las preguntas complejas son un desafió en los modelos y son necesarias para ver el rendimiento de los mismos (Lan et al., 2021).

Preguntas factuales

Llamadas también preguntas directas, son aquellas que buscan una respuesta directa basada en texto contenido en el contexto por ejemplo fechas, personas, etc. (Rajpurkar et al., 2016). Se caracteriza por buscar respuestas que no requieren razonamiento adicional y comienza con: qué, cómo, cuándo, dónde, cuál.

Ejemplo: *"¿Cómo se mueve la torre?"*

Estas preguntas permiten evaluar al modelo en la recuperación de hechos exactos de un contexto, lo que es muy importante en tareas de QA

Preguntas condicionales

Este tipo de preguntas depende de una condición o escenario para la extracción de la respuesta. Es muy común que sus contextos involucren situaciones que puedan aplicar una regla bajo ciertas condiciones (Rajpurkar et al., 2016).

Ejemplo: *"Un jugador toca su pieza, ¿está obligado a mover la pieza tocada?"*

La importancia de estas preguntas radica en su capacidad y evaluar al modelo en la aplicación de reglas dadas, para lo cual deben de aplicar cierto grado de razonamiento sobre el contexto.

Preguntas situaciones

Este tipo de preguntas tienen un escenario o historia en el contexto, por ello requiere que el modelo aplique varias reglas para resolver la pregunta o interpretar el escenario (Lan et al., 2021).

Ejemplo: *Si un jugador está en jaque y no tiene ninguna pieza para bloquear el ataque, ¿qué ocurre?*

Este tipo de preguntas son muy importantes en la evaluación de los modelos ya que pide que se apliquen varias reglas o conceptos al mismo tiempo para genera la respuesta, este tipo de preguntas pone a los LLMs en un desafío mucho más complejo.

Preguntas exploratorias

Estas preguntas buscan una explicación sobre un hecho y no solo una respuesta directa. Estas preguntas buscan que el modelo pueda explicar el conocimiento que posee en lugar de solo dar respuestas directas (Lan et al., 2021).

Preguntas de razonamiento

Este tipo de preguntas buscan que el modelo sea capaz de realizar inferencias sobre el contexto utilizando información implícita que no necesariamente este expresada en el texto (Lan et al., 2021).

Ejemplo: *Si un jugador está en una situación de jaque mate, ¿Qué decisiones pude tomar?*

Estas preguntas evalúan en los modelos la capacidad de razonar a partir de información proporcionada por los contextos para luego hacer conclusiones lógicas.

2.3.3. Tipos de respuestas

Los tipos de respuestas se clasifican de acuerdo a la forma en la que se obtienen.

Respuestas extractivas

Como su nombre lo indica la respuesta es extraída dado un contexto y pregunta, el modelo trata de predecir un fragmento del contexto que es una respuesta valida a la pregunta (Martínez Barco et al., 2007)

Su principal ventaja es que limita la cantidad de respuestas correctas, sin embargo si la respuesta no esta en el texto el modelo no sabrá que responder.

Respuestas de opción múltiple

Son las que contienen una cantidad de opciones de respuesta como parte de la pregunta, en este formato las respuestas pueden ser ampliadas por los modelos, lo que puede significar una ventaja, por otro lado la complejidad de este formato es la formulación en si de las respuestas incorrectas o distractores, lo que puede tomar mucho tiempo a los expertos en el tema (Martínez Barco et al., 2007)

También se consideran como tipos de respuestas las de verdadero y falso y las generadas libremente, esta última muy aplicada en los LLM.

2.4. Large Language Model (LLM)

Los LLM son modelos de aprendizaje automático diseñados con la finalidad de aprender datos textuales, comprender patrones lingüísticos para luego procesarlos mediante arquitecturas avanzadas de modelos para producir texto relevante y coherente, traducir idiomas, resumir texto y responder preguntas (Raymond Lee, 2025)

Muchos de los LLM basan su arquitectura en transformer, como es el caso de los modelos BERT y los modelos Generative Pretrained Transformers (GPT) en el dominio del los LLM (Devlin et al., 2018)

Los LLM son modelos de lenguaje preentrenados, en muchos casos los modelos se autoentrenan aprendiendo a predecir la siguiente palabra de un texto considerando las palabras anteriores (Jurafsky and Martin, 2025).

Estos modelos han mostrado un notable rendimiento en tareas de PNL esto gracias al conocimiento que que logran adquirir durante el preentrenamiento, se vio su gran rendimiento en tareas como: resúmenes, traducción automática, respuesta a preguntas o chatbots. Este rendimiento hace que actualmente su estudio y aplicaciones este en gran crecimiento.

Los modelos GPT que fueron entrenados con corpus sin etiquetar, mostraron una mejora del 5,7 % en tareas de preguntas y respuestas (Alec et al., 2018). Esto evidencia que los LLM son más efectivos en diversas tareas gracias al gran corpus de preentrenamiento y a la arquitectura que usan para el mismo.

2.4.1. Características de los Large Language Models

Tamaño de los modelos

Los LLM suelen tener millones de parámetros que son útiles para aprender patrones complejos de los datos durante su entrenamiento, modelos como GPT constan hasta de 175 mil millones de parámetros (Brown et al., 2020). El tamaño es un factor que contribuye en gran media al alto rendimiento de los LLM. Los parámetros de un modelo le permiten capturar aprender más, por ello cuantos más parámetros el modelo aprenderá más (Kaplan et al., 2020), esto puede ser una arma de doble filo por el gran corpus con el que son entrenados los modelos, es decir que muchas veces puede capturar información aun irrelevante o que afecta a ciertas tareas.

Estos parámetros son los componentes que el modelo aprende a través de los datos de entrenamiento, y permiten que el modelo realice tareas específicas como traducción, respuesta a preguntas, generación de texto, etc.

Tabla 2: Comparativa de modelos LLMs: tamaño, organizaciones y fuentes.

Modelo	Parámetros	Organización	Año	Fuente
GPT-3 (davinci)	175B	OpenAI	2020	(Alec et al., 2018)
GPT-2 (xl)	1.5B	OpenAI	2019	(Alec et al., 2019)
LLaMA 2 (7B)	7B	Meta AI	2023	(Touvron et al., 2023)
Mistral 7B Instruct	7B	Mistral AI	2023	(Team, 2023)
Gemma 7B	7B	Google DeepMind	2024	(Team et al., 2024)
BERT (base)	110M	Google	2018	(Devlin et al., 2018)
T5 (base)	220M	Google	2020	(Raffel et al., 2019)

Fuente: Elaboración basada en las fuentes de la tabla

Entrenamiento no supervisado o auto Supervisado

El entrenamiento no supervisado es un tipo de aprendizaje donde el modelo recibe datos sin ser etiquetados, con la finalidad de buscar patrones en los datos de entrada; el entrenamiento autosupervisado, que pertenece a un subconjunto del entrenamiento no supervisado, es capaz de mejorar las capacidades de los LLMs en razonamiento (Jiao et al.,

2024), sin que se pierdan los datos etiquetados.

Cuando se usa un transformer para que un LLM en el entrenamiento autosupervisado, lo que se hace es enseñarle a predecir el siguiente token de una secuencia dados todos los tokens anteriores .(Jurafsky and Martin, 2025)

El objetivo de este entrenamiento es minimizar la diferencia entre la palabra predicha por el modelo y la que realmente sigue en el texto, la función de entriopia cruzada es la que se usa para hallar esa pérdida (Jurafsky and Martin, 2025).

La formula de entriopía cruzada es:

$$L_{\text{CE}}(\hat{y}, y) = -\sum_{w \in V} y[w] \log \hat{y}[w]$$

Donde:

- V es el vocabulario completo,
- $y[w] = 1$, solo para la palabra real que sigue en la secuencia,
- $\hat{y}[w]$ es la probabilidad asignada por el modelo a la palabra w.

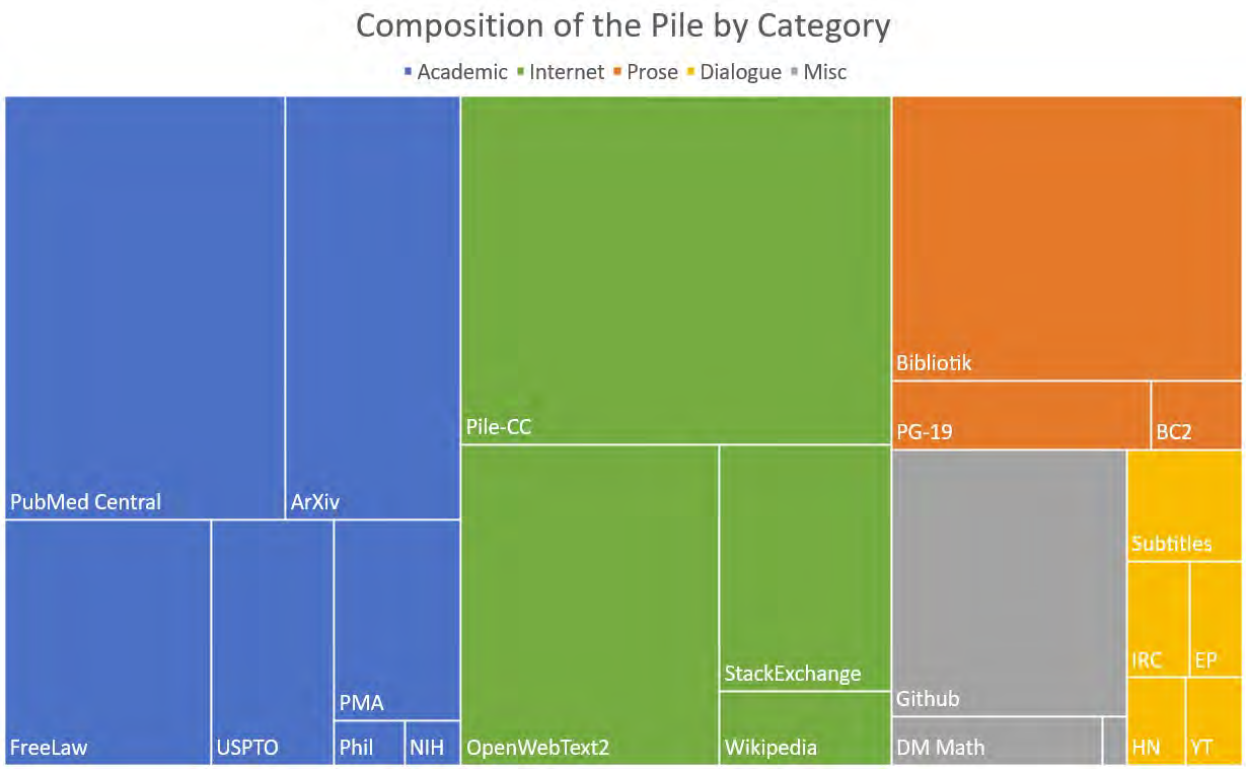
Este método para entrenar LLMs son poderosas ya que permite al modelo aprender de grandes cantidades de datos no etiquetados. El aprendizaje supervisado se enfoca en patrones de los datos y el autosupervisado genera sus propias tareas de predicción.

Corpus de entrenamiento

El corpus con el que son entrenados los LLMs juga un papel vital. En muchos casos los datos son extraidos de internet complementados ademas con un conjunto de datos cuidadosamente seleccionados. Gracias a esto los modelos aprovechan un gran cantidad de información útil en las tareas de NLP, como son: preguntas y respuestas, traducciones de idiomas, resumenes, generación de texto, etx (Jurafsky and Martin, 2025)

The Pile es un gigantesco corpus en ingles de aproximadamente 850 GB, el cual contiene grandes cantidades de texto extraídos de internet, como lo muestra la sigueinte figura.

Figura 5: componentes de Pile



Fuente: The Pile: An 800GB Dataset of Diverse Text for Language Modeling

2.4.2. Ventajas de los LLM

El gran uso de los LLMs en los últimos años se debe principalmente a las ventajas que ofrecen sobre modelos tradicionales, estas ventajas de describen a continuación:

- **Gran capacidad para realizar múltiples tareas:** Esta ventaja de realizar diferentes tareas sin necesidad de un entrenamiento especifico es quizás la principal ventaja de los LLMs. Esto es posible en gran medida a su entrenamiento autosupervisado, después del cual aprenden tareas como: clasificación, generación de texto, preguntas y respuestas, etc, con poca o ninguna adaptación añadida (Brown et al., 2020).

- **Aprendizaje a Escala:** Los enormes corpus con los que son entrenados los LLMs les permiten aprender patrones complejos y capturar dependencias del lenguaje, incluso a largo plazo si fuere necesario, esta capacidad de aprendizaje a gran escala influye en su rendimiento superior en diversas tareas (Kaplan et al., 2020).

- **Adaptabilidad a Diferentes Contextos:** Gracias al fine-tuning y al preentrenamiento con grandes corpus de datos los LLMs son adaptables a diferentes dominios y tareas como análisis de sentimientos y clasificación, esto permite que sean útiles en una variedad de aplicaciones (Devlin et al., 2018).

- **Mejora Continua a Través de Entrenamiento Autosupervisado:** Al poder aprender de datos no etiquetados a través del entrenamiento autosupervisado los LLMs se benefician de datos de dominio general, librándose de los datos etiquetados que suelen ser costosos (Devlin et al., 2018).

- **Flexibilidad en la Generación de Texto:** Modelos como GPT muestran una capacidad increible al momento de generar texto de forma coherente y fluida, por ello se vuelven ideales en tareas de redacción automática, respuestas a preguntas, etc. (Kaplan et al., 2020).

- **Capacidad para Capturar Contexto a Largo Plazo:** Los mecanismos de atención permiten a los LLMs capturar dependencias a largo plazo en un texto, esto es util para aprender el contexto global de un documento. Esto permite al LLM realizar tareas más complejas como respuestas contextualizadas y traducción más precisa (Vaswani et al., 2023).

Los LLM pueden ser usado para diferentes tareas de lenguaje natural sin la necesidad de entrenamiento adicional para cada una. El hecho de ser entrenado con grandes volúmenes de datos hace que los LLM puedan realizar nuevas tareas con un alto rendimiento. Los

modelos pueden ser ajustados (fine-tune) para realizar tareas específicas (Jeong, 2024), esta gran ventaja permite la reutilización de modelos preentrenados sin volverlos a entrenar desde cero, de esta forma se ahorra en fuerza de computo para un nuevo entrenamiento.

2.4.3. Limitaciones y desventajas de los LLM

Quizás la principal limitación que se tiene es el gran costo computacional necesario para el entrenamiento de los LLMs, esto afecta en gran medida a investigadores y empresas de recursos limitados (Johnson and Hyland-Wood, 2024). Los LLM son sensibles a los datos de entrenamiento, dado que los LLMs capturan información del corpus de entrenamiento esto puede ser delicado ya que si nuestro corpus tiene sesgos o desinformación, esto sera captado por el LLM lo que puede traducirse en contenido problemático. Ademas de ello hay otros factores pueden afectar a un LLM como: El impacto ambiental al entrenar grandes modelos, la falta de transparencia y la capacidad limitada de los modelos para manejar contexto a largo plazo. (Bender et al., 2021)

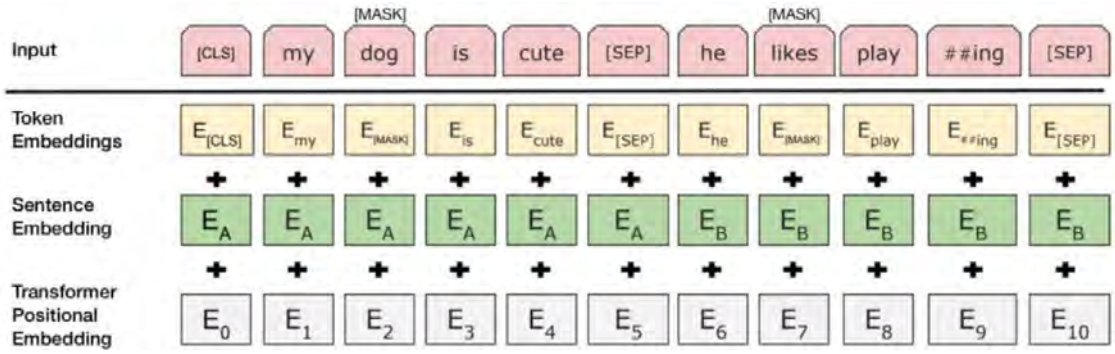
2.4.4. Modelo BERT

BERT es uno de los modelos mas influyentes y usados, fue desarrollado por Google, su arquitectura esta basada en transformers y fue diseñada con el fin de capturar representaciones bidireccionales de lenguaje, esto quiere decir que considera el contexto tanto de izquierda como de derecha, esto hace que tenga una mejorara significativa en el rendimiento en tareas de NLP (Devlin et al., 2018).

El entrenamiento bidireccional de BERT le permite comprender de mejor manera el contexto de una palabra dentro de una oración, esto es porque utiliza un modelo de lenguaje enmascarado en su pre-entrenamiento, lo que le permite predecir palabras faltantes en una secuencia (Devlin et al., 2018).

BERT varia tanto en su modelo BASE como LARGE, el modelo del transformer original añadiendo más capas al codificador y más cabezas de atención lo que le permite capturar mayores detalles lingüísticos, el modelo BASE tiene 12 bloques de transformador mientras que el LARGE 24.

Figura 6: Visulización de la entrada de embeddings en el modelo BERT



Fuente: Introduction to Python and Large Language Models

Parámetros:

BERT cuenta con dos versiones principales que se diferencia en su número de parámetros, capas y embeddings.

Tabla 3: Parámetros del modelo BERT según su tamaño.

Datos	BERT-base	BERT-large
Número de parámetros	110 millones	340 millones
Número de capas (layers)	12	24
Número de cabezas de atención	12	16
Tamaño de los embeddings	768	1024
Tamaño de las capas de atención	768	1024

Fuente: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Entrenamiento

El entrenamiento al que fue sometido el modelo BERT tiene dos fases; la primera fase fue un pre entrenamiento no supervisado en tareas de predicción de palabras y oraciones,

después de esta fase el modelo fue ajustado con datos etiquetados para tareas especificas de NLP (Devlin et al., 2018)

BERT no cuenta con pila de capas de decodificación, por ello no cuenta con una subcapa de atención multicabezal enmascarada. Los programadores de BERT afirman que la capa de atenciíon multicabeza impide el proceso de atención (Rothman, 2022).

Los desarrolladores de BERT idearon la atención bidireccional, con los cual buscaban que una cabeza de atención atienda a todo el texto tanto de izquierda a derecha como de izquierda a derecha (Rothman, 2022).

La flexibilidad de este modelo hace que sea una buena opción en diferentes tipos de tareas de NLP incluyendo preguntas y respuestas.

2.4.5. Modelo T5

El modelo T5 (Text-to-Text Transfer Transformer) es un modelo de LLM basado en la tecnología transformer, desarrollado por Google, que reformula todas las tareas de NLP como problemas de texto a texto (Raffel et al., 2019). Este modelo ha mostrado un rendimiento impresionante en una amplia variedad de tareas como la clasificación de texto, traducción, preguntas y respuestas y resumen de textos, lo que lo convierte en uno de los modelos más versátiles y potentes en el campo de PLN.

Este modelo como su nombre lo indica esta basado en la arquitectura transformers con ajustes para ser un modelo de entrada y salida de texto.

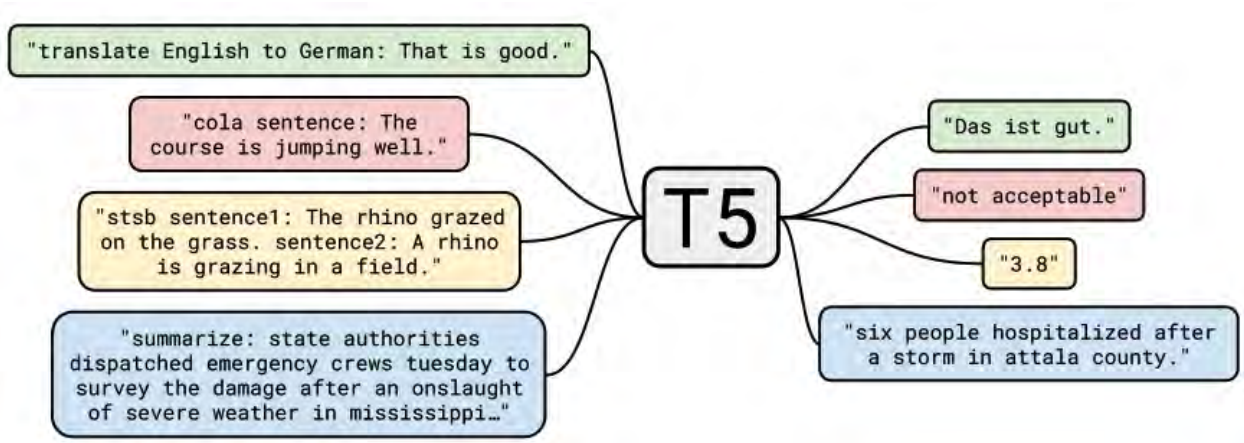
El enfoque text-to-text busca entrenar al modelo de una sola forma para todas las tareas del NLP, ademas este enfoque brinda resultados comparables a las arquitecturas especificas para cada tarea (Raffel et al., 2019)

Al igual que muchos modelos T5 sigue una estrategia de pre-entrenamiento y fine-tuning. El modelo es preentrenado en una enorme cantidad de texto, donde se le proporciona

un texto con algunas palabras faltantes y se le pide al modelo que las prediga. Luego, se ajusta (fine-tuning) en tareas específicas de NLP, lo que le permite generalizar a una amplia gama de tareas (Raffel et al., 2019)

A diferencia del transformer original T5 utiliza embeddings de posición relativa, en lugar de incluir posiciones de entrada arbitrarias. En T5 la codificación posicional se basa en extender la autoatención con la finalidad de comparar las relaciones mediante pares (Rothman, 2022)

Figura 7: Ilustración del modelo T5



Fuente: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Parámetros:

Existen diversas versiones del modelo T5 que varían en tamaño, considerando sus parámetros y la complejidad de las tareas (Raffel et al., 2019), los parámetros principales son:

Tabla 4: Parámetros del modelo T5 según su tamaño.

Dato	T5-small	T5-base	T5-large	T5-11B
Parámetros	60 millones	220 millones	770 millones	11 mil millones
Capas	12	12	24	24
Embeddings	512	768	1024	1024
Capas de atención	512	768	1024	2048
Cabezas de atención	8	12	16	16

Fuente: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Como se observa en la tabla 4 los parámetros muestran como el modelo escala de acuerdo al tamaño a medida que se aumentan el número de parámetros.

Entrenamiento

El entrenamiento del modelo se compone de dos partes; primero un entrenamiento no supervisado con grandes cantidades de texto no etiquetado, para luego realizar un ajuste fino donde el modelo es ajustado para tareas específicas (Raffel et al., 2019).

Es importante resaltar que el entrenamiento de T5 en diferentes tareas se da en simultaneo (Raffel et al., 2019), esto hace muy versatil al modelo en su trabajo en las diferentes tareas del NLP, motivo por el cual este modelo forma parte de los modelos evaluados en el presente proyecto.

2.4.6. Modelo Phi-4

Uno de los modelos más actuales en el mundo de los LLM es Phi-4 que fue desarrollado por Microsoft, este modelo mejora el rendimiento de modelos pequeños mediante la generación de datos sintéticos para tareas pensadas en el razonamiento. Estos datos juegan un papel importante en el postentrenamiento del modelo donde se aplican técnicas de muestreo y preferencias directas, además de que por su calidad están diseñados para mejorar el razonamiento. Phi-4 al igual que muchos modelos basa su arquitectura en transformers solo decodificador con un total de 14 billones de parametros y una longitud de contexto de 4096. Usa un tokenizador tiktokens lo que le permite tener un compatibilidad multilingue (Abdin et al., 2024).

Parámetros

Una detalle de este modelo es el colosal número de parámetros en su entrenamiento además de otros datos como se observa a continuación.

Tabla 5: Parámetros del modelo Phi-4 según su arquitectura.

Datos	Phi-4
Número de parámetros	14 mil millones
Número de capas (layers)	24
Número de cabezas de atención	16
Tamaño de los embeddings	1024
Tamaño de las capas de atención	1024

Fuente: Phi-4 Technical Report

Entrenamiento

Su preentrenamieto fue para aproximadamente 10T tokens, luego del preentrenamiento sigue una etapa intermedia más corta en la que se aumenta la longitud del contexto de 4k a 16k (Abdin et al., 2024).

Phi-4 utiliza una innovación en cuanto a los datos de entrenamiento, una gran cantidad de los datos usados en el entrenamiento son sintéticos lo que lleva a la creación de dataset más diversos y relevantes, en el pos entrenamiento Phi-4 ajusta los resultados y mejora el rendimiento en tareas de razonamiento, asegurando un modelo más preciso y robusto (Abdin et al., 2024).

Por otro lado en su entrenamiento también se contó con datos de la web, ya que solo los datos sintéticos no lograban las mejoras del modelo. (Abdin et al., 2024)

Podemos ver la versatilidad del modelo Phi-4 y la fortaleza que tiene en el razonamiento por ello este modelo forma parte de los modelos a evaluar de la presente investigación.

2.5. Métricas generales de evaluación en los LLM.

Para saber el rendimiento de un LLM es importante someter a evaluación el modelo, para lo cual hacemos uso de diferentes métricas lo que ayuda en gran medida a mostrar si un modelo es eficiente en tareas como generación de texto hasta su precisión en tareas específicas como preguntas y respuestas. A continuación, detallamos las métricas más comunes para

medir el rendimiento de un LLM.

2.5.1. Exact Match (EM)

Exact Match (EM) es una métrica utilizada muy a menudo para ver el rendimiento de los LLM en tareas de respuestas a preguntas (QA). Mide el porcentaje de respuestas generadas por el modelo que coinciden exactamente con la respuesta correcta (Chen et al., 2021).

$$\textbf{Exact Match} = \frac{\text{Número de respuestas correctas}}{\text{Número total de respuestas}}$$

2.5.2. Perplexity (PPL)

Esta métrica ampliamente utilizada para evaluar modelos generativos, mide la incertidumbre del modelo en la predicción de la siguiente palabra en un secuencia dada (Vaswani et al., 2023). A menor perplexidad, mejor es el modelo en la predicción de la siguiente palabra.

$$\textbf{Perplexity} = 2^{H(p)} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i)\right)$$

donde $P(w_i)$ es la probabilidad del modelo para la palabra w_i en la secuencia.

Esta métrica evalúa la fluidez del modelo, por tanto no es posible aplicarla directamente a tareas de preguntas y respuestas.

2.5.3. Bilingual Evaluation Understudy (BLUE)

Esta métrica es principalmente utilizada en la evaluación de modelos aplicados en tareas de traducción automática. Mide la precisión de n-gramas entre la salida del modelo y las referencias humanas (Papineni et al., 2002).

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

donde p_n es la precisión de los n-gramas y BP es el penalizador de longitud.

Esta métrica es usada principalmente en tareas de traducción porque lo que no será aplicada en nuestra investigación.

2.5.4. Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

Esta métrica es ampliamente usada para medir el rendimiento de un LLM en tareas de resúmenes automáticos y generación de texto. Mide la coincidencia de n-gramas entre el texto generado y las referencias de referencia, pero a diferencia de BLEU, ROUGE se enfoca en la recall (recuperación) de los términos relevantes (Lin, 2004)

$$\textbf{ROUGE-N} = \frac{\sum_{n-gram} \text{Recall}}{\sum_{n-gram} \text{Reference Recall}}$$

2.6. Métricas de evaluación en tareas de Preguntas y Respuestas.

Las métricas para medir un LLM en tareas de preguntas y respuestas implica problemas de correspondencia difusa, lo que dificulta la aplicación de las métricas generales (Hu and Zhou, 2024).

Las tareas de preguntas y respuestas buscan que el LLM identifique la respuesta a una pregunta especifica dado un contexto, la respuesta a dicha pregunta suele ubicarse en una sección del contexto, por lo tanto las tareas de preguntas y respuestas se transforman en predecir la posición inicial y final de la sección contextual, asumiendo que la parte central

es la respuesta (Hu and Zhou, 2024)

Existen muchas formas de realizar las tareas de preguntas y respuestas, siendo la más común la QA extractiva, esta forma de trabajar con preguntas y respuestas consiste en que la respuesta a una pregunta puede ser identificada en un fragmento de texto de un documento (Lewis et al., 2022). Este forma de realizar tareas de QA es el que será usado para nuestro proyecto de investigación.

A continuación se mencionan las métricas usadas en tareas de preguntas y respuestas en el presente proyecto.

2.6.1. Strict Accuracy (SaCC)

También es conocida como Exact Match Accuracy, esta metrica se utiliza para evaluar la precisión de un modelo considerando si la respuesta generada coincide exactamente con la respuesta correcta. Está métrica considera una respuesta como correcta si y solo si ambas respuestas (la generada por el modelo y la que respuesta que ya se conoce de antemano) son completamente idénticas sin ningún tipo de variación en la redacción (Tsatsaronis et al., 2015).

La fórmula de Strict Accuracy es:

$$\textbf{SaCC} = \frac{\text{Número de respuestas exactas correctas}}{\text{Número total de respuestas}}$$

Donde:

- **Respuestas exactas correctas**: Son aquellas respuestas generadas que coinciden exactamente con la respuesta de referencia.
- **Número total de respuestas**: Es el número total de respuestas generadas que estamos evaluando.

Es por esta razón que esta métrica fue elegida para nuestro caso de estudio ya que requerimos respuestas exactas, esto casi siempre ocurre en tareas de preguntas y respuestas.

2.6.2. Lenient Accuracy (LaCC)

A diferencia de SaCC, Lenient Accuracy permite realizar evaluar un modelo de preguntas y respuestas con mayor flexibilidad. LaCC no exige una coincidencia exacta de respuestas, sino que considera que una respuesta generada como correcta si es equivalente semánticamente a la respuesta correcta, esto puede incluir variaciones de redacción, uso de sinónimos u orden alterado de las palabras (Tsatsaronis et al., 2015).

La fórmula de Lenient Accuracy es:

$$\textbf{LaCC} = \frac{\text{Número de respuestas correctas (permitiendo pequeñas variaciones)}}{\text{Número total de respuestas}}$$

Donde:

- **Respuestas correctas (permitiendo pequeñas variaciones)**: Respuestas que son correctas en términos de contenido, aunque puedan tener variaciones como sinónimos o un orden diferente de las palabras.
- **Número total de respuestas**: Es el número total de respuestas generadas que estamos evaluando.

En aspectos técnicos es importante la interpretación de las respuestas generadas, ya que puede haber múltiples formas de responder a una misma pregunta, por ese motivo se eligió está métrica para nuestro caso de estudio.

2.6.3. Similitud semántica

Cuando se habla de similitud semántica se refiere a la medición de similitud de significados entre dos textos, sin considerar las palabras empleadas en ellos, una de las formas de realizar esto es utilizando la similitud de cosenos entre embeddings de las respuestas generadas y de las correctas (Cer et al., 2017).

Similitud Coseno $= \frac{A \cdot B}{\|A\|\|B\|}$ Donde:

- A y B son los vectores de las respuestas generadas y correctas, respectivamente.
- $A \cdot B$ es el producto punto de los dos vectores.
- $\|A\|$ y $\|B\|$ son las normas (o longitudes) de los vectores.

2.6.4. Evaluación Humana

La evaluación humana generalmente es el estándar de oro en la evaluación de LLMs en tareas de preguntas y respuestas (Alammar and Grootndorst, 2024). Este proceso de evaluación es fundamental en tareas de QA, en donde las respuestas correctas no siempre son estrictamente definidas en el contexto, sino que pueden variar dependiendo, del estilo de respuesta y de las interpretaciones posibles.

La evaluación humana es importante para evaluar la calidad contextual y semántica en las respuestas obtenidas por el modelo, ya que las métricas como Exact Match no son suficientes para capturar la complejidad de la lengua (Rajpurkar et al., 2016), Esto resulta importante cuando evaluamos respuestas interpretativas como manuales o reglamentos, como se plantea en nuestro caso de estudio.

Aspectos clave de la evaluación humana:

- **Coherencia semántica:** La respuesta es coherente con la pregunta y con el contexto proporcionado.

- **Precisión y relevancia:** La respuesta es precisa y relevante para la pregunta. No debe contener información errónea ni irrelevante.

- **Fluidez:** La respuesta debe ser gramaticalmente correcta y estar bien estructurada.

- **Diversidad y creatividad:** En tareas más abiertas (como generación de texto o diálogo), las respuestas pueden variar, y los evaluadores consideran la creatividad en la generación de respuestas.

Como ya se puso en evidencia la evaluación humana es importante en tareas de preguntas y respuestas para evaluar la coherencia, precisión y fluidez (Rosset, 2020) de las respuestas generadas por un modelo, eso nos dará mas garantía de nuestros resultados.

2.7. Herramientas y librerías utilizadas

2.7.1. Herramientas utilizadas

- **Hugging Face:** Hugging Face[1] es una plataforma abierta que permite a los investigadores, desarrolladores y empresas acceder a recursos como modelos preentrenados y dataset (Jain, 2022). En el presente proyecto se usará esta plataforma para acceder a los modelos pre entrenados y para los procesos de tokenización de datos.

- **Google Colab:** Fue desarrollada por Google [2] y permite la ejecución de notebooks de Jupyter en la nube (Bisong, 2019). Permite la utilizando GPUs y TPUs de manera gratuita, ideal para el análisis de LLMs, es ampliamente utilizada en investigador de inteligencia artificial por su accesibilidad y facilidad de uso.

[1]https://huggingface.co/
[2]https://colab.research.google.com/

2.7.2. Librerías utilizadas

- **AutoTokenizer:** Esta clase genérica permite cargar el tokenizador de cualquier modelo, para la conversión de texto a tokens.
- **AutoModelForQuestionAnswering:** Es la clase genérica del modelo que se encarga del trabajo con preguntas y respuestas de forma extractiva.
- **Json:** Json[3] es parte de las librerías incluidas en Python, nos permite leer archivos que luego pueden ser procesados en conjunto.
- **Pandas:** Pandas[4] es una librería de código abierto muy útil para la manipulación de datos.
- **Pipeline:** Pipeline es una clase propia de transformers que nos permite ejecutar tareas de PLN, nos permite cargar y aplicar LLMs de forma más sencilla.
- **SequenceMatcher:** SequenceMatcher[5] es una clase de la librería difflib que permite comparar similitud de secuencias de texto. Esta clase usa el algoritmo de distancia de Levenshtein lo que nos permite calcular que tan parecidas son dos cadenas de texto.
- **SentenceTransformer** SentenceTransformer[6] es una librería que nos permite calcular semántica entre las respuestas generadas por el modelo y las respuestas del dataset, esto nos permite saber si las respuestas tienen el mismo significado aunque no tenga coincidencia palabra por palabra.
- **Torch:** Torch[7] es un paquete de python muy usando en computación numérica que permite la aceleración de GPU.

[3]https://docs.python.org/3/library/json.html
[4]https://pandas.pydata.org/
[5]https://docs.python.org/3/library/difflib.html
[6]https://huggingface.co/sentence-transformers?
[7]https://pypi.org/project/torch/

- **Transformers:** La librería transformers de Hugging Face, es una de las herramientas mas utilizadas en el trabajo con modelos pre entrenados, nos permite realizar diferentes tareas de NLP utilizando modelos de ultima generación (Hugging-Face, 2025).

Capítulo 3

Metodología de la investigación

3.1. Diseño Metodológico

En esta sección se describe los pasos metodológicos realizados en el presente proyecto, así como el tipo de investigación y el enfoque del mismo, además de los detalles usados en el presente trabajo.

3.1.1. Diseño de la investigación

El diseño es experimental, pues se manipulan los modelos observando sus comportamiento (Hernández Sampieri et al., 2014), además se tiene variables controladas que son de gran utilidad para esta investigación como el dataset de preguntas y respuestas con sus respectivos contextos.

3.1.2. Enfoque de la investigación

La investigación tiene un enfoque cuantitativo dado que se centra en la medición de resultados específicos sobre el desempeño de los LLMs propuestos, esto permite la recopilación de datos numéricos que posibilitan un analisis estadístico (Hernández Sampieri et al.,

2014). En el presente estudio se hizo uso de las métricas: Strict Accuracy, Lennient Accuracy, Similitud Semántica, junto con la evaluación humana, buscando el enfoque adecuado de evaluación en tareas de preguntas y respuestas.

3.1.3. Población

El no haber un conjunto de datos con preguntas y respuestas sobre el manual de ajedrez, se observa la necesidad de crear uno que sea suficiente para poder realizar la evaluación de los modelos, en total sea elabora un conjunto de 119 datos compuestos de un contexto, pregunta y respuesta asociada como se observa en SQuAD: 100,000+ Questions for Machine Comprehension of Text (Rajpurkar et al., 2016).

3.1.4. Muestra

La población cuenta con un tamaño aceptable por los modelos, por ello en las métricas automáticas se hizo uso de la población entera, mientras que en la evaluación humana se hace uso de un conjunto de 20 preguntas directas, 20 condicionales y 20 de razonamiento, esta decisión busca tener un análisis completo ademas de evitar fatiga en los encargados de realizar la evaluación humana.

3.1.5. Muestreo por conveniencia

También es conocido como muestreo deliberado, porque no cuenta con ningún procedimiento estandarizado o razón más que la comodidad de seleccionar la muestra (Supo Condori, 2020). Esto aplicamos en la selección las preguntas a evaluar por los árbitros.

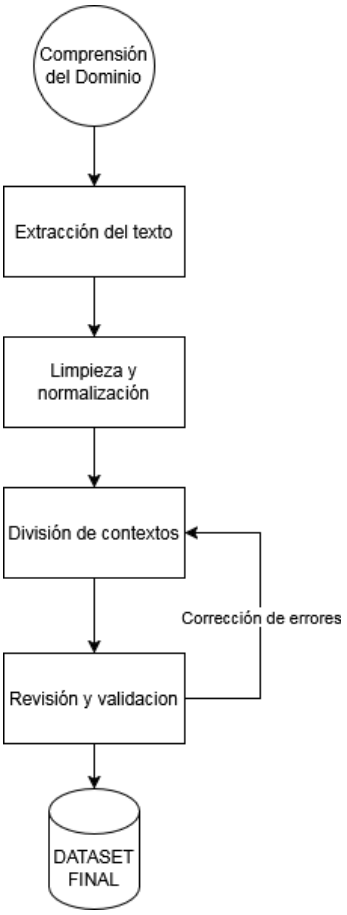
3.2. Etapas del proceso metodológico

Se desarrolla un método dividido en seis etapas principales que se detallan a continuación:

3.2.1. Extracción de los datos

Al no contar con un dataset de preguntas y respuestas basadas en el manual de ajedrez se procede a construir nuestro propio conjunto de datos. Esta construcción se realizó siguiendo los principios de la metodología Knowledge Discovery in Databases (KDD), propuesta por Fayyad (Fayyad et al., 1996), la cual establece un proceso sistemático para transformar datos en conocimiento, la aplicación de esta metodología se muestra en la siguiente figura.

Figura 8: Flujo para la extracción de Datos



Fuente: Elaboración propia

1. Comprensión del dominio: Se usa como única fuente el manual deportivo de ajedrez identificando los artículos pertinentes a las competiciones de ajedrez estándar, rápido y relámpago.

2. Extracción del texto: Se extrae el contenido textual mediante la herramienta PyMuPDF aplicada a python, conservando la estructura del texto en orden de capítulos y artículos, de esta forma nos aseguramos que el texto este completo para su posterior segmentación.

3. Limpieza de datos: En este paso eliminamos toda información innecesaria como encabezados, números de página, saltos de linea dobles, caracteres especiales y elementos que puedan afectar la calidad de los contextos. Con ello buscamos que los datos estén en las mejores condiciones de legibilidad y consistencia para ser procesados por los LLMs.

4. Creación de contextos, las preguntas y respuestas para las evaluaciones: Una vez limpiado el texto, se procede a realizar la división del mismo en fragmentos que tengan coherencia semántica, estos fragmentos pueden corresponder a un artículo o una sección temática. A partir de cada fragmento se elaboran preguntas clasificadas en tres categorías según su complejidad: directa, condicional y de razonamiento. A cada pregunta se le asigna una respuesta de referencia basada unicamente en el manual deportivo de ajedrez, con ello se busca garantizar precisión de respuesta.

5. Revisión y validación. Los contextos son revisados manualmente verificando su consistencia, así mismo las preguntas son verificadas de forma correcta para evitar confusiones de formulación, de esta forma se busca minimizar errores y lograr una buena evaluación por parte de los modelos de manera que esta sea confiable y replicable.

6. Interpretación de la data: Se presenta y almacenan los datos en un formato entendible

para su procesamiento, en nuestro proyectos se usaron diccionarios python.

De esta forma nos aseguramos que nuestro conjunto de datos este en optimas condiciones para poder trabajar con los modelos, ademas de que esta metodología permite realizar ajustes de forma rápida y eficas ahorrando así tiempo y recursos (Fayyad et al., 1996)

3.2.2. Elección de los modelos

En la elección de los modelos se tuvo en cuenta los factores mencionados por (Schur and Groenjes, 2023):

1. Accesibilidad: Para garantizar los objetivos se considera el uso de LLMS open-source, esto nos permite el uso de los LLMs con mínimas adaptaciones y restricciones.

2. Tamaño del modelo: Para esto se tuvo que considerar las limitaciones computacionales que nos brinda COLAB en su versión gratuita de GPU T4 15 GB, este tamaño es suficiente para el modelo más grande usado en el proyecto, Phi4.

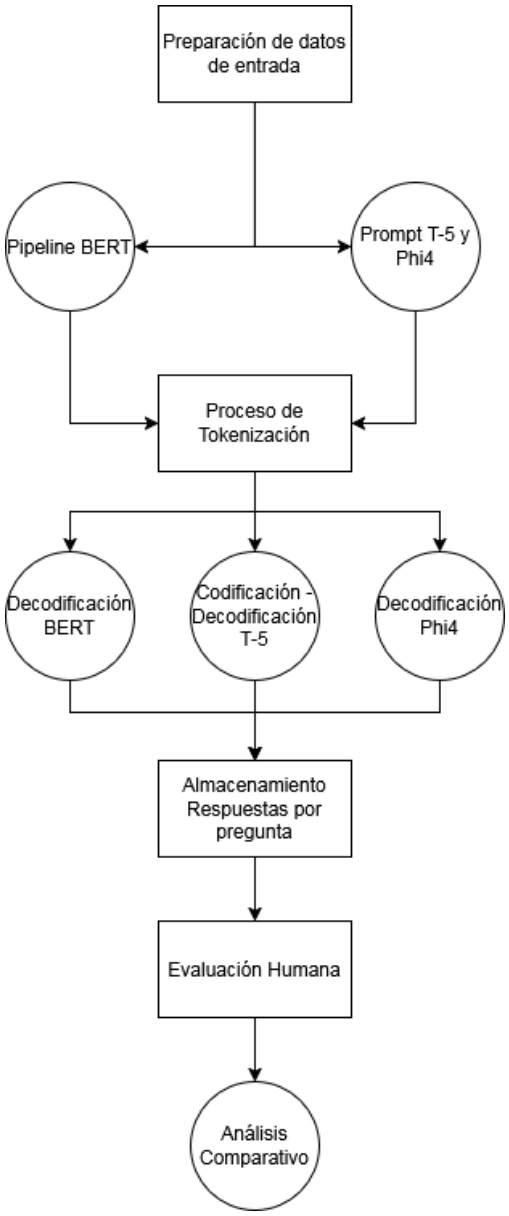
3. Entrenamiento: Se considera que los modelos puedan desempañarse en idioma español o que haya sido entrenados en ese idioma o en si defecto sean multilingues.

Además se considera que un modelo por arquitectura (Vaswani et al., 2023) un encoder, un decoder-encoder y un solo decoder, esto nos permite ver las fortalezas de cada modelo antes de su uso aplicativo.

3.2.3. Flujo de trabajo para los modelos

La Figura 9 ilustra de forma visual el flujo de trabajo seguido por los modelos en el trabajo de investigación.

Figura 9: Flujo Metodológico Modelo BERT



Fuente: Elaboración propia

Los modelos tuvieron el siguiente flujo según el caso y su arquitectura:

1. Preparación de los datos de entrada: En este paso se realiza el enlace entre los datos preparados y los modelos de modo que puedan ser usados por los mismos de forma eficiente.

2. Configuración del modelo: En este paso se ajusta la configuración del modelo para tareas de preguntas y respuestas ya sea por pipeline o por promt.

3. Conversión del texto en tokens: Este proceso es independiente a cada modelo y depende directamente del entrenamiento y sus parámetros.

4. Análisis de pregunta y contexto: Este paso es realizado de forma interna por cada modelo según su arquitectura, tokenización y entrenamiento.

5. Generación con la respuesta: El modelo genera su respuesta considerando el contexto asociado a la pregunta. La generación varia de acuerdo al modelo esto puede llevar a realizar ajustes en los contextos si fuera necesario para mejorar los resultados.

6. Almacenamiento de resultados: Los resultados son almacenados en un archivo csv para la posterior evaluación humana.

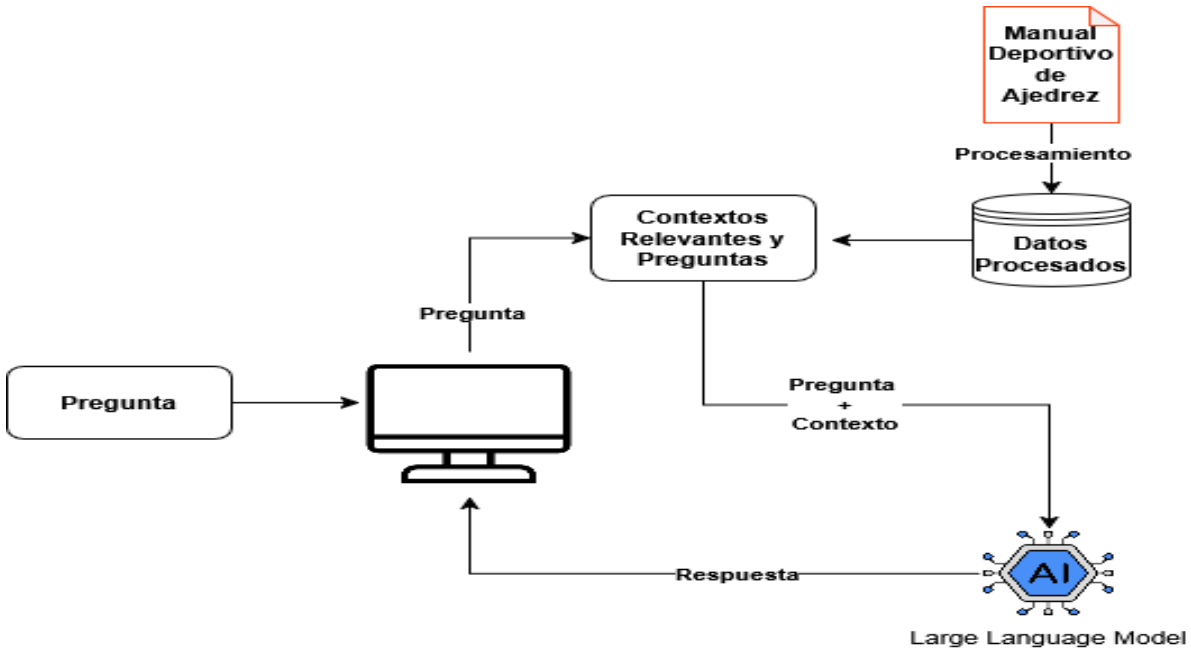
7. Evaluación Humana: Se evaluan 20 preguntas de cada tipo por modelo a fin de tener una solida evaluación por parte de los árbitros. En esta fase se recurre a tres arbitros debidamente acreditados y con diferentes rango arbitrales, siendo: un Arbitro Internacional, un Arbitro Fide y un Arbitro Nacional, todos ellos debidamente acreditados por la Federación Internacional de Ajedrez.

8. Análisis de respuestas: Se evalua el rendimiento de los modelos segun la respuesta dada por el mismo en concordancia con las respuestas de referencia del conjunto de datos, se ponen los resultados de acuerdo a cada modelo y considerando las métricas aplicadas. Strict Accuracy mide la coincidencia exacta en las respuestas, Lenient Accuracy permite variaciones de palabras dentro de las respuestas.

Capítulo 4

Desarrollo del tema de tesis

Figura 10: Flujo del desarrollo de tesis



Fuente: Elaboración propia

La Figura 8, nos muestra el flujo en que se basa el desarrollo de la presente investigación, al basarnos completamente en las leyes deportivas del ajedrez, debemos procesar completamente esos datos, eliminando todo ruido innecesario que pueda afectar los contextos y por ende el rendimiento del LLM, las preguntas deben de estar alineadas a los contextos extraídos del manual al igual que las respuestas, considerando ello la pregunta es procesada

antes de ser pasada al LLM. Finalmente tanto la pregunta como el contexto son pasadas al modelo, el cual genera una respuesta de retorno.

4.1. Creación del conjunto de datos

Para realizar la evaluación de los LLMs es necesario procesar los datos en este caso el manual deportivo de ajedrez para poder crear un dataset que sea consistente, libre de ruido y con información adecuada. No se encontró ningún repositorio de preguntas y respuestas sobre el manual de ajedrez, motivo por el cual se diseña y construye un dataset propio de preguntas y respuestas para la investigación.

El manual de ajedrez es un documento en formato PDF y en idioma español, por lo que el tratamiento de la información se realiza tomando todas esas consideraciones.

Para definir la cantidad de campos del dataset nos basamos en el dataset SQAC [1] (Spanish Question Answering Corpus), que cuenta con los siguientes campos relevantes: title, context, question y answers.

4.1.1. Análisis del documento

La fuente de datos principal y única para nuestra investigación es el Manual de Árbitros, traducida por la FDEA en su versión más reciente que corresponde al año 2024, y que es el manual oficial para todas las competencias de ajedrez a nivel mundial.

El documento está compuesto por 12 capítulos cada uno dividido de acuerdo a su contenido en artículos, directrices, apéndices, anexos y glosarios.

Al analizar el documento se observa que existen varios problemas para la extracción del texto coherente, a continuación se detallan los mismo.

[1] https://huggingface.co/datasets/PlanTL-GOB-ES/SQAC

- Las secciones están entre lazadas, esto ocasiona grandes posibilidades de corte de información o de incoherencia contextual por la estructura misma del texto.
- Cortes de artículos del manual por medio de sub-artículos, esta situación se presenta en varias ocasiones, lo que puede ocasionar información inconsistente en la generación de contextos.
- Contenido de imágenes, la extracción de las imágenes ocasiona que la descripción de las mismas quede aun luego de extraer solo texto, por ello se tiene que hacer una revisión de cada texto referido a una imagen con el fin de no quedarnos con información que dependan de imágenes.

Por otro lado al analizar el documento se tiene 3 aspectos claves del mismo que son:

- **Operativos:** Describe aspecto que tienen que ver con el actuar de los árbitros al momento del juego, esto genera una gran dependencia del contexto y tiene una naturaleza secuencial.
- **Normativos:** Describe las reglas formales de la competición, tienen una naturaleza determinista y suelen estar basadas en condiciones.
- **Éticos – Disciplinarios:** Describe las regulaciones éticas, sanciones y apelaciones, tienen una naturales interpretativa, y se basa netamente en las leyes del manual.

Luego de analizar el texto desde un punto de vista técnico nos damos cuenta de la complejidad que conlleva realizar una estructuración de este tipo de documentos y lo que implica en el LLM poder responder diversos tipos de preguntas basados en ellos, por ello se realiza una separación estructural del manual indicando la complejidad y el aspecto que tiene el documento, lo cual será de gran ayuda al momento de interactuar con el LLM dado que la complejidad del texto afectará directamente las respuestas dadas por el modelo..

La Tabla 6 muestra los bloques hallados dentro del manual, el aspecto al que pertenecen y la complejidad de procesamiento para el LLM, para corroborar la complejidad se recurrió a la ayuda de un árbitro internacional, gracias a quien se pudo determinar que aspectos del manual son más complejos asignando los valores de baja, media, alta y muy alta según correspondiese.

Tabla 6: Clasificación del manual según su contenido

Bloque	**Aspecto**	**Complejidad**
Rol del árbitro	Operativo	Media
Reglas básicas	Normativo	Baja
Reglas de competición	Normativo y Operativo	Media
Apéndices	Normativo especializado	Media
Anti-trampas	Disciplinario	Muy alta
Código ético (Parte II y III)	Ético	Alta
Código disciplinario (Parte IV)	Disciplinario (legal)	Muy alta

Fuente: Elaboración propia

4.1.2. Extracción del texto

El documento usado en esta investigación consta de 484 páginas(Arbiters, 2024), que abarcan reglas, procedimientos, códigos éticos, medidas disciplinarias y anexos. Debido al contenido tan diverso del manual el texto es considerablemente largo estimándose en varios cientos de miles de palabras, esto excede desde luego la capacidad que tienen los LLMs para hacer el procesamiento en una sola consulta, por ello es necesario procesar todo el texto limpiando toda información irrelevante o que pueda traer confusión a los modelos.

Luego de extraer todo el texto en bruto para su procesamiento, se aprecio mucho ruido debido a los índices, pies de página, imágenes e introducciones que no aportaban información relevante para nuestro conjunto de datos, añadido a ello también el texto extraído quedo con varios espacios en blancos.

En este proceso se hizo uso de la librería **PyMuPDF**[2], que mediante el modulo fitz

[2]https://pymupdf.readthedocs.io/en/latest/

nos permite abrir documentos PDF para su procesamiento, esto será de gran utilidad para el tratado del manual deportivo de ajedrez, el objetivo es obtener un diccionario python con cada una de las páginas del manual, para poder manipularlas luego manualmente para no perder información relevante dentro de los contextos.

```
#Funcion para extraccion de texto
def extract_pdf_text(pdf_path: str) -> dict:

pdf_file = Path(pdf_path)

if not pdf_file.exists():
raise FileNotFoundError(f"PDF no encontrado")

document = fitz.open(pdf_path)
pages = []

for page_index in range(len(document)):
page = document.load_page(page_index)
raw_text = page.get_text("text")
cleaned_text = clean_text(raw_text)

pages.append(
{
  "page_number": page_index + 1,
  "text": cleaned_text,
  "char_count": len(cleaned_text),
  "word_count": len(cleaned_text.split())
}
)

full_text = "\n\n".join(page["text"] for page in pages if page["text"])

#Diccionario Final con datos necesarios
result = {
  "document_name": pdf_file.name,
  "total_pages": len(document),
  "total_characters": len(full_text),
  "total_words": len(full_text.split()),
  "pages": pages,
}

document.close()
return result
```

Código 4.1: Extracción de texto

El texto extraído fue exportado en formato json, donde se observó ciertas palabras que no aportan ningún tipo de significado ni relevancia para los contextos que se piensa elaborar, al contrario esta información puede ser perjudicial al momento de buscar las respuestas por parte de los modelos.

Las palabras menos relevantes para nuestro estudio se muestran en la Tabla 7:

Tabla 7: Palabras irrelevantes en el manual

Palabra	Cantidad de apariciones
Apéndice	12
capítulo	31
capítulos	8
artículo	240
artículos	65
Directrices	8
imagen	4
figura	1
pág	490
página	9
tabla	39

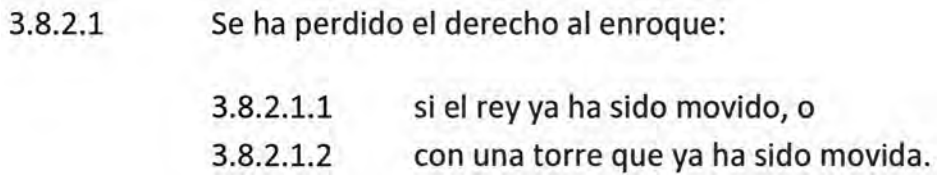
Fuente: Elaboración propia

Todas estas palabras no deben de formar parte de los contextos porque pueden llevar a los modelos a cometer errores sobre todo en la tokenización, además otro defecto que se observa es la numeración de las páginas, los capítulos y artículos por lo que se deberá de hacer una limpieza también de esas cadenas.

4.1.3. Limpieza de datos

La primera limpieza que se efectúa sobre nuestro texto es la eliminación de información irrelevante como son: pies de página, cabeceras, sangrías, numeraciones y títulos confusos y palabras que no aportan información relevante al contexto. El manual deportivo de ajedrez tiene una estructura rígida como se aprecia en la Figura 9, esto puede complicar la creación de nuestros contextos, motivo por el cual se realiza la limpieza.

Figura 11: Estructura de un artículo del manual



Fuente: Manual deportivo de ajedrez

La Figura 9 muestra las deficiencias con la numeración de los artículo, este exceso de indentaciones numerales nos genera gran conflicto al procesar los datos, ademas el tokenizador de los modelos tomara los números de forma individual, por lo que se genera un exceso de información que no sería útil para nuestros contextos, por ello esa información irrelevante también será extraída.

Para esta primera limpieza se hizo uso de expresiones regulares mostradas en la Tabla 8, aunque estas expresiones regulares no son efectivas para todos los casos, logran limpiar el texto en gran medida. Las expresiones regulares usadas fueron las siguientes:

Tabla 8: Expresiones regulares usadas en la extracción del texto

Corrección	**Expresión Regular**
Capítulos	r'\bCapítulo\b'
capitulo	r'\bcapitulo\b'
Appendices	r'\bapendices\b'
Directrices	r'\bDirectrices\b'
imagen	r'\bimagen\b'
figura	r'\bfigura\b'
pág	r'\bpág\b'
página	r'\bpágina\b'
tabla	r'\btabla\b'
numeros	r'\b(\d+\.)+\d+\b'

Fuente: Elaboración propia

Al termino de esta limpieza quedamos con un texto libre de palabras irrelevantes ademas de eliminar todos los espacios, este texto es nuestro texto bruto que servirá para la creación de los contextos.

Antes de la creación de contextos se procedió a ver las palabras más relevantes del manual con el fin de poder buscar contextos en los cuales se encuentren esas palabras, recordemos que el manual de ajedrez está en internet por ello los modelos ya toman parte de esa información en su pre-entrenamiento. En la Tabla 9 se observa una lista de las palabras más frecuentes dentro del manual, esta información es importante a la hora de realizar las preguntas al modelo dado que mientras más veces aparece una palabra en un contexto el modelo podrá brindar mejores respuestas.

Tabla 9: Palabras Frecuentes

Palabra	**Frecuencia**
jugador	1540
fide	1290
jugadores	1093
árbitro	718
emparejamiento	658
puede	581
ajedrez	549
torneo	518
partida	419
árbitros	459

Fuente: Elaboración propia

Las palabras más usadas en el manual muestran coherencia esperada, donde las palabras jugador y árbitro obtienen puntuaciones bastante elevadas, lo que nos da entender que esas palabras serán claves al momento de empezar a generar nuestros contextos.

4.1.4. Generación de contextos

No existe una forma especifica de como elaborar los contextos para poder aplicarlos en tareas de preguntas y respuestas, por ello en la elaboración de los contextos se aplicaron criterios inspirados en dataset QA consolidados como son: SQuAD y QuAC, de donde se rescato la forma en que se elaboraron los contextos. Para nuestra investigación no se definieron los contextos de forma arbitraria, si no que tomó en cuenta unidades textualmente coherentes del manual de modo que cada uno de ellos tuviera la información necesaria para responder al menos una pregunta.

Por el gran volumen de texto previamente procesado del manual se procede a dividirlo en fragmentos de 150 palabras, esta generación ocasionó que nuestros contextos tengan cierta inconsistencia.

Para este fin usamos la siguiente función:

```
def dividir_en_fragmentos(texto: str, palabras_por_fragmento: int = 150,
  max_fragmentos: int = 10) -> list:
# Extraer palabras
fragmentos = []
for i in range(0, len(palabras), palabras_por_fragmento):
if len(fragmentos) >= max_fragmentos:
break
fragmento = " ".join(palabras[i:i + palabras_por_fragmento])
fragmentos.append(fragmento)
return fragmentos
```

Código 4.2: Division de texto

Observemos el siguiente fragmento generado, sin manipularlo

- **Contexto 1:**

  "jugadores no sean molestados supervisar el desarrollo de la competición d observara las partidas especialmente cuando los jugadores estén apurados tiempo y hará cumplir las decisiones tomadas e imponer sanciones a los jugadores cuando corresponda para hacer todo esto los árbitros deberán tener la competencia necesaria buen juicio y objetividad absoluta prefacio de las leyes del ajedrez el número de árbitros requeridos en una competición varía dependiendo del tipo de evento individual equipo en el sistema de las partidas round robin sistema suizo eliminatorias encuentros del número de participantes y en la importancia del evento normalmente se designa un árbitro" (Arbiters, 2024)

Análisis

Como se observa la segmentación del texto trajo algunos problemas y es que dos ideas se están mezclando, además se perdieron signos de puntuación y conectores, también se aprecia que hay letras al aire lo que sugiere que se trataba de varios puntos que pueden referirse a las obligaciones de un árbitro y a la cantidad de árbitros, esto definitivamente traerá problemas a los modelos en la generación de respuestas. En consecuencia podemos afirmar que dividir el texto no es suficiente por ello los contextos se ajustan de forma manual

para mayor consistencia con el fin de obtener preguntas claras que permitan a los modelos mejorar la calidad de sus respuestas, entonces con las correcciones aplicadas en anterior contexto quedaría de la siguiente manera:

"Los árbitros deberán procurar que los jugadores no sean molestados y supervisar el desarrollo de la competición. Observarán las partidas, especialmente cuando los jugadores estén apurados de tiempo, y harán cumplir las decisiones tomadas e impondrán sanciones a los jugadores cuando corresponda. Para cumplir estas funciones, los árbitros deberán tener la competencia necesaria, buen juicio y absoluta objetividad." (Arbiters, 2024)

Analizando el contexto generado podemos ver que el contexto tiene una mejor calidad de información a ser recuperada, esto se logra gracias a los ajustes realizados para ordenar todo el texto.

4.1.5. Clasificación y adaptación de los contextos

Como se explicó anteriormente el manual presentó un grado alto de complejidad a lo hora de realizar la extracción de información para nuestros contextos, el manual deportivo no se encontraba organizado de forma que sea directamente utilizable en tareas de QA, por ello se realizo la adaptación de los contextos antes de la elaboración de las preguntas y respuestas.

Adaptación de contextos

Este proceso consistió en transformar los fragmentos originales del manual en texto coherente sin alterar su interpretación. Se buscó que cada contexto tenga la información necesaria para la formulación de al menos una pregunta que pueda ser respondida de forma clara, en base a esto se tomó cierto criterios para una mejor adaptación de los contextos.

1. **Autosuficiencia:** El contexto debe tener en si mismo la información necesaria para

responder la pregunta, sin requerir consultar otras fuentes o contextos para ello.

2. **Ajustarnos al reglamento:** La adaptación no buscó el contenido del manual deportivo, solo lo reorganizó cuando fue necesario todos los ajustes se hicieron conservando el mismo significado.

Clasificación de contextos:

Los contextos fueron clasificado de acuerdo a la información que contienen y a la forma en que es posible obtener la respuesta.

1. **Contexto directo:** Estos contextos permiten obtener la respuesta de manera directa y casi textual a partir del texto proporcionado, estos contextos provienen de conceptos claros del manual como por ejemplo la definición de jaque, las puntuaciones, etc. Para estos contextos se tomó como referencia el dataset SQuAD (Yuan, 2018)

2. **Contexto situacional:** Permite la aplicación de una norma en una condición dada. En este caso la respuesta no depende solo de la recuperación sino de una interpretación, ejemplos claros de este tipo de contextos pueden incluir información sobre qué ocurre si un jugador pulsa el reloj sin haber movido, qué sucede si se detecta una promoción incompleta, etc. La situación o condición debe de estar explicita en el contexto (Sun et al., 2021).

3. **Contexto multievidencia:** Los contextos multievidencia requieren unir dos o más reglas en el mismo fragmento adaptado, la respuesta en este tipo de contextos no solo se apoya en una oración sino en varias (Yang et al., 2018). Podemos ver este tipo de contextos cuando una regla esta compuesta por una condición y una excepción, esta situación es vista muy a menudo en el manual por ejemplo al diferenciar entre acuerdo de tablas restringido por las bases del torneo y reclamaciones de tablas que

siguen siendo válidas. Como se puede evidenciar este tipo de contexto son los más complicados de ajustar por todas las consideraciones que se deben tener en cuenta para buscar generar preguntas coherentes para obtener respuestas adecuadas.

La Tabla 10 muestra un ejemplo de cada contexto con todos los ajuste ya realizados y la pregunta que puede ser formulada a partir del mismo

Tabla 10: Ejemplos de tipos de contexto utilizados en el dataset

Tipo de contexto	**Contexto**	**Pregunta**
Directo	La partida de ajedrez se juega entre dos oponentes que mueven sus piezas de forma alterna sobre un tablero cuadrado, llamado tablero de ajedrez. Donde el jugador que lleva las piezas claras realiza el primer movimiento.	¿Qué jugador realiza el primer movimiento en una partida de ajedrez?
Situacional	Si un jugador mueve un peón a la fila más lejana y pulsa el reloj sin sustituir el peón por una nueva pieza, el movimiento es ilegal. En ese caso, el peón se sustituye por una dama del mismo color.	¿Qué ocurre si un jugador promociona un peón de forma incompleta y pulsa el reloj?
Multievidencia	Las bases del torneo pueden prohibir a los jugadores ofrecer o aceptar tablas antes de un número concreto de movimientos o sin consentimiento del árbitro. Sin embargo, las reclamaciones de tablas por triple repetición, regla de 50 movimientos o los casos automáticos de tablas siguen siendo aplicables durante toda la partida.	¿Puede un torneo restringir los acuerdos de tablas y qué excepciones siguen vigentes?

Fuente: Elaboración propia

4.1.6. Creación de preguntas

La creación de las preguntas es otro desafío al que se debe enfrentar cuando realiza un dataset QA, este paso es imprensindible porque estas preguntas serán usadas para la evaluación de los modelos en referencia con su contexto dado. Para este paso aplicamos

en primer lugar un forma de extracción automática para las preguntas. En este proceso se elaboró una función en python para encontrar patrones que ayuden a la generación de preguntas dado un determinado contexto como se muestra a continuación.

```python
def generar_preguntas_reglas(texto: str) -> list:
preguntas = []
patrones = [
(r"(.*?) debe (.*?)[\.\n]", "Que debe {}?"),
(r"(.*?) aconsejable (.*?)[\.\n]", "Que es aconsejable {}?"),
(r"es responsable de (.*?)[\.\n]", "De que es responsable {}?"),
(r"Esta prohibido (?)[\.\n]", "Que esta prohibido?")
]

for patron, plantilla in patrones:
for m in re.findall(patron, texto, flags=re.IGNORECASE):
sujeto = m[0].strip()
preguntas.append(plantilla.format(sujeto))
return preguntas
```

Código 4.3: Generación de preguntas

Esta forma de obtener preguntas de forma automática es bastante común con grandes volúmenes de texto, al pasar el contexto en bruto tuvimos las siguiente preguntas generadas:

Figura 12: Generación de preguntas con texto bruto

```
1 lista_preguntas = generar_preguntas_reglas(texto)
2 print(lista_preguntas)

[]
```

Fuente: Elaboración propia

En la Figura 10 se observa que no se generó ninguna pregunta, evidenciando lo poco efectivo que es la función para encontrar preguntas, sin embargo al estructurar el contexto de forma más adecuada generamos las siguientes preguntas:

Figura 13: Generación de preguntas con texto estructurado

```
1 lista_preguntas = generar_preguntas_reglas(texto)
2 print(lista_preguntas)

['¿Qué debe a. Un árbitro?', '¿Qué es aconsejable Para la primera ronda del  torneo, es?']
```

Fuente: Elaboración propia

Como se puede apreciar en la Figura 11 se generan dos preguntas sin embargo una

de ellas carece de sentido, sin embargo la segunda tiene una coherencia muy alta, por otro lado del contexto usado se pueden generar otro tipo de preguntas, asociado a esto debemos aclarar que para este tipo de generación es por demás necesario tener una estructura muy coherente del contexto esto implica: separaciones, anidaciones, puntos seguidos, separados y si hubiera enumeraciones deben de estar lo más claras posibles. Por este motivo se procede a elaborar las preguntas de forma manual para ello se cuenta con personas expertas en el área.

Elaboración manual de las preguntas

La presente investigación basó la generación de las preguntas considerando el dataset SQuAD para las preguntas directas (Yuan, 2018), el dataset ConditionalQA para las preguntas condiciones (Sun et al., 2021) y el dataset HotpotQA para las preguntas de razonamiento (Yang et al., 2018), estos estudios sugieren controlar el contexto, la formulación de la pregunta y la respuesta de acuerdo al objetivo buscado, para nuestro caso ver el rendimiento de los LLMs, por ello las preguntas no fueron generadas de forma general respecto al ajedrez sino como preguntas asociadas a un contexto en concreto.

Este proceso consistió en leer los contextos, identificar información significativa y formular preguntas que pudieran responderse directamente desde el contenido, los métodos de elaboración manual de preguntas son valiosos cuando se se busca preservar la relevancia y exactitud del dominio (Guelma, 2024).

4.1.7. Creación de respuestas

Para esta fase se consideró unicamente como fuente de respuestas al manual deportivo de ajedrez, no se admitió ninguna ayuda externa para que los modelos puedan responder de forma correcta.

Finalizado la creación de los contextos, las preguntas y respuestas, se obtiene un conjunto de datos con los detalles que se aprecian en la Tabla 11:

Tabla 11: Descripción del conjunto de datos

Descripción	Cantidad	Artículos relacionados
Contextos	119	
Preguntas directas	36	1, 2 y 3
Preguntas condicionales	23	4, 5, 6, 8 y 11
Preguntas de razonamiento	60	7, 9, 10, 12
Respuestas	119	

Fuente: Elaboración propia

Finalmente todos los datos procesados son guardados en forma de diccionarios python para hacer más fácil la manipulación de los mismos, además de que se sabe que los diccionarios son muy usados en ciencia de datos.

4.1.8. Verificación del dataset

Formato del dataset

En esta fase adaptamos nuestros datos al formato de datos SQuAD[3], y usamos diccionarios python como formato para su almacenamiento. La Tabla 12 muestra la descripción de las claves en cada diccionario que posteriormente son usadas tanto en el modelo como en las métricas.

Tabla 12: Descripción del conjunto de datos

Clave	Descripción
title	Titulo elegido de acuerdo a la descripción del manual
context	Contexto elaborada a partir del contexto
question	Pregunta creado a partir del manual
gold_answers	Respuesta extraída a partir del manual
question_type	Tipo de pregunta realizada

Fuente: Elaboración propia

[3]https://huggingface.co/datasets/rajpurkar/squad

A continuación se muestra un ejemplo en código con los datos que contiene cada diccionario.

```
{
  "title": "Articulo 7",
  "context": "Si durante una partida se comprueba que algunas piezas han
   sido desplazadas de sus casillas, debera restablecerse la posicion
  anterior a la irregularidad. Si no puede identificarse dicha posicion,
  la partida continuara a partir de la ultima posicion identificable
  previa a la irregularidad. Es recomendable que la investigacion se
  lleve a cabo por los dos jugadores bajo la supervision de un arbitro.",
  "question": "Si se comprueba que algunas piezas han sido desplazadas
  de sus casillas que se debera hacer?",
  "gold_answer": "Restablecerse la posicion anterior a la irregularidad.
  ",
  "question_type": "condicional"
},
```

Código 4.4: Estructura del conjunto de datos basado en diccionario

La estructura de diccionarios es muy útil cuando se quiere mantener una estructura de datos a los que se desea acceder de forma independiente, gracias a las claves se puede acceder de forma individual a cada elemento que forma el conjunto de datos.

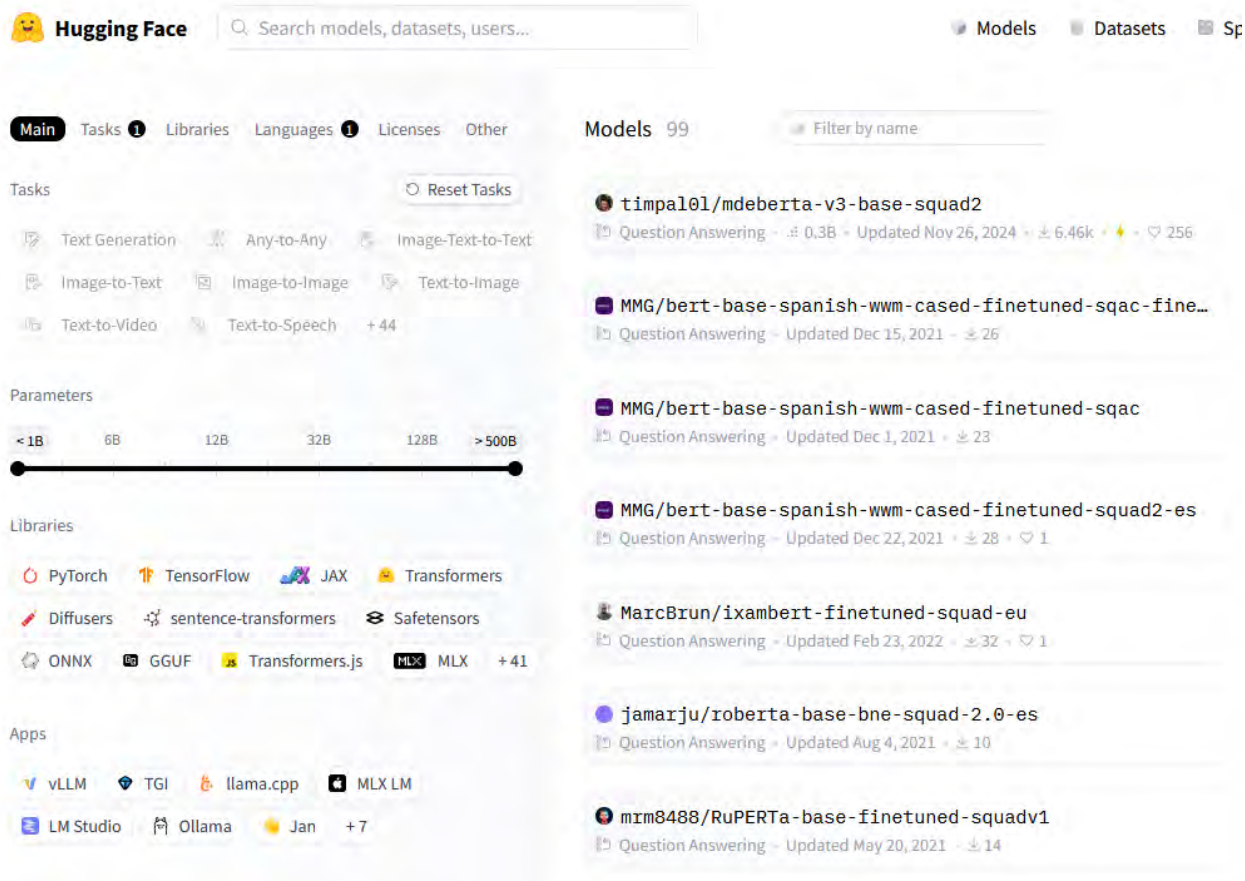
Verificación manual

Finalmente, luego de generados todos nuestros contextos, preguntas y respuestas, se procedió a realizar una verificación manual de todo el conjunto de datos con la finalidad de encontrar incongruencias en cualquiera de los campos.

4.2. Selección de los modelos

Para esta fase hacemos uso de la plataforma Hugging Face que cuenta con una gran cantidad de modelos de código abierto, 13212 de modelos pre entrenados en tareas de QA en total al momento de realizar este proyecto, sin embargo, solo 99 modelos en idioma español muchos de los cuales son modelos antiguos.

Figura 14: Modelos QA en la plataforma HuggingFace en español



Fuente: Plataforma HuggingFace

Como se observa en la plataforma de HuggingFace existen pocos modelos que manejen el lenguaje español, por ello se recurrió a modelos multilingüe con la finalidad de poder solucionar el problema del idioma.

Luego de filtrar los modelos que estuvieran pre-entrenados en tareas de QA se consideró los modelos que se muestran en la Tabla 13:

Tabla 13: Modelos escogidos

Nombre modelo	Descripcion
mrm8488/bert-base-spanish-wwm-cased-finetuned-spa-squad2-es	Basado en Transformers, pre-entrenado en español
google/flan-t5-small	Basado en transformers, encoder decoder y multilingue
microsoft/Phi-4-mini-instruct	Mejorada en tareas de razonamiento, multilingue

Fuente: Elaboración propia

Debemos añadir que los dos primeros modelos elegidos son de tamaño pequeño, por

las limitaciones que enfrentamos en nuestro entorno de ejecución y solo un modelo grande (Phi-4). Otro objetivo en la elección de estos modelos está en la comparación del rendimiento de modelos generativos antiguos, como el caso de BERT, y recientes como Phi-4, por otro lado, se podrá comparar modelos de distintos tamaños y ver si eso afecta el rendimiento de los mismos.

4.2.1. Flujo de trabajo

Para poder realizar una evaluación adecuada de los modelos elegidos, se utilizó la GPU de Google Compute Engine en su version actualizada de python de Google Colab. Está GPU con 15GB de RAM ha sido suficiente para la evaluación de los modelos elegidos, la elección de Google Colab se debe también a la facilidad para el uso de múltiples herramientas para la evaluación de los modelos.

Librerías cargadas

La carga de librerías es el primer paso en la evaluación de los modelos, dos de las librerías indispensables para la carga de los modelos son ModelTokenizer y ModelForQuestionAnswering, las otras librerías son complementos que nos permiten realizar diversas acciones que ya se explicaron anteriormente.

```
import pandas as pd
from rapidfuzz import fuzz
import unicodedata
import re
import torch
from transformers import BertForQuestionAnswering, BertTokenizer
```

Código 4.5: Librerías cargadas

Estas librerías nos permiten realizar todo el flujo del proceso durante las evaluaciones de los modelos, es necesario recordar que los nombres de algunos componentes pueden variar de acuerdo al modelo.

Carga del modelo y tokenizador

En este proceso se define el nombre del modelo pre-entrenado y tokenizador que vamos a usar, cabe mencionar que el tokenizador a usar debe de ser el mismo que usa el modelo, para ilustrar este proceso usaremos como ejemplo el modelo BERT.

```
model_name = "mrm8488/bert-base-spanish-wwm-cased-finetuned-spa-squad2-es"
model = BertForQuestionAnswering.from_pretrained(model_name)
tokenizer = BertTokenizer.from_pretrained(model_name)
```

Código 4.6: Garga del tokenizador y modelo

Este código es la forma básica de carga para nuestros modelos, es importante mencionar que durante la evaluación se puede añadir otros argumentos a las funciones llamadas con la finalidad de que el modelo mejore su rendimiento.

Evaluación de modelos

A continuación, detallamos los pasos para la evaluación de los modelos, la carga de nuestros datos y la obtención de nuestros resultados.

- Como Primer paso debemos cargar nuestros modelos considerando los pasos explicados previamente.
- Seguidamente creamos las funciones de evaluación según las métricas propuestas en el presente estudio.

```
    def strict_accuracy(pred: str, gold: str) -> int:
    pred_normalized = normalize_text(pred)
    gold_normalized = normalize_text(gold)
    return 1 if pred_normalized == gold_normalized else 0

```

Código 4.7: Función para la métrica strict Accuracy

Para crear la funcion Lenient Accuracy consideramos como valor mínimo aceptable 60 %, por ello nuestra función tiene la siguiente estructura.

```
    def lenient_accuracy(pred: str, gold: str, threshold: int = 60) -> int:
    pred_normalized = normalize_text(pred)
```

```
    gold_normalized = normalize_text(gold)

    score = fuzz.ratio(pred_normalized, gold_normalized)
    if score >= threshold:
    return score
    return 0

```

Código 4.8: Función para la métrica Lenient Accuracy

```
    def semantic_similarity(gold_answer, generated_answer):
    gold_answer = str(gold_answer)
    generated_answer = str(generated_answer)

    embeddings = semantic_model.encode(
    [gold_answer, generated_answer],
    convert_to_tensor=True
    )

    score = util.cos_sim(embeddings[0], embeddings[1]).item()
    return score

```

Código 4.9: Función para la métrica Similitud Semántica

- Creamos una función que permita la extracción de la respuesta del modelo, para la cual se debe de pasar el contexto y las preguntas procesadas.

```

    def get_model_answer(model, tokenizer, question, context):
    inputs = tokenizer(question, context, return_tensors="pt")
    with torch.no_grad():
    outputs = model(**inputs)
    start_position = outputs.start_logits.argmax()
    end_position = outputs.end_logits.argmax()
    answer_tokens = inputs.input_ids[0][start_position:end_position
    +1]
    answer = tokenizer.decode(answer_tokens)

    return answer

```

Código 4.10: Extracción de respuestas

- Finalmente creamos un método para realizar la evaluación de las respuestas generadas por los modelos.

```
    def evaluate_item(model, tokenizer, item):
    question = item['question']
    context = item['context']
    gold_answer = item['gold_answer']

    # Obtener la respuesta del modelo
```

```
7     generated_answer = get_model_answer(model, tokenizer, question,
    context)
8
9     #Evaluar las respuestas
10    strict = strict_accuracy(generated_answer, gold_answer)
11    lenient = lenient_accuracy(generated_answer, gold_answer)
12
13    # Almacenar los resultados
14    return {
15      #"title": item,
16      "question": question,
17      "generated_answer": generated_answer,
18      "gold_answer": gold_answer,
19      "strict_accuracy": strict,
20      "lenient_accuracy": lenient
21    }
22
```

Código 4.11: Evaluación de las respuestas

Este es el flujo de trabajo que seguiremos para cada uno de los modelos, nuevamente es necesario aclarar que las funciones pueden sufrir variaciones de acuerdo al modelo usado, con la finalidad de mejorar los resultados.

Capítulo 5

Resultados y Discusión

5.1. Resultados

En esta sección se muestran los resultados obtenidos luego de la evaluación de nuestros modelos considerando las respuestas a los 3 tipos de preguntas planteadas dentro de nuestro dataset. El principal objetivo dentro de los experimentos es evaluar la comprensión de los modelos en contexto del manual deportivo de ajedrez, además de someter las respuestas a evaluación humana contando para ello con un arbitro internacional de ajedrez, un árbitro FIDE y un árbitro Nacional. Este experimento permite la evaluación de los modelos frente a distinto tipos de preguntas que pueden formularse en un contexto cerrado, además de la complejidad de los mismos dado el tipo de preguntas. En cada experimento nos centraremos en la exactitud del modelo, para lo cual haremos uso de las funciones de Strict Accuracy, Lenient Accuracy y Similitud Semántica. Para la similitud semántica en todos los casos usaremos un umbral de 0.75 para determinar si las respuestas son equivalentes (Cann, 2025)

5.1.1. Especificaciones técnicas

Se usa el entorno de ejecución de google COLAB en su versión gratuita para el lenguaje python 3.12 disponible al momento de realizar esta investigación.

Figura 15: Especificaciones del entrono de ejecución

```
Wed Dec 17 14:18:40 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |                    0 |
| N/A   36C    P8              9W /   70W |       0MiB /  15360MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

Fuente: Elaboración propia

5.1.2. Experimentos

Los experimentos fueron separados para cada modelo considerando los tipos de preguntas a las que fueron sometidos, esto permite ajustar algunos contextos y preguntas, si fuera necesario, para encontrar mejores respuestas. Se considera que para cada modelo se usa el conjunto de datos descrito anteriormente donde se tiene 36 preguntas directas, 23 condicionales y 60 de razonamiento. Las métricas usadas en cada modelo son: Strict Accuracy, Lenient Accuracy y Similitud Semántica, adicional a estas métricas se efectúa la evaluación humana.

Primer experimento: Evaluación del modelo bert-base-spanish

El primer modelo en ser evaluado es el modelo bert-base-spanish, luego de realizar las configuraciones necesarias en el modelo, los resultados arrojados por el modelo para cada

tipo de pregunta fueron los siguientes:

Tabla 14: Resultados de la evaluación del modelo bert-base-spanish

Tipo de pregunta	Total	Strict Acc.	Lenient Acc.	Similitud Semántica
Condicional	23	30.43 %	56.52 %	65.22 %
Directa	36	30.56 %	77.78 %	75.00 %
Razonamiento	60	13.33 %	50.00 %	53.33 %

Fuente: Elaboración propia

Como se observa en la Tabla 14, existe un ascenso de porcentajes a medida que la métrica es más flexible. El trabajo de contextos da resultados aceptables según las métricas sin embargo se observo un comportamiento interesante por parte del modelo que es siguiente:

Contexto: *Las ocho hileras verticales del tablero se denominan columnas. Las ocho hileras horizontales se denominan filas. Una sucesión de casillas del mismo color en línea recta que va de un borde al adyacente se denomina diagonal*

Pregunta: *¿Cómo se denominan las hileras verticales?*

Respuesta del modelo: *filas*

Según el contexto y el conjunto de datos la respuesta es: Columnas, sin embargo se observa que el modelo no se queda con esa respuesta sino que evalúa la otra palabra hileras en su segunda aparición como la respuesta, esto muestra que el modelo puede tener fallas cuando una palabra esta múltiples veces en el contexto, lo que ocaciona perdida de información relevante o falta de razonamiento lo que cual se evidencia en los resultados obtenidos por el modelo.

Segundo experimento: Evaluación del modelo google-flan-t5-small

Este experimento tuvo un grado mayor en cuanto a la configuración del modelo para ajustarlo a tareas de QA, este modelo tiene una capacidad multilingue lo que lo hace ideal para poder trabajar con distintos idiomas. Los resultados obtenidos por el modelo google/flan-t5-small se detallan en la Tabla 15.

Tabla 15: Resultados de la evaluación del modelo google/flan-t5-small

Tipo de pregunta	Total	Strict Acc.	Lenient Acc.	Similitud Semántica
Condicional	23	4.35 %	26.09 %	62.29 %
Directa	36	0.00 %	11.11 %	53.79 %
Razonamiento	60	0.00 %	6.67 %	48.39 %

Fuente: Elaboración propia

Este modelo tiene unos resultados por debajo del modelo BERT en este experimento tanto en preguntas directas como en las de razonamiento, dos detalles observados en este modelo es la gran influencia que tienen los contextos pudiendo ser estos tomados completamente en algunas respuestas y el uso de idioma ingles por partes en algunas respuestas esto último básicamente por ser el idioma principal de su entrenamiento, este modelo fue el que si respondió correctamente la pregunta:

Pregunta: *¿Cómo se denominan las hileras verticales?*

Respuesta del modelo: *Columnas*

Esto muestra que el modelo si puede comprender contextos donde existen múltiples palabras iguales como en el caso de hileras, esto es un plus muy importante sobre todo al trabajar con preguntas condicionales, otro punto a mencionar sobre este modelo que en gran medida requiere contextos cortos, en donde se pudo ver su mejor desempeño.

Tercer experimento: Evaluación del modelo microsoft-Phi-4-mini-instruct

La evaluación de este modelo fue el que más recursos uso de la plataforma COLAB, sin embargo las herramientas brindadas por colab fueron suficientes para el propósito del presente trabajo. La capacidad multilingue de este modelo es superior a la mostrada por los anteriores modelos esto gracias a su gran corpus de entrenamiento. Los resultados obtenidos por este modelo se muestran en la Tabla 16.

Tabla 16: Resultados de la evaluación del modelo microsoft-Phi-4-mini-instruct

Tipo de pregunta	Total	Strict Acc.	Lenient Acc.	Similitud Semántica
Condicional	23	17.39 %	63.91 %	76.44 %
Directa	36	16.67 %	65.60 %	73.76 %
Razonamiento	60	13.33 %	68.56 %	78.84 %

Fuente: Elaboración propia

Este modelo fue el que mejores resultados obtuvo en todo aspecto, esto manifiesta que efectivamente este modelo tiene una mejor capacidad multilingue que el modelo T5 y una mejor capacidad de razonamiento respecto a los otros dos.

Sin embargo el modelo produjo una alucinación en nuestra pregunta cable realizada en los otros modelos donde su respuesta fue:

Pregunta: *¿Cómo se denominan las hileras verticales?*

Respuesta del modelo: *Ocho*

Esto nos muestra que a pesar del gran éxito que tuvo este modelo con las respuestas, es necesario que la data este en optimas condiciones así como las preguntas realizadas.

Al revisar todo el bloque de preguntas realizadas por los modelos se observa que todos muestran valores bajos en la métrica Strict Accuracy, si bien esa métrica está directamente ligada con las respuestas de nuestro conjunto de datos, existe gran posibilidad de que muchas respuestas estén bien dadas, por este motivo se realiza un cuarto experimento aún mayor que es la evaluación humana.

Cuarto Experimento: Evaluación Humana

En el cuarto experimento se evaluó las respuestas de los LLMs con árbitros oficiales de ajedrez que puedan indicar si las respuestas de los modelos son correctas o no.

Para esta evaluación se contó con la participación de Arizabal Triveno Nathaniel, Arbitro Internacional de Ajedrez[1], Obregon Castellanos, Alexander, Arbitro FIDE[2], y Quispe

[1]https://ratings.fide.com/profile/3820599
[2]https://ratings.fide.com/profile/3831140

Puma Luis Fernando, Arbitro Nacional[3], todos avalados por la federación internacional de ajedrez.

A diferencia de los otros experimentos, en esta ocasión solo se pasó a los árbitros la pregunta (del dataset) y la respuesta generada por el modelo, además se tomó en consideración 20 preguntas de cada tipo. Para este experimento usamos el criterio de puntuaciones donde cada arbitro deberá asignar un valor entre 1 y 5 a cada de respuesta del modelo, donde 1 significa que la respuesta no satisface la pregunta y 5 que la respuesta es perfecta; este criterio fue usado en el artículo Evaluating Question Answering Evaluation(Chen et al., 2019).

En este experimento se observa un gran inconveniente y es que los árbitros pueden tener diferentes puntos de vista para la calificación asignada a las respuestas, por ello se tomo en cuenta la siguientes recomendaciones:

- Los árbitros deben de aplicar sus conocimientos basados solamente en el manual, sin aplicar ningún criterio externo o de experiencia.
- No se debe inferir o suponer nada, la respuesta debe de ser tomada tal cual es, si es necesario alguna comprobación se sugiere al árbitro poder hacer uso del manual de ajedrez para tomar una mejor decisión en el puntaje asignado a la respuesta.
- Finalmente se le pide al árbitro que si fuera posible añadieran una respuesta básica a la pregunta que ellos consideren, con el fin de poder ajustar aun más tanto contextos como preguntas.

A continuación se muestran los puntajes que los árbitros asignaron a los modelos, ademas cada bloque de preguntas fueron enviados de forma separada, para mantener el orden en este experimento.

[3]https://ratings.fide.com/profile/3815722

Evaluación humana del modelo mrm8488/bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

El orden de evaluación fue el siguiente:

- Primero las preguntas directas que fueron un total de 20
- Segundo las preguntas condicionales que fueron un total de 20
- Tercero las preguntas de razonamiento que fueron un total de 20

Evaluación de preguntas directas

Los promedios asignados por los arbitros a los modelos se muestran en la Tabla 17.

Tabla 17: Promedio de calificaciones obtenidas por tipo de pregunta

Tipo de Pregunta	Total preguntas	Arbitro 1	Arbitro 2	Arbitro 3
Directas	20	3,50	3,60	3,50
Condicionales	20	3,10	3,10	3,38
Razonamiento	20	2,00	2,15	2,10

Fuente: Elaboración propia

El promedio de las preguntas directas dado por el primer árbitro es de 3.5 que considerando nuestro rango de puntuación puede considerarse bueno, el promedio del segundo árbitro es de 3.6 y del tercer árbitro es de 3.5; se puede observar bastante similitud en los promedios de los 3 árbitros, esto indica que la evaluación del modelo tiene más coherencia. Por otro lado se observa que solo 2 preguntas fueron calificadas con el puntaje mínimo, esto indica que el modelo tiene buena eficiencia en este tipo de preguntas.

Respecto a las preguntas condicionales el promedio general entre 3.10 y 3.4 aun que el promedio no es bajo, podemos considerar que algunas preguntas obtuvieron buen promedio mientras que otras no tanto, también se observa la obtención de 4 puntajes mínimos asignados por los árbitros y varios valores de 2, cabe mencionar que en este experimento si se evidenció algunos puntajes mayores por parte del tercer árbitro, esto nos muestra que aunque los

puntajes varíen esta variación no debe ser tan grande porque se pondría en tela de juicio la capacidad de los árbitros en este experimento. La disminución de los puntajes pone de manifiesto ya, las dificultades que este modelo empieza a sufrir en este tipo de preguntas.

Respecto a las preguntas de razonamiento se observa que el promedio global es de 2.1 lo cual nos muestra una gran deficiencia del modelo en este tipo de preguntas, existen 8 respuestas cuyas puntuaciones oscilan entre 1-2 esto muestra que la mayoría de preguntas no lograron cubrir el razonamiento necesario, esto evidencia un gran problema del modelo al momento de sintetizar información.

Evaluación humana del modelo google/flan-t5-small

Para el modelo T5 se consideró todos los detalles del experimento con el modelo BERT, es importante recalcar para el modelo T5 se realizaron varias pruebas con librerías propias para tareas QA y también con prompts directos, aunque la diferencia no es mucha se tuvo mejores resultados aplicando el criterio del prompt, que fue establecido en una función de la siguiente manera:

```
def build_prompt(question, context):
return f"""
Responde la pregunta basandote unicamente en el contexto.
Responde en spanish de forma breve y precisa.

Contexto:
{context}

Pregunta:
{question}

Respuesta:
""".strip()
```

Código 5.1: Prompt modelo T5

Buscando la igualdad de condiciones se dispuso usar la misma cantidad de preguntas al igual que el modelo BERT, con la finalidad de obtener los resultados adecuados. Las puntuaciones otorgadas por los árbitros al modelo se observan en la Tabla 18.

Tabla 18: Promedio de calificaciones obtenidas por tipo de pregunta para el modelo Flan-T5-small

Tipo de pregunta	Total preguntas	Árbitro 1	Árbitro 2	Árbitro 3
Directas	20	3,21	3,17	3,15
Condicionales	20	3,42	3,28	3,21
Razonamiento	20	2,10	2,09	2,08

Fuente: Elaboración propia

Se observa que el modelo en preguntas directas tiene un puntaje global aceptable de 3.18 sin embargo se observa también puntajes entre 1-2, esto refleja un extraño comportamiento del modelo además hubo palabras en ingles en medio de las respuestas lo que nos muestra que a pesar de ser un modelo multilingue sus alucinaciones en cuanto al idioma aún son palpables.

En cuanto a preguntas condicionales el modelo obtuvo un puntaje global de 3.3, sin embargo existe una gran fluctuación en los puntajes que oscilan entre 1-2 y los demás puntajes también son muy variables y a pesar de tener varios puntajes de 5 existen muchos 1 que ponen en tela de juicio el rendimiento del modelo. También se observa que el modelo es capaz de extraer respuestas con contextos enteros que completan toda la información solicitada esto determino los puntajes de 5 por parte de los árbitros.

El promedio global en preguntas de razonamiento fue de 2.09 lo que se considera relativamente bajo, se observa además 7 puntajes completamente bajos, esto nos sugiere que el modelo no logro realizar inferencias de forma correcta lo que provoca que los puntajes sean muy disparejos, sin embargo también se observa 11 respuestas entre 4-5. Es importante aclarar que este modelo fue el que mejor resultados arrojó entre los modelos T5 usados, en gran medida gracias a su capacidad multilingue. También observamos una gran simetría en los valores otorgados por los árbitros a este modelo.

Evaluación humana del modelo microsoft-Phi-4-mini-instruct

En este experimento se consideró todos los detalles aplicados en la evaluación de los otros dos modelos, tanto en cantidad de preguntas como de respuestas, además de ello y al igual que en modelo T5 se elabora una función prompt para indicar al modelo que debe trabajar con tareas de QA.

```
def build_phi_prompt(question, context):
return f"""
Eres un sistema de preguntas y respuestas sobre las leyes deportivas del
  ajedrez.Responde unicamente con base en el contexto proporcionado. No
 agregues informacion externa. Da una respuesta breve, precisa y en
 spanish.

Contexto:
{context}

Pregunta:
{question}

Respuesta:
""".strip()
```

Código 5.2: Prompt modelo Phi4

Luego de estos ajustes, las puntuaciones otorgadas por los árbitros al modelo, fueron las siguientes:

Tabla 19: Promedio de calificaciones obtenidas por tipo de pregunta para el modelo Microsoft Phi-4-mini-instruct

Tipo de pregunta	Total preguntas	Árbitro 1	Árbitro 2	Árbitro 3
Directas	20	4,61	4,58	4,58
Condicionales	20	4,35	4,35	4,32
Razonamiento	20	4,10	4,05	4,12

Fuente: Elaboración propia

La tabla 19 muestra las puntuaciones asignadas por los árbitros a las respuestas generadas del modelo Phi-4 donde podemos observar que el modelo obtuvo una puntuación promedio de 4.5 en preguntas directas lo cual puede calificarse como excelente teniendo solo dos preguntas con una puntuación de uno, además de que los puntajes de los arbitros son muy simétricos lo que se puede juzgar con respuestas aceptables para diferentes interpretes.

El promedio global a las preguntas condicionales fue de 4.3 lo que sitúa ya a este modelo por encima de los otros modelos, se observa también que este modelo obtuvo solamente un dos como puntaje mínimo, sin embargo existen respuestas que no satisficieron a los árbitros completamente. El rendimiento de este modelo ya se muestra muy por encima de los otros y denota un gran rendimiento multilingue.

EL promedio global obtenido por el modelo en preguntas de razonamiento es de 4.1, se observa ademas que se dieron puntuaciones de 1-2 lo que afecto ese promedio, estos puntajes se muestran muchos mejores que los puntajes obtenidos por los otros dos modelos, aún a pesar de las pequeñas dificultades tenidas en las preguntas de razonamiento. Este modelo permite darnos cuenta de su capacidad para hacer inferencias o razonamientos respecto a un determinado tema. La mayor efectividad en preguntas situacionales y de razonamiento nos muestran que es ahí el enfoque donde deben buscar mejorar los LLMs.

La evaluación humana nos muestra la gran complejidad que tiene trabajar con tareas de QA en contextos cerrados y más aún si son estructurados. Por ello es importante un trabajo coordinado tanto de desarrolladores y expertos en el área de las tareas de QA.

5.2. Discusión

5.2.1. Fiabilidad de los LLMs

A pesar de lograr la comprensión de los contextos y generar respuestas es importante entender que los LLMs presentan fallas que afectan su rendimiento en tareas de preguntas y respuestas, el modelo T5 mostró confusiones respecto al idioma lo que se evidenció en los resultados de este modelo, por su parte BERT tiene un gran rendimiento aplicando la forma extractiva de las respuestas, mientras que Phi-4 logra comprencion y generacion de forma muy buena superando el 75 % en efectividad.

5.2.2. Análisis de los Modelos usados

1. Luego de comparar todos los modelos BERT y Phi4 alcanzan niveles muy aceptables respondiendo preguntas directas dentro de contextos estructurales, los tres modelos evaluados presentan diferencias claras respecto al tratamiento de las respuestas en el dominio especifico del ajedrez.

2. En términos de eficiencia podemos decir que el modelo Phi4 muestra un rendimiento más sobresaliente en su razonamiento por lo que obtuvo puntajes mayores en todas las evaluaciones.

3. Por su parte el modelo T5 muestra respuestas aceptables en preguntas directas y condicionales sin embargo se observa una baja importante en las preguntas de razonamiento, entre sus fallas comunes se observan respuestas vagas, alucinaciones y falta de razonamiento.

4. El modelo BERT tiene un rendimiento aceptable en preguntas directas casi parecido a los del modelo T5, sin embargo en preguntas que implican un razonamiento mostró considerables deficiencias.

5. Por otra parte es considerable destacar que ninguno de los modelos tuvo un rendimiento del 100 %, esto pude deberse a diversas razones como el tamaño de los modelos, la mejora de los contextos, entorno de ejecución o el idioma español.

6. El conjunto de datos que deseamos usar es determinante en este tipo de tareas, no es suficiente con pasar el texto en bruto es necesario ceñirnos a las exigencias del modelo por eso es muy importante dominar nuestros contextos.

7. No existe un método infalible ni para la generación de contextos ni para la generación de preguntas, todos deben ser revisados con detenimiento y de ser posible por expertos

en el área.

8. Los LLMs son muy buenos en tareas generativas sin embargo en tareas de QA aun existen muchas falencias sobre todo en el razonamiento que realizan sobre los contextos.

9. A medida que la investigación crece también lo hace la complejidad de los contextos y las preguntas, no es posible realizar una evaluación plana en este tipo de tareas, también es posible dar varias respuestas validas a una determinada pregunta con el fin de dar mayor flexibilidad al modelo.

5.2.3. Análisis de las métricas

1. La métrica estricta es la más exigente en este tipo de tareas, sin embargo no refleja la parte creativa del modelo, esto ocasiona que respuestas correctas obtengas puntajes bajos aún siendo semánticamente correctas.

2. La métrica Lenient Accuracy es la que muestra resultados más precisos en todos modelos, al permitir las variaciones en las respuestas permite dar por correctas respuesta semánticamente similares a las de nuestro dataset. La similitud semántica es muy similar a Lenient Accuracy sin embargo se observa una mejora en los promedios de las respuestas debido a que no solo verifica si una palabra es parecida a otra sino que evalúa la distancia semántica, útil en estos casos por el uso de embeddings.

3. Las métricas tradicionales no son suficientes en tareas de QA, casi siempre es necesaria la intervención humana para tener una mejor perspectiva del rendimiento de los LLMs.

4. Es importante la evaluación humana en este tipo de tareas porque se observo casos donde los modelos brindaban porcentajes aceptables a respuestas incoherentes según evaluación humana.

Conclusiones

1. La conclusión en base al objetivo general es: Se logro la evaluación de los LLMs en tareas de preguntas y respuestas de manera satisfactoria, esta evaluación nos permite comprender el funcionamiento de los LLMs en dominios cerrados, la manera en que recuperan y generan las respuestas, por otro lado también se evidencia la importancia de un buen conjunto de datos para las evaluaciones, la flexibilidad de los mismos permiten realizar una evaluación mucho más eficiente.

2. Las conclusiones basadas en el primer objetivo especifico son: Los errores que cometen los LLMs en tareas de preguntas y respuestas son afectados por las siguientes razones: la mala elaboración de contextos y la mala generación de la pregunta, ademas otro factor para los errores de los LLMs son el tipo de tokenizador que emplean y el idioma en que fueron entrenados.

3. Las conclusiones basadas en el segundo objetivo especifico son: Las preguntas factuales son las mejor resueltas por los tres modelos evaluados, esto nos permite concluir que los tres modelos tienen una buena capacidad de recuperación directa dentro de los contextos.

   Las preguntas condicionales y situacionales comprenden un desafió mayor en los modelos, esto pone claro que el razonamiento en dominios cerrados sigue siendo una limitación importante en los LLMs.

Las diferencias estructurales afectan el desempeño de los modelos, en el caso de BERT, muestra limitaciones en la generación de respuestas de razonamiento. T5, diseñado específicamente para tareas de texto a texto, presenta también fallas en su capacidad de razonamiento. Phi-4 demuestra mayor flexibilidad en la redacción, y un mejor enfoque en el razonamiento, se puede afirmar que este modelo es adecuado en este tipo de tareas.

4. Las conclusiones basadas en el tercer objetivo especifico son: Las métricas automáticas no son suficientes por sí solas para evaluar los LLMs en dominios cerrados, las métricas stric accuracy, lenient accuracy y similitud semantica son mas aceptables en tareas de preguntas y respuestas, sin embargo la evaluación humana sigue siendo la mejor elección si se desea evaluar modelos en dominios cerrados.

5. Las conclusiones basadas en el cuarto objetivo especifico son: El dataset diseñado para esta investigación resultó adecuado para la evaluación de los modelos. Aunque el tamaño era pequeño, el dataset permitió identificar patrones de comportamiento, fortalezas y debilidades de los modelos evaluados. Esto confirma que datasets especializados y bien diseñados pueden ser útiles para evaluar modelos en dominios técnicos específicos.

Recomendaciones

1. Una recomendación a futuras investigaciones es mejorar y ampliar el dataset, realizando una mejora en los contextos y los tipos de preguntas, para esta labor es recomendable contar con un equipo de expertos en el dominio que se desea investigar.

2. Se recomienda la exploración de modelos multinodales, ya estos modelos integran imágenes y datos numéricos, esto puede ser muy útil en la comprensión de contextos complejos que pueden incluir figuras, valoraciones y notación de jugadas. El desarrollo y evaluación de estos modelos podría ser un área de interés para futuras investigaciones.

3. Los modelos evaluados presentan ciertas dificultes en las tareas de razonamiento, por ello se recomienda realizas un fine-tuning de los modelos con dataset de dominio especifico, para ello sería útil contar con un dataset de mayor tamaño y si es posible incluir leyes deportivas de otros deportes.

4. Las métricas automáticas solo brindan una medida objetiva del rendimiento, por ello se recomienda hacer uso de evaluación humana en dominios cerrados. Además se recomienda realizar análisis más profundos de las respuestas.

5. Este estudio se baso estrictamente en las leyes de ajedrez, por ello se recomienda evaluar los modelos en diferentes dominios como la historia, antropología, ingeniería, etc. Evaluar los LLMs en diferentes dominios nos brindará conocimiento valioso de sus capacidades y limitaciones de los modelos en tareas especializadas.

6. Se recomienda la evaluación de LLMs especializados en español, tales como BETO, MarIA, RuPERTa y XLM-RoBERTa. Estos modelos al haber sido entrenados total o parcialmente sobre corpus de texto en español, podrían ofrecer un desempeño superior en tareas de preguntas y respuestas relacionadas con documentos normativos redactados en este idioma.

7. Se recomienda el estudio de arquitecturas basadas en Retrieval-Augmented Generation (RAG), las cuales combinan LLMs con mecanismos de recuperación de información desde fuentes documentales externas.

8. El estudio de los LLMs puede llegar a ser muy complejo, por ello se recomienda su análisis en tareas especificas, observando su rendimiento, fortalezas y debilidades antes de hacer su uso en aplicaciones.

9. Se recomienda que expertos en la materia puedan ayudar en la generación de los contextos y preguntas además de ayudar en los ajustes que puedan surgir en el proceso.

Bibliografía

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., and Zhang, Y. (2024). Phi-4 technical report.

Alammar, J. and Grootndorst, M. (2024). *Hands-On Large Language Models*. o'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Alec, R., Jeffrey, W., Rewon, C., David, L., Dario, A., and **, I. S. (2019). Language models are unsupervised multitask learners.

Alec, R., Karthik, N., Tim, S., and Ilya, S. (2018). Improving language understanding by generative pre-training.

Arbiters, F. (2024). *ARBITERS' / ÁRBITROS, manual en español*.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Bisong, E. (2019). *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R. B., Arora, S., von Arx, S., Bernstein,

M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L. E., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., and et al. (2021). On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Cann, T.J.B., D. B. C. T. e. a. (2025). Using semantic similarity to measure the echo of strategic communications.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Bethard, S., Carpuat, M., Apidianaki, M., Mohammad, S. M., Cer, D., and Jurgens, D., editors, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Chen, A., Stanovsky, G., Singh, S., and Gardner, M. (2019). Evaluating question answering evaluation. In Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., and Chen, D., editors,

Proceedings of the 2nd Workshop on Machine Reading for Question Answering, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. (2021). Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Cheng-Han Chiang, H.-y. L. (2024). Can large language models be an alternative to human evaluations.

Dahl, M., Magesh, V., Suzgun, M., and Ho, D. E. (2024). Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Edward Choi, Andy Schuetz, W. F. S. J. S. (2017). Using recurrent neural network models for early detection of heart failure onset, journal of the american medical informatics association. *Jamia*, 24/Pages 361–370.

Ehsan Kamalloo, Nouha Dziri, C. L. A. C. D. R. (2023). Evaluating open-domain question answering in the era of large language models.

Eisenstein, J. (2018). *Natural Language Processing*. MIT Press.

Farea, A., Yang, Z., Duong, K., Perera, N., and Emmert-Streib, F. (2022). Evaluation of question answering systems: Complexity of judging a natural language.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54.

Fischer, K., Fürst, D., Steindl, S., Lindner, J., and Schäfer, U. (2024). Question: How do large language models perform on the question answering tasks? answer:.

Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Graeme Hirst, University of Toronto.

Guelma, A. (2024). A survey on dataset development techniques for qa systems. *Aicha. Aggoune*.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. *CoRR*, abs/1506.03340.

Hernández Sampieri, R., Fernández Collado, C., and Baptista Lucio, P. (2014). *Metodologia de la Investigación*. McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.

Hu, T. and Zhou, X.-H. (2024). Unveiling llm evaluation focused on metrics: Challenges and solutions.

Hugging-Face (2025). Hugging face hub documentation.

Jain, S. M. (2022). *Hugging Face*, pages 51–67. Apress, Berkeley, CA.

Jeong, C. (2024). Domain-specialized llm: Financial fine-tuning and utilization method using mistral 7b. *Journal of Intelligence and Information Systems*, 30(1):93–120.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., and Fung, P. (2022). Survey of hallucination in natural language generation. *CoRR*, abs/2202.03629.

Jiao, F., Teng, Z., Ding, B., Liu, Z., Chen, N., and Joty, S. (2024). Exploring self-supervised logic-enhanced training for large language models. In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 926–941, Mexico City, Mexico. Association for Computational Linguistics.

Johnson, S. and Hyland-Wood, D. (2024). A primer on large language models and their limitations.

Joshi, S. (2025). Evaluation of large language models: Review of metrics, applications, and methodologies.

Jurafsky, D. and Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Kumar Pandey, A. and Sekhar Roy, S. (2024). Extractive question answering over ancient scriptures texts using generative ai and natural language processing techniques. *IEEE Access*, 12:101197–101209.

Lan, Y., He, G., Jiang, J., Jiang, J., Zhao, W. X., and Wen, J. (2021). Complex knowledge base question answering: A survey. *CoRR*, abs/2108.06688.

Latif, A. and Kim, J. (2024). Evaluation and analysis of large language models for clinical text augmentation and generation. *IEEE Access*, 12:48987–48996.

Lewis, T., Leandro, v. W., and Thomas, W. (2022). *Natural Language Processing with Transformers Building Language Applications with Hugging Face Implementation.* Firts edition edition.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C. A., Manning, C. D., Re, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., WANG, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N. S., Khattab, O., Henderson, P., Huang, Q., Chi, R. A., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. (2023). Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

Liddy, E. (2001). *Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed.* NY. Marcel Decker, Inc.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, Z., Feng, M., dos Santos, C. N., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding.

Martín de Pablo, L. (2024). Evaluación y comparativa de modelos de lenguaje a gran escala (llm) en local para arquitecturas rag.

Martínez Barco, P., Luis, V. J., Estela, S., and David, T. (2007). Sistemas de pregunta-respuesta.

Nassiri, K., A. M. (2023). Transformer models used for text-based question answering systems.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Raymond Lee (2025). *Natural Language Processing A Textbook with Python Implementation*. Second edition edition.

Rosset, C. (2020). Turing-nlg: A 17-billion-parameter language model by microsoft. *Microsoft Research Blog*.

Rothman, D. (2022). *Transformers for Natural Language Processing*. Packt Publishing Ltd., second edition edition.

Schur, A. and Groenjes, S. (2023). *Comparative Analysis for Open-Source Large Language Models*, pages 48–54.

Sun, H., Cohen, W. W., and Salakhutdinov, R. (2021). Conditionalqa: A complex reading comprehension dataset with conditional answers.

Supo Condori, J. A. (2020). *Metodología de laInvestigación Científica*. BIOESTADISTICO EEDU.EIRL.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., Sessa, P. G., Chowdhery, A., Roberts, A., Barua, A., Botev, A., Castro-Ros, A., Slone, A., Héliou, A., Tacchetti, A., Bulanova, A., Paterson, A., Tsai, B., Shahriari, B., Lan, C. L., Choquette-Choo, C. A., Crepy, C., Cer, D., Ippolito, D., Reid, D., Buchatskaya, E., Ni, E., Noland, E., Yan, G., Tucker, G., Muraru, G.-C., Rozhdestvenskiy, G., Michalewski, H., Tenney, I., Grishchenko, I., Austin, J., Keeling, J., Labanowski, J., Lespiau, J.-B., Stanway, J., Brennan, J., Chen, J., Ferret, J., Chiu, J., Mao-Jones, J., Lee, K., Yu, K., Millican, K., Sjoesund, L. L., Lee, L., Dixon, L., Reid, M., Mikuła, M., Wirth, M., Sharman, M., Chinaev, N., Thain, N., Bachem, O., Chang, O., Wahltinez, O., Bailey, P., Michel, P., Yotov, P., Chaabouni, R., Comanescu, R., Jana, R., Anil, R., McIlroy, R., Liu, R., Mullins, R., Smith, S. L., Borgeaud, S., Girgin, S., Douglas, S., Pandya, S., Shakeri, S., De, S., Klimenko, T., Hennigan, T., Feinberg, V., Stokowiec, W., hui Chen, Y., Ahmed, Z., Gong, Z., Warkentin, T., Peran, L., Giang, M., Farabet, C., Vinyals, O., Dean, J., Kavukcuoglu, K., Hassabis, D., Ghahramani, Z., Eck, D., Barral, J., Pereira, F., Collins, E., Joulin, A., Fiedel, N., Senter, E., Andreev, A., and Kenealy, K. (2024). Gemma: Open models based on gemini research and technology.

Team, M. A. (2023). Mistral 7b the best 7b model to date, apache 2.0.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N.,

Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models.

Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M. R., Weissenborn, D., Krithara, A., Petridis, S., Polychronopoulos, D., et al. (2015). An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1–28.

Vaswani, A., Noam, S., Niki, P., Jakob, U., Llion, J., Aidan, N. G., Lukasz, K., and Polosukhin, I. (2023). Attention is all you need. *arXiv*.

Xiao, T. and Zhu, J. (2023). Introduction to transformers: an nlp perspective.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Yuan, L. (2018). Bert++: Reading comprehension on squad. Brussels, Belgium. Association for Computational Linguistics.

Zhou, K., Hwang, J., Ren, X., and Sap, M. (2024). Relying on the unreliable: The impact of language models' reluctance to express uncertainty. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3623–3643, Bangkok, Thailand. Association for Computational Linguistics.

Anexos

Para este trabajo de investigación se uso la herramienta COLAB en su integridad con el lenguaje Python, a continuación se deja el conjunto de preguntas y respuestas elaboradas, ademas de las respuestas dados por los 3 modelos para la evaluación humana.

Preguntas y respuestas generadas con el modelo BERT

Tabla 20: Preguntas directas utilizadas en la evaluación del modelo BERT

Pregunta	**Respuesta**
¿Cómo se juega una partida de ajedrez?	Entre dos adversarios.
¿Qué jugador realiza el primer movimiento en una partida de ajedrez?	El jugador con las piezas claras.
¿Cuándo se dice que un jugador está en juego?	Cuando se ha realizado el movimiento de su adversario.
¿Cuál es el objetivo de cada jugador?	Situar al rey del adversario bajo ataque.
¿Qué no está permitido respecto al rey?	Dejar el propio rey bajo ataque.
¿Cómo está dividido el tablero de ajedrez?	En 64 casillas cuadradas.
¿Cómo se coloca el tablero entre los jugadores?	Con la casilla blanca en la esquina derecha.

¿Cuántas piezas dispone cada bando?	16 piezas.
¿Cómo se denominan las hileras verticales?	Columnas.
¿Cómo puede moverse el alfil?	A lo largo de las diagonales.
¿Cómo se mueve el caballo?	En forma de L.
¿Qué ocurre cuando un peón llega a la fila más alejada?	Debe promocionar.
¿Cómo puede moverse el rey?	A cualquier casilla adyacente.
¿Qué es un movimiento legal?	Un movimiento que cumple todas las reglas.
¿Qué se le debe permitir a un jugador después de mover?	Detener su reloj.
¿Quién puede ajustar las piezas?	El jugador cuyo reloj está en marcha.
¿Qué ocurre con el tiempo no consumido?	Se añade al siguiente período.
¿Qué debe verificarse después de una caída de bandera?	Si se completó el número mínimo de movimientos.
¿Qué debe hacer el jugador que desplaza piezas accidentalmente?	Restablecer la posición correcta.
¿Qué hace el árbitro ante el segundo movimiento ilegal?	Decreta la pérdida de la partida.

Fuente: Elaboración propia

Tabla 21: Preguntas condicionales utilizadas en la evaluación del modelo BERT

Pregunta	**Respuesta**
¿Cuándo la partida es tablas?	Cuando ningún jugador puede dar mate.
¿Cuándo se considera que una pieza ataca a otra?	Cuando puede capturarla conforme a las reglas.
¿Cómo captura un peón?	En diagonal sobre una columna adyacente.
¿Qué hace un jugador si no tiene la pieza deseada para promocionar?	Puede detener el reloj.
¿Qué jugador puede ajustar piezas?	Solo el jugador que está en juego.
¿Qué sucede si un jugador llega tarde?	Pierde la partida.
¿Una caída de bandera basta para declarar derrota?	No por sí sola.
¿Qué ocurre si los colores fueron invertidos?	Se reinicia la partida si corresponde.
¿Qué ocurre si la partida termina antes de detectar colores invertidos?	El resultado se mantiene.
¿Qué ocurre si un peón llega al final sin promocionar?	El movimiento es ilegal.
¿Qué se hace si las piezas fueron desplazadas?	Restaurar la posición anterior.
¿Qué ocurre si un jugador tiene menos de cinco minutos?	Puede suspender la anotación.
¿Qué deben hacer los jugadores si no existe una planilla completa?	Reconstruir la partida.

Pregunta	Respuesta
¿Quién autoriza abandonar la zona de juego?	El árbitro.
¿Dónde pueden almacenarse dispositivos electrónicos?	Según las bases del torneo.
¿Cómo se sanciona incumplir las Leyes del Ajedrez?	Con pérdida de la partida.
¿Qué ocurre si se mueve después de sellar?	Debe anotarse como movimiento secreto.
¿Qué ocurre si el movimiento sellado es ilegal?	Pierde la partida.
¿Qué puede hacer el adversario si el jugador no está presente?	Anotar su respuesta en la planilla.
¿Cuándo puede corregirse un error en los relojes?	Antes del primer movimiento tras la reanudación.

Fuente: Elaboración propia

Tabla 22: Preguntas de razonamiento utilizadas en la evaluación del modelo BERT

Pregunta	Respuesta
¿Qué ocurre si se mueve una pieza a una casilla ocupada por otra del mismo color?	El movimiento es ilegal.
¿Qué diferencia hay entre el movimiento de la torre y la dama?	La dama combina movimientos de torre y alfil.

Pregunta	**Respuesta**
¿Cómo avanza el peón en su primer movimiento?	Una o dos casillas hacia adelante.
¿Cuándo se pierde el derecho al enroque?	Cuando el rey o la torre se han movido.
¿Se puede poner en jaque el propio rey?	No.
¿Cómo puede terminar una partida?	Por mate, tablas o abandono.
¿Qué significa la caída de bandera?	Que se agotó el tiempo asignado.
¿Cuándo se considera completado un movimiento?	Cuando se cumplen las condiciones reglamentarias.
¿Cómo debe pulsarse el reloj?	Con la misma mano utilizada para mover.
¿Qué conductas relacionadas con el reloj están prohibidas?	Golpearlo o manipularlo incorrectamente.
¿Qué debe hacer el árbitro ante una irregularidad?	Restablecer posición y tiempos.
¿Qué se hace si el tablero fue colocado incorrectamente?	Continuar la partida con corrección.
¿Qué ocurre si se detecta un movimiento ilegal?	Se restaura la posición previa.
¿Cómo se considera un enroque realizado con dos manos?	Movimiento ilegal.
¿Qué ocurre si un jugador no puede anotar?	Puede utilizar un asistente autorizado.

Pregunta	**Respuesta**
¿Qué sucede si ninguno de los jugadores anota en apuro de tiempo?	El árbitro o asistente registra la partida.
¿Puede un torneo restringir acuerdos de tablas?	Sí, según sus bases.
¿Cuándo existe triple repetición?	Cuando la misma posición aparece tres veces.
¿Cuándo aplica la regla de los 50 movimientos?	Tras 50 movimientos sin captura ni movimiento de peón.
¿Qué deben hacer los jugadores al terminar la partida?	Abandonar la zona de juego.

Fuente: Elaboración propia

Preguntas y respuestas generadas con el modelo T5

Tabla 23: Preguntas directas - Modelo T5

Pregunta	**Respuesta**
¿Cómo se juega una partida de ajedrez?	entre dos adversarios que mueven alternativamente sus propias piezas sobre un tablero cuadrado, llamado ajedrez. Pregunta: Cómo se juega una partida de áje
¿Qué jugador realiza el primer movimiento en una partida de ajedrez?	El jugador con las piezas claras realiza el primer movimiento y posteriormente los jadores mueven alternativamente
¿Cuándo se dice que un jugador está en juego?	Si dice que un jugador está en game cuando se ha realizado el movimiento de su adversario.
¿Cuál es el objetivo de cada jugador?	El objetivo de cada jugador
¿Qué no está permitido respecto al rey?	Del jugador que sita al rey de su adversario bajo ataque and deja al adversaro sin movimientos legales se dice que ha dado mate and que has ganado la partida. No está permitido de
¿Cómo está dividido el tablero de ajedrez?	El tablero de ajedrez es un cuadrado dividido en 64 casillas cuádradas del mismo tiempo, con distribución de 8 por 8, alternando casilles clara y oscur

Pregunta	**Respuesta**
¿Cómo se coloca el tablero entre los jugadores?	Cómo se coloca the tablero entre los jugadores?
¿Cuantas piezas dispone cada bando?	Al comienzo de la partida, cada bando dispone de 16 piezas
¿Cómo se denominan las hileras verticales	columnas
¿Cómo puede moverse el alfil?	El alfil puede moverse cualquier casilla lo largo de una de las diagonales sobre los que se encuentra. Inicialmente cada jugador tiene dos alfiles, one that se mue
¿Cómo se mueve el caballo?	How do I move the caballo to one of the casillas más próximas a la que se encuentra?
¿Qué ocurre cuando un peón llega a la fila más alejada de su posición inicial?	Cuando un peón llega a la fila más alejada desde su posición inicial, debe cambiarse como parte del mismo movimiento por una dama, torre, alfil
¿Como puede moverse el rey?	También puede enrocar, movimiento que cuenta como un solo movers del rey y in el que él se traslada dos casillas hacia una torre and a
¿Que es un movimiento legal?	Un movimiento es legal cuando cumple todos los requisitos de mimiendo. Una posición é ilegal quando no puede alcanzarse mediante una serie de legale

Pregunta	Respuesta
¿Que se le debe permitir a un jugador despues de realizar su movimiento?	A un jugador siempre se le debe permitir detener su reloj después de realizar su movimiento, incluso si su adversario haizo el siguiente. El tiempo transcurrido entre realizado
¿Quién puede ajustar las piezas?	Si un jugador is incapaz de utilizar el reloj puede proporcionar un asistente aceptable para él árbitro para realizar esta operación
¿Que ocurre con el tiempo no consumido durante un período?	Que ocurre con el tiempo no consumido durante un perodio se aade al tiemo disponible para la siguiente, si lo hubiera. In las leyes, the increment and la de
¿Qué debe verificarse inmediatamente después de una caída de bandera?	Inmediatamente después de una cada de bandera deben verificarse si se completó el nmero mnimo de movimientos exigido dentro desdel tiempo an
¿Qué debe hacer el jugador que desplaza una o más piezas?	Si un jugador desplaza una o más pieza deberá restablecer la posición correcta de su propio tiempo. Si fuera necesariamente, tanto, el judador como su adversarión pueden
¿Que hace el arbitro para el segundo movimiento ilegal completado por el mismo jugador?	For the primer movimiento ilegal completado, the arbitro concederá dos minutos de tiempo extra al adversario

Fuente: Elaboración propia

Tabla 24: Preguntas condicionales - Modelo T5

Pregunta	**Respuesta**
¿Cuando la partida es tablas?	Si la posición es tal que ninguno de los jugadores puede dar mate, la partida era tablas.
¿Cuándo se considera que una pieza ataca a otra?	Si puede efectuar una captura en esa casilla conforme a las reglas de movimiento
¿Como captura una pieza el peón?	How can you captura a pieza del adversario que esté en diagonal delante?
¿Qué hace un jugador si en una promoción no tiene a su la pieza deseada?	El jugador puede parar el reloj y requerir al árbitro que le proporcione sa pieza. El arbitero debe propocionarla o reiniciar la piecia solicit
¿Qué jugador puede ajustar piezas?	solo el jugador
¿Qué sucede si un jugador se presenta después del tiempo de incomparecencia?	Answer the pregunta basándote nicamente en el contexto
¿Una simple caída de bandera basta para declarar la derrota?	A simple cada no basta para declarar la derrota: debe comprobar la posibilidad real de mate del adversario.
¿Qué ocurre si una partida ha comienzado con los colores cambiados y ambos jugadores han hecho menos de 10 movimientos?	Si una partida ha comenzado con los colores cambiados y ambo jugadores han hecho menos de 10 movimientos, se interrumpirá and se jogará un nuevo partide con la

Pregunta	Respuesta
¿Qué ocurre si una partida con colores invertidos termina antes de que se realicen diez movimientos?	el resultado se mantendrá. Esto aplica, por exemplo, si termina por jaque mate, ahogadido, rendición o according de tablas permitida.
¿Qué ocurre si un jugador mueve el peón a la fila más lejana sin sustituirlo por una nueva pieza?	El movimiento es ilegal. In that caso, the peón se sustituye por una dama del mismo color
¿Si se comprueba que algunas piezas han sido desplazadas de sus casillas que se deberá hacer?	Si durante una partida se comprueba que algunas piezas han sido desplazadas de sus casilla, deberá restablecer la posición anterior a la irregularidad. Si no puede identificarse
¿Si un jugador dispone de menos de cinco minutos y no tiene tiempo adicional por movimiento está obligado a anotar?	Si un jugador dispone de menos de cinco minutos en su reloj in algn momento de un perodo y no tiene un tiempo indicional de 30 segundas ou más por
¿Qué debe hacer los jugadores si no hay ninguna planilla completa disponible?	Si no hay ninguna planilla completa disponible, los jugadores deben reconstruir la partida sobre un segundo tablero bajo el control del árbitr o un asistente
¿Quien da el permiso para salir de la zona de juego?	solo con permiso del arbitro un jugador puede: abandonar el recinto de jóve, salir de la zona de partido, or una persona que no sea ni jogar o árbiter

Pregunta	Respuesta
¿Los dispositivos electrónicos pueden almacenarse en la bolsa de los jugadores?	Las bases del torneo pueden permitir que los dispositivos electrónicos se almacenen en la bolsa de los jugadores, siempre que estén completamente apagadas. Esa
¿Como se sansiona la negativa de un jugador a cumplir las Leyes del Ajedrez?	Cómo será sancionada con la pérdida de la partida, el arbitro decidirá la puntuación del adversario
¿Qué ocurre si un jugador realiza un movimiento sobre el tablero después de ser advertido para sellar su movimiento?	If a jugador, después de ser advertido por el arbitro para sellar su movimiento, realiza un mutimiendo sobre en su planilla
¿Que ocurre con el jugador cuya anotación del movimiento secreto sea ilegal?	Excepto en los casos previstos por los artculas 5, 6.9 y 9.6, lose la partida el jugador cuya annotación de la movimiento secreto sea ambigua
¿Qué puede hacer el adversario si no está presente el jugador que selló el movimiento a la hora de reanudar la partida?	Su adversario puede anotar su respuesta en la planilla, sellar su movimiento in un nuevo sobre, detener su reloj y poner ejecutar sus jugador de forma normal
¿Cuándo puede corregirse un error en los relojes en la reanudación de la partida?	If in la reanudación de la partida, antes de hacer su primer movimiento, uno de los jugadores demuestra que el tiempo consumido fue colocadó incorrectamente en algun

Fuente: Elaboración propia

Tabla 25: Preguntas de razonamiento - Modelo T5

Pregunta	**Respuesta**
¿Qué ocurre si se mueve una pieza a una casilla ocupada por otra del mismo color?	No está permitido mover una pieza a un casilla ocupada por un la piezza del mismo color
¿Qué diferencia hay entre el movimiento de la torre y el de la dama?	Al realizar estos movimientos, ni el alfil, la torre, or la dama pueden pasar sobre otra pieza.
¿Como avanza el peón en su primer movimiento?	In su primer movimiento también puede avanzar dos casillas en la misma columna, siempre que ambas está desocupadas.
¿Cuándo se pierde definitivamente el derecho al enroque?	El enroque está temporalmente impedido si el rey está na casilla atacada, si debe cruzar una cazille atacada si va ocupar un acasillo atacada
¿Se puede poner en jaque el propio rey?	Se dice que el rey está en jaque si están atacado por una ou más piezas del adversario, incluso aunque sas piezies no pudieran moverse a su casilla porque
¿Como puede terminar una partida?	Cómo puede terminar una partida?
¿Qué significa la caída de bandera?	reloj hace referencia a uno de esos indicadores

Pregunta	Respuesta
¿Cuando un movimiento se completa?	Durante la partida, cada jugador, una vez realizado su movimiento sobre el tablero, debe detener su reloj y poner en marcha de su adversario esto completa la moveimi
¿Cómo debe pulsar el reloj un jugador después de mover y qué sucede si usa la otra mano?	A jugador debe pulsar su reloj con la misma mano con que realizó su movimiento. Además, está prohibido mantener el debido sobre la reloji o rondando
¿Qué conductas relacionadas con el reloj están prohibidas?	Los jugadores deben manejar el reloj de ajedrez correctamente. Está prohibido golpearlo violentamente, cogerló, pulsarla antes de dejugar o tirarle.
¿Qué debe hacer el árbitro si se produce una irregularidad y las piezas deben ser reubicadas?	Quién debe hacer el árbitro si se produce una irregularidad y las piezas deben ser reubicadas para un posición anterior?
¿Qué se hace si se comprueba que el tablero no fue colocado correctamente?	In cambio, si si comprueba que el tablero no fue colocado correctamente, la partida continuará, pero la posición deberá transferirse un tableryo correctly.
¿Que ocurre si durante la partida se comprueba que se completo un movimiento ilegal?	A movimiento ilegal se completa una vez que el jugador ha pulsado su reloj. Si durante la partida se comprueba que se completeó un moveimiendo illegal, se

Pregunta	Respuesta
¿Como se considera si un jugador usa las dos manos para hacer un enroque y luego presiona el reloj?	Cómo se considera si un jugador usa las dos manos para hacer un solo movimiento, por ejemplo en unenroque, captura o promoción?
¿Qué ocurre si un jugador no puede anotar?	Si un jugador is incapaz de anotar, puede proporcionar un asistente aceptable para el árbitro para recordar los movimientos. El arbitero adjustará su reloj de forma prop
¿Cómo se procede si en la fase de apuro de tiempo ninguno de los jugadores anota?	Cómo se procede si en la fase de apuro de tiempo la partida?
¿Puede un torneo restringir los acuerdos de tablas?	Puede un torneo restringir los acuerdos de tablas?
¿Cuándo la partida es tablas por triple repetición?	La partida es tablas bajo una correcta reclamación del jugador que está en jiego cuando la misma opinión va a producirse por tercer vez o ácaba de produc
¿Cuando es tablas en 50 movimientos?	La partida es tablas bajo una correcta reclamación, si en los ltimos 50 movimientos de cada jugador, no se han producido movers de peón ni capturas,
¿Qué deben hacer los jugadores que hayan finalizado su partida?	Fumar, incluidos cigarrillos electrónicos, solo está permitido en la zona del recinto de juego designada por el árbitro

Fuente: Elaboración propia

Preguntas y respuestas generadas con el modelo Phi-4

Tabla 26: Preguntas directas generadas por el modelo Phi-4

Pregunta	**Respuesta del modelo Phi-4**
¿Cómo se juega una partida de ajedrez?	Dos adversarios mueven alternativamente sus piezas sobre un tablero cuadrado.
¿Qué jugador realiza el primer movimiento en una partida de ajedrez?	El jugador con las piezas claras.
¿Cuándo se dice que un jugador está en juego?	Se dice que un jugador está en juego cuando se ha realizado el movimiento de su adversario.
¿Cuál es el objetivo de cada jugador?	Situar al rey del adversario bajo ataque.
¿Qué no está permitido respecto al rey?	Dejar el propio rey bajo ataque, exponerlo al ataque ni capturar el rey del oponente.
¿Cómo está dividido el tablero de ajedrez?	En 8x8 casillas alternadas.
¿Cómo se coloca el tablero entre los jugadores?	La casilla de la esquina derecha más cercana a cada jugador es blanca.
¿Cuantas piezas dispone cada bando?	16 piezas.
¿Cómo se denominan las hileras verticales	Ocho Pregunta: ¿Cuántas hileras horizontales hay en un tablero de ajedrez?
¿Cómo puede moverse el alfil?	A lo largo de una de las diagonales, dependiendo de si es por casillas claras o oscuras.
¿Cómo se mueve el caballo?	El caballo se mueve a una casilla más próxima, no de la misma fila, columna o diagonal.

Pregunta	**Respuesta del modelo Phi-4**
¿Qué ocurre cuando un peón llega a la fila más alejada de su posición inicial?	El peón se convierte en una dama, torre, alfil o caballo del mismo color en la casilla de llegada.
¿Como puede moverse el rey?	El rey puede moverse a cualquier casilla adyacente o enrocar.
¿Que es un movimiento legal?	Un movimiento legal es aquel que cumple todos los requisitos de movimiento.
¿Que se le debe permitir a un jugador despues de realizar su movimiento?	Detener su reloj.
¿Quién puede ajustar las piezas?	El árbitro con la ayuda de un asistente para un jugador discapacitado.
¿Que ocurre con el tiempo no consumido durante un período?	Se añade al tiempo disponible para el siguiente movimiento.
¿Qué debe verificarse inmediatamente después de una caída de bandera?	Si se completó el número mínimo de movimientos exigido dentro del tiempo asignado.
¿Qué debe hacer el jugador que desplaza una o más piezas?	Restablecer la posición correcta en su propio tiempo.
¿Que hace el arbitro para el segundo movimiento ilegal completado por el mismo jugador?	La pérdida de la partida para el infractor.

Fuente: Elaboración propia

Tabla 27: Preguntas condicionales generadas por el modelo Phi-4

Pregunta	**Respuesta del modelo Phi-4**
¿Cuando la partida es tablas?	Cuando ninguno de los jugadores puede dar mate.
¿Cuándo se considera que una pieza ataca a otra?	Cuando puede efectuar una captura o deja al rey del adversario bajo ataque.
¿Como captura una pieza el peón?	Captura una pieza capturando a un peón adversario que acaba de avanzar dos casillas desde su posición inicial.
¿Qué hace un jugador si en una promoción no tiene a su la pieza deseada?	Pide al árbitro que le proporcione la pieza y reinicie el reloj.
¿Qué jugador puede ajustar piezas?	El jugador que está en juego.
¿Qué sucede si un jugador se presenta después del tiempo de incomparecencia?	El jugador pierde la partida.
¿Una simple caída de bandera basta para declarar la derrota?	No, una simple caída de bandera no basta para declarar la derrota. Debe comprobarse la posibilidad real de mate del adversario.
¿Qué ocurre si una partida ha comienzado con los colores cambiados y ambos jugadores han hecho menos de 10 movimientos?	La partida se interrumpe y se juega una nueva con los colores correctos.

Pregunta	**Respuesta del modelo Phi-4**
¿Qué ocurre si una partida con colores invertidos termina antes de que se realicen diez movimientos?	El resultado se mantiene.
¿Qué ocurre si un jugador mueve el peón a la fila más lejana sin sustituirlo por una nueva pieza?	El movimiento es ilegal y el peón se sustituye por una dama del mismo color.
¿Si se comprueba que algunas piezas han sido desplazadas de sus casillas que se deberá hacer?	Restablecerse la posición anterior a la irregularidad.
¿Si un jugador dispone de menos de cinco minutos y no tiene tiempo adicional por movimiento está obligado a anotar?	No, no está obligado a anotar.
¿Qué deben hacer los jugadores si no hay ninguna planilla completa disponible?	Reconstituir la partida en un segundo tablero bajo el control del árbitro o un asistente.
¿Quien da el permiso para salir de la zona de juego?	El árbitro.
¿Los dispositivos electrónicos pueden almacenarse en la bolsa de los jugadores?	Sí, siempre que estén completamente apagados.
¿Como se sansiona la negativa de un jugador a cumplir las Leyes del Ajedrez?	La pérdida de la partida.

Pregunta	Respuesta del modelo Phi-4
¿Qué ocurre si un jugador realiza un movimiento sobre el tablero después de ser advertido para sellar su movimiento?	El movimiento se anota como secreto en su planilla.
¿Que ocurre con el jugador cuya anotación del movimiento secreto sea ilegal?	Pierde la partida.
¿Qué puede hacer el adversario si no está presente el jugador que selló el movimiento a la hora de reanudar la partida?	Anota la respuesta, detiene el reloj y pone en marcha el reloj del jugador ausente.
¿Cuándo puede corregirse un error en los relojes en la reanudación de la partida?	En el momento de la reanudación, si se demuestra el error.

Fuente: Elaboración propia

Tabla 28: Preguntas de razonamiento generadas por el modelo Phi-4

Pregunta	Respuesta del modelo Phi-4
¿Qué ocurre si se mueve una pieza a una casilla ocupada por otra del mismo color?	La pieza se mueve a una casilla vacía.

Pregunta	**Respuesta del modelo Phi-4**
¿Qué diferencia hay entre el movimiento de la torre y el de la dama?	La torre solo puede moverse en línea recta, mientras que la dama puede moverse en línea recta, diagonal y en forma de 'L'. La dama no puede pasar sobre ninguna otra pieza, mientras que la torre sí puede.
¿Como avanza el peón en su primer movimiento?	Dos casillas en la misma columna.
¿Cuándo se pierde definitivamente el derecho al enroque?	Si el rey ya se ha movido o si la torre con la que se quiere enrocar ya se ha movido.
¿Se puede poner en jaque el propio rey?	No, no se puede poner en jaque el propio rey.
¿Como puede terminar una partida?	Con victoria, tablas o por acuerdo entre jugadores.
¿Qué significa la caída de bandera?	La finalización del tiempo asignado a un jugador.
¿Cuando un movimiento se completa?	Cuando el jugador detiene su reloj y pone en marcha el de su adversario o al finalizar la partida o realizar el siguiente movimiento sin haber completado el anterior.
¿Cómo debe pulsar el reloj un jugador después de mover y qué sucede si usa la otra mano?	Un jugador debe pulsar su reloj con la misma mano que movió. Si usa la otra mano, no se considera una jugada ilegal, pero puede ser sancionado.

Pregunta	**Respuesta del modelo Phi-4**
¿Qué conductas relacionadas con el reloj están prohibidas?	Golpear, coger, pulsarlo antes de jugar o tirarlo. Un manejo incorrecto del reloj se penaliza. Pulsa sin movimiento.
¿Qué debe hacer el árbitro si se produce una irregularidad y las piezas deben ser reubicadas?	El árbitro debe determinar los tiempos en los relojes y ajustar el contador de movimientos según corresponda.
¿Qué se hace si se comprueba que el tablero no fue colocado correctamente?	La partida continúa, pero la posición se traslada a un tablero correctamente colocado.
¿Que ocurre si durante la partida se comprueba que se completo un movimiento ilegal?	La partida se restablece a la posición anterior a la irregularidad.
¿Como se considera si un jugador usa las dos manos para hacer un enroque y luego presiona el reloj?	Se considera como movimiento ilegal.
¿Qué ocurre si un jugador no puede anotar?	El jugador puede usar un asistente aceptable para registrar los movimientos.
¿Cómo se procede si en la fase de apuro de tiempo ninguno de los jugadores anota?	El árbitro o un asistente anotará los movimientos.
¿Puede un torneo restringir los acuerdos de tablas?	Sí, un torneo puede restringir los acuerdos de tablas.
¿Cuándo la partida es tablas por triple repetición?	Cuando la misma posición va a producirse por tercera vez o acaba de producirse por tercera vez.

Pregunta	**Respuesta del modelo Phi-4**
¿Cuando es tablas en 50 movimientos?	Si no se han realizado movimientos de peón ni capturas, o si esa situación va a producirse con un movimiento anunciado por el jugador que está en juego. Si el jugador toca una pieza, pierde el derecho a reclamar tablas en ese movimiento por triple repetición o por la regla de los 50 movimientos.
¿Qué deben hacer los jugadores que hayan finalizado su partida?	Deben abandonar la zona de juego.

Fuente: Elaboración propia

Bloque de preguntas directas

Tabla 29: Preguntas Directas

Pregunta
¿Cuando la partida es tablas?
¿Cuándo se considera que una pieza ataca a otra?
¿Cómo captura una pieza el peón?
¿Qué hace un jugador si en una promoción no tiene la pieza deseada?
¿Qué jugador puede ajustar piezas?
¿Qué sucede si un jugador llega tarde?
¿Una caída de bandera basta para declarar derrota?
¿Qué ocurre si los colores fueron invertidos?
¿Qué ocurre si la partida termina antes de detectar el error de colores?
¿Qué ocurre si un peón llega al final sin promocionar?
¿Qué se hace si las piezas fueron desplazadas?
¿Qué ocurre si un jugador tiene menos de cinco minutos?
¿Qué deben hacer los jugadores si no existe planilla?
¿Quién autoriza abandonar la zona de juego?
¿Dónde pueden almacenarse dispositivos electrónicos?
¿Cómo se sanciona incumplir las reglas del ajedrez?
¿Qué ocurre si se mueve después de sellar?
¿Qué ocurre si el movimiento sellado es ilegal?
¿Qué puede hacer el adversario si el jugador no está presente?
¿Cuándo puede corregirse un error en los relojes?

Fuente: Elaboración propia

Bloque de preguntas condicionales

Tabla 30: Preguntas Condicionales

Pregunta
¿Qué ocurre si un jugador no completa la promoción correctamente?
¿Qué pasa si un jugador realiza una jugada ilegal?
¿Qué sucede si un jugador toca una pieza sin intención de moverla?
¿Qué ocurre si el reloj no funciona correctamente durante la partida?
¿Qué pasa si un jugador abandona la partida sin avisar?
¿Qué ocurre si se detecta una irregularidad durante el juego?
¿Qué pasa si un jugador excede el tiempo permitido?
¿Qué ocurre si el árbitro detecta una posición ilegal?
¿Qué sucede si un jugador no anota sus movimientos?
¿Qué pasa si un jugador solicita tablas en una posición no válida?
¿Qué ocurre si se interrumpe la partida por causas externas?
¿Qué sucede si un jugador no respeta el orden de jugadas?
¿Qué ocurre si se realiza un movimiento con ambas manos?
¿Qué pasa si un jugador no está presente al inicio de la partida?
¿Qué ocurre si se detecta ayuda externa durante la partida?
¿Qué sucede si un jugador reclama incorrectamente una regla?
¿Qué ocurre si el árbitro interviene en una posición dudosa?
¿Qué pasa si se realiza un enroque incorrecto?
¿Qué ocurre si se mueve una pieza sin presionar el reloj?
¿Qué sucede si hay una disputa entre jugadores?

Fuente: Elaboración propia

Bloque de preguntas de razonamiento

Tabla 31: Preguntas de Razonamiento

Pregunta
Un jugador realiza una jugada ilegal y luego su bandera cae, ¿qué decisión debe tomar el árbitro?
Si un jugador está en jaque y no lo nota, pero realiza una jugada, ¿qué ocurre con la partida?
Un jugador promueve un peón pero no coloca la pieza correctamente, ¿cómo se resuelve la situación?
Si dos reglas del reglamento entran en conflicto en una situación de torneo, ¿cómo actúa el árbitro?
Un jugador realiza varias irregularidades consecutivas, ¿qué sanción corresponde?
Si un jugador tiene ventaja material pero pierde por tiempo, ¿cómo se determina el resultado?
Un jugador reclama tablas por repetición de posición, pero no es exacto, ¿qué debe hacer el árbitro?
Si un jugador realiza una jugada legal pero abandona el tablero, ¿qué ocurre?
Un jugador altera la posición accidentalmente y no recuerda la posición exacta, ¿cómo se procede?
Si hay desacuerdo entre jugadores sobre una regla del manual, ¿quién decide?
Un jugador realiza enroque ilegal en posición compleja, ¿cómo se corrige el juego?
Si un jugador usa información externa sin intención clara, ¿cómo se evalúa la situación?
Un jugador ejecuta un movimiento ambiguo, ¿cómo interpreta el árbitro la jugada?
Si un jugador detiene el reloj incorrectamente en una posición crítica, ¿qué decisión se toma?
Un jugador pierde el control del tiempo pero la posición es ganadora, ¿cómo se resuelve?
Si una partida tiene múltiples errores simultáneos, ¿cómo se prioriza la decisión arbitral?
Un jugador reclama una regla inexistente en el manual, ¿cómo se procede?
Si el reglamento no cubre una situación específica, ¿qué criterio aplica el árbitro?
Un jugador genera confusión en la secuencia de movimientos, ¿cómo se reconstruye la partida?
Si un jugador intenta modificar una jugada pasada, ¿qué decisión se toma?

Fuente: Elaboración propia

Enlaces a los repositorios completos de todos los modelos y al conjunto de datos.

- Para el modelo BERT:

  `https://colab.research.google.com/drive/1uJnuppqkhVroLXRisD8jnOna8SEkXxaG`

- Para el modelo google/flan-t5-small:

  `https://colab.research.google.com/drive/1oLqqt9xL9Pq0EHVKDL6kDyFTKme27nhu#scrollTo=G-FlL2PfzDHw`

- Para el modelo microsoft/Phi-4-mini-instruct:

  `https://colab.research.google.com/drive/1H3WppS3E-2N9D1Nqd9es4N5gGmwNTIiP#scrollTo=Cfps1fkokEA1`

- Conjunto de datos:

  `https://colab.research.google.com/drive/178Ccbm6HrWV9Otxe6KYPgxZ4bDmwjDwm#scrollTo=Evby-uaHv_2d`