Universidad ORT Uruguay
Facultad de Ingeniería

Estudio de cumplimiento ambiental aumentado por LLM y RAG

Entregado como requisito para la obtención del título de Master en Big Data

Mariano Chicatun - 301379

Tutor: Sergio Yovine

2024

Yo, Mariano Chicatun declaro que el trabajo que se presenta en esta obra es de mi propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el Proyecto Final de Master en Big Data;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

10-09-2024

Dedicatoria

Este trabajo está dedicado con todo mi cariño a mi familia, quienes siempre me han brindado su amor y apoyo incondicional en cada paso de mi vida. A mis amigos, por su compañerismo y constante motivación durante este proceso, y a todas las personas que de alguna manera contribuyeron a mi crecimiento personal y académico. Su presencia ha sido fundamental en este viaje.

Agradecimientos

Agradezco a mi universidad, que me ha dio el espacio y las herramientas para desarrollar este trabajo, así como a mis compañeros, por su colaboración, apoyo y las experiencias compartidas a lo largo de esta etapa. Principalmente, quiero expresar mi sincero agradecimiento a mi director de tesis, Sergio Yovine, por su guía, paciencia y conocimientos, sin los cuales este proyecto no habría sido posible. Su dedicación y asesoría fueron esenciales para la realización de este trabajo.

Abstract

El objetivo de este trabajo es optimizar la validación del cumplimiento ambiental en grandes empresas de sectores como minería, petróleo, alimentación, y papelera. Estas industrias enfrentan desafíos significativos para verificar el cumplimiento de leyes y estándares ambientales, ya que el proceso es lento y demanda mucho tiempo.

Este estudio propone utilizar técnicas de Inteligencia Artificial y Machine Learning para agilizar y mejorar este proceso. Con los avances recientes en modelos de lenguaje de gran escala (LLM), técnicas como la Generación Aumentada por Recuperación (RAG) han demostrado ser eficaces para abordar problemas complejos relacionados con datos textuales no estructurados.

Se exploran técnicas avanzadas de RAG, incluyendo variantes como Graph RAG y Hybrid RAG, y esquemas multiagente secuenciales y jerárquicos. Para implementar estos agentes, se utilizan frameworks como Langchain, LlamaIndex y LangGraph.

Graph RAG está ganando relevancia en aplicaciones prácticas y científicas. Nuevas implementaciones, como Microsoft GraphRAG [1], buscan superar las limitaciones de RAG tradicional, que a menudo pierde conexiones entre diferentes partes de la información y carece de una comprensión holística de los datos. Graph RAG utiliza grafos para vincular información relevante y nodos de resumen que sintetizan temas similares, permitiendo responder preguntas generales y holísticas sobre los documentos.

Se diseña una Prueba de Concepto (PoC) basada en una arquitectura de microservicios con tres componentes principales: un constructor de grafos, un generador de preguntas, y un sistema de QA (Question Answering). Estos componentes son modulares y pueden modificarse, mejorarse, escalarse y ejecutarse de manera independiente, lo que facilita el desacoplamiento del sistema.

El constructor de grafos procesa la documentación normativa del proyecto y extrae los requisitos necesarios para garantizar el cumplimiento ambiental.

El proceso de generación de preguntas, inspirado en el concepto de Chain-of-Questions [2], genera preguntas que cubren exhaustivamente el conjunto de datos para minimizar y acotar el espacio de búsqueda dentro de la documentación del proyecto en estudio.

Finalmente, el sistema de QA genera respuestas basadas en estas preguntas, utilizando la documentación del proyecto en evaluación para determinar el cumplimiento ambiental.

Palabras clave

LLM; RAG; GraphRAG; Cumplimiento Ambiental;

Índice general

1 Introducción

1.1 Introducción

El presente trabajo busca mejorar el proceso de validación del cumplimiento ambiental para grandes empresas en rubros como la industria de grandes capitales. Empresas mineras, petróleo y gas, alimentarias, papeleras, etc. tienen grandes complicaciones a la hora de saber si están incumpliendo leyes y estándares ambientales. El proceso de validar el cumplimiento ambiental es lento y consume mucho tiempo.

El proceso de validación de cumplimiento ambiental empieza por la selección del estándar que tiene las reglas con las que se tiene que cumplir, para este trabajo se selecciona el estándar Environmental and Social Framework (ESF). Luego el experto en la materia recupera toda la documentación existente relacionada con el proyecto que está analizando y luego de un estudio profundo tanto del estándar como del proyecto, y en caso de encontrar gaps de información, crea un listado de preguntas para obtener más información y complementar los documentos. En muchos casos esto termina con una visita a sitio para hacer entrevistas a los stakeholders y recabar más información.

El objetivo principal de este trabajo es utilizar técnicas de Inteligencia Artificial y Machine Learning para facilitar y mejorar este proceso. Con la introducción y la mejora de los grandes modelos de lenguaje (Large Language Models - LLM), técnicas como Retrieval Augmented Generation (Retrieval Augmented Generation - RAG) han vuelto posible resolver problemas complejos sobre datos de texto no estructurados.

En este trabajo se exploran técnicas avanzadas de Retrieval Augmented Generation (RAG), esquemas multiagente como Chain of Thoughts (CoT), Mixture of Experts (MoE), e implementación de esquemas multiagente con nuevos frameworks como LangGraph, CrewAI and LlamaIndex.

Estos frameworks apunta a abstraer parte de la complejidad del uso de los LLM y técnicas como RAG, CoT, etc. y modularizar componentes para poder intercambiarlos fácilmente entre distintos modelos, bases de datos vectoriales o híbridas, la carga de distintos tipos de documentos y su procesamiento. Dado el creciente ritmo de avance por el interés que genera la automatización con LLMs, las librerías sufren cambios constantes y generan profundos problemas de manejo de dependencias y mantenimiento de una base de código para el desarrollo de aplicaciones. Se espera que con el tiempo las librerías tiendan a la estacionareidad y con ello mejore la experiencia de desarrollo de aplicaciones.

Retrieval Augmented Generation (RAG) es una técnica que permite a un modelo de lenguaje (LLM) acceder a información contextual, como documentos o datos privados, al responder preguntas del usuario, lo que aumenta las probabilidades de generar una respuesta más precisa y relevante. El RAG estándar (Baseline RAG) emplea representaciones vectoriales y algoritmos de búsqueda vectorial, como BM25, para recuperar información contextual relevante cada vez que se formula una pregunta.

Durante el desarrollo de este trabajo se introdujo y comenzó a tomar relevancia en la comunidad científica y de los desarrolladores, una nueva técnica de extracción de información, como también de retrieval llamada GraphRAG, o Graph Retrieval Augmented Generation. Esta técnica permite extraer nodos, entidades y relaciones de un documento de forma automática, o de forma estructurada a través de proveer un esquema definido. Una vez construido este grafo se implementan técnicas de retrieval de grafos que permiten recuperar las entidades y relaciones relevantes al momento de responder una pregunta. Comparado con un Baseline RAG permite obtener una visión más holística de la información de los documentos que se están procesando y captura mucho mejor las relaciones subyacentes en la data de los documentos de contexto vinculando los puntos de otra manera "sueltos"que tiene Baseline RAG, que no guarda ningún tipo de relación entre las partes de los documentos más allá del nombre del documento y página.

La evaluación y obtención de métricas para medir el rendimiento de una aplicación con LLMs y RAG es una de las partes más importantes cualquier aplicación RAG, pero a su vez una de las más complejas. La revisión manual y de un experto continúa siendo la mejor herramienta para la evaluación de una aplicación de LLMs, pero al ser lenta y costosa continuamente se buscan nuevas técnicas que mejoren este proceso de evaluación. Distintas técnicas como el uso de comparaciones de distancias de las representaciones vectoriales de texto, el uso de LLM-as-a-judge que utiliza a un LLM con una salida estructurada para validar las respuestas de un RAG comparándolo contra una "ground truth", etc. han surgido para automatizar este proceso.

El avance permanente de este campo, tanto en el entrenamiento y aplicación de nuevos modelos continuamente, como avances en las técnicas como RAG o GraphRAG abren nuevas posibilidades continuamente, aunque con un gran costo asociado de tiempo de desarrollo y estudio para estar al día con las tecnologías, dada la inmadurez del campo del desarrollo de aplicaciones con LLM y RAG.

1.2 Objetivos

El objetivo principal de este trabajo es explorar técnicas de Machine Learning e Inteligencia Artificial que posibiliten un aumento en la productividad y en la velocidad de análisis de los expertos en estudios ambientales, con el objetivo de ayudarlos a generar un impacto positivo en como las empresas realizan hoy sus estudios ambientales. La implementación de un PoC (Prueba de Concepto) volcado en un pipeline end-to-end para la aplicación de LLMs y RAG en estudios ambientales es novedosa y trae consigo varios posibles caminos futuros de desarrollo a otros campos como la abogacía, la seguridad de los trabajadores y hasta la contaduría, dada la similitud en la naturaleza de la información.

El cumplimiento ambiental es de suma importancia para cualquier empresa de tamaño mediano, y su importancia es aún mayor con el tamaño de las operaciones. Industrias como la minería, Oil & Gas, alimentación, papeleras, etc. generan grandes impactos ambientales, con justa razón los gobiernos continuamente buscan aumentar el control añadiendo más regulación ambiental. Esto genera grandes cuellos de botella a la hora de empezar un proyecto, teniendo que hacer estudios ambientales caros y lentos para poder dar una respuesta a las crecientes obligaciones.

1.3 Enfoque de la solución

1.3.1 Descripción de alto nivel

Con el objetivo de poder correlacionar dos tipos de documentos, el estándar ambiental y la documentación ambiental del proyecto bajo estudio, se propone la siguiente solución de alto nivel: 1. Procesamiento y extracción de la información respectiva de un estándar ambiental (se utilizará el ES Framework previamente mencionado) en una Knowledge Base. 2. Generación de preguntas pertinentes al proyecto bajo estudio (contemplando tipo de industria, proceso y particularidades) a partir de la Knowledge Base. Las preguntas tratan de forma más atómica

cada uno de los requerimientos planteados dentro del estándar. 3. Una a una se utilizan las preguntas generadas para interrogar la documentación del proyecto y generar respuestas a cada una de las preguntas. En los casos donde la información recuperada del proyecto no sea suficiente para responder si está en cumplimiento o no, se instruye al modelo de lenguaje que genere preguntas extra para que el experto recabe más información.

1.3.2 Descripción técnica de la solución final

El proyecto se divide en 3 microservicios con el objetivo de poder iterar, mejorar y evaluar de forma independiente en cada uno de ellos. Cada uno de estos microservicios se despliega en Python en forma de API REST utilizando el framework FastAPI.

Se utiliza el framework Llamaindex para el desarrollo de la aplicación, debido a su avanzada implementación de GraphRAG llamada Property Graph.

Los 3 microservicios son los siguientes: 1. 'graph-generation': Procesa el estándar ambiental transformando la información relevante en un grafo de entidades y relaciones. Extrae los requerimientos de cada una de las partes mencionadas en el ES Framework. La carga y extracción de los documentos, así como la generación del diagrama de grafos, se realiza con LlamaIndex. Para esta prueba de concepto no se implementa una base de datos de grafos para persistir el grafo generado, pero la implementación utilizando Neo4j, Nebula, o algunas otras bases de datos de grafos está contemplada en esta librería. De las dos etapas de GraphRAG este microservicio representa la ingesta de la información en la Knowledge Base, persistida de forma local. 2. 'question-generator': Genera una lista de preguntas en función de los requerimientos extraídos del estándar ambiental en el paso anterior. Recibe los nodos, entidades y relaciones en el paso anterior y genera una lista de preguntas que pretende cubrir cada uno de los requerimientos planteados por el estándar. Implementado utilizando LlamaIndex y Property Graph, este microservicio implementa la etapa de retrieval de GraphRAG. 3. 'project-compliance': Para cada una de las preguntas obtenidas en el paso anterior, se interroga a la documentación del proyecto procesada como un Baseline RAG y retrieval utilizando la comparación de representaciones vectoriales con el algoritmo BM25 en una base de datos vectorial persistida de forma local.

Una de las ventajas de esta implementación es que cada etapa podría ser supervisada y mejorada por un experto de forma relativamente simple e iterativa, ya que para grafos de tamaño pequeño a mediano, se pueden realizar inspecciones

visuales de los gráficos de entidad relación extraídos por el modelo de lenguaje, como así también inspeccionar las listas de preguntas generadas en la segunda etapa. La tercera etapa puede mejorarse y validarse de forma continua con frameworks como RAGAS que permite evaluar las aplicaciones RAG.

1.3.3 Evaluación

Para poder medir el rendimiento del sistema a la hora de resolver el problema, es necesario desarrollar un pipeline de evaluación. La evaluación de sistemas RAG o de la generación de LLM está en constante cambio y evolución en este momento. Algunas técnicas como comparación vectorial de las representaciones vectoriales del contenido o el uso de LLM-as-judge son las más usuales, siendo aún la evaluación manual y experta de las respuestas la técnica más fiable. En este trabajo, se utilizan dos pipelines de evaluación, uno para la generación de preguntas y otro para la generación de respuestas.

En el generador de preguntas se utiliza la distancia vectorial de los embeddings entre las preguntas generadas por el sistema y las generadas por un experto (en adelante, el SME - Subject Matter Expert) en la materia que participo en este proyecto, para determinar si las preguntas generadas cubren en la totalidad del estándar bajo estudio. Es decir, se realiza una comparación entre las preguntas generadas por el sistema y las generadas por un experto, para determinar el çoverage"del estándar. El estudio o evaluación de las preguntas generadas por un LLM, es distinto al canal normal de investigación, que se enfoca en la evaluación de respuestas generadas, por lo que se adaptó una de las técnicas que se utilizan para evaluar respuestas de sistemas RAG.

En el generador de respuestas se utilizó un framework de evaluación de respuestas denominado RAGAS, que separa en dos la evaluación: evaluando la parte de generación, más orientada a evaluar la generación de los LLM, y la evaluación del RAG, orientado a evaluar la recuperación de los documentos.

2 Marco Teórico

2.1 Large Language Models - LLMs

Los models de lenguaje están diseñados para comprender y generar texto en lenguaje natural, se entrenan en grandes cantidades de datos textuales para aprender patrones, estructuras y reglas del lenguaje, para predecir o generar texto coherente y comprensible.

La mayoría de los modelos de lenguaje hoy están diseñados con la arquitectura de Transformers que se detalla en la siguiente sección.

2.1.1 Arquitectura de Transformers

La arquitectura de Transformers es un modelo de deep learning que esta revolucionando el campo, y principalmente, las tareas de Natural Language Processing (NLP). Algunos otros campos de aplicación de Deep Learning como Time Series Forecasting, Computer Vision, Audio Processing y Video Processing también se han visto beneficiadas con esta nueva arquitectura.

La arquitectura de Transformers introducida por [3] se ha convertido en la columna vertebral de muchas aplicaciones de NLP como la traducción de idiomas, la generación de texto, código, etc. y LLMs como GPT y BERT.

La principal diferencia con otras arquitecturas de procesamiento de secuencias como Redes Neuronales Recurrentes (RNN) y Long-Short-Term-Memory (LSTM) es que procesan todos los tokens en simultáneo, y no de forma secuencial. Esto permite una paralelización y eficiencia mayor.

Los componentes principales de una arquitectura de transformers se observan en la figura 2.1.

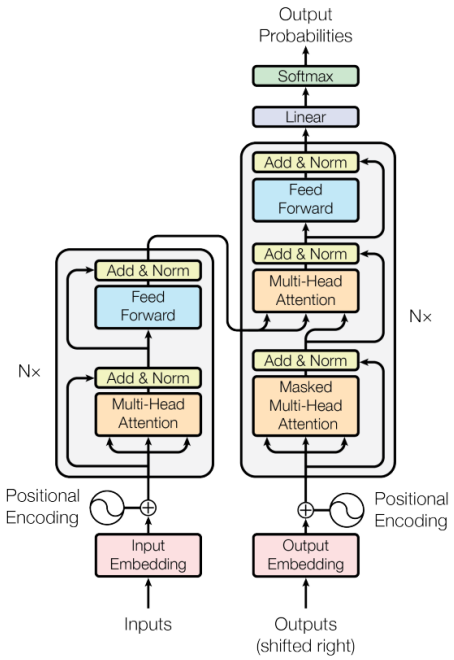

Figura 2.1: Arquitectura de Transformers

2.1.1.1 Encoder

El encoder toma una secuencia de entrada (que pueden ser palabras, los datos de una serie temporal o cualquier tipo de data secuencial) y produce una representación vectorial o embedding de él, que el Decoder puede utilizar para generar una salida. El encoder varias campas de encoder, compuestas de las siguientes sub capas.

Multi-Head Attention

Esto permite al modelo atender a distintas partes de la secuencia de entrada de forma simultánea. Utiliza múltiples "heads"(capas de atención paralelas) para aprender diferentes representaciones de la data de entrada.

Calcula una suma ponderada de los **Value**s, con los pesos siendo determinados porque tan compatible es la **Query** con una **Key**, esto es, cuan relevantes son distintas partes de la secuencia de entrada entre sí. La **Query**, **Key** y **Value** son derivadas de los embeddings de entrada.

Feed Forward Neural Network

Es una red neuronal Fully connected aplicada a cada posición (token) en la secuencia independiente. Se aplica según la posición, para evitar mezclar información entre tokens, si no que opera en la representación individual de cada token.

Típicamente, consiste de dos transformaciones lineales con una función de activación ReLU en el medio, que permite al modelo aprender mappings no lineales en la data de entrada.

2.1.1.2 Decoder

Utiliza la representación de salida del Encoder 2.1.1.1 y genera una secuencia de salida (por ejemplo, el texto traducido, o la siguiente palabra en la generación). El decoder también consiste de múltiples capas de decoder, con las siguientes subcapas:

Masked Multi-Head Attention

Este mecanismo de atención asegura que cuando se genera un token de salida en la posición `i`, el modelo solo tiene información de los tokens anteriores, y no de los futuros. Esto se hace enmascarando (haciendo 0) los tokens futuros, para prevenir el Leakage de información.

Multi-Head Attention

Cumple la misma función que 2.1.1.1.1, pero teniendo como entrada las representaciones de salida del encoder. Esto permite al decoder enfocarse en las partes relevantes de la secuencia de entrada mientras se genera la salida.

Feed Forward Neural Network

En el encoder y en el decoder sirven el mismo propósito, expuesto previamente en 2.1.1.1.2.

2.1.1.3 Attention Mechanism

Esta es la innovación principal de la arquitectura de transformers y permite al modelo enfocarse en partes específicas de la entrada cuando se procesa cada token.

Tiene dos sub capas:

Self Attention

En el encoder permite a cada token de la entrada atender a todos los otros tokens, haciendo posible capturar las relaciones entre palabras en una secuencia de forma más efectiva.

Ejemplo

En la sentencia "The cat sat on the mat", el mecanismo self-attention permite al modelo entender que çat.ªtá más relacionado con "sat"mientras que le preste menos atención a otras palabras como "the.º "mat".

Encoder-Decoder Attention

En el Decoder, la atención se aplica entre la salida del Encoder 2.1.1.1 y el estado actual del Decoder 2.1.1.2. Esto permite al decoder enfocarse en las partes relevantes de la entrada cuando se genera cada token de salida.

Ejemplo

Si la secuencia de entrada es en francés y la secuencia de salida en inglés, el mecanismo encoder-decoder attention ayuda al modelo a saber que partes de la entrada en francés enfocarse cuando se genera cada palabra en inglés.

2.1.1.4 Positional Encoding

Un modelo implementado con la arquitectura de transformers no entiende necesariamente el orden de los tokens porque los procesa en simultáneo (en paralelo). Para sobreponerse a esta limitante, esta arquitectura introduce el **Positional Encoding**.

Este mecanismo introduce una señal a los embeddings de los tokens que da a cada token información sobre su posición en la secuencia. Esta información se añade a los embeddings de entrada antes de pasarse al encoder o decoder.

Típicamente, utiliza funciones seno o coseno de diferentes frecuencias para codificar la información posicional, resultando en una señal suave y continua que el modelo puede utilizar para diferenciar entre las posiciones de los tokens.

2.1.2 Ventanas de Contexto en LLMs

La ventana de contexto es un aspecto clave de los modelos de lenguaje, ya que define la cantidad de texto que el modelo puede considerar al generar respuestas. Se considera una "memoria de trabajo"que permite al modelo mantener coherencia y contexto durante interacciones largas. Una ventana de contexto más grande le permite procesar indicaciones más complejas, mientras que una más pequeña puede limitar la capacidad que tienen de mantener conversaciones extensas de forma efectiva.

Los LLMs basados en la arquitectura de Transformers utilizan el mecanismo de atención para procesar y generar texto, permitiendo al modelo asignar importancia a diferentes palabras en una secuencia. Sin embargo, están limitados por la ventana de contexto, que define cuántos tokens pueden considerar a la vez. Una ventana de contexto pequeña puede restringir la capacidad del modelo para manejar documentos extensos o mantener coherencia en interacciones prolongadas, lo que lleva a problemas como:

- Alucinaciones: Generación de información incorrecta o inventada.
- Conocimiento desactualizado: Dependencia de datos de entrenamiento estáticos, lo que puede causar respuestas basadas en información obsoleta.
- Faltas en dominios específicos: Carencia de conocimientos especializados para responder con precisión en áreas específicas.

Aunque la ventana de contexto y el mecanismo de atención son esenciales para la coherencia y comprensión en los LLMs, también limitan su capacidad para manejar interacciones largas, información actualizada y dominios específicos. Esto se debe a que a medida que aumenta el rango contextual, los valores de atención

tienden a volverse más pequeños y uniformes, lo que reduce la capacidad del modelo para capturar dependencias a largo plazo. También, la complejidad computacional del mecanismo de Self-Attention 2.1.1.3.1 aumenta con la longitud de la secuencia, lo que lo hace ineficiente para manejar textos muy largos.

2.1.2.1 Importancia de la Ventana de Contexto

El aumento y mejora de la ventana de contexto en LLMs, tiene los siguientes beneficios:

- Comprensión: Una ventana de contexto más amplia permite a los LLMs captar detalles y matices en el texto, generando respuestas más precisas y relevantes.
- Coherencia: Al poder hacer referencia a más texto previo, el modelo puede mantener una narrativa o argumento coherente a lo largo de la conversación.
- Prompts Complejos: Modelos con ventanas de contexto extensas pueden abordar preguntas sobre varios temas en simultáneo que requieren entender intercambios previos.

2.1.2.2 Limitaciones de memoria de largo plazo

La naturaleza sin estado o stateless de los LLM limita su capacidad para retener y recuperar información de interacciones pasadas o del contexto proporcionado. Esto afecta la capacidad de mantener coherencia y generar respuestas personalizadas en diálogos de múltiples turnos.

Se han propuesto varios enfoques para resolver estos desafíos:

Mecanismos de memoria externa

Integrar redes de memoria o mecanismos de atención con una memoria externa puede mejorar la capacidad del modelo para actualizar y acceder a la información a lo largo de interacciones.

Métodos a nivel de modelo

Entrenar a los LLMs con textos de mayor longitud puede ser efectivo, pero implica costos de entrenamiento elevados y el riesgo de perder detalles importantes en contextos largos.

Enfoques a nivel de agente

Transformar el texto en árboles o páginas puede ayudar a manejar el contexto, pero estos métodos a menudo no capturan dependencias a largo plazo, limitando su efectividad en secuencias muy largas.

Retrieval-Augmented Generation (RAG)

RAG permite atacar las limitaciones de contexto de los LLMs integrando mecanismos de retrieval de información externa. En lugar de depender exclusivamente del contexto disponible en la ventana de contexto del modelo, RAG permite buscar y recuperar información relevante desde bases de datos o documentos externos en tiempo real, con el objetivo de mejorar la coherencia y precisión de las respuestas, especialmente en tareas de diálogos largos o consultas complejas.

2.1.2.3 Ventana de Contexto Afirmada vs Efectiva

La ventana de contexto efectiva se refiere a la máxima longitud en la que un LLM puede superar a un modelo de baseline fuerte.

Según muchos estudios actuales se ha demostrado que se sobrestimó la longitud efectiva de contexto en muchos modelos. Por ejemplo, GPT-4 afirma tener una ventana de 128K [4], pero la longitud efectiva es de 64K según [5].

Se han desarrollado varias pruebas o test para el benchmarking de la ventana de contexto efectiva:

Pruebas de Needle-in-a-Haystack (NIAH)

Este método se utiliza para evaluar la capacidad de los LLMs para recuperar información específica dentro de grandes volúmenes de texto irrelevante. Los resultados de estas pruebas indican que la recuperación simultánea de información relevante y el razonamiento complejo son tareas difíciles para los LLMs actuales.

Ideales en el procesamiento de grandes ventanas de contexto

- Capacidad de Needle-in-a-Haystack: Los LLMs deben ser capaces de encontrar información necesaria en contextos extensos.
- Capacidad de razonamiento en contextos largos: Los LLMs deberían ser capaces de hacer inferencias complejas a partir de entradas crudas en un solo paso.

2.1.3 Embeddings o representaciones vectoriales

Las representaciones vectoriales, comúnmente denominados embeddings, son una técnica clave en el campo del NLP y otras aplicaciones de Inteligencia Artificial. Los embeddings transforman elementos como las palabras, frases o hasta imágenes en vectores numéricos, haciendo que sea más fácil procesarlos y entender las relaciones entre ellos.

Son fundamentales para permitir que las máquinas procesen, interpreten y razonen de forma eficiente sobre datos no-numéricos, capturando relaciones complejas entre la entrada y mejorando la performance en muchas tareas de IA.

2.1.3.1 Como funcionan los Embeddings

Los embeddings convierte data no numérica en una representación matemática de múltiples dimensiones. Por ejemplo, para el caso de modelos de lenguaje, cada palabra o frase se transforma en un vector numérico (valores de punto flotante). La idea es que las palabras sean similares o tengan significados similares estén más cerca entre sí en este espacio vectorial, mientras que palabras no relacionadas estén más alejadas.

Un ejemplo clásico es palabras como rey.º reina"deberían estar más cerca entre sí en este espacio vectorial, y la diferencia entre "hombrez "mujer"debería ser similar a la diferencia entre reyz reina".

2.1.3.2 Ventajas de los Embeddings

Algunas ventajas del uso de embeddings:

- **Reducción de la dimensionalidad**: Los embeddings permite que múltiples palabras se representen en un espacio dimensional reducido, donde las relaciones semánticas entre las palabras son capturadas.
- **Capturar relaciones semánticas**: Los embeddings permiten que el model capture relaciones semánticas complejas como sinónimos o antónimos entre palabras que aparecen normalmente en contextos similares.
- **Eficiencia**: Al representar data compleja como vectores numéricos, las tareas de procesamiento (como búsqueda, clasificación o generación de texto) se vuelven mucho más rápidas y eficientes, ya que las operaciones matemáticas entre vectores que se computan de forma eficiente.

2.1.3.3 Aplicaciones

- Búsqueda semántica: Los embeddings permiten buscar no solo coincidencias exactas, sino también significados o relaciones semánticas, mejorando significativamente la precisión de las consultas.
- Clasificación de texto: Al transformar el texto en vectores, los algoritmos de clasificación pueden detectar patrones con mayor facilidad.
- Recuperación de información: En sistemas como RAG, los embeddings permiten indexar grandes cantidades de datos y recuperar información relevante comparando vectores.

2.2 Retrieval Augmented Generation - RAG

Retrieval Augmented Generation (RAG) es una técnica que provee a un LLM con información contextual de documentos o información que puede ser privada al momento de contestar una pregunta del usuario, aumentando las probabilidades de llegar a una mejor respuesta al proveerle información de contexto relevante.

Esta técnica se ha vuelto de suma importancia debido a varias razones clave:

- **Limitaciones de la ventana de contexto**: Los LLMs tienen limitaciones a la hora de procesar grandes volúmenes de información debido a su ventana de contexto limitada. RAG supera esta limitación recuperando información relevante de bases de datos externas sin depender del tamaño del contexto.
- **Acceso a datasets privados**: RAG permite a los LLMs utilizar datos privados y propietarios de manera segura, integrando información específica de la organización o dominio para aumentar la precisión de las respuestas.
- **Mejora en precisión y relevancia**: RAG enriquece las respuestas al recuperar información de fuentes actualizadas o especializadas, ideal cuando los datos públicos son insuficientes o están desactualizados.
- **Actualización dinámica**: RAG permite la actualización continua de fuentes de datos en tiempo real, manteniendo el sistema siempre actualizado sin necesidad de reentrenamiento.
- **Reducción de costos computacionales**: RAG reduce los costos al acceder de manera selectiva a información relevante, evitando la necesidad de ampliar el contexto del modelo y optimizando los recursos.

2.2.1 Componentes de RAG

Un sistema RAG se divide en 3 componentes principales:

1. Ingestión: Limpiar, dividir, generar representaciones vectoriales y cargar la data en una Base de Datos de Conocimiento (Knowledge Base).
2. Retrieval: Hacer consultas a la Knowledge Base para obtener la información pertinente a la pregunta.
3. Generación: Pasar el contexto recuperado a un prompt y pasárselo al LLM.

2.2.2 Baseline RAG

Baseline RAG utiliza representaciones vectoriales que utilizando alguno de los muchos algoritmos de búsqueda vectorial (BM25, Contriever, LLM-Embedder) para recuperar la información de contexto relevante a la pregunta del usuario, cada vez que esto ocurre. La ingesta de documentos de Baseline RAG en general sigue el siguiente proceso:

2.2.2.1 Ingestion

1. Carga de los distintos documentos posibles en memoria, estos pueden ser pdf, txt, audios, imágenes, etc.

2. En caso de ser necesario, procesamiento de estos documentos, como limpieza de caracteres irrelevantes, formato de los títulos, tablas, etc.

3. Separación de los documentos en partes más pequeñas (comúnmente denominados chunks) que permiten generar representaciones vectoriales o embeddings más representativos del texto y no llenar la ventana de contexto del LLM al momento del retrieval.

4. Generación de representaciones vectoriales o embeddings de las partes de los documentos. Existen grandes cantidades de modelos para representar vectorialmente un texto, con distinto rendimiento en función de la aplicación específica y el tamaño de la representación.

5. Almacenamiento o persistencia de estas representaciones vectoriales como parte de la creación de Knowledge base del agente.

2.2.2.2 Retrieval

Al momento de la recuperación, el proceso es el siguiente:

1. El usuario hace una pregunta o consulta en el sistema.

2. Se genera una representación vectorial con el mismo modelo que genero las representaciones de la Knowledge base.

3. Usando el algoritmo seleccionado de búsqueda vectorial, se compara la representación vectorial de la pregunta contra los vectores de la Knowledge base, y se selecciona los documentos que menos distancia vectorial tengan a la pregunta del usuario.

4. Los documentos seleccionados se pasan como contexto al LLM en conjunto con el prompt base que indica el comportamiento base del LLM. El LLM genera la respuesta final que es finalmente entregada al usuario.

2.2.3 Técnicas RAG avanzadas

Existen 3 tipos de técnicas avanzadas de RAG:

1. Optimización pre-retrieval: Modificar la creación de las particiones del documento.

2. Optimización del retrieval: Mejorar las queries a la DB vectorial.

3. Optimización post-retrieval: Procesar las particiones para filtrar el ruido.

La parte de generación puede mejorarse con fine-tuning o con prompt engineering.

2.2.3.1 Pre-Retrieval

Query Expansion

Esta técnica es una variante de Baseline RAG. El proceso que siguen es muy similar, con la diferencia de que en esta técnica, se utiliza a un LLM para generar n preguntas sobre la pregunta original, con el objetivo de hacer preguntas más específicas que aumenten la probabilidad de recuperar los documentos en el proceso de retrieval. El separar una pregunta o consulta del usuario en varias, que puede ser compleja, o compuesta de varios temas, una vez transformados a su representación vectorial, permiten una búsqueda más certera en la base de datos vectorial, al dirigirse a distintas partes del espacio de embeddings que también son relevantes a la pregunta inicial.

La creación de las n preguntas puede realizarse de forma muy sencilla con un prompt zero-shot a un LLM bien detallado, como el siguiente:

AI

Sos un asistente de AI, tu tarea es generar tres diferentes versiones de la pregunta del usuario para recuperar los documentos relevantes de una base de datos vectorial, tu objetivo es ayuda al usuario a superar algunas limitaciones de la búsqueda en bases de datos vectoriales.
Proporciona estas preguntas alternativas separadas por "\"
Pregunta del usuario:

Ejemplo:

1. **Consulta del usuario** Estoy haciendo una biografía de Einstein y necesito saber donde nación, cuál era su nacionalidad y cuando murió.
2. **Query Expansion**: Genera tres preguntas:
   a) Donde nació Einstein?
   b) De qué nacionalidad era Einstein?
   c) Cuando murió Einstein?
3. **Embeddings**: El model de embeddings genera la representación vectorial de cada una de las preguntas.
4. **Retrieval** Para cada una de las preguntas, se hace la búsqueda vectorial comparando con los vectores almacenados en la Vector Database.

Self-Query

Se utiliza al LLM para extraer tags dentro de la consulta del usuario y se los utiliza con el embedding de la query. El LLM extrae varios campos de metadata que sean relevantes para el caso de uso de negocio, por ejemplo tags, IDs, artículos, leyes, nombres, etc.

Usando zero-shot prompt engineering se puede llegar a buenos resultados, para extracciones simples.

Ejemplo:

AI

Eres un asistente de modelo de lenguaje de IA.
Tu tarea es extraer información de una pregunta del usuario. La información que se debe extraer es el número de ley mencionada en el texto. Tu respuesta debe consistir únicamente en número de ley mencionada, nada más.
Pregunta del usuario:

Hybrid RAG

Esta técnica combina la técnica de búsqueda vectorial con una o más estrategias de búsqueda complementarias, lo que ayuda con la búsqueda de palabras exactas (keyword search). La estrategia más usual es la combinación de la búsqueda por keywords y la búsqueda vectorial. Esta relación puede darse a través de un parámetro prefijado, de la siguiente forma:

$$hybrid_{score} = (1 - \alpha) * keywords_{score} + \alpha * vector_{score}$$

α es el parámetro predefinido que puede optimizarse en función de los resultados, toma valores entre 0 y 1, por lo que: - $\alpha = 0$: Búsqueda por Keywords. - $\alpha = 1$ Búsqueda Vectorial.

2.2.3.2 Post-Retrieval

Reranking

Esta técnica funciona, se suele utilizar en conjunto con Query Expansion, que genera n prompts que a su vez devuelven k documentos, por lo que en total tenemos $n*k$ particiones de documentos. Rerank se utiliza para ordenar los $n*k$ documentos basados en la relevancia relativa a la pregunta inicial, de más relevante a menos relevante, y se elegirán las k particiones más relevantes.

El flujo es el siguiente: 1. Buscar las $n * k$ particiones 2. Reordenar utilizando rerank 3. Guardar solo los top k

Para hacer reranking, hay varias técnicas (sentence-transformers), o usar un LLM + Prompt Engineering como reranker.

AI

Eres un asistente de modelo de lenguaje de IA.
Tu tarea es reordenar los pasajes relacionados con una consulta según su relevancia. Los pasajes más relevantes deben colocarse al principio.
Solo debes seleccionar un máximo de k pasajes.
Los siguientes son pasajes relacionados con esta consulta: pregunta.
Pasajes:

2.2.4 GraphRAG

Construye un entendimiento semántico sobre todo el conjunto de documentos o datos, para que cuando hacemos consultas, el Knowledge Graph guía el descubrimiento de contenido para proporcionar al LLM contenido relevante sobre el tópico y de manera comprehensiva al nivel correcto de abstracción.

2.2.4.1 Knowledge Graph

Son representaciones que se componen de triplas (sujeto, predicado, objeto), pero que no son tan expresivas o representativas como podrían ser. No tiene la habilidad de: 1. Asignar etiquetas y propiedades a los nodos y las relaciones. 2. Representar nodos de texto con sus representaciones vectoriales (embeddings). 3. Realizar retrieval vectorial y simbólico.

2.2.4.2 Property Graph

Es una colección de conocimiento de nodos etiquetados (por ejemplo, categorías, etiquetas de texto, etc.) con propiedades (metadata) vinculados juntos por relaciones en caminos estructurados. Pueden utilizar representaciones vectoriales en un Vector Store funcionando "por debajo"de la Property Graph.

Funcionan de la siguiente forma:

- Categorizan nodos y relaciones en tipos con metadata asociada.
- Tratan al grafo como un superset de una base de datos vectorial para una búsqueda híbrida.
- Para queries más complejas se podrían utilizar Cypher queries.

Construcción del Grafo

1. Guiado por un esquema

El usuario define los tipos de entidades, los tipos de relaciones y sus conexiones en el esquema. El LLM solo completo con data estas relaciones y entidades

2. Extracción implícita

Se utiliza al LLM para construir relaciones entre los nodos de la data. Puede ser basado en la estructura del documento, o del nodo específico, etc.

3. Extracción libre

Se utiliza al LLM para inferir las entidades, tipos de relaciones y el Schema directamente de la data en una forma libre.

Retrieval del Grafo

1. Keyword/Synonym-Based Retrieval

Expande la consulta en keywords y sinónimos relevantes para encontrar los nodos de interés.

2. Vector Similarity

Recupera los nodos basados en la similitud de la representación vectorial de la consulta.

3. Cypher Queries

Utiliza el lenguaje de consultas de grafos Cypher para especificar patrones de grafos complejos y recorrer múltiples relaciones.

Se puede proporcionar un template para que el LLM complete, o dejar que este mismo genera la consulta Cypher.

4. Recorrida de Grafo mixta

Se define una lógica de recorrido de grafos a medida utilizando varios componentes distintos de recuperación.

Estos recuperadores pueden combinarse y componerse para realizar una búsqueda híbrida que aproveche tanto la estructura del grafo como las representaciones vectoriales de los nodos.

2.3 Evaluación y Observabilidad de RAG

La evaluación de aplicaciones RAG generalmente y según el estado del arte, se suele separar en dos partes:

- Evaluación de la recuperación: Evaluar si los documentos recuperados son los necesarios y relevantes para contestar a las consultas del usuario.
- La evaluación de la generación: Lo que conlleva evaluar si la generación del LLM responde de forma confiable (no está alucinando) y basada en el contexto proporcionado, además de también evaluar la completitud y el resumen de la respuesta generada.

Aunque todavía no hay una respuesta única a esta problemática de la evaluación, varios distintos enfoques han tomado relevancia desde el advenimiento de LLM en las aplicaciones comerciales

2.3.1 Técnicas

2.3.1.1 LLM-as-a-judge

Existen varias técnicas distintas entre sí del cómo usar a un LLM como juez, pero de forma resumida, buscan utilizar al modelo de lenguaje para validar parcial o totalmente si la respuesta generada cumple con los requerimientos.

2.3.1.2 Comparación de representaciones vectoriales

Utilizando representaciones vectoriales, se puede comparar la similitud entre la consulta original del usuario, los documentos recuperados, y la respuesta generada

por el LLM. Esta técnica se basa en calcular distancias o similitudes en el espacio vectorial para determinar si la respuesta generada está alineada con el contenido de los documentos recuperados y la intención de la pregunta. Algoritmos como la distancia coseno o la distancia euclidiana son comunes para medir la similitud entre vectores, donde se espera que los vectores más cercanos correspondan a respuestas y documentos más relevantes.

2.3.1.3 Evaluación Humana

Aunque el objetivo es automatizar la evaluación, la intervención humana sigue siendo crítica para validar la precisión y relevancia en aplicaciones RAG. Los evaluadores humanos comparan las respuestas generadas por el LLM con las fuentes originales y determinan si la respuesta es coherente, completa y relevante. Este tipo de evaluación se suele utilizar como referencia de calidad para ajustar las métricas automáticas y como una fuente de feedback para mejorar la precisión de los modelos.

2.3.2 Métricas

2.3.2.1 Generación

RAGAS [6] es un framework que ha tomado relevancia últimamente como una forma para lograr la evaluación de aplicaciones RAG y LLM. Utilizando técnicas como LLM-as-a-judge (2.3.1.1) y Comparaciones vectoriales (2.3.1.1) permite obtener métricas de rendimiento de las aplicaciones LLM con recuperación aumentada.

Confiabilidad

Evalúa la consistencia con los hechos de la respuesta generada, comparándolo con el contexto dado. Se calcula utilizando la respuesta y el contexto recuperado.

utiliza el siguiente proceso para la evaluación:

1. Se generan con un LLM una serie de claims o declaraciones sobre las respuestas generadas.
2. Para cada uno de esos claims se verifica si pueden inferirse del contexto dado (el resultado de la etapa de retrieval).

3. Se obtiene un score dado por:

$$Faithfulness = \frac{|claims\ in\ the\ generated\ answer\ inferred\ from\ given\ context|}{Total\ number\ of\ claims\ in\ the\ generated\ answer}$$

Relevancia

Se evalúa cuan pertinente es la respuesta generada por el LLM, dado el prompt. Se calculan utilizando la pregunta, el contexto y la ground truth. La respuesta se considera relevante cuando contesta de forma directa y apropiada la pregunta original. Esta métrica penaliza la falta de completitud o el contenido de detalles redundantes.

RAGAS implementa esto utilizando un LLM para generar preguntas apropiadas para la respuesta generada múltiples veces, y mide la similitud de coseno promedio entre las preguntas generadas y la pregunta original. Si la pregunta genera la respuesta de forma precisa la pregunta original, el LLM debería poder generar preguntas de las respuestas que se alineen con la pregunta original. Luego calcula la métrica de la siguiente forma:

$$\begin{aligned} \text{Answer Relevancy} &= \frac{1}{N}\sum_{i=1}^{N}\cos(E_g, E_o) \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{E_g \cdot E_o}{||E_{g_i}|| \cdot ||E_o||} \end{aligned} \tag{2.1}$$

Context Recall

Mide cuanto del contexto recuperado se alinea con la respuesta generada, tratada como la ground truth.

Se calcula utilizando los parámetros `question`, `ground truth` y el `context` recuperado. Varía entre 0 y 1, con valores más altos indicando mejor performance.

Para estimar el recall del contexto de la `ground truth`, cada afirmación in la respuesta `ground truth` es analizada para determinar si puede atribuirse al contexto recuperado o no.

$$\text{Context Recall} = \frac{|GT\ claims\ that\ can\ be\ attributed\ to\ context|}{|Number\ of\ claims\ in\ GT|} \tag{2.2}$$

Context Precision

Evalúa si los elementos relevantes de la ground-truth presentes en el contexto están rankeados más arriba o no. Todos estos fragmentos relevantes deben aparecer en las primeras posiciones.

Utiliza `question`, `ground truth` y `context`. Varía entre 0 y 1, donde un valor más alto indica mejor precisión.

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} Precision@K \cdot v_k}{Total\ number\ of\ relevant\ items\ in\ top\ K\ results} \tag{2.3}$$

$$\text{Precision@k} = \frac{true\ positives@k}{true\ positives@k + false\ positives@k} \tag{2.4}$$

3 Desarrollo

3.1 Proceso de Estudio Ambiental

3.1.1 Roles

Para el sistema, los principales actores son:

1. El **estándar** es un documento que establece los requisitos que deben cumplir la empresa, organización o proyecto en términos de cumplimiento ambiental. Estos estándares suelen tener un carácter general, ya que están diseñados para ser aplicables a diversas industrias y procesos. Debido a esta naturaleza general, es responsabilidad del experto ambiental adaptar y contextualizar los requerimientos específicos del estándar al proceso o industria particular que se está evaluando.

2. El **proyecto** es un documento que recopila toda la información ambiental relevante de la empresa o del proyecto específico. Este documento incluye el análisis de todos los aspectos ambientales que la empresa ha considerado importantes, y sirve como la base sobre la cual el experto ambiental realiza su evaluación. A menudo, el análisis de esta documentación ambiental genera nuevas preguntas, lo que puede llevar al experto a realizar visitas de campo para verificar la exactitud de la información documentada y para reunirse con las partes involucradas en el proceso, asegurándose de que todos los aspectos relevantes estén dentro del alcance del estudio.

3. El **experto ambiental** es el encargado de interpretar y aplicar los estándares ambientales al proyecto específico. Su rol es fundamental, ya que debe adaptar los requisitos generales del estándar al contexto particular del proyecto, asegurando que se cumplan todas las normativas pertinentes. El experto también es responsable de llevar a cabo el **estudio ambiental**, donde compara y se valida el cumplimiento del **proyecto** en relación con el **estándar**.

Además, el experto realiza inspecciones en terreno y mantiene interacciones directas con las partes involucradas, lo que le permite garantizar que la documentación refleje fielmente la realidad y que se identifiquen y corrijan posibles áreas de no conformidad.

4. El **estudio ambiental** es el proceso mediante el cual el **experto ambiental** evalúa y se valida que el **proyecto** cumpla con los requisitos establecidos en el **estándar**. Este estudio es crucial para asegurar el cumplimiento normativo y la sostenibilidad del proyecto, identificando cualquier área que necesite ajustes o mejoras.

3.2 Sistema Automático de Cumplimiento Ambiental

La solución técnica final de la Prueba de Concepto abarca la generación de los siguientes tres componentes:

- Construcción del grafo: Genera el grafo utilizando el documento.
- Generador de preguntas: Utiliza el grafo construido para generar preguntas que interroguen la documentación del cliente.
- Compliance del Proyecto: Utilizar el grafo para evaluar el cumplimiento.

El código del proyecto puede accederse del siguiente link:

Repositorio de GitHub https://github.com/MarianoChic09/msc-ort-uy-thesis

La motivación de esta separación de responsabilidades es disminuir la complejidad necesaria para completar estas tareas al dividir y hacer más pequeñas las tareas.

3.2.1 Construcción del grafo

Desarrollo del microservicio `graph-generator`.

El **Graph Generator** es un componente clave en la solución que se encarga de

procesar documentos y convertir la información relevante en grafos de entidades y relaciones. Estos grafos representan de manera estructurada las conexiones y dependencias entre diferentes elementos del documento, lo que facilita la posterior generación de preguntas y la validación de cumplimiento en el proyecto.

Estructura General del Proyecto

La estructura del microservicio está organizada de manera modular, teniendo en mente la escalabilidad y facilidad en el mantenimiento. La solución se compone de varios archivos y carpetas clave, cada uno con una función específica:

- `src/`: Este es el directorio principal del código fuente, donde se encuentran los módulos que implementan la lógica central del Graph Generator.
  - `config.py`: Maneja la configuración del sistema, incluyendo la carga de secretos y variables de entorno, permitiendo una configuración flexible basada en el entorno (desarrollo o producción).
  - `index.py`: Implementa la lógica para crear y manejar los Graph Index. Es responsable de transformar los documentos en grafos utilizando modelos de lenguaje (LLMs) y modelos de embeddings.
  - `main.py`: Define la API REST utilizando FastAPI, permitiendo la carga de documentos y la creación de grafos mediante endpoints. Es el punto de entrada para la interacción con el sistema, con dos rutas principales entre las existentes en FastAPI:
    - `/predefined-schema-extractor`: Este utiliza entidades, relaciones y un esquema predefinido que puede pasarse como parámetros de la request o utiliza un esquema por defecto. Este tipo de extracción se refiere a la mencionada en el Capítulo **Marco Teórico** como extracción por esquema predefinido.
    - `/free-form-extractor`: Esta ruta implementa el extractor libre mencionado también en el Capítulo **Marco Teórico** y se refiere a la implementación de un extractor de grafos que deja que el LLM seleccione las entidades, relaciones y el esquema de forma completamente libre.
  - `models.py`: Inicializa los modelos de lenguaje y de embeddings de Azure OpenAI, con el objetivo de utilizarlos para la generación y el análisis de los grafos.

    - `schemas.py`: Define los esquemas y validaciones que se aplican a las entidades y relaciones extraídas de los documentos, utilizado en caso de usar el endpoint de generación de grafo con esquemas predefinidos. Permite como entrada las entidades, relaciones y el esquema predefinido.
    - `utils.py`: Contiene funciones utilitarias, como el guardado de archivos subidos con la API de FastAPI.

- `notebooks/schemas.ipynb`: Contiene notebooks para definir y probar los esquemas de entidades y relaciones que serán utilizados por el sistema.
- `storage/`: Este directorio se utiliza para almacenar los datos persistentes del sistema, como los índices de grafos y los almacenes de vectores. Los diferentes archivos aquí representan las distintas versiones o tipos de almacenamiento utilizados por el sistema.
    - `property_graph_store.json`: Almacena los grafos generados que representan la información extraída de los documentos.

Funcionamiento del Graph Generator

El proceso del Graph Generator se puede resumir en los siguientes pasos:

1. **Carga del Documento**: El sistema permite la carga de documentos a través de la API REST. Estos documentos son almacenados en el sistema y preparados para su procesamiento.
2. **Procesamiento del Documento**: Una vez cargado, el documento es leído y procesado para extraer las entidades y relaciones que forman el grafo. Este procesamiento se realiza utilizando el LLM `gpt-4o` a través de la API de Azure OpenAI, que identifica los componentes clave en el texto.
3. **Generación del Grafo**: Con la información extraída, se genera un grafo que representa las entidades y relaciones encontradas en el documento. Este grafo puede ser ajustado para cumplir con un esquema predefinido o generado libremente dependiendo de la configuración solicitada.
4. **Validación del Esquema**: En el caso de utilizar un esquema predefinido, el sistema verifica que las entidades y relaciones extraídas cumplan con las reglas definidas. Esto garantiza la consistencia y precisión del grafo generado.

5. **Almacenamiento y Persistencia**: El grafo generado se almacena en el sistema, para su posterior recuperación y utilización. La estructura del grafo y la información de los índices se persisten en el directorio de almacenamiento, asegurando que se pueda acceder a ellos para futuras consultas o análisis.

Resumen de la Funcionalidad

El **Graph Generator** permite transformar documentos en representaciones gráficas estructuradas, para facilitar el análisis y la validación de cumplimiento dentro del sistema. Su diseño modular y flexible permite la integración de diferentes modelos y esquemas dentro del sistema, con el objetivo de permitir una rápida iteración y mejora continua.

3.2.2 Generador de preguntas

El **Generador de preguntas** tiene como objetivo generar preguntas clave en los requerimientos extraídos del estándar y que están relacionados con el proyecto. Estas preguntas evalúan la conformidad del proyecto con las regulaciones ambientales. El **Generador de preguntas** está diseñado para generar automáticamente listas de preguntas que luego pueden ser validadas o ajustadas por un experto ambiental.

Estructura General del Proyecto

El Question Generator sigue una estructura modular para facilitar su mantenimiento, escalabilidad y la mejora continua. A continuación, se describe la organización del proyecto:

- `notebooks/`: Contiene notebooks que permiten realizar pruebas y experimentos con el generador de preguntas. Estos notebooks son útiles para evaluar el pipeline de preguntas, probar distintos enfoques de generación y realizar ajustes basados en los resultados obtenidos.
  - `evaluation-pipeline.ipynb`: Se enfoca en evaluar el pipeline de generación de preguntas.
  - `questions_testing.ipynb`: Permite probar diferentes escenarios de generación de preguntas para verificar la calidad y cobertura de las mismas.

- `src/`: Este es el directorio principal del código fuente donde reside la lógica del Question Generator.
    - `config/`: Aquí se encuentra el archivo de configuración que carga los secretos y las variables de entorno, permitiendo la inicialización del sistema en distintos entornos.
    - `index.py`: Gestiona el índice de los documentos procesados y permite la carga y consulta de estos índices.
    - `main.py`: Define la API REST utilizando FastAPI, permitiendo la interacción con el sistema a través de endpoints para generar preguntas y consultar índices. Es el punto de entrada para la generación de preguntas basadas en los estándares.
    - `models.py`: Inicializa los modelos de lenguaje y de embeddings de Azure OpenAI, que se utilizan para la generación de las preguntas y la evaluación de los documentos.
    - `schemas.py`: Define los esquemas de consulta para las preguntas, asegurando que la estructura de los datos enviados y recibidos sea la correcta.

Funcionamiento del Question Generator

El proceso de generación de preguntas empieza por la carga del documento hasta la creación de preguntas específicas basadas en los estándares:

1. **Inicialización del Modelo**: Al iniciar la aplicación, se cargan los modelos de lenguaje y embeddings de Azure OpenAI. Estos modelos se utilizan para interpretar los documentos cargados y generar las preguntas relevantes basadas en la información procesada.
2. **Generación del Índice**: Una vez procesados los documentos, se generan índices que permiten consultas eficientes sobre los documentos. Estos índices son consultados mediante la API para recuperar la información más relevante que guiará la generación de preguntas.
3. **Generación de Preguntas**: El *system prompt* guía al modelo de lenguaje en la generación de preguntas basadas en las responsabilidades y requerimientos extraídos de los documentos. Este prompt está diseñado para generar preguntas que cubran todas las áreas relevantes del cumplimiento ambiental. Las preguntas se generan utilizando el Property Graph Store, que devuelve una lista de preguntas detalladas y específicas.

4. **Almacenamiento y Evaluación**: Las preguntas generadas se almacenan junto con el documento original y el grafo de entidades, para posteriormente realizar la evaluación y ajustar en caso de ser necesario. Un experto puede revisar las preguntas y realizar modificaciones si es necesario, asegurando que cubran todos los aspectos importantes del estándar.

5. **Endpoints de Consultas**: El sistema utiliza distintos endpoints para generar las preguntas. Uno de ellos permite la generación de preguntas basadas en un esquema predefinido (*schema-guided* 2.2.4.2.1.1), mientras que otro permite preguntas más libres basadas en el análisis general de los documentos (*free-form*) 2.2.4.2.1.3, como también utilizando *Baseline RAG* 2.2.2.

Resumen de la Funcionalidad

El **Question Generator** permite automatizar la creación de preguntas relevantes basadas en los requisitos del estándar y los documentos de proyecto. Estas preguntas permite evaluar el cumplimiento de las normativas y ayudar a que no se omitan aspectos clave. Gracias a su diseño modular y flexible, el sistema puede adaptarse a diferentes esquemas de preguntas y estilos de generación.

3.2.3 Compliance del Proyecto

El componente **Project Compliance** es una parte crucial del sistema diseñado para validar si un proyecto cumple con los requisitos ambientales establecidos en los estándares regulatorios. Este módulo procesa los documentos relacionados con el proyecto, utiliza las preguntas relevantes generadas por el **Question Generator** basadas en los estándares y proporciona un análisis detallado sobre el cumplimiento del proyecto con el estándar.

Aunque **Project Compliance** también se vería beneficiado de los beneficios de Graph RAG 2.2.4, por simplicidad y costos se implementó como un Baseline RAG. Esta decisión también se genera a partir de que la generación de las preguntas se encargan de tener una visión más holística sobre la norma, y con esto generar preguntas más específicas, con lo cual el espacio de búsqueda de Project Compliance ante cada pregunta debería ser más acotado.

Estructura General del Proyecto

El sistema **Project Compliance** sigue una arquitectura modular que facilita su mantenimiento y extensibilidad. Los principales directorios y archivos del proyecto son los siguientes:

- `data/`: Contiene los documentos procesados, como informes ambientales, que son analizados por el sistema para generar preguntas y validar el cumplimiento.
- `notebooks/`: Incluye notebooks para realizar pruebas, validaciones y análisis de la conformidad del proyecto.
  - `evaluation-pipeline.ipynb`: Se utiliza para evaluar el pipeline de validación de conformidad del proyecto.
- `src/`: Es el directorio principal del código fuente, donde se encuentra la lógica del sistema.
  - `main.py`: Define la API REST utilizando FastAPI. Proporciona los endpoints para cargar documentos, generar preguntas, y consultar el índice para verificar el cumplimiento del proyecto. Es el punto de entrada del sistema.
  - `index.py`: Implementa la lógica para crear y gestionar los índices de los documentos procesados, utilizando un sistema de almacenamiento de vectores.
  - `question_generator.py`: Se comunica con el generador de preguntas para obtener preguntas relevantes basadas en los documentos del proyecto.
  - `models.py`: Inicializa los modelos de lenguaje y embeddings de Azure OpenAI, que son fundamentales para procesar los documentos y generar respuestas a las consultas.
  - `schemas.py`: Define la estructura de las consultas y peticiones que se realizan al sistema.
  - `utils.py`: Proporciona funciones auxiliares, como la verificación de la existencia de directorios y la gestión de archivos.
- `storage_vector_store/`: Este directorio almacena los datos persistentes del sistema, como los vectores de los documentos procesados y las estructuras de los grafos.

Funcionamiento del Graph Generator

El proceso de validación de conformidad del proyecto se puede dividir en varias etapas:

1. **Carga de Documentos**: El sistema permite la carga de documentos relacionados con el proyecto a través de la API REST. Estos documentos son almacenados en el sistema y se procesan para generar un índice basado en vectores, que facilita la búsqueda y consulta de información.
2. **Generación de Preguntas**: Utilizando el módulo **Question Generator**, se generan preguntas relevantes basadas en los documentos del proyecto y los estándares regulatorios. Estas preguntas ayudan a verificar que todos los aspectos del cumplimiento hayan sido abordados en el análisis.
3. **Consultas sobre el Índice**: El sistema utiliza un índice de almacenamiento vectorial que permite realizar consultas basadas en las preguntas generadas. El índice es clave para identificar las partes relevantes de los documentos que pueden responder a cada pregunta.
4. **Validación de Cumplimiento**: El sistema compara las preguntas generadas con la información almacenada en los documentos, evaluando si el proyecto cumple con los requisitos del estándar. Esta validación puede realizarse a través de múltiples consultas al índice y la búsqueda de respuestas específicas para cada pregunta.
5. **Interacción con el Sistema**: El componente **Project Compliance** expone varios endpoints en la API REST que permiten:
    - **Cargar documentos**: Actualizar el índice con nueva documentación.
    - **Generar preguntas**: Basadas en un estándar o un conjunto de preguntas predefinidas.
    - **Consultar respuestas**: Buscar respuestas a las preguntas generadas y validar el cumplimiento del proyecto.

Resumen de la Funcionalidad

El componente **Project Compliance** permite automatizar el proceso de validación de conformidad de un proyecto en relación con los estándares regulatorios. Además, facilita la verificación de respuestas específicas y la identificación de posibles áreas de incumplimiento o áreas donde falte información.

3.3 Experimentación y Resultados

Se separa la experimentación en las tres partes que contempla el pipeline: `graph-generation`, `question-generator`, y `project-compliance`, con el objetivo de analizar por separado el rendimiento de cada parte y su contribución al resultado final.

El objetivo es evaluar la capacidad del sistema de extraer información del documento, utilizarla para generar preguntas sobre el proyecto con base en el estándar, y generar las respuestas a dichas preguntas utilizando los documentos ambientales con los que cuente la organización o proyecto bajo estudio.

Para esta experimentación se utiliza solo una de las secciones del ES Framework, la sección 3: Resource Efficiency and Pollution Prevention and Management.

3.3.1 Graph Generator

Para este servicio, los parámetros que podemos modificar para generar cambios en los resultados son los siguientes:

1. La forma de extracción del grafo.
2. El prompt utilizado.

3.3.1.1 Free Form extraction

La extracción libre trajo los siguientes resultados mostrados en la Figura 3.1. Estos modelan un diagrama de nodos extraídos con el LLM de forma libre, permitiendo que elija cuáles son los nodos, tipos de relaciones y las relaciones entre nodos que pueden existir.

3.3.1.2 Extracción guiada por un esquema

Se utilizaron varios esquemas distintos como parte de la creación del grafo usando este tipo de extracción, finalmente el esquema utilizado fue el siguiente:

```
schema = {
```

Figura 3.1: Extracción Free Form

```
    "entities": ["BORROWER", "REQUIREMENT"],
    "relations": ["MUST_COMPLY_WITH", "IS_OBLIGATED_TO"],
    "validation_schema": {
        "BORROWER": ["MUST_COMPLY_WITH"],
        "CONTRACTOR": ["IS_OBLIGATED_TO"],
        "REQUIREMENT": [],
    },
}
```

3.3.2 Question Generator

El cambio principal que puede ocurrir acá es el prompt utilizado para generar la lista de preguntas. Además, como parte de los trabajos futuros, se podría clasificar las preguntas generadas por área, eliminar las preguntas duplicadas (semánticamente distintas, pero iguales en significado) y añadir más información de contexto para la generación de las respuestas.

Para comparar los resultados se generaron respuestas utilizando tanto Baseline RAG como fuente de base de datos vectorial para interrogar al Vector Database, como GraphRAG.

Se utilizó el mismo prompt para generar preguntas sobre el estándar tanto utilizando la base de datos vectorial, como la Graph Property Store extraída con un schema y la Graph Property Store de forma libre, instruyéndole al LLM que genere una lista de preguntas que cubra todos los requerimientos utilizados.

El prompt es el siguiente:

AI

As an expert in environmental compliance, you are tasked with ensuring that a project adheres to all relevant environmental regulations and guidelines. Below is a summary of the borrower's responsibilities related to environmental compliance.
Based on this information, generate a comprehensive list of questions that you would ask to validate the project's compliance. Your questions should cover all aspects mentioned in the summary. Ensure that the questions are thorough and specific enough to identify any potential compliance issues.
Treat every question generated as a separate entity.

3.3.2.1 Baseline RAG

Utilizando una base de datos vectorial con 5 particiones de 1000 tokens el agente genero 20 preguntas, algunas de las cuales se observan a continuación:

Question Generator Response

1. What measures have been implemented to address potential adverse impacts on human health and the environment, considering factors such as existing ambient conditions, land use, and proximity to biodiversity areas?
2. How has the project evaluated and selected location alternatives to minimize negative effects in already degraded areas?
3. What technically and financially feasible options have been considered to avoid or minimize project-related air emissions during the design, construction, and operation phases?
4. Can you provide an estimate of the gross GHG emissions resulting from the project, and what methods were used to develop this estimate?
5. How does the project plan to manage and minimize the generation of hazardous and nonhazardous waste, and what measures are in place for the safe disposal of such waste?
6. What procedures are in place to ensure compliance with national legislation and international conventions for the management of hazardous waste?
7. How does the project plan to avoid the manufacture, trade, and use of chemicals and hazardous materials subject to international bans or restrictions?
8. What steps have been taken to minimize and control the release and use of hazardous materials, and what less hazardous substitutes have been considered?
9. What integrated pest management (IPM) or integrated vector management (IVM) approaches are being used, and how are they being implemented?
10. How has the project assessed the risks associated with the procurement and use of pesticides, and what criteria were used to select the pesticides?

3.3.2.2 Resultados con GraphRAG

De forma similar, se instruyó a un LLM, pero utilizando la base de conocimiento de grafos como fuente de información, para que genere una lista de preguntas detalladas para interrogar posteriormente al proyecto.

Utilizando Free-Form Extraction

Las preguntas generadas utilizando el grafo generado por el Free Form extractor, cuyos resultados de extracción se mencionan en la sección anterior, son las siguientes:

El agente generó 57 preguntas utilizando este enfoque.

Question Generator Response

1. Has the borrower conducted an assessment of hazardous materials for project activities?
2. Are there significant pest management issues associated with the project?
3. Is the borrower complying with all relevant environmental requirements?
4. Has the borrower adopted Good International Industry Practice (GIIP) alternatives where applicable?
5. Has the borrower considered existing and future land use in the project planning?
6. How does the borrower plan to destroy waste generated by the project?
7. Are licensed disposal sites being used for waste disposal?
8. What measures has the borrower implemented to minimize the release of hazardous materials?
9. What steps has the borrower taken to avoid or minimize water usage?
10. Has the borrower considered existing ambient conditions in the project area?

Utilizando Schema guided extraction

Las preguntas generadas (25 preguntas) utilizando el grafo generado con el schema guided extractor son las siguientes:

Question Generator Response

1. Have all relevant factors been considered to address potential adverse project impacts on human health and the environment?
2. Has the project characterized and estimated sources of air pollution as part of the environmental and social assessment?
3. What additional strategies and measures have been adopted to avoid or minimize negative effects in areas where the project constitutes a significant source of emissions?
4. Are there technically and financially feasible and cost-effective options implemented to avoid or minimize project-related air emissions during the design, construction, and operation phases?
5. Has a health and safety risk assessment of existing pollution been undertaken?
6. Is the project in compliance with existing requirements for the management of hazardous wastes, including national legislation and applicable international conventions?
7. Has the site been remediated in accordance with national law and Good International Industry Practice (GIIP)?
8. Are contractors used for waste disposal reputable and legitimate enterprises?
9. Are waste materials treated, destroyed, or disposed of in an environmentally sound and safe manner, including appropriate control of emissions and residues?
10. Is there a process established to identify the responsible party for historical pollution?

3.3.2.3 Evaluación

Comparación Asistida

El objetivo de la evaluación de las preguntas generadas por Question-Generator es entender que porcentaje del documento çubren"las preguntas generadas (en adelante denominaremos a esta métrica coverage). El coverage nos permite entender, que tanto se están capturando los requerimientos del estándar que el proyecto bajo estudio tiene que cumplir.

Debido a la falta de bibliografía en este tema, se desarrolló una metodología para calcular esta métrica, donde se obtiene Para determinar y calcular el coverage se sigue el siguiente proceso:

1. Se le solicitó al experto ambiental que genere una lista de preguntas que contemplen los requerimientos mencionados en la ESS 3.
2. Se generan las representaciones vectoriales o embeddings de todas las preguntas del experto y las preguntas generadas por el sistema.
3. Se comparan vectorialmente los embeddings de las preguntas del experto contra las preguntas generadas por el sistema utilizando la diferencia de coseno.
4. Se genera un heatmap para identificar visualmente y guiar al experto en las preguntas más similares. El resultado puede observarse de la figura 3.2.

El objetivo de este heatmap es guiar al experto en la búsqueda de las preguntas más similares, mediante la similitud de coseno, para que fácilmente determine si todas las preguntas que él generó están cubiertas dentro de las preguntas generadas por el sistema.

Comparación Manual

Por otro lado, también se realizó un estudio y comparación manual con el experto, al cual se le envió tres sets de preguntas generadas sobre el ESS3, utilizando los siguientes extractores:

- Set 1: Baseline RAG 2.2.2.
- Set 2: Graph RAG con un extractor libre 2.2.4.2.1.3.

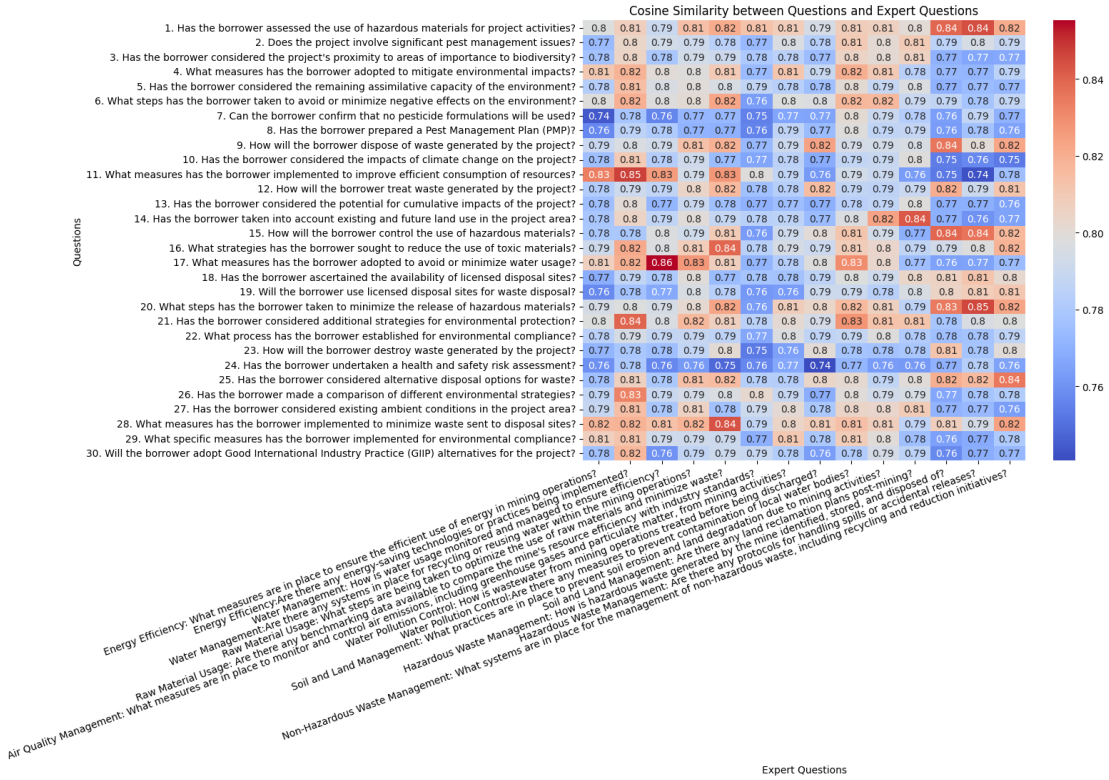

Figura 3.2: Heatmap de Pipeline de Evaluación de Question Generator

- Set 3:Graph RAG con un extractor guiado por esquema 2.2.4.2.1.1.

Esto fue enviado al experto sin que tenga noción de cuál era el sistema que estaba generando las preguntas, se le solicitó comparar los 3 sets de preguntas generadas y elegir la mejor.

Los resultados de la opinión del experto pueden observarse del siguiente cuadro:

Expert's Opinion

When comparing the three sets of questions I would say that:
Set 1 has a good coverage of ESS3 topics, but includes some questions less relevant to the standard. Set 2 focuses well on ESS3 topics but has fewer questions overall and some that are less relevant. Set 3 has the highest concentration of questions directly related to ESS3, with very few questions that fall outside the scope of the standard.
Based on this analysis, I would say that Set 3 is the best set of questions in terms of relevance and applicability to ESS3. It covers the widest range of topics within the standard, including resource efficiency, pollution prevention, waste management, hazardous materials, pest management, and emissions. The questions in Set 3 are also more detailed and specific, aligning closely with the requirements and objectives of ESS3.

El experto concluyó que las preguntas que mejor representaban el ESS3 fueron las extraídas con Graph RAG Schema Guided, cuya Knowledge Base se puede ver de la figura 3.1 y se explica en la subsección 3.3.1.2.

3.3.3 Project-Compliance

Por simplicidad, el sistema de validación de compliance se realiza como un sistema Baseline RAG, que sigue el procedimiento descrito en la sección 2.2.2 donde el documento ambiental del proyecto se pre procesa, se particiona, se generan las representaciones vectoriales y se almacenan en un Vector Store.

Se interroga a esta base de conocimientos las preguntas generadas por el sistema con Question-Generator, y se obtienen las respuestas que proporciona el sistema RAG basándose en el contexto recuperado, lo que puede observarse de la tabla 3.1.

Así mismo, varias preguntas no pudieron ser respondidas porque el sistema RAG no encontró la información suficiente, o no existe la información en el documento, como se observa de la tabla 3.2. En varios casos, esto simplemente supone que no hay suficiente información para responder a dicha pregunta y en realidad eso podría suponer un no cumplimiento de alguno de los requerimientos, a validar por el experto.

De las 30 preguntas generadas por el sistema que se utilizaron para este ejemplo, 25 pudieron ser respondidas, mientras que 5 dieron como resultado que el contexto provisto no es suficiente para responder a la pregunta.

3.3.3.1 Evaluación

Para la evaluación de las respuestas generadas en `Project Compliance` se utilizó el framework de evaluación RAGAS [6], se utiliza un set de preguntas generadas por el experto sobre el ESS 3 que se utiliza para la mayoría de los experimentos de este trabajo. Con esto, se utiliza el pipeline de evaluación para generar las respuestas a estas preguntas y calcular las métricas mencionadas en 2.3.2. El proceso de evaluación se observa en la figura 3.3.

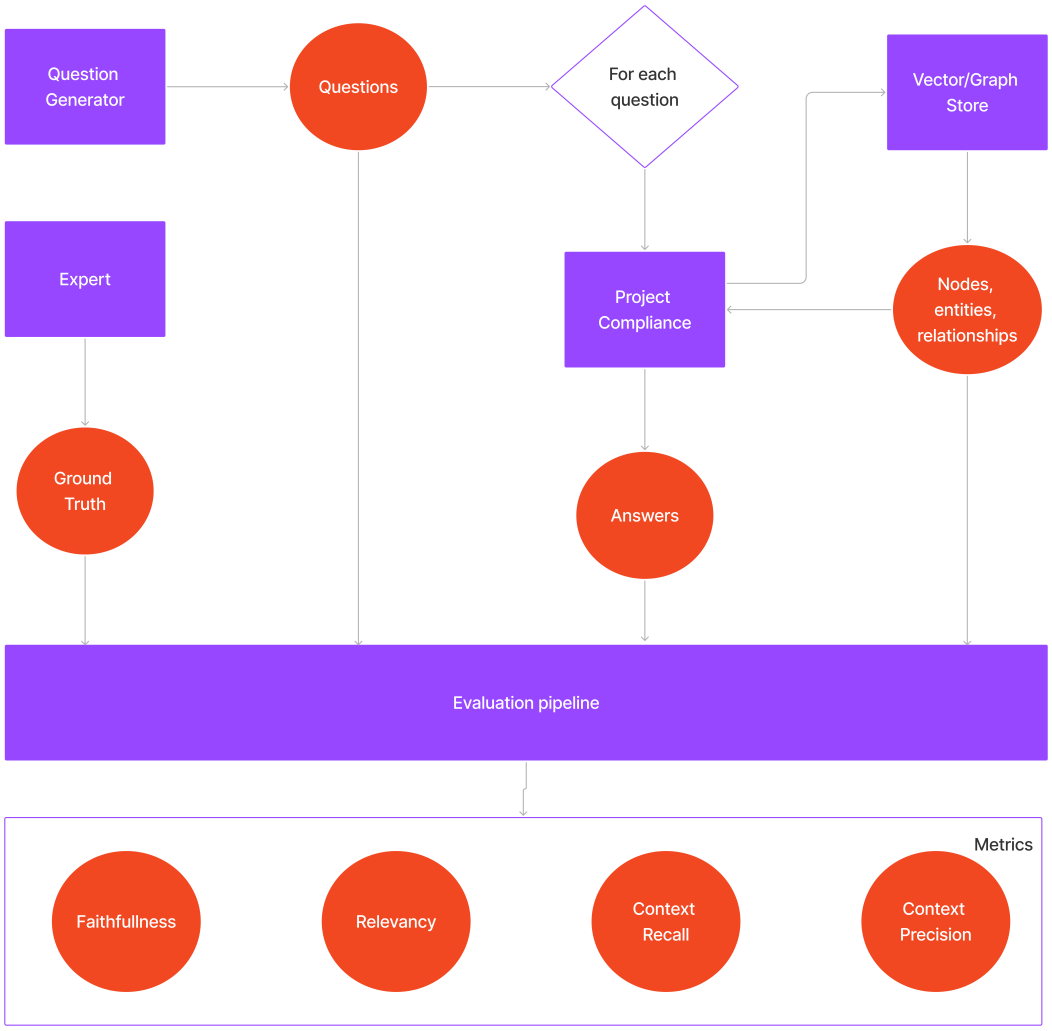

Figura 3.3: Pipeline de Evaluación de Project Compliance

A continuación se observa una comparación entre la pregunta, la respuesta del

experto y la respuesta generada por el sistema. Comparándolas, la respuesta del experto contiene mayor cantidad de temas cubiertos y mayor detalle de cada uno de los temas mencionados, la respuesta del sistema cubre una cantidad significativa de los temas mencionados por el experto.

Pregunta

Are there technically and financially feasible and cost-effective options implemented to avoid or minimise project-related air emissions during the design, construction, and operation phases?

Respuesta del Experto

The EIA and EMP report provides some information about measures that should be implemented to avoid or minimise project-related air emissions during the design, construction, and operation phases of the Jindal MIOP. The design phase should include technical specifications for process emissions from the proposed milling plant.

Design Phase

- **Mitigation Hierarchy:** Specific details about air emission mitigation measures during the design phase, such as utilising renewable energy sources or incorporating air quality considerations in infrastructure placement, are not mentioned.

Construction Phase

- **Dust Management:** The report emphasises dust control during construction, including minimising drop heights for materials, utilising water suppression techniques, and covering vehicles transporting dry materials.
- **Timing of construction:** Restricting construction activities to daytime hours as defined by the DMRE is mentioned as a way to mitigate noise pollution. While this primarily targets noise reduction, it indirectly helps minimise air emissions by avoiding activities during potential temperature inversions which can trap pollutants.
- **Vehicle Maintenance**: This indirectly contributes to air quality control because exhaust emissions from well-maintained vehicles should not exceed the design limits.

Respuesta del Experto - Continuación

Operational Phase:

- **Air Quality Monitoring:** Continuous air quality monitoring for dust and other pollutants is required, with a focus on areas near the project boundary and potential sensitive receptors like nearby farms.
- **Dust Suppression:** The plan includes ongoing dust suppression measures, including the use of surfactants and dust suppressants during watering, especially near project boundaries. Maintaining existing vegetation and planting additional native trees is also encouraged to act as windbreaks and reduce dust.
- **Operational Practices:** The plan mandates practices to minimise dust generation, such as covering trucks, carrying dry materials, using chutes at material transfer points, minimising vehicle speeds on paved and unpaved roads, and maintaining haul routes with less erodible material.
- **Blasting Management:** The plan outlines measures to control dust during blasting, including limiting blasting during specific wind conditions, avoiding blasting in fog or low clouds, restricting blasting to daytime hours, and considering pre-watering or palliative application on blast areas.
- **Energy Use Reduction:** While specific details are limited, the plan encourages the exploration of energy reduction options, including decarbonising the electricity supply (potentially through on-site power generation through renewables) and electrifying the mining vehicle fleet.

Conclusion:
The EIA and EMP outline various measures to manage and mitigate air emissions, but the sources lack specific details about the technical specifications of some proposed solutions, particularly regarding energy use reduction strategies. There is also no mention about emissions from the proposed milling and magnetic separation processing plant.

Respuesta del Sistema

Yes, there are several technically and financially feasible and cost-effective options implemented to avoid or minimize project-related air emissions during the design, construction, and operation phases. These include:

1. Reducing Vehicle Speeds: Limiting vehicle speeds to 40 km/hr on paved site roads and 20 km/hr on unpaved haul routes.
2. Utilizing Chutes: Using chutes at material handling transfer points to minimize dust.
3. Enclosing Processing Plant: Ensuring the processing plant is enclosed and designing bag filters on extraction points for 30 mg/Nm3.
4. Covering Vehicles: Covering vehicles carrying dry soil and other materials during travel.
5. Maintaining Haul Routes: Using less erodible aggregate material for haul routes and maintaining them properly.
6. Air Quality Monitoring: Implementing continuous ambient air quality monitoring, during the test phase and long-term monitoring campaigns.
7. Blasting Restrictions: Restricting blasting to low wind speeds, avoiding certain wind directions, and not blasting during fog, low overcast clouds, or in the dark.
8. Wind Speed Reduction: Reducing wind speed around dusty areas through early planting of native tree species and strategic use of barriers.
9. Grievance Mechanism: Implementing a grievance procedure for air quality issues to be raised and addressed transparently and promptly.

These measures are designed to mitigate air emissions effectively throughout the various phases of the project.

Comparando la respuesta generada por el sistema con la generada por el experto, vemos que aunque la respuesta del experto es más profunda y detallada si comparten similitudes en los temas que se basan para generar la respuesta. Dado que en el servicio Project Compliance la implementación es de Naive o Baseline

RAG 2.2.2, si bien estas no son respuestas perfectas, son prometedoras, con lo que potencialmente se puede lograr implementado técnicas de RAG avanzadas o utilizando Graph RAG.

Se utilizó RAGAS y dos preguntas generadas por el sistema para comparar la respuesta del experto contra las respuestas que generó el sistema, que se observan en la tabla 3.3. Las métricas calculadas se desarrollan en la sección 2.3.2.

La generación de respuestas por parte del experto es larga y complicada debido al nivel de detalle necesario para poder responder una pregunta de este estilo. Por esto se generaron solo dos respuestas sobre un proyecto de ejemplo de la minera de acero y carbón Jindal Steel & Power con minas en Sudáfrica, Mozambique y Botswana. Estos recursos son de acceso libre y gratuito. El resultado de las métricas de RAGAS no es concluyente respecto de la calidad de las respuestas debido a que solo se comparan dos respuestas contra la del experto. Aunque no sea concluyente, supone un primer paso en el desarrollo de un pipeline que permita la validación y comparación de las respuestas generadas por el sistema. El experto colaboró en comparar la respuesta generada con su respuesta y concluyó que esta podría funcionar como una primera respuesta para luego profundizar más.

Question	Response
14. Are GIIP alternatives adopted for the environmentally sound and safe management and disposal of hazardous waste where existing requirements are absent?	Yes, alternatives that align with Good International Industry Practice (GIIP) are adopted for the environmentally sound and safe management and disposal of hazardous waste where existing requirements are absent. This includes measures such as proper storage and handling of hazardous substances, use of drip trays, designated washing and cleaning areas, immediate cleanup of spillages, and appropriate disposal of contaminated materials.
18. Is the project in compliance with national legislation regarding environmental and social impacts?	The project has taken into account various national legislations and guidelines regarding environmental and social impacts. These include regulations on waste management, water quality, air quality, and biodiversity. The project planning and assessments have been informed by these regulations, indicating compliance with national legislation.
2. Has the project characterized and estimated sources of air pollution as part of the environmental and social assessment?	Yes, the project has characterized and estimated sources of air pollution as part of the environmental and social assessment. It identified various local sources of air pollution, including agricultural activities, biomass burning, domestic fuel burning, unpaved roads, and vehicle emissions. Additionally, assumptions and methodologies for air quality dispersion modeling were outlined, including considerations for peak model years, buffer zones, truck emissions, and particle size distribution.

Tabla 3.1: Resultados del sistema RAG de Project Compliance

Question	Response
28. Are reputable and legitimate contractors being used for the transportation of hazardous materials?	The provided information does not specify whether reputable and legitimate contractors are being used for the transportation of hazardous materials.
10. Is there a process established to identify the responsible party for historical pollution?	There is no specific process mentioned for identifying the responsible party for historical pollution.

Tabla 3.2: Resultados del sistema RAG de Project Compliance

Métrica	Resultado
Faithfulness	0.3333
Answer Relevancy	0.9825
Context Recall	0.3891
Context Precision	1.0000
Harmfulness	0.0000

Tabla 3.3: Resultados métricas RAGAS de Project Compliance

4 Experimentación adicional y Posibles Mejoras

4.1 Agentic LLMs

Los agentes basados en LLM son aplicaciones que pueden ejecutar tareas complejas mediante una arquitectura que combina los LLM con módulos clave como planificación y memoria. Al construir agentes LLM, un LLM actúa como el controlador principal o "cerebro" que dirige el flujo de operaciones necesarias para completar una tarea o solicitud del usuario. El agente LLM puede requerir módulos clave como planificación, memoria y uso de herramientas.

Un marco de trabajo para un agente LLM consiste en los siguientes componentes clave:

- Solicitud del usuario - Una pregunta o solicitud realizada por el usuario.
- Agente/Cerebro - El núcleo del agente que actúa como coordinador.
- Planificación - Ayuda al agente a planificar acciones futuras.
- Memoria - Gestiona los comportamientos pasados del agente.

Han surgido varias implementaciones de frameworks que intentan abstraer la complejidad de desarrollar este tipo de agentes, algunos de ellos son:

- LangGraph
- CrewAI
- LlamaIndex

- Devin - Generación de código
- ChatDev - Generación de código
- AutoGPT

4.1.1 LangGraph

LangGraph es una librería open-source para la construcción de aplicaciones LLM stateful y multi-actor con LLMs, utilizado para crear flujos de trabajo de agentes o multi-agentes. LangGraph permite definir grafos cíclicos, que son esenciales para la mayoría de las arquitecturas de agentes, a diferencia de las soluciones basadas en DAG (Directed Acyclic Graph). Permite un control fino sobre el flujo y el estado de la aplicación, que es crucial para crear agentes confiables.

4.1.1.1 Flujo de trabajo con LangGraph

- Inicializar el modelo y las herramientas si las hubiera.
- Inicializar el grafo con el estado inicial.
- Definir los nodos del grafo.
- Definir el punto de entrada y las aristas del grafo.
- Compilar el grafo.
- Ejecutar el grafo.

4.1.1.2 Experimentación

Se implementó el flujo de postprocesamiento con LangGraph de la figura 4.1. El objetivo de este postprocesamiento es elegir cuáles de las preguntas generadas son aplicables al contexto de la empresa o proyecto bajo estudio. La aplicabilidad de los requerimientos ambientales se modifican bastante en función de la industria o proceso del contexto de aplicación.

Este proceso es un comienzo del proceso de filtrado de las preguntas que no apliquen o una reformulación de las preguntas que lo ameriten para ser más específico en el proceso o industria de aplicación.

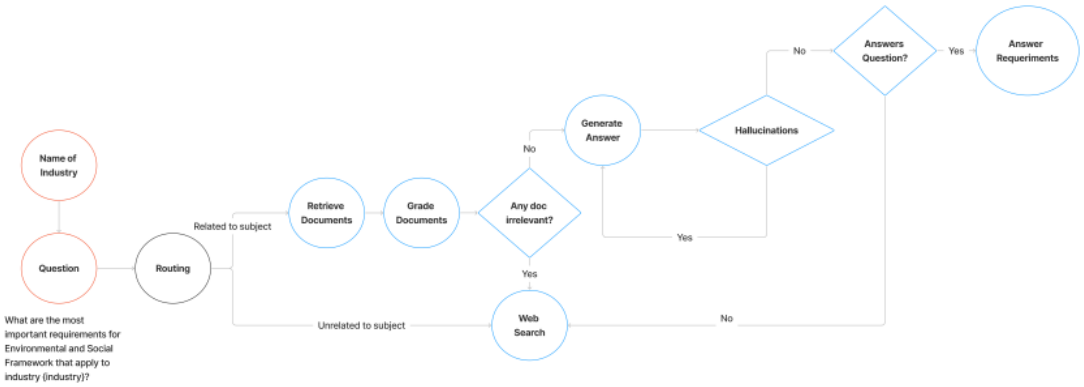

Figura 4.1: Flujo de trabajo Post processing con LangGraph

Por falta de tiempo de experimentar más en profundidad, principalmente, esta experimentación no fue incluida y desarrollada en la prueba de concepto principal. Sin embargo, ofrece posibilidades interesantes para orientar el campo de aplicación de las preguntas a las del proyecto, y también para remover preguntas que no apliquen.

4.2 Mejoras y Trabajos Futuros

1. Utilizar un modelo de lenguaje con fine-tuning para la creación del Grafo. Implementaciones como **Triplex - SOTA LLM for Knowledge Graph Construction**. Esto permitiría disminuir el costo alto de construcción y actualización de grafos comparado al uso de `gpt-4o`. Por ejemplo, para **Graph-Rag by Microsoft**, el costo estimado es de un token generado de salida por cada token de entrada.

2. Implementar GraphRAG como base de conocimiento también en project-compliance para la búsqueda de respuestas a las preguntas.

3. La inclusión de LangGraph para el postprocesamiento de las preguntas, que podría incluir un usuario que interactúe con el sistema para mejorar la calidad de las preguntas antes de que sean respondidas por el sistema de project-compliance.

4. Se podría desarrollar una funcionalidad donde el mismo usuario genere por ejemplo 3 preguntas más en detalle respecto de alguna pregunta seleccionada de las generadas, para los casos donde las preguntas todavía sean de muy alto nivel. Esto ayudaría a interrogar el proyecto con mayor nivel de detalle.

5. Teniendo en cuenta las conclusiones obtenidas en la evaluación del servicio Project Compliance 3.3.3.1 se considera la posibilidad de implementar las siguientes características en el sistema:

    - Hierarchy of Questions: Utilizar las preguntas generadas por el Question Generator y el Graph Knowledge Base como base para generar otras preguntas que profundizan en el tema de la pregunta de base. El objetivo de esto es lograr una estructura jerárquica de preguntas, donde cada una vaya desglosando o especificando más la pregunta inicial. De esta forma, el sistema podría guiar al usuario en la exploración de temas complejos, proporcionando no solo respuestas inmediatas, sino también un enfoque más profundo y detallado del problema o pregunta original.
    - Human-in-the-loop Question Generation: Integrar una funcionalidad que permita a los usuarios intervenir directamente en la generación de preguntas. Los usuarios podrán revisar, modificar, agregar o eliminar preguntas propuestas por el sistema antes de que se finalice el set de preguntas. Esto les permite tener mayor control sobre la relevancia y calidad del set de preguntas generado, asegurando que el sistema se ajuste más finamente a los requerimientos del proyecto. Esta capacidad no solo optimiza la calidad de las preguntas, sino que también genera un ciclo de retroalimentación continuo donde el sistema pueda aprender de las ediciones hechas por los usuarios para mejorar futuras iteraciones.
    - User Feedback Loop: Incorporar un sistema de feedback donde el usuario pueda marcar preguntas que considera incorrectas o fuera de contexto, permitiendo al sistema aprender de los errores y ajustar los servicios Question Generator y Project Compliance. Esto mejora la calidad del sistema a largo plazo y asegura que las preguntas generadas sean cada vez más relevantes.
    - Adaptive Question Filtering: Desarrollar una funcionalidad que permita al sistema filtrar las preguntas generadas en función de criterios predefinidos, como la relevancia, el nivel de detalle o el enfoque del proyecto. Esto le permitirá al usuario ajustar las preguntas según el contexto o los objetivos específicos del proyecto.

5 Conclusiones

+ontext vs RAG

Se ha desarrollado una prueba de concepto que implementa una solución basada en técnicas avanzadas de RAG, y principalmente Graph RAG, con el objetivo de extraer los requerimientos ambientales relevantes a una empresa para poder estar al día con el cumplimiento regulatorio. Si bien esta prueba de concepto se enfoca en el aspecto ambiental, lo desarrollado puede extenderse a muchas otras áreas donde es necesario el cumplimiento con algún tipo de regulación.

Desde la perspectiva de negocio, el enfoque utilizado demuestra la posibilidad de automatizar parte del proceso de análisis documental y generación de preguntas y respuestas relevantes para asistir a expertos ambientales cuando realizan estudios ambientales. Con este tipo de sistemas, se podría asistir a los expertos, disminuyendo el tiempo de realización de los estudios, aumentar la eficiencia y asegurando una cobertura mayor sobre los requerimientos que estándares como el Environmental and Social Framework puedan tener para su organización.

El rol del experto fue fundamental para la evaluación y validación de los resultados obtenidos en este trabajo. Su opinión permitió tomar decisiones de diseño clave a la hora de finalizar la prueba de concepto end-to-end. Una de las partes más interesantes de este proceso fue aprender como interactuar con un SME para lograr desarrollar soluciones sobre un dominio complejo y que requiere muchos años de experiencia para trabajar en él, tanto en el establecimiento de una rutina de trabajo eficiente como en mantener una comunicación clara y efectiva.

GraphRAG ha demostrado ser especialmente efectivo para modelar y capturar relaciones complejas entre entidades, superando a los métodos de RAG convencionales en escenarios donde el contexto es extenso y las relaciones interdependientes son cruciales.

La conclusión principal en la comparación de RAG y Graph RAG es que la respuesta de GraphRAG en la generación de preguntas tiene una mayor cober-

tura de los requerimientos ambientales que plantea el ES Framework. El experto con el que trabajamos en este proyecto validó esto confirmando que la respuesta en la generación de preguntas de GraphRAG cubría más y con mejor calidad el estándar ambiental. GraphRAG aparece como una respuesta superior a poder tomar en cuenta una base documental completa en consideración a la hora de generar respuestas a un usuario. La naturaleza relacional de los grafos permite modelar estructuras y capturar relaciones en información más compleja y de forma más completa.

Finalmente, aunque la evaluación de las respuestas generadas por estos sistemas ha sido ampliamente estudiada, aún queda mucho por avanzar en la evaluación de las preguntas que los sistemas generan. En este sentido, la inclusión de un experto en el proceso resulta crucial, ya que no solo se busca la precisión en las respuestas, sino también el enfoque adecuado en las preguntas, según los intereses y necesidades específicas de este dominio de aplicación.

El estado del arte en LLMs y RAG está evolucionando constantemente, con avances que permiten manejar ventanas de contexto más amplias y mejorar la eficiencia de procesamiento. Sin embargo, la capacidad de los LLM de mantener la atención precisa disminuye con el largo del rango contextual. Por esto es que RAG, GraphRAG y sus continuas mejoras destacan, al combinar la búsqueda inteligente en bases de datos externas, con la información modelada a medida del problema, con el fin de acceder a grandes volúmenes de información y de carácter privado sin perder precisión. Destaca también la posibilidad de actualizar continuamente la información e incrementalmente mejorar la respuesta del sistema, sin tener que entrenar de cero o hacer fine tuning sobre un modelo de lenguaje que es muy costoso en tiempo y procesamiento.

6 Bibliografía

[1] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson, "From local to global: A graph rag approach to query-focused summarization," 2024. [Online]. Available: https://arxiv.org/abs/2404.16130

[2] W. Zhu, J. Thomason, and R. Jia, "Chain-of-questions training with latent answers for robust multistep question answering," 2023. [Online]. Available: https://arxiv.org/abs/2305.14901

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[4] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich,

A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[5] C.-P. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, and B. Ginsburg, "Ruler: What's the real context size of your long-context language models?" *arXiv preprint arXiv:2404.06654*, 2024.

[6] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," 2023. [Online]. Available: https://arxiv.org/abs/2309.15217