Uso de modelos de lenguajes de gran tamaño, para consultar bases de datos en lenguaje natural

Use of large language models to query databases in natural language

Alejandro Pasos Ruíz[1], Emilio Gabriel Rejón Herrera[2]
Universidad Autónoma de Yucatán, Facultad de Matemáticas.
Periférico Norte, Tablaje Cat. 13615, Mérida, Yucatán, México.
[1]apasos@correo.uady.mx, [2]rherrera@correo.uady.mx.

Fecha de recepción: 20 de julio de 2024
Fecha de aceptación: 14 de septiembre de 2024

Resumen. El objetivo de este trabajo, es utilizar Inteligencia Artificial Generativa (GenIA), para optimizar la experiencia de usuario. Lo anterior, mediante la implementación de un procedimiento, con modelos de lenguajes de gran tamaño (LLM, Large Language Model), en consultas de bases de datos utilizando lenguaje natural. De tal forma, al proporcionar la estructura de la base de datos y el tipo de visualización requerida, se mejora dicho procedimiento. Por lo tanto, su efectividad es del 93% en la generación de consultas SQL y del 100%, para las visualizaciones. Finalmente, para proyectos futuros se sugiere obtener retroalimentación directa de los usuarios, así como ampliar el esquema e implementar el procedimiento en organizaciones educativas, para probar su factibilidad en una mayor escala.

Palabras Clave: GenAI, LLM, ChatGPT, SQL, Educación.

Summary. The objective of this work is to use Generative Artificial Intelligence (GenIA) to optimize the user experience. The above, through the implementation of a procedure, with large language models (LLM, Large Language Model), in database queries using natural language. In this way, by providing the structure of the database and the type of visualization required, this procedure is improved. Therefore, its effectiveness is 93% in generating SQL queries and 100% for visualizations. Finally, for future projects it is suggested to obtain direct feedback from users, as well as expand the scheme and implement the procedure in educational organizations, to test its feasibility on a larger scale.

Keywords: GenAI, LLM, ChatGPT, SQL, Education.

1 Introducción

En los últimos años, se ha observado un gran avance en el campo de la Inteligencia Artificial (IA), particularmente, en la disciplina de Aprendizaje Automático (ML, Machine Learning), que, a través de algoritmos, permite identificar patrones en datos masivos y elaborar predicciones (o análisis predictivos); de tal forma, se han desarrollado modelos basados en redes neuronales artificiales (RNA´s); los cuales, han demostrado su utilidad cuando son aplicados para resolver problemas en dominios específicos. Sin embargo, dichos modelos no eran escalables para problemas generales, tales como detección de imágenes, traducción de lenguajes, conversación en lenguaje natural, entre otros. Es así, que la incorporación de modelos de redes neuronales profundas, en combinación con el procesamiento en paralelo utilizando GPU (Graphical Processing Units), encaminó a la creación de la Inteligencia Artificial Generativa.

Asimismo, una de las empresas más importantes en IA Generativa (OpenAI, 2024), la define como “Sistemas de inteligencia artificial, que tienen la capacidad de crear contenido nuevo y original, como imágenes, música, texto o incluso diseños. Estos sistemas, utilizan algoritmos de aprendizaje automático para generar datos realistas y creativos, que no han sido vistos previamente, imitando la capacidad humana de crear”. Por otro lado, (Google, 2024), la define como “Una rama de la IA que se centra en la creación de nuevos datos, a partir de modelos estadísticos y de aprendizaje profundo. Dichos sistemas pueden generar contenido original e innovador, como imágenes de rostros humanos inexistentes, música compuesta automáticamente o textos creativos, generados por computadora”.

Los tipos de IA Generativa más comunes son:

1. Redes Generativas Adversativas (GANs, Generative Adversarial Networks): Son un tipo de arquitectura de red neuronal que consta de dos modelos distintos, uno generador y otro discriminador, que compiten entre sí. Es así, que el generador puede crear muestras de datos nuevas, mientras que el discriminador intenta distinguir

entre las muestras generadas y las muestras reales. A medida que ambos modelos se entrenan juntos, el generador mejora su capacidad para producir datos, cada vez más realistas.

2. Redes Neuronales Recurrentes (RNNs, Recurrent Neural Network): Son un tipo de red neuronal que puede procesar secuencias de datos, lo que las hace ideales para la generación de texto, música o secuencias de vídeo. Estas redes son capaces de aprender de secuencias pasadas, para generar información nueva y coherente.

3. Transformadores (Transformers): Son arquitecturas de redes neuronales que han demostrado ser altamente efectivas en tareas de procesamiento del lenguaje. Uno de sus modelos más conocido, son los modelos de lenguaje de gran tamaño (LLM).

También, para problemas de conversación en lenguaje natural, los tipos transformers LLM son más utilizados y se han aplicado en proyectos, tales como GTP, Llama y Google Antrophic. Asimismo, en la generación automática de contenido, traducción a diferentes idiomas, sistemas de recomendación en lenguaje natural, análisis de sentimientos, investigación de mercados y en elaboración de programas; en particular, ésta última se usa para generar, optimizar y depurar código. De tal forma, en el procedimiento propuesto se considera exclusivamente, la generación de código a partir de una consulta. En este caso, son dos tipos:

a) un primer código generado es en lenguaje SQL. Existen varios trabajos que combinan el uso de los LLM para generar código SQL; por ejemplo, Gaolis, es un prototipo que trabaja en conjunto con un motor de base de datos y optimiza consultas SQL, para proporcionar resultados de forma eficiente (Mohammed et al., 2023). Además, se propone escribir en lenguaje natural e implementar tres diferentes estrategias, para combinar y generar una consulta tipo SQL (Xiang, 2024). Por otro lado, existen enfoques para realizar consultas simples y genéricas, evitando mostrar el esquema de la base datos y dando como resultado un nuevo LLM, que mejora este problema específico (Youssef et al., 2024). Asimismo, en una extensión del trabajo anterior, se amplía el modelo para la creación de árboles sintácticos, haciendo más eficiente la consulta SQL. Al respecto, dichos árboles sintácticos se enfocan en la optimización de las consultas para bases de datos complejas. Cabe señalar, que, en este último no se pretende optimizar la consulta SQL, sino la experiencia de usuario para proporcionar resultados que sean útiles para la organización.

b) el segundo código generado, es en lenguaje Python, para crear una gráfica o una tabla usando la biblioteca Plotly; es así, que otros modelos de lenguaje, como ChatGPT, Llama y Anthropic, tienen la capacidad de procesar este tipo de consultas.

Considerando lo descrito, en este proyecto, se utilizó los modelos tipo Transformers, específicamente los LLM, para implementar un proceso de consultas en lenguaje natural, en bases de datos de instituciones educativas y facilitar la obtención de información.

Objetivo

El objetivo de este trabajo, es aplicar lenguajes de gran tamaño (LLM, Large Language Model) para mejorar la experiencia de usuario. Lo anterior, implementando un procedimiento, que utilice dichos modelos para realizar consultas a bases de datos, utilizando lenguaje natural.

2 Metodología

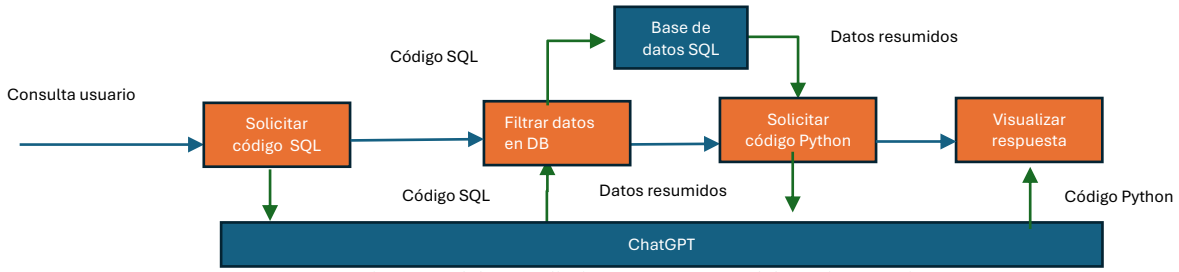

Figura 1. Diagrama del procedimiento propuesto. Elaboración propia.

En la figura 1, se observa el procedimiento, que se describe a continuación:

1. El usuario escribe en lenguaje natural la consulta sobre la información requerida. Además, selecciona una tabla especifica a consultar y envía el esquema de dicha tabla; por lo tanto, esto simplifica el proceso para obtener la consulta correcta.

2. El LLM GPT-3.5-turbo ejecuta la primera consulta y retorna el código SQL. Esto se hace por medio de la librería Langchain, que es una interfaz entre los LLM y código Python.

3. El código SQL obtenido se usa para obtener los datos resumidos.

4. El LLM GPT-3.5-turbo ejecuta la segunda consulta mediante Langchain, retorna el código Python y procesa la visualización, utilizando la librería Plotly.

Cabe señalar, que este procedimiento se implementó en una aplicación web utilizando la librería Fast API, la cual se integra fácilmente con las otras herramientas requeridas (Langchain y Plotly).

Por otro lado, se obtuvo un conjunto de datos de libre acceso del ámbito educativo (Munyiri, 2024), con la finalidad de realizar las pruebas para validar el procedimiento. También, se contó con la disponibilidad de bases de datos en formato SQL, así como las tablas que contienen datos de estudiantes, asignaturas, profesores, calificaciones, etc. Dicha estructura se puede observar en la figura 2. Además, se consideró utilizar la información contenida en la tabla de calificaciones y se agruparon todas las demás, en una única, tomando como pivote, la de calificaciones (Marks). Lo anterior, se puede observar en la tabla 1, donde se muestra un subconjunto de los datos ya agrupados. Esto se hace para contar con un esquema común, que pueda ser replicado en otras instituciones educativas.

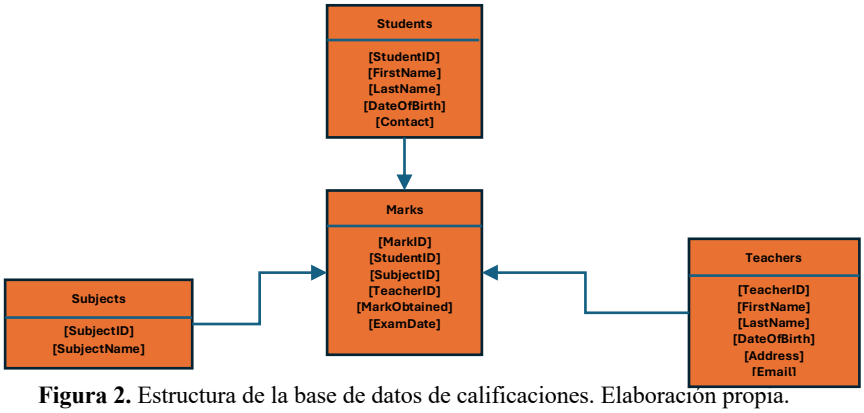

Figura 2. Estructura de la base de datos de calificaciones. Elaboración propia.

Tabla 1. Subconjunto de la tabla en la base de dato. Elaboración propia.

ID	Estudiante	Asignatura	Profesor	Calificación	Fecha
1	John Makori	Mathematics	David Mwangi	85	22/11/2023
2	John Makori	English	Grace Kimani	78	22/11/2023
3	John Makori	Science	Mohammed Abdi	92	22/11/2023
4	John Makori	History	Alice Wambua	88	22/11/2023

Por lo tanto, a) se convierten las consultas proporcionadas por el usuario, al lenguaje SQL, para extraer la información solicitada y b) se visualiza la información obtenida. Es así, que en lugar de requerir un desarrollador con conocimientos en SQL para a) y otro, con conocimientos de visualización de datos, para b); un usuario inexperto, podría obtener los mismos resultados.

3 Resultados

Utilizando el procedimiento, se generaron un total de 20 diferentes consultas relativas a los datos, para verificar que se obtenga el resultado correcto. En estas consultas, se verifica el uso de varias operaciones SQL, como las agrupaciones (GROUP BY), filtros (WHERE), limites (LIMIT), entre otros. Es así, que cada consulta se ejecutó tres veces para validar la consistencia de los resultados, siendo un total de 60 ejecuciones. A continuación, en la tabla 2 y en la figura 3, se ilustran los ejemplos de consultas y de visualización.

Tabla 2. Ejemplos de consultas. Elaboración propia.

Consulta	Query SQL	Resultado
¿Cuál es el estudiante con mejor calificación en la asignatura de science?.	`SELECT Student FROM marks_joined WHERE Asignatura = 'Science' ORDER BY Calificacion DESC LIMIT 1;`	`Error: Columna incorrecta Student`
Proporciona el promedio de calificación por profesor.	`SELECT Profesor, AVG(CAST(Calificacion AS FLOAT)) AS PromedioCalificacion FROM df GROUP BY Profesor`	`Error: Tabla no valida df`
Proporciona los cinco estudiantes, con mejor promedio.	`SELECT Estudiante, AVG(Calificacion) AS Promedio FROM marks_joined GROUP BY Estudiante ORDER BY Promedio DESC LIMIT 5`	`Correcto`
¿Cuál es la asignatura con el promedio de calificación más bajo y cual es dicho promedio?.	`SELECT Asignatura, AVG(Calificacion) AS promedio_calificacion FROM marks_joined GROUP BY Asignatura ORDER BY promedio_calificacion ASC LIMIT 1;`	`Correcto`
Proporciona los estudiantes que obtuvieron menos de 75 puntos, así como el nombre de la asignatura y sus calificaciones.	`SELECT Estudiante, Asignatura, Calificacion FROM marks_joined WHERE Calificacion < 75`	`Correcto`

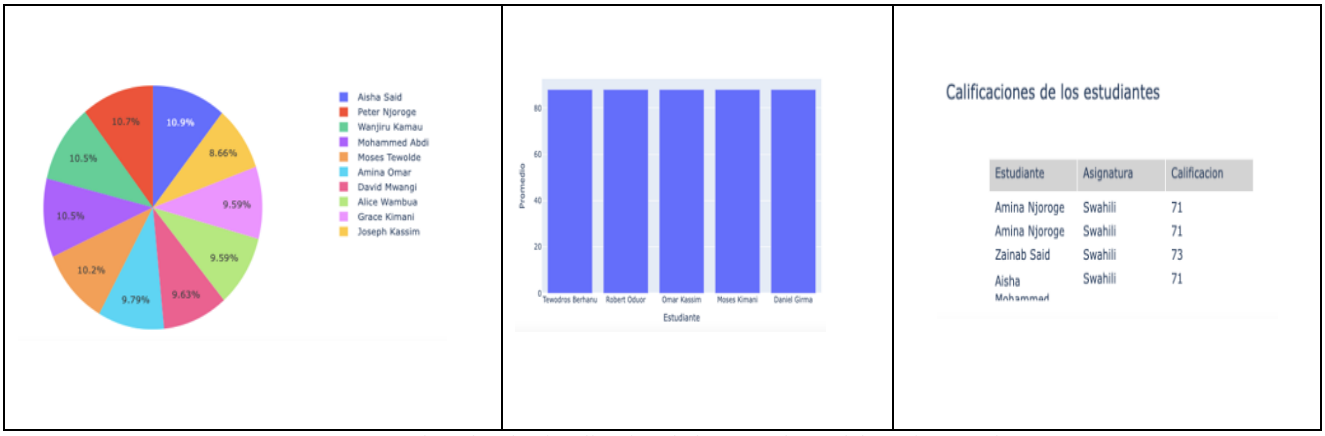

Figura 3. Ejemplos de visualización de las consultas. Elaboración propia.

Asimismo, la primera consulta al LLM tuvo 4 resultados incorrectos (dos en la primera ejecución y otra, en las dos primeras), de las de las 60 invocaciones. Por otro lado, la segunda consulta LLM (para generar el gráfico), no tuvo ningún error. Por lo tanto, la efectividad del procedimiento es del 93% (56 ejecuciones correctas de 60 realizadas) para las consultas SQL y el 100% para las visualizaciones.

4 Conclusiones

El procedimiento propuesto en este trabajo, ejecuta consultas en una tabla consolidada y visualiza información útil, mediante un ejemplo para organizaciones educativas. En las pruebas realizadas, se observó una eficiencia del 93% para generar consultas SQL y el 100% para visualizaciones. Tradicionalmente, cuando se necesita un nuevo requerimiento de información en un sistema de control escolar, los desarrolladores aplican herramientas SQL y de visualización; de tal manera, el beneficio de utilizar LLM, mediante lenguaje natural, es que puede generar todo el flujo de información, sin contar con un especialista. Además, al usar una estructura tipo estrella en el esquema de base de datos, el modelo puede extenderse fácilmente cuando se requiera añadir una nueva dimensión a la tabla de calificaciones consolidada (Marks); por ejemplo, cuando se añaden distintos criterios de acreditación en una asignatura o asignan asignaturas, a carreras o tipos; lo cual, se puede implementar, con mínimos cambios.

Por otro lado, en el contexto escolar, las calificaciones de un estudiante son consideradas como un dato personal y no deben publicarse sin el consentimiento expreso del individuo, siendo indispensable considerar la anonimidad. De tal manera, al enviar los datos resumidos, se propone implementar un mecanismo de ocultamiento de los datos personales, para preservar este derecho.

Finalmente, para proyectos futuros, se sugiere obtener retroalimentación directa de los usuarios, así como ampliar el esquema e implementar el procedimiento en diversas organizaciones, para probar su factibilidad en una mayor escala. Esto, también permitiría comprobar la escalabilidad de la estructura de base de datos, la generación de consultas más específicas y validar su utilidad.

Referencias

Munyiri Bernard Njiru (2024). School Database: Comprehensive Educational Data. https://www.kaggle.com/datasets/bernardnm/great-school.

Google (2024). Introducción a los modelos de lenguaje grandes. https://developers.google.com/machine-learning/resources/intro-llms?hl=es-419.

Mohammed Saeed, Nicola De Cao, Paolo Papotti: Querying Large Language Models with SQL. *ArXiv:2304.00472s* 2023.

Xiang, Lili. 2024. SQL Query Evaluation with Large Language Model and Abstract Syntax Trees. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 1890. https://doi.org/10.1145/3626253.3635408.

OpenAI (2024). How ChatGPT and our language models are developed. https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-language-models-are-developed?q=llm.

Youssef Mellah, Veysel Kocaman, Hasham Ul Haq, David Talby. Efficient schema-less text-to-SQL conversion using large language models. AIH 2024, 1(2), 96–106. https://doi.org/10.36922/aih.2661.