Universidad de La Habana
Facultad de Matemática y Computación

Enfoques Zero-Shot para la Extracción de Conocimiento a partir de Lenguaje Natural

Autor: **Rolando Sánchez Ramos**

Tutor: **Dr. Alejandro Piad Morffis**

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencias de la Computación

Noviembre de 2023

github.com/rolysr/nl2ql

Agradecimientos

Opinión del tutor

Dr. Alejandro Piad Morffis
Facultad de Matemática y Computación
Universidad de la Habana
Noviembre, 2023

Resumen

Esta tesis se centra en abordar la complejidad inherente a la consulta de bases de datos en forma de grafo, como Neo4J. Estas bases de datos a menudo requieren un conocimiento especializado en lenguajes de consulta, lo que limita su accesibilidad a un grupo reducido de usuarios con habilidades técnicas avanzadas. Para superar esta limitación, proponemos la aplicación del aprendizaje Zero-Shot, un enfoque innovador en el procesamiento del lenguaje natural. En esta investigación, se lleva a cabo un experimento basado en el modelo <variable>para traducir consultas de lenguaje natural a código *Cypher*. La evaluación se realiza utilizando el conjunto de datos de evaluación <variable>, que abarca una amplia variedad de ejemplos de consultas. Los resultados obtenidos, <variable>, establecen un punto de referencia esencial para el uso de modelos de lenguaje en la traducción de lenguaje natural a código *Cypher*.

Abstract

This thesis focuses on addressing the inherent complexity of querying graph databases like Neo4J. Such databases often require specialized knowledge of query languages, limiting accessibility to a small group of users with advanced technical skills. To overcome this limitation, we propose the application of Zero-Shot learning, an innovative approach in natural language processing. In this research, an experiment is conducted based on the <variable>model to translate natural language queries into *Cypher* code. Evaluation is carried out using the <variable>evaluation dataset, which encompasses a wide variety of query examples. The obtained results, <variable>, establish a crucial benchmark for the use of language models in translating natural language to *Cypher* code.

Índice general

Índice de figuras

Introducción

En la época actual, asistimos a un constante aumento en la producción de información en diversos formatos: visual, auditivo y textual, que abarca todos los ámbitos de la sociedad [4]. De manera particular, resulta sumamente intrigante la información generada a través del ingenio creativo y la investigación humana. Estos tipos de datos se almacenan debido a su relevancia y a la necesidad de acceder a ellos en el futuro, pudiendo optar por una organización estructurada o no. Sorprendentemente, solo alrededor del 20% de la información a nivel mundial se encuentra estructurada [1].

Las bases de conocimiento constituyen un tipo particular de bases de datos diseñadas para la administración del saber. Estas bases brindan los medios para recolectar, organizar y recuperar digitalmente un conjunto de conocimientos, ideas, conceptos o datos [2]. La ventaja fundamental de mantener la información de manera estructurada radica en su facilidad para ser consultada, ampliada y modificada. Debido a su utilidad y prevalencia, la recuperación de información a través de consultas en bases de conocimiento se ha convertido en una tarea esencial.

Es esencial que la información almacenada en bases de conocimiento adopte un formato adecuado para permitir búsquedas ágiles y precisas. Entre los formatos más comunes se encuentran los modelos de Entidad-Relación y el modelo Relacional. A pesar de ser enfoques más antiguos, el modelo Relacional (BDR) sigue siendo el más ampliamente utilizado en la actualidad [3]. No obstante, en ocasiones, las características específicas del problema demandan un formato más expresivo, y es en este punto donde las bases de datos orientadas a grafos (BDOG) [5] entran en juego.

Las BDOG han ganado progresivamente popularidad como una manera efectiva de almacenar información en los últimos años. Estas bases tienen la capacidad de modelar una diversidad de situaciones del mundo real al tiempo que mantienen un alto nivel de simplicidad y legibilidad para

los seres humanos. Las BDOG presentan numerosas ventajas en comparación con las bases de datos relacionales. Esto incluye un mejor rendimiento, permitiendo el manejo más rápido y eficaz de grandes volúmenes de datos relacionados; flexibilidad, ya que la teoría de grafos en la que se basan las BDOG permite abordar diversos problemas y encontrar soluciones óptimas; y escalabilidad, ya que las bases de datos orientadas a grafos permiten una escalabilidad eficaz al facilitar la incorporación de nuevos nodos y relaciones entre ellos. Ejemplo de un sistema de gestión de BDOG es *Neo4J* [**?**], a través del cual es posible construir instancias de este tipo de base de datos e interactuar con las mismas a través del lenguaje de programación *Cypher* [**?**], el cual posee una sintaxis declarativa similar a *SQL* [**?**].

Por otro lado, el avance en la comprensión del lenguaje natural se ha visto potenciado con el surgimiento de los grandes modelos de lenguajes (LLMs) [**?**] como GPT-4 [**?**] o LLaMA-2 [**?**], los cuales presentan una serie de habilidades emergentes como elaboración de resúmenes de textos, generación de código, razonamiento lógico, traducción lingüística entre otras [**?**]. Dichas herramientas constituyen modelos de *Machine Learning* entrenados con un gran volumen de datos, lo cual es posible gracias al número de parámetros con los que estos son configurados [**?**].

Usualmente, para el uso de los LLMs basta con ofrecerles como dato de entrada un texto (*prompt*), el cual describe la tarea que se espera que estos realicen. Además, son muchas las técnicas existentes para elaborar una entrada de calidad, esto con el objetivo de que la respuesta por parte de dicho modelo de lenguage ofrezca resultados alentadores al respecto, lo cual se conoce como *prompt engineering* [**?**]. Una técnica bastánte común es *Zero-Shot Learning* (ZSL) [**?**], la cual consiste en describirle a un LLM un procedimiento a realizar sin ofrecer de antemano ejemplos de cómo resolverlo, como por ejemplo, en tareas relacionadas con la generación de código, donde algunos de estos son capaces de generar algoritmos expresados en un lenguaje de programación formal a partir de una sentencia o consulta en lenguaje natural sin recibir como entrada del usuario algunos ejemplos de código, o especificaciones de cómo funciona el lenguaje objetivo a generar [**?**] [**?**].

En lo que respecta a la comprensión del lenguaje natural y su uso en consultas a bases de conocimiento, existen diversas vías llevadas a cabo y con resultados diversos, donde se hacen análisis sintácticos y semánticos sobre la consulta, muchas veces asistidos por diccionarios o mapas sobre la base de conocimiento en cuestión. Se usan modelos de paráfrasis como técnica de aumento de datos y finalmente Transformers [**?**] o incluso LLMs para llevar de la consulta ya curada al lenguaje de consulta formal o a un

lenguaje intermedio capaz de expresar a esta a alto nivel [**?**] [**?**] [**?**] [**?**].

Por las razones anteriormente expuestas, resulta interesante la investigación sobre la tarea de generación de código de consulta formal a partir de una sentencia en lenguaje natural mediante el uso de LLMs, especialmente el diseño e implementación un experimento capaz de demostrar las capacidades reales de estos para dicho acometido, lo cual designará la importancia de continuar el estudio de dichas herramientas con el objetivo de mejorar los sistemas de extracción de conocimientos en BDOG.

Problemática

Para utilizar el lenguaje de consulta formal *Cypher* se requiere de conocimientos básicos de programación, lo cual consume cierto tiempo y esfuerzo. Esto tiene como consecuencia que, solo aquellas personas con experiencia en el uso de lenguajes de programación puedan hacer uso de la mayoría de los sistemas de almacenamiento de datos desarrollados con esta tecnología y teniendo en cuenta la necesidad de poseer un conocimiento del dominio sobre el cual está construida la base de datos a consultar. Por lo tanto, llevar a cabo una mejora en las herramientas orientadas a democratizar dicho proceso permitiría hacer más rápido y eficiente dicho proceso de consulta en cuanto a tiempo y recursos computacionales. Debido a dicha situación, se propone una experimentación basada en un LLM capaz de traducir una consulta en lenguaje natural a un código en *Cypher*, donde a su vez se verifique la efectividad de este a partir de enfoques basados en ZSL, los cuales intuitivamente pueden ofrecer como resultado una cota inferior para la efectividad de sistemas desarrollados en base a dichos algoritmos de aprendizaje. Además, actualmente la implementación de sistemas de generación de código de consulta formal está principalmente orientada al lenguaje *SQL*, mientras que para el lenguaje *Cypher*, no existen suficiente estudios recientes que avalen la calidad de tales herramientas para dicho caso de uso.

Objetivos

Dadas las ideas anteriores, los objetivos principales del trabajo consistirá en diseñar e implementar una estrategia experimental capaz de evaluar la capacidad mínima de los LLMs para la consulta en lenguaje natural a bases de conocimiento estructuradas con independencia del dominio, para lo cual se empleará un enfoque basado en ZSL.

Para lograr los objetivos generales se trazaron los siguientes objetivos específicos:

1. Estudiar el estado del arte de los modelos de Aprendizaje Automático capaces de hacer predicciones de tipo texto-a-texto.
2. Analizar el trabajo de tesis sobre este tema anteriormente desarrollado en la facultad.
3. Implementar un modelo de Aprendizaje Automático capaz de convertir una consulta en lenguaje natural humano a un lenguaje formal que permita obtener datos a partir de una base de conocimiento.
4. Explorar las capacidades de enfoques Zero-Shot para la traducción de lenguaje natural al lenguaje Cypher.
5. Mejorar el sistema de evaluación de resultados permitiendo que el conjunto de datos de prueba y evaluación sea lo más realista posible y con una mayor complejidad.

Organización de la tesis

[Hablar sobre la estructuracion del documento]

Capítulo 1

Estado del Arte

Con el incremento constante de la cantidad de información generada en todo el mundo, la recuperación de información se ha convertido en un aspecto de creciente relevancia tanto en el ámbito industrial como en el académico. En consecuencia, la reducción del tiempo transcurrido entre el momento en que un usuario desea acceder a la información y el momento en que efectivamente puede hacerlo ha sido objeto de un número creciente de investigaciones científicas en los últimos años. Este capítulo se dedica a la evaluación de diversas estructuras de interfaces entre el ser humano y bases de conocimiento, las cuales tienen como objetivo abordar esta problemática.

Las Interfaces de Lenguaje Natural a Bases de Datos (NLIDB, por sus siglas en inglés) representan un campo de investigación dinámico centrado en facilitar las interacciones entre humanos y computadoras con bases de datos relacionales utilizando consultas en lenguaje natural. A lo largo de las últimas décadas, el desarrollo de NLIDB ha pasado por varias fases transformadoras, impulsadas por avances tecnológicos y metodológicos, así como por una creciente demanda de una mejor accesibilidad a las bases de datos.

Las etapas iniciales del desarrollo de NLIDB se caracterizaron por sistemas específicos de dominio. Estos sistemas fueron diseñados para trabajar dentro de áreas de conocimiento bien definidas, donde se utilizaba el procesamiento de lenguaje natural controlado para garantizar la comprensión de las consultas y la interacción con la base de datos. Por ejemplo, algunos trabajos pioneros [**?**] [**?**] demostraron NLIDBs que se adaptaban a dominios específicos, lo que los hacía altamente efectivos pero limitados en alcance. Del mismo modo, en años posteriores [**?**] [**?**] se continuó la exploración del

uso de interfaces de lenguaje natural controlado dentro de dominios de conocimiento particulares.

Otro enfoque durante esta fase implicó NLIDBs basados en reglas. Algunos sistemas propuestos al respecto [**?**] dependían de reglas predefinidas para traducir consultas en lenguaje natural en declaraciones *SQL* para la recuperación de datos en la base de datos. Si bien estos sistemas ofrecían ciertas ventajas, carecían de versatilidad para manejar una amplia gama de consultas de usuarios en diferentes dominios.

A medida que avanzaba la investigación en NLIDB, hubo un cambio hacia la independencia de dominio y la flexibilidad. Los sistemas recientes han buscado reducir la dependencia del conocimiento específico del dominio y las reglas. Algunos investigadores [**?**] [**?**]) han desarrollado NLIDBs que utilizan técnicas de aprendizaje supervisado, lo que los hace más adaptables a varios dominios y entradas de usuario. Además, un avance significativo se ha producido con la integración de redes neuronales profundas en el desarrollo de NLIDBs, donde varias investigaciones [**?**] [**?**] han demostrado el potencial del aprendizaje profundo en NLIDB, aprovechando vastos repositorios de texto y código para el entrenamiento. Este enfoque ha mejorado significativamente el rendimiento de NLIDB, permitiendo un procesamiento de consultas más natural y contextual.

Para el caso específico de NLIDB con respecto a BDOGs inicialmente se desarrollaron trabajos enfocados en técnicas similares a las empleadas para *SQL* [**?**] [**?**], donde se realizaba un preprocesamiento en la consulta dada, se aprovechaba la información ofrecida por el esquema [**?**] de la base de datos a consultar y finalmente dicho conocimiento era utilizado por un modelo de aprendizaje profundo entrenado sobre un conjunto de pares de lenguaje natural y lenguaje de consulta formal como por ejemplo *Cypher*.

Recientemente, los trabajos orientados a esta área de estudio han estado enfocados en dos metodologías principales:

1. Enfoques neurosimbólicos basados en representación intermedia de la consulta dada en lenguaje natural.

2. Enfoques basados en técnicas de *prompt engineering* mediante LLMs.

1.1. Enfoques neurosimbólicos basados en representación intermedia (IR, por sus siglas en inglés) de la consulta dada en lenguaje natural

Con respecto a la recuperación de información de una base de conocimiento con este enfoque, mediante consultas hechas en lenguaje natural se han hecho varias investigaciones científicas que se pueden agrupar bajo el patrón de la Figura 1.1.

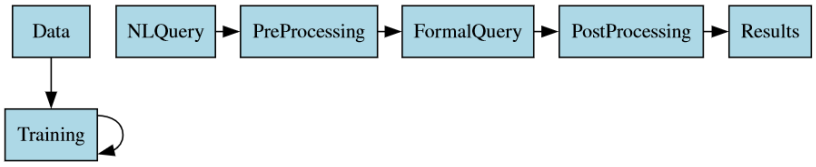

Figura 1.1: Secuencia de flujo representativa del Estado del Arte de traducción de Lenguaje Natural a Lenguaje Formal utilizando el enfoque neurosimbólico basado en (IR).

1.1.1. Conjuntos de entranamiento y evaluación

En cuanto a los datos para el entrenamiento existen dos opciones. Básicamente se puede buscar un conjunto de datos de referencia(*Benchmark*) en los que entrenar y probar el sistema, o se crea uno. La mayoría de las investigaciones existentes elijen la primera opción. Algunos de los *Benchmarks* más populares son:

- **WikiSQL**: WikiSQL [**?**] es el banco de datos más grande y más utilizado, contiene 26531 tablas y 80654 pares de consultas en lenguaje natural y lenguaje *SQL*. Las tablas se extraen de tablas *HTML* de Wikipedia. Luego, cada consulta en *SQL* se genera automáticamente para una determinada tabla bajo la restricción de que la consulta produce un conjunto de resultados no vacío.
- **Spider**: Este *Benchmark* [**?**] es un punto de referencia multidominio a gran escala con 200 bases de datos de 138 dominios diferentes y 10.181 pares de consultas.

- **MetaQA**: El conjunto de datos METAQA-Cypher, originalmente conocido como METAQA [**?**], contiene más de 400000 pares de preguntas y respuestas de múltiples pasos obtenidos de la base de conocimiento WikiMovies [**?**]. Mientras que investigaciones previas se han centrado principalmente en la anotación SPARQL [**?**], nuestra innovación implica reconfigurar METAQA en *Cypher*, estableciéndolo como un valioso punto de referencia para el aprendizaje de pocos ejemplos.

En el caso alternativo, se suelen usar las propias bases de conocimiento objeto de estudio para crear un conjunto de entrenamiento. Una de las técnicas empleadas para esto es Random Walk [**?**], en la que se hace un recorrido aleatorio sobre un subconjunto de las entidades y relaciones de la base de conocimiento, y se elaboran consultas artificiales que respondan a dichas entidades y relaciones [**?**].

1.1.2. Preprocesamiento

En general las investigaciones en el área realizan algún tipo de preprocesamiento a la consulta. Algunos realizan parafraseo para llevar la consulta a representaciones canónicas [**?**], en su lugar otros realizan un análisis morfológico léxico, con toquenización, lematización, eliminación de stopwords, tagueo de partes de la oración (*POS-taguing*), etc. [**?**]. También se encuentran las investigaciones que usan en esta etapa traducciones basadas en diccionarios especializados en el dominio, o de propósito general, al igual que ontologías, para "suavizar"vocablos difíciles de entender por el resto del sistema [**?**]. Además se usan técnicas como vectorización de palabras (*word2vector*), y transformación de la consulta a una representación en grafos [**?**].

1.1.3. Postprocesamiento

En la fase de Post-Procesamiento, se observan diversas enfoques en las investigaciones. La mayoría de los investigadores realizan un análisis semántico que implica la clasificación según tipos de datos, el uso de ontologías y bases de conocimiento externas para realizar mapeos [**?**] [**?**]. Algunos optan por convertir la consulta en una representación canónica, mientras que otros la transforman en un lenguaje intermedio antes de transpilarla al lenguaje objetivo [**?**]. También hay quienes la codifican directamente en forma de grafo, y algunos la convierten en embeddings [**?**].

1.1.4. Consulta

En la fase de construcción de la consulta, existen tres enfoques principales. El primero es un enfoque manual que implica la búsqueda y formateo de palabras clave como *"where"* y *"select"*, además del mapeo de atributos a tablas [**?**]. Otra vía se centra en el uso de modelos, incluyendo decodificadores y en algunos casos redes neuronales convolucionales [**?**]. También se emplea la construcción de la consulta formal mediante un compilador, especialmente cuando se ha utilizado un lenguaje intermedio entre el lenguaje natural y el formal de consulta[**?**].

1.2. Enfoques basados en técnicas de *prompt engineering* mediante LLMs

Con la creciente atención dada a los modelos de lenguaje a gran escala, estos se han convertido en un componente esencial en el procesamiento del lenguaje natural. A medida que aumenta el tamaño de los modelos preentrenados, también está cambiando gradualmente su uso. A diferencia de modelos como BERT [**?**] y T5 [**?**], que requieren un proceso de entrenamiento con una pequeña cantidad de datos, modelos como GPT-3 [**?**] requieren un diseño de un texto de entrada para generar resultados deseados. El reciente modelo de *ChatGPT* [**?**], que emplea Aprendizaje por Reforzamiento para la Retroalimentación Humana (RLHF) [**?**], simplifica el diseño de textos de entrada de calidad, lo que permite una mejor utilización de la capacidad de ZSL de modelos preentrenados a gran escala de manera conversacional. Debido a la sólida capacidad de dichos en la generación de código [**?**] y al hecho de que los modelos de generación de código suelen requerir una gran cantidad de datos anotados para producir buenos resultados [**?**], un modelo de generación de código de ZSL se considera fundamental.

Para la tarea específica de generar código de un lenguaje de consulta formal como *SQL* y *Cypher* se han utilizado distintos enfoques basados en varias de las principales técnicas de *prompt engineering*.

1.2.1. Zero-Shot Learning

Esta técnica se enfoca en la capacidad de un modelo para comprender y generar código en un lenguaje de consulta sin requerir ejemplos específicos de entrenamiento en ese lenguaje en particular. En otras palabras, el modelo puede realizar esta tarea "desde cero", sin conocimiento previo del

lenguaje. Algunos estudio interesantes se han realizado principalmente en la tarea de traducir lenguaje natural a lenguaje *SQL* [**?**] [**?**], los cuales permiten inferir la calidad mínima que estos modelos pueden alcanzar en la realización de dicha tarea [**?**].

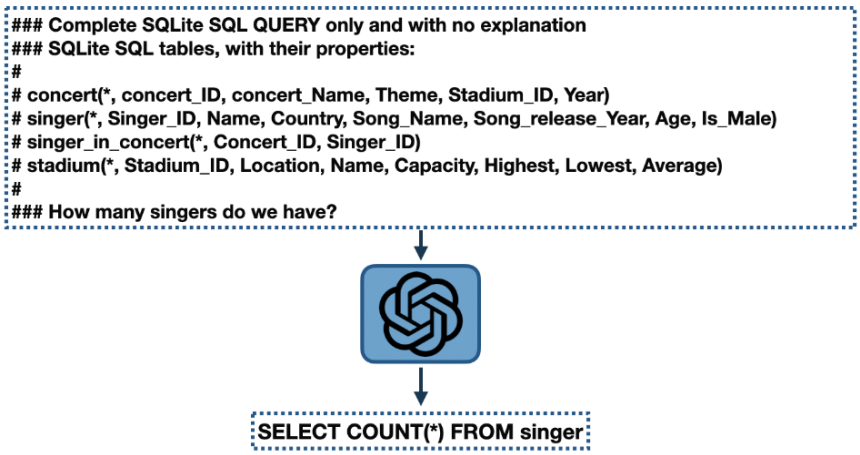

Figura 1.2: Ejemplo del flujo de trabajo de experimentos basados en ZSL para la traducción de lenguaje natural a código en *SQL*.

1.2.2. Few-Shot Learning

En contraste, el enfoque de Few-Shot Learning (FSL) se basa en la idea de que el modelo tiene acceso a un pequeño número de ejemplos (pocos ejemplos) en el lenguaje de consulta deseado para mejorar su capacidad de generar código en ese lenguaje. Esto puede ser especialmente útil cuando se necesita una adaptación rápida a un nuevo lenguaje o contexto. Tal y como muestran algunos resultados experimentales, este enfoque puede tener resultados superiores a varios modelos basados en *fine-tuning* [**?**].

Un ejemplo notable de esta aplicación es el modelo Codex [**?**], que ha demostrado ser un fuerte baseline en el *benchmark* Spider sin ninguna fase de entrenamiento. Además, se ha observado que proporcionar un pequeño número de ejemplos en el dominio en el prompt permite a Codex superar a los modelos de estado del arte cuyos parámetros han sido ajustados con pocos ejemplos de pocos dominios [**?**].

Además, se ha propuesto un marco de selección de demostraciones ODIS [**?**] que utiliza tanto ejemplos fuera de dominio como ejemplos generados sintéticamente en el dominio para construir demostraciones. Este

enfoque ha demostrado ser efectivo en comparación con los métodos de línea de base que se basan en una única fuente de datos [?].

En cuanto a los conjuntos de datos, existen varios conjuntos de datos de texto a SQL que han sido propuestos para evaluar el rendimiento de los modelos LLM. Algunos de estos conjuntos de datos incluyen CoSQL, TableQA, DuSQL, CHASE, y BIRD-SQL [?] [?] [?] [?] [?] [?].

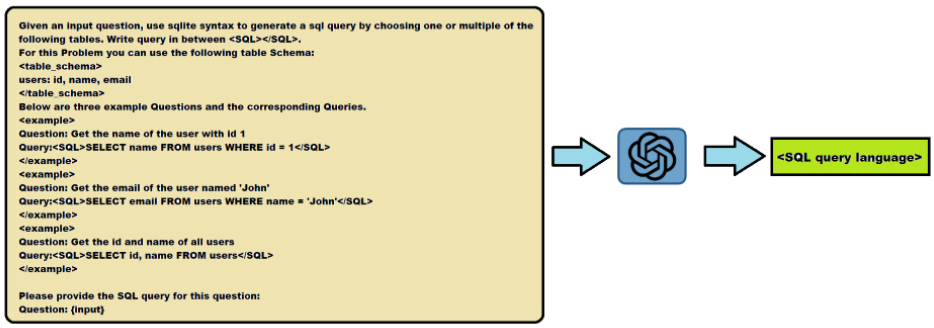

Figura 1.3: Ejemplo del flujo de trabajo de experimentos basados en FSL para la traducción de lenguaje natural a código en *SQL*.

1.2.3. Chain-of-Thought Prompting

El enfoque de la cadena de pensamiento (*Chain of Thought, CoT*) [?] en la conversión de texto a *SQL* ha demostrado ser una estrategia prometedora para mejorar la capacidad de los modelos de lenguaje de gran tamaño (LLMs) para realizar razonamientos complejos. Este enfoque se ha utilizado en varias investigaciones recientes para mejorar la capacidad de los LLMs para realizar tareas de razonamiento complejo, como la conversión de texto a SQL [?].

Un estudio propuso un nuevo paradigma para la generación de consultas *SQL* a partir de texto, llamado *Divide-and-Prompt*, que divide la tarea en subtareas y luego aborda cada subtarea a través de la cadena de pensamiento. Se presentaron tres métodos basados en la indicación para mejorar la capacidad de los LLMs para generar consultas *SQL* a partir de texto. Los experimentos mostraron que estas indicaciones guían a los LLMs para generar consultas *SQL* a partir de texto con mayor precisión de ejecución [?].

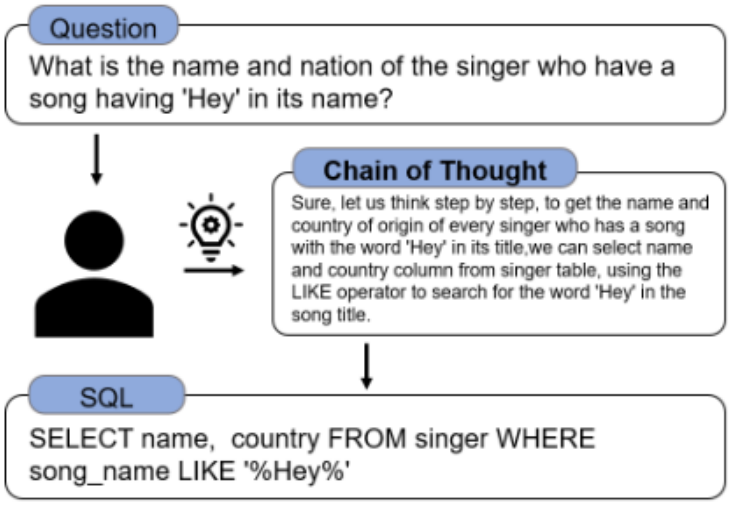

Figura 1.4: Ejemplo del flujo de trabajo de experimentos basados en CoT para la traducción de lenguaje natural a código en *SQL*.

1.2.4. *Fine-Tuning*

El entrenamiento (*fine-tuning*) de un modelo es un proceso que se utiliza para mejorar el rendimiento de un modelo de aprendizaje automático en una tarea específica. En el contexto de la conversión de texto a *SQL*, el ajuste fino puede ser utilizado para mejorar la precisión y la eficacia de los modelos de lenguaje de gran tamaño (LLMs) en la generación de consultas *SQL* a partir de texto. Se han realizado estudios [**?**]ropuso un método de *fine-tuning* para los modelos LLMs utilizando un conjunto de datos de texto a *SQL*, donde el método principal consistía en entrenar el modelo en el conjunto de datos de entrenamiento y luego ajustar el modelo en un conjunto de datos que contenía consultas *SQL* generadas por humanos. Este método demostró ser efectivo para mejorar la precisión de los modelos LLMs en la generación de consultas *SQL* a partir de texto [**?**].

Este enfoque constituye una poderosa herramienta en la conversión de lenguaje natural a *SQL*. Sin embargo, no es una solución mágica, ya que pocas organizaciones tienen conjuntos de datos de entrenamiento *NL-2-SQL* disponibles de manera inmediata. Algunos expertos consideran que las mejores arquitecturas se podrían lograr combinando modelos ajustados con agentes RAG (*Retrieval Augmented Generation*) [**?**].

1.3. Consideraciones generales

A pesar del uso comprobado de dichos enfoques, todavía existe una escasez de estudios orientados a la traducción de lenguaje natural a código de consulta a BDOG como por ejemplo *Cypher* [**?**], por lo tanto, es visible la necesidad de elaborar experimentos enfocados en dicha tarea, tomando como bases las ideas anteriormente expuestas. Debido a esto, resulta imprescindible desarrollar las primeras experimentaciones de la tarea en cuestión utilizando el enfoque ZSL, el cual en promedio permite obtener una cota inferior de qué tan efectivos pueden ser los LLMs con respecto a una tarea específica [].

Capítulo 2

Propuesta de Solución

En el presente capítulo se abordará la metodología seguida para diseñar el experimento propuesto en este trabajo **??**. Primeramente, se expondrá un marco teórico que formaliza la definición del problema a tratar, esto con el objetivo de presentar los conocimientos base tenidos en cuenta para los enfoques probados. Luego, se detallan los primeros acercamientos desechados **??**, argumentando las deficiencias de estos a la hora de arrojar resultados consistentes para la tarea que se desea desarrollar. Finalmente, se detalla la metodología definitiva a implementar, teniendo en cuenta la experiencia obtenida de las anteriores y mostrando su robustez para el análisis experimental [].

De forma general, el componente común para cada vía de solución constituye la presencia de un Gran Modelo de Lenguaje, pues representan los modelos más recientes utilizados para la tarea en cuestión; además, ofrecen resultados alentadores para el caso de traducción a lenguaje *SQL* según lo visto en la sección 1.2. Por lo tanto, tiene sentido probar su eficacia para traducir a código en *Cypher*, ya que ambos presentan similitudes como lenguajes formales declarativos para consultar bases de datos. Dicho modelo será analizado como una "caja negra" capaz de hacer tareas de traducción de lenguaje natural a una consulta semánticamente equivalente en el lenguaje *Cypher*.

Para cada vía de solución se deberá considerar el despliegue de un sistema de gestión de bases de datos para alguna BDOG, ya que es en este componente donde se almacenará la información a extraer por consultas en un lenguaje orientado a este tipo de almacenamiento. En el caso particular de este trabajo, se considerará el uso de *Neo4J*, con el cual se puede interactuar a partir del lenguaje *Cypher* ya mencionado. Por esto, es importante

considerar la implementación de un módulo intermedio para interactuar con una instancia del sistema de gestión *Neo4J*.

El enfoque de *prompt engineering* a utilizar será ZSL, por lo tanto, los textos de entrada que se le darán al modelo para la generación de código *Cypher* no contendrán ejemplos de pares de lenguaje natural con su correspondiente traducción al lenguaje de consulta objetivo. Por lo tanto, se tomarán algunas ideas experimentadas en el estado del arte para *SQL* vistas en la sección **??**.

2.1. Preliminares

2.1.1. Bases de Datos *Neo4J*

2.1.2. Lenguaje *Cypher*

2.1.3. Grandes modelos de Lenguaje

2.1.4. Definición del problema *Text-to-Cypher*

Dentro del ámbito de las bases de datos y las consultas que las acceden, existe una tarea particularmente compleja: traducir preguntas formuladas en lenguaje natural a un lenguaje de consulta estructurado como *Cypher*. Dada una pregunta Q, formulada en lenguaje cotidiano, y un esquema de base de datos S, el cual se compone a partir del tuplo N, A, R, donde encontramos múltiples nodos N (que representan instancias de una entidad en la base de datos), atributos C (tanto en nodos como en relaciones) y relaciones entre pares de nodos R. La problemática subyacente en el proceso de convertir *Text-to-Cypher* se centra en generar una consulta en lenguaje *Cypher* Y que sea equivalente y responda adecuadamente a la pregunta inicial Q realizada por un usuario humano.

2.1.5. LLM para resolver *Text-to-Cypher*

Con el auge de la inteligencia artificial y el aprendizaje automático, la tarea de convertir texto en lenguaje natural a código en *Cypher* ha sido recientemente abordada a través de técnicas modernas. En trabajos más recientes, algunos investigadores, **[?]** **[?]**, han abogado por formular esta tarea como un desafío de generación. Utilizando lo que se conoce como "*prompts*.º indicaciones P, es posible dirigir y guiar a un gran modelo de lenguaje M en esta labor. Este modelo, una vez entrenado, puede estimar una distribución de probabilidad sobre posibles consultas *Cypher* Y. De esta forma,

el modelo es capaz de generar, paso a paso y token por token, una consulta apropiada.

La fórmula subyacente para generar la consulta Y se estructura como:

$$P_M(Y|P,S,Q) = \prod_{i=1}^{|Y|} P_M(Y_i|P,S,Q,Y<i)$$

Para simplificar, $Y < i$ se refiere al fragmento inicial o prefijo de la consulta *Cypher* que se está construyendo. Mientras que $P_M(Yi|*)$ denota la probabilidad condicional asociada con la generación del i-ésimo token, considerando factores como el prefijo existente $Y < i$, la indicación P, el esquema S y la pregunta original Q.

Uno de los hallazgos más reveladores en el campo reciente es el concepto de aprendizaje en contexto (ICL, por sus siglas en inglés) [**?**], en el cual, grandes modelos de lenguaje pueden adaptarse y aprender de unos pocos ejemplos presentados en un contexto específico. Esta estrategia, defendida por varios investigadores [**?**] [**?**] [**?**], ha mostrado que los LLMs pueden abordar y dominar una amplia variedad de tareas complejas con una cantidad limitada de datos. Sin embargo, hay un equilibrio que mantener: agregar más ejemplos conlleva un aumento en los costos, tanto en términos de mano de obra para preparar esos ejemplos como en términos de costos de procesamiento y tokens al interactuar con APIs avanzadas como la de OpenAI. En este estudio, el foco recae en trabajar eficientemente con indicaciones al modelo de lenguaje sin requerir ejemplos adicionales.

2.2. Diseño de la información entrada al LLM

2.3. Selección del modelo

2.4. Enfoques considerados en la evaluación

2.4.1. Elaboración manual de consultas de prueba

2.4.2. Generación de consultas de prueba sintéticas

2.4.3. *Benchmark* orientado a *Cypher*

2.5. Propuesta de solución diseñada

Capítulo 3

Detalles de Implementación

Capítulo 4

Análisis Experimental

Conclusiones y Recomendaciones

Bibliografía

[1] Amazon. What is structured data? https://aws.amazon.com/es/what-is/structured-data/, 2023. (Citado en la página 10).

[2] Web Archive. Creación de una base de conocimiento. https://web.archive.org/web/20180315025341/http://es.ccm.net/faq/2158-organizacion-crear-una-base-de-conocimientos, 2023. (Citado en la página 10).

[3] SAP insights. ¿qué es el modelado de datos? https://www.sap.com/latinamerica/products/technology-platform/datasphere/what-is-data-modeling.html, 2023. (Citado en la página 10).

[4] Fundación MAPFRE. ¿cuánta información se genera y almacena en el mundo? https://www.fundacionmapfre.org/blog/cuanta-informacion-se-genera-y-almacena-en-el-mundo/, 2023. (Citado en la página 10).

[5] Wikipedia. Graph database. https://en.wikipedia.org/wiki/Graph_database, 2023. (Citado en la página 10).