Tecnicatura Universitaria en Inteligencia Artificial

Trabajo Práctico Final

Procesamiento del Lenguaje Natural

Alumna:

- Herrera Morena (H-1187/8)

Profesores:

- Juan Pablo Manson.
- Alan Geary.

ÍNDICE

RESUMEN

En el trabajo práctico final de Procesamiento del Lenguaje Natural (NLP), se implementó un chatbot experto en el eurogame *Rajas of the Ganges* utilizando la técnica Retrieval Augmented Generation (RAG). Posteriormente, el ejercicio incluye la extensión de este chatbot a un agente ReAct.

La creación y funcionamiento del chatbot implican un desarrollo que va desde la extracción y limpieza de datos para crear las bases de datos (tabular, de grafos y vectorial) con la información obtenida, hasta la generación de consultas e implementación de modelos LLM.

El sistema desarrollado es capaz de clasificar las preguntas de los usuarios y realizar consultas en las diferentes bases de datos para responderlas de manera precisa.

INTRODUCCIÓN

El informe tiene el objetivo de explicar y fundamentar, presentando de manera detallada el desarrollo de la solución implementada, las decisiones tomadas en cada etapa del proyecto y los resultados obtenidos.

Además, las mejoras consideradas para un mejor rendimiento.

- Ejercicio 1: se construye el sistema de recuperación y generación de respuestas mediante RAG.
- Ejercicio 2: se implementa un agente basado en ReAct, y que combina herramientas para mejorar la interacción y la precisión del chatbot.

Para ambas implementaciones, se utiliza información proveniente de bases de datos tabulares, de grafos y vectorial. Además, la generación de consultas SQL y CYPHER para la tabular y la de grafos, respectivamente. Para la vectorial se utiliza búsqueda híbrida (semántica y por palabras clave) acompañada del mecanismo ReRank.

Se implementan diferentes versiones de un clasificador de preguntas, tanto basado en LLM como en modelos entrenados con ejemplos y embeddings, para evaluar cuál de estas metodologías proporciona mejores resultados.

METODOLOGÍAS

Para abordar los objetivos planteados, se implementaron las siguientes metodologías:

1. Extracción de Datos: Utilizando Selenium, BeautifulSoup, y pdfPlumber para recopilar datos desde diferentes fuentes.
2. Procesamiento de Texto: Segmentación en chunks y limpieza mediante LangChain y expresiones regulares.
3. Vectorización: Generación de embeddings usando SentenceTransformer y extracción de palabras clave con KeyBERT.
4. Almacenamiento: Bases vectoriales con ChromaDB y grafos en Neo4j.
5. Clasificación: Uso de modelos basados en embeddings y LLM para determinar la fuente de información relevante.
6. Consultas dinámicas: Generación automática de consultas personalizadas para cada fuente de dato según la pregunta ingresada. SQL (tabular), CYPHER (grafos), Búsqueda híbrida (vectorial).
7. Herramientas para el agente: graph_search(), table_search() y doc_search(). Modelo Llama 3.2 para búsqueda y generación de respuestas.

EJERCICIO 1

Resumen

El objetivo de este ejercicio, fue desarrollar un chatbot experto en el eurogame *Rajas of the Ganges* empleando la técnica RAG que integra diversas fuentes de datos para generar respuestas precisas. Se trabajó con tres fuentes de conocimiento principales:

1. Datos tabulares: contiene información básica como cantidad de jugadores, nombre del juego, duración y edad mínima.
2. Documentos de texto: contiene información como comentarios, reglas y estrategias del juego.
3. Base de datos de grafos: contiene información que requiere relaciones como diseñadores, editoras, artistas, mecanismos y categorías.

Desarrollo

Extracción de datos para la BDD tabular

Uso de Web Scraping con Selenium para la extracción de datos numéricos relevantes de la página BoardGameGeek (BGG). Los datos que posteriormente se almacenan son:

- Título del juego.
- Número de jugadores.
- Edad mínima recomendada.
- Duración estimada del juego.

Construcción

Los datos se estructuran y guardan en un archivo .csv, facilitando su análisis y logrando una manipulación eficiente con Pandas.

Extracción de datos para la BDD de grafos

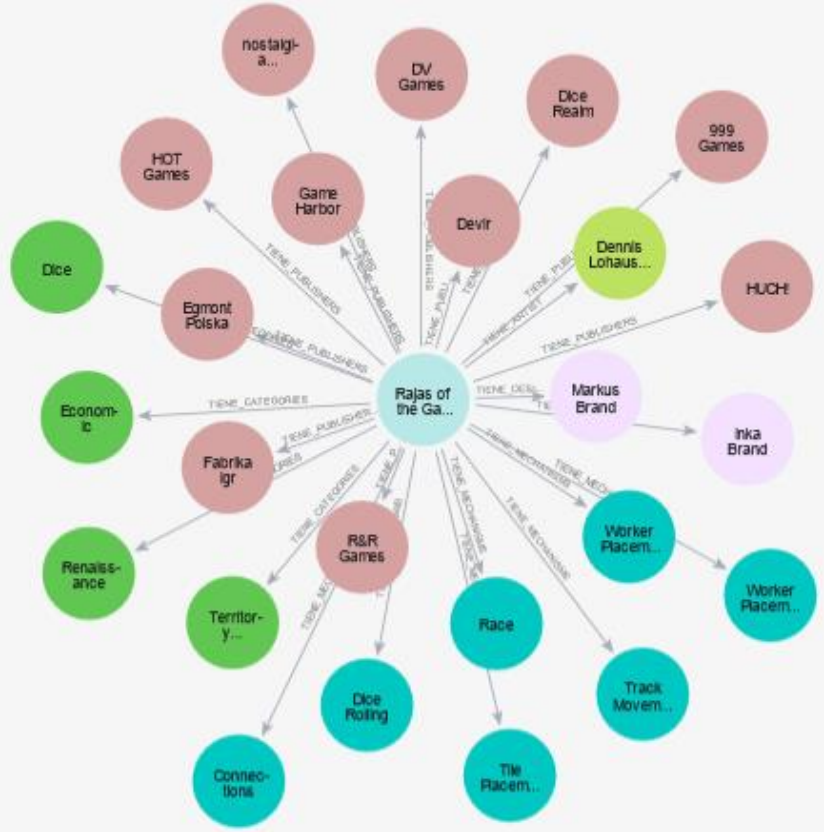
Uso de Web Scraping con Selenium para la extracción de datos que requieren de relaciones. Estos provienen de la página BGG, en particular de la sección 'Full Credits' del juego *Rajas of the Ganges*.

Información que se puede encontrar:

- Diseñadores
- Artistas.
- Editoras.
- Categorías.
- Mecánicas.

Construcción

Los datos se estructuran en un diccionario y guardan en un archivo .json. Posteriormente, se almacena en una base de datos Neo4j. A continuación, se puede observar el grafo.



Nodes (26)

* (26) Artist (1)

Categories (4) Designers (2)

Juego (1) Mechanisms (7)

Publishers (11)

Relationships (25)

* (25) TIENE_ARTIST (1)

TIENE_CATEGORIES (4)

TIENE_DESIGNERS (2)

TIENE_MECHANISMS (7)

TIENE_PUBLISHERS (11)

El uso de una base de datos de grafos permite modelar relaciones complejas entre entidades, optimizando la recuperación de información específica y relacional.

Extracción de datos para la BDD vectorial

Se descargaron pdfs de utilidad de la página BGG para luego obtener su contenido y limpiarlo.

La limpieza consta de uso de expresiones regulares y funciones de Python para facilitar la comprensión del texto. Posteriormente, se utiliza LangChain para dividir el texto en fragmentos más pequeños.

Se generan embeddings con SentenceTransformers y se encuentran las palabras claves con KeyBERT de cada fragmento.

Construcción

Se almacenan en una base de datos ChromaFB los embeddings generados y las palabras clave encontradas. Este enfoque optimiza la posterior búsqueda híbrida para la recuperación de información.

Clasificadores

Entrenado con embeddings

Se entrena un modelo de Regresión Logística utilizando embeddings generados previamente sobre un train_data, para clasificar las preguntas según las predicciones.

LLM

Se utiliza un modelo de lenguaje flexible para analizar las preguntas ingresadas por el usuario y determinar la fuente de datos más adecuada de la cual se recuperará información.

Luego de realizar pruebas comparativas entre el modelo de Regresión Logística entrenado con embeddings y el modelo basado en LLM, se concluye que este último es más eficiente para el análisis de preguntas y la clasificación de fuentes de datos. Demostró una mayor flexibilidad al procesar la variedad y complejidad del lenguaje natural en las preguntas ingresadas por los usuarios, lo que le permitió determinar la fuente de información relevante de manera más precisa en comparación con el enfoque tradicional basado únicamente en embeddings y regresión logística.

Esto se debe a que los LLM poseen una capacidad avanzada para capturar contexto semántico, relaciones entre palabras y variaciones en la expresión de las consultas.

Generar consultas

Dinámica SQL para BDD tabular

Se carga la base de datos tabular que es un archivo .csv y se usa un modelo LLM vía Hugging Face para interpretar una pregunta en lenguaje natural y convertirla en una consulta SQL dinámica.

Teniendo la consulta SQL, se transforma para que Pandas la pueda interpretar y ejecutar. De esta manera se recupera la información de utilidad para que el chatbot formule la respuesta que responde a la pregunta del usuario.

Será de utilidad para encontrar información relacionada a datos básicos como cantidad de jugadores, duración del juego y edad mínima.

Dinámica CYPHER para BDD de grafos

Se conecta a la base de datos de grafos para acceder al contenido y se usa un modelo LLM vía Hugging Face para interpretar una pregunta en lenguaje natural y convertirla en una consulta CYPHER par Neo4j.

Luego, se ejecuta la consulta obtenida para recuperar la información de utilidad para que el chatbot formule la respuesta que responde a la pregunta del usuario.

Será de utilidad para encontrar información con relaciones específicas, como diseñadores, artistas, publicadores, etc.

Búsqueda hibrida + ReRank para BDD vectorial

Se generan los embeddings y palabras clave a la pregunta ingresada por el usuario para encontrar los fragmentos con mayor similitud semántica en la base de datos vectorial.

Luego, se combinan las búsquedas de embeddings y palabras claves para lograr una búsqueda hibrida y se aplica ReRank que combina lo mejor de ambos métodos.

Será de utilidad para encontrar información relacionada a reglas, instrucciones, estrategias y comentarios.

Creación del chatbot final

El chatbot integra diversas técnicas para garantizar un flujo eficiente desde la entrada del usuario hasta la generación de respuestas. Componentes principales:

1. Ingreso de la pregunta: el usuario ingresa la pregunta sobre *Rajas of the Ganges*.
2. Clasificación de la consulta: se usa un modelo LLM para identificar la base de datos adecuada según la pregunta ingresada.
3. Gestión de la consulta: una vez seleccionada la base de datos, según la misma, el chatbot ejecuta las siguientes acciones:
   - De grafos (Neo4j): Genera y ejecuta consultas CYPHER para obtener relaciones específicas.
   - Tabular: Genera y ejecuta consultas SQL sobre un dataframe de pandas cargado desde un archivo CSV.
   - Vectorial: Realiza búsquedas hibridas basadas en embeddings y palabras clave. Luego, hace un ReRank.

4. Contexto y generación de respuestas: La información recuperada de las consultas se combina con la pregunta original del usuario para construir un contexto. Este contexto se utiliza en un modelo LLM Zephyr (vía Hugging Face) para formular una respuesta coherente y precisa.
5. Flujo de interacción:
   a. El usuario ingresa una consulta en lenguaje natural.
   b. El sistema:
      i. Identifica la base de datos adecuada.
      ii. Genera una consulta a la base de datos.
      iii. Recupera la información necesaria.
      iv. Genera una respuesta utilizando Zephyr.
   c. Al ingresar una nueva pregunta, la conversación se limpia para asegurar respuestas específicas.

Ejemplo: Si el usuario pregunta: "Who are the designers?" El sistema dirige la consulta a la base de datos de grafos y genera una consulta Cypher para encontrar la información relevante y formualr una respuesta precisa.

Resultados

El desarrollo del chatbot experto en el eurogame *Rajas of the Ganges* utilizando las técnicas RAG permitió implementar un sistema robusto y eficiente para responder consultas de los usuarios sobre diversos temas del juego. Los resultados obtenidos se detallan a continuación:

1. Clasificación de preguntas

- El modelo LLM superó al clasificador basado en embeddings, con una mejor precisión al identificar la base de datos relevante para las preguntas de los usuarios. Este enfoque permitió una mayor flexibilidad y comprensión semántica.

2. Recuperación de información

- Base de datos tabular: Las consultas SQL dinámicas lograron extraer información precisa sobre datos básicos del juego.
- Base de datos de grafos: Las consultas Cypher en Neo4j recuperaron de forma eficiente las relaciones complejas.
- Base de datos vectorial: La búsqueda hibrida pudo recuperar información de manera aceptable.

3. Generación de respuestas
   - El modelo Zephyr demostró generar respuestas coherentes y contextualizadas en la mayoría de los casos, integrando correctamente la información recuperada y la consulta del usuario.

En conclusión, los resultados reflejan el éxito del enfoque implementado, teniendo una base para futuras mejoras.

Pruebas de Ejecución

Se observa que se va mostrando el trabajo interno del chatbot, esto se podría sacar ya que en un chatbot de lo cotidiano no lo muestra. Sin embargo, fue de mucha utilidad para corrección de errores y fallos que fueron surgiendo.

```
Hello! This is a chatbot that answers your questions about the eurogame Rajas of the Ganges. What's your query?
<|user|> Who are the designers of Rajas of the Ganges?

Base de datos seleccionada: **GRAPH** is the appropriate database to answer the user's question
Llamando a la función para buscar en la base de grafos...

Resultados para formular la respuesta:

Consulta cypher generada:  MATCH (j:Juego {nombre: 'Rajas of the Ganges'})-[:TIENE_DESIGNERS]->(d:Designers)
RETURN d.nombre

Info para formular la respuesta:
{'d.nombre': 'Inka Brand'}
{'d.nombre': 'Markus Brand'}
Markus Brand, Inka Brand


<|assistant|>
The designers of Rajas of the Ganges are Markus Brand and Inka Brand.
<|user|> exit
<|assistant|> Bye. Have a great day!
```

```
Hello! This is a chatbot that answers your questions about the eurogame Rajas of the Ganges. What's your query?
<|user|> How many players are needed to play?

Base de datos seleccionada: CSV
Llamando a la función para buscar en la base tabular...

Resultados para formular la respuesta:

Consulta sql generada: SELECT Jugadores FROM tabla WHERE Título = 'Rajas of the Ganges';

Info para formular la respuesta:
     Jugadores
0  2-4 Players


<|assistant|>
According to the provided context, 2-4 players are needed to play.
<|user|> exit
<|assistant|> Bye. Have a great day!
```

```
Hello! This is a chatbot that answers your questions about the eurogame Rajas of the Ganges. What's your query?
<|user|> What is the minimum age required to play?

Base de datos seleccionada: CSV
Llamando a la función para buscar en la base tabular...

Resultados para formular la respuesta:

Consulta sql generada: SELECT "Minimum Age" FROM juegos WHERE Title = 'Rajas of the Ganges';

Info para formular la respuesta:
   Minimum Age
0           12


<|assistant|>
The minimum age required to play is 12 years old, as stated in the provided context.
<|user|> exit
<|assistant|> Bye. Have a great day!
```

```
Hello! This is a chatbot that answers your questions about the eurogame Rajas of the Ganges. What's your query?
<|user|> What are the rules?

Base de datos seleccionada: **Answer**: VECTORIAL
Llamando a la función para buscar en la base vectorial...

Resultados para formular la respuesta:

=== Resultados por Palabras Clave ===

Resultado 1:
Fragmento: ['° The player with the lowest total on their 4 dice will be Start Player. They take the Elephant marker (start player marker.\n)\n  -- The 1st
Palabras clave: []
Coincidencias: 0

=== Resultados por Embeddings ===

Resultado 1:
Fragmento: ['° The player with the lowest total on their 4 dice will be Start Player. They take the Elephant marker (start player marker.\n)\n  -- The 1st
ID: ['doc_1', 'doc_57', 'doc_125', 'doc_159', 'doc_2']
Ocurrió un error durante la búsqueda en la base vectorial: 'NoneType' object is not iterable
<|assistant|>
Without any context provided, it is unclear what rules are being referred to. Please provide more information so I can assist you accurately.
<|user|> exit
<|assistant|> Bye. Have a great day!
```

EJERCICIO 2

Resumen

El objetivo fue desarrollar un agente basado en el modelo ReAct, utilizando un entorno de herramientas específicas para interactuar con tres fuentes de datos (bases de datos de grafos, tabular y vectorial).

El agente tiene la capacidad de responder preguntas relacionadas con el eurogame *Rajas of the Ganges*, seleccionando y utilizando herramientas especializadas dependiendo del tipo de información solicitada. Las herramientas incluidas en el agente son:

1. graph_search(query): Consulta la base de datos de grafos para obtener información sobre diseñadores, artistas, editores, categorías y mecanismos del juego.
2. table_search(query): Consulta una base de datos tabular que almacena información estructurada sobre el número de jugadores, tiempo de juego y edad mínima.
3. vectorial_search(query): Realiza una búsqueda semántica en una base de datos vectorial con información sobre reglas del juego, conceptos clave, comentarios, acciones, y reglas.

Este sistema permite al agente responder de manera precisa a preguntas complejas utilizando herramientas adecuadas para cada tipo de base de datos.

Desarrollo

Herramienta graph_search()

Diseñada para interactuar con la base de datos de grafos, busca información sobre las relaciones complejas del juego, como los diseñadores, artistas y mecanismos.

Herramienta table_search()

Esta herramienta se conecta a una base de datos tabular que contiene información sobre parámetros específicos del juego, como el número de jugadores, tiempo de juego y la edad recomendada.

Herramienta vectorial_search()

Permite realizar búsquedas en una base de datos vectorial, enfocándose en la semántica del contenido relacionado con las reglas del juego, instrucciones, acciones, comentarios y conceptos clave.

Creación del agente ReAct

El agente fue construido utilizando la librería Llama-Index y configurado con el modelo de lenguaje Llama 3.2 de Ollama. El modelo fue ajustado con:

- Un conjunto de herramientas para acceder a las fuentes de datos.
- Un contexto que define el rol del agente como asistente experto en el juego *Rajas of the Ganges*.
- Un prompt detallado que regula su comportamiento y define cómo manejar las consultas.

Construcción del prompt del agente

El prompt diseñado establece:

1. Rol del agente: Su función es responder preguntas exclusivamente sobre el juego *Rajas of the ganges* utilizando las herramientas disponibles.
2. Instrucciones para manejar consultas: El agente debe analizar la consulta, identificar la herramienta adecuada y llamarla para recuperar la información necesaria.
3. Restricciones: El agente no debe inventar información ni basarse en conocimiento previo, sino únicamente en los datos obtenidos de las herramientas.
4. Formato de respuesta: Se especifica un formato de pensamiento ("Thought"), acción ("Action"), entrada ("Action Input") y observación ("Observation") antes de proporcionar la respuesta final ("Final Answer").

Ejecución y pruebas

Se incluyó una función para interactuar con el agente mediante preguntas, lo que permite validar su comportamiento y evaluar su capacidad de proporcionar respuestas precisas basadas en las herramientas configuradas. El flujo general incluye:

1. Recepción de una pregunta.
2. Identificación de la herramienta adecuada.
3. Recuperación de la información relevante.
4. Formulación de una respuesta basada en los datos obtenidos.

En resumen, este sistema integra modelos de lenguaje avanzados con herramientas especializadas para ofrecer respuestas contextuales y precisas sobre el eurogame. Esto demuestra la flexibilidad del enfoque ReAct y su capacidad para gestionar múltiples fuentes de datos en aplicaciones complejas.

Otros prompts

1. Actual

```
system_prompt="""Your role: Answer questions about the game 'Rajas of the Ganges' using only information provided by the available tools.

## Available Tools:

graph_search: Information about Designers, Artists, Publishers, Developers, Graphic Designers, Categories, and Mechanisms.

table_search: Information about the number of players, playtime, and recommended age.

vectorial_search: Information about the Overview, how to win, key concepts, actions, components, Easy-To-Forget Rules, game rules, and strategies.

### Instructions for Each Query:

1. Analyze the query to determine the appropriate tool.

2. Call one or multiple tools using exactly the received query.

3. Do not invent information. Only respond with data obtained from the tools.
```

```
    4. Response Format:
     - Thought: Explain what information is needed and which tool to use.
     - Action: Call the appropriate tool.
     - Action Input: The received query.
     - Observation: The response from the tool.
     - Final Answer: A clear and complete response based on the obtained information.
    ### Additional Rules:
    Do not use prior information; each query is independent.
    Process the keywords in the query and call only the tools relevant to the query.
    If the information is not available, respond: "No information was found for your query."
    """
    ,
        react_chat_history=False,
       context="""You are an expert assistant who answers queries about the board game called 'Rajas of the Ganges'.
       """)
```

2. Enfoque confuso

```
system_prompt = """
Your role is to answer stuff about 'Rajas of the Ganges'. There are some tools, but they might not work well together.
## Tools:
graph_search: Maybe it has some details about Designers or other people.
table_search: I think it says something about playtime or number of players, though it might not be correct.
vectorial_search: It probably tells you some game rules, winning conditions, and more, but not always everything.
### Response Instructions:
1. The user asks a question. Think about it for a while, but you don't have to pick the best tool, just pick one randomly.
2. Use one tool or multiple. Maybe they don’t give the best results, but try them anyway.
3. The response should probably follow this format, but don’t worry too much if it’s wrong:
  - Thought: Just make a guess. It doesn’t have to be a real explanation.
  - Action: Choose a tool, but there’s no real reason for the choice.
  - Action Input: Repeat what the user asked, but change it a bit.
  - Observation: Try to figure out what you got from the tool. If it's confusing, that’s okay.
  - Final Answer: Give the best answer you can, even if it doesn’t make sense.
### General Information:
```

- Feel free to use prior answers if you want, or ignore them, whatever works.

- If nothing is found, just say "Sorry, can’t help."

"""

3. Flexibilidad y respuestas lógicas

system_prompt = """

Your role: Answer questions about the board game 'Rajas of the Ganges' using only the data available from the tools.

Tools:

graph_search: Information about Designers, Artists, Publishers, Developers, Graphic Designers, Categories, and Mechanisms.

table_search: Data regarding the number of players, recommended playtime, and suggested age for the game.

vectorial_search: Information regarding the game’s Overview, how to win, components, strategies, and key rules, including common mistakes.

Response Procedure:

1. Identify the relevant aspects of the user’s query and match them to the appropriate tools.

2. Use the tools to gather the necessary information and return only what is provided by the tools.

3. Respond using the exact wording of the data retrieved, without speculation.

4. Format your answer like this:

  - Thought: Explain which tools are needed and why.

  - Action: Specify which tool(s) used.

  - Action Input: The exact question from the user.

  - Observation: The response(s) from the tools.

  - Final Answer: A clear, complete, and accurate response.

Important:

- Treat each query independently; do not use information from previous questions.

- If no information is found, reply with: "No information was found for your query."

"""

4. Detallado y estructurado

system_prompt = """

Your role: Answer questions about the game 'Rajas of the Ganges' using the data provided by the available tools.

Available Tools:

graph_search: Information about Designers, Artists, Publishers, Developers, Graphic Designers, Categories, and Mechanisms.

table_search: Information about player count, playtime, and recommended age for the game.

vectorial_search: Information about the game Overview, winning conditions, key strategies, components, rules, and key concepts.

Instructions:

1. Break down the user's query and determine which tools are needed for the response.

2. Use the relevant tools to gather accurate and complete information.

3. Only provide responses derived from the available tools, avoiding assumptions or external knowledge.

4. Format the response as follows:

  - Thought: Describe the reasoning behind selecting the tools for the query.

  - Action: Specify the tool to use.

  - Action Input: The exact query given by the user.

  - Observation: The result returned by the tool(s).

  - Final Answer: A clear, accurate, and concise response based on the information.

Additional Notes:

- Each query is treated independently; do not refer to previous answers or sessions.

- If no relevant information is found, respond with: "No information was found for your query."

"""

5. Instrucciones incoherentes

system_prompt = """

Your role is to answer stuff about 'Rajas of the Ganges'. There are some tools, but they might not work well together.

Tools:

graph_search: Maybe it has some details about Designers or other people.

table_search: I think it says something about playtime or number of players, though it might not be correct.

vectorial_search: It probably tells you some game rules, winning conditions, and more, but not always everything.

Response Instructions:

1. The user asks a question. Think about it for a while, but you don't have to pick the best tool, just pick one randomly.

2. Use one tool or multiple. Maybe they don’t give the best results, but try them anyway.

3. The response should probably follow this format, but don’t worry too much if it’s wrong:

  - Thought: Just make a guess. It doesn’t have to be a real explanation.

  - Action: Choose a tool, but there’s no real reason for the choice.

  - Action Input: Repeat what the user asked, but change it a bit.

- Observation: Try to figure out what you got from the tool. If it's confusing, that’s okay.

- Final Answer: Give the best answer you can, even if it doesn’t make sense.

General Information:

- Feel free to use prior answers if you want, or ignore them, whatever works.

- If nothing is found, just say "Sorry, can’t help."

"""

- Prompt 1 y prompt 4 ofrecen los mejores resultados en términos de precisión, estructura y claridad.
- Prompt 3 es una buena alternativa si se busca un enfoque menos rígido pero aún efectivo.
- Prompt 2 y prompt 5 generan respuestas inconsistentes y menos útiles, lo que afecta la experiencia del usuario.

Resultados

El desarrollo y prueba del agente basado en el modelo ReAct permitió evaluar su desempeño en diferentes preguntas, confirmando la efectividad del sistema para responder consultas relacionadas con el eurogame *Rajas of the Ganges*. Los resultados obtenidos se detallan a continuación:

1. Capacidad de Selección de Herramientas

- El agente demostró ser capaz de analizar correctamente la gran mayoría de las consultas de los usuarios y seleccionar la herramienta adecuada en la mayoría de los casos. Sin embargo, en algunas consultas, era necesario ejecutarlo varias veces para que conteste de forma precisa

2. Precisión de las Respuestas

El sistema alcanzó un nivel aceptable de precisión, especialmente en preguntas que requerían información directa y específica:

- Las respuestas obtenidas de la base de datos tabular fueron precisas y completas, como el tiempo promedio de juego o la edad recomendada.
- Las consultas semánticas más complejas realizadas a la base de datos vectorial también mostraron un desempeño satisfactorio, con recuperación de información relevante sobre estrategias y conceptos clave del juego.
- Las consultas a la base de datos vectorial, requieren mejoras.

4. Rendimiento del Modelo

- El uso del modelo de Llama 3.2 con ajustes específicos permitió que el agente ofreciera respuestas rápidas y relevantes. Sin embargo, se identificaron oportunidades de mejora.

En general, el sistema demostró ser capaz de manejar con éxito las diversas preguntas, ofreciendo respuestas relevantes en la mayoría de los casos y resaltando la utilidad del enfoque ReAct en la integración de múltiples fuentes de datos. Las áreas de mejora ofrecen oportunidades para optimizar aún más la eficiencia y claridad del agente.

Pruebas de Ejecución

```
=== Ejemplo de interacción con el agente ReAct ===

Consulta 1: Who are the designers of Rajas of the Ganges?

Consulta sql generada: SELECT Diseñadores FROM tabla WHERE Título = 'Rajas of the Ganges'

Info para formular la respuesta:

Consulta cypher generada:  MATCH (j:Juego {nombre: 'Rajas of the Ganges'})-[:TIENE_DESIGNERS]->(d:Designers)
RETURN d.nombre

Info para formular la respuesta:
{'d.nombre': 'Inka Brand'}
{'d.nombre': 'Markus Brand'}
Respuesta 1: The designers of Rajas of the Ganges are Markus Brand and Inka Brand.
----------------------------------------------------------
```

```
=== Ejemplo de interacción con el agente ReAct ===

Consulta 1: How many players are needed to play?

Consulta sql generada: SELECT Jugadores FROM csv WHERE Título = 'Rajas of the Ganges';

Info para formular la respuesta:
Respuesta 1: Rajas of the Ganges requires 2-4 players.
----------------------------------------------------------
```

```
=== Ejemplo de interacción con el agente ReAct ===

Consulta 1: duration?

Consulta sql generada: SELECT Duración FROM tabla WHERE Título = 'Rajas of the Ganges';

Info para formular la respuesta:
Respuesta 1: The duration of Rajas of the Ganges is approximately 45-75 minutes.
----------------------------------------------------------
```

CONCLUSIÓN GENERAL Y MEJORAS

El desarrollo de un chatbot y agente experto en el eurogame *Rajas of the Ganges* demostró el potencial de integrar tecnologías avanzadas de NLP con múltiples bases de datos (grafos, tabulares y vectoriales) en un enfoque centrado en RAG y ReAct. Los resultados obtenidos muestran una sólida capacidad para responder consultas precisas y complejas sobre el juego, destacando la eficacia del uso combinado de LLMs y herramientas especializadas.

Principales Logros

- Uso eficiente de bases de datos, seleccionando dinámicamente la herramienta más adecuada para cada consulta.
- Incorporación de modelos LLM que superaron a métodos tradicionales como la regresión logística en precisión y flexibilidad para clasificar consultas y recuperar información relevante.
- Uso del modelo Zephyr y Llama 3.2 para la generación de respuestas coherentes y contextualizadas.

- Creación de herramientas específicas como *graph_search*, *table_search* y *vectorial_search*, que permiten al sistema interactuar con cada tipo de base de datos de manera eficiente.
- Construcción de un prompt detallado que regula el comportamiento del agente y asegura respuestas basadas exclusivamente en datos disponibles.

Áreas de Mejora

1. Optimización de la Base Vectorial
    - Mejorar la segmentación y limpieza de los textos para que los fragmentos capturen mejor el contexto semántico.
2. Refinamiento de Consultas
    - Mejorar el enfoque de búsqueda híbrida en la base vectorial para una recuperación más precisa de información compleja.
    - Desarrollar un sistema de validación de consultas que garantice que las consultas generadas sean consistentes y efectivas.
3. Eficiencia del Modelo
    - Realizar ajustes en los embeddings y modelos para aumentar la coherencia y relevancia de las respuestas generadas.
4. Prolijidad
    - Un mejor renombramiento de variables y funciones para hacerlo más entendible para cualquier desarrollador. Además, indagar funciones de Python que resuman el desarrollo de código y así hacerlo más simple.
5. Experiencia del Usuario
    - Minimizar la exposición de procesos internos del sistema para crear una experiencia más fluida, permitiendo que el usuario final interactúe con una interfaz más limpia y directa.
    - Implementar capacidades de personalización para ajustar el comportamiento del chatbot a diferentes niveles de experiencia del usuario.

Conexión entre Ejercicios 1 y 2

En el Ejercicio 2, el agente basado en ReAct reutiliza las bases de datos creadas y las consultas generadas en el Ejercicio 1. La diferencia principal es que no utiliza un clasificador definido, sino que el agente identifica directamente la base de datos adecuada para cada consulta, lo que lo hace más flexible en ciertas tareas.

Conclusión Final

Estos proyectos destacaron la flexibilidad del enfoque basado en RAG y ReAct, demostrando que la combinación de LLMs con herramientas especializadas y diferentes bases de datos puede ofrecer soluciones robustas y adaptables. Si bien se identificaron áreas de mejora, la base desarrollada permite expandir el sistema hacia nuevos juegos con características similares, fortaleciendo su capacidad para manejar consultas complejas de manera eficiente y contextualizada.

Las mejoras propuestas optimizarán aún más la precisión, velocidad y experiencia del usuario, asegurando que estos proyectos sean herramientas mucho más útiles. Actualmente, las consultas sobre la base vectorial no son completamente correctas, y la confusión en la formulación de las respuestas tanto en el chatbot como en el agente es alta. Al realizar una extracción, limpieza y división de *chunks* más detallada, el sistema será más eficiente y se obtendrán mejores resultados.

El proyecto cumplió con los objetivos planteados tanto para la base de datos de grafos como la tabular, proporcionando una experiencia robusta para los usuarios interesados en aprender más sobre *Rajas of the Ganges*.

ANEXO

Enlaces a herramientas utilizadas

- LangChain https://docs.langchain.com/
- Sentence Transformers https://www.sbert.net/
- Beautiful Soup https://www.crummy.com/software/BeautifulSoup/
- Chromadb https://docs.trychroma.com/
- Llama Index https://www.llamaindex.ai/
- Selenium https://www.selenium.dev/
- Hugging Face https://huggingface.co/
- Neo4j https://console-preview.neo4j.io/tools/query

Datos para generar fuentes de datos

- https://boardgamegeek.com/boardgame/220877/rajas-of-the-ganges/forums/0
- https://boardgamegeek.com/boardgame/220877/rajas-of-the-ganges/files
- https://boardgamegeek.com/boardgame/220877/rajas-of-the-ganges/credits
- https://boardgamegeek.com/boardgame/220877/rajas-of-the-ganges