

TRABAJO PRÁCTICO FINAL

IA 4.2 Procesamiento del Lenguaje Natural

Tecnicatura en Inteligencia Artificial



Facundo López Crespo

Legajo N° L-3339/1

13/02/2025

Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Índice

Introducción	2
Ejercicio 1 - RAG	2
Resumen	2
Desarrollo	3
Base de datos vectorial	3
Base de datos de grafos	4
Base de datos tabular	5
Clasificador	5
Queries	5
Implementación RAG	6
Conclusiones	6
Anexo	7
Ejercicio 2 - Agente	11
Introducción	11
Desarrollo	11
Consultas	12
Referencias	14

Introducción

El presente trabajo a través de distintas herramientas propias del Procesamiento del Lenguaje Natural crea un chatbot experto en el juego de mesa Memoir '44 a partir del cual un usuario puede obtener información relevante para interiorizarse sobre el juego. A su vez, en una instancia siguiente, se incorpora el concepto de Agente basado en ReAct para atender las consultas de un usuario relativas a este juego.

Memoir '44 es un wargame en el que cada partida representa un escenario de La Segunda Guerra Mundial. Cada jugador controla a uno de los bandos: Aliados o el Eje. Mediante una gestión de mano de cartas, los jugadores irán ejecutando órdenes sobre las distintas unidades para intentar conseguir cumplir antes que el rival la condición de victoria del escenario, normalmente, alcanzar un determinado número de medallas, siendo la mayoría conseguidas mediante la eliminación de unidades rivales. El combate entre unidades se resuelve mediante una tirada de dados, siendo el número de dados dependiente del tipo de unidad y la distancia de ataque, más las modificaciones que aplique el terreno.

Ejercicio 1 - RAG

Resumen

El chatbot puede responder consultas de distinta índole en torno al juego. Por ejemplo, es capaz de informar quién lo diseñó, en cuáles otros juegos trabajó este diseñador, en qué año fue lanzado, qué puntaje tiene en la plataforma board game geek, cuáles son sus componentes, cómo se juega, cuáles son las opiniones más favorables y menos favorables de las personas que lo han jugado, entre tanta otra información. Para cumplir esta tarea se hace uso de la herramienta RAG (Retrieval Augmented Generation).

La información del dominio específico del juego se almacena en distintos tipos de base de datos. Se utiliza una base de datos vectorial para guardar información relativa a su reglamento, las tácticas que se pueden desplegar en una partida, una guía rápida para jugadores sin experiencia y diversas reseñas obtenidas sobre el mismo. Además, se utiliza una base de datos tabular para contar con estadísticas sobre el juego. Y por último, se utiliza una base de datos de grafos para acceder a información ligada principalmente a las personas que participaron en su creación.

A partir de la pregunta del usuario se recurre a un clasificador basado en Inteligencia Artificial (IA) para identificar la base de datos que contiene la información relevante para poder atender esta pregunta. Luego se obtiene la query que se corresponde con la información que se está buscando. Así se puede hacer el llamado a la base de datos que corresponda para extraer la información útil.

La información recuperada será incluida en el prompt elaborado para que el Modelo de Gran Escala (LLM) genera la respuesta adecuada.

Desarrollo

Para la implementación del chatbot se recurre a la técnica RAG (Retrieval Augmented Generation). Esta técnica permite suministrar información concreta sobre el juego, conocimiento no paramétrico, al Modelo de Gran Escala (LLM) elegido para atender las consultas del usuario¹.

En relación al LLM, se opta por recurrir a un modelo de la familia Qwen. Los modelos Qwen son desarrollados por la empresa Alibaba. En particular, el elegido para esta aplicación es el Qwen2.5-72B-Instruct, el cual presenta las siguientes especificaciones técnicas:

- Tipo: Modelos de lenguaje causal
- Etapa de entrenamiento: Preentrenamiento y post-entrenamiento
- Arquitectura: Transformers con RoPE, SwiGLU, RMSNorm y sesgo de atención QKV
- Número de parámetros: 72.7 mil millones
- Número de parámetros (sin incluir embeddings): 70.0 mil millones
- Número de capas: 80
- Número de cabezas de atención (GQA): 64 para Q y 8 para KV

Este LLM se destaca por ser competitivo en comparación con modelos de última generación, como el Llama-3-405B-Instruct, a pesar de ser cinco veces más pequeño. A su vez, tiene grandes habilidades para ejecutar tareas vinculadas con la matemática, la codificación y la conversación y comprensión de instrucciones. En particular, esta última habilidad es la razón por la que se escoge este modelo para la construcción del chatbot.

La información específica sobre el juego se obtiene principalmente de la plataforma board game geek y del blog misut meeple. Para extraer esta información se realizan diversas tareas como la lectura de documentos PDF, el web scraping y la transcripciones de videos².

Según la naturaleza de la información que se extrae se elige qué tipo de base de datos es conveniente utilizar para alojarla. Producto de esta diferenciación se recurre a una base de datos vectorial, una base de datos de grafos y una base de datos tabular.

Base de datos vectorial

Los textos que se utilizan son:

- reglamento del juego en un documento pdf
- guía de referencias para novatos disponible en un documento pdf
- explicación de los diferentes escenarios de batallas del juego en un documento pdf
- hilo de discusión en torno a las tácticas que se pueden implementar para ganar una partida disponible en un foro de la plataforma board game geek
- reseña disponible en un foro de la plataforma anterior
- reseña disponible en el blog misut meeple
- reseña disponible en You Tube

¹ El uso del LLM es de forma remota vía API. Se utiliza el servidor de Hugging Face.

² El scraping se realiza mayormente con Selenium, aunque cuando la página web no utiliza carga dinámica se opta por usar BeautifulSoup por su mayor sencillez.

No se aplican técnicas avanzadas de limpieza y normalización de texto que faciliten la tarea de comprensión semántica asignada a los algoritmos. En particular, las utilizadas se encargan de eliminar tanto caracteres no codificados, como espacios múltiples o espacios que se presentan al inicio o al final del texto.³

Para que la información recuperada sea valiosa y atienda con precisión el tema sobre el cual el usuario consulta, previo a poblar esta base de datos, se procede a segmentar el texto. Se elige como criterio de segmentación definir un tamaño fijo de trozo de texto establecido en 400 caracteres con un solapamiento de 30 caracteres entre segmentos haciendo uso de la herramienta RecursiveCharacterTextSplitter de langchain.⁴

La base de datos vectorial que se utiliza es ChromaDB, aunque esta herramienta cuenta con su propio modelo de vectorización, se decide acudir a un modelo particular de embedding para lograr la representación numérica del texto. El modelo utilizado es google/universal-sentence-encoder-multilingual/3, el cual es la tercera versión de la serie multilingüe del Universal Sentence Encoder desarrollada por la empresa Google. Tiene como principales características que está basado en la arquitectura transformers y que sus embeddings son de alta calidad para tareas como la búsqueda semántica, la clasificación de texto y el clustering⁵.

Base de datos de grafos

En esta base de datos se coleccionan los hechos asociados al juego⁶. Por ejemplo, se considera como un hecho que Memoir '44 fue diseñado por Richard Borg. Se recurre a la obtención de tríadas para constituir la ontología correspondiente. Dicha ontología se representa mediante el modelo RDF (Resource Description Framework). El grafo que se obtiene por la aplicación de este modelo se serializa para garantizar la persistencia de la información. El lenguaje de consulta que se utiliza para recuperar dicha información es el SPARQL.

Se utiliza un LLM para la extracción de tríadas. Como información de contexto se suministra texto obtenido del web scrapping. A través de esta técnica se obtiene texto vinculado con los créditos del juego y su vez con información particular sobre las entidades (diseñadores, artistas, editores, etc) asociadas al juego⁷.

³ Aunque esta decisión no afecta las búsquedas semánticas que se realicen, sí tiene un efecto sobre las búsquedas por palabras claves. En este sentido, se puede avanzar en la conversión a minúscula, la eliminación de la puntuación, los acentos o palabras de parado, la derivación o lematización del texto para mejorar los resultados obtenidos con este tipo de búsqueda.

⁴ Se podría haber optado por utilizar una segmentación más sofisticada que permita trozar a nivel frase

⁵ La elección de un modelo multilingüe se basa en que ciertas fuentes de información están en español. Una alternativa a considerar es avanzar en la traducción al inglés para poder utilizar un modelo de embedding donde la lengua excluyente sea el inglés. Así se obtiene una mejor performance. El mismo razonamiento se aplica para elegir el LLM.

⁶ Por hecho se entiende a la unidad básica de información que puede ser almacenada en un grafo de conocimiento.

⁷ Se torna dificultoso poder definir el tipo de entidad del objeto de la tríada cuando el sujeto no es Memoir 44. Esto trae aparejado problemas para consultar la base de datos en esos casos.

Particularmente interesa volcar en esta base de datos información sobre las personas que participan en los créditos del juego. Y así determinar qué otros juegos se pueden asociar a Memoir '44.

Base de datos tabular

Se hace uso de un dataframe de Pandas para guardar información inherente a las estadísticas del juego. Por ejemplo, en esta base de datos puede encontrarse información sobre el puntaje medio asignado al juego, la cantidad de comentarios recibidos, la cantidad de personas que lo jugaron o que compraron el juego, siempre teniendo como referencia que estas estadísticas son las informadas por la plataforma board game geek.

Clasificador

La aplicación necesita conocer a cuál base de datos debe acudir para recuperar la información. Con este propósito se implementa un modelo clasificador basado en LLM para a partir de la consulta del usuario inferir qué base de datos debe consultarse.

Se prueba un modelo de regresión logística provisto por la librería scikit-learn para evaluar su habilidad para clasificar consultas. Para ello, se construye un dataset con ejemplos de clasificación con el fin de entrenar al modelo de manera supervisada.

Con la finalidad de evaluar a ambos modelos se testea sus respectivos desempeños, encontrándose que el modelo clasificador basado en LLM obtiene una mejor métrica, por lo cual, se opta por este modelo para clasificar la consulta del usuario.

Queries

En caso que la recuperación debe llevarse adelante sobre la base de datos vectorial se seleccionan los textos que realizan el mayor aporte para que el LLM pueda generar una respuesta satisfactoria. Esta selección pasa por dos instancias.

En una primera instancia, se realiza una búsqueda híbrida para la cual se combinan los resultados tanto de la búsqueda semántica como los de la búsqueda por palabras claves. La búsqueda semántica utiliza como criterio para rankear la similitud de coseno dándole mayor puntaje a aquellos textos que se acercan al significado de la consulta del usuario. En cambio, la búsqueda por palabras claves rankea los textos en base al índice BM25 donde lo que se pondera es la semejanza en los términos utilizados entre los textos y la consulta del usuario.

En una segunda instancia, los resultados obtenidos por la búsqueda híbrida se rerankean utilizando un modelo Cross-Encoder, el cual permite examinar de manera más sofisticada la similitud semántica referida.

Por lo general, a partir de la búsqueda híbrida se obtienen 10 textos con sus respectivos puntajes. Por el reranqueo, se vuelve a asignar un nuevo puntaje a cada texto para finalmente elegir los mejores 5 textos que van a ser utilizados para aumentar el prompt.

En caso que la recuperación debe llevarse adelante sobre la base de datos de grafos se solicita al LLM que genere aquella consulta en SPARQL que permita recuperar la información relevante de la base de datos de conocimiento para atender la consulta del usuario. En el llamado al LLM se suministra como contexto la antología propia del juego.

El mecanismo anterior también aplica para la búsqueda en la base de datos tabular con la distinción que, a la hora de definir la ingeniería de prompt para este caso, se le indica al LLM que debe devolver un script propio para consultar un dataframe de Pandas y se le comunica cuáles son las columnas del mismo como así también las variables estadísticas que éste contiene.

Implementación RAG

Para su implementación se crea la clase *llm_memoir_44* con cuatro atributos. El primero `-client_hugging_face-` se utiliza para crear el cliente de Hugging Face, el segundo `-info-` para guardar la información de contexto, el tercero `-chat_prompt-` para guardar propiamente el prompt aumentado gracias a la información de contexto y el cuarto `-historial-` guarda el hilo de la conversación que desarrolla el chatbot con el usuario.

A su vez, la clase dispone del método `hablar` que se encarga de poner en funcionamiento el chatbot ejecutando el resto de los métodos de la clase. Ante cada pregunta del usuario, la secuencia es la siguiente, se ejecuta `set_context` el cual setear el valor correspondiente para el atributo `info`, luego se ejecuta `make_prompt` setear el valor correspondiente para el atributo `chat_prompt` y por último se llama al LLM pasándole como mensaje el prompt aumentado.

En la implementación se considera el escenario en el cual no es posible recuperar información, ante este caso, se le indica al LLM que le indique al usuario que no es posible responder la pregunta.

A su vez, se implementó la funcionalidad para que el chatbot tenga memoria, lo que permite que el prompt aumentado también cuente con la serie de preguntas y respuestas que se dan en la conversación. Esto permite que el LLM observe cuáles fueron las anteriores preguntas y respuestas dadas con el propósito de mejorar la respuesta en curso.

El idioma utilizado por el chatbot es el español, por lo cual, es caso que se detecte que la respuesta del LLM es en otro idioma se procede a su traducción para no alterar la lengua utilizada en la conversación.

Conclusiones

El RAG implementado resulta útil para evacuar una considerable gama de preguntas que un usuario puede realizar en torno al juego. Se puede constatar una precisión aceptable en las respuestas otorgadas. Para cumplir con este objetivo es clave suministrar textos de calidad que cubran la mayor parte posible de los tópicos en los cuales puede estar interesado el usuario.

Implementar una buena ingeniería de prompt es imprescindible a la hora de llamar al LLM, se puede verificar que mostrándole ejemplos precisos sobre cómo debe responder ante

ciertas preguntas puntuales (few shots) mejora significativamente las respuestas del modelo.

Anexo

Se copia una conversación mantenida con el chatbot (en negrita se remarcán las preguntas del usuario)

`llm.hablar()`

Bienvenido a Memoir 44 Chatbot! para salir escribe "chau"

Escribe tu consulta, para salir despídete con chau:¿Qué opina Trump de ese juego?

User: ¿Qué opina Trump de ese juego?

Agente: No tengo información específica sobre las opiniones de Donald Trump sobre Memoir '44'.

Dado que las opiniones personales de figuras públicas sobre juegos de mesa no suelen ser ampliamente documentadas o divulgadas, no puedo proporcionarte una respuesta precisa sobre lo que Donald Trump piensa de este juego en particular.

Si estás interesado en opiniones generales sobre Memoir '44, puedo ofrecerte las vistas de jugadores y críticos, pero no tengo datos sobre las opiniones de Donald Trump.

Escribe tu consulta, para salir despídete con chau:¿Quién diseñó el juego?

User: ¿Quién diseñó el juego?

Agente: El juego Memoir '44 fue diseñado por Richard Borg.

Escribe tu consulta, para salir despídete con chau:¿qué opinión positiva tiene el juego ?

User: ¿qué opinión positiva tiene el juego ?

Agente: Una opinión positiva sobre el juego Memoir '44 es que es un juego maravilloso si lo tomas por lo que es. No es un 'Juego de Guerra' en el sentido más estricto del término, sino un juego impulsado por cartas que ofrece una gran rejugabilidad y dinámica. La combinación de mecánicas de juego simples y la temática histórica de la Segunda Guerra Mundial hace que sea atractivo y accesible para una amplia gama de jugadores. Además, el juego es fácil de aprender y rápido de jugar, lo que lo convierte en una excelente opción para sesiones de juego tanto casuales como más serias. La interacción directa y la tensión que se genera durante los combates y las decisiones tácticas añaden un nivel de emoción y estrategia que mantiene a los jugadores enganchados.

Escribe tu consulta, para salir despídete con chau:y negativa

User: y negativa

Agente: Una opinión negativa sobre el juego Memoir '44 es que, a pesar de su atractivo y su jugabilidad, puede resultar demasiado simplificado para los entusiastas de los juegos de estrategia militar más detallados. Algunos jugadores pueden encontrar que las reglas y el sistema de combate no reflejan con precisión los aspectos tácticos y estratégicos de la Segunda Guerra Mundial, lo que puede limitar la profundidad y el realismo del juego. Además, la dependencia de las cartas de comando puede hacer que el juego sea menos predecible y más aleatorio, lo que puede frustrar a aquellos que buscan un mayor control y planificación en sus juegos de estrategia. La simplicidad del juego, aunque es un punto fuerte para algunos, puede ser visto como una limitación para otros que buscan una experiencia más compleja y detallada.

Escribe tu consulta, para salir despídete con chau:¿En cuáles otros juegos participó el diseñador de Memoir?

User: cuáles son las instrucciones del juego

Agente: Claro, aquí tienes un resumen detallado de las instrucciones básicas de Memoir '44:

Preparación del Juego

1. ****Elegir un Escenario:****

- Elige un escenario de la libreta de escenarios. Cada escenario tiene una configuración específica del tablero, unidades iniciales, y objetivos.

2. ****Configurar el Tablero:****

- Coloca el tablero de juego según las instrucciones del escenario. Utiliza las fichas de terreno para crear el campo de batalla.

3. ****Colocar las Unidades:****

- Cada jugador coloca sus unidades en las posiciones iniciales indicadas en el escenario. Las unidades incluyen infantería, tanques, y artillería.

4. ****Repartir las Cartas de Comando:****

- Cada jugador recibe una mano de cartas de comando, generalmente cinco cartas al inicio del juego.

Turno del Jugador

1. ****Jugar una Carta de Comando:****

- El jugador que inicia el turno (definido en el escenario) juega una carta de comando de su mano. La carta indica qué unidades pueden mover y/o atacar.

2. ****Mover las Unidades:****

- Mueve las unidades según las indicaciones de la carta de comando. Las unidades pueden moverse hasta un número de espacios determinado por el tipo de unidad y el terreno.

- ****Infantería:**** Puede moverse 2 espacios.

- ****Tanques:**** Pueden moverse 3 espacios.

- ****Artillería:**** Puede moverse 1 espacio.

- Las unidades pueden girar al final de su movimiento, pero cada giro cuenta como un espacio adicional movido.

3.

Escribe tu consulta, para salir despídete con chau:para qué sirven las Medallas de Victoria

User: para qué sirven las Medallas de Victoria

Agente: Las Medallas de Victoria en Memoir '44 sirven para rastrear el progreso y el éxito de cada jugador durante el juego. El objetivo principal del juego es ser el primero en ganar un número específico de Medallas de Victoria, que generalmente varía entre 4 y 6, dependiendo del escenario seleccionado. Las Medallas de Victoria se pueden ganar de las siguientes maneras:

1. ****Eliminación de Unidades Enemigas:****

- Cada vez que una unidad enemiga es completamente eliminada del campo de batalla, el jugador que la eliminó gana una Medalla de Victoria. La última figura de la unidad eliminada se coloca en el Medallero del jugador ganador.

2. ****Objetivos Específicos:****

- Algunos escenarios tienen objetivos específicos que, si se cumplen, otorgan Medallas de Victoria. Por ejemplo:

- Capturar y mantener ciertos edificios o hexágonos del tablero.

- Hacer que una unidad salga del campo de batalla por un hexágono de salida designado.

- Controlar ciertos puntos estratégicos al final de un turno.

3. ****Control de Objetivos:****

- En algunos escenarios, controlar ciertos hexágonos del tablero al final de un turno puede otorgar Medallas de Victoria.

Ejemplos de Cómo Se Ganan las Medallas de Victoria:

- ****Eliminación de Unidades:****

- Si un jugador logra eliminar una unidad de infantería enemiga, gana una Medalla de Victoria.

- ****Captura de Objetivos:****

- Si un escenario requiere que un jugador capture y mantenga tres de los cuatro edificios alrededor del Aeródromo, y lo logra, gana una Med

Escribe tu consulta, para salir despídete con chau: cómo se usan los dados

User: cómo se usan los dados

Agente: En Memoir '44, los dados de combate se usan para resolver los ataques entre las unidades.

Aquí tienes un resumen detallado de cómo se usan los dados:

Proceso de Ataque con Dados

1. ****Elegir la Unidad Atacante:****

- Después de mover las unidades, elige una unidad que desees atacar con. La unidad atacante debe tener línea de visión directa a la unidad enemiga que desees atacar.

2. ****Determinar el Número de Dados:****

- El número de dados que se lanzan depende del tipo de unidad atacante y del tipo de unidad atacada, así como de la distancia y el terreno:

- ****Infantería:**** Lanza 2 dados.

- ****Tanques:**** Lanza 3 dados a una distancia de 1-3 hexágonos.

- ****Artillería:**** Lanza 1 dado a una distancia de 1-5 hexágonos.

3. ****Modificar el Número de Dados:****

- El número de dados puede ser modificado por el terreno y otras condiciones:

- ****Bosques, Pueblos, y Búnkeres:**** Restan 1 dado.

- ****Montañas:**** Restan 2 dados.

- ****Camino y Campos Abiertos:**** No modifican el número de dados.

4. ****Lanzar los Dados:****

- Lanza los dados de combate. Cada dado tiene símbolos que representan diferentes resultados:

- ****Cruz (X):**** Elimina una figura de la unidad atacada.

- ****Bandera (F):**** La unidad atacada debe retirarse un hexágono en la dirección opuesta al atacante.

-

Escribe tu consulta, para salir despídete con chau: cuál es el puntaje del juego

User: cuál es el puntaje del juego

Agente: El puntaje del juego Memoir '44 es 7.532. Este puntaje generalmente se refiere a la calificación promedio que el juego ha recibido en sitios de reseñas y comunidades de juegos de mesa, como BoardGameGeek.

Escribe tu consulta, para salir despídete con chau: cuántas personas lo jugaron

User: cuántas personas lo jugaron

Agente: La información proporcionada no especifica cuántas personas han jugado Memoir '44. El número 172,716 que se menciona podría referirse a diversas métricas, como el número de votos o calificaciones en un sitio web de juegos de mesa, pero no se puede confirmar que sea el número de personas que han jugado el juego.

Si necesitas información más precisa sobre cuántas personas han jugado Memoir '44, te recomiendo consultar sitios de juegos de mesa como BoardGameGeek, donde puedes encontrar estadísticas detalladas sobre la popularidad y el número de jugadores.

Escribe tu consulta, para salir despídete con chau:chau

Gracias por usar Memoir 44 Chatbot, hasta la próxima!

Ejercicio 2 - Agente

Introducción

ReAct es un enfoque para integrar razonamiento lógico y uso de herramientas en agentes basados en modelos de lenguaje (LLMs). Su nombre proviene de la combinación de "Reasoning" (razonamiento) y "Acting" (acción).

Está diseñado para permitir que un agente resuelva problemas de manera más efectiva al alternar entre el razonamiento explícito y la ejecución de acciones, como el uso de herramientas externas.

El agente puede dividir tareas complicadas en subtareas manejables, en lugar de intentar resolver todo de una vez.

Se trata de un proceso iterativo que culmina cuando encuentra una respuesta adecuada o por límite de tiempo, si no la encuentra.

El proceso comienza cuando ingresa una pregunta del usuario y desencadena la siguiente secuencia de tareas:

- Query: pregunta del usuario
- Thought: el agente evalúa la pregunta del usuario y decide qué pasos debe seguir, utilizando su conocimiento general y las herramientas disponibles. Esto ayuda a que el sistema sea transparente y a que podamos entender por qué decide tomar una acción específica.
- Action: El agente toma medidas concretas basadas en su razonamiento y decide usar una herramienta para buscar la información.
- Action Input: El agente especifica el parámetro de entrada que pasará a la herramienta.
- Observation: La herramienta devuelve un resultado. Este resultado se envía de vuelta al agente para que lo procese. Esta es una respuesta parcial y debe ser validada o utilizada para construir la respuesta final.

Las trayectorias constan de múltiples pasos de pensamiento-acción-observación (Thought-Act-Observation).

Los pensamientos de forma libre se utilizan para diferentes tareas, como descomponer preguntas, extraer información, realizar razonamientos de sentido común/aritmético, guiar la formulación de búsqueda y sintetizar la respuesta final.

Desarrollo

Se hace uso de la librería llama_index para construir al agente. Para este proceso, principalmente, se utiliza un LLM, se suministra un conjunto de herramientas y se especifica a través de un prompt cómo debe actuar el agente.

En relación al LLM, a través de la plataforma Ollama se ejecuta localmente el modelo Llama 3.2. Se utiliza este modelo porque se caracteriza por su buen rendimiento y su poco peso⁸. Además se utiliza una temperatura de 0.1 para evitar alucinaciones y así obtener respuestas precisas basadas en la información de contexto. A su vez, se setea una ventana de contexto de 4096 para evitar respuestas largas.

En relación a las herramientas, el agente puede ejecutar estas funciones para obtener información relacionada a las preguntas del usuario:

- `Get_info_graph_db`: en base a la consulta aplica una query a la base de datos de grafos.
- `Get_info_tabular_db`: en base a la consulta aplica una query a la base de datos tabular.
- `Get_info_vector_db`: en base a la consulta aplica una query a la base de datos vectorial.

Por último, en relación al prompt, se comunican las temáticas abordadas en cada una de las bases de datos, se dan instrucciones sobre cómo debe actuar ante cada pregunta, se aclaran algunas reglas adicionales y se dan algunos ejemplos de cómo se espera que el agente responda ante ciertas preguntas del usuario.

Consultas

Los ejemplos siguientes sirven para evidenciar el desempeño del agente:

1. ¿Quién diseñó el juego?

Primero busca en la base de datos vectorial, aunque razona que la información recolectada no está vinculada con la temática de la consulta. Entonces consulta la base de datos de grafos, como sí encuentra información útil responde de manera correcta quién es el diseñador del juego.

2. ¿qué opinión positiva tiene el juego ?

Busca primero en la base de datos vectorial, como la información recolectada sí está ligada a la temática consultada responde sin acudir a otra herramienta. La respuesta es precisa.

3. ¿cuántas personas lo jugaron?

Primero busca en la base de datos vectorial, aunque razona que la información recolectada no está vinculada con la temática de la consulta. Entonces consulta la base de datos de grafos, aunque obtiene un error en el proceso. Por tal motivo, consulta la base de datos tabular y obtiene la información necesaria para responder adecuadamente la pregunta.

4. ¿cuántas personas lo jugaron?

⁸ Llama 3.2 tiene dos versiones ligeras de 1 y 3 billones de parámetros.

Distingue bien cuál herramienta debe usar para responder la pregunta. Directamente consulta en la base de datos tabular, encuentra el dato que precisaba y responde acertivamente sin acudir a otras herramientas.

5. ¿cuántos comentarios tiene el juego?

Distingue bien la herramienta que tiene que utilizar, aunque al buscar en la base de datos tabular la información que recupera es errónea e informa una estadística inapropiada.

6. ¿cuáles son las instrucciones del juego?

Primero consulta en la base de datos de grafos, luego en la base de datos vectorial, en ambos casos no puede recuperar información por errores de sintaxis. Al consultar la base de datos vectorial, recupera información, pero la respuesta no está estrictamente ligada a la temática de la consulta.

Conclusión

El agente en pocos casos puede distinguir cuál es la herramienta apropiada a utilizar para responder la consulta del usuario. Por tal motivo, en la mayoría de los casos tiene que recurrir a más de una herramienta para responder las consultas. Sí tiene un buen razonamiento para determinar que la información obtenida no es útil para responder las consultas.

Se puede constatar que en algunos casos, aún utilizando la herramienta adecuada, no responde correctamente.

Referencias

- Board Game Geek. Memoir 44

<https://boardgamegeek.com/boardgame/10630/memoir-44>

- Hugging Face: Qwen2-7B-Instruct

<https://huggingface.co/Qwen/Qwen2-7B-Instruct>

- Ollama : llama 3.2

<https://ollama.com/library/llama3.2>

- Chroma

<https://www.trychroma.com/>

- Modelo de embbeding universal-sentence-encoder-multilingual/3

https://www.tensorflow.org/hub/tutorials/retrieval_with_tf_hub_universal_encoder_qa