Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Tecnicatura Universitaria en Inteligencia Artificial
Procesamiento de Imágenes I - IA 4.4

TRABAJO PRÁCTICO N°2
Procesamiento del Lenguaje Natural
Chatbot experto en el juego Rajas del Ganges

Integrantes:

Cima, Nancy Lucía
C-7379/2

Docentes:

Manson, Juan Pablo
Geary, Alan
Sollberger, Dolores.
Ferrucci, Costantino

Fecha de entrega: 18/12/2024

Índice

Resumen

El presente proyecto desarrolla un chatbot experto en el juego de mesa "Rajas of the Ganges", utilizando la técnica de Retrieval Augmented Generation (RAG). El objetivo principal fue diseñar un sistema capaz de responder preguntas sobre el juego en español o inglés, aprovechando diversas fuentes de datos como documentos textuales, bases tabulares y bases de datos de grafos, con el fin de ofrecer respuestas precisas y contextualizadas.

Los métodos empleados incluyen web scraping para la obtención de información de la página web de BGG, la segmentación y limpieza de textos mediante herramientas de procesamiento de lenguaje natural como Langchain, generación de embeddings con modelos preentrenados y almacenamiento en una base de datos vectorial ChromaDB. Además, se implementaron dos clasificadores: uno basado en modelos de lenguaje (LLM) y otro entrenado con ejemplos y embeddings, permitiendo comparar su rendimiento y seleccionar el más efectivo. La integración con bases de datos de grafos y tabulares se logró mediante consultas dinámicas (SPARQL y SQL) que optimizan el uso de contexto relevante.

Los resultados demuestran que el chatbot puede interpretar y responder preguntas complejas, combinando información de múltiples fuentes. Si bien ambos clasificadores ofrecieron buen desempeño, el basado en LLM mostró mayor flexibilidad ante preguntas abiertas.

Introducción

Rajas of the Ganges es un juego de mesa ambientado en la antigua India, durante el período de expansión del Imperio Mogul. En este juego de colocación de trabajadores, los jugadores, representando a Rajas y Ranis, deben equilibrar prestigio y riquezas para cumplir con su rol de soberanos y alcanzar la victoria. A través de la mejora de sus provincias y la gestión del Karma, los jugadores buscan convertirse en los líderes más destacados de la nación.

En el contexto de este proyecto, el objetivo principal fue crear un chatbot capaz de responder preguntas sobre el juego en español o inglés, utilizando un enfoque de *Retrieval Augmented Generation* (RAG). Esta tecnología integra fuentes de datos heterogéneas, incluyendo documentos textuales, datos tabulares y bases de datos de grafos, para generar respuestas precisas y adaptadas al usuario.

Para alcanzar este objetivo, se definieron las siguientes acciones:

1. Selección de herramientas adecuadas.
2. Construcción de una base de datos de grafos.
3. Creación de una base de datos vectorial para búsqueda semántica.
4. Diseño de una base de datos tabular.
5. Implementación de consultas dinámicas y un sistema de *reranking* para optimizar la recuperación de datos.
6. Desarrollo del chatbot con capacidad multilingüe.
7. Construcción de un asistente inteligente para gestionar interacciones complejas.

Este informe documenta cada paso del desarrollo, desde la recolección y preparación de datos hasta la implementación de las tecnologías necesarias. Se detalla cómo se abordaron los retos técnicos, los resultados obtenidos y las conclusiones derivadas de la experiencia. La estructura del informe incluye una introducción al contexto del juego y el sistema, el desarrollo técnico dividido en etapas clave, y una discusión sobre los logros, limitaciones y posibles mejoras del proyecto.

Metodología

Fuentes de datos y contexto

Para desarrollar el sistema, fue necesario crear y estructurar diversas fuentes de datos, aprovechando la riqueza de información disponible sobre *Rajas of the Ganges*. Estas fuentes incluyen una base de datos de grafos, una base tabular y una base vectorial, cada una diseñada para modelar distintos aspectos del juego y su contexto.

Base de datos de Grafos

La base de datos de grafos se utilizó para representar relaciones complejas no directamente relacionadas con la jugabilidad, como conexiones entre diseñadores, artistas, empresas editoras, mecanismos de juego, categorías y nombres alternativos del juego. Para obtener esta información, se empleó Web Scraping con la biblioteca Selenium, extrayendo datos del sitio *BoardGameGeek*.

Se implementó un proceso automatizado con Selenium en modo *headless* para explorar las páginas web y extraer información relevante. Las plantillas dinámicas de XPath se utilizaron para localizar elementos como nombres, enlaces y descripciones. Para enriquecer la información recolectada, se empleó la biblioteca langdetect, identificando el idioma de los nombres alternativos del juego.

Los datos extraídos se estructuraron como nodos, aristas y propiedades. Las relaciones entre nodos, como "Diseñado por" (*Designed_by*) o "Publicado por" (*Published_by*), fueron definidas dinámicamente mediante funciones específicas.

Cada nodo representa una entidad del juego, como diseñadores, artistas, mecánicas, editoriales y nombres alternativos, mientras que las aristas reflejan las relaciones entre estas entidades. Se utilizó una función para asignar relaciones específicas según el tipo de nodo.

La implementación de la base de datos de grafos se llevó a cabo utilizando Redis, elegido por su integración sencilla con Google Colab y su eficiencia al manejar datos relacionales. Redis permitió representar la información en forma de nodos y relaciones, lo cual es ideal para modelar estructuras complejas.

Para construir el grafo, se diseñaron funciones que permitieron crear nodos y relaciones dinámicamente, partiendo de un nodo principal que representa el título del juego y conectándolo con las entidades recolectadas.

Adicionalmente, se utilizó la biblioteca networkx para generar visualizaciones del grafo, facilitando la comprensión de las relaciones creadas. Estas gráficas mostraron cómo se interconectan las entidades, proporcionando un esquema claro del modelo.

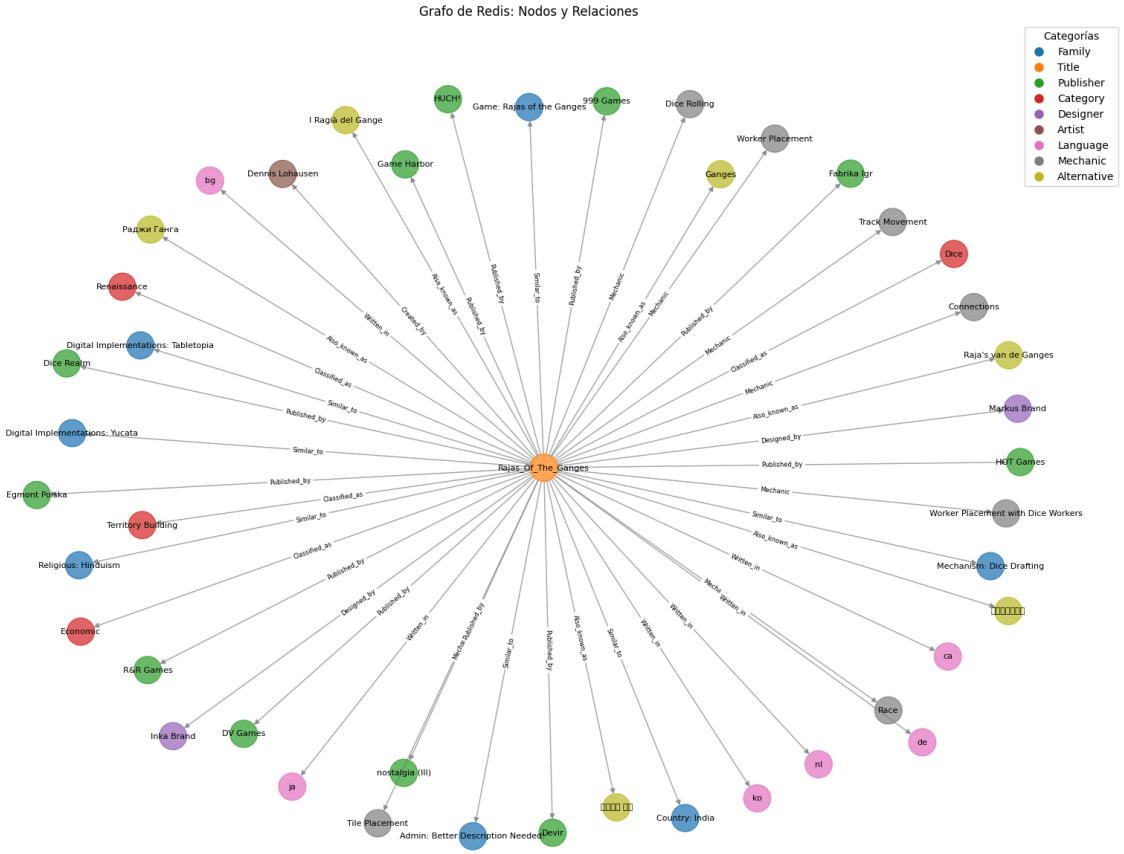

Figura 1: Diagrama de la base de datos de grafo

Se implementaron funciones para recuperar y validar los datos almacenados, asegurando la consistencia de la información. Esto incluyó la extracción de nodos y relaciones en un formato estructurado, lo que permitió integrar la base de datos al sistema de chatbot desarrollado.

La base de datos de grafos modeló relaciones como:

- Diseñadores que diseñaron el juego (*Designed_by*).
- Editoriales que publicaron el juego (*Published_by*).
- Artistas que crearon el juego (*Created_by*)

- Mecánicas y categorías (*Mechanic*, *Classified_as*).
- Familia (*Similar_to*)
- Conexiones con nombres alternativos en diferentes idiomas (*Also_known_as*, *Written_in*).

Esta estructura permitió una representación eficiente de la información, optimizando el acceso y la recuperación de datos para el chatbot desarrollado en etapas posteriores.

Base de datos Tabular

La base tabular recopiló datos estructurados como puntuaciones, clasificaciones y estadísticas, dado que este tipo de base de datos organiza la información en tablas, donde cada fila representa un registro y las columnas describen atributos relacionados. Su estructura es ideal para manejar datos numéricos y categóricos de forma ordenada.

Entre las estadísticas recopiladas se incluyen:

- Calificación promedio.
- Número de calificaciones.
- Desviación estándar.
- Cantidad de comentarios.
- Número de fans.
- Visitas a la página.
- Rankings del juego.

La información fue obtenida directamente desde la página de estadísticas del juego en *BoardGameGeek* mediante Web Scraping.

Para recolectar las estadísticas, se utilizó Selenium. La función desarrollada accedió al sitio web y extrajo los datos relevantes utilizando selectores CSS para identificar los elementos de interés.

Una vez obtenidos los datos, se procedió a realizar una limpieza y transformación para garantizar su utilidad. Por un lado, se renombraron las columnas, los nombres originales de las métricas fueron reemplazados por términos más descriptivos, como "Fans" por "Número de Fans".

Por otro lado, se transformaron los datos de formato ancho (columnas para cada métrica) a formato largo, donde cada fila corresponde a una métrica específica con su valor. Este formato simplifica el análisis y la visualización de los datos.

Base de datos Vectorial

Una base de datos vectorial es una herramienta diseñada para almacenar y gestionar datos en forma de vectores, los cuales representan información en un espacio multidimensional. Este tipo de base de datos es ideal para aplicaciones que requieren búsqueda y recuperación de información basada en similitudes, como texto, imágenes o videos. En este proyecto, la base de datos vectorial se utilizará para almacenar documentos de texto en formato PDF, videos tutoriales, reseñas, y opiniones extraídas de foros relacionados con *Rajas of the Ganges* y transcripciones de videos de YouTube.

En particular, como documentos de texto se incluyeron reglamentos y guías del juego, descargados en formato PDF desde una carpeta de Google Drive. Estos documentos contienen información detallada sobre las reglas y mecánicas del juego, fundamentales para la comprensión del chatbot.

El texto de los documentos fue extraído mediante la biblioteca PyPDF2, y se aplicaron técnicas de limpieza para eliminar caracteres innecesarios y unificar el formato. Se limpiaron caracteres como \uXXXX y se consolidaron textos eliminando saltos de línea y espacios múltiples.

Por otro lado, se extrajeron comentarios de la sección "Estrategia" del foro de *BoardGameGeek*, que contienen consejos y debates sobre tácticas y estrategias del juego.

Se utilizó Selenium para navegar por páginas dinámicas y extraer comentarios de usuarios clasificados como "Estrategia". La función desarrollada asegura que los comentarios sean visibles antes de extraerlos y aplica un proceso de limpieza para eliminar etiquetas HTML y otros elementos no deseados.

También se utilizaron transcripciones de dos videos relevantes:

- **Tutorial:** *Learn to Play Rajas of the Ganges (Plus Variants, Mini-expansions)*
  Ver en YouTube
- **Reseña:** *Rajas of the Ganges Review - JonGetsGames*
  Ver en YouTube

Las transcripciones se obtuvieron utilizando la API de subtítulos de YouTube, procesando cada video para convertir sus contenidos en texto limpio.

Los textos extraídos fueron procesados para eliminar caracteres no deseados, limpiar formatos y dividirlos en fragmentos (chunks) manejables. Estos fragmentos se transformaron en vectores mediante el modelo preentrenado *SentenceTransformers: paraphrase-multilingual-mpnet-base-v2*.

Se dividieron los documentos en fragmentos de 400 caracteres con un solapamiento de 30 caracteres para preservar el contexto con ayuda de RecursiveCharacterTextSplitte.

Cada fragmento fue convertido en un vector utilizando el modelo mencionado, que es compatible con múltiples idiomas.

Los vectores generados se almacenaron en ChromaDB, organizados por categorías como "reglas", "estrategias", "tutoriales" y "reseñas".

Como resultado, la base de datos vectorial incluye una amplia variedad de documentos relevantes, procesados y organizados para facilitar búsquedas eficientes:

- **Reglamentos y guías:** Proporcionan información clave sobre las reglas del juego.
- **Comentarios de foros:** Añaden perspectivas estratégicas de jugadores experimentados.
- **Transcripciones de videos:** Incluyen explicaciones detalladas y opiniones sobre el juego.

Esta integración permite que el sistema identifique contenido relevante basado en consultas semánticas, optimizando las respuestas del chatbot y enriqueciendo la experiencia del usuario.

Clasificadores

Se implementaron dos versiones del clasificador con el objetivo de determinar cuál ofrecía un mejor desempeño en la tarea de categorizar consultas de los usuarios.

La primera versión utilizó un modelo entrenado con ejemplos y embeddings. Para ello, se generaron varios ejemplos de consultas que fueron procesados y vectorizados utilizando embeddings semánticos. El modelo fue ajustado para identificar patrones en los datos y asociar las consultas con las categorías correspondientes. Este enfoque demostró ser eficiente al trabajar con un conjunto fijo de categorías y consultas relativamente predecibles, ya que la clasificación dependía directamente de los ejemplos proporcionados durante el entrenamiento.

Por otro lado, se desarrolló un clasificador basado en un modelo de lenguaje (LLM), que aprovechaba su capacidad para interpretar consultas complejas y formular clasificaciones a partir de un entendimiento contextual más amplio. Este enfoque resultó ser más flexible, especialmente frente a consultas menos estructuradas o que no estaban contempladas en el conjunto de ejemplos del modelo anterior.

Al comparar los resultados, el clasificador basado en LLM destacó por su mayor capacidad de generalización y su habilidad para manejar consultas abiertas o ambiguas. Sin embargo, el modelo entrenado con ejemplos mostró ventajas en cuanto a velocidad y eficiencia en contextos más controlados. En base a estas observaciones, se optó por utilizar el clasificador basado en LLM para este proyecto, priorizando su adaptabilidad y su capacidad para ofrecer respuestas más precisas en un entorno dinámico.

Resultados

Conclusiones

Anexos

Como material de apoyo para la realización del trabajo práctico se usaron los apuntes de clase.

Link a la pagina de BGG del juego:
https://boardgamegeek.com/boardgame/220877/rajas-of-the-ganges