Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

T.U.I.A

Trabajo Práctico Final

Procesamiento del Lenguaje Natural
Sistema RAG para el Juego "Lost Ruins of Arnak"

Integrante:

Masciangelo, Lucía

2024

Informe del Trabajo Práctico Final: chatbot experto en Las Ruinas Perdidas de Arnak con técnicas de RAG y Agentes.

Introducción

Este informe registra el desarrollo de un sistema de Recuperación y Generación de Información (RAG) enfocado en el juego de mesa "Las Ruinas Perdidas de Arnak". Este sistema combina métodos sofisticados de Procesamiento de Lenguaje Natural (NLP), Aprendizaje Automático y bases de datos para proporcionar respuestas a preguntas vinculadas al juego. Los datos procesados incluyen las reglas del juego, opiniones de los jugadores, información estructurada y relaciones modeladas en grafos. El sistema emplea clasificadores y modelos de generación para responder de manera contextual.

A continuación se va a explicar paso a paso cada una de las acciones que se fueron realizando para llevar a cabo todo el proyecto.

Ejercicio 1: Sistema RAG

Resumen

En esta parte se trata la creación de un sistema basado en RAG para responder consultas vinculadas a "Lost Ruins of Arnak". Los datos utilizados incluyen datos de texto, tabulares y grafos. El sistema combina la clasificación de consultas, la recuperación de información y la generación de respuestas empleando modelos de NLP ya entrenados y herramientas de bases de datos.

Desarrollo detallado de los pasos para la solución del problema

1. Extracción de Datos:
    - Se descargó y procesó un PDF del reglamento del juego utilizando *pdfplumber*, extrayendo las reglas en formato de texto.
    - Mediante *BeautifulSoup*, se realizó web scraping sobre reseñas del juego, obteniendo títulos y contenidos asociados.

2. **Construcción de Bases de Datos:**
    - **Texto:** Los textos extraídos del PDF y del web scraping se combinaron y dividieron en fragmentos manejables (500 chuncks) mediante *RecursiveCharacterTextSplitter.*
    - **Tabular:** Se creó manualmente un DataFrame con información estructurada sobre jugadores, duración del juego, puntuaciones, y precios, exportándolo a un archivo CSV.
    - **Grafos:** Usando *NetworkX*, se modelaron relaciones entre creadores, mecánicas, premios y juegos similares en un grafo dirigido.

3. **Clasificación de Consultas:**
    - Se implementó un clasificador basado en SVM entrenado con embeddings generados por el modelo *intfloat/e5-large*.

Reporte de clasificación para tu modelo entrenado:

	precision	recall	f1-score	support
graph	1.00	1.00	1.00	2
table	0.67	1.00	0.80	2
text	0.00	0.00	0.00	1
accuracy			0.80	5
macro avg	0.56	0.67	0.60	5
weighted avg	0.67	0.80	0.72	5

    - Adicionalmente, se utilizó el modelo *Flan-T5* para clasificar consultas mediante prompts.

Reporte de clasificación para HuggingFace:

	precision	recall	f1-score	support
graph	0.00	0.00	0.00	2
table	1.00	1.00	1.00	2
text	0.33	1.00	0.50	1
accuracy			0.60	5
macro avg	0.44	0.67	0.50	5
weighted avg	0.47	0.60	0.50	5

Se terminó decidiendo quedarnos con el modelo entrenado a través de SVM, ya que accuracy, (y los ejemplos evaluados), seleccionaron mejor con este modelo entrenado con ejemplos que el LLM ya entrenado.

4. **Flujo RAG:**
    - Una vez clasificada la consulta, el sistema recupera información relevante, de forma dinámica, desde las bases de datos:
        - **Base textual:** Búsqueda semántica en ChromaDB.
        - **Base tabular:** Filtrado de información en el DataFrame.
        - **Base de grafos:** Consultas sobre relaciones en NetworkX.
    - Una vez extraído el contexto de alguna de las bases de datos, la respuesta final se genera utilizando el modelo *Flan-T5-small*.

5. Evaluación:
    - Se evaluaron los clasificadores con consultas de prueba, midiendo su precisión mediante reportes de clasificación.

Conclusiones

El sistema desarrollado permite clasificar consultas en tres categorías (texto, tabla y grafo) y recuperar información de manera precisa y contextual. La integración de múltiples fuentes de datos y modelos NLP demuestra un enfoque robusto para resolver problemas complejos relacionados con la información de un juego de mesa.

Ejercicio 2: AGENTES

Resumen

Esta sección describe la implementación de un agente ReAct que combina herramientas para responder consultas relacionadas con *"Lost Ruins of Arnak"*, utilizando un enfoque basado en prompts y LLMs recuperando información desde las bases de datos.

Desarrollo detallado de los pasos para la solución del problema

1. Definición de Herramientas:
    - Se crearon funciones para buscar información en las bases de datos (texto, tabla y grafo).
    - Cada función está asociada a una herramienta registrada con *llama-index*.

2. Modelo Generativo:
    - Se configuró el modelo *Flan-T5-small* para procesar prompts y generar respuestas.

3. Agente ReAct:
    - Usando *llama-index*, se implementó un agente que combina herramientas para recuperar información y generar respuestas.
    - El agente clasifica consultas, selecciona la herramienta adecuada y proporciona respuestas basadas en el contexto recuperado.

4. Evaluación:
    - Se probaron consultas relacionadas con reglas del juego, información estructurada y relaciones en grafos, validando la capacidad del agente para manejar múltiples tipos de consultas.

Conclusiones

El sistema desarrollado representa una solución robusta y versátil para la interacción basada en consultas relacionadas con "Lost Ruins of Arnak". A través de la integración de múltiples fuentes de datos y el uso de algoritmos avanzados para clasificación y generación de respuestas, se logra ofrecer información precisa y contextualizada.

El agente ReAct demuestra ser más efectivo en la gestión de consultas diversas, integrando herramientas especializadas para recuperar información de distintas bases de datos. Este enfoque modular permite extender el sistema a otros dominios y casos de uso.

Enlaces a las herramientas utilizadas

- LlamaIndex: GitHub
- Modelo Flan-T5-small: Hugging Face
- Modelo intfloat/e5-large para embeddings: Hugging Face
- Modelo Flan-T5 para generación de texto: Hugging Face
- Librerías: pdfplumber, BeautifulSoup, transformers, NetworkX, ChromaDB