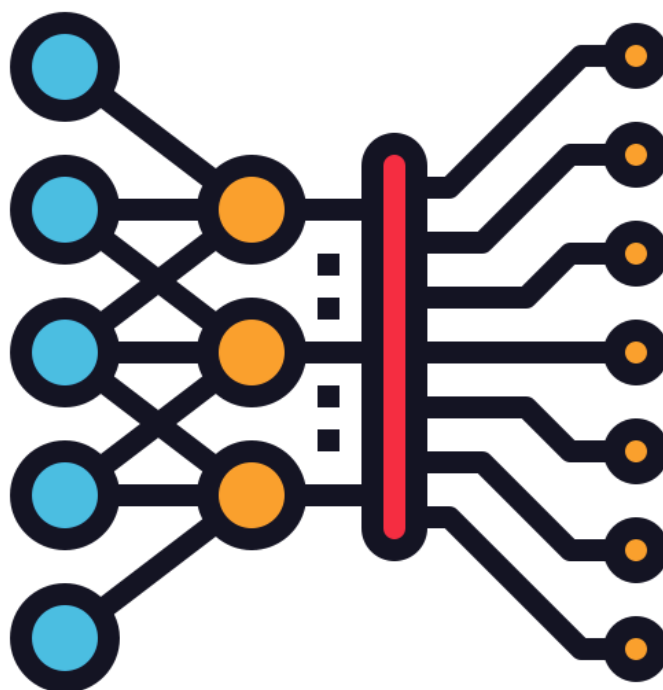




| **UNR** Universidad
Nacional de Rosario

Trabajo práctico Procesamiento del Lenguaje Natural

Tecnicatura Universitaria en Inteligencia Artificial



Docentes

- Juan Pablo Manson
- Andrea Carolina Leon Cavallo
- Alan Geary



Índice

Sistemas multiagentes	3
Sistema Multiagente	3
Funcionamiento de un Sistema Multiagente	3
Aplicaciones Prácticas en Diversos Sectores	3
Aplicación open-source	4
AutoGen	4
Principales Características	4
Problemática a solucionar con un sistema multiagente	5
Problemática	5
Esquema multiagente	6
Ejemplo de despliegue y conversación entre agentes y usuarios	6
fuentes	7

Informe del desarrollo RAG

Datos utilizados

El sistema RAG que implementó contiene información respecto a reservas ecológicas, animales en peligro de extinción y peritos o consultores expertos en materia ambiental de Argentina.

Los datos fueron extraídos de un libro de más de 500 páginas y dos archivos csv. Estos se almacenaron en dos bases de datos distintas, una vectorial (chromadb) y otra de grafos (neo4j)

Carga y preparación de datos

En el caso de la base de datos chromadb luego de extraer los datos de sus respectivos archivos, no se aplicaron modificaciones o tratamiento al texto debido a que no era necesario, solo se realizó un split y tokenización para luego realizar los embeddings.

Una vez hechos los embeddings se realizó la carga a la base de datos vectorial creando un vector store y configurandolo como retriever.

Mientras que en la base de datos neo4j utilizando la interfaz que provee su página web se cargaron los datos de forma directa del archivo csv estableciendo sus respectivas relaciones y nodos.

Crear funciones del sistema RAG

Utilice varias funciones para coordinar las respuestas del rag y hacerlo más eficiente. las funciones son:

- grafos()
Es una función que se encarga de hacer una consulta a la base de datos de grafos para obtener información acerca de las reservas naturales que se encuentran en una provincia.
- clasificador()
La función clasificador se encarga de clasificar la pregunta que recibe el RAG y esta devuelve una tupla indicando la fuente de datos que se debe consultar para poder responder la pregunta (aquí se utiliza la técnica few-shots).

- `obtener_contexto()`
Esta función recibe la respuesta del clasificador y obtiene el contexto de las fuentes para responder la consulta.
- `LLM()`
Con esta función se genera la instancia del LLM para generación de texto.
- `sistema()`
La función se encarga de llamar al resto de funciones para generar la respuesta a la consulta.
- `responder()`
Esta función solo se encarga de formatear la respuesta del modelo.

Sistemas multiagentes

Sistema Multiagente

Funcionamiento de un Sistema Multiagente

Se toma la idea de agente como una entidad que percibe e interactúa con un entorno, esto cambia por completo el enfoque que se tenía antes sobre la resolución de problemas.

Ya no hablamos de un programa que sigue instrucciones para encontrar una posible solución, sino que hablamos de una entidad que cumple un rol en una situación y buscará cumplir sus objetivos de la forma que sea, brindando soluciones adaptativas en la materia del rol que está cumpliendo el agente.

En un Sistema Multiagente (SMA) los agentes IA son autónomos, flexibles, capaces de tomar decisiones independientes y trabajan de manera colaborativa cumpliendo diversos roles para alcanzar sus objetivos. Estos sistemas son esenciales para abordar problemas complejos, intratables para un agente individual o sistemas monolíticos. La interacción entre múltiples agentes permite aprovechar sus habilidades y conocimientos individuales, mejorando la eficiencia de las soluciones y volviéndolas adaptativas.

Su visión local del sistema les permite adaptarse a entornos complejos. La descentralización evita un control monolítico, fomentando la interacción entre agentes para lograr objetivos comunes. Estos agentes pueden tener capacidad inferencial, trabajar con metas abstractas, ser reactivos para responder a condiciones ambientales y proactivos al tomar iniciativas para lograr sus objetivos.

Aplicaciones Prácticas en Diversos Sectores

Sistemas multiagentes se aplican ampliamente en juegos de computadora, animación cinematográfica y defensa. En transporte, coordinan rutas para optimizar tiempos de viaje, mientras que en logística gestionan la distribución eficiente de mercancías. En sistemas de información geográfica, analizan grandes conjuntos de datos espaciales, generando mapas inteligentes. En diagnóstico médico, analizan síntomas y datos clínicos, asistiendo a los médicos en el proceso diagnóstico.

Las aplicaciones no tienen límites, los sistemas multiagentes pueden resolver problemas de cualquier tipo con buena coordinación y roles claves para resolver el problema.

Aplicación open-source

AutoGen

AutoGen es un framework de código abierto, que se posiciona como una herramienta esencial para el desarrollo de aplicaciones LLM de próxima generación que se centran en la conversación entre múltiples agentes. Su potencial se destaca por su capacidad para simplificar la orquestación, automatización y optimización de flujos de trabajo complejos de LLM, maximizando así el rendimiento de los modelos y superando sus limitaciones.

Principales Características

- **Maximización del Rendimiento y Automatización Avanzada:** AutoGen facilita la orquestación de flujos de trabajo complejos de LLM y maximiza su rendimiento, permitiendo la automatización eficiente y la optimización de tareas.
- **Admite Diversos Patrones de Conversación y Agentes:** Con la capacidad de manejar patrones de conversación complejos, AutoGen permite a los desarrolladores crear interacciones conversacionales autónomas, variando en cantidad de agentes personalizables y topologías de conversación.
- **Colección de Sistemas de Trabajo:** AutoGen proporciona una colección diversa de sistemas de trabajo, abarcando aplicaciones de diversos dominios y complejidades, demostrando su capacidad para adaptarse a una variedad de escenarios.
- **Inferencia LLM Mejorada:** Ofreciendo inferencia LLM mejorada, AutoGen incluye utilidades como unificación y almacenamiento en caché de API, así como patrones de uso avanzados como manejo de errores, inferencia de múltiples configuraciones y programación contextual.

Problemática a solucionar con un sistema multiagente

Problemática

Uno de los desafíos recurrentes en sistemas de gestión y logística de recursos es la capacidad de administrar eficientemente estos activos, tanto en situaciones normales como en respuesta a catástrofes o crisis. La gestión adecuada de recursos se vuelve crucial para mantener operaciones fluidas y, aún más, para abordar situaciones de emergencia.

La magnitud y complejidad de los datos relacionados con los recursos, ya sean materiales, humanos o financieros, a menudo supera la capacidad de un solo agente. En particular, la administración eficaz de grandes volúmenes de datos y la logística precisa de recursos requieren un enfoque más colaborativo y distribuido. Aquí es donde entra en juego la aplicación de sistemas multiagente como una solución estratégica.

Descentralización para una Gestión Efectiva

Un solo agente se vería abrumado por la tarea de procesar y administrar vastas cantidades de información relacionada con los recursos. La descentralización, característica clave de los sistemas multiagente, permite distribuir la carga de trabajo entre varios agentes autónomos. Cada agente, especializado en ciertos aspectos o tipos de recursos, puede contribuir a la gestión global de manera eficiente.

Optimización de Respuestas a Consultas y Aplicación

En situaciones críticas, la capacidad de responder de manera rápida y precisa a consultas es esencial. Un sistema multiagente se convierte en una herramienta valiosa al optimizar la respuesta a consultas por parte de diversos usuarios, que pueden ser organizaciones, entidades gubernamentales, personal de instituciones o incluso empleados de comercios.

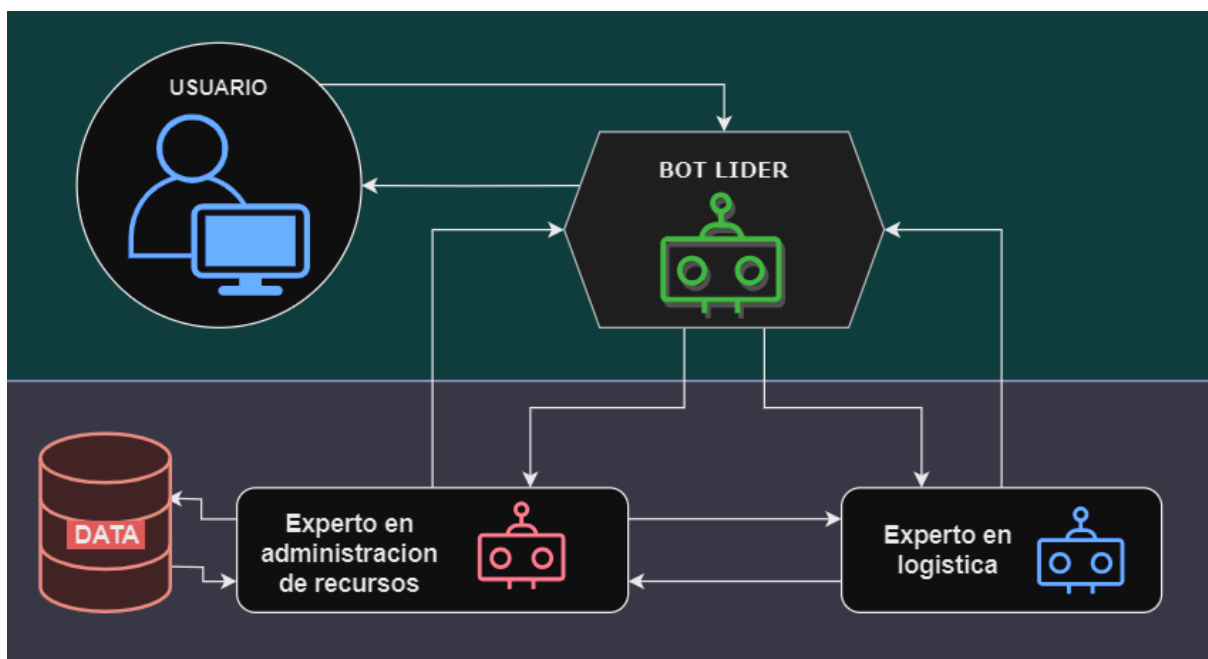
La colaboración entre los agentes permite una evaluación más holística de la situación, teniendo en cuenta la diversidad de recursos disponibles y las necesidades específicas de cada usuario. La flexibilidad de estos sistemas facilita la adaptación a diferentes contextos y escenarios, desde la gestión cotidiana de recursos hasta la respuesta coordinada en situaciones críticas.

Esquema multiagente

El esquema del sistema multiagente está compuesto por un agente líder, un agente experto en administración de recursos y un agente experto en logística.

El rol del agente líder es interactuar con el usuario y coordinar con el resto de agentes expertos. Este se encarga de considerar las consultas y enviarlas a los respectivos agentes para realizar una respuesta, este enfoque de resolución de consultas en esta situación problemática es crucial ya que la consulta del usuario es procesada anteriormente, analizada y enviada con un contexto óptimo a los agentes. La coordinación es crucial para que funcionen de manera correcta, por eso también es necesario un agente líder que organice al resto.

La integración de los agentes expertos entre sí también es crucial ya que permite interactuar entre ellos generando una retroalimentación, eliminando la necesidad de dar respuestas individuales sin tener conocimientos de lo que propone el otro agente y pasando por el agente líder como intermediario en las respuestas.



Ejemplo de despliegue y conversación entre agentes y usuarios

Usuario: Inicia la interacción solicitando asesoramiento sobre la gestión de recursos y logística.

Bot Líder: Analiza la solicitud del usuario y, tras una evaluación inicial, decide involucrar al bot experto en administración de recursos para abordar la complejidad de la pregunta.

Bot Experto en Administración de Recursos: Recibe la señal del bot líder, examina la situación planteada por el usuario y comienza a desplegar estrategias especializadas para mejorar la gestión de recursos, centrándose en la eficiencia y la coordinación.

Bot Experto en Logística: Detecta la necesidad de optimizar la logística en el proceso ya que puede interactuar de forma directa con el agente experto en administración de recursos y, en consecuencia, interviene para ofrecer orientación específica sobre la distribución de recursos y la adaptación de rutas, ajustándose a las demandas particulares del usuario.

Bot Líder: Devuelve una respuesta que aborda la problemática de una forma integral por la coordinación e interacción de los agentes involucrados.

Este flujo de interacción refleja un enfoque jerárquico y especializado en el cual cada bot desempeña un papel específico en función de sus habilidades y conocimientos. La coordinación entre los bots permite proporcionar respuestas precisas y contextualizadas, abordando la problemática planteada por el usuario de manera integral.

fuentes

Web

<https://iccsi.com.ar/multiagentes-inteligencia-artificial/#::~:~:text=En%20el%20C3%A1mbito%20de%20la,geogr%C3%A1fica%20y%20el%20diagn%C3%B3stico%20m%C3%A9dico.>

<https://microsoft.github.io/autogen/docs/Getting-Started>

<https://arxiv.org/abs/2308.08155>

Youtube

https://www.youtube.com/watch?v=XK8GR2iI_4E

<https://www.youtube.com/watch?v=10FCv-gCKug>