



Universidad
Nacional
de Rosario



TECNICATURA UNIVERSITARIA EN INTELIGENCIA ARTIFICIAL (TUIA)

PROCESAMIENTO DEL LENGUAJE NATURAL (NLP)

Trabajo Práctico 2 (TP2)

Estudiante:

Martín L. Perrone Leone

Legajo: P-5197/7

AÑO 2023 (2do CUATRIMESTRE), Rosario.

Ejercicio 1

Para realizar el chatbot primero descargos los pdf que uso como fuente de conocimiento externa. Para este ejercicio uso tres archivos que tienen los reglamentos de tres deportes, fútbol, basquet y tenis. Son los siguientes:

- Futbol_fifa.pdf
- 2023-rules-of-tennis-spanish.pdf
- Reglas Oficiales de Baloncesto FIBA 2022_V2.pdf

Una vez levantados los tres pdf, los junto en una variable, les hago split y después genero Embeddings utilizando el modelo **Paraphrase-multilingual-mpnet-base-v2** de Hugging Face

(<https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2>).

Es un es un modelo de sentence-transformers que mapea oraciones y párrafos a un espacio vectorial de 768 dimensiones y puede utilizarse para tareas como clustering o búsqueda semántica.

Ya teniendo la base de datos vectorial, para realizar **La Generación Aumentada por Recuperación (RAG)** utilizamos la librería llama_index y el modelo **Zephyr de Hugging Face** (<https://huggingface.co/HuggingFaceH4/zephyr-7b-beta>), es una serie de modelos de lenguaje entrenados para actuar como asistentes útiles.

Por último uso **ChatInterface de Gradio**, para crear una interfaz de usuario de chatbot.

El chatbot no siempre responde con el contexto que se le pasa, a pesar de que en el armado del prompt le indico que solamente responda con la información que le paso o diga que no puede responder. Por otro lado hay partes de los textos que parece no encontrar todas las veces que se le consulta, cambia en cada uso. También hay errores que parecen generarse por el google colaboratory, que cada tanto hace reiniciar la sesión, y en algunos casos no lo carga. Todo esto puede solucionarse con más tiempo de desarrollo y utilizando medios más adecuados.

https://github.com/martinperrone/TP2_NLP

Ejercicio 2

Qué es un Agente Inteligente? Características, tipos y cómo funciona

En el desarrollo constante de la inteligencia artificial, los agentes inteligentes cobran cada vez más relevancia en el ámbito tecnológico, y es que, su extrema utilidad en los numerosos aspectos del mundo real, hacen de esta **entidad autónoma** un sistema que se aplicará cada vez más, y seremos testigos de sus grandes ventajas cuando experimentemos los beneficios que aportan a la humanidad.

Definición de agente inteligente

En inteligencia artificial, un agente inteligente es un **sistema perceptivo** capaz de interpretar y procesar la información que recibe de su entorno, actuando en consecuencia de acuerdo a los datos que recoge y procesa. La forma de actuar de esta entidad es lógica y racional basándose en las reacciones del comportamiento normal de un sistema en concreto. Utiliza **sensores** para recibir información y **actuadores** para ejercer sus funciones.

Características de los agentes inteligentes

- El agente tiene capacidad de raciocinio.
- El agente aprende por sí mismo en función de la información que recibe y guarda.
- Puede interactuar con el entorno que le rodea.
- Tiene la capacidad de cooperar con otros agentes para cumplir una labor en concreto.
- Un agente puede tomar decisiones propias y obrar según su criterio.
- El comportamiento de un agente es determinado de acuerdo a la información percibida del entorno.
- Si el agente tiene una secuencia de actuaciones, es debido a la captación continua de información de la cual es sensible.

Tipos de agentes inteligentes

Los agentes inteligentes se clasifican en 6 tipos diferentes, donde cada uno distingue su utilidad y características que lo definen:

1. Agente de reactivo simple: Cuando una percepción en concreto coincide con una regla programada, el agente responde según la forma en que fue predispuesto. Este accionar se conoce como condición-acción. Toman decisiones basadas únicamente en la percepción actual, respondiendo a señales ambientales inmediatas sin ninguna memoria interna de eventos pasados.

Ejemplo: Un termostato que enciende el aire acondicionado cuando la temperatura actual supera un determinado umbral es un simple agente reflejo.

2. Agente reactivo basado en modelo: Este tipo de agente permite simular su acción de respuesta y sus diversas interacciones en un entorno preparado. De esta forma se estudia su comportamiento y sus efectos en el espacio de actuación. Utilizan sensores para recabar información y tienen en cuenta el historial de percepciones, lo que permite tomar decisiones más sofisticadas.

Ejemplo: Una IA que juega al ajedrez y que tiene en cuenta el historial de jugadas y el estado actual del tablero para decidir la siguiente jugada es un agente basado en modelos.

3. Agente basado en metas: Combina características del agente reactivo simple y agente reactivo basado en modelo. En este caso, este tipo de agente tiene un objetivo en concreto, por lo tanto, está programado para buscar la vía más óptima y planificar un conjunto de acciones para cumplir dicho propósito.

4. Agente basado en utilidad: Este agente tiene varios propósitos como sistema inteligente, además, posee una herramienta para medir el valor de su comportamiento en el cumplimiento de sus metas establecidas. Sus estándares de conducta garantizan alta calidad en sus acciones.

5. Agente que aprende: Es un tipo de agente que busca aprender de sus acciones mientras se encuentra en funcionamiento. Es un sistema altamente complejo, ya que está programado para interactuar con el mundo real, además de tener preestablecidas varias metas a alcanzar. Posee en su interior un elemento que indica el éxito de la entidad, y tiene la capacidad de interactuar en entornos que no conoce.

Ejemplo: Un filtro de spam que aprende a identificar nuevos tipos de correos basura basándose en los comentarios de los usuarios es un agente de aprendizaje.

6. Agente de consulta: Se dedica a responder consultas por parte de las personas que interactúan con este sistema. Tiene la peculiaridad de **crear varios agentes**, y dividir la pregunta del usuario en varias tareas para su respectiva solución. Además, en caso de que los agentes asignados no sean capaces de responder con exactitud la incógnita enviada, se crearán más agentes y se buscarán en más bases de datos para ofrecer una **resolución completa de la problemática**.

¿Cómo funciona un agente inteligente?

Un agente inteligente funciona mediante la **obtención constante de datos percibidos del entorno** en el que se encuentra. De acuerdo a los datos obtenidos, la entidad inteligente tiene un comportamiento u otro. Además, debido a que posee autonomía propia, puede asignarse nuevas reglas para continuar su crecimiento como sistema intelectual.

Agentes inteligentes: sus aplicaciones en el mundo

- Mantenimiento preventivo y correctivo de diferentes tipos de maquinaria.
- Aplicación en la gestión de riesgos.
- Para el cuidado de la reputación online.
- Ayuda en los procesos internos de la inteligencia territorial.
- Sistemas de análisis de la competencia laboral.
- Elaboración de agentes con la capacidad de mantener una comunicación fluida con usuarios.

Algunas empresas:

ChatDev

ChatDev es una empresa de software virtual que opera a través de varios agentes inteligentes que desempeñan diferentes roles, incluyendo CEO, CPO, CTO, programador, revisor, probador y diseñador. Estos agentes forman una estructura organizativa de múltiples agentes y están unidos por una misión para "revolucionar el mundo digital a través de la programación". ChatDev IDE es una extensión de navegador que permite conectar sin problemas las conversaciones entre varios agentes dentro de un navegador web. Consta de tres partes: Modo de Juego, Modo de Chat y Prompt IDE2. Puedes personalizar estos NPCs, personalizar el indicador de ubicación y construir tus GPTs con el editor de indicadores visuales. Además, ChatDev se lanzó como una plataforma SaaS que permite a los desarrolladores de software e innovadores empresarios construir software de manera eficiente a un costo muy bajo y con una barrera de entrada muy baja.

AutoGPT

AutoGPT cambia drásticamente la relación entre la inteligencia artificial y el usuario final (es decir, tú). Mientras que ChatGPT se basa en un intercambio entre la IA y el usuario, donde tú le das una solicitud y la IA devuelve un resultado, Auto-GPT solo necesita un mensaje inicial de tu parte. A partir de ahí, el agente de IA generará una lista de tareas que cree que necesita para cumplir con lo que le pediste, sin requerir más

información o mensajes. Esencialmente, encadena “pensamientos” de modelos LLM según el desarrollador Toran Bruce Richards.

Auto-GPT es un sistema complejo que depende de múltiples componentes. Se conecta a internet para obtener información y datos específicos (algo que la versión gratuita de ChatGPT no puede hacer), cuenta con gestión de memoria a corto y largo plazo, utiliza GPT-4 para la generación de texto más avanzada de OpenAI y GPT-3.5 para almacenamiento y resumen de archivos.

Ejecutar AutoGPT es bastante simple, solo debes seguir los pasos descritos en el repositorio de **Github**: <https://github.com/Significant-Gravitas/Auto-GPT> Una vez que lo ejecutas te pedirá el nombre, el rol, y sus objetivos (goals). Ejemplo: el nombre es AutoGPT-Demo, el rol es “Una inteligencia artificial diseñada para enseñarme sobre Auto gpt” y sus objetivos son “buscar auto gpt, buscar el repositorio en github y averiguar de que se trata el proyecto, explicar que es autogpt en un archivo autogpt.txt, finalizar”.

De esta forma podemos ver como el agente se pone a trabajar y crea una AI que tiene un rol y objetivos específicos, y hará todo lo necesario para cumplir con sus metas.

BabyAGI

BabyAGI es otro proyecto que se centra en el desarrollo de agentes autónomos con habilidades de planificación a largo plazo y uso de memoria. Al igual que AutoGPT, BabyAGI busca crear agentes que puedan aprender y adaptarse a su entorno utilizando técnicas de aprendizaje por refuerzo. BabyAGI es una versión simplificada que demuestra cómo la inteligencia artificial puede mejorar la eficiencia en la gestión de tareas.

¿Cómo funciona BabyAGI?

El script de BabyAGI opera en un bucle infinito que realiza los siguientes pasos: Extrae la primera tarea de la lista de tareas. Envía la tarea al agente de ejecución, que utiliza la API de OpenAI para completarla en función del contexto. Enriquece el resultado y lo almacena en Pinecone. Crea nuevas tareas y reorganiza la lista de tareas según el objetivo y el resultado de la tarea anterior.

Integración con OpenAI y Pinecone

BabyAGI aprovecha las capacidades de procesamiento del lenguaje natural (NLP) de OpenAI para crear nuevas tareas basadas en el objetivo predefinido. Pinecone, por otro lado, se utiliza para almacenar y recuperar los resultados de las tareas, proporcionando contexto para futuras tareas.

Simulaciones de Agentes: CAMEL y Generative Agents

Las simulaciones de agentes son entornos en los que se pueden probar y evaluar agentes autónomos. Los proyectos CAMEL y Generative Agents se centran en la creación de entornos de simulación específicos para agentes autónomos.

CAMEL

CAMEL es un proyecto que se centra en hacer que dos agentes con personalidades distintas interactúen entre sí de manera colaborativa en un entorno de simulación específico.

Los agentes tienen memoria a corto y largo plazo y utilizan pasos de reflexión para asignar puntuaciones de importancia a las observaciones y generalizar lo que han aprendido. La novedad de CAMEL radica en la interacción entre dos agentes y la colaboración en un entorno de simulación.

Generative Agents

Generative Agents es otro proyecto que se centra en la creación de entornos de simulación para agentes autónomos. Estos agentes se basan en modelos generativos para imitar comportamientos individuales y grupales que son similares a los humanos, y se fundamentan en sus “identidades, entorno y experiencias cambiantes”. Como parte de un estudio, investigadores de Stanford y Google crearon un entorno interactivo inspirado en el popular videojuego de simulación The Sims.

Este entorno de “sandbox” permitió a los investigadores estudiar los comportamientos individuales y sociales de los agentes de inteligencia artificial. Algunos de los agentes demostraron preferencias y objetivos similares a los humanos, como el caso de la Agente Alice, que es pintora.

¿Cómo funciona Generative Agents?

Los Generative Agents siguen una arquitectura que consta de componentes como observación, reflexión y planificación. La arquitectura también puede almacenar registros de las experiencias de un agente mediante el uso de lenguaje natural. Según el artículo de investigación, los usuarios pudieron interactuar con los agentes a través del lenguaje natural. Además, los agentes generativos también tenían la capacidad de entablar conversaciones. En el experimento, se descubrió que los agentes generativos pueden realizar diversas tareas, como preparar desayunos, ir al trabajo, pintar, formar relaciones, compartir opiniones e iniciar conversaciones. Son sorprendentemente similares a los humanos, ya que pueden recordar, recuperar y reflexionar sobre el pasado y el presente. Los Generative Agents son un concepto innovador que combina lo mejor de la inteligencia artificial y la simulación interactiva. Tienen la capacidad de crear simulaciones realistas del comportamiento humano que los usuarios pueden explorar. Sin embargo, al igual que cualquier otra tecnología, los agentes generativos también requieren una consideración cuidadosa en cuanto a sus

implicaciones éticas, ya que la idea de que los humanos coexistan con seres virtuales conlleva una serie de desafíos.

LangChain

LangChain es un marco para desarrollar aplicaciones impulsadas por modelos de lenguaje. Permite aplicaciones que: Son conscientes del contexto: conectan un modelo de lenguaje a fuentes de contexto (instrucciones de comandos, ejemplos breves, contenido para fundamentar sus respuestas, etc.). Razonan: dependen de un modelo de lenguaje para razonar (sobre cómo responder basado en el contexto proporcionado, qué acciones tomar, etc.). Este marco consta de varias partes. Bibliotecas de LangChain: Las bibliotecas de Python y JavaScript. Contienen interfaces e integraciones para una multitud de componentes, un tiempo de ejecución básico para combinar estos componentes en cadenas y agentes, e implementaciones listas para usar de cadenas y agentes. Plantillas de LangChain: Una colección de arquitecturas de referencia fácilmente desplegables para una amplia variedad de tareas. LangServe: Una biblioteca para implementar cadenas de LangChain como una API REST. LangSmith: Una plataforma para desarrolladores que te permite depurar, probar, evaluar y monitorear cadenas construidas en cualquier marco de modelos de lenguaje grande (LLM) e integra perfectamente con LangChain. Las bibliotecas de LangChain en sí mismas se componen de varios paquetes diferentes.

Problemática: Gestión eficiente de un sistema de transporte público en una ciudad grande.

Un sistema multiagente podría ser útil para gestionar y optimizar un sistema de transporte público en una ciudad grande. Los agentes podrían trabajar juntos para minimizar los tiempos de viaje, equilibrar la demanda y la oferta de vehículos, y mejorar la satisfacción general de los pasajeros.

Agentes involucrados en esta tarea:

- 1. Agente de Rutas:** Agente responsable de gestionar las rutas de los vehículos de transporte público. Utiliza datos en tiempo real sobre el tráfico y la demanda de pasajeros para optimizar las rutas y minimizar los tiempos de viaje.
- 2. Agente de Horarios:** Agente se encarga de crear y ajustar los horarios de los vehículos de transporte público. Trabaja en conjunto con el Agente de Rutas para asegurar que los vehículos estén disponibles cuando y donde se necesiten.
- 3. Agente de Mantenimiento:** Este agente programa el mantenimiento regular de los vehículos para asegurar su funcionamiento óptimo. También maneja cualquier problema o avería imprevista.
- 4. Agente de Información al Pasajero:** Este agente proporciona información en tiempo real a los pasajeros sobre los horarios de los vehículos, las rutas y cualquier cambio o retraso. Esto podría hacerse a través de una aplicación móvil.
- 5. Agente de Control de Tráfico:** Este agente monitorea el tráfico en la ciudad y proporciona esta información a los otros agentes. Esto permitiría al Agente de Rutas ajustar las rutas en tiempo real para evitar el tráfico pesado y al Agente de Horarios ajustar los horarios en consecuencia.