AUTOGEN STUDIO: una herramienta de desarrollo sin código para crear y Depuración de sistemas multiagente

Victor Dibia, Jingya Chen, Gagan Bansal, Suff Syed,
Adam Fourney, Erkang Zhu, Chi Wang, Saleema Amershi
Microsoft Research, Redmond, Estados Unidos
{victordibia, jingyachen, gaganbansal, suffsyed, adam.fourney, erkang.zhu, chiw,
Samershi}@microsoft.com

Abstracto

Los sistemas multiagente, donde colaboran múltiples agentes (modelos de IA generativa + herramientas), se están consolidando como un patrón eficaz para resolver tareas complejas y de larga duración en numerosos dominios. Sin embargo, especificar sus parámetros (como modelos, herramientas y mecanismos de orquestación , etc.) y depurarlos sigue siendo un reto para la mayoría de los desarrolladores. Para abordar este reto, presentamos AUTOGEN STUDIO, una herramienta de desarrollo sin código que permite prototipar, depurar y evaluar rápidamente flujos de trabajo multiagente, basada en el framework AUTOGEN .

AUTOGEN STUDIO ofrece una interfaz web y una API de Python para representar agentes compatibles con LLM mediante una especificación declarativa (basada en JSON) . Proporciona una interfaz intuitiva de arrastrar y soltar para la especificación del flujo de trabajo del agente, la evaluación y depuración interactivas de flujos de trabajo, y una galería de componentes reutilizables del agente. Destacamos cuatro principios de diseño para herramientas de desarrollo multiagente sin código y contribuimos con una implementación de código abierto.[1]

1 Introducción

Cuando se combinan con la capacidad de actuar (p. ej., usando herramientas), los modelos de IA generativa funcionan como agentes, lo que permite capacidades complejas de resolución de problemas. Es importante destacar que investigaciones recientes han demostrado que la transición de tuberías de agentes prescritas (fijas) a una configuración de múltiples agentes con capacidades autónomas puede dar como resultado comportamientos deseables, como una mejor factualidad y razonamiento (Du et al., 2023), así como un pensamiento divergente (Liang et al., 2023). Estas observaciones han impulsado el desarrollo de marcos de aplicación como AutoGen (Wu et al., 2023), CAMEL (Li et al., 2024) y TaskWeaver (Qiao et al., 2023), que simplifican el proceso de creación de aplicaciones multiagente expresadas como código Python. Sin embargo, a medida que las aplicaciones multiagente avanzan

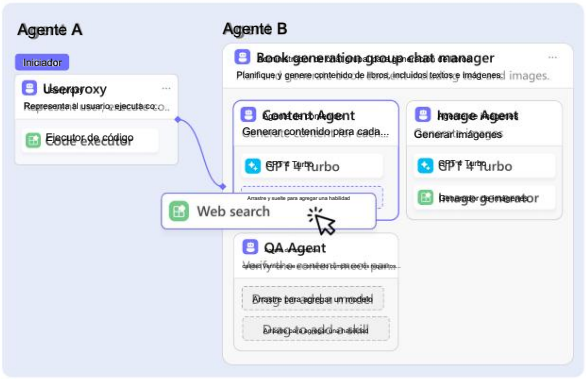

Figura 1: AUTOGEN STUDIO proporciona una interfaz de usuario de arrastrar y soltar donde se pueden definir modelos, habilidades/ herramientas y componentes de memoria , adjuntarlos a agentes y adjuntarlos a flujos de trabajo.

Si bien nuestra capacidad para resolver problemas complejos también presenta nuevos desafíos, los desarrolladores ahora deben configurar una gran cantidad de parámetros para estos sistemas, incluyendo la definición de agentes (p. ej., el modelo a utilizar, las indicaciones, las herramientas o habilidades disponibles para el agente, el número de pasos que puede realizar, las condiciones de finalización de la tarea, etc.), y los mecanismos de comunicación y orquestación, es decir, el orden o la secuencia en que los agentes actúan al colaborar en una tarea. Además, los desarrolladores necesitan depurar y comprender las interacciones complejas de los agentes para extraer señales que permitan mejor
Todos estos factores pueden crear importantes barreras de entrada y hacer que el proceso de diseño multiagente sea tedioso y propenso a errores. Para abordar estos desafíos , hemos desarrollado AUTOGEN STUDIO, una herramienta para prototipado, depuración y evaluación rápida de flujos de trabajo multiagente . Nuestras contribuciones se destacan a continuación:

- AUTOGEN STUDIO : una herramienta enfocada en el desarrollo (IU, backend, web y API de Python) para especificar y depurar de forma declarativa flujos de trabajo multiagente (con intervención humana y sin interacción) . AUTOGEN STUDIO ofrece una novedosa

[1] https://github.com/microsoft/autogen/tree/autogenstudio/muestras/aplicaciones/autogen-studio

Experiencia de arrastrar y soltar (Figura 1) para una rápida creación de flujos de trabajo complejos de agentes MULTI-AGENTE , herramientas para perfilar/depurar sesiones de agente y una galería de componentes MULTI- AGENTE reutilizables/compartibles .

- Introducimos capacidades de creación de perfiles con visualizaciones de mensajes/acciones por agentes y métricas (costos, invocaciones de herramientas y estado de salida de herramientas) para depurar flujos de trabajo MULTIAGENTE .
- Basándonos en nuestra experiencia construyendo y dando soporte AUTOGEN STUDIO como herramienta de código abierto con una base de usuarios significativa (más de 200.000 descargas) dentro de un período de 5 meses), describimos patrones de diseño emergentes para desarrolladores MULTI-AGENTE Herramientas y futuras direcciones de investigación.

Hasta donde sabemos, AUTOGEN STU- DIO es el primer proyecto de código abierto que explora un Interfaz sin código para MULTIAGENTE autónomo desarrollo de aplicaciones, proporcionando una plataforma adecuada para la investigación y la práctica en MULTI-AGENTE herramientas para desarrolladores.

2 trabajos relacionados

2.1 Agentes (LLMs + Herramientas)

Los modelos de IA generativa enfrentan limitaciones, incluidas Alucinación: generar contenido sin fundamento De hecho, y un rendimiento limitado en el razonamiento. tareas o nuevos problemas fuera de distribución. Para abordar estas cuestiones, la práctica ha virado hacia Implementaciones de agentes donde se dan modelos acceso a herramientas para actuar y aumentar su rendimiento (Mialon et al., 2023). Las implementaciones de agentes, como React (Yao et al., 2022), exploran Un paradigma de Razón y Acción que utiliza LLMs para generar tanto rastros de razonamiento como tareas específicas acciones de forma intercalada. Como parte de esto proceso, los desarrolladores han explorado marcos que construir canales prescriptivos entrelazando modelos y herramientas (por ejemplo, LIDA (Dibia, 2023), LangChain (Chase, 2022)). Sin embargo, a medida que las tareas se vuelven más complejas, Al requerir un contexto extenso y la capacidad de adaptarse de manera independiente a espacios de problemas dinámicos, las tuberías predefinidas demuestran un rendimiento limitado. (Liu et al., 2024). Esta limitación ha llevado a la Exploración de agentes más flexibles y adaptativos arquitecturas.

2.2 Marcos multiagente

Se han propuesto varios marcos para proporcionar abstracciones para crear tales aplicaciones. AutoGen (Wu et al., 2023) es un marco extensible de código abierto que permite a los desarrolladores crear grandes aplicaciones MULTIAGENTE . CAMEL (Li et al., 2024) está diseñado para facilitar la autonomía cooperación entre agentes comunicativos a través de Juego de roles, utilizando indicaciones iniciales para guiar agentes de chat hacia la finalización de tareas mientras se alinean con las intenciones humanas. OS-Copilot (Wu et al., 2024) introduce un marco para construir agentes generalistas capaces de interactuar con elementos integrales en un sistema operativo, incluido el web, terminales de código, archivos, multimedia y varios aplicaciones de terceros. Explora el uso de una Módulo planificador dedicado, un configurador y un ejecutor, así como el concepto de herramientas (Python funciones o llamadas a puntos finales de API) o habilidades (herramientas que se puede aprender y reutilizar sobre la marcha).

Conceptos básicos de múltiples agentes

1. Modelo: Modelo de IA generativa utilizado para Impulsar los comportamientos básicos de los agentes.
2. Habilidades/Herramientas: Código o API utilizadas para ad- tareas específicas del vestido.
3. Memoria: a corto plazo (por ejemplo, listas) o a largo plazo. término (bases de datos vectoriales) utilizado para guardar y recordar información.
4. Agente: Una configuración que une El modelo, las habilidades, los componentes de la memoria y comportamientos.
5. Flujo de trabajo: una configuración de un conjunto de agentes y cómo interactúan para abordar tareas (por ejemplo, orden o secuencia en que agentes actúan, planificación de tareas, terminación condiciones, etc.).

En conjunto, estas herramientas respaldan un conjunto de funciones básicas: capacidades - definición de parámetros del agente - tales como modelos de IA generativos , habilidades/herramientas o memoria, y flujos de trabajo de agentes : especificaciones de cómo estos Los agentes pueden colaborar. Sin embargo, la mayoría de estos Los marcos admiten principalmente una representación de flujos de trabajo de agentes de código primero , lo que presenta un alto barrera de entrada y prototipado rápido. También No proporcione herramientas ni métricas para la depuración del agente y evaluación. Además, carecen de estructura Plantillas reutilizables para iniciar o acelerar el Proceso de creación del flujo de trabajo del agente. AUTOGEN STU-

DIO aborda estas limitaciones proporcionando una interfaz visual para definir y visualizar declarativamente los flujos de trabajo de los agentes, probarlos y evaluarlos, y ofrecer plantillas para tareas comunes de MULTI-AGENTE para optimizar el desarrollo. Si bien este trabajo se basa en la biblioteca de código abierto AUTOGEN (Wu et al., 2023) y hereda las abstracciones principales para la representación de agentes, los patrones de diseño propuestos en herramientas de desarrollo sin código están diseñados para ser aplicables a todos los frameworks MULTI-AGENTE .

3 objetivos de diseño

AUTOGEN STUDIO está diseñado para mejorar la experiencia del desarrollador MULTI-AGENTE centrándose en tres objetivos principales: Creación rápida
de prototipos: proporcionar un espacio donde los desarrolladores puedan especificar rápidamente configuraciones de agentes y componer estos agentes en flujos de trabajo multiagente efectivos.
Herramientas para desarrolladores: ofrece herramientas diseñadas para ayudar a los desarrolladores a comprender y depurar los comportamientos de los agentes, lo que facilita la mejora de los sistemas multiagente .

Plantillas reutilizables: Presentamos una galería de plantillas reutilizables y compartibles para impulsar la creación de flujos de trabajo de agentes. Este enfoque busca establecer estándares compartidos y mejores prácticas para el desarrollo de sistemas multiagente , promoviendo una mayor adopción e implementación de soluciones multiagente .

4 Diseño del sistema

AUTOGEN STUDIO se implementa en dos componentes de alto nivel: una interfaz de usuario (UI) frontend y una API backend (web, Python y línea de comandos). Se puede instalar mediante el gestor de paquetes PyPI (listado 1).

```
pip install autogenstudio autogenstudio
ui -- puerto 8081
```

Listado 1: AUTOGEN STUDIO se puede instalar desde PyPI (pip) y la interfaz de usuario se puede iniciar desde la línea de comandos.

4.1 Interfaz de usuario

La interfaz web frontend en AUTOGEN STU-
DIO está construido utilizando React e implementa tres vistas principales que admiten varias funcionalidades clave.
La vista de compilación permite a los usuarios crear (definir y componer) flujos de trabajo multiagente
La vista permite la ejecución interactiva de tareas y la depuración del flujo de trabajo, con opciones para exportar e implementar.
La vista de galería facilita la reutilización y el uso compartido de plantillas de artefactos del agente.

4.1.1 Creación de flujos de trabajo

La vista de creación en la IU (ver Figura 1) ofrece una experiencia de definición y composición, que permite a los desarrolladores definir de forma declarativa componentes de bajo nivel y componerlos iterativamente en un flujo de trabajo. Por ejemplo, los usuarios pueden definir configuraciones para modelos , habilidades/herramientas (representadas como funciones de Python que abordan tareas específicas) o almacenes de memoria (p. ej., documentos organizados en una base de datos vectorial). Cada entidad se guarda en una base de datos para su uso en las interacciones de la interfaz. Posteriormente, pueden definir un agente, adjuntándole modelos, habilidades y memoria . Se proporcionan varias plantillas predeterminadas de agente siguiendo las abstracciones de AUTOGEN : un agente UserProxy (tiene una herramienta de ejecución de código de forma predeterminada), un AssistantAgent (tiene un modelo de IA generativo predeterminado) y un agente GroupChat (un contenedor de abstracción para definir una lista de agentes y cómo interactúan). Finalmente, se pueden definir flujos de trabajo, con agentes existentes asociados a estos. Los patrones de flujo de trabajo predeterminados admitidos son el chat autónomo (los agentes intercambian mensajes y acciones durante los turnos de conversación hasta que se cumple una condición de finalización) y el chat secuencial (una secuencia de agentes definida, cada agente procesa su entrada en orden y envía un resumen de su salida al siguiente agente).
El proceso de composición del flujo de trabajo se mejora aún más al admitir una interacción de arrastrar y soltar; por ejemplo, las habilidades/modelos se pueden arrastrar a los agentes y los agentes a los flujos de trabajo.

4.1.2 Flujos de trabajo de prueba y depuración. Los

flujos de trabajo pueden probarse in situ en la vista de compilación o explorarse de forma más sistemática en la vista de prueba. Esta vista permite a los usuarios crear sesiones, adjuntar flujos de trabajo a la sesión y ejecutar tareas (de una sola vez o en varios turnos). Las sesiones se pueden compartir (para ilustrar el rendimiento del flujo de trabajo) y comparar varias sesiones. AUTOGEN STUDIO ofrece dos funciones para facilitar la depuración. En primer lugar, proporciona una vista de observación donde, a medida que avanzan las tareas, los mensajes y las acciones realizadas por los agentes se transmiten a la interfaz y se muestran todos los artefactos generados (p. ej., archivos como imágenes, código, documentos, etc.). En segundo lugar, se proporciona una vista de perfilador post-hoc donde se visualiza un conjunto de métricas para cada tarea abordada por un flujo de trabajo.

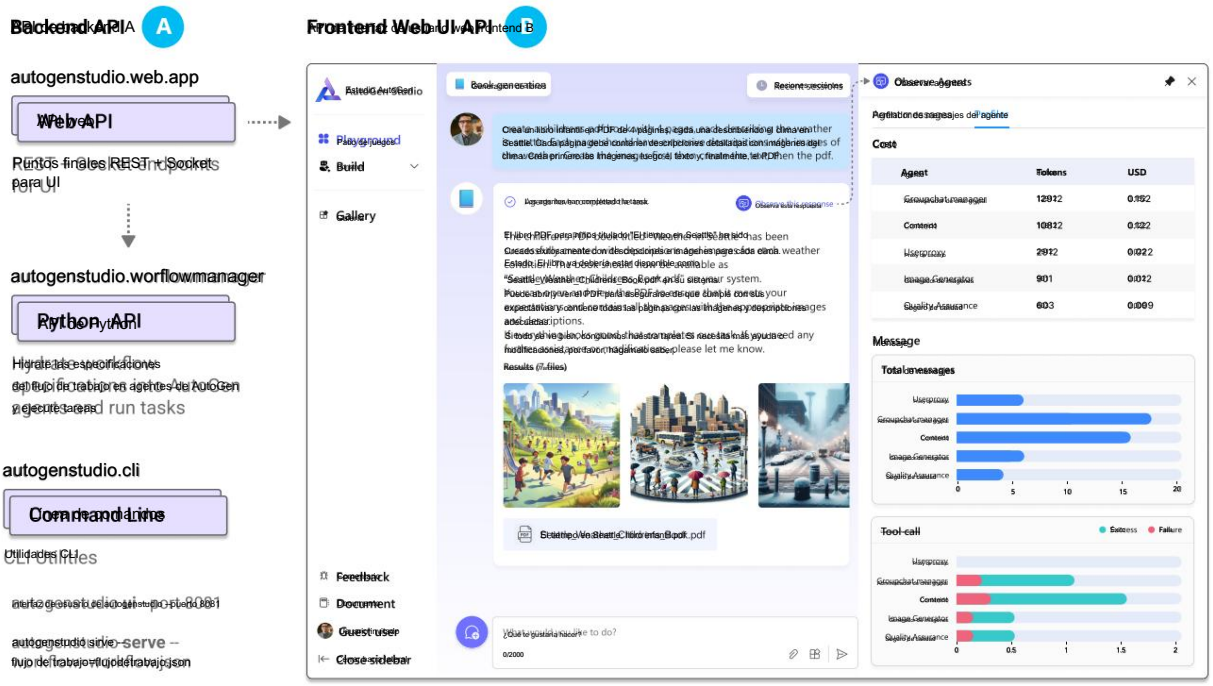

Figura 2: AUTOGEN STUDIO proporciona una API de backend (web, Python, CLI) y una interfaz de usuario que implementa un área de juegos (mostrado), vista de compilación y galería. En la vista de área de juegos, los usuarios pueden ejecutar tareas en una sesión según un flujo de trabajo. Usuarios También puede observar las acciones realizadas por los agentes, revisando los mensajes y las métricas de los agentes en función de un módulo generador de perfiles.

número total de mensajes intercambiados, costos (tokens del modelo de IA generativa consumidos y costos en dólares), Con qué frecuencia los agentes utilizan herramientas y el estado de las herramientas uso (éxito o fracaso), para cada agente.

4.1.3 Implementación de flujos de trabajo

AUTOGEN STUDIO permite a los usuarios exportar flujos de trabajo como un archivo de configuración JSON. Un archivo exportado El flujo de trabajo se puede integrar perfectamente en cualquier Aplicación Python (listado 2), ejecutada como una API punto final utilizando el comando AUTOGEN STUDIO interfaz de línea (figura 2a), o envuelta en un Docker contenedor para implementación a gran escala en varios plataformas (Azure, GCP, Amazon, etc.).

```
desde la importación de autogenstudio
    Administrador de flujo de trabajo
wm = WorkflowManager ("flujo de trabajo .
    json ")
. ejecutar ( mensaje =" ¿Cuál es el
    altura de la Torre Eiffel ")
```

Listado 2: Los flujos de trabajo se pueden importar en aplicaciones de Python.

4.1.4 Galería de plantillas

La interfaz de usuario también cuenta con una vista de galería: un repositorio de componentes (habilidades, modelos, agentes, flujos de trabajo) que los usuarios pueden importar, ampliar y reutilizar en sus propios flujos de trabajo. Dado que cada especificación de componente es declarativo (JSON), los usuarios también pueden exportar fácilmente, versión y volver a compartirlas.

4.2 API de backend: Web, Python y Línea de comandos

La API de backend consta de tres componentes principales: una API web, una API de Python y una interfaz de línea de comandos. La API web consta de REST. puntos finales creados utilizando la biblioteca FastAPI[2] , que admiten los métodos HTTP GET, POST y DELETE. Estos puntos finales interactúan con varias clases clave: Un administrador de DBM realiza CRUD (Crear, Leer, Operaciones (actualizar, eliminar) sobre varias entidades como como habilidades, modelos, agentes, memoria, flujos de trabajo y sesiones. La clase WorkflowManager maneja la ingesta de flujos de trabajo de agente declarativo, los convierte en objetos de agente AUTOGEN y ejecuta tareas (ver listado 2). Una clase Profiler analiza Mensajes del agente para calcular métricas. Cuando un usuario... inicia una tarea dentro de una sesión, el sistema recupera El historial de sesiones, instancia agentes en función de sus representaciones serializadas de la base de datos, ejecuta la tarea, transmite mensajes intermedios a la interfaz de usuario a través de websocket y devuelve los resultados finales. AUTOGEN STUDIO también proporciona una línea de comandos

[2]API rápida: https://fastapi.tiangolo.com/

Interfaz con utilidades para iniciar la interfaz de usuario incluida.
y ejecutar flujos de trabajo exportados como puntos finales de API.

5 Uso y evaluación

En este proyecto hemos adoptado un enfoque iterativo in situ.
Enfoque de evaluación. Desde su lanzamiento en GitHub
(5 meses), el paquete AUTOGEN STUDIO tiene
Se ha instalado más de 200.000 veces y se ha mejorado
iterativamente en función de los comentarios de uso (>
135 problemas de GitHub). Problemas destacados por varios usuarios
puntos críticos que se abordaron posteriormente , incluidos: (a) desafíos
para definir, persistir y
reutilización de componentes, resuelta mediante la implementación de un
capa de base de datos; (b) dificultades en la creación de componentes ,
resueltas mediante el apoyo a la generación automatizada de herramientas a
partir de descripciones y la integración de un IDE para
herramientas de edición; (c) frustraciones causadas por los componentes
Fallos durante las pruebas de extremo a extremo, solucionados mediante la
incorporación de un botón de prueba para componentes (por ejemplo, modelos).
y flujos de trabajo en la vista de compilación. La Figura 3 muestra
Un gráfico de todos los números de AUTOGEN STUDIO . Cada punto
representa un problema, basado en una incrustación de su
texto (título + cuerpo) usando la incrustación de texto de
OpenAI - 3-large modelo. Las incrustaciones se redujeron a
dos dimensiones utilizando UMAP, agrupadas con K-
Means (k = 8) y etiquetas de agrupamiento generadas utilizando
GPT-4 (basado en 10 muestras de su centroide).
Finalmente, en el Apéndice A , demostramos cómo
AUTOGEN STUDIO se puede utilizar eficazmente para apoyar
Una personalidad de ingeniero en prototipado rápido, pruebas,
y depurar iterativamente un flujo de trabajo MULTIAGENTE e
implementarlo como un punto final de API para abordar
una tarea concreta (generar libros).

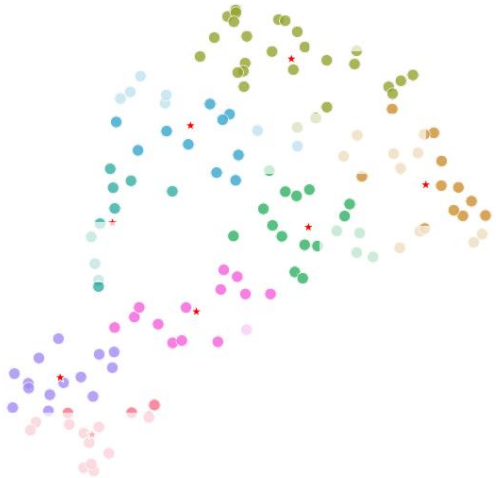

Figura 3: Gráfico de problemas de GitHub (n = 8 clústeres) de
El repositorio de AUTOGEN STUDIO . Los comentarios de los usuarios variaron.
del soporte con herramientas de creación de flujos de trabajo (por ejemplo, el
Capacidad de configurar y probar modelos) para la instalación general.

6 patrones de diseño emergentes y Direcciones de investigación

En la siguiente sección, describimos algunas de las
Patrones emergentes de alto nivel que esperamos puedan
ayudar a informar el diseño de interfaces sin código para
Construyendo aplicaciones multiagente de próxima generación.

6.1 Flujos de trabajo de definición y composición

> Permitir a los usuarios crear flujos de trabajo mediante
> definir componentes y componer
> ellos (a través de acciones de arrastrar y soltar) en
> flujos de trabajo multiagente.

Un sistema multiagente puede tener una amplia gama de
parámetros para configurar. Hemos descubierto que seleccionar la
presentación visual correcta del flujo de trabajo para
Ayudar a los usuarios a comprender qué parámetros configurar (descubrimiento)
y cómo configurarlos. En concreto , hemos descubierto que un método de
definición y composición
flujo de trabajo, donde las entidades primero se definen y persisten de forma
independiente, y luego se componen en última instancia
En flujos de trabajo multiagente, proporciona una buena experiencia
para el desarrollador. Esto incluye proporcionar herramientas.
para apoyar a las entidades de creación, por ejemplo, la capacidad de
definir y probar modelos, un IDE para generar/editar
herramientas (código) y un diseño visual basado en lienzo
de flujos de trabajo con interacción de arrastrar y soltar para
asociar entidades en el flujo de trabajo.

6.2 Herramientas de depuración y búsqueda de sentido

> Proporcionar herramientas robustas para ayudar a los usuarios a depurar,
> interpretar y racionalizar el comportamiento y
> Salidas de sistemas multiagente.

Los flujos de trabajo de múltiples agentes pueden ser frágiles y fallar por
Hay múltiples razones, que van desde modelos configurados
incorrectamente hasta instrucciones deficientes para los agentes,
configuración incorrecta de herramientas para los agentes o terminación.
condiciones. Una solicitud crítica ha sido la de herramientas
para ayudar a los usuarios a depurar y comprender las respuestas
del agente .

6.3 Exportación e implementación

> Habilite la exportación e implementación sin inconvenientes de flujos de trabajo de múltiples agentes para diversas plataformas y entornos.

Si bien una herramienta sin código como AUTOGEN STUDIO Permite la iteración rápida y la demostración de flujos de trabajo, la progresión natural para la mayoría de los casos de uso. es que los desarrolladores quieren replicar los mismos resultados, pero integrados como partes de sus aplicaciones principales. Esta etapa requiere una exportación fluida y Implementación de flujos de trabajo multiagente para varios plataformas y entornos.

6.4 Colaboración y compartición

> Facilitar la colaboración de los usuarios en el desarrollo del flujo de trabajo de múltiples agentes y permitir Fácil intercambio de creaciones dentro de la comunidad .

La colaboración y el intercambio son claves para acelerar la innovación y mejorar los sistemas multiagente. Al permitir que los usuarios colaboren en el desarrollo del flujo de trabajo, compartan sus creaciones y desarrollen sus conocimientos. A través del trabajo de cada uno se puede cultivar un entorno de desarrollo más dinámico e innovador. Herramientas y funciones que admiten la colaboración en tiempo real, el control de versiones y el uso compartido continuo de Los flujos de trabajo y los componentes son esenciales para fomentar un enfoque impulsado por la comunidad. Además, ofrecer un repositorio o galería donde los usuarios pueden publicar y compartir sus flujos de trabajo, habilidades y agentes promueve el aprendizaje comunitario y la innovación.

7 futuras direcciones de investigación

Si bien hemos explorado las primeras implementaciones de los requisitos de diseño mencionados anteriormente, nuestro Los esfuerzos para construir AUTOGEN STUDIO también han identificó dos importantes áreas de investigación futuras y Preguntas de investigación asociadas.

- Herramientas de evaluación fuera de línea: esto abarca Preguntas como cómo podemos medir el rendimiento , la confiabilidad y la reutilización de los agentes. ¿Cómo podemos comprender mejor las tareas? ¿Sus fortalezas y limitaciones? ¿Cómo podemos explorar escenarios y resultados alternativos? Y ¿Cómo podemos comparar diferentes arquitecturas de agentes y protocolos de colaboración?
- Comprender y cuantificar el impacto de decisiones de diseño de sistemas multiagente: Estas Las preguntas incluyen determinar el número y la composición óptimos de agentes para un problema dado , la mejor manera de distribuir responsabilidades y coordinar acciones entre los agentes, y la Compensaciones entre centralizado y descentralizado control o entre agentes homogéneos y heterogéneos.
- Optimización de sistemas multiagente: Investigación Las direcciones aquí incluyen la generación dinámica de agentes en función de los requisitos de la tarea y los recursos disponibles, ajustando las configuraciones del flujo de trabajo Para lograr el mejor rendimiento y adaptarse equipos de agentes para entornos cambiantes y usuarios preferencias. Además, ¿cómo podemos aprovechar Supervisión humana y retroalimentación para mejorar el agente ¿Fiabilidad, rendimiento de la tarea y seguridad?

8 Conclusión

Este documento presentó AUTOGEN STUDIO, una herramienta de desarrollo sin código para crear prototipos, depurar y evaluar rápidamente flujos de trabajo de múltiples agentes. Clave Las características incluyen una interfaz de arrastrar y soltar para el agente. composición del flujo de trabajo, capacidades de depuración interactivas y una galería de componentes de agente reutilizables. A través de una adopción generalizada, identificamos patrones de diseño emergentes para herramientas de desarrollo de múltiples agentes : un enfoque de definición y composición para la creación Flujos de trabajo, herramientas de depuración para comprender el agente comportamientos, herramientas para habilitar la implementación y funciones de uso compartido colaborativo. AUTOGEN STUDIO reduce la barrera de entrada para el desarrollo de aplicaciones multiagente , lo que potencialmente acelera la innovación en el campo. Finalmente, describimos las futuras direcciones de investigación, incluido el desarrollo de herramientas de evaluación fuera de línea, Estudios de ablación para cuantificar el impacto de las decisiones y métodos de diseño de sistemas MULTIAGENTE para Optimización de sistemas multiagente.

9 Declaración de ética

AUTOGEN STUDIO está diseñado para proporcionar un entorno sin código para la creación rápida de prototipos y pruebas de flujos de trabajo multiagente. Nuestro objetivo es impulsar responsablemente la investigación y la práctica en la resolución de problemas con múltiples agentes y desarrollar herramientas que Contribuir al bienestar humano. Junto con AUTOGEN , AUTOGEN STUDIO se compromete a implementar funciones que promuevan un entorno seguro y confiable. Resultados. Por ejemplo, AUTOGEN STUDIO ofrece herramientas de creación de perfiles para comprender las acciones de los agentes.

y salvaguardas, como la compatibilidad con entornos Docker para la ejecución de código. Esta función ayuda a garantizar que los agentes operen en entornos controlados y seguros, lo que reduce el riesgo de acciones no deseadas o dañinas. Para más información sobre nuestro enfoque de IA responsable en AutoGen, consulte las preguntas frecuentes sobre transparencia aquí. Finalmente, AUTOGEN STUDIO no está listo para producción, es decir, no se enfoca en implementar la autenticación y otras medidas de seguridad que se requieren para implementaciones listas para producción.

Expresiones de gratitud

Agradecemos a los miembros de la comunidad de software de código abierto (OSS) y a la organización AI Frontiers de Microsoft Research por las conversaciones y los comentarios que nos brindaron durante el proceso. En concreto, agradecemos a Piali Choudhury, Ahmed Awadallah, Robin Moeur, Jack Gerrits, Robert Barber, Grace Proebsting, Michel Pahud, Qingyun Wu, Harsha Nori y otros por sus comentarios.

Referencias

Harrison Chase. 2022. LangChain. Github.

Victor Dibia. 2023. Lida: Una herramienta para la generación automática de visualizaciones e infografías gramaticales independientes utilizando modelos lingüísticos grandes. Preimpresión arXiv arXiv:2303.02927.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum e Igor Mordatch. 2023. Mejora de la facticidad y el razonamiento en modelos lingüísticos a través del debate multiagente. Preimpresión de arXiv arXiv:2305.14325.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin y Bernard Ghanem. 2024. Camel: agentes comunicativos para la exploración de la "mente" de una sociedad modelo de lenguaje amplio. Avances en los sistemas de procesamiento de información neuronal, 36.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu y Shuming Shi. 2023. Fomentar el pensamiento divergente en grandes modelos lingüísticos a través del debate entre múltiples agentes. Preimpresión de arXiv arXiv:2305.19118.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape , Michele Bevilacqua, Fabio Petroni y Percy Liang. 2024. Perdidos en el medio: Cómo los modelos lingüísticos utilizan contextos largos. Transacciones de la Asociación de Lingüística Computacional, 12:157–173.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Modelos de lenguaje aumentado : una encuesta. Preimpresión de arXiv arXiv:2302.07842.

Bo Qiao, Liqun Li, Xu Zhang, Shilin He, Yu Kang, Chaoyun Zhang, Fangkai Yang, Hang Dong, Jue Zhang, Lu Wang, et al. 2023. Taskweaver: un marco de agente que prioriza el código. Preimpresión de arXiv arXiv:2311.17541.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadal-lah, Ryen W White, Doug Burger y Chi Wang. 2023. Autogen: Habilitación de aplicaciones llm de próxima generación a través del marco de conversación de múltiples agentes. archivo

Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu y Lingpeng Kong. 2024. Os-copilot: Hacia agentes informáticos generalistas con superación personal. Preimpresión de arXiv arXiv:2402.07456.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan y Yuan Cao. 2022. React: Sinergizando razonamiento y acción en modelos de lenguaje. Preimpresión de arXiv arXiv:2210.03629.

Un personaje de Jack, el ingeniero de software

Caso de uso

Jack es un ingeniero de software júnior que se unió recientemente a SoftwareCon. Como parte de sus tareas, debe crear una aplicación que genere diversos libros cortos. La versión inicial debe centrarse en la generación de libros infantiles (de 5 a 8 años) a partir de una consulta específica (p. ej., crear un libro infantil sobre el funcionamiento del sol) con la expectativa de que se generalice para dar soporte a otras tareas genéricas. Jack ha oído hablar de un enfoque MULTIAGENTE para construir sistemas que puedan abordar una variedad de tareas a través de la colaboración autónoma entre agentes. Para explorar este enfoque, comienza leyendo la documentación de AUTOGEN STUDIO , lo instala, inicia la interfaz de usuario y realiza los siguientes pasos;

A.1 Paso 1: Definir y componer un flujo de trabajo

Jack comienza con la vista de construcción, donde revisa las habilidades predeterminadas que vienen con AUTOGEN STUDIO. Él ve que hay dos habilidades relevantes Generar_pdfs y generar_imágenes. Verifica que tenga las claves API adecuadas para la habilidad Generar_imágenes . A continuación, crea un modelo GPT3.5 y añade una clave API.

Siguiendo las mejores prácticas, Jack sabe que el equipo básico de agentes con AUTOGEN consta de un UserProxyAgent que puede ejecutar código y un AssistantAgent que puede resolver tareas, además de escribir código o invocar las herramientas/habilidades disponibles. Crea ambos agentes; para su AssistantAgent, se asegura de adjuntar el modelo GPT4 que creó previamente y también de adjuntar ambas habilidades. Jack pasa a la pestaña de flujo de trabajo y crea un nuevo flujo de trabajo de chat autónomo donde especifica el UserProxyAgent como iniciador y su AssistantAgent como receptor.

A.2 Paso 2: Probar e iterar

En la pestaña de flujo de trabajo, Jack prueba el flujo de trabajo inmediatamente y observa rápidamente algunos problemas. Usando la herramienta de perfilado y visualizando los mensajes intercambiados por los agentes, observa que parece haber problemas de calidad con el contenido del libro; concretamente, el Agente Asistente parece generar mensajes muy cortos, por lo que las páginas del libro contienen solo dos oraciones por página, a pesar de que los requisitos indican que los niños son un poco mayores y pueden leer textos mucho más largos.

Para remediar estos problemas, Jack toma dos medidas. Primero, intenta ampliar las instrucciones base. de su Agente Asistente, pero aún no obtiene páginas con más de 3 oraciones en las pruebas interactivas.

Recuerda que usar más agentes puede ayudar a separar el enfoque y mejorar el rendimiento de las tareas. Luego, pasa a crear cuatro agentes: un UserProxy y un ContentAssistant con instrucciones detalladas sobre... Genera el contenido de cada página, un Asistente de Control de Calidad verifica que las páginas cumplan con los parámetros y un Asistente de Generación de Imágenes genera imágenes para el libro. Posteriormente, crea un agente de GroupChat y le añade su lista de agentes. A continuación, crea un nuevo flujo de trabajo donde el receptor es el agente de GroupChat y prueba la aplicación varias veces. Jack está satisfecho con los resultados, ya que las historias de página completa se generan correctamente. Además, a Jack le preocupan los costos, pero puede usar fácilmente el botón de observación de mensajes para explorar la duración, los tokens utilizados por los agentes, el uso de herramientas/habilidades y los costos en dólares de LLM para cada ejecución de tarea.

A.3 Paso 3: Exportar y compartir En este

punto, Jack tiene dos tareas finales: quiere compartir su trabajo con colegas para recibir retroalimentación y luego proporcionar una API con la que puedan crear prototipos. AUTOGEN STUDIO facilita compartir; primero, Jack puede simplemente exportar y compartir un enlace a sesiones exitosas . Segundo, también puede descargar su flujo de trabajo y compartirlo con colegas, guardándolo en un sistema de control de versiones como Git. Tercero, puede activar un punto final de API donde los agentes pueden responder a las solicitudes de tareas usando comandos de CLI 'autogenstudio serve –port 8000'. También puede activar un contenedor docker usando el comando AUTOGEN STUDIO serve y escalarlo en cualquier plataforma de su elección (Azure, AWS, GCP, Hugging Face).