Introduction to AI Agents and Agent Use Cases

**ИИ агенты** - это системы, которые позволяют **LLM** выполнять действия расширяя их возможности за счёт предоставления им доступа к инструментам и данным.

Составляющие определения:

* Система - это не одна компонента, а набор компонент. В которые входят:
  + Окружение - определенное пространство, где ИИ агент действует. Например, система бронирования билетов. Окружение предоставляет информацию и механизмы обратной связи.
  + Сенсоры - это то, что использует агент для получения информации о текущем состоянии окружения. Например, работа с API для сбора информации о доступных рейсах.
  + Исполнительный механизм - как только агент получает состояние окружения, он определяет список действий к выполнению. Например, бронирование отелей для пользователя.

Преимущество построения агентов с ллм - это способность ллм понимать человеческий язык и данные.

Это позволяет определять состояние окружения и определять список действий направленных на изменение этого окружения.

* Выполняющие действия - вне системы ИИ агента ллм ограничена генерацией контента и информацией основанной на подсказке (промпте) пользователя. Внутри системы ИИ агента система может интерпретировать запросы пользователей и выполнять их посредством доступных инструментов в окружении.
* Доступ к инструментам - определяется: окружением системы, разработчиком ИИ системы. Например, для агента резервирования поездок - окружение ограничивает ИИ систему агента доступными функциями окружения и разработчиком (списком подходящих рейсов).
* Память и знания - коротко временная, это контекст пользователя и агента. Длинно-временная - ИИ агент может брать знания не из окружения, а из других систем, агентов, инструментов. Например, в контексте ИИ агента для брони билетов, знаниями могут быть предпочтения пользователей в выборе стран.

Типы ИИ-агентов:

1. Простой реагирующий агент (simple reflex agents) – выполняет действие за счёт предписанных правил.

Пример: агент путишествий интерпретирует жалобу пользователя и пересылает пользовательскому сервису.

1. Модельный реагирующий агент (model-based agents) – выполняет действие на основе модели мира и изменений этой модели.

Например: агент путишествий определяет приоритеты маршрутов со значительными историческими изменениями цен на основе доступной исторической информации цен.

1. Целевой агент (goal-based agents) – агент создаёт план достижения определённой цели за счёт интерпретации цели и действий её достижения.

Например: агент бронирует путишествие определяя необходимые условия (автобус, такси, самолёты) для заказа.

1. Полезный агент (utility-based agents) – считает предпочтения и веса численных компромиссов для определения того, как достигнуть цели.

Например: агент путишествий максимизирует пользу взвешиванием удобств против цены при выборе поездки.

1. Обучающийся агент (learning agents) – улучшается со временем реагируя на обратную связь и регулировании соответствующие действия.

Например: агент улучшается засчёт обратной связи от пользующегося пользователя за счёт опроса после того, как клиент приехал с путишествия за счёт внесения улучшений в будущие тур поездки.

1. Иеррархический агент (hierarchical agents) – агент мультиагентной системы, который разбивает задачи на подзадачи для простых агентов.

Например: агент путишествий отменяет поездку за счёт разбиения здаания на подзадания и низкоуровневые агенты выполняют эти хадания при этом отдавай ответ высокоуровненвому агенту.

1. Мульти-агентная система (multi-agent systems) – агенты выполняют задачи уникальные для них, или вместе, или конкуретно.

Например: коооперативно – каждый агент выполняет свою задачу для общей цели, конкурентно – агенты соревнуются в поиске лучших вариантов резервирования отеля.

Когда использовать ИИ-агентов:

* Открытые проблемы (open-ended problems) - может позволить LLM определить необходимые шаги.
* Мульти-шаговые проблемы (multi-step processes) - задачи, требующие несколько этапов выполнения ИИ агентом работы, когда агенту нужно использовать инструмент и информацию множественных запросов.
* Задачи улучшающиеся со временем (improvement over time) - это задачи, которые улучшаются со временем, благодаря обратной связи пользователей или окружения.

Шаги проектирования ИИ-агентов:

* Проектирования системы ИИ агентов: определение инструментов, действий, и поведений.
* Второй шаг шаблонизация агентов: взаимодействие с LLM чрез подсказки, мы не меняем промпты со временем, а агент сам меняет их для работы LLM.
* Третий шаг: инструменты агентов: эти инструменты позволяют реализовывать паттерны агентов чрез код. Они предлагают шаблоны, инструменты для лучшего взаимодействия с ИИ. Они позволяют лучше рассматривать и диагнозировать системы агентов.

Explore AI Agent Frameworks

Инструменты ИИ агентов – это программные платформы разработанные для: упрощения работы, размещения и управления ИИ агентами. Они обеспечивают разработчиков готовыми блоками, абстракциями и инструментами, которые упрощают разработку сложны ИИ систем.

Эти инструменты помогают разработчикам фокусироваться на уникальности их приложений за счёт обеспечения стандартными подходами к обычным задачам в ИИ разработке. Они усиливают масштабируемость, доступность и эффективность построения ИИ систем.

Инструменты ИИ агентов позволяют:

* Персонализация: ИИ может анализировать поведение пользователя и предпочтения, чтобы обеспечить персональные рекомендации, контент.

Например: Стриминговые сервисы как Netflex используют ИИ для предложения фильма основанное на истории просмотров, усиливает удовлетворение пользователя.

* Автоматизация и эффективность: ИИ могут автоматизировать повторяющиеся задачи, оптимизировать процессы и улучшить оптимизацию вычислений.

Например: приложение пользователя использует оснащённые ИИ чатботы для управления обычными запросами, сокращения времени ответов, освобождает людей для более сложных задач.

* Улучшение пользовательского опыта: ИИ может улучшить пользовательский опыт засчёт обеспечения интеллектуальными функциями: распознование текста, обработка естественного языка и предугадывание текста.

Например: виртуальный помощник как Siri используют ИИ для понимания и отклика на голосовые команды, которые делают жизнь пользователей легче при взаимодействии с устройствами.

Инструменты ИИ агентов представляют что-то большее, чем просто ИИ инструменты. Они разработаны, чтобы дать возможность создать интеллектуальных агентов, которые могут взаимодействовать с пользователями, другими агентами, и окружением, чтобы достичь определённых целей. Эти агенты могут показать автономное поведение, принимать решения и адаптироваться под изменяющиеся условия.

Ключевые возможности доступные в инструментах ИИ агентов:

* Взаимодействие агентов и их координация: возможность создать множество агентов, которые могут работать вместе, коммуницировать и координироваться для решения сложных задач.
* Автоматизация задач и управления: обеспечивает механизмами для автоматизации многошагового рабочего процесса, разбиения задач и динамического управления задачами среди агентов.
* Понимание контекста и адаптация: наделить агентов возможностями понимать контекст, адаптироваться под изменения окружения и принимать решения на основе свежей информации.

Итого, агенты позволяют Вам делать больше, для получения автоматизации следующего уровня, для создания более интеллектуальных систем, которые могут адаптироваться и учиться по окружению.

Это быстродвижущаяся почва, но существуют вещи которые обычно есть в инструментах ИИ агентов, которые могут помочь Вам быстро прототипировать и повторять компоненты модулей, совместные инструменты, и обучение в реалном времени:

* Использование модульных компонент: ИИ инструменты предоставляют подготовленные компоненты, такие как ИИ и памятные коннекторы, вызов функций используюищх LLM или программные плагины, шаблоны подсказок.
* Использование совместных инструментов: разработка агентов с особыми ролями и задачами, позволящие им тестировать улучшать совместный рабочий процесс.
* Обучение в реальном времени: осуществление циклов обратной связи, где агенты учатся при взаимодействии и адаптируются в реальном времени.

Использование модульных компонент. Инструменты Semantic Kernel и LangChain предоставляют готовые компоненты, такие как коннекторы, шаблоны подсказок и управление памяти. Как команды могут это использовать? Команды могут быстро собрать эти компоненты для создания функционального прототипа без боли, позволяюще быстро экспериментировать и итерироваться.

Как это работает на практике?

Вы можете использовать предразработанный ИИ коннектор с семантическим ядром, которые использую авто вызовы функций, чтобы модель могла откликнуться на пользовательский ввод.

Использование совместных инструментов.

Инструменты такие как: CrewAI, MS AutoGen и Semantic Kernel содействуют созданию множеством агентов, которые могут работать вместе.

Как команды могут использовать это: команды могут разрабатывать агентов с особыми ролями и задачами, позволяющие им тестировать и улучшать взаимодействие в рабочих процессах и улучшать общую эффективность системы.

Как это работает на практике?

Вы можете создать команду агентов, где каждый агент имеет спициализированные функции, такие как извлечение данных, анализ или принятие решений. Эти агенты могут коммуницировать и делится информацией для достижения обычных целей, таких как ответы на вопросы пользователя или выполнения заданий.

Обучение в реальном времени.

Современнные инструменты обеспечивают возможности понимание контекста в реальном времени и адаптации.

Как команды могут это использовать?

Команды могут выполнять циклы обратной связи, где агенты учатся при взаимодействии и адаптируются своё поведение динамически, ведя длительное улучшение и улучшению возможностей.

Как это работает на практике?

Агенты могут анализировать обратную связь, данные окружения, и задачи из вне для обновления базы знаний, адаптировать алгоритмы принятия решений и улучшать работу со временем. Это итерративный процесс обучения позволяет агентам адаптироваться под изменяющиеся условия и предпочтения пользователей, усилия полную эффективность системы.

В чём разница между AutoGen, Semantic Kernel, Azure AI Agent Service?

Существует множество отличающихся признаков, но мы рассмотрим следующие ключевые: дизайн, возможности и цель использования.

AutoGen – инструмент открытого источника от Microsoft. Он нацелен на управление событиями, распределённых приложений агентов, поддерживает множество LLM и SLMs, инструментам, и продвинутые шаблоны разработки мульти-агенства.

Он построен вокруг основной концепции ИИ агентов, которые является автономными сущностями, которые могут воспринимать их окружения, принимать рещения, и брать действия для достижения особых целей. Агенты коммуницируют через асинхронные сообщения, позволяющие им работать с независимо и параллельно, усиливая мастабируемость и восприимчивость системы.

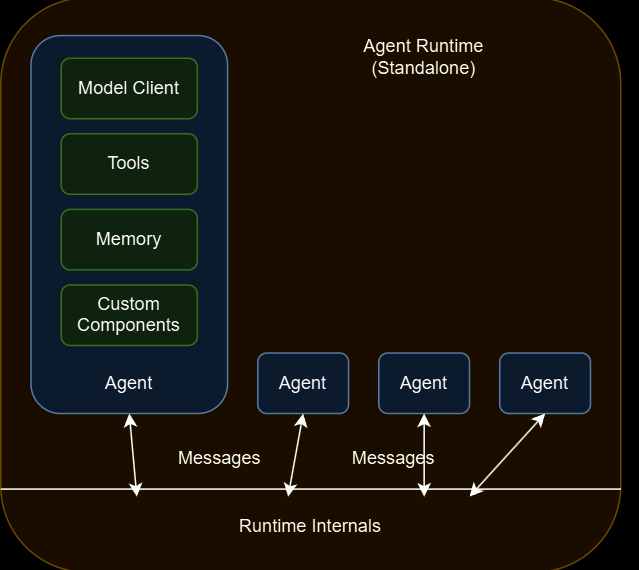
Агенты основанные на актёрской модели. Актёр – это базовый строительный блок параллельных вычислений.

В ответ на полученное сообщение актёр может: сделать локальные вычисления, создать больше актёров, отправить больше сообщений и определить как реагировать на следующие полученные сообщения.

Использование: автоматическая генерация кода, задачи анализа данных и построение особых агентов для планирования и исследования функций.

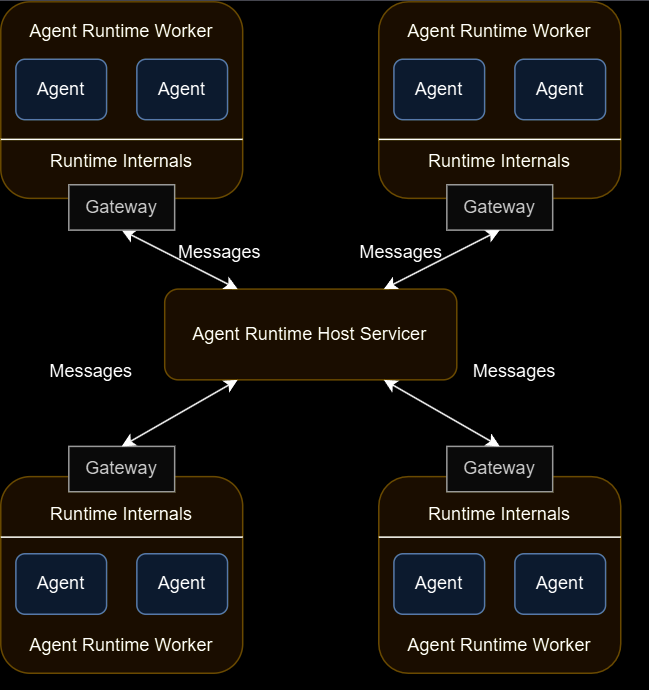
Особые концепции AutoGen:

* Агенты. Это программное приложение, которое:
  + Общаются через синхронные или асинхронные сообщения.
  + Поддерживают своё состояние, которое может быть изменено входящими сообзениями.
  + Выполняют действия в ответ на полученное сообщение или меняют состояние. Эти действия могут улучшить состояние и создать внещние эффекты, такие как обновление логов, отправки нового сообщения, запуск кода или использование API.
* Мульти агенты (multi agents). AutoGen поддерживает создание множества агентов, которые могут работать на выполнение одной цели. Агенты могут коммуницироваться, делиться информацией и координироваться их действия для эффективного решения проблемы. Чтобы создать мульти агентную ситему вы можете определить разлчиные типы агентов со специальными функциями и ролями такими как; извлечение данных, анализ, принятие решений и взаимодействие с пользователем.
* Агент времени выполнения (agent runtime). Инструмент обеспечивает а окружение времени выполнения позволяя коммуницировать агентам, управлять их личностями и жизнями, и принуждать границы безопасности и приватности. Это означает, что вы можете запустить своего агента в безопасной и контроллируемой среде, обеспечивая безопасность и эффективность. Виды интересов агента времени выполнения.
  + Автономная среда выполнения (stand-alone runtime). Это хороший выбор для одного процесса где агенты реализованы на одном программном языке и выполняют похожий процесс.



Агенты коммуницируют чрез сообщения в времени выполнения и время выполнения управляет их жизнями.

* + Распределённое время выполнения (Distributed runtime). Это подходит для мультипроцессных приложений, где агенты могут реализованы в разных программных языках и запускаться разными машинами.



Semantic Kernel + Agent Framework

Semantic Kernel – корпоративный SDK для оркестрирования ИИ. Он состоит из ИИ и коннекторов памяти, вместе с ИИ инструментом.

* ИИ коннекторы – это интерфейс с расширенным ИИ сервисом и источником данных.
* Плагин – это инкапсулированная функция которую может использовать приложение. Существуют как готовые плагины так и вы можете написать свой сами. Родственное понятие – функция быстрого доступа (prompt function). Вместо обеспечения подсказок на естественном языке для вызова функций, вы даёте определённые функции модели. Основываясь на текущем контексте чата модель может выбрать для вызова одну из этих функций чтобы выполнить запрос.
* Родные функции (native functions) – существуют также родные функции которые инструменты могут вызвать напрямую чтобы выполнить задачу.
* Память. Абстрагирует и упрощает управление контестом ИИ приложений.