Складена AI-система із використанням агентів

Сучасний розвиток штучного інтелекту призвів до появи складених AI-систем (Compound AI Systems), які об'єднують кілька агентів або моделей для виконання складних завдань. Ці системи дозволяють створювати **гнучкі**, **адаптивні** та **інтелектуальні рішення**, що перевершують можливості окремих AI-моделей.

У цьому розділі розглянемо, що таке складені AI-системи, які їхні основні характеристики, та як у них працюють агенти.

2. Що таке складені AI-системи?

Складені AI-системи — це **інтегровані багатокомпонентні рішення**, що поєднують кілька моделей штучного інтелекту або агентів для вирішення складних завдань. На відміну від монолітних AI-моделей, ці системи базуються на **взаємодії між різними компонентами**, які можуть бути спеціалізовані для різних типів завдань.

2.1. Основні характеристики складених AI-систем:

- **Модульність** – система складається з автономних компонентів, які можуть працювати незалежно або у взаємодії.
- **Гнучкість** – можливість додавання, заміни та адаптації окремих агентів.
- **Розподілене прийняття рішень** – різні агенти відповідають за різні аспекти обробки даних та прийняття рішень.
- **Оркестрація агентів** – центральний модуль або агент керує координацією всієї системи.
- **Здатність до навчання та адаптації** – використання механізмів постійного оновлення знань та поліпшення точності роботи.

2.2. Приклади складених AI-систем

- **Автономні транспортні системи**: поєднання комп'ютерного зору, прогнозування траєкторій та оптимізації маршрутів.
- **AI-системи для наукових досліджень**: інтеграція алгоритмів обробки природної мови (NLP) для аналізу наукових публікацій, генерації гіпотез та перевірки їх експериментальними даними.
- **AI-системи в бізнесі**: об'єднання агентів для фінансового прогнозування, аналізу ризиків та автоматизації робочих процесів.

3. Що таке агенти в складених AI-системах?

Агенти в складених AI-системах – це автономні програмні сутності, які виконують окремі функції та взаємодіють між собою для досягнення спільної мети. Вони можуть мати різні рівні складності – від простих правилових агентів до самонавчальних моделей.

3.1. Основні типи агентів

- **Оркестратор (Orchestrator Agent)** – центральний керуючий агент, який розподіляє завдання між іншими агентами та координує роботу системи.
- **Агенти збору даних (Data Retrieval Agents)** – відповідають за пошук, вилучення та попередню обробку інформації з різних джерел (API, бази даних, веб-ресурси).
- **Генеративні агенти (Generative Agents)** – використовують моделі GPT, LLaMA або PaLM для створення текстових відповідей, прогнозів або звітів.
- **Аналітичні агенти (Analytical Agents)** – аналізують отриману інформацію, виявляють закономірності та здійснюють статистичну обробку.
- **Верифікаційні агенти (Validation Agents)** – перевіряють коректність даних, узгодженість отриманих результатів та їх відповідність запиту користувача.
- **Комунікаційні агенти (Interface Agents)** – забезпечують взаємодію з користувачем, формують інтерфейси для введення та виведення даних.

3.2. Взаємодія агентів у складених AI-системах

Складені AI-системи можуть працювати за різними архітектурними моделями:

1. **Ієрархічна модель** – агенти працюють у чіткій структурі, де кожен рівень виконує певні завдання (наприклад, аналіз → генерація → перевірка).
2. **Розподілена модель** – агенти взаємодіють між собою без центрального керування, розподіляючи завдання динамічно.
3. **Гібридна модель** – поєднує централізовану координацію із гнучким розподілом ресурсів між агентами.

3.3. Використання агентів у складених AI-системах

- **Оркестратор отримує запит**, аналізує його та визначає необхідні кроки для виконання.
- **Агенти збору даних отримують релевантну інформацію** з баз знань, API або документів.
- **Генеративні агенти створюють відповідь** на основі оброблених даних.
- **Верифікаційні агенти перевіряють точність відповіді** та коригують помилки.
- **Комунікаційні агенти забезпечують взаємодію** між користувачем та системою.

4. Переваги використання складених AI-систем

- **Гнучкість**: можна додавати або видаляти компоненти без значних змін у загальній архітектурі.

- **Масштабованість**: легко адаптується до великих обсягів даних та складних завдань.
- **Модульність**: дозволяє інтегрувати різні AI-технології та фреймворки.
- **Ефективність**: агенти працюють паралельно, що знижує час обробки інформації.
- **Висока точність**: за рахунок спеціалізації агентів система мінімізує помилки.

Складені AI-системи – це новий етап розвитку штучного інтелекту, що дозволяє інтегрувати кілька агентів для вирішення складних завдань. Вони забезпечують модульність, масштабованість та високу ефективність у порівнянні з традиційними AI-моделями. Агенти в таких системах відіграють ключову роль, виконуючи спеціалізовані завдання, що дозволяє досягати високої продуктивності та точності в автоматизованих процесах. Подальші дослідження можуть зосередитися на оптимізації взаємодії агентів та підвищенні рівня автономності таких систем.

Останні досягнення в галузі штучного інтелекту (AI) суттєво розширили можливості автоматизації складних процесів, зокрема у сфері підтримки життєвого циклу сервісів (ЖЦС). Використання великих мовних моделей (LLM) дозволило значно покращити взаємодію людини з інформаційними системами, але наявні підходи мають низку обмежень. Сучасні LLM орієнтовані переважно на генерацію природномовних текстів, що обмежує їхню здатність до глибокого аналізу та прийняття складних рішень у реальному часі.

Одним із перспективних напрямів розвитку є використання **складених AI-систем на основі агентної архітектури**, які інтегрують LLM із багатокомпонентними AI-агентами. Такі системи не лише розширюють функціональність окремих моделей, а й забезпечують ефективну оркестрацію завдань, адаптивність до змін і автономність у прийнятті рішень. Зокрема, останні дослідження, проведені в Берклі [1], а також розробки в LangChain, CrewAI та інших платформах демонструють потенціал **багатоагентних AI-систем** у вирішенні складних завдань.

У цій статті ми пропонуємо **концепцію AI-агентної системи**, яка використовує багаторівневу агентну архітектуру для підтримки ЖЦС у динамічному середовищі. Головна ідея полягає в створенні **ієрархічної системи агентів**, де кожен агент має чітко визначену роль і взаємодіє з іншими агентами через стандартизовані механізми координації та обміну даними. Запропонований підхід забезпечує масштабованість, модульність і гнучкість системи.

Концепція AI-агентної системи

1. Основні принципи

Складена AI-система базується на таких ключових принципах:

- **Модульність** – система складається з незалежних агентів, кожен з яких має певні функції та може бути вдосконалений без впливу на інші компоненти.

- **Ієрархічна координація** – агенти розподіляються за рівнями відповідальності: від стратегічних агентів, що керують процесами, до тактичних агентів, які виконують конкретні завдання.
- **Автономність** – агенти здатні приймати рішення на основі локальних даних та взаємодії з іншими агентами.
- **Інтерактивність** – агенти спілкуються з користувачем і між собою за допомогою природної мови, використовуючи LLM.
- **Адаптивність** – система навчається на основі історичних даних та змін у середовищі.

2. Архітектура AI-агентної системи

Пропонована архітектура складається з кількох рівнів агентів:

1. **Головний агент (Core Agent)** – центральний координатор, що приймає стратегічні рішення, контролює цілісність системи та керує комунікацією між агентами.
2. **Функціональні агенти (Task Agents)** – спеціалізовані агенти, які виконують певні завдання у межах життєвого циклу сервісів (аналіз, проєктування, впровадження, моніторинг, підтримка тощо).
3. **Експертні агенти (Specialist Agents)** – агенти, які інтегрують спеціалізовані моделі та алгоритми (нейромережі, оптимізаційні методи, аналітичні моделі) для вирішення специфічних проблем.
4. **Комунікаційні агенти (Interface Agents)** – агенти, що взаємодіють із користувачами, отримують команди та надають рекомендації на основі аналізу даних.

3. Взаємодія агентів

Взаємодія агентів у системі відбувається за допомогою **асинхронної обробки запитів** та механізмів управління намірами (Intention Management). Це дозволяє агентам не лише реагувати на команди користувача, а й проактивно пропонувати рішення, ґрунтуючись на отриманих даних.

Ключові механізми взаємодії:

- Використання **LLM** для розуміння запитів та генерації відповідей у природній мові.
- Інтеграція **Retrieval-Augmented Generation (RAG)** для доступу до зовнішніх знань у процесі ухвалення рішень.
- Реалізація **Multi-Agent Reinforcement Learning (MARL)** для оптимізації взаємодії між агентами.

Запропонована архітектура забезпечує можливість використання AI-агентів у широкому спектрі бізнес-процесів, дозволяючи автоматизувати складні завдання та зменшити залежність від людського втручання.

Архітектура AI-агентної системи

1. Загальний опис архітектури

Запропонована архітектура AI-агентної системи базується на **централізованій оркестрації** завдань, використанні **мультимодальної Retrieval-Augmented Generation (RAG) системи** та **взаємодії між агентами** через механізм делегування та перевірки результатів.

Основні принципи архітектури:

- **Модульність**: усі агенти незалежні, що дозволяє їх масштабувати або замінювати без впливу на інші компоненти системи.
- **Ієрархічна структура**: агенти згруповані за рівнями відповідальності, що дозволяє ефективно керувати інформаційними потоками.
- **Мультимодальність**: підтримка роботи з текстом, зображеннями, аудіо та іншими форматами даних.
- **Семантична індексація**: застосування ChromaDB для ефективного пошуку релевантної інформації.

Основними компонентами системи є:

- **Центральний агент (Orchestrator)** – отримує запити від користувача, розбиває їх на підзадачі та розподіляє між іншими агентами.
- **Мультимодальна RAG-система** – працює із векторною базою даних (ChromaDB) для пошуку релевантної інформації з документації.
- **Виконавчі агенти (Task Agents)** – спеціалізовані агенти, що виконують конкретні завдання (аналітика, генерація контенту, обробка даних тощо).
- **Агенти перевірки (Validation Agents)** – контролюють якість результатів та забезпечують їх відповідність вихідному запиту.
- **Інтерфейсний агент (Interface Agent)** – взаємодіє з користувачем, пояснює отримані результати та уточнює додаткові запити.

Технологічний стек включає **LangChain** для управління агентами та запитами, **CrewAI** для координації агентів, **ChromaDB** для ефективного векторного пошуку, а також додаткові бібліотеки для обробки мультимодальних даних.

2. Компоненти та їх взаємодія

2.1. Центральний агент (Orchestrator)

Центральний агент є ключовим елементом системи, що виконує наступні завдання:

- **Прийом запитів** від користувача та їх розбиття на підзадачі.
- **Оркестрація робочих процесів** через розподіл завдань між відповідними агентами.

- **Моніторинг виконання завдань** та адаптація стратегії управління залежно від проміжних результатів.
- **Інтеграція з мультимодальною RAG-системою** для пошуку та перевірки знань.

Для реалізації центрального агента використовується **LangChain Agents**, який дозволяє створювати динамічні ланцюжки взаємодії агентів та адаптувати їхню поведінку на основі результатів попередніх операцій.

2.2. Мультимодальна RAG-система

Retrieval-Augmented Generation (RAG) використовується для комбінування генеративних моделей (GPT-4, LLaMA 2) з потужною пошуковою інфраструктурою. Вона включає:

- **ChromaDB** – високопродуктивну векторну базу даних, яка дозволяє швидко знаходити релевантні документи.
- **Retriever** – механізм для обробки запитів та семантичного пошуку в документах.
- **Generator** – система генерації текстових відповідей на основі витягнутих релевантних даних.
- **Фільтрація та ранжування результатів** – використання технологій TF-IDF, BM25 та глибоких нейромережевих моделей.

2.3. Виконавчі агенти (Task Agents)

Виконавчі агенти мають спеціалізовані ролі:

- **Агенти аналізу даних** – використовують методи машинного навчання (ML) для обробки запитів.
- **Генеративні агенти** – LLM, які створюють текстові відповіді на основі знайдених матеріалів.
- **Агенти оптимізації** – відповідають за побудову моделей прийняття рішень для складних завдань.

Агенти реалізуються через **CrewAI**, що забезпечує ефективну комунікацію між ними та їх динамічне керування.

2.4. Агенти перевірки (Validation Agents)

Агенти перевірки відповідають за:

- **Контроль якості результатів** – перевірка відповідності вихідному запиту.
- **Виявлення помилок** – оцінка узгодженості отриманих відповідей.
- **Автоматичне тестування** – перевірка працездатності згенерованого коду.

2.5. Інтерфейсний агент (Interface Agent)

Він забезпечує:

- **Прозору комунікацію з користувачем**.
- **Форматування відповідей** у зручний для розуміння вигляд.
- **Динамічне уточнення запитів**.

3. Потік виконання запиту

1. **Користувач вводить запит** через Інтерфейсний агент.
2. **Центральний агент (Orchestrator)** розбиває його на підзадачі та розподіляє між відповідними агентами.
3. **Мультимодальна RAG-система** виконує пошук у ChromaDB та надає релевантну інформацію.
4. **Виконавчі агенти** аналізують дані та формують відповідь.
5. **Агенти перевірки** здійснюють валідацію та коригування отриманих результатів.
6. **Центральний агент агрегує результати** та передає їх користувачеві.

4. Використані технології

- **LangChain** – динамічне керування агентами.
- **CrewAI** – організація багатоагентних процесів.
- **ChromaDB** – семантичний пошук у векторній БД.
- **GPT-4 / LLaMA 2** – генерація текстових відповідей.
- **FastAPI** – розгортання API для взаємодії із зовнішніми сервісами.

Запропонована архітектура забезпечує **гнучкість, адаптивність та ефективність** в обробці складних запитів. Використання багатоагентного підходу з інтеграцією RAG-системи дозволяє динамічно адаптувати систему до змінних умов. Подальші дослідження можуть зосередитися на **оптимізації взаємодії агентів** та розширенні мультимодальних можливостей.

Порівняння AI-агентної системи з іншими AI-підходами

У цьому розділі розглянемо, як запропонована AI-агентна система порівнюється з іншими підходами в штучному інтелекті, такими як традиційні великі мовні моделі (LLM), класичні системи інформаційного пошуку (IR), багатокомпонентні нейромережеві системи та традиційні чат-боти. Ми також розглянемо сильні та слабкі сторони кожного підходу та обґрунтуємо вибір запропонованої архітектури.

2. Традиційні великі мовні моделі (LLM)

2.1. Особливості LLM

Великі мовні моделі, такі як GPT-4, LLaMA 2, PaLM 2, є потужними генераторами тексту, які використовують глибоке навчання для моделювання мовних закономірностей. Основні характеристики:

- **Генеративність**: створюють текст на основі вхідного запиту.
- **Контекстна обізнаність**: розуміють попередні частини тексту та будують узгоджені відповіді.
- **Обмежене використання зовнішніх знань**: не мають прямого доступу до актуальних чи специфічних знань, якщо вони не були включені у навчальні дані.
- **Відсутність можливості перевірки фактів**: можуть генерувати правдоподібний, але некоректний контент (галюцинації).

2.2. Порівняння з агентною системою

Характеристика	Традиційні LLM	AI-агентна система
Доступ до зовнішніх знань	❌ Обмежений	✅ Використовує RAG та ChromaDB
Генерація відповідей	✅ Потужна	✅ Потужна + контроль результатів
Адаптивність	❌ Фіксована після навчання	✅ Динамічна завдяки агентній архітектурі
Управління складними процесами	❌ Відсутнє	✅ Делегування завдань між агентами
Перевірка фактів	❌ Відсутня	✅ Валідація через спеціалізованих агентів

3. Класичні системи інформаційного пошуку (IR)

3.1. Особливості IR-систем

Системи інформаційного пошуку (Google Search, Elasticsearch) працюють на основі індексації та ранжування документів за запитами користувачів. Основні характеристики:

- **Працюють на основі ключових слів** та використовують статистичні моделі (TF-IDF, BM25).
- **Забезпечують швидкий пошук великого обсягу інформації**.
- **Не виконують генерацію відповідей** – лише повертають релевантні документи.
- **Потребують ручної обробки результатів користувачем**.

3.2. Порівняння з агентною системою

Характеристика	Класичні IR-системи	AI-агентна система
Релевантність відповідей	❌ Вимагає ручної обробки	✅ Інтегрує пошук із генерацією
Використання нейромереж	❌ Мінімальне	✅ Глибока інтеграція
Контекстуальне розуміння	❌ Відсутнє	✅ Використовує LLM
Можливість уточнення запитів	❌ Лімітоване	✅ Інтерактивна взаємодія

4. Багатокомпонентні нейромережеві системи

4.1. Особливості багатокомпонентних AI-систем

Ці системи поєднують кілька підходів, зокрема нейронні мережі для аналізу тексту, зображень і звуку. Вони використовуються в голосових асистентах (Siri, Alexa) та аналітичних платформах.

Основні характеристики:

- **Комбінують кілька моделей (NLP, CV, ASR)** для різних типів даних.
- **Мають складну архітектуру з модулями для різних завдань**.
- **Вимагають значних обчислювальних ресурсів**.

4.2. Порівняння з агентною системою

Характеристика	Багатокомпонентні AI-системи	AI-агентна система
Гнучкість	❌ Фіксовані під конкретні завдання	✅ Модульні, легко адаптуються
Розширюваність	❌ Важко інтегрувати нові компоненти	✅ Легка інтеграція нових агентів
Автономність	❌ Потребує централізованого управління	✅ Децентралізоване прийняття рішень

5. Традиційні чат-боти

5.1. Особливості чат-ботів

Базові чат-боти (rule-based, intent-based) працюють за заздалегідь визначеними сценаріями та не володіють глибоким розумінням контексту.

Основні характеристики:

- **Мають обмежений набір правил** та не можуть імпровізувати.
- **Не здатні працювати з довгостроковою пам’яттю**.
- **Мають високу точність для простих запитів, але низьку для складних сценаріїв**.

5.2. Порівняння з агентною системою

Характеристика	Традиційні чат-боти	AI-агентна система
Гнучкість відповідей	❌ Фіксовані сценарії	✅ Динамічне формування
Контекстуальність	❌ Обмежена пам’ять	✅ Використовує LLM + векторну базу

Адаптація до нових задач	❌ Потребує додаткового навчання	✅ Гнучка система з модульними агентами

Запропонована AI-агентна система поєднує сильні сторони різних AI-підходів, усуваючи їхні основні недоліки. Вона перевершує традиційні LLM у точності відповідей через використання RAG, перевищує IR-системи у розумінні запитів, гнучкіша за багатокомпонентні AI-системи та значно перевершує класичні чат-боти у динамічності та адаптації. Це робить її перспективним рішенням для складних інформаційних завдань.