

# ПОРІВНЯЛЬНИЙ АНАЛІЗ PYTHON ФРЕЙМВОРКІВ ДЛЯ СТВОРЕННЯ АГЕНТІВ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ

Вінницький національний технічний університет

## Анотація

*В роботі проведено порівняльний аналіз Python фреймворків для створення ШІ агентів. Оцінено LangGraph, LlamaIndex, CrewAI, Semantic Kernel, AutoGen, OpenAI Swarm, досліджено їх сильні та слабкі сторони, можливості.*

**Ключові слова:** Штучний інтелект, великі мовні моделі (LLM), ШІ агенти, LangGraph, LlamaIndex, CrewAI, Semantic Kernel, AutoGen, OpenAI Swarm.

## Abstract

*In the paper, a comparative analysis of Python frameworks for creating AI agents is conducted. LangGraph, LlamaIndex, CrewAI, Semantic Kernel, AutoGen, OpenAI Swarm are evaluated, their strengths and weaknesses, and capabilities are investigated.*

**Keywords:** Artificial intelligence, large language models (LLM), LangGraph, LlamaIndex, CrewAI, Semantic Kernel, AutoGen, OpenAI Swarm.

## Вступ

В останні роки в галузі штучного інтелекту спостерігається значне зростання розробки та застосування агентів ШІ для різних завдань, таких як пошук інформації, обробка даних і прийняття складних рішень. Поява кількох фреймворків Python, розроблених спеціально для створення агентів ШІ, надала розробникам різноманітні інструменти для створення складних систем на основі агентів. Однак вибір відповідного фреймворку може суттєво вплинути на продуктивність, масштабованість і можливості кінцевих агентів ШІ.

Ця теза робота спрямована на проведення комплексного порівняльного аналізу шести основних фреймворків Python для створення агентів ШІ: LangGraph, LlamaIndex, CrewAI, Semantic Kernel, AutoGen і OpenAI Swarm. Дослідження зосереджено на оцінці цих фреймворків з точки зору їх основної спрямованості, статусу відкритого вихідного коду, підтримки мови програмування, сильних і слабких сторін, можливостей керування пам'яттю, інтеграції інструментів, підтримки втручання під час виконання та найкращих варіантів використання.

## Результати дослідження

Дослідження оцінювало такі показники відстані:

1. LangGraph: цей фреймворк наголошує на оркестровці стану та графах станів. Як рішення з відкритим вихідним кодом, що підтримує Python і JavaScript, LangGraph вирізняється можливістю компонування, безперервністю та складним керуванням станом. Він містить графіки зі збереженням стану з постійною пам'яттю та добре інтегрується з інструментами LangChain. Однак його складність і обмежена підтримка певних програм можуть створити проблеми для нових користувачів.

2. LlamaIndex: орієнтований на індексування та пошук даних, LlamaIndex є фреймворком з відкритим кодом, який підтримує Python і JavaScript. Його сильні сторони полягають у ефективному прийомі та обробці даних, а також потужних можливостях пошуку документів. Фреймворк пропонує різні методи індексування та надає інструменти для підключення даних до механізмів LLM. Його основні обмеження включають обмежений вибір контексту та високі вимоги до ресурсів для великих наборів даних.

3. CrewAI: ця структура зосереджена на взаємодії агентів на основі ролей і доступна як рішення з відкритим кодом, що підтримує Python. CrewAI пропонує зручну платформу з хорошими інтерфейсами абстракції, що робить її доступною для розробників. Однак він має обмежені можливості налаштування та обмеження щодо інтеграції рольової пам'яті, а також високі вимоги до пам'яті для складних робочих процесів.

4. Semantic Kernel: зосереджується на інтеграції навичок із плануванням природної мови, є фреймворком із відкритим кодом, що підтримує C#, Python і Java. Він демонструє модульну архітектуру, готову до корпоративного використання, і має потужні можливості зосередженості, пам'яті та планування. Його слабкість полягає в складній установці, потенційно крутій кривій навчання та субсимволічних (несемантичних) обмеженнях.

5. AutoGen: AutoGen базується на багатоагентній архітектурі та доступний як фреймворк із відкритим кодом, що підтримує Python. Він пропонує настроювані агенти, розширену оркестровку та комплексні функції для виробничих середовищ. Його основні слабкі сторони включають складність процесів і неоптимальні можливості втручання під час виконання.

6. OpenAI Swarm: спеціалізується на легкій координації агентів, OpenAI Swarm є фреймворком з відкритим кодом, що підтримує Python. Його переваги включають легкі настроювані компоненти та експериментальні, невизначені технічні межі. Однак він не має можливостей керування пам'яттю та має обмежену підтримку втручання людини.

| Feature                   | LangGraph                                                                            | LlamaIndex                                                                                | CrewAI                                                                          | Semantic Kernel                                                                   | AutoGen                                                                                         | OpenAI Swarm                                                                                    |
|---------------------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Primary focus             | Stateful orchestration and cyclic graphs                                             | Data indexing and retrieval                                                               | Role-based agent collaboration                                                  | Integrating LLMs with conventional programming languages                          | Multi-agent conversations                                                                       | Lightweight agent coordination                                                                  |
| Open source               | Yes                                                                                  | Yes                                                                                       | Yes                                                                             | Yes                                                                               | Yes                                                                                             | Yes                                                                                             |
| Programming languages     | Python, JavaScript                                                                   | Python, JavaScript                                                                        | Python                                                                          | C#, Python, Java                                                                  | Python                                                                                          | Python                                                                                          |
| Strengths                 | Controllability, continuity, and LangChain interoperability                          | Efficient data ingestion and querying, flexibility                                        | User-friendly platform, role-based architecture                                 | Enterprise-ready, modular, future-proof                                           | Customizable agents, asynchronous messaging                                                     | Lightweight, customizable, open-source                                                          |
| Weaknesses                | Complexity, limited support for distributed systems, potential for supervisor issues | Limited context retention, focus on search and retrieval, file size and processing limits | Limited orchestration strategies, rate limits, potential for incomplete outputs | Limited focus, memory limitations, challenges with reusing existing functions     | Complexity of prompts, subpar conversational aspect, limited capabilities in specific scenarios | Experimental, stateless, limited novelty, potential for divergence                              |
| Memory management         | Stateful graphs with persistent data                                                 | Various indexing techniques for efficient memory management                               | Built-in memory management with vector store and SQLite database                | VolatileMemory (short-term) and Qdrant support                                    | Not applicable                                                                                  | Not applicable                                                                                  |
| Tool integration          | Integrates with LangChain tools and models                                           | Provides tools for data connectors, indexes, and engines                                  | Offers a range of built-in tools and allows custom tool creation                | Supports plugins and connectors for integrating external services                 | Allows tool use through multi-agent conversations                                               | Supports direct function calls as tools                                                         |
| Human-in-the-loop support | Allows human intervention in workflows                                               | Supports human-in-the-loop through agents and workflows                                   | Enables human participation in agent tasks and collaboration                    | Supports human-in-the-loop through interactive agents and workflows               | Supports human participation in multi-agent conversations                                       | Not applicable (human-as-a-tool)                                                                |
| Best use cases            | Complex workflows requiring fine-grained control, state management, and adaptability | Data-centric applications, knowledge retrieval, and question-answering                    | Collaborative agent systems with specialized roles and automated workflows      | Enterprise-grade applications, integrating AI with existing systems and codebases | Next-generation LLM applications with multi-agent conversations and autonomous task solving     | Experimentation with multi-agent coordination, lightweight prototypes, and educational purposes |

Рисунок 1 – Порівняння фреймворків для створення ШІ агентів

Можливості керування пам'яттю суттєво відрізняються в різних структурах. LangGraph пропонує графіки зі збереженням стану з постійною пам'яттю, тоді як LlamaIndex пропонує різні методи індексування для обробки даних. CrewAI реалізує рольову пам'ять із пам'яттю для конкретного агента. Semantic Kernel містить підсимволічні моделі пам'яті. AutoGen не наголошує на керуванні пам'яттю як на основній функції, а OpenAI Swarm вважає керування пам'яттю незастосовним до свого дизайну.

Усі фреймворки забезпечують певну форму інтеграції інструментів, але з різними підходами. LangGraph інтегрується з інструментами LangChain, LlamaIndex пропонує підключення даних до механізмів LLM, а CrewAI надає низку вбудованих інструментів для створення агентів. Semantic Kernel підтримує інтеграцію плагінів і навичок, AutoGen дозволяє використовувати інструменти через вибрані служби, а OpenAI Swarm підтримує прямі виклики функцій.

Можливість людського втручання під час виконання відрізняється між фреймворками. LangGraph і LlamaIndex дозволяють людині втручатися в робочі процеси, CrewAI забезпечує участь людини через агентів на інтерфейсах, а Semantic Kernel підтримує участь людини в розмовах між кількома агентами. AutoGen надає інструменти для процесів, що здійснюються людиною в циклі, тоді як OpenAI Swarm має обмежену підтримку людського втручання.

## Висновок

Порівняльний аналіз підкреслює важливість вибору відповідної структури агента AI на основі конкретних вимог програми. LangGraph пропонує надійне керування станом для складних робочих процесів, тоді як LlamaIndex забезпечує ефективні можливості пошуку даних. CrewAI сприяє співпраці на основі ролей, Semantic Kernel чудово підходить для корпоративної інтеграції, AutoGen пропонує комплексну багатоагентну архітектуру, а OpenAI Swarm забезпечує легкі механізми координації. Для додатків, які вимагають складного керування станом і оркестровки робочого процесу, LangGraph є сильним вибором. Проекти, зосереджені на обробці та пошуку даних, виграють від LlamaIndex. Для спільних багатоагентних систем із визначеними ролями CrewAI надає доступне рішення. Корпоративні програми з модульними вимогами до навичок добре узгоджуються з семантичним ядром. Складна взаємодія агентів і розширена автоматизація знаходять підтримку в AutoGen, тоді як експериментальні програми з акцентом на легких протоколах можуть віддати перевагу OpenAI Swarm.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. AI agent frameworks: Choosing the right foundation for your business [Електронний ресурс] – Режим доступу: <https://www.ibm.com/think/insights/top-ai-agent-frameworks> (дата звернення: 18.03.2025).
2. AI agents — what they are, and how they'll change the way we work [Електронний ресурс] – Режим доступу: <https://news.microsoft.com/source/features/ai/ai-agents-what-they-are-and-how-theyll-change-the-way-we-work/> (дата звернення: 18.03.2025).
3. AI Agents: Frameworks (Part-3) [Електронний ресурс] – Режим доступу: [https://medium.com/@vipra\\_singh/ai-agents-frameworks-part-3-ca8ce33c2f35#dac6](https://medium.com/@vipra_singh/ai-agents-frameworks-part-3-ca8ce33c2f35#dac6) (дата звернення: 18.03.2025).
4. AI Agent Frameworks: A Detailed Comparison [Електронний ресурс] – Режим доступу: <https://www.turing.com/resources/ai-agent-frameworks> (дата звернення: 18.03.2025).

**Герасімов Євген Євгенович** – студент групи ІАКІТР-24м, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: [yevhen.gerasimov@gmail.com](mailto:yevhen.gerasimov@gmail.com).

**Богач Ілона Віталіївна** – к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: [ilona.bogach@gmail.com](mailto:ilona.bogach@gmail.com).

**Herasimov Yevhen Yevhenovych** – student of ІАКІТР-24М group, Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [yevhen.gerasimov@gmail.com](mailto:yevhen.gerasimov@gmail.com).

**Bogach Ilona Vitaliivna** – Associate Professor of Automation and Intelligent Information Technologies Department, Faculty of Computer Systems and Automatics, Vinnytsia National Technical University, Vinnytsia, e-mail: [ilona.bogach@gmail.com](mailto:ilona.bogach@gmail.com).