UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA

INGENIERÍA INFORMÁTICA

MENCIÓN EN COMPUTACIÓN

PETICIÓN DE TEMA TRABAJO DE FIN DE GRADO

Metacrew, an organization agents for the setting up agent organizations

Metacrew, una organización de agentes para la confección de organizaciones de agentes

Author: Jesús Núñez de Arenas Llamas

Co-Director: Carlos García Martínez

Co-Director: Aurora Ramírez Quesada

February 9, 2025

Abstract

Los sistemas de agentes basados en modelos de lenguaje de gran tamaño(LLM) son una forma emergente de uso de la inteligencia artificial(IA) que permite abordar problemas complejos de forma más eficiente. Estos sistemas se basan en la comunicación de varios agentes especializados y los recursos permitidos para tomar decisiones de forma autónoma. El uso de varios agentes permite una mayor variabilidad en la resolución de los problemas en comparación a los enfoques más tradicionales y comunes del uso de la IA.

En este Trabajo de Fin de Grado vamos a abordar la automatización de la creación de cadenas de agentes capaces de dar solución a problemas de distintos ámbitos. Para ello, usaremos las librerías de CrewAI [9] y LangChain [2] principalmente, con las que usaremos sus herramientas para construir el sistema.

El sistema lo probaremos comparando la repuesta del grupo creado con diferentes modelos de LLM. Para evaluarlos recurriremos al sistema automático de FairEval [13], que tiene en cuenta el sesgo de los LLM para evaluar su calidad, y una evaluación humana, para una valoración más subjetiva de la utilidad de estas respuestas. Con este enfoque tenemos un sistema riguroso de evaluación del sistema desarrollado.

Este trabajo es una obra de carácter original realizada por Jesús Núñez de Arenas Llamas bajo la tutela de Carlos García Martínez con tipología de investigación aplicada, o de desarrollo, sobre algún tema en concreto de la tecnología específica de cada titulación, incluyendo los de naturaleza profesional recogidos en la Orden CIN 351/2009, de 9 de febrero de 2009. Cualquier uso de fuentes externas está y será citado de la manera correspondiente.

The systems of agents based on large language models(LLM) are an emerging way of use of the artificial intelligence(AI) that allow to approach complex problems in a more efficient way. These systems are based on the communication of various specialized agents y the allowed resources to make autonomous decisions. The use of some agents allows a wider variety in the resolution of problems compared with more traditional and common approaches of the use of AI.

In this Final Degree Project we are going to automatize the creation of groups of agents that are able to give answers to problems of different fields. To do it, we are going to use mainly the libraries of CrewAI [9] and LangChain [2], whose tools we will use to build the system.

It will be tested comparing the answers given by the created crew against the answer of different LLMs. To evaluate it we will use the automated system of FairEval [13], that takes into account the bias of the LLMs to evaluate its quality, and a human evaluation, for a more subjective evaluation of the utility of the answers. This approach allow us to have a rigorous approach of the developed system.

This work is an original project made by Jesús Núñez de Arenas Llamas under the supervision of Carlos García Martínez with the typology of applied research, or development research, on a specific topic in the specific technology of each degree specific to each degree, including those of a professional nature included in Order CIN 351/2009, of 9 February 2009. Any type of use of external sources is and will be quoted appropriately.

Contents

1. Introduction

Nowadays, the use of Artificial Intelligence(AI) is extremely widespread through different fields of study. Moreover, it is used as a tool to hasten the process of tedious tasks or to even automatize them.

The AIs are so potent that companies are starting to rely on them to make analyses about the market, to accept or deny automatically curriculums vitae based on a series of criteria... We can take a look at some articles that explore the impact of AI in the ambit of candidate selection, being one of them: Exploring the impact of AI on candidate selection: a two-phase methodological approach with CRITIC-WASPAS [4]. However, their implementation and maintenance in everyday life in the office are privative and normally only the big companies rely on this software. The commonly used models are LLMs, like ChatGPT [10], which are able to communicate through natural language with the user to explain or solve the user's request.

As the population continues to use AI, there were bound to appear tools that try to simplify its use for automation of the processes. The default way is to use a single agent to automate a process. It may seem an appropriate solution; however, the use of a single agent has its limitations. The quantity of information that the agent can process is limited and, due to being an only agent, it will have to generalize the solution to the problems. We can read the blog of 'Single Agent Systems Versus Multi-Agent AI Teams' [1] to have a better understanding of the differences between their use. An example of the organization of agents is ChatDev [11], that works as a virtual software organization which makes use of different intelligent agents to take care of different roles in an organization. In Figure 1.1 taken from the article, we can see a possibility of execution where the first agents receive a list of requirements for a software that are decomposed into smaller requirements and passed to the next agents. In the latest phases, the code is written and tested by specialized agents in computer science that comply with the requirements.

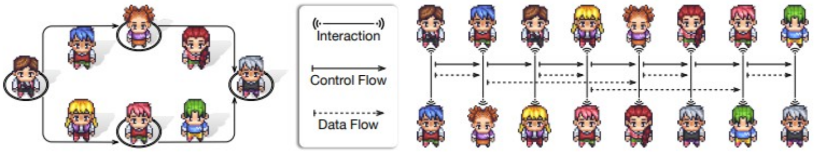

Figure 1.1: Topology of an organization of agents with dual agent interaction

To avoid the generalization issues and limitations of information it can receive, we can create a network of interconnected agents that can communicate with each other. To do it, there are vanguardist tools that we can use, CrewAI [9] and LangChain [2], whose rising popularity has made companies start to use them. These tools, instead of using a single agent of AI, like ChatGPT [10], use multiple agents that are given a background and a task to complete. The configuration of the crew also passes by how it is organized internally, which agents can communicate with which and which agent will synthesize the information gathered. Also, its structure could change depending on the task field. Hypothetically, after the crew is run, they give a solution that is more complex than a single agent could accomplish.

The creation of an agent organization can be a challenging task and difficult to approach if we need to repeat it for different fields. This has led us to our investigation, where we are going to use these tools in order to create a crew of agents that is able to create new crews to automatize different processes in different fields. We will study up to what point the crew benefits from the use of different agents of AI to give solutions to complex problems.

2. Objectives

The goal of this project is to generate and study a group of agents(crew) to automatize the creation of different crews to solve tasks of different fields accurately. To achieve it, we have created the next set of objectives that need to be completed.

- Definition of tasks: The system needs to search for information about the tasks that a real company decompose from a main task.
- Definition of agents: The system has to search for the type of professionals that take care of a specific task in the main task.
- Definition of a communication workflow: The system needs to search for information on how an organization is structured when approaching certain tasks.
- Design and implementation of the system. We need to create an organization of agents that can comply with the previous points.
- Implementation of series of tests like the ones used in EvoAgent [14] and AutoAgents [3].
- Execution and study of the solutions given by the implemented system over the tests.

3. Antecedents

As the field of study is so new and specific due to tools being developed not long ago, there is little study about the topic of the project. There are some studies about the agents of AI, which we are going to use as antecedents of this study.

The two most important found articles are EvoAgent [14] and AutoAgents [3]. They give us a head start on how there are different ways to create a group of agents with different approaches. EvoAgent [14] creates multi-agents by using evolutionary algorithms to create various agents with a wide range of skills. The agents are evolved to have a range of refined skills, having a population with different skills as the objective. AutoAgents [3] creates agents in two phases: a drafting phase, where a group of three agents creates a plan and the agents to use, and an execution phase, where the agents created are executed to complete the original task. Our project is differentiated by the inclusion of custom tools that the agents created can use, and the use of a framework library like CrewAI [9] with a high level of abstraction allows our work to be understood more easily.

Other project that piqued our interest is Auto-GPT [12], even if it works with a sole agent, its broad fields of use and the characteristic of creating its own subtasks to reach the target imposed by the user are awe-inspiring. The ability to create small tasks to defuse a major problem is one of the functionalities needed to complete our project.

These projects give us a good ground start to make the project, on one hand, how to use a chain of agents to achieve a goal and, on the other hand, how to break down the problem to make it more pleasant to work with. With this knowledge, we should be able to start to work with our tools and adapt their knowledge.

4. Project phases

4.1. Prior knowledge acquisition

Before getting hands-on with the code, we need to obtain the relevant information that would make the project viable. So it is necessary to have a phase where we obtain knowledge to get started, that we will later broaden to add more complexity to the project.

We will familiarize ourselves with the Integrated Development Environment (IDE) chosen, in this case, Visual Studio Code [7], and with the libraries used. Once we are able to make good use of these tools, we can start to study the problem at hand and how to approach it. This will encompass the behavior of each agent, their exchange of information, and the solution they reach. We will familiarize ourselves with CrewAI's [9] framework and we will study how to make custom tools using the library of LangChain [2].

4.2. Analysis and design

Once we acquire knowledge of its nature, we can start to design different workflows, backgrounds, tasks, and the number of agents to optimize their outputs. Due to the complexity of the problem, multiple drafts about the workflow will need to be made and discarded before reaching a middle point between the complexity of the crew and satisfaction with the solutions.

Firstly, we will try different combinations of workflows with basic agents. This will be an iterative process where we will make use of linear, parallel and hierarchical workflows and their combinations to reach a state where the computational cost is not too high and the solutions that are given are good.

After the workflow is chosen, we will work on preparing backgrounds and tasks that are associated to try and optimize the output of each individual agent. To create a crew whose job is to create crews for different problems, we will need agents that are able to identify different tasks for each problem and generate a background for each agent to solve that task.

4.3. Development

During the development phase, we will implement the models defined in analysis and design. When we have a working crew that is able to create new crews effectively, we will optimize it to try to reduce the time it takes or to make interactions between the agents more agile. This will give us an edge to reduce the time we take when starting testing and analyzing the results.

4.4. Testing and analysis of results

Once the crew is developed, we will choose a variety of problems to solve. Our crew will create another crew with its own workflow, the appropriate agents whose backgrounds and tasks are defined to solve a concrete problem. This newly created crew will try to solve the problem, storing the results of their interactions, the outputs of all the agents, and the solution given to the problem.

Later on, we will compare the solutions given by the created crews and the existing models such as ChatGPT 3.5, ChatGPT 4, Microsoft Copilot [8], Gemini [6]... We will use two separate score systems to measure the performance of the crews and the models. The first form of evaluation will be using FairEval [13] software which is going to give us a mark between 0 and 10. The second form will be using volunteers to rate the accuracy, reliability, helpfulness, and level of detail of the answers.

4.5. Documentation

The documentation of the project will start at the same time as the knowledge acquisition starts. We will document our findings and thoughts on each phase and explain how we reach each conclusion in order to make the study more understandable. The final code will be given to explain some of the decisions and a quick guide on how to use it and some expected results.

5. Resources

5.1. Human Resources

Author:

- Jesús Núñez de Arenas Llamas, student of Ingeniería Informática en la mención de Computación

Under the direction of:

- Carlos García Martínez, titular professor of the department of Informática y Análisis Numérico
- Aurora Ramírez Quesada, assistant professor doctor of the department of Informática y Análisis Numérico

5.2. Hardware Resources

The testing will be run on the personal computer of the author, with the following characteristics:

- CPU: 11th Gen Intel® Core™ i7-11370H @ 3.30GHz
- RAM: 16 GB DDR4-3200
- GPU: RTX 3060 6GB
- Storage: 1 TB + 524 GB

5.3. Software Resources

The development environment used will be:

- Python 3.10.11

- Visual Studio Code [7]

The different libraries used in the environment will be:

- CrewAI 0.67.1

- CrewAI-tools 0.12.1

- LangChain 0.2.16

The next tool will be used to manage the packages and their dependencies:

- Poetry [5] 1.8.3

Lastly, to document the project, it will be used:

- Overleaf, LaTex editor

6. Temporal distribution

According to the regulations of the Polytechnic School of Córdoba, the Final Degree Project should cover a total of 300 hours, with a valuation of 12 credits ECTS, which is 25 hours per credit ECTS. Taking into account the hourly limitation, a probable itinerary could be the one shown in Table 6.1.

Phase	Hours
Prior knowledge acquisition	60
Analysis and design	40
Development	90
Testing and analysis of results	70
Documentation	40
Total	**300**

Table 6.1: Hourly distribution

Dividing the work hours into 12 different weeks, we obtain that each week we will have to work an average on 25 hours per week. In Figure 6.1, we can observe a possible distribution of the phases in the 12 weeks.

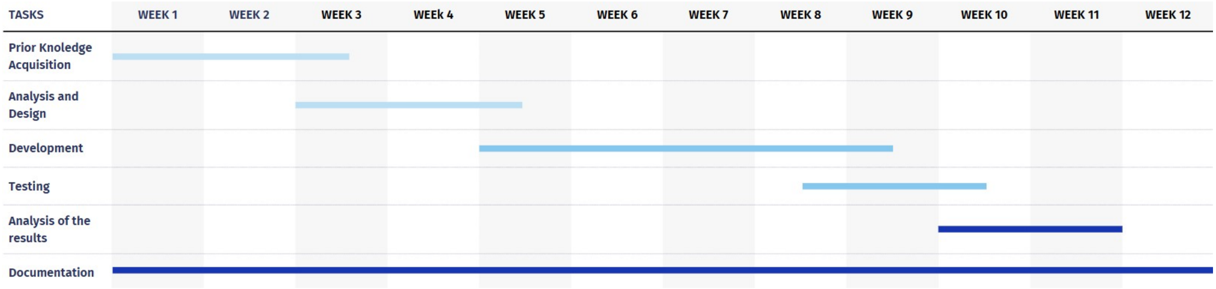

Figure 6.1: Gantt chart of the expected time distribution

Bibliography

[1] Dawid Bednarski. Single agent systems versus multi-agent ai teams. *Taskade*, 2024. URL: `https://www.taskade.com/blog/single-agent-systems-versus-multi-agent-ai-teams/`.

[2] Harrison Chase. LangChain, October 2022. URL: `https://github.com/langchain-ai/langchain`.

[3] Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. *ArXiv*, 2024. URL: `https://arxiv.org/html/2309.17288v3#bib`, `arXiv:2309.17288v3`.

[4] Luis Hernan Contreras Pinochet, Nathalia Sertorio Lopes, Luciana Massaro Onusic, Marcos dos Santos, Vanessa Itacaramby Pardim, and Andresa Silva Neto Francischini. Exploring the impact of ai on candidate selection: A two-phase methodological approach with critic-waspas. *Procedia Computer Science*, 242:920–927, 2024. 11th International Conference on Information Technology and Quantitative Management (ITQM 2024). URL: `https://www.sciencedirect.com/science/article/pii/S1877050924019938`, `doi:10.1016/j.procs.2024.08.274`.

[5] Sébastien Eustace and The Poetry contributors. Poetry: Python packaging and dependency management made easy. URL: `https://github.com/python-poetry/poetry`.

[6] Google. Google gemini, 2023. URL: `https://gemini.google.com/`.

[7] Microsoft. Visual studio code, 2015. URL: `https://code.visualstudio.com/`.

[8] Microsoft. Microsft copilot, 2023. URL: `https://copilot.microsoft.com/`.

[9] João Moura. Crewai: Multi-agent collaboration framework, 2024. URL: `https://github.com/joaomdmoura/crewai`.

[10] OpenAI. Chatgpt, 2023. URL: `https://chat.openai.com/chat`.

[11] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023. URL: `https://arxiv.org/abs/2307.07924`.

[12] Significant Gravitas. AutoGPT. URL: `https://github.com/Significant-Gravitas/AutoGPT`.

[13] Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *ArXiv*, abs/2305.17926, 2023.

[14] Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Dongsheng Li, and Deqing Yang. Evoagent: Towards automatic multi-agent generation via evolutionary algorithms. *ArXiv*, 2024. URL: `https://arxiv.org/pdf/2406.14228`, `arXiv:2406.14228`.