Universidad
Rey Juan Carlos

Development and implementation of an ECM:
A generalist approach to user-AI interaction.

School: Escuela de Ingenieria de Fuenlabrada.
Degree: Robotics Software Engineering.

Author: Sebastián Mayorquín Posada.
Tutor: Julio Vega.

September 19, 2024

License

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Acknowledgements

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Abstract

[En proceso] Los ultimos lanzamientos de Modelos Grandes de Lenguaje (LLM por sus siglas en ingles), da paso a nuevas implementaciones de Inteligencias Artificiales Generalistas (AGI). Mas allá the optimizar modelos de machine learning, el desarrollo de AGI's requiere de aplicaciones cognitivas que habiliten a las LLM a operar de forma efectiva en aplicaciones del mundo real.

Esta memoria intruduce la Máquina de Congición-Ejecución(MCE) como un marco teórico que descompone el diseño de un AGI como un problema the aproximacion definido por 3 variables clave: El *espacio de ejecución*, textitespacio de cognición y *espacio de ejecución*. Como aproximación a este marco teórico se propone *[SystemName]* como un entorno de desarrollo para probar y diseñar AGI's implementando diferentes aproximaciones en multiples capas. Finalmente se implementan multiples técnicas incluyendo *Prompt Engineering*, codigo *Exelent*, o *Auto-Entrenamiento* con el fin de construir un AGI que soporte la interación usuario-IA en un entorno no controlado.

#TODO: Incluir en el abstract mencion a la implementacion en RaspBerry, y resultados

Acronyms

- **AI**: Artificial Intelligence
- **AP**: Agent Protocol
- A_1: Execution Layer Algorithm
- A_2: Cognition Layer Algorithm
- **LLM**: Large Language Model
- **ECM**: Execution-Cognition Machine
- **AGI**: Artificial General Intelligence
- **RAG**: Retrieval Augmented Generation
- **PMPA**: Profile-Memory-Plan-Action

Contents

1. Introducción

La palabra *"inteligencia artificial"* ha sido seleccionada por el diccionario Collins como palabra del año en el 2023. Miles de empresas ahora estan integrando tecnologías con *"IA"*. Del mismo modo, multiples medios de comunicación informan de los posibles problemas y riesgos de estas tecnologías en caso de no ser implementadas de forma apropiada. A pesar de su creciente popularidad en los ultimos años, el concepto de inteligencia artificial es dificil de definir y categorizar incluso dentro del sector cientifico.

Dado este contexto de creciente interés y debate, el presente capítulo abordará los fundamentos de la inteligencia artificial, definiendo sus características principales, su historia y las razones detrás de su surgimiento. Este marco conceptual es clave para que el lector pueda comprender el estado del arte, que será tratado en profundidad en los capítulos siguientes.

1.1 Inteligencia Artificial - Conceptos Básicos

Las dos palabras acuñadas en el concepto de Inteligencia Artificial hacen referencia a la simbiosis de dos campos de estudio totalmente diferentes. Es tanto así, que en distintos idiomas esta composición lingüistica se mantiene constante (véase la terminología en ingles: *"Artificial Intelligence"*).

En este sentido, la *inteligencia* ha sido estudiada historicamente por ramas como la psicología, filosofía o educación donde se converge en multiples definiciónes que aunque distintas, resultan "intuitivamente" faciles de relacionar: la inteligencia hara referencia a la capacidad para entender, comprender o resolver problemas, al conjunto de habilidades cognitivas que incluyen la autoconciencia, creatividad o razonamiento logico. Fuera del ámbito cientifico resulta sencillo distinguir aquellas entidades que demuestran inteligencia de aquellas que no. Así pues, aunque puede conformarse un debate en torno a las capacidades intelectuales de dos especies de similar origen (como lo podría ser un delfín y un tiburon); es posible afirmar con certeza que una unidad morfológica simple como lo es una celula eucariota, es menos inteligente que un humano promedio.

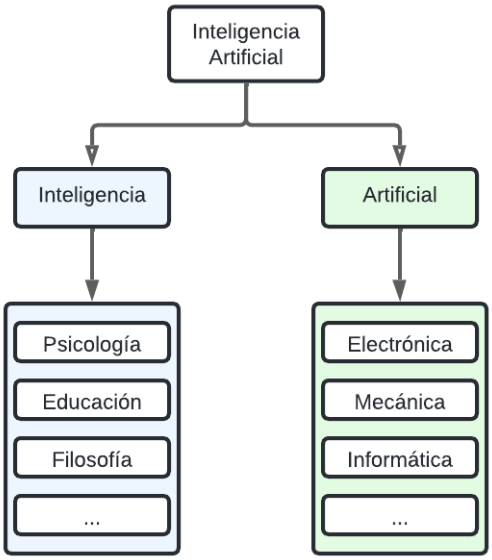

Figure 1.1: ”Inteligente” vs ”Artificial”. Elaboración Propia

De forma similar, todo aquello que es *artificial* hace referencia a aquello que no es natural, generalmente en relación con artefactos o productos creados con un proposito determinado. Mientras que un delfín se puede considerar producto de la naturaleza, un robot aspirador se considera artificial al ver que este es consecuencia de una composición de elementos electronicos diseñados y organizados por el ser humano con un fin concreto: limpiar el escenario en el que se encuentre.

Tras la Conferencia de Dartmouth organizada en 1956 por el matemático John McCarthy junto con otro investigadores como Marvin Misnky, se definió por primera vez el concepto de inteligencia artificial.

> *”El estudio procederá sobre la base de la conjetura de que cada aspecto del aprendizaje o cualquier otra característica de la inteligencia puede, en principio, ser descrito con tal precisión que se pueda crear una máquina capaz de simularlo.” McCarthy et al. (2006)*

1.2 Aplicaciones de la IA

1.3 Aprendizaje Automatico vs Inteligencia Artificial

2. State of Art

2.1 IA Suave y Fuerte

La Inteligencia Artificial (IA) ha revolucionado la forma de utilizar y desarrollar nuevas herramientas. Aunque este concepto no es nuevo, es importante diferenciar entre dos conceptos clave: inteligencia *fuerte* e inteligencia *suave*

Para definir la Inteligencia Artificial Searle (1980) definió dos tipos de IA. La IA *suave*[1] hace referencia a un conjunto de algoritmos o herramientas que aproximan una solucion a un problema determinado utilizando estrategias basadas en analisis y reacción a traves de múltiples estados. En contraste, la IA *fuerte* esta caracterizada por sus capacidades para entender de forma profunda, razonar y reaccionar a estados impredecibles, adaptandose y aprendiendo a lo largo del proceso de resolución del problema.

Desde el diseño de las *redes neuronales* en 1949, nuevas tecnicas para el desarrollo de IA fuerte han estado en auge. Aunque la IA fuerte muestra propiedades aparentemente utópicas, muchos de los subproblemas que encierra su diseño han fomentado la evolución de multiples sectores. Como ejemplo de esto, el campo del aprendizaje automatic se ha expandido creando nuevos campos de estudio como el aprendizaje profundo. Será este nuevo campo es el responsable de la creación de los Modelos Grandes de Lenguaje (LLMs), donde modelos como GPT-4 (Achiam et al. (2023)), LLAMA2 o (Touvron et al. (2023)), o GEMINI (Saeidnia (2023)) utilizan el entrenamiento de millones de iteraciones sobre el conocimiento humano para recrear propiedades emergentes como lo son el entendimiento del sentido comun, habilidades de razonamiento y resolución de problemas, generación de respuestas con contexto, etc. Estos modelos aun estan en etapa de desarrollo, donde el objetivo apunta a propiedades como la multimodalidad[2], reconocimiento de patrones y/o explicabilidad, acercando a estos modelos a la integración de agentes cognitivos completos.

#TODO: En el parrafo, menciono 3 ejemplos de LLMs del estado del arte: GPT4,

[1] Nótese que aunque parecidos, es importante no confundir IA suave con IA clásica.

[2] (Capacidad para trabajar con multiples tipos de entrada, como audios o imagenes.)

Llama2 y Gemini ¿Deberia explicar cada una por separado?

2.2 Agents and Cognition

By formatting the output of the models, we make LLMs able to use multiple *tools*. Tools are a set of functions or utilities that allow the LLM to interact with its environment. While an LLM can only generate an output from a given query, agents translate those outputs into actions that modify the world and optionally the agent's own configuration.

> *"Agents are only as good as the tools they have"* (LangChain, 2024).

The use of agents has been extended in recent years, where a remarkable implementation is the *Retrieval Augmented Generation (RAG)*. This architecture is supported over an agent connected to a tool that retrieves information from a database, empowering the LLM with field- or company-specific knowledge and improving the accuracy of the models in controlled environments.

The apparition of AI agents enhances the opportunity for further research in the field of cognition. *Cognition* is a branch of computer science that investigates the development of systems capable of replicating human intelligence properties such as reasoning, perception, memory, planning, and decision-making. Classical cognitivism has been studied since 1956; however, the interdependent relationship between multiple modules required for a cognitive architecture has posed a constraint for further research.

> *"We have circles within circles. The central difficulty for problem solving was ill-defined problems. In order to deal with them we turned to restructuring. In order to address this we turned to analogy and at its heart we find ill-defined problem solving as a crucial component. Analysis and formaIization have been seriously frustrated in this whole endeavour." Vervaeke (1999)*

Multiple paths have emerged from attempts to define some form of cognitive intelligence while avoiding a fully intelligent system. For example, problem-solving has been approached using classical AI (Russell and Norvig, 2016), STRIPS (Fikes and Nilsson, 1971), or PDDL (Aeronautiques et al., 1998). Although these

approaches have been successful in multiple applications, the advent of AI agents opens the door to revisiting the original study of cognition.

LLMs *break the loop* on the interdependency restriction by approaching reasoning as a computable optimization problem. AI agents enable these LLMs to be integrated into a cognitive system, partially solving some architectural constraints. In line with these architectures, a *cognitive AI agent* is defined as an agent from which cognitive capabilities emerge, and which is able to interact, learn, and modify its behavior or environment.

2.3 GPS vs AGI

A *General Problem Solver (GPS)* has been studied in the field of cognition since 1959 with Newell et al. (1959) who introduced the GPS as an hypothetical algorithm or set of techniques that decomposes a problem into the execution of a sequence of operators that combined in the proper way can explore states and subgoals of the problem until reaching a solution. Although multiple advances were made in the design of a GPS, further research on this topic were discontinued due to the limitations of the classical AI and cognition described in the Section 2.2.

Similar to the GPS, *Artificial General Intelligence (AGI)* is a new approach based in the modern AI techniques (LLM based) which is able to understand, learn and apply knowledge in any cognitive task emulating the capabilities of a human. Thus, because AGI focus on replicating human intelligence in a broader sense, problem-solving capabilities emerge as part of the generalization of the human knowledge, where it is required a new framework that interacts with this AGI in order to bring those capabilities to the reality.

> *"Demonstrating that a system can perform a requisite set of tasks at a given level of performance should be sufficient for declaring the system to be an AGI; deployment of such a system in the open world should not be inherent in the definition of AGI" Morris et al. (2023)*

It is important to note that, although AGI does not necessarily require interaction with the environment, this term is commonly used by frameworks that deploy AGI as their main functionality (other common names include autonomous AI, AGI agents, etc.). Henceforth, we will refer to these architectures as *AGI deployments*.

There have been multiple implementations of AGI deployments. Some focus on specializing general knowledge into a specific application, where it is encapsulated inside a cognitive agent that guides the AI by informing it about the current status of the environment or the available tools.

An example of a specialized deployment is VOYAGER. It uses a three-module-based architecture: the *automatic curriculum*, which describes the current state of the agent and saves relevant information; the *iterative prompting mechanism*, which maintains a feedback loop between the actions coded by the AI and the feedback obtained; and the *skill library*, which enables the AI to learn and store previous actions and completed subgoals to encourage tool reuse. By using this architecture, VOYAGER demonstrates the capacity to autonomously play the videogame *Minecraft*, achieving multiple requested goals.

> *"VOYAGER exhibits superior performance in discovering novel items, unlocking the Minecraft tech tree, traversing diverse terrains, and applying its learned skill library to unseen tasks in a newly instantiated world. VOYAGER serves as a starting point to develop powerful generalist agents without tuning the model parameters." Wang et al. (2023)*

Another relevant framework is AUTOGPT. This framework eases the deployment of autonomous agents for minor tasks. It handles task management, tool selection, multiple prompting techniques, and more. For deploying agents, AUTOGPT provides the so-called FORGE, which automatically connects all the mechanisms and servers needed to not only start running the agent but also provide the user with various tools to interact and debug in real-time.

> *"AutoGPT uses the concept of stacking to recursively call itself [...], using this method and with the help of both GPT 3.5 and GPT 4, creates full projects by iterating on its own prompts." Fezari and Ali-Al-Dahoud (2023)*

Similar to AutoGPT, AGI deployments are being extended into projects for company software assistance (MetaGPT, Wu (2023)), autonomous deployments (SuperAGI, TransformerOptimus (2023)), design-creativity assistance (AgentGPT, Reworkd (2023)), and more.

#TODO: Seccion sobre los 5 Niveles de AGI, definido por OpenAI: https://www.tomsguide.com/ai/chatgpt/openai-has-5-steps-to-agi-and-were-only-a-third-of-the-way-there

2.4 AGI Behavior Techniques

It has been discussed in previous sections how LLMs can be used to build and deploy AGIs. However, there are multiple ways to modify or *tune* the behavior of an AI so it can be successfuly implemented into agents. The following highlights the key techniques in the state-of-the-art for building Agents from LLMs:

- **Fine-Tuning:** All LLMs consist of one or multiple layers in a deep learning-based neural network. These layers contain a set of parameters that define the *knowledge* or behavior of the AI. By training an already stable LLM with a field-specific dataset, we can improve the accuracy of the AI with the provided knowledge and define the format or guidelines it must follow. Although this method obtains better results than the following techniques, it may lose emergent properties from the original LLM and requires a previous analysis of the provided dataset (bias, expectations vs. results, data cleaning, etc.). In this field, techniques such as custom training or freezing can be used to improve the results of fine-tuning.

- **Prompt Engineering:** Without modifying the core parameters of the LLM, we can still change the expected behavior by introducing a goal-crafted prompt that the AI will receive as input and use as guidelines. Although this method does not ensure improved accuracy over fine-tuning, it does not alter the model's parameters and enables the definition of complex reasoning structures. It is important to note that, while security risks for AI misbehavior can arise from this method, techniques such as Chain of Thought (CoT), Retrieval-Augmented Generation (RAG), or Few-Shot prompting lead to reasoning properties that have not been achieved with other methods. Sahoo et al. (2024) delves further into this field.

- **Composition:** By using both prompt engineering and fine-tuning methods, it is possible to streamline multiple agents by splitting the expected behavior into multiple subgoals. These agents are usually referred to as *Experts* and reduce the hallucination of LLMs by distributing the attention needed for completing a given query over multiple runs instead of a single instance. Some AGI frameworks, such as AutoGen Wu et al. (2023), are fully based on this method.

- **Interpretable Feature Excitation:** When using LLMs, abstract features appear to have a relationship with visible patterns within the parameters of the neural network. These parameters can be excited (e.g., increasing the

influence of those parameters on the output) to regulate behaviors related to that feature. The research conducted by Viteri et al. (2024) in this topic opens the door for using this method in future agent designs.

2.5 Cognitive Architectures

Regardless of the methodology used for building the LLM/Agent, the combination of mechanisms, tools, and agents constitutes a *cognitive architecture*.

Using classical AI and reinforcement learning, Laird (2019) introduced SOAR as a unified cognitive architecture that integrates various cognitive functions such as learning, memory, and problem-solving into a single framework. SOAR employs a decision cycle that involves proposing, evaluating, and selecting operators based on the current state.

In contemporary research, cognitive architectures have evolved by incorporating more tools and integrating the capabilities of LLMs as the primary mechanisms. Wang et al. (2024) provide an exhaustive survey of the main architectures for AGI deployments in recent years, describing the PMPA model as an unified framework that encompasses all the studied architectures.

The PMPA model addresses four main topics that should be implemented by the cognitive architecture:

- *Profile*: Defines the main guidelines and rules for the agents implemented, targeting the main goals, knowledge base, and behavior.
- *Memory*: Defines the information and data obtained from the agent's environment and establishes a structure and mechanism to retrieve, codify, and classify the relevant knowledge.
- *Planning*: Defines the mechanism which enables the agent to decompose the target goal into multiple subgoals, emulating human planning capabilities.
- *Action*: Defines the mechanism which connects or translates the orders requested by the agent into a set of tools that will interact with the agent's environment or behavior.

It is important to note that, unlike SOAR, this framework does not prescribe specific methods for connecting each module. Instead, it outlines the primary properties that a cognitive architecture must possess to be viable for AGI deployment. Besides PMPA, there are other AGI frameworks, such as the Agent Protocol-AI-Engineer-Foundation (2023)-, which propose alternative API, behavior, and module standards. However, further advancements in cognitive architectures are necessary to establish which standard will ultimately be adopted.

#TODO: Añadir informacion sobre arquitecturas mas relevantes del estado del arte: ReAct, VerifyAgain y/o expectativas sobre Q* de OpenAI

3. Objectives

3.1 Problem Description

3.2 Requirements

3.3 Methodology

3.4 Workplan

4. Development Platform

4.1 Python

4.2 OpenAI (library)

4.3 Langchain

4.4 LangGraph

4.5 FastAPI and Requests

4.6 AgentProtocol

4.7 ROS2

4.8 PyAutoGUI and Pynput

4.9 OpenCV

5. Software Development

6. Hardware Development

Bibliography

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aeronautiques, C., Howe, A., Knoblock, C., McDermott, I. D., Ram, A., Veloso, M., Weld, D., Sri, D. W., Barrett, A., Christianson, D., et al. (1998). Pddl— the planning domain definition language. *Technical Report, Tech. Rep.*

AI-Engineer-Foundation (2023). Agentprotocol.

Fezari, M. and Ali-Al-Dahoud, A. A.-D. (2023). From gpt to autogpt: a brief attention in nlp processing using dl. *Preprint*.

Fikes, R. E. and Nilsson, N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208.

Laird, J. E. (2019). *The Soar cognitive architecture*. MIT press.

LangChain (2024). Langchain documentation: Agents module. Accessed: 2024-07-05.

McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4):12–12.

Morris, M. R., Sohl-Dickstein, J., Fiedel, N., Warkentin, T., Dafoe, A., Faust, A., Farabet, C., and Legg, S. (2023). Levels of agi: Operationalizing progress on the path to agi. *arXiv preprint arXiv:2311.02462*.

Newell, A., Shaw, J. C., and Simon, H. A. (1959). Report on a general problem solving program. In *IFIP congress*, volume 256, page 64. Pittsburgh, PA.

Reworkd (2023). Agentgpt.

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson.

Saeidnia, H. R. (2023). Welcome to the gemini era: Google deepmind and the information industry. *Library Hi Tech News*, (ahead-of-print).

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3(3):417–424.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

TransformerOptimus (2023). Superagi.

Vervaeke, J. A. (1999). *The naturalistic imperative in cognitive science*. PhD thesis.

Viteri, S., Nanda, N., and Smith, J. (2024). Scaling monosemanticity in transformer circuits. Accessed: 2024-07-08.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. (2023). Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Wu, A. (2023). Metagpt.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., and Wang, C. (2023). Autogen: Enabling next-gen llm applications via multi-agent conversation framework.