Exploring the Use of LLMs for Teaching AI and Robotics Concepts at a Master's Degree

Miguel Angel Gonzalez-Santamarta, Francisco J Rodríguez Lera,
Miguel Ángel Conde, Francisco Rodríguez-Sedano and
Camino Fernández

April 24, 2024

Exploring the use of LLMs for teaching AI and Robotics concepts at a Master's Degree

Miguel Á. González-Santamarta[0000−0002−7658−8600], Francisco Javier
Rodríguez-Lera[0000−0002−8400−7079], Miguel Á.
Conde-González[0000−0001−5881−7775], Francisco
Rodríguez-Sedano[0000−0001−5909−1566], and Camino
Fernández-Llamas[0000−0002−8705−4786]

Robotics Group, University of León, León, 24006, Spain
{mgons,fjrodl,mcong,fjrods,cferll}@unileon.es

Abstract. This article explores the use of Large Language Models (LLM) as transformative tools for teaching Artificial Intelligence (AI) and Robotics concepts at the master's level. LLMs, exemplified by models like ChatGPT, present a unique opportunity to revolutionize the pedagogical landscape by offering advanced capabilities in any service robot. The study investigates the integration of LLMs in the instructional framework, through the llama_ros tool, capable of replacing different classic cognitive functions in a transversal project across different subjects of an official master's degree. The research presents as an example the creation of an LLM-based chatbot on an open hardware platform called Mini Pupper. The reader will find how to emphasize the potential of LLMs to shape their inclusion in bachelor's or master's programs.

Keywords: LLM · robotics · llama_ros · chatbot · ROS 2

1 Introduction

Bachelor's and Master's degrees in computer science span a wide spectrum of courses, encompassing subjects such as Path Planning and Navigation, Robot Perception, and Human-Robot Interaction. Students will also explore the intricacies of Machine Learning for Robotics, delving into algorithms that enable robots to learn from and adapt to their environment. Ethics in AI and Robotics is integrated into the curriculum, encouraging students to critically evaluate the societal implications and ethical considerations associated with these rapidly advancing technologies. The collaboration between Artificial Intelligence (AI) and robotics [6] has played a pivotal role in propelling the progress of robotic systems, granting machines the capability to execute intricate tasks and acclimate to diverse surroundings.

Embarking these concepts in a Master's degree in Robotics and AI represents an exciting journey into the realm of cutting-edge technology, where students dig into the convergence of robotics and AI. The students analyze the realm of classic AI, exploring algorithms, and computer vision techniques. A few years ago, some of these approaches changed for employing machine learning in different components of robot parts, for reaching end-to-end machine learning for robotics. Nowadays, the community is pushing for introducing in almost every part of a robot a component based on Large Language Models (LLM) [7]. Thus, this paper goes through the use of LLMs in a service robot as a part of a traversal project that runs in the Master's Degree of Robotics and AI at the University of León and involves different courses in parallel for two semesters.

1.1 Contribution

This paper provides a comprehensive overview of key AI artifacts utilized in Robotics and AI master's degree programs, emphasizing the substitution of traditional methodologies with LLMs. This exploration aims to shed light on the evolving landscape of educational tools within these programs and the transformative impact brought about by integrating LLMs.

On the technical front, our contributions are twofold. Firstly, it introduced the llama_ros tool[1], a dedicated solution designed to encapsulate the state-of-the-art functionalities of llama.cpp which is publicly available on GitHub[2]. This package serves as a tool for advancing the capabilities and accessibility of the llama.cpp framework in ROS 2 [20] context. Secondly, we present a practical example, showcasing the integration of this package in a project that unifies multiple courses into a cohesive multi-subject proposal. This practical application demonstrates the versatility and practicality of the llama_ros package in real-world, interdisciplinary scenarios.

2 State of the Art

During the last two years, different researchers have been working on using advanced AI-based tools, such as ChatGPT [1], to students, emphasizing their availability for generating original written content to aid academic assessments. The surge in the adoption of LLMs is particularly exemplified by the popularity of OpenAI's ChatGPT [21].

For instance, [3] presents the existing gap in studies examining students' use of LLMs as learning tools leading to the primary objective of the paper: conducting an in-depth case study on the application of ChatGPT in engineering higher education. The objectives encompass investigating whether engineering students can produce high-quality university essays with LLM assistance, evaluating the effectiveness of current LLM identification systems in detecting such essays, and exploring students' perceptions of the usefulness and acceptance of LLMs in learning. In this case, the paper focuses on identifying current parts of a robot and evaluating the integration of LLMs instead of classic AI solutions.

---

[1] `https://github.com/mgonzs13/llama_ros`

[2] `https://github.com/ggerganov/llama.cpp`

Generative AI tools, notably exemplified by ChatGPT, have witnessed a surge in popularity and widespread utilization across diverse sectors [2]. As presented by Aruleba, ChatGPT is recognized by UNESCO as the fastest-growing app in history [24]. It played a pivotal role in introducing the Generative AI concept to the general public. With its unique features and the ability to engage in discussions on a myriad of topics, ChatGPT stands out as one of the most powerful AI applications, especially for education [8]. Serving as a chatbot capable of persuasive conversations, it is employed for various tasks such as essay writing, literature reviews, paper enhancement, and even computer code generation. Of course, its use for education is out of doubt.

Wensheng et al. work [10] presents a complete vision of all capacities such as learning assistance tools, cross-language communications, or personalized learning experiences among others. However, here it is proposed how to use as a tool for students and teachers to update the current models and state-of-the-art engines such as PDDL (Planning Domain Definition Language) [9] or Behavior Trees, thus, the new changes motivated by the Generative AI should be visited not only by students but also by teachers and their syllabus. The study presented here seeks to fit in educational contexts as in [2], emphasizing the prevalent issues of integrity and loss of knowledge.

3 Materials And Methods

The massive adoption of LLMs has had a significant impact across various subjects in any Master's Degree program devoted to Robotics and AI. In this section, it is presented the Master and the main subjects involved in this first step.

3.1 Courses

The Master's Degree in Robotics and AI is taught at the University of León, Spain. It is organized following a mixed structure of common courses and a set of subjects focused on specific itineraries, with a total duration of one academic year, divided into two semesters. The semester is the basic temporal unit and consists of 30 ECTS credits. Each ECTS credit corresponds to 25 hours of student work. To obtain the degree, the student must complete (or have previous studies recognized for) a total of 60 ECTS credits.

The teachings are structured into two modules: Robotics and AI. These two modules contain the subjects presented in tables 1 and 2. In addition to these courses, the degree has 9 ECTS options for those pupils that seek a more practical perspective and also the Final Project which has 9 ECTS.

3.2 LLMs Impact

This section performs a concise analysis of the primary subjects impacted by the integration of LLMs and its possible negative impact on students. Afterwards, the research transitions into the exploration of practical integration strategies

Table 1. Courses in AI Module

ECTS	Sem.	Course
3	1	Data Preprocessing
4.5	1	**Artificial Intelligence**
4.5	1	**Computer Vision**
4.5	1	Machine Learning
4.5	2	Programming in Distributed Data Environments
3	1	Deep Learning
3	2	**Object Recognition**
3	2	Machine Learning Applications (Opt)
3	2	Reinforcement Learning (Opt)
3	2	**Trends in Deep Learning** (Opt)

Table 2. Courses in Robotics Module

ECTS	Sem.	Course
4.5	1	**Service Robotics**
4.5	1	Middleware for Software Design in Robots
3	2	Human-Robot Interaction
3	2	**Cognitive Robotics**
3	2	Multi-Agent Systems (Opt)
3	2	Trends in Robotics I (Opt)
3	2	**Trends in Robotics II (Opt)**

- Artificial Intelligence: Incorporating LLMs alongside classical AI methods in an AI course can have both positive and negative effects. Relying solely on LLMs may limit the depth of understanding of traditional AI concepts such as machine learning algorithms, vision and robotics. Students might miss out on foundational knowledge in these areas.
- Computer Vision and Object Recognition: The integration of LLMs in these two courses, specifically Large Visual Models in the Computer Vision course could change the practical process for students. The ability of these models to comprehend linguistic contexts has empowered the interpretation of images and videos, leading to a deeper level of understanding and visual analysis. However, students might lose the in-depth understanding of foundational concepts such as image processing, feature extraction, object detection, and image segmentation, which are crucial for computer vision applications, as well as classy vision-specific frameworks.
- Service Robotics: The adoption of LLMs has transformed human-robot interaction. The robots' ability to understand and generate natural language has elevated the quality of communication in personalized services. Students explore how LLMs can be used to enhance the understanding of commands, questions, and responses, redefining how robots provide adaptive and user-centered services. However, LLMs may not inherently address challenges related to real-world service environments, such as dynamic and unstructured spaces, which may lead to a narrow focus on language-centric applications or hinder their ability to design and deploy fundamental algorithms.
- Cognitive Robotics: This course has undergone a significant shift with the integration of LLMs. Now, robots not only execute tasks but also understand and generate natural language more intelligently. Students explore how LLMs can be applied to improve environmental perception and the ability of robots to interact more contextually and adaptively in complex environments. Relying solely on LLMs may limit students' exploration of dedicated cognitive architectures and models, hindering a comprehensive understanding of cognitive robotics principles, such as classic Deliberative or Reactive approaches, and avoiding state-of-the-art solutions such as BTs or FSMs.
- Trends in Robotics: A course with this title has to overview LLMs. Students analyze how LLMs can forecast and model current trends, providing valuable insights for anticipating and adapting to the dynamic evolution of robotics.

As a result of these courses, we have the Figure 1. It presents an overview of a simple robot with its Application System Structure with the cognitive functions of a robot, a single presentation of a model employed in the course and the engines applied.

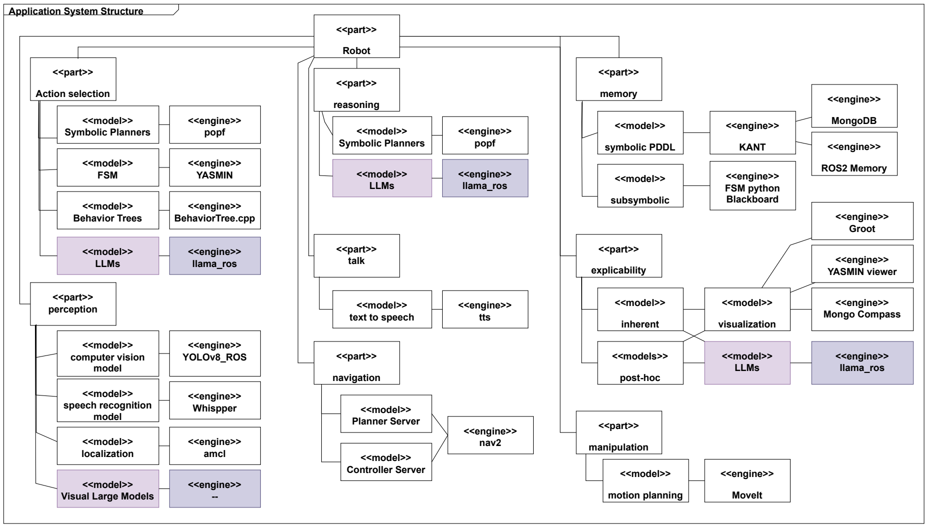

Fig. 1. Application System Structure of a robot with models and engines for a robotics course (it was simplified for the sake of visualization). This structure includes the following cognitive functions: action selection, perception, reasoning, talk, memory, explicability and manipulation.

4 Practical Case

4.1 Project Description

The project takes into account several subjects from both semesters, including Service Robotics, Middleware, and Artificial Intelligence in the first semester, and Trends in Robotics II and Trends in Machine Learning in the second.

During the second semester in the Cognitive Robotics course, Finite State Machines, such as SMACH [4] and YASMIN [15], were used for performing a simple task to provide a service robot that can move between a set of points of interest marked in an apartment. These points of interest are defined a priori in a pre-loaded file.

Behavior trees are also introduced. Deliberative systems, based on PDDL [9], are incorporated to execute tasks in a non-deterministic manner, enabling the capability to overcome issues that may arise during the typical deployment of a robot. Additionally, these systems facilitate interaction with humans with grammar and rule-based NLP is integrated to incorporate dialogue capabilities and reasoning using PDDL.

The proposal is to introduce the use of LLMs in the different components grayed in Figure 1 where it is possible to change action selection and reasoning as is presented in [14]. Here we describe how to integrate the LLMs in the robot in the shape of a chatbot in order to provide a level of natural interaction to students, opening the door to use the robot as their tutor.

Building a ROS-based chatbot for a robot involves integrating various software components to enable effective communication and interaction. Here are the main software components typically used in a ROS-based chatbot for a robot. First, the Speech Recognition Module processes spoken language and converts it into text. Popular libraries like PocketSphinx or Google's Speech Recognition API were utilized in previous research. Second, the Natural Language Processing (NLP) Engine is essential for understanding and interpreting user input. Common NLP frameworks include Dialogflow, Rasa NLU, or Wit.ai, enabling the chatbot to comprehend and extract meaning from natural language. Third, the Text-to-Speech (TTS) Module converts textual responses generated by the chatbot into spoken words. Popular TTS engines like Google Text-to-Speech or Festival can be integrated. Fourth, the dialogue management component which orchestrates the conversation flow, manages context, user responses, and system actions. This component ensures a coherent and context-aware interaction.

Previous to the use of LLMs, a knowledge base was required to store information that the chatbot could reference during interactions. This may include data about the robot's capabilities, environmental information, or responses to frequently asked questions. At this point, the knowledge base is used for other components in our proposal and is presented as the "memory" part.

4.2 llama.cpp

llama.cpp, as outlined in the GitHub repository [11], constitutes a project dedicated to executing the LLaMA model utilizing integer quantization on hardware-constrained machines. This implementation is rooted in plain C/C++ and operates independently of external dependencies, facilitating the deployment of these models across diverse platforms, including the potential for GPU acceleration.

Engaging with llama.cpp involves a thoughtful approach to prompt engineering. Drawing inspiration from Microsoft's methodology [5], adeptly crafting prompts emerges as a critical aspect in achieving desired outcomes when working with LLM models. This endeavor demands a blend of creativity and precision, involving the careful selection of textual prompts to guide the model toward generating text pertinent to the specified objectives. These prompts are intricately linked with tokens.

Tokens, the elemental units of text or code, serve as the building blocks for language processing and generation within LLM AI. Depending on the chosen tokenization method, these units may encompass characters, words, subwords, or other text/code segments. Tokens are assigned numerical values or identifiers and organized into sequences or vectors, functioning as both inputs and outputs for the model. They represent the foundational elements that convey meaning in language to the model.

Embeddings, in this context, denote the representations or encodings of tokens, ranging from sentences to entire documents. These embeddings capture past event representations generated during the prompting interaction, existing within a high-dimensional vector space where each dimension corresponds to a learned linguistic feature. Embeddings play a central role in preserving and deciphering the meaning and interrelationships within textual data and they act as the mechanism through which the model discerns and distinguishes between various tokens or language components.

4.3 llama_ros

During this research for the right integration of llama.cpp in ROS 2 [20] projects, it is created the tool Llama_ros. The llama_ros tool, detailed in [12], encompasses a suite of ROS 2 packages designed to integrate llama.cpp [11] into the ROS 2 framework to use quantized [19] LLMs locally while operating efficiently within their hardware constraints, leveraging both CPU and GPU. This integration facilitates the utilization of ROS 2 messages for text generation with LLMs, tokenization of text, and the creation of embeddings from textual data.

To illustrate, the UML class diagram of llama_ros is depicted in Figure 2. The primary class, the Llama, encapsulates all the functionalities of the llama.cpp independently of ROS 2. This class serves as the foundation for the LlamaNode class, a ROS 2 node that exposes ROS 2 interfaces, enabling the invocation of the llama.cpp functions from other ROS 2 nodes.

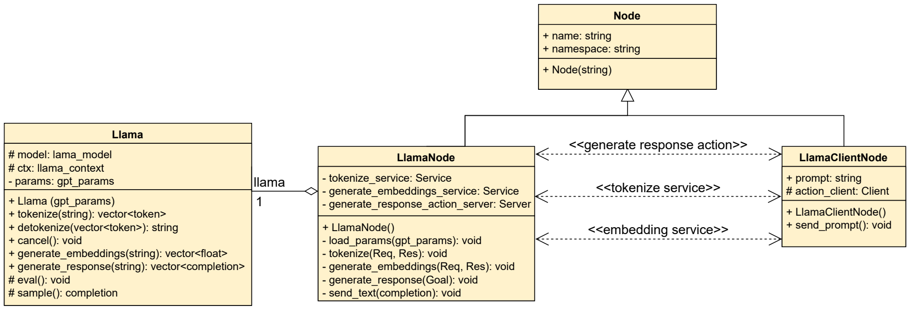

Fig. 2. UML diagram of llama_ros. The diagram includes the Llama class, which wraps llama.cpp, the LlamaNode class, which presents the ROS 2 interfaces; and the LlamaClientNode, which is a node example to use llama_ros.

This tool presents a ROS 2 action server for text generation, alongside two services dedicated to text tokenization and embedding generation. These services prove invaluable when implementing more intricate prompt engineering techniques [18,16]. For example, the embedding service facilitates the conversion of text into vectors, allowing the creation of a vector database. Subsequently, retrieval functions can be employed to search for vectors similar to a given text. This capability enhances the precision of prompt crafting by enabling the retrieval of more accurate prompts based on vector similarity.

4.4 Mini Pupper

Mini Pupper is an economical and open-source quadruped robot inspired by the Stanford Pupper model. Designed for scientific exploration, it enables the assembly of a customizable robotic platform with processing capacity for diverse applications. It has the option of adding lidar or a standard USB camera.

Examining the hardware specifications, Mini Pupper presents the following dimensions: a compact 210x110x165mm frame, a weight of 509g for optimal portability, a 1000mAh rechargeable battery through micro-USB, a Raspberry Pi 4B processor for computational robustness, a 240x320 ISP LCD screen for interactive displays, a 5V 1A input charger for efficient recharging, and compatibility with Ubuntu and ROS 2, offering a versatile environment for scientific exploration and experimentation. Its unassuming design and technical specifications make it an ideal subject for some practicing course competencies.

4.5 ChatBot Application

The proposed chatbot application is an integrated system within ROS 2, comprising speech recognition for human input, LLMs for generating responses, and text-to-speech for human interaction.

For speech recognition, the chatbot leverages whisper_ros [13], based on whisper.cpp[3]. This implementation utilizes the whisper-large-v3 model [22] from OpenAI, coupled with Silero VAD [23] for voice activity detection. Conversely, llama_ros [12] serves the purpose of response generation. This allows flexibility in selecting LLMs tailored to various domains. Additionally, VITS [17] facilitates text-to-speech conversion, although alternative models can be integrated.

Finally, by integrating into ROS 2 as a distributed system, the chatbot enables the execution of LLMs, the Whisper model, and text-to-speech on dedicated machines, while audio functionalities, such as capturing and playing audio, can be managed on the robot's main computer, for instance inside the Raspberry of the Mini Pupper.

5 Discussion and Conclusions

In incorporating LLMs, in local computers thanks to llama_ros, into our Master's-level courses on AI and Robotics, we have witnessed both positive and challenging aspects. The use of LLMs in teaching has proven beneficial in certain domains. In the context of AI, LLMs offer students an insightful exploration

---

[3] `https://github.com/ggerganov/whisper.cpp`

into natural language understanding, generation, and their applications, providing a hands-on experience with cutting-edge language models. Additionally, the integration of LLMs in Service Robotics has showcased advancements in human-robot interaction, emphasizing the significance of natural language communication in service-oriented scenarios.

However, the exclusive reliance on LLMs does present limitations, such as the potential neglect of foundational concepts, the overemphasis on language-centric tasks, and the challenges in multi-modal integration, highlighting the need for a balanced approach. Similarly, in Cognitive Robotics, the broader spectrum of cognitive capabilities, including perception and reasoning, requires a more comprehensive approach beyond language-centric tasks. To ensure a comprehensive education in AI and Robotics, it is imperative to supplement LLMs with traditional methods, core algorithms, and practical applications that constitute the holistic landscape of these fields.

As we move forward, we recommend an integrated approach that leverages the strengths of LLMs while addressing their limitations. By combining the advancements offered by LLMs with a well-rounded curriculum encompassing traditional methods and emerging technologies, we can empower students to navigate the dynamic landscape of AI and Robotics with a thorough understanding of both the foundational principles and the latest innovations.

Acknowledgment

This work has been partially funded by an FPU fellowship provided by the Spanish Ministry of Universities (FPU21/01438) and the Grant PID2021-126592OB-C21 funded by MCIN/AEI/10.13039/501100011033.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Aruleba, K., Sanusi, I.T., Obaido, G., Ogbuokiri, B.: Integrating chatgpt in a computer science course: Students perceptions and suggestions. arXiv preprint arXiv:2402.01640 (2023)
3. Bernabei, M., Colabianchi, S., Falegnami, A., Costantino, F.: Students' use of large language models in engineering education: A case study on technology acceptance, perceptions, efficacy, and detection chances. Computers and Education: Artificial Intelligence **5**, 100172 (2023). https://doi.org/10.1016/j.caeai.2023.100172
4. Bohren, J., Cousins, S.: The SMACH high-level executive. IEEE Robotics & Automation Magazine **17**(4), 18 – 20 (2011). https://doi.org/10.1109/MRA.2010.938836
5. Bolaños, M., Maeda, J., Buck, A.: Orchestrate your AI with Semantic Kernel, `https://learn.microsoft.com/en-us/semantic-kernel/overview/`
6. Brady, M.: Artificial intelligence and robotics. Artificial intelligence **26**(1), 79–121 (1985)

7. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al.: A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology (2023)
8. Diego Olite, F.M., Morales Suárez, I.d.R., Vidal Ledo, M.J.: Chat GPT: origen, evolución, retos e impactos en la educación. Educación Médica Superior **37**(2) (2023)
9. Fox, M., Long, D.: PDDL2.1: An extension to PDDL for expressing temporal planning domains. J. Artif. Intell. Res. (JAIR) **20**, 61–124 (12 2003). https://doi.org/10.1613/jair.1129
10. Gan, W., Qi, Z., Wu, J., Lin, J.C.W.: Large language models in education: Vision and opportunities. In: 2023 IEEE International Conference on Big Data (BigData). pp. 4776–4785. IEEE (2023)
11. Gerganov, G.: GitHub - ggerganov/llama.cpp: Port of Facebook's LLaMA model in C/C++ — github.com. `https://github.com/ggerganov/llama.cpp` (2023)
12. González-Santamarta, M.A.: llama_ros. `https://github.com/mgonzs13/llama_ros` (Apr 2023)
13. González-Santamarta, M.A.: whisper_ros. `https://github.com/mgonzs13/whisper_ros` (Apr 2023)
14. González-Santamarta, M.Á., Rodríguez-Lera, F.J., Guerrero-Higueras, Á.M., Matellán-Olivera, V.: Integration of Large Language Models within Cognitive Architectures for Autonomous Robots (2023)
15. González-Santamarta, M.Á., Rodríguez-Lera, F.J., Matellán-Olivera, V., Fernández-Llamas, C.: Yasmin: Yet another state machine. In: Tardioli, D., Matellán, V., Heredia, G., Silva, M.F., Marques, L. (eds.) ROBOT2022: Fifth Iberian Robotics Conference. pp. 528–539. Springer International Publishing, Cham (2023)
16. Gupta, U.: GPT-InvestAR: Enhancing Stock Investment Strategies through Annual Report Analysis with Large Language Models. arXiv preprint arXiv:2309.03079 (2023)
17. Kim, J., Kong, J., Son, J.: Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech (2021)
18. Li, N., Kang, B., De Bie, T.: SkillGPT: a RESTful API service for skill extraction and standardization using a Large Language Model. arXiv preprint arXiv:2304.11060 (2023)
19. Lin, D., Talathi, S., Annapureddy, S.: Fixed point quantization of deep convolutional networks. In: International conference on machine learning. pp. 2849–2858. PMLR (2016)
20. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot Operating System 2: Design, architecture, and uses in the wild. Science Robotics **7**(66), eabm6074 (2022). https://doi.org/10.1126/scirobotics.abm6074, `https://www.science.org/doi/abs/10.1126/scirobotics.abm6074`
21. OpenAI: GPT-4 Technical Report. `https://arxiv.org/abs/2303.08774` (2023)
22. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust Speech Recognition via Large-Scale Weak Supervision (2022). https://doi.org/10.48550/ARXIV.2212.04356
23. Team, S.: Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier. `https://github.com/snakers4/silero-vad` (2021)
24. UNESCO: UNESCO: Governments must quickly regulate Generative AI in schools (Sep 2023), `https://www.unesco.org/en/articles/unesco-governments-must-quickly-regulate-generative-ai-schools`