2024 SinfonIA Uniandes Team Description Paper

SinfonIA Uniandes

November 2023

Abstract

1 Introduction

The SinfonIA Pepper Team is an innovative partnership between Universidad de Los Andes, Celcia, Colsubsidio and Bancolombia, dedicated to pioneering education and research in social robotics, machine learning, and AI throughout Colombia. Our mission is to integrate robots and AI in practical services like information sourcing, guidance, and entertainment, thereby enlightening the Colombian community about the positive impacts of social robots in everyday scenarios. The RoboCup@Home Social Standard Platform League serves as our experimental arena, facilitating a rich knowledge exchange within the RoboCup network and positioning us at the forefront of international social robotics research aimed at real-world application. Additionally, our research introduces an inventive method for object segmentation tailored for platforms with limited computational resources, potentially setting a new benchmark for efficiency in the field.

2 Background and previous experiences in RoboCup

The SinfonIA Uniandes team, with a strong background in RoboCup, participated in the Small Size League (SSL) from 2011 to 2017, becoming the first Latin American team in the hall of fame. In 2019, they debuted in the Social Standard Platform League (SSPL) with the SinfonIA Pepper team, showcasing skills such as verbal interaction and object recognition. Despite technical challenges, they secured a tied fourth place in RoboCup 2019 Sydney. In 2022, despite not updating Pepper's skills due to the COVID pandemic, they were the runners-up in RoboCup Bangkok, demonstrating their potential despite persistent technical challenges. Finally, at the RoboCup in France in 2023, The SinfonIA participated by showcasing a wide variety of tasks, including receptionist, carry my luggage, stickler of the rules, serve breakfast, storing groceries, Gpsr and Egpr. This participation was an environment of abundant learning opportunities to apply on team's future work.

3 Software Architecture

3.1 Distributed ROS architecture

Our robotic architecture is structured in layers, starting with a Toolkit at the base that connects the robot's physical components to ROS, allowing AI to interface with these elements. This layer ensures the robot is equipped for its intended social roles. Subsystems built upon this include Manipulation, using C++ and Moveit for movement precision; Perception, for recognizing objects and individuals; Navigation, for autonomous traversal; and Speech, which applies NLP for smoother human-robot dialogue. The Manipulation subsystem utilizes inverse kinematics for accurate arm movements. The Speech subsystem stands out with its ability to convert spoken language into text, enhanced with cloud-based NLP models and neural networks for nuanced conversations. The Interface subsystem offers both desktop and web-based controls, the latter developed with Python and Django, ensuring flexible robot management. Perception is advanced with video recognition capabilities, identifying objects, faces, and postures, and is supported by a lightweight Yolo v7 model. The central task module interface orchestrates these subsystems for complex task performance, vital for adaptability in scenarios like RoboCup@Home, enhancing the robot's operational efficiency. At the top, the Large Language Model module integrates with the task module, facilitating nuanced human interactions by processing natural language, positioning our robot at the cutting edge of social robotics technology.

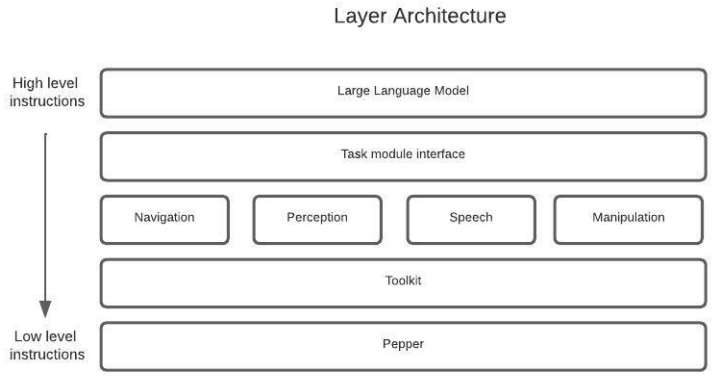

Figure 1: Software Architecture

3.2 Cloud and local redundancy

The team has been working in a robust software architecture consisting of three levels of redundancy within their subsystems: local, remote, and cloud. At the local level, the software architecture consists of the basic infrastructure for the robot to function properly by itself. At the remote level, the architecture is designed to provide additional competencies to the robot by implementing different pre-trained models in a remote machine. At the cloud level, the architecture includes cloud-based applications for the subsystems and remote backups, allowing for secure data access and reliable recovery capabilities. This architecture allows us to have redundancy in complex tasks such as face recognition or natural language processing, being the cloud solutions the most accurate but the less available.

4 Research on Robowflex implementation for manipulation team

In the context of improving our planning and motion control capabilities for Pepper from the manipulation area, the need arises to migrate from MoveIt to Robowflex. MoveIt, although robust, may have certain limitations, especially in terms of intuitiveness for new members and efficiency in code creation. The transition to Robowflex is justified by its intuitive approach, code development optimization, and advanced motion planning capabilities.

By implementing Robowflex, we aim to achieve several key objectives. Firstly, we want to enhance accessibility for our members, especially for future newcomers, by providing a more user-friendly interface and a smoother learning curve compared to MoveIt. Additionally, we seek to optimize efficiency in code creation, enabling faster and maintainable development. We also aspire to utilize Robowflex's benchmarking capabilities to continuously evaluate and improve our motion planning algorithms.

Robowflex is a high-level C++ library for MoveIt. Its API simplifies common use cases while providing low-level access when needed. This tool is particularly useful for the development and evaluation of motion planners, as well as addressing complex problems involving motion planning as a subroutine. Furthermore, it offers visualization capabilities, integrations with other robotics libraries, and complements other robotics packages. [1]

This transition not only benefits our internal operations but also supports research and opens up the possibility of new applications for other robots. By adopting Robowflex, we aim to actively contribute to the field of robotics research by providing a more accessible and efficient environment. This not only propels our own exploration in the area but also paves the way for future research and developments in the field of robotics.

This decision is supported by the consideration of specific limitations of MoveIt, such as potential difficulties in motion planning for robots with limited

mobility, complexity in robot configuration, and challenges in sensor integration—critical aspects in dynamic and collaborative environments.

The implementation of Robowflex follows an object-oriented approach. Robots are constructed and implemented through objects, inheriting properties and behaviors from a base class. This not only simplifies robot configuration through loading files, URDF, and SRDF but also facilitates interoperability with Robowflex.

We have recently successfully adapted Robowflex to the Pepper robot, significantly improving planning and motion control capabilities. Currently, we are working on implementing new ROS services from the manipulation node using Robowflex, marking the gradual migration from MoveIt and other tools to this more versatile platform. This process ensures a smooth transition while maintaining and enhancing tool capabilities.

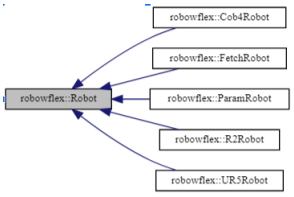

Figure 2: Implementations

5 Research on Natural Interaction with a Pepper Robot through LLMs

5.1 Problem

The field of social robotics is focused on investigating robots as social agents that interact and assist humans in a common environment. Similarly, social robotics over the years has been dedicated to building robots capable of performing a greater variety of tasks, with the aim of creating robots with greater autonomy that can help humans more effectively. In this line of thought, this research arises from the issues present in the discipline of social robotics, which primarily revolve around the misalignment of directives given by developers and the actual behavior of robots in the contexts in which they operate. Likewise, the addressed issue is derived from the highly complex challenge that social robotics faces in enabling robots to clearly understand the environment and instructions given by humans. Consequently, robots may perform effectively in certain specific tasks for which they have been prepared in advance, but they may exhibit undesired behavior in the execution of different activities. For this reason, the present research aims to make use of a Pepper-type robot, equipped with attributes such as dual cameras, a 3D camera, microphones, a gyroscope in its torso, touch and collision sensors, facial recognition and the ability to interpret human expressions. Finally, the use of this robot is included for the implementation of

language models that enable the robot to handle a broader range of tasks it can fulfill.

5.1.1 Objectives

In broad terms, the present research aims to apply and evaluate a system on the Pepper-type robot, enabling it to perform general tasks through instructions given in natural language. In more detail, this research aims to construct a mediating interface between primitive language and the robot's external behaviors. This interface relies on the implementation of language models, specifically both commercial and open-source Large Language Models (LLMs), as a mediator code generator for the robot [2]. Similarly, the goal is to create levels of abstraction in the robot's base code that allow for modifications in the outputs of the LLMs' implementation. Finally, the intention is to evaluate and compare the results obtained by the code generated by both commercial and open-source LLMs [2].

5.2 Methodology

The methodology for evaluating the natural interaction with the Pepper robot is based on a streamlined process proposed by Rojas et al [2] as members of the SinfonIA team. It starts off with the transcription of verbal instructions into textual format (Speech-to-Text process). Subsequent to this transformation, the textual instructions undergo processing by a large language model (LLM). This model is tasked with generating precise code instructions tailored for execution by the robot. Finally, the Pepper robot executes the requested actions, culminating in the successful completion of the designated task. This method aims for the transformation of human instructions into executable commands, leveraging LLMs to bridge the gap between natural language inputs and robot actions.

5.2.1 Prompting

For achieving the transformation of spoken commands to the completion of the requested task, the most important step is the code generation based on natural language processing. To facilitate this step, it is necessary to have a code interface which includes high level functions which align more with the model of the world that the LLM understands, as it was observed in previous trials that a low abstraction level caused more errors. The interface should be easy to use, and the parameters should be as intuitive and well-explained as possible. The interface is provided via prompts to the LLM where Long-String-Prompting and Chain-Prompting were both tried out.

5.2.2 Experimentation

Both open-source and commercial LLMs were evaluated in their capabilities for evaluating spoken requests and transforming them into generated code for task

execution. For this purpose, 720 tasks were evaluated, where 400 of which had no syntax or runtime errors, and thus underwent manual evaluation.

5.3 Results and Analysis

5.3.1 LLMs Performace

The results depicted Fig. 3 show that Llama 2 had the poorer performance out of the three LLMs evaluated, where none of its outputs managed to complete a single task.

GPT3.5 had an improved performance in comparison to Llama, where about half of its outputs passed the automated evaluation, and 40% of the tests where at partially or fully completed. Nonetheless, the GPT4 model far outperformed its counterparts; around 90% of its outputs had a successful automatic evaluation and half of its outcomes resulted in a successful completion of the desired task.

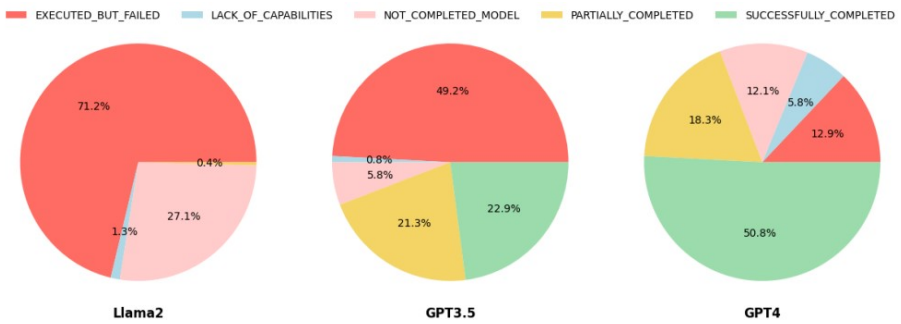

Figure 3: Task results.

5.3.2 Prompting Types Evaluation

The results of the use of both prompting methods can be observed in the graphs of Fig. 4, where the best overall performance was obtained by the use of Long-String-Prompting, as the count of successfully completed tasks is greater in both the GPT3.5 and GPT4 models for Long-String-Prompting. This observation signifies the efficacy of the Long-String-Prompting method in eliciting better performance and successful task completions compared Chain-Prompting as evaluated in the study.

5.4 Conclusions and Future Work

SinfonIA's exploration into Large Language Models (LLMs) for use in GPSR applications, particularly with Pepper-like robots, reveals compelling implications for social robotics. Employing natural language as a conduit for human-robot interaction stands as an intuitive means to convey instructions. Particularly,

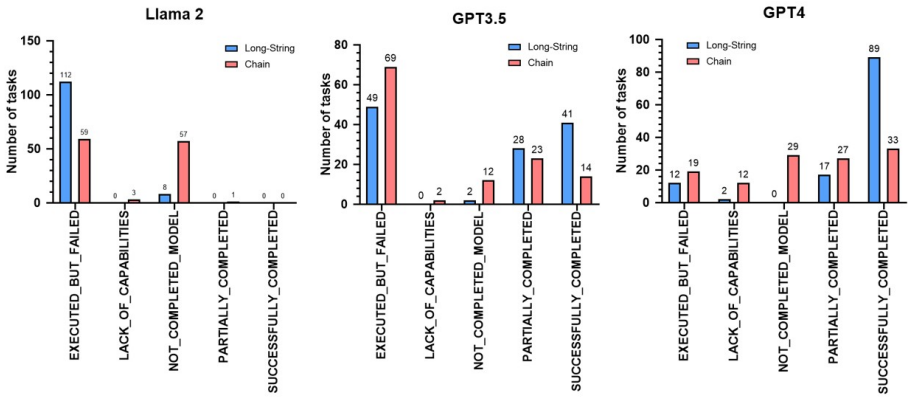

Figure 4: Long-String-Prompting and Chain-Prompting results.

the study showcases the advantageous application of LLMs, notably GPT3.5 and GPT4, allowing for code generation aligned with their understanding of the world. The performance evaluation unveils GPT4 as the frontrunner, achieving a significantly higher success rate in task completion compared to GPT3.5 and Llama 2, specially when utilizing the Long-String-Prompting method. Despite GPT3.5 exhibiting faster response times, its higher failure rate compared to GPT-4 accentuates the latter's suitability for GPSR robots like Pepper. However, contextual understanding remains a challenge, evident in errors primarily attributed to responses beyond the robot's functional scope. GPT4's robust performance with the Long-String approach establishes it as the current optimal choice for GPSR robots, underscoring its potential for enhanced human-robot interactions.

The future trajectory of this research holds immense promise in tandem with AI advancements. Integrating newer models like GPT4 Turbo and exploring specialized language solutions, as unveiled in the OpenAI Dev Day [3], offers avenues for immediate improvement. The prospect of empowering models with retained memory for informed decision-making and expanding contextual windows promises more nuanced, contextually informed solutions. Addressing limitations in the robot's capabilities involves continual enhancement of the task module, thereby expanding the scope of actions and behaviors. Beyond Pepper, this research extends to all social robots, providing a module adaptable to each robot, underscoring a significant stride in social robotics. This adaptation signifies a crucial evolution toward robots seamlessly integrating into daily life, fostering natural interactions based on simple commands, propelling the integration of robots into our societal framework alongside the advancement of large language models.

References

[1] Z. Kingston and L. E. Kavraki, "Robowflex: Robot Motion Planning with MoveIt Made Easy," in IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 2022

[2] L. Rojas and J. A. Romero, "Improving Autonomy and Natural Interaction with a Pepper Robot through the Evaluation of Different Large Language Models", unpublished, Nov. 2023.

[3] OpenAI. (2023, November 6). New models and developer products announced at DevDay. https://openai.com/blog/ new-models-and-developer-products-announced-at-devday

Pepper Software and External Devices

We use a standard *SoftBank Robotics* Pepper robot unit.

Figure 5: Pepper Robot

Robot's Software Description

For Pepper robot we are using the following software:

- ◇ Platform: Ubuntu 9.1

External Devices

Pepper relies on the following external hardware:

- ◇ Atom E3845 Quad core 1.91 GHz
- ◇ Intel HD graphics up to 792 MHz
- ◇ 4 microphones, 2 RGB HD cameras, 5 tactile sensors, touch screen on the breast

Cloud Services

Pepper connects the following cloud services:

- ◇ Object detection and recognition: Yolo and Lightnet.
- ◇ Face Recognition: Microsoft Azure Face API.
- ◇ Speech recognition: Google Speech Recognition Service.

Team Members

SinfonÍa Uniandes

Juan José García	Alexandra Gutierrez	Luccas Rojas
Juan Andrés Carrasquil	David Zamora	Angie Polanía Arias
Camilo Rey	Juan David Guevara	David Cuevas Alba
Valeria Torres Gomez	Maria Paula Estupiñan	Julian Andres Mendez
Maria Alejandra Angulo	Sofia Acosta	Camilo Murcia
David Tobón Molina	David Santiago Rodríguez	Sergio Andrés Cañar
Abel Arismendy	Ana María Gómez	Marcio Padilla
Juan Manuel Silva	Javier Ronaldo Prieto	Andres Julian Bolivar
Santiago Rodríguez	Maria Lucia Benavides	Juan David Villota
Felipe Valencia	Jesus Sandoval	Jose Florez
Catalina Páramo	Cesar Bustos	Daniel Pedraza
Manuela Lovera	Gabriel Padilla	