Social Robots and Edge Computing: Integrating Cloud Robotics in Social Interaction

Theodor-Radu Grumeza(✉), Thomas-Andrei Lazăr, and Alexandra-Emilia Fortiş

West University of Timişoara, Faculty of Mathematics and Informatics, Timişoara 300223, Romania
theodor.grumeza@e-uvt.ro

Abstract. In this study, the authors explore the integration of cloud robotics for social interactions using the SoftBank Robotics Pepper robot in relation with an academic environment. The main objective is to enhance social robots ability to interact with humans by providing guidance and information using the Pepper robot as a case study. The initial steps assess Pepper's language processing capabilities, identifying limitations in its vocabulary. The novelty of this research lies in employing Meta LLAMA 2, a large language model, trained on educational content, to enhance audio-to-text coversion. This approach aims to boost Pepper's comprehension and response to human inquiries, marking a step forward in the practical application of cloud-edge robotics in social contexts.

Keywords: Edge computing · Cloud Robotics · LLAMA 2 · Natural Language Processing · Social Robots · Large Language Models

1 Introduction

Cloud-based robotics [1] combines cloud computing with robotic systems, creating a new approach in cloud-edge applications. By transferring control logic from the robot's local hardware to cloud platforms, it facilitates a more effective process for exchanging services and information among a network of agents.

This research is centered around the cloud-edge integration of social robots, employing the SoftBank Robotics Pepper as use-case [2], interaction with humans being one of its key features. The main idea is to place the humanoid robot into a crowded academic environment to test its abilities in offering guidance and other information. An approach is being explored where the robot should help students or visitors, provide them information on services, and assist them in filling out forms or applications, ensuring that they have up-to-date information on policies and procedures. By implementing the application of Natural Language Processing (NLP), the Meta's LLAMA 2 [3], a collection of pretrained and fine-tuned large language models (LLMs), the authors are trying to transform the social robot's behavior, by training more interaction skills and the use of a broader dictionary of words.

L. Barolli (Ed.): AINA 2024, LNDECT 204, pp. 55–64, 2024.
https://doi.org/10.1007/978-3-031-57942-4_7

Understanding the integration of social robots using a Large Language model with cloud edge computing is covered by the literature review in the second section of this research. The third section refers to the implied methods, the system specifications, highlights some open problems that could be of use in implementing a new approach in social robots and, finally, the phases in implementation of the system. The fourth section offers the results of the research, emphasizing the precision, similarity and F1 score which are of use in determining the correctitude of predictions of the trained model from a lexical and semantic point of view. The last section is dedicated to conclusions and future work.

2 Related Work

Recent scientific research [4–6] emphasizes the importance of edge computing, representing an imperative technology in today's intelligent society, enabling stable and efficient artificial intelligence services for the rapidly expanding array of terminal devices and data streams. Edge devices have integrated into numerous fields, such as social robots enhancing interactive experiences in education and customer service [7], advanced robotics in healthcare [8], warehouse management systems employing robots for efficient inventory control [9], smart agricultural robots revolutionizing precision farming and many more. This widespread adoption illustrates the diverse and innovative applications of edge technology in contemporary industries, extending beyond conventional uses like smart homes and autonomous vehicles [10,11].

Cloud-based robotics, as an application of cloud-edge, represents an innovative convergence of cloud computing technologies with robotic systems [12]. This approach wants to enhance the possibility to reconfige robotic applications while reducing their intrinsic complexity and cost. By migrating the control logic from the local robot hardware to cloud-based platforms, it enables a more efficient mechanism for service and information exchange across a network of robots or agents.

Recent advancements in robotics have been driven using machine learning, especially data-driven methods like Deep Learning and Reinforcement Learning [13], which require repetitive task execution by robots in order to gather data. Simulations capable of handling complex environments and robot dynamics could simplify data collection and protect real robots from damage. They also help identify potential issues scenarios in a cloud environment, involving robots before real-world deployment. While simulation tools like Gazebo, V-REP, Webots, or Choregraphe exist, they often struggle with complex environment simulation as presented in the work of Ayala et al. [14].

Large Language Models (LLMs) are advanced AI assistants, excelling in complex reasoning tasks and expert knowledge in various fields, including specialized areas like robotics in social contexts. They are built using auto-regressive transformers and are initially trained on a vast, self-supervised data sets. This is followed by an alignment process with human preferences through methods like

Reinforcement Learning with Human Feedback (RLHF). Despite the simplicity of this methodology, the high computational demands have restricted LLM development to a limited number of entities. Several publicly released pretrained LLMs, such as BLOOM [15] or LLaMa-1 [3]. However, these models do not fully replace closed "product" LLMs like ChatGPT, BARD, and Claude.

3 Proposed Methods

In what follows, it can be stated that the computational framework employed for this study consisted in a computer equipped with an AMD Ryzen 9 7950X CPU, 32GB of DDR4 RAM, operating at a speed of 3200MHz, and a 1TB NVMe SSD. Additionally, an Nvidia RTX 3070 GPU, featuring 8GB of VRAM, was employed for the data training processes. To manage and process the input data from the robot, a setup of two Virtual Machines (VMs) was established. Each VM was configured with 8 virtual CPUs, 16GB of RAM, and 50GB of NL-SAS storage.

The workflow involved capturing the robot's input in a WAV file format on one VM, followed by processing the audio data and subsequently transmitting the output back to the robot. The operating systems chosen for this setup were Ubuntu 22.04 LTS Desktop for the main computer and Ubuntu 22.04 LTS Server for the VMs. The implementation and coding tasks were carried out using Jupyter Notebook.

Regarding interaction with social robots one needs to take into account some open problems:

- How the incorporation of NLP (Natural Language Processing) could improve human-robot interaction?
- How the response generation in social robots could be made more natural and context-aware?
- What methods can be employed for improving the accuracy of language processing in robots without significantly increase the computational costs?

Using only the main board of the robot and a preset of dictionaries will not be sufficient to offer that type of information. Initial approaches we to evaluate the performance of the language processing feature on the Pepper robot which is of most importance when interacting with humans. The primary objective was to assess Pepper's capability in accurately comprehending spoken questions from humans. Observation indicated that the efficiency of language processing on Pepper is somewhat limited due to its constrained vocabulary also presented in [16]. This vocabulary, essentially a database of words and phrases, determines the range of speech that the recognition system can understand; a broader vocabulary allows for more extensive comprehension.

To facilitate this evaluation, a Python script was designed to record the user's voice. This script prompts the robot to record for a set duration, subsequently transferring the audio file to the same directory as the running script. Following this, a second script comes into play, utilizing the Pinecone API as a dynamic

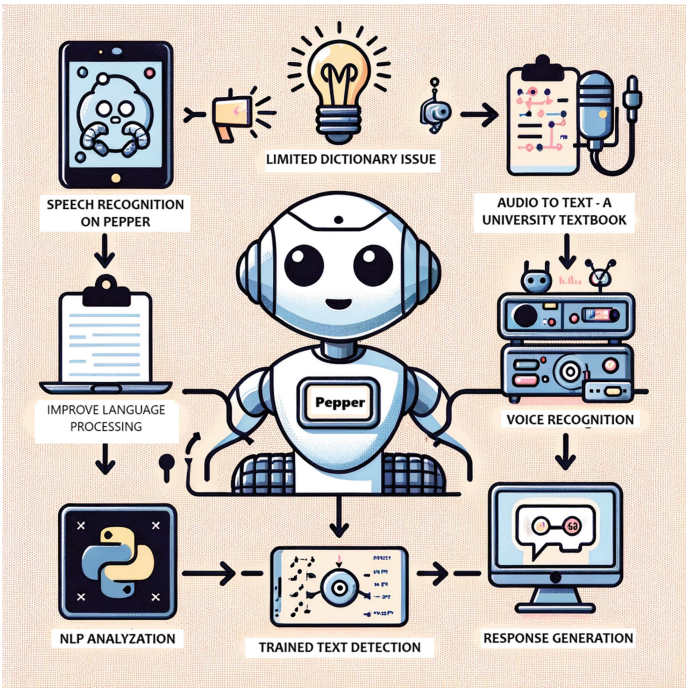

Fig. 1. System diagram

data vector, enabling the model to learn more effectively to extract text from the audio file.

The LLAMA 2 1.7B model was trained on a dataset that includes a PDF file of an academic textbook, such that it will interpret the text extracted from the audio file. The process involves the NLP analysis of the extracted text to identify the user's query. Once it comprehends the question, the NLP generates an appropriate response. This methodology not only tests Pepper's ability to interact with users but also explores the integration of advanced NLP techniques in enhancing robot-human communication (Fig. 1).

3.1 Advanced Library Integration for Improved Robot-Human Interaction

This research explored the integration of cloud-edge technology with SoftBank Robotics' Pepper robot, utilizing a suite of Python libraries like 'langchain', 'pypdf', 'unstructured', `'sentence_transformers'`, `'pinecone-client'`, and `'huggingface_hub'`. These tools were pivotal in augmenting Pepper's functionalities, particularly in processing complex text data, crucial for effective interaction with university students and visitors.

3.2 Customizing Pepper for Interactive Settings

The authors harnessed the 'PyPDFLoader'module to equip Pepper with the capability to parse and analyze text from PDF documents, such as university guidelines and protocols. This feature is of major importance for providing up to date, accurate information to the university community. It sets the foundation for Pepper's advanced comprehension and interpretation abilities relevant to a university context.

3.3 Context-Aware Data Processing Modules

Modules like "RecursiveCharacterTextSplitter" and 'HuggingFaceEmbeddings' were implemented, to enable Pepper to process and understand text in a detailed and nuanced manner. These modules were crucial for handling various tasks, such as responding to student inquiries, assisting in document completion, and navigating information requests.

3.4 Enhanced Interaction Through Advanced NLP Techniques

A significant improvement in Pepper's natural language processing (NLP) capabilities was obtained by incorporating transformer-based models from "HuggingFaceHub" and "SentenceTransformer". This enhancement was essential for tasks like semantic search and automated question answering, facilitating natural and relevant interactions between Pepper and its users.

3.5 Developing a Reliable Question Answering Framework

A sophisticated question-answering system was integrated, evaluated using metrics like `'accuracy_score'`, `precision_score`, `'recall_score'`, `'f1_score'`. This system was instrumental in allowing Pepper to accurately address queries regarding university services, ensuring the dissemination of trustworthy and prompt information.

4 Results

Regarding text classification and machine learning model evaluation, F1 score, precision, and similarity are important metrics, each offering a unique perspective on model accuracy. The F1 score, in particular, is a composite measure that accounts for both precision and recall. Precision refers to the proportion of positive predictions that are correctly identified, while recall is about the proportion of actual positive predictions that the model accurately identifies. The F1 score is calculated as the harmonic mean of precision and recall, with a higher score indicating greater model precision and recall.

The similarity score, on the other hand, assesses the similarities in two text parts. It can be computed using various metrics like cosine similarity, Jaccard

similarity, or Levenshtein distance. A higher similarity score suggests greater resemblance between the two text segments.

The F1 score is particularly useful when both precision and recall are vital. Precision is the preferred metric when correctly identifying positive predictions and it is more important than avoiding incorrect negative predictions. For instance, in loan approval models, a high precision score implies a lower likelihood of approving loans to those likely to default.

In a parallel approach, the similarity score is valuable for evaluating text classification models, aiding in assessing the model's ability to accurately identify similar text segments. For example, in product classification, a high similarity score suggests the model is more likely to correctly group products with similar features.

The Fig. 2 serves to demonstrate the method of assessing similarity between two distinct lists of character strings, named `'supposed_answers'` and `'ai_generated_answers'`. This assessment is executed through the formation of a new list, entitled `'similarity.scores'`. Each entry in this list is assigned a binary value, either 1 or 0. Specifically, a value of 1 is allocated if the corresponding elements in both lists match exactly, indicating a perfect correlation. Conversely, a value of 0 is assigned if the elements differ, signaling a lack of similarity.

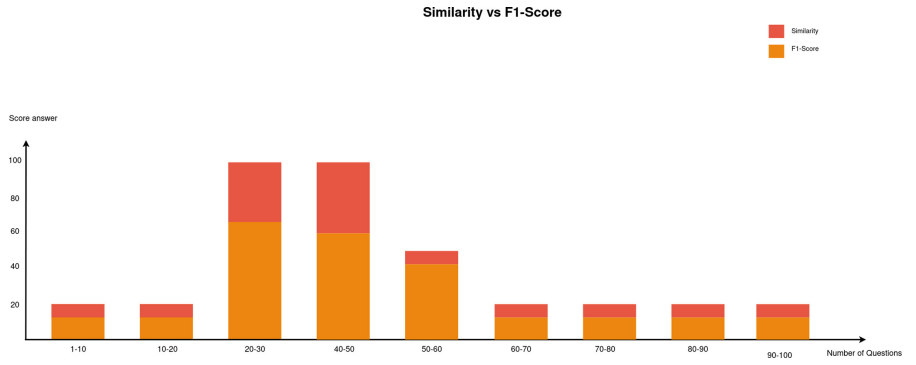

Fig. 2. Similarity assessing method

This binary scoring system is necessary in revealing certain patterns in the model's performance. It becomes apparent that while the model may occasionally fail to produce answers that closely mirror the supposed answers, it is capable of generating true positive responses. This implies that the model can provide accurate and relevant information on specific subjects, even if it does not always align perfectly with the anticipated answers.

To illustrate this point, a specific example is depicted in 3.

Although the answer provided by the robot is succint it touches upon an aspect of the University Senate's role in policy-making process, but without the depth and breadth of the complete answer.

Answer vs Prediction

	Question	Answer provided by the book	Answer provided by the robot
Human interface	What role does the UVT Senate play in shaping university policies?	The UVT Senate is pivotal in shaping university policies. It holds significant responsibilities such as approving academic regulations, endorsing strategic plans, and making key institutional decisions. This role ensures that the university is governed and directed effectively, maintaining its overall vision and mission.	Promoting institutional communication, endorsing strategic plans, and making key institutional decisions

Fig. 3. Answer vs Prediction

This example highlights the AI's capability to grasp the essence of a topic and provide relevant, if not comprehensive, responses. Its performance is reflective of its training on a dataset that comprises various segments or chunks of information. This structure allows the AI to access and relay pertinent information, even if it doesn't always achieve the level of detail or specificity found in the ideal answers (Fig. 4).

The F1 score, as mentioned earlier, reflects the authors' findings with values between 0.02 up to 0.10 in some cases. This indicates that the model, trained on a chunk of a PDF document, accurately generates responses 40% of the time as showed in Fig. 5. This suggests a need for more refined fine-tuning.

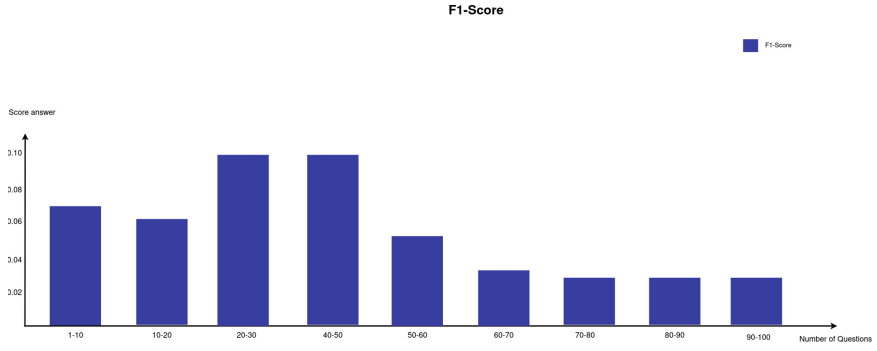

Fig. 4. Precision Score comparison

Pinecone was also utilized as a dynamic data vector, enabling the model to learn more effectively. The benefits of this approach include:

1. Model Suitability for Limited Hardware: Given that the version of Pepper trained for this research is a recent one, with limited processing power, the model provides responses via the cloud, offering somewhat relevant answers based on the results. However, the accuracy of these responses can not be guaranteed.

2. Enhanced Autonomy for Pepper: By integrating a LLMA model, Pepper gains assistance with information processing, though this does not involve

physical movement. Taking into consideration that the authors integrated the language model on a computational model, the robot does not have to compute the answer by itself. Making a request is needed in order to give the answer.

3. Optimization for Simplicity: The first node is specifically designed to optimize Pepper's behavior, considering its limited ability to comprehend complex sentences. Taking into account that the robot has a limited dictionary and understanding of the words it was trained in such a manner that improvements audio-to-text extraction were observed.

Precision and similarity are fundamentally the same concept, yet precision incorporates an additional aspect: it examines the grammar of each sentence. In the current study, a method of lexical similarity was employed, involving a comparison in order to identify the words check if they can be found in the primary dataset. Precision aids this process by also verifying the positional arrangement of the words to determine if they are placed in correct ordered and if the spelling is in accordance with the dictionary. Furthermore, the gathered data, as illustrated in Fig. 3, indicates that in cases of similarity, the response is also grammatically correct.

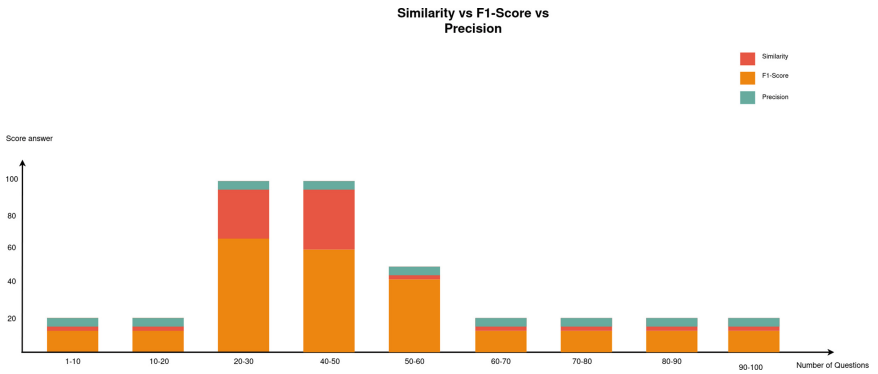

Fig. 5. Comparison between Similarity, F1-Score and Precision score

5 Conclusions and Future Work

5.1 Conclusions

The performed study demonstrates an approach of optimizing AI learning based on LLAMA 2, which is a NLP, and it is focused on learning chunks of information from a document. This method proves to be efficient enough to use the gathered information such that the response could be used as guidance to individuals who require assistance.

Implementing Cloud-Edge and problem segmentation techniques conducted to a better response time in terms of latency. If the entire model was implemented

to run on the Pepper's robot hardware system, it would be nearly impossible to achieve such a rapid response. A significant step forward in this study is that success was obtained by using Cloud-Edge technology for responses, enhancing the interactivity of the social robot.

In conclusion, the tests that were conducted have shown that the Pepper robot can be effectively used as a guide demonstrating a considerable ability to accurately respond to questions by offering semantic and lexical correct answers. Given that the pretrained model on data chunks is capable of addressing specific types of questions will enable the robot to guide users by providing certain information.

Furthermore, after integrating and enhancing the speech recognition from a limited dictionary to a much broader but slightly slower one it is significant that the social robot will understand and will be able to interact with more people, not responding to just a limited set of words.

5.2 Future Work

Next steps in this research will be focused on enhancing the interactivity by enabling Natural Language Processing (NLP) commands for the Pepper social robot. This would involve instructing the robot with data associated to the locations of different rooms and guiding it efficiently through the requested direction. Additionally, another goal is to optimize and upgrade the hardware component. Identifying the most suitable variant will improve the training of the model or even replace the current model entirely to achieve superior responses from alternative NLP models.

Starting from the current results, the authors' intention is to perform additional evaluations on typical edge devices, such as integrating Cloud-Edge computing on an Raspberry Pi 4 or NodeMCU ESP32 microcontroller in order to assess their response times.

Acknowledgement. This article was partially supported by the UVT 1000 Develop Fund of the West University of Timişoara.

References

1. Wan, J., Tang, S., Yan, H., Li, D., Wang, S., Vasilakos, A.V.: Current status and open issues: cloud robotics. IEEE Access **4**, 2797–2807 (2016)
2. Pandey, A.K., Gelin, R., Robot, A.: Pepper: the first machine of its kind. IEEE Robot. Autom. Mag. **25**(3), 40–48 (2018)
3. Touvron, H., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
4. Cao, K., Liu, Y., Meng, G., Sun, Q.: An overview on edge computing research. IEEE Access **8**, 85714–85728 (2020)
5. Shi, W., Cao, J., Zhang, Q., Li, Y., Lanyu, X.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
6. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)

7. Elfaki, A.O., et al.: Revolutionizing social robotics: a cloud-based framework for enhancing the intelligence and autonomy of social robots. Robotics **12**(2), 48 (2023)
8. Wan, S., Zonghua, G., Ni, Q.: Cognitive computing and wireless communications on the edge for healthcare service robots. Comput. Commun. **149**, 99–106 (2020)
9. Queralta, J.P., Qingqing, L., Zou, Z., Westerlund, T.: Enhancing autonomy with blockchain and multi-access edge computing in distributed robotic systems. In: 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), pp. 180–187. IEEE (2020)
10. Biswas, A., Wang, H.-C.: Autonomous vehicles enabled by the integration of IoT, edge intelligence, 5G, and blockchain. Sensors **23**(4), 1963 (2023)
11. Nasir, M., Muhammad, K., Ullah, A., Ahmad, J., Baik, S.W., Sajjad, M.: Enabling automation and edge intelligence over resource constraint IoT devices for smart home. Neurocomputing **491**, 494–506 (2022)
12. Dr Subarna Shakya: Survey on cloud based robotics architecture, challenges and applications. J. Ubiquitous Comput. Commun. Technol. **2**(1), 10–18 (2020)
13. Morales, E.F., Murrieta-Cid, R., Becerra, I., Esquivel-Basaldua, M.A.: A survey on deep learning and deep reinforcement learning in robotics with a tutorial on deep reinforcement learning. Intell. Serv. Robot. **14**(5), 773–805 (2021)
14. Ayala, A., Cruz, F., Campos, D., Rubio, R., Fernandes, B., Dazeley, R.: A comparison of humanoid robot simulators: a quantitative approach. In: 2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), pp. 1–6. IEEE (2020)
15. BigScience Workshop, Le Scao, T., et al.: Bloom: a 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100 (2022)
16. Pande, A., Mishra, D.: The synergy between a humanoid robot and whisper: bridging a gap in education. Electronics **12**(19), 3995 (2023)