A Novel User-Friendly Pipeline for Enhanced Natural Language Understanding in Human-Robot Interaction

Dorin Clisu[1,†], Iulia Farcas[1,*,†], Andrei Rusu[1,*,†] and Mihai Hulea[2,*,†]

[1] *NTT DATA Romania*
[2] *Technical University of Cluj-Napoca, Romania*

Abstract

This paper presents an innovative Natural Language Understanding (NLU) pipeline for human-robot interactions (HRI), optimized for on-premises deployment in industrial settings. The proposed system integrates an end-to-end Automated Speech Recognition (ASR) system, a transformer-based model for intent and entity recognition, and a dynamic dialogue management system. These components operate on commodity hardware, ensuring real-time responsiveness without cloud dependency. The pipeline is uniquely extensible via an automated, offline training module that uses large language models like ChatGPT to generate datasets, reducing the need for specialized machine learning expertise.

Keywords

speech-to-text, text-to-speech, transformers, human-machine collaboration

1. Introduction

Human-robot interaction is becoming increasingly critical in industrial settings where efficiency, accuracy, and adaptability are paramount [1]. As industries shift towards greater automation, the demand for more intuitive and natural communication methods between humans and robots grows. NLU serves as a vital tool in bridging this gap, allowing robots to interpret and respond to human language [2]. However, many existing NLU systems depend on cloud-based services [3], which can introduce unwanted latency and security risks—issues particularly problematic in industrial environments.

This paper introduces a specialized NLU pipeline designed for human-robot interaction within industrial settings and ASR system [4], a transformer-based model for intent and entity recognition, and a dynamic dialogue management system. These components are optimized to function on commodity hardware, forming a robust and scalable solution for real-time HRI.

RuleML+RR'24: Companion Proceedings of the 8th International Joint Conference on Rules and Reasoning, September 16–22, 2024, Bucharest, Romania

* Corresponding author.
† These authors contributed equally.
dorin.clisu@nttdata.com (D. Clisu); iulia.farcas@nttdata.com (I. Farcas); rusu.andrei@nttdata.com (A. Rusu); mihai.hulea@aut.utcluj.ro (M. Hulea)

2. Solution architecture

To develop a robust and efficient NLU pipeline for human-robot interaction, we have employed a state-of-the-art technology stack: Python with deep learning libraries PyTorch and Transformers for working with BERT [5], OpenAI Whisper [6] for ASR and Mozilla TTS [7] for Text-to-Speech, Pydantic for data validation, FastAPI and Streamlit for creating user-friendly interfaces. These components are used to create the training and inference pipelines and are detailed in Figure 1 below.

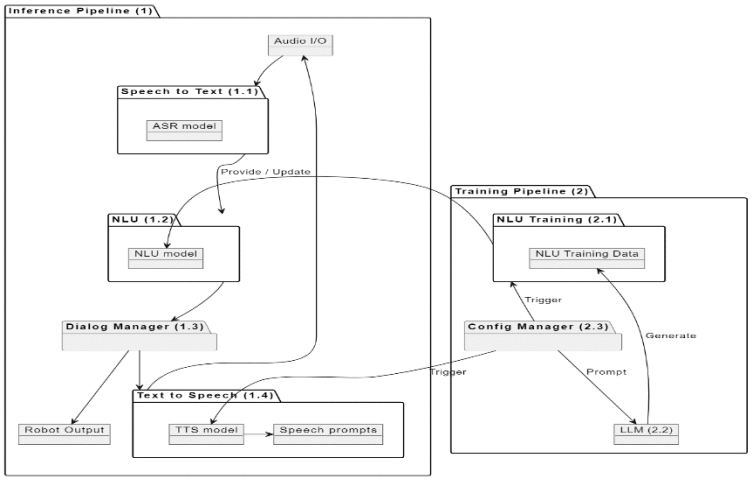

Figure 1 – High level architecture of the proposed system.

2.1. Inference pipeline

The Inference Pipeline is designed for on-edge deployment, enabling real-time processing of vocal commands with a target maximum end-to-end delay of two seconds. Key components include: Audio I/O: Captures and outputs audio, serving as the interface for human-robot communication; Speech to Text (ASR): Converts spoken language into text, optimized for minimal latency; NLU: Extracts intents and entities from the transcribed text to understand and execute commands; Dialog Manager: Manages conversational context and directs the flow of interactions; Text to Speech: Converts text responses back into speech, allowing for seamless communication with operators; Robot Output: Executes the understood commands, affecting robot actions directly. This component is out-of-scope.

2.2. Training pipeline

The Training Pipeline effectively trains the ML models, enables new voice commands and is configurable to operate either via a cloud-based LLM API or via an open-source model like Llama-3-8b on a powerful standalone computing station. This flexibility ensures an optimum balance between performance, reliability and security. Main components are: Config Manager: Provides technical user interface for updating commands, creates data models with validation and LLM prompt according to the commands structure; LLM: Generates annotated data in the form of natural utterances expressing the commands,

according to the prompt.; NLU Training: Trains the NLU model using the generated data. The hyperparameters are fixed during the research phase, so that users get a usable model without any ML engineer intervention.

2.3. Speech to Text

After studying the state-of-the art (as of 2024) in ASR models, we chose Whisper from OpenAI as it has very good accuracy with a real-time factor > 1 on commodity hardware. Additionally, it comes in multiple model sizes for an optimum compromise between accuracy and resource consumption depending on application.

The fundamental limitation of Whisper is that it can only transcribe a pre-recorded audio clip. Because a robot needs to continuously listen for commands, we tried some workarounds and found Silero VAD (voice activity detection) model to be satisfactory.

2.4. NLU

Transformer based language models have revolutionized NLP in the last few years [8], from language translation, sentiment analysis, knowledge extraction to complex text generation. Despite being relatively old (2018), the BERT language model remains a solid workhorse for many NLP tasks because it has a low inference cost. One uses BERT as a pre-trained text encoder which captures an abstract understanding of the language in its hidden state vectors. Then, according to the task, one trains a relatively small neural network on top of the BERT encodings, requiring a similarly small amount of task-specific data. In our case of intent and slot recognition, we can use one sentence classifier head for the intent, and another token-level classifier head for the slot identification. Then for each slot that must fit to a pre-defined list of values we can run the extracted tokens through a zero-shot classifier leveraging vector similarity.

2.5. Text to Speech

After studying the state-of-the art (as of 2024) in TTS models, it resulted that cloud API's offer more than enough speech quality, low latency and very low price for this application (OpenAI, Google, AWS, etc.). Since most utterances can be generated as part of the training flow, there is not much reliance on the internet for regular operation. However, in some cases such as mentioning the run-time slot options, we need on-the-fly generation therefore we investigated some open-source models that can run locally, of which Mozilla TTS seems promising. If the latency of generating locally is higher than the cloud latency, then the cloud API will be used primarily, with an automatic fallback (when offline) to the local model.

2.6. Dialog Manager

To have a coherent system, a core module is needed to stitch together the AI functionalities of the individual modules. A rule-based implementation is developed, covering the following fixed scenarios with appropriate templates applicable to any command that is later added: Too much audio noise: When the recognized speech is unintelligible, ask the operator to repeat or simply ignore and keep listening; Unclear intent: When the speech is

recognized but cannot be classified as one of the existing intents, ask the operator to revise and repeat (optionally informing them about the list of intents); Missing slot: When the intent is clear but one of the mandatory slots for this intent is missing, ask the operator to say the respective slot, or repeat the entire utterance if there is more than one missing slot (to be able to separate which slot is which); Unclear slot option: When everything is clear except an invalid slot option (ex. A, B, C are the options, but they provide D), ask the operator to revise and say the slot (optionally informing them about the list of values).

The dialog manager uses a local database and memory variables to keep track of the conversation. It also logs all successes and failures (tied to the user input and context) so that the system can be analyzed, improved and updated.

3. Conclusions

In this paper, we have presented a novel Natural Language Understanding (NLU) pipeline designed for human-robot interaction in industrial settings.

Future work will focus on further enhancing the system's capabilities, including improving the accuracy and latency of the ASR and TTS components, expanding the range of supported commands, and integrating more advanced dialogue management features. Additionally, we aim to conduct extensive field testing in various industrial settings to refine the system and ensure its reliability and effectiveness in real-world applications.

4. Acknowledgements

This work was supported by the European Union's Horizon Europe research and innovation programme under grant agreement No 101058589 (AI-PRISM).

References

[1] "Human-Robot Interaction Review: Challenges and Solutions for Modern Industrial Environments | IEEE Journals & Magazine | IEEE Xplore." Accessed: Jul. 15, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9493209

[2] R. Bamdale, S. Sahay, and V. Khandekar, "Natural Human Robot Interaction Using Artificial Intelligence: A Survey," in 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), Mar. 2019, pp. 297–302. doi: 10.1109/IEMECONX.2019.8877044.

[3] S. Pais, J. Cordeiro, and M. L. Jamil, "NLP-based platform as a service: a brief review," J Big Data, vol. 9, no. 1, p. 54, Apr. 2022, doi: 10.1186/s40537-022-00603-5.

[4] D. Yu and L. Deng, Automatic Speech Recognition: A Deep Learning Approach. in Signals and Communication Technology. London: Springer, 2015. doi: 10.1007/978-1-4471-5779-3.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv, May 24, 2019. doi: 10.48550/arXiv.1810.04805.

[6] "Introducing Whisper." Accessed: Jul. 15, 2024. [Online]. Available: https://openai.com/index/whisper/

[7] “mozilla/TTS.” Mozilla, Jul. 15, 2024. Accessed: Jul. 15, 2024. [Online]. Available: https://github.com/mozilla/TTS
[8] B. Min et al., “Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey,” ACM Comput. Surv., Sep. 2023, doi: 10.1145/3605943.