ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2024) 000–000

www.elsevier.com/locate/procedia

57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)

Integrating Large Language Model for Natural Language-Based Instruction toward Robust Human-Robot Collaboration

Fanru Gao[a], Jianjing Zhang[a], Liqiao Xia[a], Robert X. Gao[a,*]

[a]*Department of Mechanical and Aerospace Enginerring, Case Western Reserve University, Cleveland, OH, 44106, USA*

* Corresponding author. Tel.: +1-216-368-6045; fax: +1-216-368-6445. *E-mail address:* robert.gao@case.edu

Abstract

Human-Robot Collaboration (HRC) aims to create environments where robots can understand workspace dynamics and actively assist humans in operations, with the human intention recognition being fundamental to efficient and safe task fulfilment. Language-based control and communication is a natural and convenient way to convey human intentions. However, traditional language models require instructions to be articulated following a rigid, predefined syntax, which can be unnatural, inefficient, and prone to errors. This paper investigates the reasoning abilities that emerged from the recent advancement of Large Language Models (LLMs) to overcome these limitations, allowing for human instructions to be used to enhance human-robot communication. For this purpose, a generic GPT 3.5 model has been fine-tuned to interpret and translate varied human instructions into essential attributes, such as task relevancy and tools and/or parts required for the task. These attributes are then fused with perceived on-going robot action to generate a sequence of relevant actions. The developed technique is evaluated in a case study where robots initially misinterpreted human actions and picked up wrong tools and parts for assembly. It is shown that the fine-tuned LLM can effectively identify corrective actions across a diverse range of instructional human inputs, thereby enhancing the robustness of human-robot collaborative assembly for smart manufacturing.

Keywords: Human-robot collaboration; Natural language processing; Error correction; Large language models

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2024) 000–000

www.elsevier.com/locate/procedia

57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)

Integrating Large Language Model for Natural Language-Based Instruction toward Robust Human-Robot Collaboration

Fanru Gao[a], Liqiao Xia[a], Jianjing Zhang[a], Sichao Liu[b], Lihui Wang[b], Robert X. Gao[a,*]

[a]*Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, 44106, USA*
[b]*Department of Production Engineering, KTH Royal Institute of Technology, Stockholm, 10044, Sweden*

* *Corresponding author. Tel.: +1-216-368-6045; fax: +1-216-368-6445. E-mail address: robert.gao@case.edu*

Abstract

Human-Robot Collaboration (HRC) aims to create environments where robots can understand workspace dynamics and actively assist humans in operations, with the human intention recognition being fundamental to efficient and safe task fulfillment. Language-based control and communication is a natural and convenient way to convey human intentions. However, traditional language models require instructions to be articulated following a rigid, predefined syntax, which can be unnatural, inefficient, and prone to errors. This paper investigates the reasoning abilities that emerged from the recent advancement of Large Language Models (LLMs) to overcome these limitations, allowing for human instructions to be used to enhance human-robot communication. For this purpose, a generic GPT 3.5 model has been fine-tuned to interpret and translate varied human instructions into essential attributes, such as task relevancy and tools and/or parts required for the task. These attributes are then fused with perceived on-going robot action to generate a sequence of relevant actions. The developed technique is evaluated in a case study where robots initially misinterpreted human actions and picked up wrong tools and parts for assembly. It is shown that the fine-tuned LLM can effectively identify corrective actions across a diverse range of instructional human inputs, thereby enhancing the robustness of human-robot collaborative assembly for smart manufacturing.

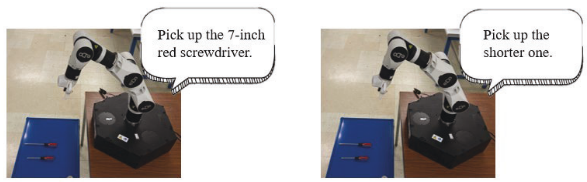
Keywords: Human-robot collaboration; Natural language processing; Error correction; Large language model

1. Introduction

Human-Robot Collaboration (HRC) focuses on the physical and cognitive interactions between humans and robots working towards common objectives. In HRC communication, a cognitive model is commonly employed to collect data from both the environment and human operators, before transforming it into commands for robot management [1]. Advances in sensor technologies and machine learning, such as Convolutional Neural Network (CNN), has enhanced robots' abilities to process image data, recognize human actions, and predict future activities, thereby facilitating cooperative tasks [2]. However, the unpredictable nature of human behavior and potential errors in AI model predictions require an adaptive mechanism to correct errors and accurately interpret human intentions.

Natural human language offers a promising solution for correcting robotic predictions, similar to human-to-human interactions. Unlike brainwaves or gesture-based control, language-based communication provides a natural, real-time, and convenient method for error correction that is independent of ongoing actions. However, ambiguity and polysemy that are inherent to human language pose challenges for robots, as conventional language control protocols require precise and rigid syntactic structures. These protocols necessitate exact descriptions of the objects for a robot to act on, such as the length of a screwdriver, instead of only saying "shorter". As depicted in Fig. 1, a command using rigid syntax provides detailed quantitative information (Fig. 1. (a)), whereas a command presented in natural human language may lack specific details (Fig. 1. (b)).

In the situation described above, language understanding is essential in HRC, where the analysis of grammatical structures has emerged as a promising method to pave the way. For instance, categorizing language into noun and verb clauses facilitates the transformation of flexible natural language into structured statement expressions [3]. A limitation of the grammatical approach is its ineffectiveness when processing terms not seen during training. The advent of Large Language Models (LLMs) offers enhanced understanding and reasoning capabilities infused with common sense, enabling them to interpret meanings without extensive manual annotations. Over the past few years, LLMs have gained attention across various fields, from healthcare to finance and manufacturing [4].



(a) Rigid, Predefined Syntax Command (b) Human Language Command

Fig. 1. Predefined command and human natural language

Despite these advantages, integrating LLMs into HRC poses unique challenges. The main issue is content awareness: robots must understand both the workplace environment and workflow conditions to act properly. Since LLMs were originally designed for general use, adapting them to the diverse conditions of the HRC domain is complex. Additionally, the need for deterministic responses is critical, as variability in LLMs can result in discrepancies during interpretation and application, potentially leading to unintended robotic actions.

To address these challenges, an HRC-adapted LLM has been developed by fine-tuning the GPT-3.5 model from OpenAI with an HRC-based dataset. This process converts human language input into a configuration for robot manipulation, which is subsequently integrated with ongoing robot actions through a control module, ultimately producing executable commands for the robot to correct errors. Contributions from this study include:

(1) Established an HRC-based data framework to guide the generation of natural language for specific HRC tasks, synthesizing varying completion levels of communication. The language model is utilized to refine the LLM, bridging the gap between general applications and the specifics for HRC.

(2) Proposed a language control method that corrects prediction errors by fine-tuned LLM with a specifically prepared language model. This enhanced LLM is integrated into a robotic communication module, improving cooperation efficiency between the robot and human operators (expert or non-expert) through natural language commands.

(3) Developed an error correction control module that combines current robot state with fine-tuned LLM's outputs to generate a sequence of corrective actions, enabling the robot to correct tool selection errors, and then pick up the correct tool.

2. Related work

2.1. Development of language models

Natural language serves as the predominant mode of communication among humans. The objective of Natural Language Processing (NLP) is to transform human language into symbolic forms comprehensible to computers, or to render machine language into formats understandable by humans. Bengio *et al*. proposed the first neural language model, which was implemented using a feedforward neural network, generating a probability distribution of a word [5]. Advancing this field, in 2013, Mikolov *et al*. proposed Word2vec, which enables large-scale word-embedding to become feasible [6]. In 2017, Vaswani *et al.* developed a groundbreaking neural network called transformer, which has demonstrated significantly improved performance in parallel computing and long-term dependencies [7]. Building on the success, Devlin *et al*. proposed bidirectional encoder representations from transformers (BERT) in 2018, which was able to generate deep bidirectional language representations and showed excellent advantages in 11 NLP tasks [8]. Simultaneously, the Generative Pre-trained Transformer (GPT from OpenAI) was proposed, efficiently encoding contextual relationships and exhibiting superior reasoning abilities [9,10].

One limitation of the general GPT is that it may not adequately recognize the specific environmental settings in manufacturing such as HRC, or the varied linguistic practices of operators. This underscores the need for domain-specific LLMs to capture and interpret nuances accurately.

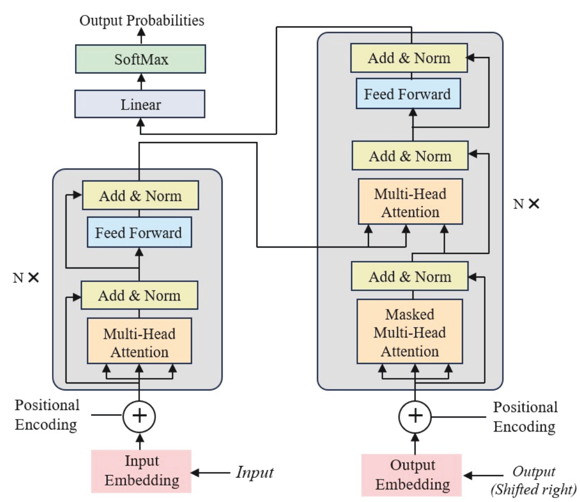


Fig. 2. Working process of Transformer model

2.2. Human intention Recognition in HRC

One of the principal research interests in HRC is the investigation of collaboration processes between humans and robots [4]. Unlike humans, who can easily perceive changes in their environment and status, robots fundamentally lack this flexible perceptual capability that is critical to collaborative interactions. Generally, there are three main methods for robots to perceive and understand human intentions: gesture control [11], brainwave control [12], and language control [13].

Among these, language control stands out as it is not affected by the current human actions, thereby less vulnerable to noise in model predictions. Park *et al*. contributed to this area by developing a framework that improves HRC on construction sites through the integration of Bidirectional Long Short-Term Memory with Conditional Random Field (BiLSTM-CRF) and BERT models. This framework uses the BiLSTM-CRF to capture contextual information in sequence labeling, while employing the BERT model for language understanding [13]. Additionally, the development of LLM has enabled more effective communication between humans and robots using language, allowing robots to process environmental information through verbal commands [14].

On the other hand, language control in HRC has been performed via rigid, predefined syntax, making it difficult for robots to interpret instructions naturally. This limitation poses a challenge and restricts its broader application.

3. Methodology

3.1. Fine-tuned language model

LLM has shown significant potential in communication like answering questions, text summarization, and translation, due to the pre-trained model technologies and the Transformer architecture [7]. Through pre-training, LLMs develop a common sense that boosts performance across various downstream tasks [15]. This capability enables the effective integration of LLMs into HRC through task-specific fine-tuning. The outcome of such adaptation is enhanced interaction and comprehension between humans and robots, leading to increased collaborative efficiency and functionality.

The fine-tuned LLM leverages the transformer architecture, introduced by Vaswani *et al.* [7]. As shown in Fig. 2, the Transformer model comprises two main components: an encoder and a decoder. The encoder processes the input sequence and includes six identical layers, each has two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. The decoder also contains six identical layers but adds a masked multi-head attention layer, ensuring that the outputs depend only on previous sequence elements. This structure enables the transformer-based model to generate outputs through a series of computations detailed in subsequent sections. The output of the transformer-based model can be computed as follows:

$$a_0 = UW_e + W_p \tag{1}$$

where $\mathbf{W_e}$ is the token embedding matrix and $\mathbf{W_p}$ is the position embedding matrix. $\mathbf{U}$ includes the input sequence to the transformer-based model. The working process of transformer-based model involves the generation of the Query ($\mathbf{Q}$), Key ($\mathbf{K}$), and Value ($\mathbf{V}$) matrix for each token in the input:

$$Q = W_Q * a_{l-1}, K = W_K * a_{l-1} \quad V = W_V * a_{l-1}, \tag{2}$$

where $\mathbf{W_Q}$, $\mathbf{W_K}$, $\mathbf{W_V}$ are the weight matrices specific to each head in the multi-head attention mechanism, and $\mathbf{a_{l-1}}$ is the previous output from the attention mechanism of the corresponding input. Calculating the similarity by comparing the Query and the Key of other tokens through the dot product:

$$similarity = Q \cdot K^T \tag{3}$$

where T denotes the transpose of $\mathbf{K}$ matrix, the *attention_score* using *SoftMax* function with the output from Eq. 3 yields:

$$attention\_socre = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right), \tag{4}$$

where d_K represents the dimension of $\mathbf{K}$. The output of the attention mechanism $\mathbf{Y_{l-1}}$ can be computed by multiplying *attention_ score* with $\mathbf{V}$, where $\mathbf{V}$ is the size of the vocabulary:

$$Y_{l-1} = attention\_score * V \tag{5}$$

In addition to the attention sub-layers, the Transformer model includes a position-wise fully connected feed-forward network, which comprises two linear transformations with a *ReLU* activation function applied in between. The term $\mathbf{Y_{l-1}}$ is fed through a separate feed-forward neural network with its own parameter $\mathbf{W_2}$ and $\mathbf{b_2}$. The final output $\mathbf{a_l}$ of the feed-forward network is:

$$a_l = ReLu(Y_{l-1}) * W_2 + b_2, \tag{6}$$

and the parameters of the network can be optimized with the cross-entropy loss function as follows:

$$L = -\sum_{i=1}^{N} Y_i \log(\check{Y}_i) \tag{7}$$

where Y_i is the true token embeddings and $\check{Y}_i$ is the predicted one, N represents token numbers.

3.2. HRC natural language command instruction framework

To train the model for interpreting natural language instructions with missing information, an HRC-based command framework has been developed. Because human instruction may not contain all the critical details, robots must resort to additional environmental context to properly and correctly respond to the human instruction. Contemporary cognitive theories emphasize that the structured nature of human language, rather than its versatility, is its most significant attribute [16]. These theories posit that human linguistic behavior adheres to grammatical principles, especially the Subject-Verb-Object (SVO) word order [16,17]. In this presented study, the language commands have been structured into components of Subject, Object, and Context, as illustrated in Fig. 3. Notably, the Verb is integrated within the Context component to align closely with the specific demands of the HRC scenario.

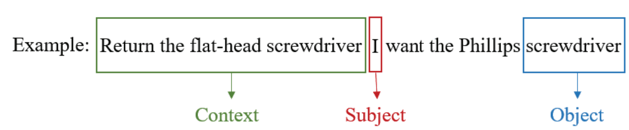


Fig. 3. Syntax structure of language instruction

Understanding the Subject is essential for a LLM to correctly predict robotic actions. Humans naturally use environmental and contextual cues to interpret instructions, which is a capability often lacking in general LLM.

Contextual understanding is critical for an LLM to choose the appropriate action among multiple possibilities. Without this, robots may face ambiguities, unsure whether to hand over a tool to the operator or return it to the toolbox.

Additionally, accurate identification of the Object in a command is crucial. The Object typically represents the tool required by human operators, underscoring the need for clear and precise communication to ensure successful HRC.

To make the instructions abstract and natural for practical communication in HRC, the natural language commands were synthesized by minimizing the explicit content, specifically by omitting the Subjects, Objects, or Contexts, as illustrated in Table 1, to mimic the conversational style in natural human communication. Such an approach, however, poses significant challenges for robots in discerning human intentions due to the ambiguity of the commands.

3.3. Control Module

The Control Module's primary function is to receive and process the output from the fine-tuned LLM, converting it to a robotic manipulation configuration. Additionally, the module records the robot's current state, such as the actions being executed (e.g., picking up or returning) and the characteristics of the tool currently held by the robot. By integrating the LLM's output with the robot's current state, the Control Module effectively infers the human operator's actual intentions, generating the final configuration that includes the expected robot actions, tool names, and features.

Table 1. Natural language command after removing corresponding part

Removed component	Command	Challenges
Initial command	Return the flat-head screwdriver, I want the Phillips screwdriver.	N/A
Subject	Not the flat-head screwdriver, the Phillips screwdriver.	What is the desired action?
Object	Return the flat-head one, I want the Phillips one.	What is the tool human wants?
Context	I want the Phillips screwdriver.	What to do with the tool in hand?
Subject and Object	Not the flat-head one, the Phillips one.	What is the tool human wants and the desired action?
Subject and Context	The Phillips screwdriver.	What is the desired action and what to do with the tool in hand?
Object and Context	I want the Phillips one.	What is the tool human wants and what to do with the tool in hand?
Subject, Object and Context	The Phillips one.	What to do first? What to do next and what is the tool human actually wants?

As shown in Fig. 4, the current state stores the robot's activity status, and the Control Module can determine the ongoing action, such as picking up a blue Phillips screwdriver, by reading this state.

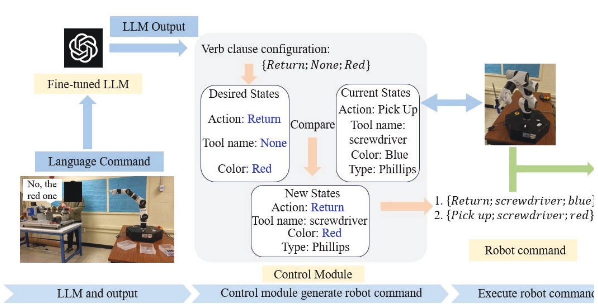


Fig. 4. Workflow of the control module

When an instruction lacks specific information, such as the tool name expected by the human operator, the Control Module plays the role of facilitating the correct understanding and interpretation of the human instruction. By analyzing the working status output from the LLM, the module identifies discrepancies in the human language command and identify errors in the robot's current operational state. Further integration of the robot's current status indicates that the human command did not correct the name of the tool, but rather its attribute, such as the color. Consequently, the Control Module synthesizes the data from the LLM with the robot's operational status, conducts a comparison, and adjusts the status accordingly. This process results in a finalized sequence of robotic commands: (1) return the currently held tool; (2) retrieve and deliver the tool as specified by the human operator.

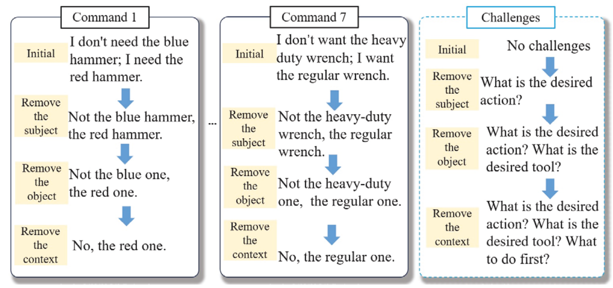


Fig. 5. Training data examples

4. Case study

To implement the developed method for HRC, the GPT-3.5-1106 model was fine-tuned using the HRC natural language command instruction framework, including evaluation of an unseen test dataset. Subsequently, an experiment on an Edo robot was platformed to evaluate the effectiveness of using unstandardized, natural language commands for correcting robotic errors.

4.1. Dataset

To facilitate the evaluation, a fine-tuned dataset that includes ambiguous but natural human language commands was paired with expected robotic configurations. Initially, a total of seven instructions were generated using rigid syntax with complete information to train the LLM to produce outputs accurately. Recognizing that a tool can possess multiple attributes, such as color and shape, the instructions also encompass these features, allowing the LLM to learn the diverse ways humans might describe tools. This approach aligns with the developed HRC natural language command framework, from which multiple sets of natural human language instructions were generated.

As illustrated in Fig. 5, the standard training dataset comprises 28 command samples, including tool names such as hammer and wrench, and color features such as red and blue, with additional characteristics such as heavy-duty and regular. Moreover, to examine the impact of the size of the dataset size on model generalization, an advanced training set with 41 instructions was generated, following the same criteria. For the testing set, different tool names and features from the training set were selected to ensure credible evaluation. It is noted that the order of reduction in the dataset is not fixed. The intended output from this dataset is structured as a dictionary containing four keys: Input status, Working status, Tool name, and Tool feature. Input status is binary (0 or 1), indicating whether the natural language instruction is relevant to the robot's task (1 for yes, 0 for no). Working status reflects the accuracy of the tool selection by the robot based on the human's instructions, with 1 denoting correct and 0 incorrect. Tool name and Tool feature provide details about the tool specified by the human operator, with Tool feature detailing the characteristics of the tool as described in the human command.

4.2. Fine-tuned LLM performance

The GPT-3.5-turbo-1106 model, which has 175 billion parameters, was selected for the presented study due to its optimal balance of performance and computational efficiency. For the fine-tuned process, the standard training set was utilized, with the learning rate multiplier set to 2. A cross-entropy loss function was used. As depicted in Fig. 6, the loss value decreased significantly from 3.9532 to 0.2481 after 50 training steps, demonstrating the model's ability to accurately learn from the training samples and generate the expected output.

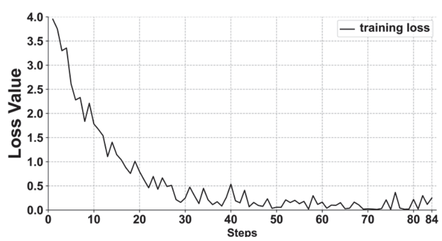


Fig. 6. Loss value during the training

To evaluate the natural language understanding ability of the proposed model, comparative evaluations with other versions of GPT-3.5 were performed, including GPT-3.5-turbo (not fine-tuned), GPT-3.5-0613, and GPT-3.5-0125, all trained on the same dataset. Model performance was measured on the test dataset using accuracy metrics: human intention accuracy, tool name accuracy, and tool feature accuracy, based on the output's Working status, Tool name, and Tool feature. These metrics evaluate how well the model identifies human intentions and the specific tool attributes mentioned in the commands. The accuracy is calculated as:

$$accuracy = \frac{Number\ of\ test\ cases\ predicted\ correctly}{Total\ number\ of\ test\ cases} \quad (8)$$

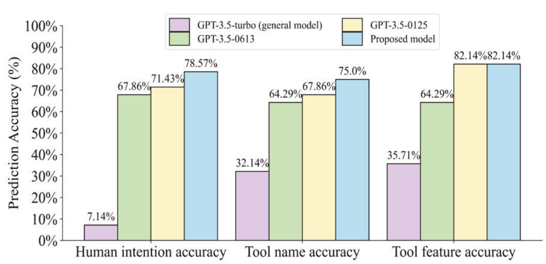


Fig. 7. Comparison of prediction accuracy with other models

Fig. 7 illustrates the prediction accuracy of these models. In each group, from left to right, the first, second, and third models constitute the control group, with the fourth model being fine-tuned proposed model. Compared to the control group, the proposed model has shown comparatively the best performance in discerning the expected tool name and features from instructions, particularly those with a flexible format and missing information. This enhanced understanding underscores the improved model responsiveness to complex, real-world commands in HRC.

Comparative analysis was also performed of performance of the fine-tuned model when trained on a standard training set versus an advanced training set, to examine the impact of training sample volume on the model's generalization ability. As depicted in Fig. 8, within each group, the left bar represents the model trained on the standard training set, and the right bar corresponds to the model trained on the advanced training set. While the model trained on the standard training set was capable of making reasonable predictions and producing the expected output format, the model trained on the advanced training set exhibited enhanced accuracy. This improvement demonstrates that the volume of training samples significantly influences model performance, with a larger number of samples leading to better generalization.

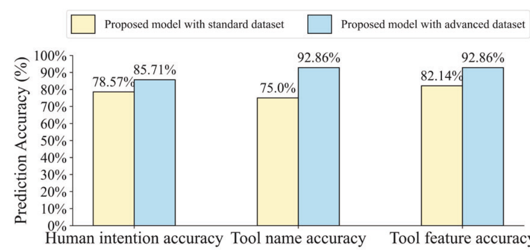


Fig. 8. Comparison of prediction accuracy between different data volume

The model's ability to interpret new input formats where human operators describe tools using multiple features was also evaluated. For example, if a robot delivers a wrong tool, e.g., a red flat head instead of a blue Phillips screwdriver, the operator might specify both color and shape to correct the error and help the robot to distinguish the desired tool between a red Phillips and a blue Phillips screwdriver in the toolbox. These features were added in the test dataset to assess the model's accuracy in understanding such detailed commands. As shown in Fig. 9, while all models can generate outputs in a fixed format, the developed model trained on advanced training set achieved higher accuracy.

Within each group, the first two models are control group

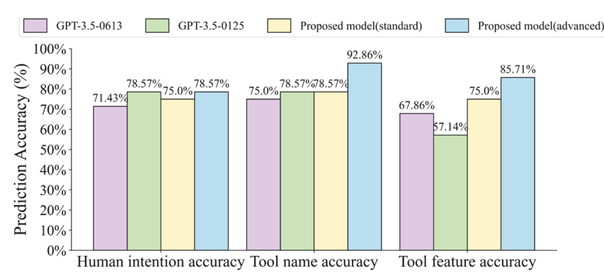


Fig. 9. Comparison of prediction accuracy with multiple features

models trained on the standard set, and the two on the right are the developed models trained on both the standard and advanced sets. Results show that all the models can interpret adjectives due to pre-training and fine-tuning has enhanced their ability to link adjectives with the correct tool features, improving understanding of complex commands.

4.3. Real-robot experiment

To evaluate the performance of the developed model, it was integrated into an Edo robot platform for assisting human operators in assembly tasks. The system employs a camera to capture images and predict the next step in the assembly process, enabling the robot to provide the required tool to the operator [18]. When prediction errors occur, operators can use natural language commands to correct the robot's action. This correction mechanism, illustrated in Fig. 10, involves six steps: ① capturing the operator's current action to predict the next action; ② robot retrieving the predicted tool; ③ operator identifying and correcting tool selection errors; ④ robot returning the incorrect tool (e.g., a screwdriver) to where it was picked up; ⑤ robot retrieving the correct tool (e.g., an Allen key); ⑥ robot delivering the right tool to the operator.

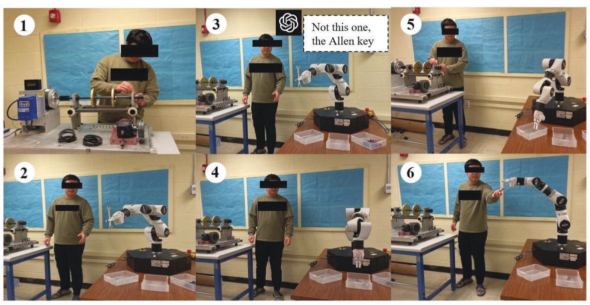


Fig. 10. Edo robot platform-based tool selection error correctness

5. Conclusion

To address the challenge posed by rigid and predefined syntax that is typically required for robots to understand human language input and facilitate the development of efficient HRC communication, a fine-tuned LLM trained on a tailored HRC communication model was developed, which equips the LLM with task-specific knowledge and engineering behaviors. Experimental results have demonstrated that the fine-tuned LLM model can accurately interpret human intentions from incomplete language commands, enabling the robot to execute effectively action sequences and correct erroneous movements.

Future research will expand the training dataset with different human language behaviors, and explore hint-based voice feedback and explanatory mechanisms to enhance the intuitiveness and transparency of human-robot communication and interaction towards more effective, efficient, and safer operations on the factory floors.

Acknowledgements

The authors gratefully acknowledge support from the US National Science Foundation under awards CMMI-1830295 and EEC-2133630 (Engineering Research Center, HAMMER), and the Swedish Vetenskapsrådet under award 2023-00493.

References

[1] F. Semeraro, A. Griffiths, and A. Cangelosi, "Human–robot collaboration and machine learning: A systematic review of recent research," Robotics and Computer-Integrated Manufacturing, vol. 79, p. 102432, 2023.

[2] S. Li, P. Zheng, S. Liu, Z. Wang, X. V. Wang, L. Zheng, and L. Wang, "Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives," Robotics and Computer-Integrated Manufacturing, vol. 81, p. 102510, 2023.

[3] R. Gupta, P. Pande, I. Herzog, J. Weisberger, J. Chao, K. Chaiyasate, and E. S. Lee, "Application of ChatGPT in cosmetic plastic surgery: ally or antagonist?," Aesthetic Surgery Journal, vol. 43, no. 7, pp. NP587–NP590, 2023.

[4] D. K. Misra, J. Sung, K. Lee, and A. Saxena, "Tell me dave: Context-sensitive grounding of natural language to manipulation instructions," The International Journal of Robotics Research, vol. 35, no. 1-3, pp. 281–300, 2016.

[5] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," Advances in neural information processing systems, vol. 13, 2000.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[9] Y. Qiu and Y. Jin, "ChatGPT and finetuned bert: A comparative study for developing intelligent design support systems," Intelligent Systems with Applications, vol. 21, p. 200308, 2024.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal,A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-shot learners," Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.

[11] R. X. Gao, L. Wang, P. Wang, J. Zhang, and H. Liu, "Human motion recognition and prediction for robot control," in Advanced Human-Robot Collaboration in Manufacturing, pp. 261–282, Springer, 2021.

[12] L. Wang, S. Liu, C. Cooper, X. V. Wang, and R. X. Gao, "Function block-based human-robot collaborative assembly driven by brainwaves," CIRP Annals, vol. 70, no. 1, pp. 5–8, 2021.

[13] S. Park, X. Wang, C. C. Menassa, V. R. Kamat, and J. Y. Chai, "Natural language instructions for intuitive human interaction with robotic assistants in field construction work," Automation in Construction, vol. 161, p. 105345, 2024.

[14] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2998–3009, 2023.

[15] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., "Improving language understanding by generative pre-training," 2018.

[16] G. Gazdar, Generalized phrase structure grammar. Harvard University Press, 1985.

[17] Y. Grodzinsky, "Language deficits and the theory of syntax," Brain and language, vol. 27, no. 1, pp. 135–159, 1986.

[18] S. Liu, J. Zhang, L. Wang, and R. X. Gao, "Vision AI-based human-robot collaborative assembly driven by autonomous robots," CIRP Annals, vol. 73, no. 1, 2024