Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 16, No. 2, 2026

Research and Implementation of a Human-Computer Interaction AI Intelligent Robot Based on Speech

Weihang Mu *

Department of Computer Science, School of Computing and Engineering, University of Huddersfield, Huddersfield, H1 3DH, UK
*** Corresponding author Email:** U2389766@unimail.hud.ac.uk

Abstract: With the advancement of deep learning, language model training, and edge computing, speech-based human-robot interaction (HRI) systems are expanding their applications in manufacturing and medical escort service robots. In this paper, a modular speech-driven human-robot interaction system is proposed and implemented, which integrates speech recognition (ASR), natural language understanding (NLU), dialogue management, and action execution modules. To evaluate the engineering trade-offs of different technology options, three sets of comparative trials were designed and implemented: ASR (cloud-based commercial vs. local models), NLU (traditional statistical methods vs. Transformer fine-tuning), and dialogue management (rule-driven vs. reinforcement learning). Simulation experiments were conducted in the ROS/Gazebo environment. Subjective tests were also performed, with evaluation metrics including word error rate (WER), accuracy, F1-score, response latency, task completion rate, and user satisfaction. The results show that the Transformer-based NLU is significantly better than the traditional methods in semantic parsing. ASR in the cloud has obvious advantages in recognition quality, but the local model is more suitable for real-time control scenarios in terms of delay and privacy. Dialogue management is recommended to adopt a hybrid strategy of "rule first + reinforcement learning enhancement." Finally, the problems of system engineering deployment, model compression, edge-cloud collaboration, and ethical compliance were discussed.

Keywords: Human-computer Interaction; Speech Recognition; Natural Language Understanding; Dialogue Management; Robot Control.

1. Introduction

The quality of human-computer interaction is the key to the widespread adoption of intelligent robots. In recent years, end-to-end speech recognition, pre-trained language models, and multimodal perception technologies have matured, providing a new path for robots to understand natural language and make stable responses in dynamic environments [1]. In practical engineering, however, the system still faces challenges such as low latency, real-time performance, robustness against noise (including dialect), and the tradeoff between interpretability and security. This paper focuses on engineering reproducibility and evaluability. It proposes a modular system and, through comparative experiments on different technology selections, aims to provide a practical design and evaluation template for subsequent academic research and industrial deployment.

2. Related Work

Speech recognition (ASR): From traditional acoustic models and language models to end-to-end Conformer, RNN-T and Transducer models, especially the advantages of Wav2Vec2 and Conformer in low-resource migration studies [2].

Natural language understanding (NLU): Intention recognition and slot extraction are the core of task-oriented dialogue systems; in recent years, joint models based on BERT/RoBERTa have shown significant effects in Chinese scenes [3].

Dialogue management (DM): Rule-driven solutions have strong interpretability and high reliability; reinforcement learning (RL) has potential in policy optimization, but training costs and safety constraints need additional treatment [4].

Multimodal fusion: Vision-language alignment can reduce instruction ambiguity, which is particularly important in "pointing to objects + verbal instructions" interaction tasks [5].

3. System Design and Overall Architecture

3.1. Overall Architecture

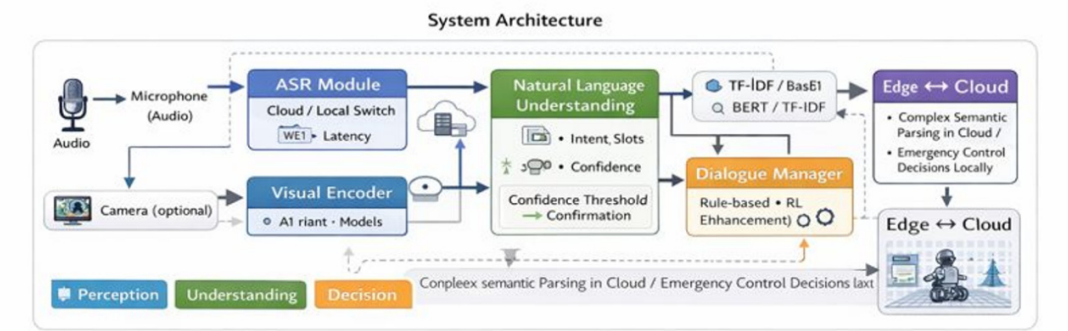

Figure 1. Overall system architecture of the speech-based HRI robot

The system is divided into the following modules:

Sound acquisition: microphone array + voice activity detection

Speech recognition (ASR): supports cloud/local deployment switching

Natural language understanding (NLU): intent classification + slot filling (joint learning)

Dialogue management (DM): rule-driven + reinforcement learning enhanced (triggering confirmation based on confidence threshold)

Action planning and control: global planning (A*) + local obstacle avoidance (dynamic window method) + low-level control (PID/model predictive control)

Logging and monitoring: recording raw audio, ASR text, NLU confidence, performed actions and results

Figure 1 illustrates the proposed modular system architecture for a spoken dialogue system operating in edge-cloud collaborative environments. The system comprises four core modules: an **Automatic Speech Recognition (ASR) module** with a configurable cloud/local switch to balance latency and accuracy; a **Visual Encoder** for processing multimodal inputs; a **Natural Language Understanding (NLU) unit** responsible for intent detection and slot filling, outputting confidence scores; and a **Dialogue Manager** that combines rule-based policies with reinforcement learning enhancements. A key feature of the architecture is the dynamic edge-cloud collaboration: complex semantic parsing is offloaded to the cloud, while time-sensitive emergency control decisions are executed locally at the edge. The workflow follows a **Perception–Understanding–Decision** pipeline, with a confidence-based confirmation mechanism triggered when NLU confidence falls below a predefined threshold.

3.2. Design Points

Low latency: ASR supports streaming decoding, small frame length (20 ms), and guarantees first packet delay below about 200 ms (locally).

Confidence mechanism: NLU outputs confidence; when it falls below a threshold, it triggers confirmation or degradation of the service.

Edge-cloud collaboration: Emergency control decisions are placed locally, while complex semantic parsing is handled by the cloud or the near-end server [6].

Security constraints: Multi-factor confirmation (such as voice + gesture or key press confirmation) is enabled for critical motion commands [7].

4. Algorithm Implementation

4.1. Speech Acquisition and ASR (Example)

Input: Audio stream, language settings (e.g., zh-CN)

Output: Recognized text or empty string

Steps:

Initialize the speech recognition engine (supports streaming decoding, frame length 20 ms).

Start the microphone and set the sampling rate to 16000 Hz.

Loop:

a. Detect voice activity (VAD).

b. If speech is detected, start recording audio for up to 5 seconds.

c. Call the recognition engine (cloud/local model can be switched).

d. Return the recognition result text.

If recognition fails (e.g., no voice or network error):

a. Return an empty string.

b. Log errors.

Close.

4.2. Simple NLU (Vectorization + Classifier Example)

Input: User text statement

Output: Intent labels

Training phase:

Collect annotated corpus (text + intent labels).

Extract TF-IDF features.

Train a naive Bayes classifier.

Save the model (vec, clf).

Prediction phase:

Load the TF-IDF vectorizer and classifier.

Input text → TF-IDF feature vectors.

Classifier predicts intent labels.

Return the label.

4.3. Dialogue Management (Rule Engine Example)

Input: Intent label, slot information, confidence

Output: Response statement and perform the corresponding action

Steps:

Initialize the robot controller.

If confidence < 0.7:

a. Return "Please confirm the instruction."

Otherwise:

a. Perform actions based on intention:

If intention is "forward" → robot.forward().

If intention is "stop" → robot.stop().

Other intentions → Return "not identified".

b. Return execution results feedback statement.

4.4. Robot Control (Abstract)

Input: Action commands (e.g., "go forward", "stop")

Output: Robot state updates

Steps:

Initialize robot state to "idle".

If "go" command received:

a. Set status to "moving forward".

b. Execute forward control commands (e.g., issuing speed commands).

If "stop" command received:

a. Set status to "stopped".

b. Send stop signal.

Keep status change log.

5. Experiment Design

5.1. Experiment Objectives

Compare performance of different ASR deployments (cloud vs. local) in terms of word error rate, latency, and impact on downstream NLU.

Compare traditional NLU and Transformer models in intent recognition and slot extraction.

Compare rule-driven and reinforcement learning-driven dialogue management in task completion rate and interaction efficiency.

5.2. Data and Subjects

Simulation experiment: Robot (navigation task, object manipulation task) controlled by synthesized/recorded audio in ROS + Gazebo with at least 500 instruction samples for each configuration.

Subjective assessment: 20 participants (including dialects and different age groups) completed a subjective questionnaire (Likert 1-5), each completing 10 interactions in imperative and natural language. Studies like highlight the importance of diverse user groups in HRI evaluation [8].

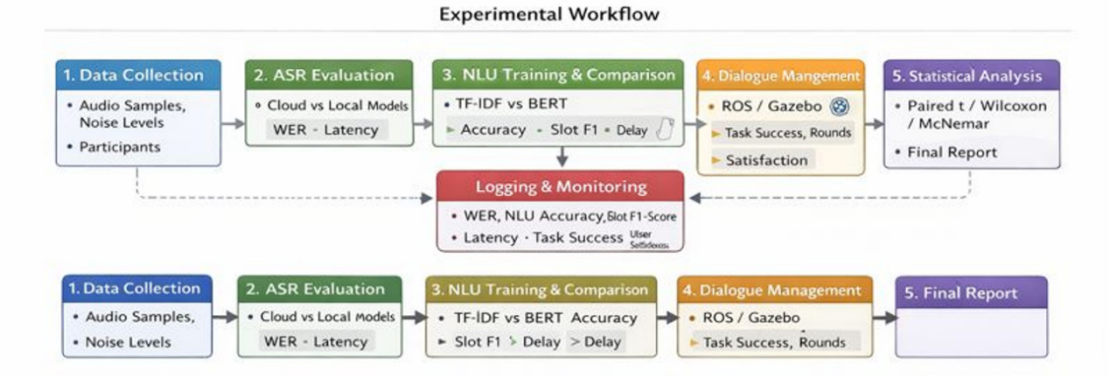

Figure 2. Workflow of the experimental evaluation process.

Figure 2 outlines the end-to-end experimental workflow designed to evaluate the system's performance. The process consists of five sequential stages:

Data Collection: Gathering audio samples under varying noise levels with participant involvement. **ASR Evaluation**: Comparing cloud-based and local ASR models in terms of Word Error Rate (WER) and latency.**NLU Training & Comparison**: Training and evaluating NLU models (TF-IDF vs. BERT) on accuracy, slot F1-score, and inference delay. **Dialogue Management Testing**: Deploying the system in a simulated environment (ROS/Gazebo) to measure task success rate and dialogue round efficiency.**Statistical Analysis & Reporting**: Applying paired t-tests, Wilcoxon signed-rank tests, and McNemar's test to validate results, culminating in a final performance report.

5.3. Noise Setting

Three levels of environmental noise: silent (0-30 dB), indoor (~45 dB), and noisy (~65 dB). For each noise level, system performance degradation was assessed, considering challenges noted in distant interaction scenarios [9].

5.4. Indicators and Statistical Methods

Objective indicators: Word error rate (WER), intent recognition accuracy, slot F1-score, average response latency (ms), mission success rate.

Subjective indicators: User satisfaction, ease of use, trustworthiness.

Statistical test: Paired t-test or Wilcoxon test (non-normal) for continuous variables; McNemar test or Bootstrap method for accuracy differences. Report p-value and effect size (Cohen's d), significance level $\alpha = 0.05$.

6. Comparative Tests

6.1. Experiment A: ASR Comparison (Cloud vs. Local)

Interpretation: Cloud ASR has a slight advantage in word error rate, but higher latency; scenarios with high real-time and privacy requirements are more inclined to local models or edge-cloud hybrid deployment, aligning with discussions on deployment trade-offs [10].

Table 1. ASR Comparison (Sample Data)

Noise Level	Google ASR WER (%)	Wav2Vec2 WER (%)	Google Latency (ms)	Wav2Vec2 Latency (ms)
Silence	4.8	5.6	320	110
Indoor (~45 dB)	7.6	9.1	340	125
Noisy (~65 dB)	12.4	15.2	380	150

6.2. Experiment B: NLU Comparison (TF-IDF + NB vs. BERT Fine-tuning)

Table 2. NLU Contrast (Sample Data)

Model	Accuracy (%)	Slot F1(%)	Inference Delay (ms)
TF-IDF + NB	86.4	78.9	6.2
BERT Fine-tuning	93.1	91.2	48.7

Interpretation: BERT model significantly improves semantic understanding, but inference delay and resource occupation increase, suitable for edge-cloud collaboration or offline complex reasoning, as explored in the context of large language models for robotics [11].

6.3. Experiment C: Dialogue Management (Rule-driven vs. Reinforcement Learning)

Table 3. DM Comparison (Example Data)

Strategy	Task Completion Rate (%)	Average Number of Rounds	User Satisfaction (1-5)
Rule-driven	92.1	2.15	3.9
RL (DQN)	95.3	1.82	4.1

Reading: Reinforcement learning improves efficiency and

success rate, but needs to address training sample and policy constraint issues; actual deployment suggests incremental optimization with rules prioritized and reinforcement learning enhancement, a strategy discussed in guidelines for LLM use in HRI [12].

7. Result Analysis and Error Sources

ASR error sources: Noise, accent/dialect, speech rate variation, speech endpoint detection failure. Improvements include data augmentation (noise mixing), multi-microphone beamforming, and adaptive acoustic models, also noting specific challenges with child speech [2].

NLU error sources: Syntactic diversity and multi-intention scenes. Robustness can be improved by adding annotated data with slots, applying sequence labeling + conditional random fields, and using context windows, benefitting from advances in LLMs [3, 11].

Dialogue management risk: Reinforcement learning easily affected by distribution deviation causing "unexplainable behavior." Protective measures include fallback strategy constraints and rules, sandbox simulation, and manual audit processes, as emphasized in responsible reporting guidelines [4, 12].

Engineering suggestions: Model compression (distillation/quantization), ONNX/TensorRT acceleration, and edge-cloud collaborative deployment framework (emergency control local, complex semantics cloud) are feasible paths.

8. System Engineering, Privacy and Security

Model acceleration: Knowledge distillation, INT8 quantization, pruning, and ONNX conversion; can be used for BERT family TinyBERT/DistilBERT [13].

Privacy compliance: Voice and data encryption, minimize data storage, user consent and anonymity; deployment subject to regional regulations (e.g., user authorization, data deletion mechanisms, etc.).

Security policy: Enable confirmation mechanisms for critical commands (dual-mode confirmation); record audit logs (timestamps, recognized text, confidence, performed actions), and maintain a rollback mechanism.

9. Conclusion and Future Work

This paper proposes and implements a speech-based human-robot interaction system, and through comparative experiments systematically assessed the trade-offs of three groups of different combinations in accuracy, latency, and user experience.

Main conclusions: Transformer-based NLU is superior in semantic understanding; cloud ASR has advantages in pure recognition quality, but latency and privacy limitations favor local models in some scenarios; dialogue management recommends a rule-first + reinforcement learning enhancement strategy.

Future work directions: (1) Large-scale online testing in real scenes and collecting multi-source voice and data (dialect, children's voices); (2) Implementing multimodal fusion (visual + voice) to reduce command ambiguity; (3) Research on miniaturized model deployment and online adaptive learning mechanisms; (4) Improving ethics and privacy compliance frameworks.

References

[1] Wang, J., Wu, Z., Li, Y., et al. (2024). Large language models for robotics: Opportunities, challenges, and perspectives. *arXiv, 2401.04334*.

[2] Janssens, R., Verhelst, E., Abbo, G.A., et al. (2024). Child speech recognition in human-robot interaction: Problem solved? *arXiv, 2404.17394*.

[3] Zeng, F., Gan, W., Huai, Z., et al. (2023). Large language models for robotics: A survey. *arXiv, 2311.07226*.

[4] Matuszek, C., Williams, T., DePalma, N., et al. (2025). Reporting guidelines for large language models in human-robot interaction. *ACM Transactions on Human-Robot Interaction, 15: 1-24.*

[5] Wang, T., Zheng, P., Li, S., & Wang, L. (2024). Multimodal human-robot interaction for human-centric smart manufacturing: A survey. *Advanced Intelligent Systems, 6*(3), 2300359.

[6] Garcia, R., Mahu, R., Grageda, N., et al. (2024). Speech emotion recognition with deep learning beamforming on a distant human-robot interaction scenario. In *Proc. Interspeech 2024*, 3215–3219.

[7] Gong, T., Chen, D., Wang, G., et al. (2024). Multimodal fusion and human-robot interaction control of an intelligent robot. *Frontiers in Bioengineering and Biotechnology, 12*, 1310247.

[8] Mauliana, M., Ashok, A., Czernochowski, D., & Berns, K. (2025). Exploring LLM-powered multi-session human-robot interactions with university students. *Frontiers in Robotics and AI, 12*, 1585589.

[9] Kim, C.Y., Lee, C.P., & Mutlu, B. (2024). Understanding large-language model (LLM)-powered human-robot interaction. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, 371–380.

[10] Wang, C., Hasler, S., Tanneberg, D., et al. (2024). LaMI: Large language models for multi-modal human-robot interaction. *arXiv, 2401.15174*.

[11] Liu, H., Zhang, Y., Li, C., et al. (2023). Challenges and applications of large language models in robotics control. *Journal of Artificial Intelligence Research, 77*, 145–189.

[12] Amershi, S., Weld, D., Vorvoreanu, M., et al. (2019). Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 3.

[13] Jiao, X., Yin, Y., Shang, L., et al. (2020). TinyBERT: Distilling BERT for natural language understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 4163–4174.