ISSN: 2249-7196
IJMRR/March-May 2024/ Volume 14/Issue 3/186-193
G. Deepthi / International Journal of Management Research & Review

Building a Social Interactive Robot with Speech Recognition and a Custom Chatbot

D. Sreenivasa Rao[1], J. Sujatha[2], G. Deepthi[3], Ch. Pavani[4], J. Srinivas[5]

[1,2]Assistant Professor, Department of Electronics and Communication Engineering Avanthi Institute of Engineering & Technology, Tamaram, Makavarapalem, Narsipatnam, Anakapalli District – 531113, Andhra Pradesh, India

[3,4,5]UG Student, Department of Electronics and Communication Engineering Avanthi Institute of Engineering & Technology, Tamaram, Makavarapalem, Narsipatnam, Anakapalli District – 531113, Andhra Pradesh, India

Abstract

This paper describes a social interactive robot that combines speech recognition with a custom-built chatbot to support natural spoken interaction between a user and a machine. Speech recognition algorithms interpret spoken commands and queries, forming the input path of the system. Rather than using a generic conversational agent, a chatbot was built specifically for this robot, trained on an intent dataset defined for the tasks the robot is expected to handle, so that its responses stay within a scope it can actually serve. A hotword detector listens continuously for a trigger phrase and activates the recognition pipeline only when that phrase is heard, which keeps the robot from responding to ambient conversation and avoids running the full pipeline continuously on limited hardware. The robot is built on a Raspberry Pi running ROS, with an ESP32 for peripheral control, a microphone and 5 V speakers for audio, a Raspberry Pi camera, an HC-SR04 ultrasonic sensor for obstacle detection, a DHT11 for temperature and humidity, DC and SG90 servo motors driven through an L298N motor driver, and a touch-enabled TFT LCD. The chatbot is implemented in Python using NLTK for language processing and a TFLearn neural network trained on a JSON intent file, with Google Text-to-Speech generating the spoken reply. The mechanical design was produced in Fusion 360.

Index Terms— *Chatbot, hotword detection, human-robot interaction, natural language processing, NLTK, Raspberry Pi, ROS, social robot, speech recognition, text-to-speech.*

I. Introduction

Social interactive robots sit at the intersection of robotics and human-computer interaction. The aim is to give a machine the ability to take part in natural, human-like social interaction, which is useful across healthcare, education and customer service. Advances in artificial intelligence and hardware design have made it possible to build robots that understand and respond to social cues in real time. Alongside the technical work, ethical questions about privacy, autonomy and the nature of human relationships with machines deserve attention.

A social robot is one capable of interacting with people and with other robots in a natural and interpersonal manner. It is equipped with sensors, cameras and microphones so that it can respond to touch, sound and visual input in a way

that resembles how a person would. Robots already perform basic and repetitive tasks more efficiently and accurately than humans, which suits them to manufacturing; adding artificial intelligence extends that to situations complex enough to require interpretation rather than repetition.

A. Objectives and Scope

The objective is to build a robotic system that engages in natural interaction with a user, providing assistance and support through spoken conversation. The scope covers the design, development and deployment of a robot with sensory capability — camera, microphone and touch input — that allows it to perceive speech and interpret user input, and to adapt its responses to the context of the interaction.

B. Applications

Education. The robot can act as a tutor, giving personalized assistance through speech recognition and chatbot features, helping students practise language skills, giving feedback on pronunciation and running interactive learning activities.

Healthcare. In a care setting the robot can provide companionship, assist with cognitive therapy exercises, remind patients about medication and offer support through conversation.

Customer service. In retail and hospitality the robot can serve as a virtual assistant, answering enquiries and giving product information, with the chatbot responding from a predefined knowledge base.

Language learning. The robot can give learners conversational practice, correct pronunciation and explain grammar, with the chatbot supplying targeted exercises.

II. Literature Survey

Mitra et al. [1] present a neural network model that estimates articulatory trajectories from speech signals, trained on synthetic speech generated by a task-dynamic model of speech production. The trained model was applied to natural speech, and the estimated trajectories were used together with standard cepstral features to train acoustic models for large-vocabulary recognition. Results on two English datasets showed that articulatory information improves recognition performance not only under clean conditions but also against noisy backgrounds, with the best results obtained when articulatory, cepstral and perceptually motivated features were combined. Most earlier work using articulatory information had concentrated on estimating it from speech rather than using it for recognition, and had been confined to digit or medium-vocabulary tasks.

Deng and Sun [2] propose a statistical approach to automatic speech recognition based on atomic speech units constructed from overlapping articulatory features. Speech sounds are represented by atomic units derived from these overlapping features, treated as the smallest units of speech and corresponding to basic articulatory gestures. A recognition framework built on statistical modelling of these units decodes the speech signal and recognizes words or phrases from the sequence of atomic units detected. The contribution is a different perspective on how speech is represented and modelled.

Augello et al. [3] introduce the Humorist Bot, a chatbot designed to bring computational humour into human-computer interaction. Recognizing the role humour plays in human communication, the authors built an agent that both generates humorous sentences and recognizes humorous expressions introduced by the user during dialogue. It

uses established computational humour techniques, is implemented on the ALICE framework embedded in a messaging client, and includes an avatar whose facial expression changes according to the humorous content of the dialogue. The relevance here is the demonstration that a conversational agent benefits from responses shaped to the character of the interaction rather than to information content alone.

III. Hardware

The robot is built around a Raspberry Pi as the central control unit, which interfaces with the motors, sensors and actuators and runs the recognition and chatbot software. Programming is done in Python, and the board's wireless capability supports remote access. The components used are described below.

Raspberry Pi. The central controller, running the operating system, the ROS nodes, the speech recognition pipeline and the chatbot. It coordinates the robot's movement by interfacing with the motors and sensors.

ESP32. A microcontroller providing wireless connectivity, low power consumption and real-time processing. It handles sensor integration and motor control, taking the timing-sensitive work off the Raspberry Pi, and its compatibility with IoT platforms supports remote monitoring.

Raspberry Pi camera. Gives the robot vision, supporting object recognition and face recognition, which the chatbot uses when greeting a person by name.

Microphone. Captures the user's speech, which is the input to both the hotword detector and the recognition pipeline. It also lets the robot detect other sounds in its environment.

5 V speakers. Deliver the synthesized spoken reply, along with any sound effects or audio content.

DC motors and L298N motor driver. Provide movement. The driver's dual H-bridge gives independent bidirectional control of two motors with PWM speed control, and includes protection features that prevent damage to the motors.

SG90 servo motors. Give precise positional control for the robot's articulated movements, in a compact and low-power package.

HC-SR04 ultrasonic sensor. Measures distance without contact, allowing obstacle detection and avoidance while the robot moves.

DHT11 sensor. Measures temperature and humidity. These readings are exposed to the chatbot, so that a user asking about conditions in the room receives a live value rather than a canned reply.

8 × 8 dot matrix display. Gives visual feedback on the robot's status and supports simple expressive output during interaction.

3.5-inch touch TFT LCD. Provides a touch interface as an alternative to speech, with virtual buttons, sensor readouts and system status.

Li-ion battery. Supplies power, chosen for high energy density, compact size and stable voltage output across the discharge cycle.

IV. Software Tools

The Raspberry Pi runs Raspberry Pi OS, installed from an image written to a microSD card and then configured for the robot's peripherals. On top of this the system uses ROS for inter-component communication, with the speech recognition, chatbot and sensor nodes each running separately and exchanging messages through topics and services.

NLTK, the Natural Language Toolkit, provides the language processing: tokenization, stemming and the text handling that precedes classification. TFLearn and TensorFlow provide the neural network used to classify user input into intents. NumPy handles the numeric work. Google Text-to-Speech synthesizes the spoken reply from the chatbot's text output, and supports voice selection and multiple languages.

JSON is used as the data format for the intent definitions and for structured exchange within the system. Keeping intents in a JSON file rather than in code means the robot's conversational scope can be extended by editing data and retraining, without changing the program.

The mechanical design was produced in Fusion 360, which was used for 3D modelling of the robot's body and components, for simulating movement before fabrication, and for iterating on the design through successive prototypes.

V. Speech Recognition

The recognition pipeline proceeds through several stages, shown in Fig. 1.

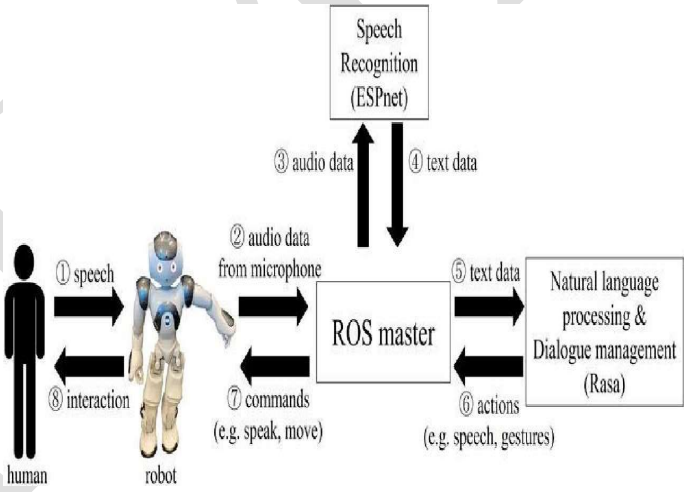

Fig. 1. Block diagram of the speech recognition process.

A. Audio Input

The system captures audio from the microphone. This raw audio is the starting point for everything that follows, so its quality matters: distortion or background noise at this stage degrades every later step. Sampling rate and signal normalization are handled here.

B. Preprocessing

Preprocessing prepares the raw audio for analysis. Background noise is removed to improve signal clarity, volume levels are normalized for consistency, and frequencies irrelevant to speech are filtered out.

C. Feature Extraction

The preprocessed audio is transformed into a form suitable for analysis by extracting features that capture the essential characteristics of the speech signal, particularly its frequency content and spectral shape. Common techniques are the spectrogram, which shows how frequency components vary over time, and Mel-frequency cepstral coefficients (MFCCs), which capture the spectral envelope. These features are the input to the modelling stages.

D. Acoustic Modelling

Acoustic modelling builds statistical models connecting the acoustic features derived from the signal to linguistic units such as phonemes or words. The models learn the probabilistic relationship between observed features and the corresponding units. Hidden Markov Models are effective for the temporal dependencies in sequential data, while deep neural networks capture more complex patterns within it.

E. Language Modelling

Language modelling estimates the probability of word sequences occurring in a language, allowing the system to predict the most likely sequence given the input. N-gram models calculate the probability of a word from its preceding words; neural language models use deep learning to capture longer-range patterns. Trained on large text corpora, these models learn the statistical structure of natural language. This stage is what resolves acoustically similar alternatives, since the acoustic model alone cannot distinguish between phrases that sound alike but differ in plausibility.

F. Post-Processing and Response Generation

Post-processing refines the transcribed text, correcting recognition errors through spell checking, grammar checking and contextual analysis. The corrected text is then used to generate a response, which may be information, an answer to a question, or a command executed by the robot.

A feedback loop collects user corrections and uses them to refine the models over time, so the system adapts to the particular users and environment it operates in.

G. System Flow

In operation, the user speaks and the microphone converts the words into digital audio. The speech recognition module analyses the audio and converts it to text. Natural language processing and dialogue management interpret the text, maintaining context across turns. The recognized text is published to the ROS master, which coordinates commands and data between the components of the system. From there the robot executes actions — speaking, gesturing or moving — according to the commands the text represents.

The recognition node publishes to a topic that the chatbot subscribes to, and the chatbot exposes a voice service that the recognition node calls with the transcribed text. Structuring it this way means the two can be developed and restarted independently, which is the practical benefit of running them as separate ROS nodes rather than as one program.

VI. Hotword Detection

Hotword detection identifies a particular trigger word or phrase that starts an interaction. Machine learning is used to recognize the pattern of the chosen word in spoken audio, so that the user activates the system by speaking it. This

enables hands-free operation, and it also solves a practical problem: without it, the robot would either respond to every conversation within earshot or would have to run the full recognition pipeline continuously, which is costly on a Raspberry Pi. Snowboy was used for this function.

A. Hotword Model

The hotword model is a trained machine learning model that recognizes the predefined trigger word within audio input and initiates further processing when it is detected.

B. Detection Threshold

The detection threshold sets the sensitivity of the system: the minimum confidence required before a detection triggers a response. Adjusting it trades false positives against false negatives. Set too low, the robot wakes to unrelated speech; set too high, the user has to repeat the trigger word.

C. Activation, Context and Response

Activation occurs when a hotword is recognized above the threshold, at which point the system begins further processing. Context establishment then prepares the environment for the interaction that follows, gathering contextual information and initializing the relevant modules. The response phase executes the action associated with the detected trigger, whether that is confirming to the user that the robot is listening or beginning a dialogue.

VII. Custom Chatbot

The chatbot is built rather than adopted, so that its responses match the tasks this robot performs. The workflow covers package installation, creation of the intent dataset, loading that data, defining the chatbot's purpose, training, and use.

A. Intent Dataset

A JSON file, intents.json, holds the patterns and responses that let the chatbot interpret user input. Each intent groups a tag, a set of example phrasings a user might use, and the responses the robot may give. This file is the dataset the model is trained on, and it is also what defines the boundary of what the robot can discuss. Defining the purpose of the chatbot before training matters for the same reason: a narrow, well-specified scope produces reliable behaviour, whereas an open-ended one produces a system that answers everything badly.

B. Training

User input is tokenized with NLTK and stemmed with the Lancaster stemmer, then converted to a bag-of-words vector against the vocabulary built from the training patterns. A feedforward network defined in TFLearn takes this vector as input, passes it through three fully connected layers of sixteen units, and produces a softmax output over the intent labels. Training on the intent data teaches the network to map an input phrase to the intent it belongs to. The trained model and the processed vocabulary are saved so that the robot loads them at startup rather than retraining each time.

C. Interpretation and Response

At run time the chatbot node exposes two ROS services: a text service returning a text reply, and a voice service that publishes the reply for speech synthesis. An incoming query is converted to a bag-of-words vector and passed through the model. The highest-scoring output gives the intent tag, and a response is chosen at random from the responses listed for that tag, so that repeated questions do not produce identical wording.

Some intents are resolved dynamically rather than from fixed text. Where the tag is a temperature or humidity query, a placeholder in the stored response is replaced with the current reading from the DHT11, received on a ROS topic. Where the tag is a greeting, farewell or identification request, the node calls a face recognition service and substitutes the recognized person's name. Where the tag is an object detection request, it calls the detection service and appends the result. This is what separates the robot's chatbot from a purely textual one: several of its answers depend on what its sensors report at the moment it is asked.

VIII. Results

The assembled robot is shown in Fig. 2. The recognition pipeline transcribed spoken input and passed it to the chatbot, which returned a response that was then spoken through the text-to-speech module. Hotword detection worked as intended, keeping the robot idle until addressed by name and then activating the recognition pipeline.

Fig. 2. The assembled social interactive robot.

The chatbot answered queries within the scope of its intent dataset, and the sensor-linked intents returned live values: asked about the temperature, the robot reported the current DHT11 reading rather than a stored figure. Face recognition allowed it to greet a known person by name.

Two limitations were apparent. Recognition accuracy depends on the acoustic conditions, and the modest microphone and the fan noise of the robot itself both work against a clean signal. The chatbot answers only what its intent file covers; a question outside that set returns a fallback response rather than an answer. Both are consequences of the design choices made, and both are addressable — the first by better audio hardware and noise handling, the second by extending the intent dataset.

IX. Conclusion

Combining speech recognition with a purpose-built chatbot lets a social robot understand and respond to spoken input, making interaction more natural than a button or screen interface allows and accessible to users of varying ability. The custom chatbot adds a layer of function beyond recognition alone, since its responses are shaped to the tasks the robot performs and, for several intents, to what its sensors report at the moment of asking.

The approach is applicable across healthcare, education and customer service, where a machine that can be spoken to is more useful than one that must be operated. Ethical questions around privacy, autonomy and societal impact deserve attention as such systems move from prototype to deployment, particularly given that a robot of this kind listens continuously in order to detect its trigger word.

X. Future Scope

Advanced natural language understanding. Sentiment analysis, entity recognition and semantic understanding would let the robot handle queries more complex than the intent classification used here supports.

Multimodal interaction. Adding gesture recognition, facial expression analysis and emotion detection to the speech input would give the robot a fuller picture of user intent.

Personalization and adaptation. Machine learning applied to interaction history could tailor responses to individual users over time.

Domain-specific deployment. Specialized intent sets would suit the robot to particular settings, such as patient care and rehabilitation exercises in healthcare, or tutoring in education.

Integration with IoT and smart environments. Connecting the robot to smart home devices and connected sensors would let it control devices and retrieve information beyond its own sensors.

Collaborative interaction. Communication protocols and coordination strategies would allow several robots to work together with humans in a shared environment.

References

[1] V. Mitra, W. Wang, A. Stolcke, H. Nam, C. Richey, J. Yuan, and M. Liberman, “Articulatory trajectories for large-vocabulary speech recognition,” in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 7145–7149.

[2] L. Deng and D. X. Sun, “A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features,” Journal of the Acoustical Society of America, vol. 95, no. 5, p. 2702, 1994.

[3] A. Augello, G. Saccone, S. Gaglio, and G. Pilato, “Humorist bot: bringing computational humour in a chat-bot system,” in Proc. Int. Conf. Complex, Intelligent and Software Intensive Systems, 2008, pp. 703–708.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.