MedCrave
Step into the World of Research

International Robotics & Automation Journal

Research Article

Open Access

CrossMark

Design and implementation of smart voice assistant and recognizing academic words

Abstract

This paper approaches the use of a Virtual Assistant using neural networks for recognition of commonly used words. The main purpose is to facilitate the users' daily lives by sensing the voice and interpreting it into action. Alice, which is the name of the assistant, is implemented based on four main techniques: Hot word detection, Voice to Text conversion, Intent recognition, and Text to Voice conversion. Linux is the operating system of choice, for developing and running the assistant because it is in the public domain, also, Linux has been implemented on most Single-board computers. Python is chosen as a development language due to its capabilities and compatibility with various APIs and libraries, which are deemed necessary for the project. The virtual assistant will be required to communicate with IoT devices. In addition, a speech recognition system is created in order to recognize the significant technical words. An artificial neural network (ANN) with different structure networks and training algorithms is utilized in conjunction with the Mel Frequency Cepstral Coefficient (MFCC) feature extraction technique to increase the identification rate effectively and find the optimal performance. For training purposes, the Levenberg-Marquardt (LM) and BGFS Quasi-Newton Resilient Backpropagation are compared using 10 MFCC, utilizing from 10 to 50 neurons increasing in increments of 10 similarly for 13MFCC the training is done utilizing from between 10 to 50 neurons.

Keywords: chatbots, IoT devices, MFCC features, neural networks, voice assistant

Volume 8 Issue 1 - 2022

Ahmed J Abougarair,[1] Mohamed KI Aburakhis,[2] Mohamed O Zaroug[1]

[1]Department of Electrical and Electronic Engineering, University of Tripoli, Libya

[2]Department of Engineering Technology, Clark State College, USA

Correspondence: Ahmed J Abougarair, Electrical and Electronic Engineering, University of Tripoli, Tripoli, Libya, Tel +218925385942, Email a.abougarair@uot.edu.ly

Received: October 18, 2021 | **Published:** February 24, 2022

Introduction

Technological advances have made people's lives easier by making ordinary chores such as booking a flight, managing the home temperature, or obtaining instant information simultaneously with other tasks. This assistant saves time by using the oral communication with the computer such as answer queries and manage some operating system commands so as to communicate with peripheral devices. Practically, the user can check the weather, manage IoT devices, and handle various system activities with their voice remotely and with less effort. Software interfaces, for instance, Conversational Interfaces (CIs) have evolved with the goal of simplifying human-machine interactions by allowing humans to engage with computers using human words.[1] They are especially important when individuals are preoccupied with other tasks, such as driving. CIs, also, attempts to support businesses in many ways in aiding customers.[2] Chatbots and voicebots are the two most common forms of CIs. Chatbots utilize natural language to mimic human interaction with a user via text, which is generally done via websites or mobile apps. Voicebots, on the other hand, understand natural language commands using speech recognition technology.[3] The author in[4] introduced multi-layer feed forward NN for spectrum sensing to identify the main users. Through their study, they figured out that the design structure of the neural network controls the accuracy of the detection, where they trained multi-layer feed forward NN with different back propagation algorithms. An overview of the mel-frequency cepstral coefficients, vector quantization and their relationship are presented in[5] where vector quantization works as a classifier of the speech signals. It can be combined with the mel-frequency cepstral coefficients and work as speaker recognition. Companies, consumers, and several scientific organizations have all prioritized the usage of conversational interfaces.[6] As businesses increasingly employ these conversational agents to communicate with customers, it's critical to understand the variables that influence people to use chatbots. This necessitates a higher sense of urgency, especially in light of recent research demonstrating the disadvantages and high failure rates of chatbots used on social media and messaging applications.[7] Since the debut of its bot API on Messenger, Facebook has revealed that their Artificial Intelligence (AI) bots have had a 70% failure rate. For instance, they do not accurately answer particular queries.[8] The objective of this paper is to create an app to ease the daily life of the users. The following is a description of how the paper is structured: System structure and requirements in section II, Chatbots systems are presented in section III, The integration procedure presented in section IV, The implementation methodology provided in V, The MFCC combined with neural networks presented in VI and the conclusion is provided in section VII.

System structure and requirements

The first step that needs to take place is to record the spoken words and convert it to text, followed by ascertaining that the is able to extract the intent using its artificial intelligence algorithms. The next logical step is to confirm that the Virtual Assistant is able to respond based on the intent deduced in the prior step. Some responses need to execute system commands, others need to get information from third party Application Programming Interface (API) (like weather, and other applications) or changing some values on the Internet of things (IoT) devices. The Software will react based on the confidence that it has in the intent. If it is not too confident about it, it will ask the user to repeat the spoken words in a different form. Lastly, the Software will take action and playback a voice to indicate what action it takes or reply with an answer if needed. This visualizes in the system architecture below in Figure 1. The system requirements necessary for the design and implementation process are:

a. Computer with a Linux operating system installed on it.

b. (Wit, Wolframalpha, and Snowboy) accounts.

c. Python version 3 with libraries (VLC, speech recognition, Snowboy, Wit, WolframAlpha, gtts, urllib3, swig, pyaudio, portaudio, sox).

d. Thingspeek account and channels.

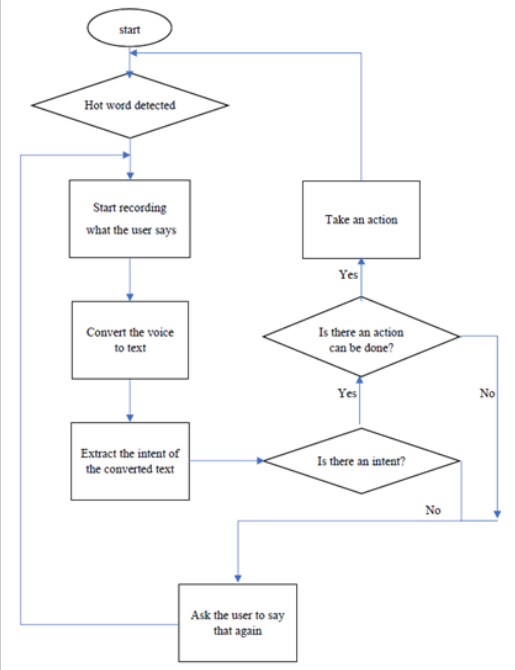


Figure 1 System architecture.

Chatbots system

A chatbot is a software program that performs tasks, provides services, or initiates activities in response to orders or queries from the user. It uses natural language processing (NLP) to maintain a dialogue with people. The chatbot is extensively utilized for various purposes, including amusement, teaching, and information retrieval.[9] Artificial intelligence is at the heart of the chatbot system's technology. A user-friendly, conversational, educated, and rapid answer are among the chatbot's key characteristics.

Voice-driven virtual assistants, such as Google Assistant, Apple's Siri, Amazon's Alexa, and Microsoft's Cortana, are the most well-known use of this technology. A voice assistant is Software that can assist you with certain simple activities that are not difficult to do but take time, so you may tell the assistant to complete them while you do something else. The general assumption that the chatbot designers have to keep in mind is, that when users initiate a conversation with a chatbot, they usually have a goal in mind that they want to achieve by the end of the conversation. This affects the pace and themes of the discussion, so as to reach the desired outcome.[10] The overall interaction of the chatbot system is depicted in Figure 2. To begin, the system receives the user's text input and compares it to its database to determine the appropriate answer. The system will transmit the correct text output to the middle device once it has found it. As a result, the most difficult element is defining the precise meaning.[11]

Chatbots are evaluated using assessment metrics, which are: Dialogue efficiency, Dialogue quality metrics, and above all User satisfaction. The first statistic, Dialogue efficiency, assess the system's ability to reply to user inquiries effectively. Second, the system's rationality is measured by the conversation quality metric. It divides responses into three categories: reasonable reply, weird but understandable, and nonsensical reply. User satisfaction is the last statistic, which is based on direct user feedback.[9]

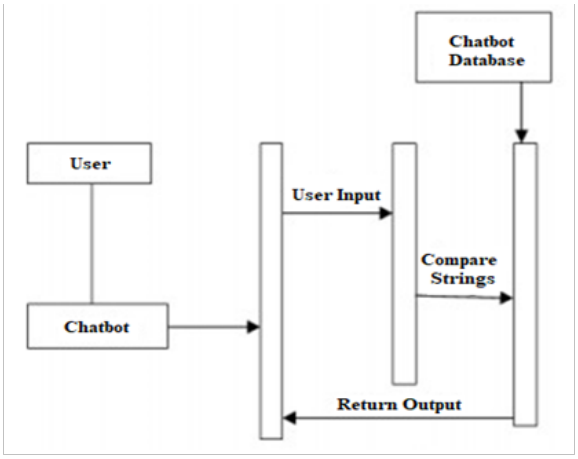


Figure 2 Design flowchart for a chatbot.[11]

Intent recognition

Intent recognition, also known as intent classification, is the process of categorizing a written or spoken input according to the user's goals. Intent recognition is a critical component of chatbots, with applications in sales conversions, customer service, and a variety of other domains.[12] The system uses a natural language processing (NLP), Machine Learning (ML) system that has been trained on a database of numerous phrases, which are identified manually by humans.

Speech recognition and voice assistant

The process of converting or translating spoken voice to text is known as speech recognition. It's possible to term it "Speech to text" (STT). You'll still need a microphone to record the voice and convert it to an electrical signal. ML will compare the acoustics of each phrase until it finds a match, then send the phonemes as text to NLP to predict the right word based on the complete material provided. A voice assistant is a piece of Software that integrates everything we've spoken about so far. It also employs text-to-speech (TTS) technology. These days, Alexa, Siri, Cortana, and Google Assistant are all quite popular. For the time being, Alexa is the best option. By dissecting its features and capabilities, the advantages and disadvantages may be identified, resulting in a basic vision for this paper.[10]

Preparation procedure

The idea is to create Software that is triggered by a certain word or phrase, which for this paper is "Hey Alice", which initiates listening process, and it, begins to listen to what the user is saying, evaluates what is said, and then takes the appropriate action. Finally, it and then responds, articulating, the action that was taken or it returns a response, that it did not comprehend the command. A trigger word detection ML model may be used to create the new model. The appropriate method to accomplish that is to compare the audio of the word with the audio from the recording, however, it could also be done using the STT model paired with the NLP model to filter and locate the word. A large variety of data is required for a successful model, and the data set must be filtered to meet the specific application, removing poor words, repetitive and superfluous phrases. And this work is difficult due to the large amount of data required for training. Snowboy in Figure 3 is one of several ready-made options available. Because it solves the problem mentioned above, the off-the-shelf product is therefore an obvious choice. The off-the-shelf approach provides a

model that only has to be trained relatively less frequently, before it can be used. Snowboy was utilized for this study because it can operate in the offline mode; it also allows the community to assist train the model, resulting in a better model.

Figure 3 Snowboy logo.[13]

When the system is activated, it starts recording and stops recording when the user stops, and then sends the record to the STT model as shown in Figure 4 which is constructed using Python. The STT model is based on the NLP, Hidden Markov model, and the long-short-term memory (LSTM) neural network. The reason why Google Web Speech API was chosen is the ability to use it without making an account or getting an API key. It could work with around 120 languages but is limited to only 50 speech-to-text transactions per day, which is good enough.

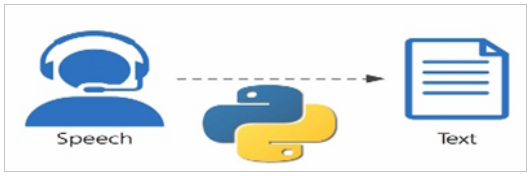
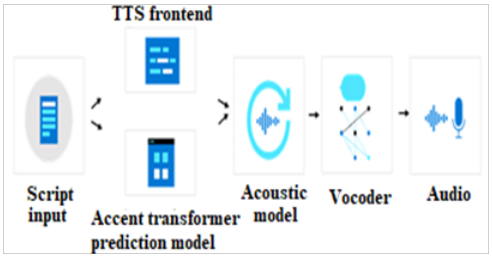


Figure 4 Speech to Text model based Python.[14]

After this training, the model will have a stronger sense, thanks to NLP and the recurrent neural network RNN. The advantage of utilizing RNN, in this case, is that it considers the interdependence of all terms in the phrase. Because the sequence of words in the phrases might affect the meaning or aim, it's critical to extract the sentence's intent. The Wit.ai proved adequate for intent recognition. It's straightforward to use, and it's the only free option that was discovered before the start of the project's implementation. To use any Application Programming Interface (API), the script either needs to send a request or receive a response from the API. However, both operations need some preparation to work correctly. After a successful request, the API will send a response back. This response usually comes in JSON format, so it needs to be filtered because it contains a lot of unnecessary information that needs to be filtered and eliminated. After the filtration, the script has the information that it will use to make a decision. Text to speech is a way of transforming a response to convey what was done if an action was taken or to show that it did not understand. Text to speech (TTS) has a variety of options. There are several non-AI alternatives available, but the Google "gtts" AI-based solution was the best choice because it does not require an account or a large library to be loaded on the machine.[15,16] Figure 5 & 6 shows the steps for converting text to speech using AI. Table 1 present the comparison between STT solutions.



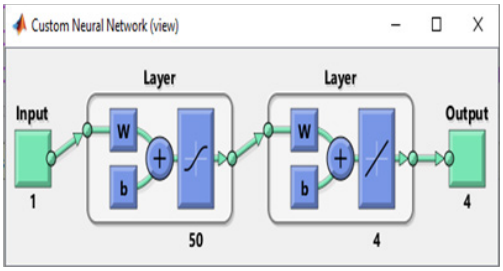


Figure 5 How AI-based TTS works.[15]

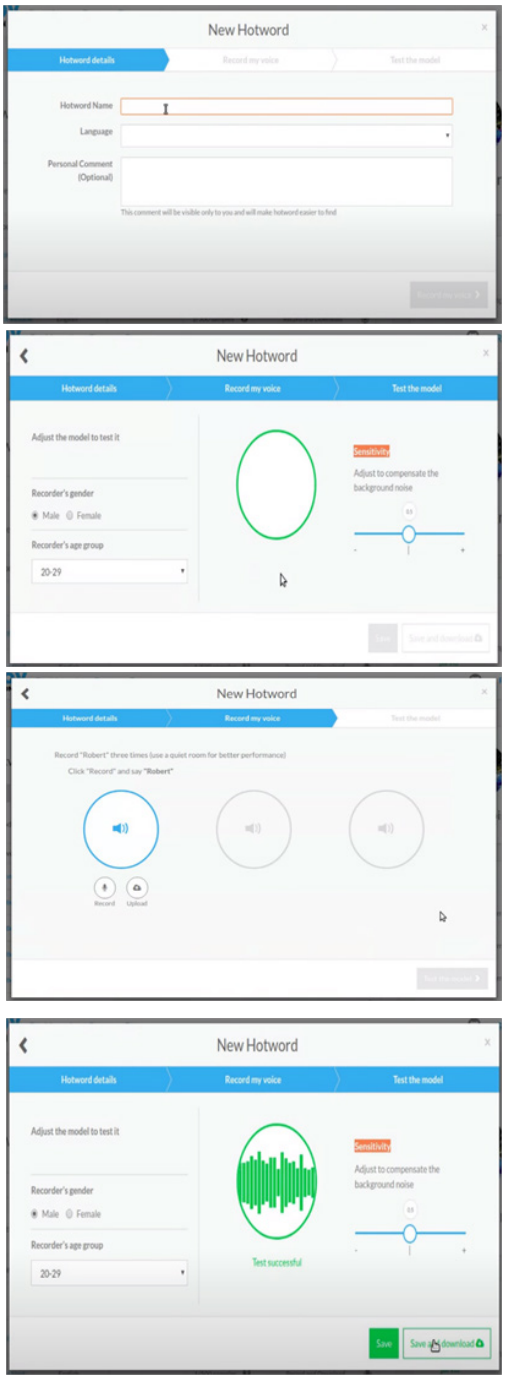


Figure 6 Process of creating a new model.

Table 1 Comparison between STT solutions

	Accuracy	Transactions limit	An additional package	An account is needed
Microsoft Bing Speech	80	5000 transactions monthly	No	Yes
Google Web Speech API	70	50 transactions daily	No	No
Google Cloud Speech	90	60 minutes monthly	Yes	Yes
Houndify	unknown	400 seconds daily	No	Yes
IBM Speech to Text	69	500 minutes monthly	No	Yes

Table 2 Recognation of Academic Words for 10 MFCC

Num. of Neurons	RP %			
	Training Algorithm Levenberg-Marquardt (LM)			
	Estimate	Source	Process	Variable
10	70	79	78	80
20	73	82	81	83
30	81	87	84	87
40	84	88	88	90
50	85	90	92	93
Num. of Neurons	Training Algorithm BFGS Quasi-Newton Resilient Backpropagation (BFGS)			
10	65	71	71	73
20	71	77	73	77
30	81	80	75	80
40	77	81	80	82
50	79	82	83	83

Implementation methodology

Before we talk about the Implementation, let us divide the Software into 5 main pieces. The first piece is "Activation", so Alice gets activated once it hears the keyword "Hey Alice. The second piece is "Speech-to-text", Alice will start recording what the user is saying then convert it to text. The third piece is "Intent recognition", it will try to figure out the move to the next step. Otherwise, it will ask the user to say that again and go back to the speech-to-text. The fourth piece is "processing and execution", which will try to take the proper action, if any, based on the intent of what has been said. Otherwise, it will also tell the user that it could not take a proper action and request the user to repeat the command again, and then, go back to the speech-to-text part. The last and the fifth piece is "Text-to-speech", at this point, Alice will say something back either it is an acknowledgment or a response to what has been requested. Snowboy is a highly customizable keyword detection engine that is set up to listen in real-time, and is always listening (even when offline) compatible with Raspberry Pi, (Ubuntu) Linux, and Mac OS X. A keyword (also known as wake word or trigger word) is a keyword or phrase that the computer constantly listens for is, a signal to trigger further actions. To start using Snowboy, a keyword must be selected or created.

Either way, an account is needed. For the paper, the keyword "hey Alice" needs to be created. After making an account on the site, we created a "Hey Alice" model and tested it, which worked successfully, Fig. 6, represents the steps for creating a new model. The speech-to-text part is done using Google Web Speech API. Google Web Speech is an API that converts speech to text based on ML algorithms with a good accuracy of around 70%, thus less work needed to make it work, as shown in Table 1. The intent recognition part is done using Wit.ai API. Wit, is a natural language interface for applications, and is capable of turning sentences into structured data. The Wit.ai API allows HTTP GET calls to return the extracted meaning from a given sentence based on examples. The API authenticates with OAuth2 and returns JSON formatted responses. Figure 7 below shows the steps needed to train the API.

The processing and execution part is done to select the most proper action/response if there are intent and entity to choose from, and the action/response based on these. This is accomplished based on a sequence conditional statement. To make Alice more interactive and enjoyable to use, we varied her answers slightly so that, but used interchangeable words conveyed the same meaning, and got the job done.

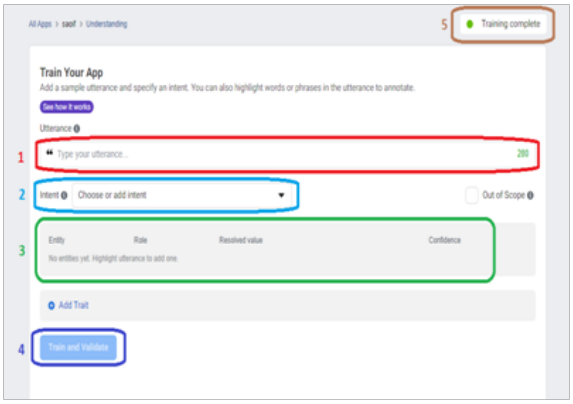


Figure 7 Wit training.

Lastly, to reply, Alice needed to convert text to speech for which, we used, google text to speech API, successfully allowed her the ability to speak.

As a result of all that work, Alice has a good accuracy of Activation or keyword detection. Based on our usage experience, we approximated the accuracy of Activation to be at least 80%. The STT has less accuracy, which is not good, but it gets the job done, especially considering it's a free service. The intent recognition was also good, if you trained it well, and what makes it even more fascinating is the ability to train the model while your app is working normally. The processing, taking action, and reply work very well for almost everything except questions because the response to the questions is based on Wolfram Alpha API, which is a unique engine for computing answers and providing knowledge. So from time to time, if the user does not ask a straightforward question, Wolfram Alpha is not sophisticated enough and, so it might respond with a wrong answer.

Citation: Abougarair AJ, Aburakhis MKI, Zaroug MO. Design and implementation of smart voice assistant and recognizing academic words. *Int Rob Auto J.* 2022;8(1):27–32. DOI: 10.15406/iratj.2022.08.00240

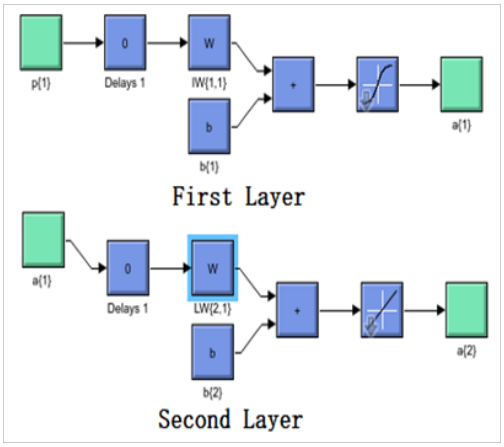


Figure 8 Structure of two layer feedforward neural networks.

Mel frequency cepstral coefficient (MFCC) combined with neural neyroks

The initial stage in any automatic speech recognition system is to extract features or identify the audio signal components that are useful for detecting linguistic content while ignoring everything else, such as background noise, emotion, and so on. The most important thing to remember about speech is that the form of the vocal tract, which includes the tongue and teeth, filters the sounds produced by a human. The sound that emerges is determined by this shape. If we can precisely establish the shape, we should be able to accurately represent the phoneme being produced. The envelope of the short-time power spectrum reflects the curvature of the vocal tract, and MFCCs' purpose is to appropriately represent this envelope. Mel Frequency Cepstral Coefficient (MFCC) can be used to extract the distinctive properties of the human voice, and this MFCC also represents the short-term power spectrum of the human voice. MFCC is used to produce the coefficients that describe the frequency Cepstral, which are based on the linear cosine transform of the log power spectrum on the nonlinear Mel frequency scale. Mel scale approximates the human voice more accurately since the frequency bands are evenly spaced Mel frequency is evenly spaced on the Mel scale, and this frequency is used to linearize Mel scale values below 1000 Hz and to find the log power of Mel scaled signal above 1000 Hz using linear space filters. Mel frequency wrapping is beneficial for better voice representation.[17] Pattern training and pattern comparison are two crucial elements in the pattern-matching process. In the pattern-comparison stage of the technique, the unknown talks are directly compared to each conceivable pattern learned in the training stage in order to infer the identity of the unknown based on the patterns' goodness of fit.[18] In pattern recognition, there are many basic models that are used. One of these methods is artificial neural networks, which are considered one of the most important methods in the training and learning process. Using MATLAB, the recognition performance was assessed for various MFCC coefficient values, as well as varying training algorithms and numbers of neurons in hidden layers of neural networks.

The following are the database's primary features: English spoken by non-native speakers, a single session of sentence reading, and relatively large speech samples ideal for learning individual speech characteristics. Different MFCC coefficients and two-layer feedforward neural network with a different neuron in each layer were used for classifications shown in Fig. 8. The database consists of 150 speakers, 70 female, and 80 males.

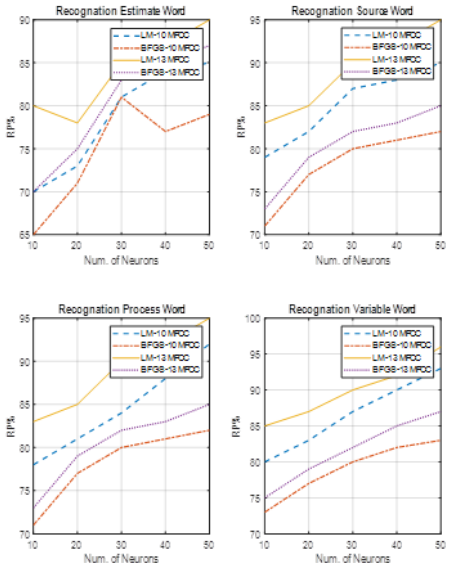


Figure 9 Recognition percentage Vs number of neurons and MFCC coefficients.

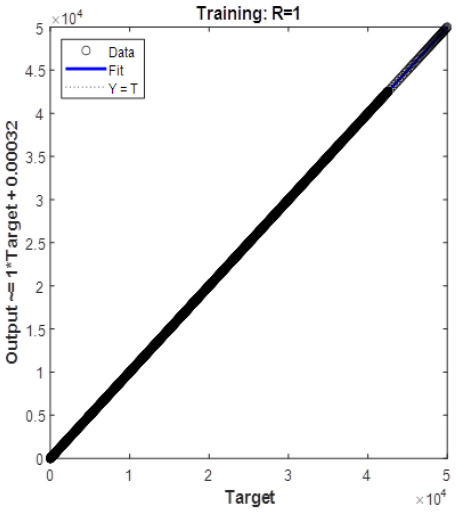


Figure 10 Recognation between target and actual data based LM.

The recording was divided into part training (70%), validation (15%), and test (15%). In addition, a set of algorithms for the training process will be applied, for example, Levenberg-Marquardt (LM), Bayesian Regularization, BFGS Quasi-Newton Resilient Backpropagation and Scaled Conjugate Gradient. Table 2 present recognition percentage (RP%) Vs number of neurons for 10 MFCC coefficients with different training algorithms. Also, TABLE 3 presents recognition percentage (RP%) Vs the number of neurons for 13 MFCC coefficients with different training algorithm networks. As seen from Figure 9 the highest recognition percentage of all the academic words was based LM training algorithm with 13 coefficients of MFCC. Figure 10 display the high-performance correlation between the target and actual database LM with 50 neurons in the hidden layer of neural networks. Figure 11 explains the training data error decrease as the NN learns.[19,20]

Citation: Abougarair AJ, Aburakhis MKI, Zaroug MO. Design and implementation of smart voice assistant and recognizing academic words. *Int Rob Auto J.* 2022;8(1):27–32. DOI: 10.15406/iratj.2022.08.00240

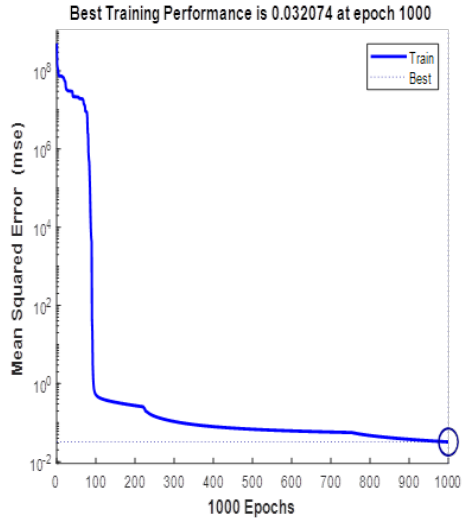


Figure 11 The training data error decrease as the NN learns.

Conclusion

The virtual assistant is a beneficial thing to have in your home. It could help with many tasks and save you a lot of time. What has been achieved in this paper is building a virtual assistant that could answer many types of questions, control IoT devices, and control the peripheral devices like speaker and camera. The work in this field is tremendous, but at the same time, it is shrouded in secrecy, and there is not much information in this field. The reader can find a lot of solutions as API or as Software but not open-source projects or how to do it yourself tutorial which shows the competitiveness in this field. Lastly, the ability to improve the Software is endless. The limit to what you can do, is how much time, effort, and knowledge you want to devote to it. For detecting some spoken academic words, this experiment compared different parameters of a speech recognition system with an artificial neural network. For the purpose of classification, a feedforward neural network was deployed, and MFCC methods were used for feature extraction. The best performance-based recognition percentage was found to be 13 MFCC coefficients with two-layer feedforward and 50 neurons in the hidden layer.

Conflicts of interest

The authors declare there are no conflicts of interest.

Acknowledgments

None.

Funding

None.

References

1. Conversational UI - A paradigm shift in business communication. *Maruti Techlabs*. 2017.
2. Conejos F. Conversational Interfaces: The Guide (2020) Landbot.io. Landbot.io. 2019.
3. Chatbot: What is a Chatbot? Why are Chatbots Important? Expert System. 2020.
4. Chatterjee S, Mandal R, Chakraborty M. A Comparative Analysis of Several Back Propagation Algorithms in Wireless Channel for ANN-Based Mobile Radio Signal Detector. *BibSonomy*. 2013.
5. Trivedi Vaibhavi, Chetan Singadiya. Isolated Word Speech Recognition Techniques and Algorithms. *IJSRD - International Journal for Scientific Research & Development*. 2013;1(2):2321–0613.
6. Dale R. The return of the chatbots. *Natural Language Engineering*. 2016;22(5):811–817.
7. Knight W. How to Prevent a Plague of Dumb Chatbots. 2016.
8. Orlowski A. Facebook scales back AI flagship after chatbots hit 70% f-AI-lure rate. 2017.
9. Shawar BA, Atwell E. Different measurement metrics to evaluate a chatbot system. In Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies. 2007:89–96.
10. Ina, The History Of Chatbots – From ELIZA to ALEXA. 2017.
11. Dahiya M. A tool of conversation: Chatbot. *International Journal of Computer Sciences and Engineering*. 2017;5(5):158–161.
12. Marshall C. What is intent recognition and how can I use it? super.AI. 2020.
13. kitt, "kitt.ai," kitt , 2016.
14. Samson. 7 steps to Converting Speech To Text with Python. 2020.
15. Liao Q. Neural Text to Speech extends support to 15 more languages with state-of-the-art AI quality. 2020.
16. Ahmed J Abougarair, Gnan H, Oun A. Implementation of a Brain-Computer Interface for Robotic Arm Control. *IEEE*. 2021.
17. Rekha Hibare, Anup Vibhute. Feature Extraction Techniques in Speech Processing: A Survey. *International Journal of Computer Applications*. 2014;107(5): 1047–1054.
18. Karpagavalli Chandra. A Review on Automatic Speech Recognition Architecture and Approaches. *International Journal of Signal Processing, Image Processing and Pattern Recognition*. 2016;9(4):393–404.
19. Ahmed J Abogarair. Neural Networks Identification and Control of Mobile Robot Using Adaptive Neuro Fuzzy Inference System. ICEMIS'20: Proceedings of the 6th International Conference on Engineering & MIS 2020, September 2020.
20. Ahmed J Abogarair. Model Reference Adaptive Control And Fuzzy Optimal Controller For Mobile Robot. *Journal of Multidisciplinary Engineering Science and Technology*. 2019;6(3):9722–9728.