JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

VICARIOUS FRIEND: ADVANCED VIRTUAL ASSISTANT USING AUTOMATION AND DEEP LEARNING

[1]K.K.A.PADMASHRI[1], [2]Mrs.M.KARTHIHAYINI[2]

Department of Information Technology, Meenakshi Engineering College,
Tamil Nadu
(Affiliated to Anna University Tamil Nadu)

Abstract:

People are using their smartphones, laptops, tablets, and other digital assistants more and more in today's technologically advanced society, and those who used to live with them are becoming less and less common. The technologies that engage with people more these days than real people in close proximity include Alexa, Siri, and Cortana. From the Turing test through the current deep face technologies, artificial intelligence has advanced. The word vicarious is a synonym for virtual. A friend is someone who responds to questions and is informed of amicable subjects, assists with tasks, and lightens the load. Since the virtual friend is more capable, working together lessens the workload, and when the helper is able to respond amiably to inquiries from the user, the assistant takes on the role of the virtual or vicarious friend.

The use of deep neural networks for learning has the advantage of increasing system accuracy to 94% when responding to the system's friendly queries. It would be highly desirable to extend this study in order to center the research on real-world datasets instead of just theoretical approaches and simulations.

I. INTRODUCTION

The primary objective of this project is to improve the field of making virtual assistants aware of user intents and responding accordingly by utilizing deep learning techniques. Automation is the use of technology to carry out operations or tasks with the least amount of human involvement. With the use of robotics, software, and artificial intelligence, it seeks to improve productivity, decrease errors, and streamline operations. A subfield of artificial intelligence known as "deep learning" trains multilayered neural networks to learn from input. It performs exceptionally well in applications like generative models, natural language processing, and picture and audio recognition.

To reduce errors and enhance performance, deep learning models iteratively modify their parameters using backpropagation and optimization methods. Project is a state-of-the-art virtual assistant that was developed at the cutting edge of technology and artificial intelligence. With its capacity to automate tasks, sophisticated deep learning algorithms, and neural network technology, chatbots act as a surrogate buddy, providing support and company while you work through the challenges presented by artificial intelligence. The comprehensive use of natural language processing (NLP) text classification allows the virtual assistant to be precisely customized to comprehend user inquiries and offer solutions. Whether chatbot is prepared to be your informed friend, providing chats and insights.

II. EXISTING SYSTEM

Virtual assistants, sometimes referred to as chatbots or desktop assistants, have advanced significantly in the last several years. To comprehend and react to user requests or orders, these systems use natural language processing (NLP), machine learning, and human-computer interaction strategies. While their functionality may differ, they often include things like speech recognition for hands-free interaction, task automation for managing emails, appointments, and reminders, and information retrieval for accessing web data and getting news, weather, and other updates. Their usefulness is further increased by integration with different apps, such as calendars and productivity tools.

Desktop assistants are always learning from their interactions with users and utilize machine learning to improve their comprehension and replies over time. Scholars are always looking for ways to enhance these systems in order to increase their capacity to comprehend context, add more language support, and handle more difficult tasks.

III. PROPOSED SYSTEM

This sophisticated virtual assistant was created by combining a deep-learning chatbot with an automated system. The deep learning chatbot is made more human-friendly by using various human conversational expressions during training. the comprehension of the input in speech and text. If the input is speech, the system translates it to text and categorizes it based on whether or not it is a command. If the input is a command, the user-intentioned command will be carried out by utilizing various commands that have been defined in utility functions. If the user input consists of a chat or a pleasant message, the text will be pre-processed using natural language processing (NLP) and transformed into a numerical word bag to be fed into a sequentially trained neural network (a deep learning technique). In context with the user's statements, the model accurately determines the intent and responds. The virtual assistant is upgraded by the deep learning chatbot to become a virtual friend or vicarious assistant, giving it intelligence.

IV. SYSTEM SPECIFICATION

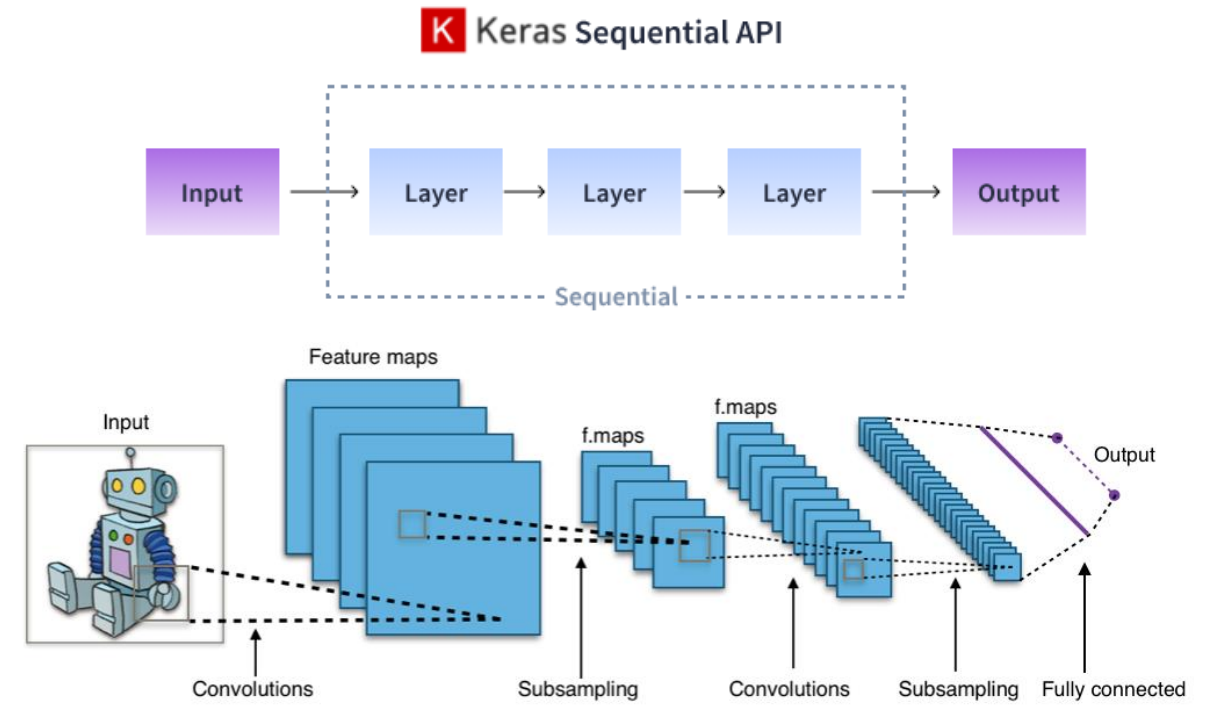
HARDWARE CONFIGURATION:

Processor	-	I5
Speed	-	3 GHz
RAM	-	8 GB(min)
Hard Disk	-	500 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	LCD

SOFTWARE CONFIGURATION

Operating System: Linux, Windows/10/11
Server: Anaconda, Jupyter, Visual studio code
Front End: Html, CSS , Java-srcipt
Server-side Script: Python .

V. ABOUT THE SEQUENTIAL API OF KERAS



A sequential model's step-by-step procedure from TensorFlow's Keras API, in addition to a few other things:

1. Import Libraries: TensorFlow and Keras, two libraries that offer tools for creating and training neural networks, should be imported first.

2. Define the Sequential Model: The layers of the neural network are housed in a Sequential model object, which must be created next. You can add layers to this model in a sequential order.

3. Add Layers: The `add()} method must be used to add layers to the model. These layers can be of any kind that Keras supports, including recurrent, convolutional, and dense (completely connected) layers. Every layer has a part in transforming the input data.

4. Configure Layers: A variety of parameters, including the number of neurons, activation functions, kernel sizes, and strides, must be specified for each layer. The neural network's behavior and functionality are determined by these configurations.

5. Compile the Model: The `compile()` method must be used to compile the model after layers have been added. You can set up more parameters during compilation, like the optimizer, loss function, and training metrics.

6. Train the Model: After the model has been constructed, the `fit()` method can be used to train it on training data. In order to minimize the loss function, this entails feeding the model with the input data and the related labels and iteratively changing the layer weights.

7. Evaluate the Model: The `evaluate()` method can be used to assess the model's performance on untrained data once it has been trained. This tells us if the model is overfitting or underfitting, as well as how well it generalizes to new data.

8. Create Predictions: Lastly, the `predict()} method can be used to apply the trained model to new data in order to create predictions. This enables you to use the patterns you've learned to draw conclusions or categorizations for cases that you haven't seen.

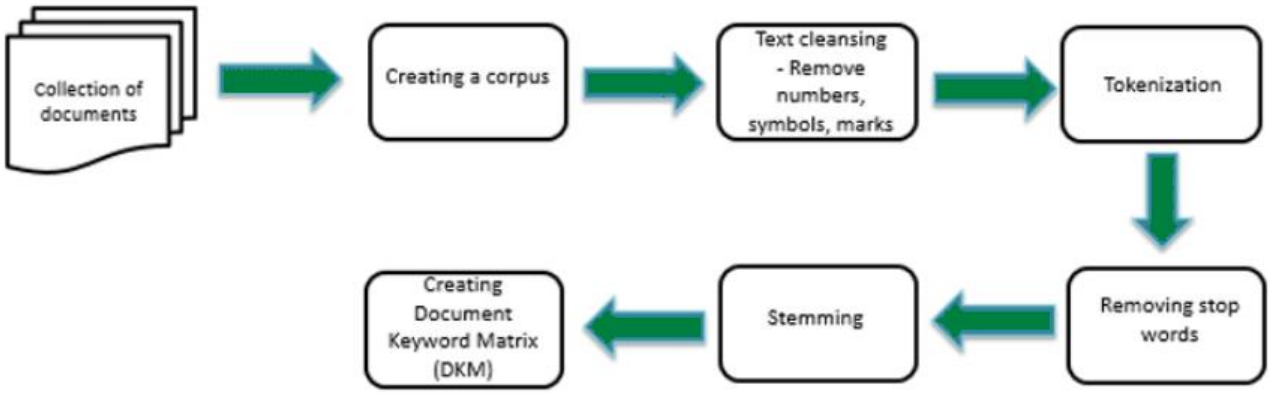
Regularization: To avoid overfitting during training, one might use strategies like dropout or L2 regularization.

Callbacks: Keras offers callbacks that enable training monitoring and actions like early training termination and model saving. Examples of these callbacks are EarlyStopping and ModelCheckpoint.

Visualization: For improved comprehension and analysis, tools like TensorBoard make it possible to see metrics, model architectures, and training progress.

Transfer Learning: Sequential models can accelerate development and boost performance by using pre-trained models for tasks like feature extraction or fine-tuning on fresh datasets.

Hyperparameter tuning: To further improve the model's performance, methods like grid search or random search can be used to identify the ideal hyperparameters.



Natural language processing (NLP) preprocessing of text using a step-by-step method that only requires five steps:

1. Tokenization: Divide the text into manageable chunks known as tokens, which could be individual words, phrases, or characters.

By doing this, the book is divided into digestible chunks for additional study.

2. Lowercasing: To guarantee language consistency, change every token to lowercase.

This makes it easier to treat words with multiple cases—like "Hello" and "hello"—as one token.

3. Removing Punctuation and Special Characters: Eliminate punctuation marks, symbols, and other non-alphanumeric characters.

This step helps in cleaning the text and removing noise that might interfere with analysis.

4. Stop-word Removal: Remove common words that do not carry significant meaning, known as stopwords (e.g., "the", "and", "is").

Stop-word removal reduces the dimensionality of the data and focuses on more meaningful content.

5. Stemming or Lemmatization: Reduce words to their base or root form to normalize variations of the same word.

Stemming chops off suffixes from words (e.g., "running" becomes "run"), while lemmatization maps words to their dictionary form (e.g., "ran" becomes "run").

These steps collectively prepare the text data for further NLP tasks such as sentiment analysis, text classification, or topic modeling by cleaning, standardizing, and reducing noise in the text.

5.1 ADVANTAGE OF PROPOSED SYSTEM

1. Automation Capability: Boosts productivity by mechanizing tedious jobs, freeing up time for more crucial ones. minimizes human error and guarantees correct and consistent completion of tasks.

2. Web search: Offers immediate access to a plethora of information available on the internet.
saves time by delivering precise and pertinent search results quickly.

3. Capability to Take Screenshots Voice Command: Provides convenience by enabling users to take screenshots without using their hands; this is particularly helpful when multitasking or in scenarios when manual input is challenging.
speeds up the transfer of data or graphic materials.

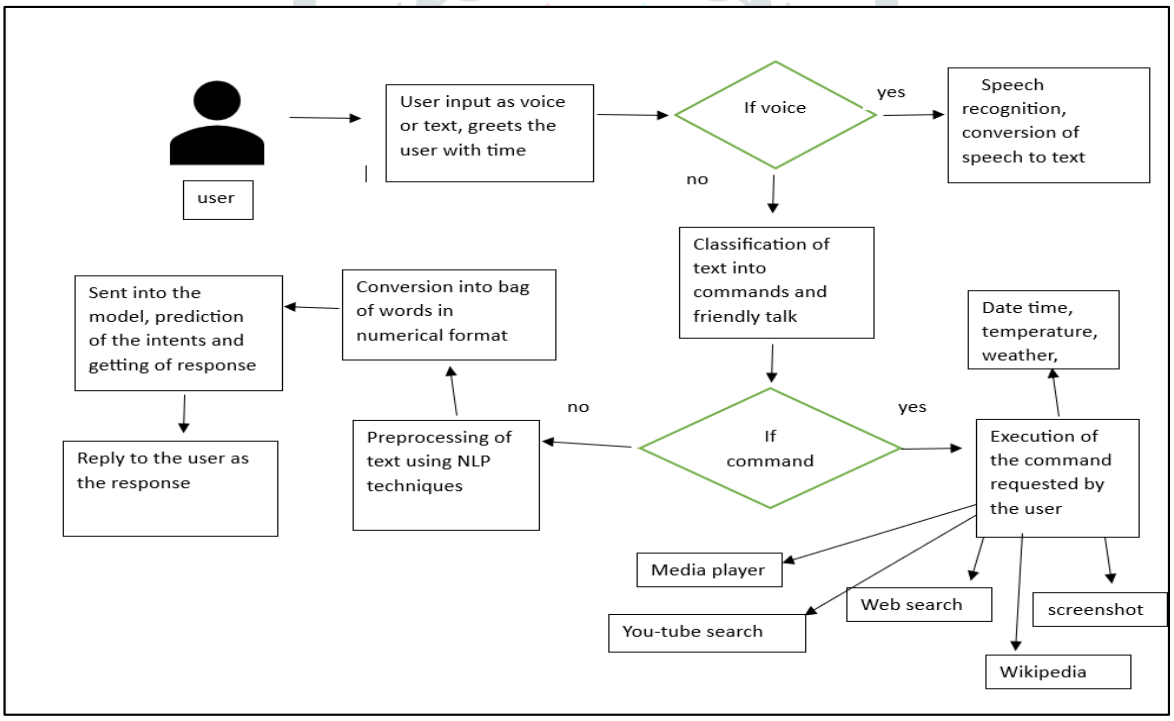
4. Speech Recognition: Provides hands-free functionality, especially useful for individuals with limited movement or those performing tasks that need manual input increases accessibility by enabling spoken instructions to be used to interact with apps or devices.

6. Datetime Telling: This feature makes it easier to schedule and manage your time by giving you fast access to the current date and time. saves time and effort by doing away with the need to manually check clocks or calendars.

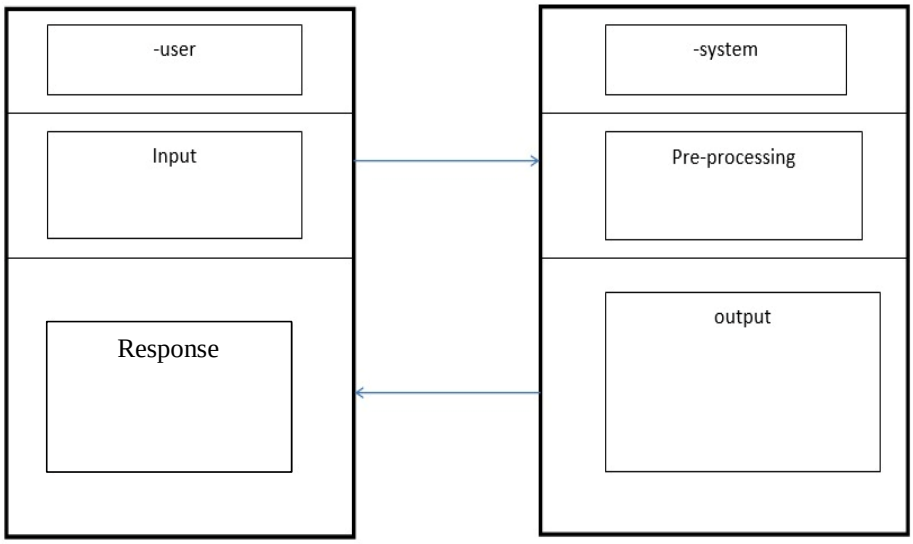
7. Amicable Chatbot Responding to Users as If They Were Friends: Develops a friendly and individualized user experience, building connection and camaraderie reduces feelings of isolation and loneliness, especially for those who communicate with the assistant regularly increases user loyalty and pleasure by establishing a rapport with them and responding sympathetically.

Adding web searching as a function increases the virtual assistant's usefulness by giving users access to a multitude of online resources, which boosts learning and productivity.

VI. ARCHITECTURE DIAGRAM



6.1 CLASS DIAGRAM



DATASET MODULES

A collection of data that has been arranged in a certain manner is called a dataset. A dataset can contain any kind of data, including database tables and array series. A tabular dataset is comparable to a database table or matrix, in which the fields of the dataset are represented by the rows, and the variables are represented by the columns. The file type that is most frequently supported for tabular datasets is "Comma Separated File," or CSV. But storing "tree-like data" in a JSON file is a more efficient use of its space. sorts of data that are present in datasets .

A dataset is a group of data that has been organized into a specific order. Any type of data, from an array series to a database table, can be included in a dataset. A tabular dataset is similar to a database table or matrix, where each row represents the dataset's fields and each column represents a specific variable. "Comma Separated File," or CSV, is the file type that is most commonly supported for tabular datasets. However, we can use the JSON file more effectively to store a "tree-like data." Data types found in datasets .

Numerical information: property value, temperature, etc.

Data that can be classified as True/False, Yes/No, Blue/Green, etc.

Ordinal data: They are comparable to categorical data, but they cannot be assessed through comparison.

Requirement for a dataset

Because ML/AI models cannot be trained without data, working on machine learning projects requires a massive amount of data. One of the most crucial phases in the development of an ML/AI project is gathering and preparing the dataset.In any machine learning project, the technology will not work as intended if the dataset is not properly prepared and pre-processed.The developers' only resource for working on the machine learning project is the datasets. Two categories are used to divide datasets when developing deep learning learning apps:

Training Dataset

Test Collection

VII. IMPLEMENTAION

The following steps are included in the implementation process:

1. Data collection: Greetings, inquiries, flowers, education, sleep, free counsel, and other human-friendly contact terms will all be captured in the JSON format.

2. Preparing the text and loading the data: The material will be transformed into a bag of words (numerical format to be fed into the model) using the sequential flow of NLP preprocessing approaches.

3. Providing the input parameters for the sequential model defined by Keras

4. Using a dense layer and up to 100 epochs of processed data to train the neural network.

5. Using data to test the model and ensuring that it is accurate

6. Frontend development with HTML, CSS, and JavaScript

7. Building an automation system through the definition of several automation utility functions to be used as needed.

8. Using Python's EEL module to connect the front end and back end.

VIII. RESULT ANALYSIS

The "Vicarious Friend" project, which includes automation to help users complete activities and a chatbot with deep learning capabilities that answers nice questions, would have its results analyzed by taking into account multiple important factors.

1. User Interaction and Satisfaction: Having the ability to talk and type was a positive user experience that could be utilized by a broad range of users.

2. Task Completion Efficiency: The ability to do a task faster than the allotted time. It is highly catchy to give helpful advise, act as a stress reliever, and motivate someone to finish a task and take a break.

3. Accuracy of Chatbot Responses: The deep learning model recorded 92% accuracy for Aquaray, which aids in providing the most relevant answers.

4. Effect on User Productivity: Encourages the user to feel less alone by offering free counsel, cracking jokes, and easing their anxiety of being alone.

5. Compatibility: Suitable with all computers, laptops, workstations, and systems.

IX. REFERENCES

[1] Vishal Kumar Dhanraj and Semal Mahajan Lokeshkriplani, "Research Paper onDesktop Voice Assistant", *International Journal of Research in Engineering and Science (IJRES)*, vol. 10, no. 2, pp. 15-20, 2022.
[2]Nivedita Singh, Dr. Diwakar Yagyasen, Mr. Surya Vikram Singh, Gaurav Kumar and Harshit Agrawal, "Voice Assistant Using Python", *International Journal of Innovative Research in Technology*, vol. 8, July 2021.
[3] A.M. Sermakani, J. Monisha, G. Shrisha and G. Sumisha, "Creating Desktop Speech Recognization Using Python Programming", *IJARCCE*, vol. 10, no. 3, pp. 129-134, March 2021.
[4]T. -K. Kim, "Short Research on Voice Control System Based on Artificial Intelligence Assistant", *2020 International Conference on Electronics Information and Communication (ICEIC)*, pp. 1-2, 2020.
[5] Sutar Shekhar, Pophali Sameer, Kamad Neha and Prof. Devkate Laxman, "An Intelligent Voice Assistant Using Android Platform", March 2015.