

Voice Assistant Powered by AI

Piyush Arora Kartik Choudhary Kartik Chippa

IIT Jodhpur, India

Introduction

We are all well aware of Cortana, Siri, Google Assistant, and many other virtual assistants designed to aid users with Windows, Android, and iOS platforms. These Artificial-Intelligence based voice assistants have become plentiful over the past couple of years. But to our surprise, there is no complete voice assistant available on Linux platforms with an appreciable number of users. Moreover, why are we still learning terminal commands? Is there a way to use terminal commands without cramming them? Is there a way to teach beginners to operate with terminal statements? New users generally find it hard to learn terminal commands. Can we use advancements in AI to tackle this problem and provide a user-friendly solution? What if my terminal is my new friend? To whom I can to? We are daring to put a step forward in this direction and thus our project proposes a solution for the same.

Proposed Solution

We propose to fine-tune pre-trained BERT (Bidirectional Encoder Representations from Transformers)[1] which is an open-source large language model for natural language processing (NLP) on our own dataset using Google's Dialogflow (which provides a model hosted on the cloud). We fine-tune our model to get a context and the target token out of a command fed to the model. We fetch these outputs from the cloud-hosted BERT model using python scripts giving an option of voice input as well. The response is stored in a JSON file and then processed by our scripts. The results are then passed through a module that converts the text into a voice that is heard by the user. To make the user comfortable with our service, we also provide a user-friendly interface built using the Tkinter python library.

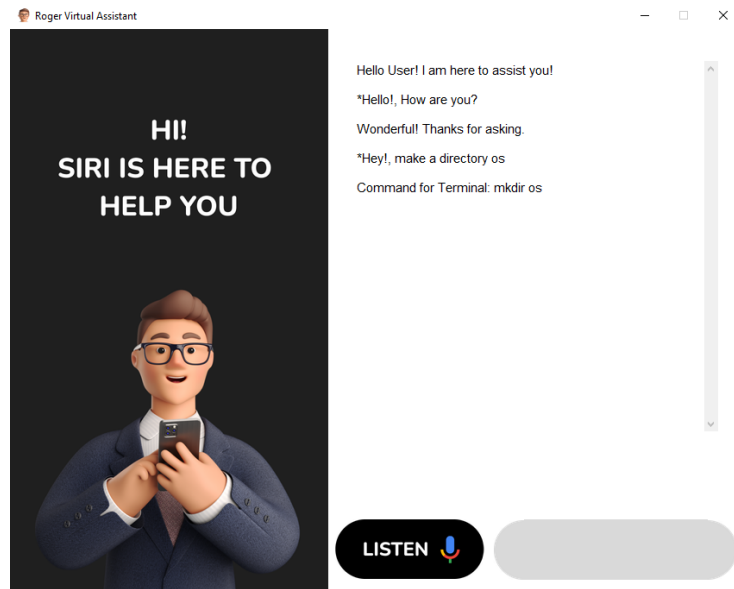


Figure 1: A snapshot of GUI

We provide two modes for our product:

- An advanced terminal with basic functionalities like `mkdir`, `rmdir`, `ls`, etc. Along with these, we provide additional features like googling an unknown command, playing music on Spotify, youtube your favorite song, playing chess, snake and ladders, and many more! All these commands can be given through voice.
- We also provide an AI-assisted mode, which when turned on, converts our common input language into terminal commands. For example, if a user asks the terminal, "Hey!, make a directory named ABC", the program will be able to convert it into `mkdir ABC`. The fine-tuned BERT model will output the context of the statement (`mkdir`), as well as the target folder name (ABC). The same functionality has been implemented for statements like `rmdir`, `ls`, `cd`, `pwd`, etc. All these commands can be given through voice.

Output of all these requests is visible on the terminal as well as the GUI. The output is recited by the python scripts and thus provides a user-friendly environment. This proposed pipeline helps us to interact with the system using voice commands and establishes a friendly connection between AI and the terminal.

Outcomes

The project intends to build a fully functional voice assistant that understands human voice commands and completes tasks requested by the user by using AI-

powered techniques, and Natural Language Processing(NLP). BERT converts the natural language input like "Hey! Create a directory abc" into "mkdir abc" for the terminal which can be understood by the OS terminal. Based on our understanding and study of numerous Linux general-purpose commands and human conversations, we have created our own dataset using the Google Service, Dialogflow. Elements of NLP, AI, and a friendly user interface are added to enhance the functionality of terminal commands. Modular coding technique was promoted so that the code-base could be maintained over several years, without making any heavy changes. Meeting the needs of a wider number of users, we have proposed this assistant to help beginners learn various terminal commands with a convenient graphic user interface. As this assistant can interpret natural language, and not only hot words (which other assistants do, get activated for only specific commands), we get an edge over other assistants.

Lessons Learned

Different lessons learned by us can be seen below:

AI to understand Language: We learned to use the cloud-based language model in order to take the terminal to the next level.

A friendly user interface: We learned to use Tkinter which is a popular and standard GUI library for Python. It provides various functionalities, widgets, and controls such as buttons, labels, and text boxes which we have used in our GUI application.

I/O using Voice: We learned to use voice as a source for input and output to Linux-based systems.

OS system calls: In order to execute various terminal commands we learned to use the system calls to interact with the operating system.

Text to Speech and Speech to Text Learned how to use python libraries to convert text to speech and speech to text.

Code for our Assistant

https://github.com/PiyushArora1010/Voice_CMD

References

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. N. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: <https://arxiv.org/abs/1810.04805>