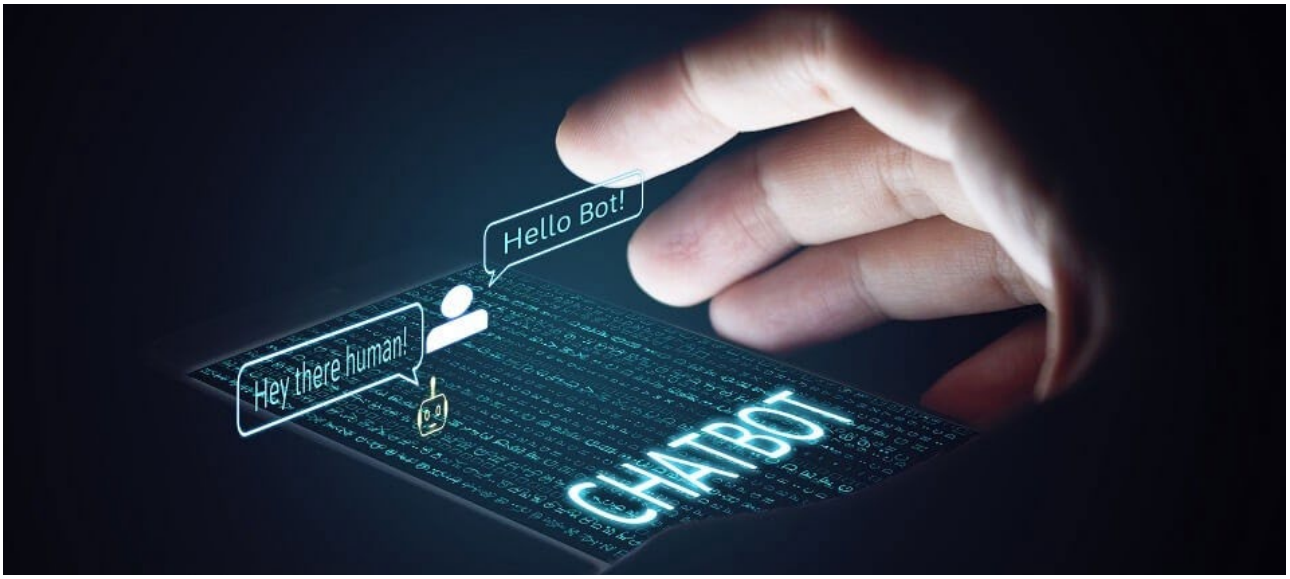
Software Engineering
Project Write-up

EVOQUE: An intelligent chatbot



Submitted by:

20103120 Bharat Singh
20103121 Neeraj Kumar
20103122 Sushant Singh
20103123 Divyansh Gurtoo

Problem Definition:

Typically, a user must manually manage many groups of programs in order to finish a task. For instance, a person attempting to plan a trip must look up the airport codes for local airports before checking travel websites for tickets between various airport combinations to get to the destination. There is a need for a system that can manage jobs effortlessly.

There are already many chatbots available. But we don't often use them. Many people experience difficulties with speech recognition. Although these computers are capable of understanding English, they are unable to distinguish some words in our accent. Our pronunciation differs greatly from theirs. Additionally, using them on mobile devices is simpler than using desktop computers. A chatbot that can comprehend English spoken with an Indian accent and operate on a desktop system is required.

When a chatbot is unable to provide appropriate answers, it's usually because it lacks the necessary context or doesn't comprehend the question's intent. Only via rigorous optimization, combining both humans and machine learning is it capable of providing pertinent answers to inquiries. Constantly establishing effective quality control methods will aid in managing the potential for the bot to learn undesirable negative behavior For them to operate well, a lot of information must be input into them.

Complex task dependencies should be modelled by chatbots, who should then use these models to suggest the user's best course of action. When a task includes several sub-tasks and each sub-task can have its own sub-tasks, it needs to be tested for finding the best paths. There may be several viable options in such a situation, and the system should be able to take the user's preferences, other ongoing tasks, and priorities into account before recommending a specific course of action.

Solution:

Personal assistant software is required to act as an interface into the digital world by understanding user requests or commands and then translating into actions or recommendations based on agent's understanding of the world.

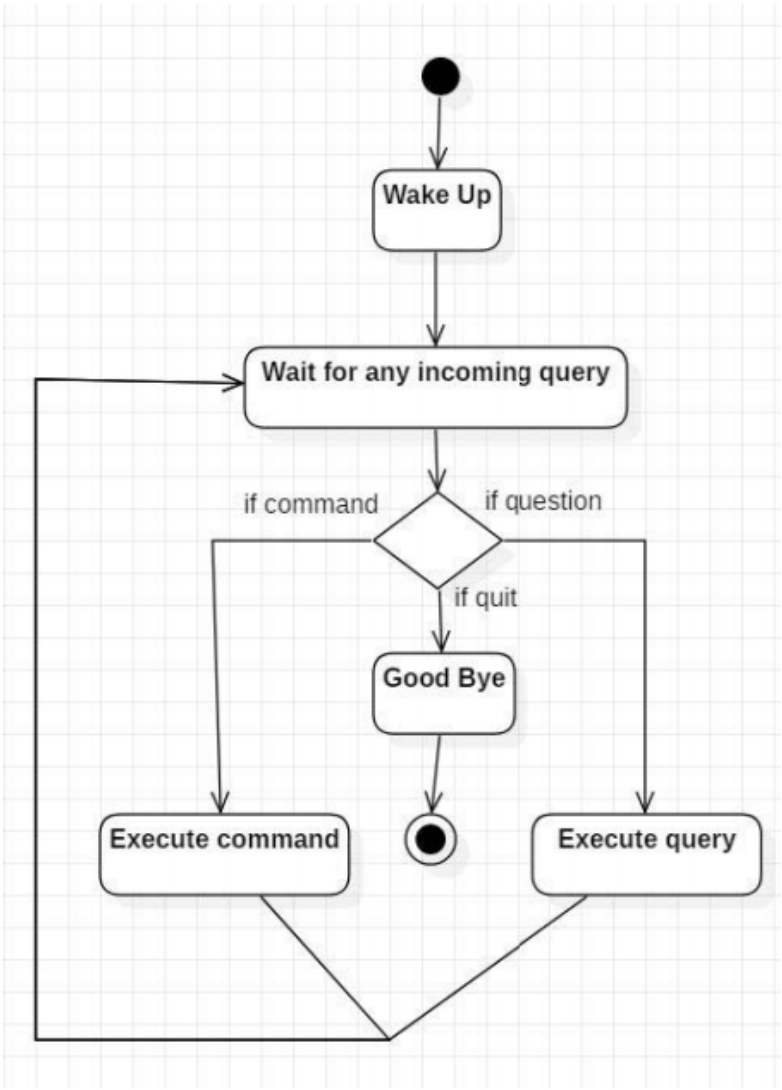
Evoque focuses on relieving the user of entering text input and using voice as primary means of user input. Agent then applies voice recognition algorithms to this input and records the input. It then use this input to call one of the personal information management applications such as task list or calendar to record a new entry or to search about it on search engines like Google, Bing or Yahoo etc. Focus is on capturing the user input through voice, recognizing the input and then executing the tasks if the agent understands the task. Software takes this input in natural language, and so makes it easier for the user to input what he or she desires to be done.

Voice recognition software enables hands-free use of the applications and lets us to query or command the agent through a voice interface. This helps users to have access to the agent while performing other tasks and thus enhances value of the system itself. Evoque also have ubiquitous connectivity through Wi-Fi or LAN connection, enabling distributed applications that can leverage other APIs exposed on the web without a need to store them locally.

Evoque provides variety of services. These include:

- Providing information such as weather, facts from e.g. Wikipedia etc.
- Set an alarm or make to-do lists and shopping lists.
- Remind you of birthdays and meetings.
- Play music from streaming services such as Saavn and Gaana.

• Play videos, TV shows or movies on televisions, streaming from e.g. Netflix or Hotstar.
• Book tickets for shows, travel and movies.



The user sends command to Evoque in audio form. The command is passed to the interpreter. It identifies what the user has asked and directs it to task executer. If the task is missing some info, Evoque asks user back about it. The received information is sent back to task and it is accomplished. After execution feedback is sent back to user.