

IVR system with Multimodal Interaction

Likhitha(IMT2014030)

Nikunj Gupta(IMT2014037)

February 10, 2018

Contents

1	Introduction	2
2	Approach	3
	2.1 Most frequent topic	3
	2.2 Short codes	3
3	System Architecture	3
	3.1 Work Flow	4
	3.2 Tools	5
4	Evaluation and Results	6
5	Future Work	6
	5.1 Qualitative inference engine	7
	5.2 Dialog systems	7
6	References	7

1 Introduction

The IVR Systems have changed the way we have been communicating with call centres. Yet, it is quite frustrating to stay online and listen to the same set of options again and again. For cases where the user wants to repeat the task he had accomplished the previous time, it will be monotonously annoying to traverse the same path on touch tone system every time he calls. Also, there are users who just want to talk to the executive, then why do they have to traverse various paths to find where they could talk to some human. Rather, it would be so nice, if he could just say "talk to executive" and the system directly takes him to the node where he gets to talk to an executive of the call center.

The primary focus of any business idea is to enhance customer satisfaction. The Interactive Voice Response systems serve as a bridge between people and the company database connecting telephone users with the information they need, from anywhere at any time. These systems have already been making an impact since a decade and their usage is evident in movie ticket booking, food delivery applications and telecom operator services. IVR systems have evolved from touch-tone or dual-tone multifrequency (DTMF) user interface to text and speech input.

But, a survey revealed that 83 percent of respondees viewed IVRs as having no benefit to the customer. The trade off between usability and functionality results in cumbersome, time-consuming and infuriating user interfaces. Complicated IVR tress result in tedious selection from multiple options available. The traditional interfaces not only have a bias towards technically skillful users but also seem to be unable to prioritize the needs of people in this fast-paced society. An inefficient IVR system may drive customers to unnecessarily select live agent support over self-service, raising inbound call volume, prolonging average handle time and poorly affecting customer experience.

This project is an attempt to make such things easier for the frequent users of the touch-tone systems. The user must be able to save some tasks so that, in future, when he calls again, he can just ask for the previous accomplished tasks to be iterated again. Also, we would like our users to be able to speak out the names of their short cut codes, rather than remembering and typing some number assigned to it. In addition to that, the system must be made intelligent enough to suggest the most frequent nodes to the callers. All this is aimed to make the touch-tone system more easy-to-use and less frustrating. The multimodal interface streamlines the use of an IVR application enabling the possibility to interact by voice or fingers-typing text and reduce the waiting time hence ensuring better customer satisfaction using previously saved paths and most frequent paths.

2 Approach

It is desirable to reach a node in the IVR tree as fast as possible; involving minimum frustration. In order to address the scenarios where the user would like to save the previously achieved tasks or for the engine that could suggest the callers a path (task) which is currently most visited, the following features have been implemented.

2.1 Most frequent topic

This is often the path that is most frequently accessed by users or the path the company wants most of its users to traverse. Hence, the most frequent path is statistically set and users can have direct access to it.

2.2 Short codes

If a user is a frequent caller and also wants to traverse the same path multiple times, he can save that path as a short code. A user can save multiple short codes with different names and use them whenever required. Using these short codes will save a lot of the on call time and of course, the frustration too.

3 System Architecture

The system works on python and allows multimodal inputs. The multiple modes are keyboard input and speech. The caller is prompted with questions which he can listen and answer in the form of voice or keyboard input whichever he finds comfortable. The prompts are spoken out (similar to that in existing systems) by the system using the Playsound library in Python and corresponding speech input is accepted using Google Speech to Text API calls. The approach defined above is implemented as follows:

Most frequent topic For suggesting the most frequent path to a user we came up with storing a 'global frequency' file for all users, which accumulates the count of the number of times a node is hit in the IVR tree. The path with the maximum number of hits at each level is finally suggested to the caller as the most frequent path.

Short codes Users can save their short codes as they wish and select them when revisiting. These paths are saved in a dictionary as key-value pairs for each user. A user can use multimodal input system to achieve his motive of traversing an already saved path; he just has to type 'y' for saying yes to using short codes and then speak up the name of the short code he wants to run. For suggestions of new short codes to users by the system, we maintain a 'frequency' array in a file for each user which maintains the frequencies of the nodes traversed by the user. Judging by these frequencies we can suggest a new path to the user at the beginning of the call.

3.1 Work Flow

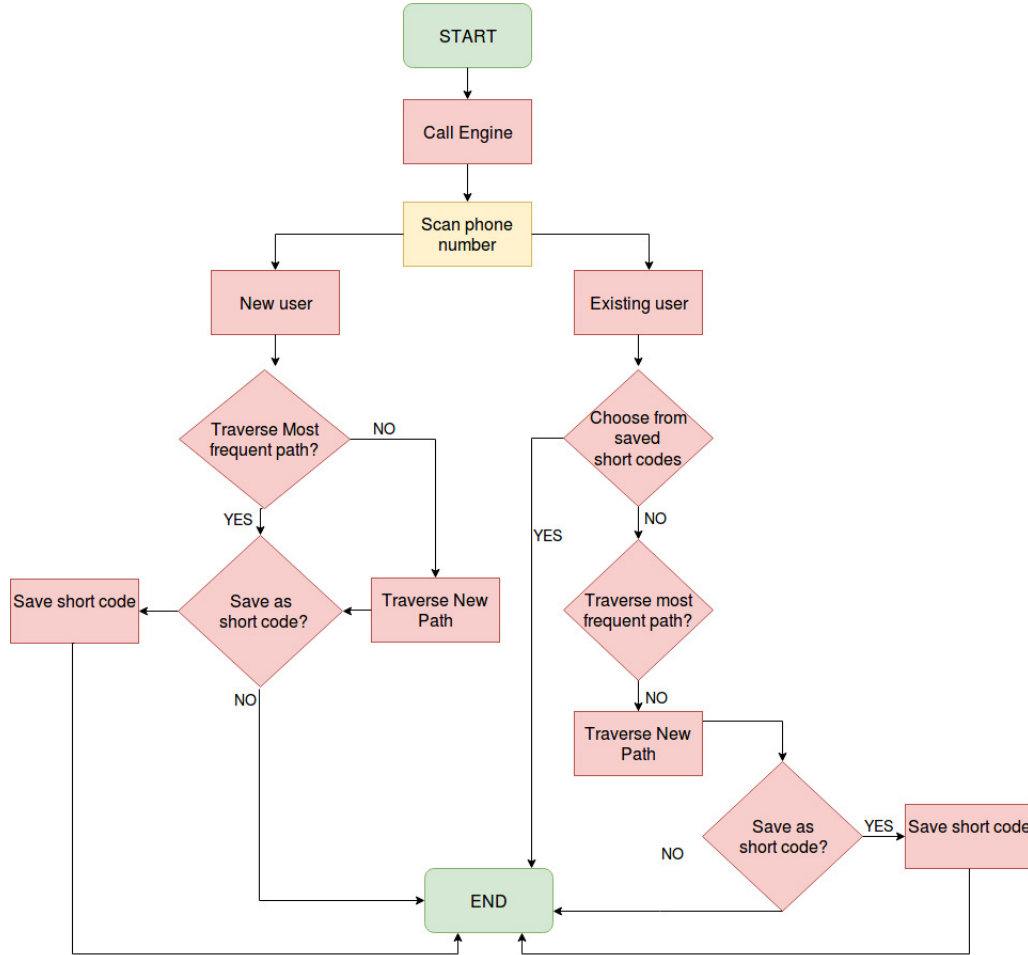


Figure 1: Functional Work flow

The engine involves three important structures. Firstly, the set of users who would want to call the engine. Secondly, the IVR system that interacts with the user via questions and input answers. And thirdly, the storage, where the frequencies and short codes are stored which are used to make the lives of the users easier. All these are displayed in the diagram above.

Figure 1 shows the **step-wise flow** of our engine (explained below).

- **Step 1:** A user calls the system.
- **Step 2:** The engine checks his details in the storage file. He/She could

be a new caller or a revisiting one. Given that he is a

- **New user:** He is given an option of pressing y or n to answer the question "Do you want to hit the most frequent topic?" If he presses
 - * **Yes:** The system chooses options corresponding to the most frequent path on his behalf thus giving the user his desired output as soon as possible.
 - * **No:** the user is let to choose options for his new path that he wants to traverse in the IVR tree.
- **Existing user:** As he is a revisiting user, he would have presumably stored a short code for himself the previous time he had called. So, he is given an option of pressing y or n to answer the question "Do you want to run your short codes?" If he presses,
 - * **Yes:** The system lists the short code options stored by the existing user over a period of time. After the user inputs the short code he wants, the system traverses the corresponding path on his behalf thus saving his time which would have gone in reiterating the same path on his own.
 - * **No:** Again, the user is let to choose options for his new path that he wants to traverse in the IVR tree.
- **Step 3:** Whatever path the user takes, he is asked whether he wants to save that as his short code or not. If yes, then he is asked to speak out a name for that short code and then it gets saved. If he says no, the the program gets over without saving this path which he had taken.

3.2 Tools

This is a Python based project. Some of the tools used were:

- Google Speech Recognition library
- gTTS Python package
- Python Playsound
- IVR tree (using tree data structure)

4 Evaluation and Results

SCENARIO	USER TYPE	INTERACTION TIME (in min)
Check mobile balance and validity	New user	2 min 55 sec
	Existing user	23 sec
Installation of data card /dongle on Windows(longest path)	New user	4 min 12 sec
	Existing user	26 sec
Details about special call rates and full talk time rates(shortest path)	New user	1 min 51 sec
	Existing user	25.3 sec
Most frequent path	Any user	27.84 sec

Mean interaction time	New user	2 min 21sec
	Existing user	25.53 sec

Table 1: Results

Table 1 shows the results of interaction time with our IVR engine. We simulated for 2 kinds of users, namely- new and existing, for accomplishing 4 different tasks. The results show that the mean interaction time for an existing user utilizing our short codes feature is 81.8% faster as compared to the time he would have taken in the conventional IVR system.

Our project is an attempt to achieve the following:

- *Reduced operational costs:*
 - The short codes feature implemented enables easier traversal of path avoiding taking the user input again and traversing the entire path from the beginning.
 - Most frequent path is directly accessible to the users avoiding several re-computations during the peak time of the server.
- *Better customer satisfaction:*
 - Personalised short codes give the user flexibility as well as hassle free usage of the IVR system.
 - Any customer can get the context of the most frequent topic and easy access to it through this system.

5 Future Work

There is need for an improvement of this engine in the direction of natural language processing as well as understanding the intent of the user in a better

way.

5.1 Qualitative inference engine

This involves understanding the utterance of the word and mapping it to semantically closest possible node and take care of the qualitative jargon involving words like 'little', 'much more'.

5.2 Dialog systems

Often when a task is extended to more than a sentence, the effectiveness of the system is reflected in remembering the context which the user intends. Dialog systems enforce a context centric environment.

6 References

[1] IVR Tree Reference:
<https://telecomtalk.info/detailed-analysis-on-customer-care-ivr-of-airtel-infographic/111313/>