Voice Based chatbot for open ended data collection

Introduction:

In this project, we aim to build an app which conversates with the user in a friendly manner and collect data(Employment Survey). For doing this, we take help of below mentioned softwares. Finally, we built an app which takes employment survey from a user in a conversational manner, in the same way 2 people talk on their phone.
We had very diverse app users such as guards in a hostel, mess workers, night mess people, teachers, retail shopkeepers etc. Their age lies in the range of 22-54 year. Their educational qualifications varied from 10th pass to post-graduate level. Most of them were married. The responses we got from them helped us to get a holistic view of how people answer a survey, and we had to train our DialogFlow agent according to the responses. The varied range of surveys conducted helped us to identify how people answer a particular question based on their educational qualifications, age, etc.

- **DialogFlow**:This is a google owned developer of chatbots which aims to build natural and rich human-computer interaction experiences.This is built entirely on Google infrastructure and is powered by Google's machine learning which makes it one of the best chatbot creations platform.Entire DialogFlow works on the principle of intents and entities.Entities are Dialogflow's mechanism for identifying and extracting useful data from natural language inputs.An *intent* categorizes an end-user's intention for one conversation turn. While intents allow your agent to understand the motivation behind a particular user input, entities are used to pick out specific pieces of information that your users mention -- anything from street addresses to product names or amounts with units.For eg. User Utterance-"Book a flight to New Delhi",Intent is to book a flight and entity is New Delhi.Our main aim was to build a conversational chatbot and DialogFlow's entity and intent makes it best suitable for the same.
- **Android Studio**-It is an IDE from Google that helps developers to build applications for the Android OS platform. We had to use it to incorporate the DialogFlow agent into an app which is very handy and easy to use. Using the help of https request, we have been able to integrate the DialogFlow agent into the app easily. Many versions of the app were created like; initial one opened a new browser window for the DialogFlow agent, another opened it in webview.And, then we have the final version where in, the HTTP request is sent(in JSON

format) from the android app to DialogFlow, and the response is also received from DialogFlow as JSON format. In the final version of the app, only the user utterance and response is displayed over a blank screen, and the app speaks out the response received from Dialogflow agent.so, it will be more like the conversation between two users talking on the phone. A button is provided in the final version of the app to start the conversation between the user and chatbot.The app also saves entire conversation transcript in a text file named (“survey”+starting time of survey).txt .

- **Google Speech to text API**: Google provides for two types of speech to text API-one is the Regular API, and another is the Streaming API. Regular API can be used to convert local audio files stored in the device while Streaming API can be used to convert both local audio files as well as streaming audio from the microphone. We had measured the latency time of both the APIs to transcribe local audio file(short files with less than 30s duration) and found out that Streaming API takes around 300-400ms while the Regular one takes about 700-800 ms to transcribe the same files. We also found out the latency time of DialogFlow agent using an app developed by us, and it came to be around 600-700 ms. This motivated us to explore a way by which we could record the audio files(of user utterance) and then use the Google’s Streaming API to convert the audio file to text and then give this transcript as input to the DialogFlow agent. This new method could have helped us to decrease the latency time very significantly and improve the usability of the app. But, alas! We could not find any way out to use Streaming API in Android Studio. We tried everything available online and even posted our query on all major Android Developer’s forum but to no avail.

Analysis of Responses:

Q1.) नमस्ते! कृपया अपना नाम बताएं

->All the people were very comfortable with this question as this was a pretty straight forward one and did not require any knowledge whatsoever. There were a different type of responses such as some of them just speaking out their name like “दिनेश कुमार”, “आशुतोष”, while some of them answered it like “मेरा नाम मोहित है”.

Q2.)क्या मैं आपका लिंग जान सकता हूँ

->This was also a simple question like the first one and responses received were like “मैं पुरुष हूं”, “पुरुष”, “मेल” which was pretty much expected to standard questions like these.

Q3.)कृपया अपनी उम्र बताईये
->Again a standard question, most people answered in Hindi but some people preferred to answer in English like “थर्टी”, for which we had adequate provisioning. We had accommodation for an answer like “29 साल”, “22 वर्ष”.

Q4.)आप कहाँ तक पढ़े हुए हैं
->Our target users had varied educational qualifications like some of the shopkeeper and teacher were graduate and postgraduate while guard and mess workers had a maximum qualification of 10+2 or intermediate or 12th pass. Initially, we didn’t have provision to accommodate “इंटरमीडिएट”, and so the app gave an error to the user who used this.

Q5.)क्या आप शादीशुदा/विवाहित हैं ,कृपया हाँ या नहीं में उत्तर दीजिये
->This was a simple binary question that required only “हां” or “नहीं” but some user replied as “मैं शादीशुदा हूँ” , this was not expected by us as this was expected to be a binary question but after this response we had to make provision for this question considering it to be a subjective question

Q6.)पिछले साल आप कितने दिन अपने क्षेत्र में रहे है
->In this question we have made provision to accept months, days and weeks.If a person says that “पूरा साल”, then the machine repeats “कृपया दोबारा बोलिये और अगर पूरा साल काम किया है तो एक साल बोलिये” and then the person has to reply “1 साल” to continue the conversation.

Q7.)आप के घर के आय का प्रधान श्रोत क्या है
->The accepted responses to this question were very limited and we had to add new intents in the DialogFlow model on the go. There were no major issues with this dialogue during the conversation.

Q8.)कौन से महीने में आपने यह काम शुरू किया था
->This was an ambiguous question as most of the people could not remember the exact month of starting their job. Most likely the responses given by the user were any month they wished. This question is especially relevant for the farmers and labourers whose employment is never fixed and varies according to season and demand. This case is more relevant for the farmer who sows different seeds during different seasons.

Q9.)आपके नियोगकर्ता का नाम क्या है ?
->This question was asked to only those people who were working as a self-employed. This was made possible by using follow up in DialogFlow.This question accepted the only name of people. One of our users who is a teacher answered Delhi Public School to this question and got the error there.

Q10.)आप दिन में कितने घंटे इस पर काम करते है
->Regular question but some user replied their duration as well as the timing of their work but this did not cause any error. But, some user gave only their workplace timing as the answer which was rejected by the DialogFlow agent

Q11.)क्या यह काम आप अपने क्षेत्र में करते है या कही और, कृपया एक उत्तर दीजिये, अपने क्षेत्र में या फिर कहीं बाहर
->This was also expected to be a binary question but user gave some replies as "कहीं बाहर मतलब थोड़ा दूर घर से", which was also expected by the app. Apart from this, other responses were pretty straight forward to this standard question.

Q12.)इस काम का यह क्षेत्र ग्रामीण है या शहरी
->This was also a binary question but there was a very peculiar error occurring that sometimes a person was trying to say "शहरी" but the google speech to text converted it into "शायरी" which raised an error. But this was not such a thing that could be overcome by us. This was google speech to text's error but apart from this single mistake, otherwise speech to text was working absolutely fine.

Q13.)आप दिहाड़ी पर काम करते है या प्रति पीस के दर से
->We could not get any response regarding this question as this was asked only if the person was a casual labourer.

Q14.)हर दिन आप को कितना पैसा मिलता है (per month for salaried)
->This was also a disputed question as some of the people gave their annual salary or annual profit. This has been corrected in the latest version of the app.

Q15.)आप को सामान के तौर पर क्या क्या मिलता है [Options: paddy, wheat, barley, maize, gram, other]
->We could not get any response regarding this question as this was asked only if the person was a farm labourer either casual or long term.

Q16.)क्या आप के नियोगकर्ता आप को खाना या नास्ता उपलब्ध करवाते है
->This was also very straight forward and binary question and the responses also indicate the same. People either responded in “हां” or “नहीं” , there was not an error in this step of the process.

Q17.)क्या यह काम मनरेगा/ प्रधान मंत्री आवास योजना या फिर किसी और सरकारी सहयता वाली रोज़गार का हिस्सा है
->This was also very straight forward and binary question but some user responded as “सहकारी सहायता” and “सरकारी” which the app failed to recognize and gave the error.

Q18.)क्या इस दौरान आप दूसरे नियोगकर्ता के साथ काम करने के लिए खली रहते है
Q19.)क्या आप के घर के दूसरे सदस्स्य भी आप के नियोगकर्ता के साथ काम करते है
Q20.)(if salaried):क्या सेवनिबृत्त होने के बाद आप को पेंशन मिलेगा
Q21.)(if salaried):क्या आप के नियोगकर्ता बिना नोटिस के आप को काम से निकल सकते है

->The above 4 questions were very straight forward and required an only binary response in the form of “हां” or “नहीं”.There was not even a single error in answer corresponding to these questions.