IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI POWERED CONVERSATIONAL SHOPPING AND HUMAN PLUS BOT HYBRID SUPPORT

[1]Rishindra Mani Katiyar [2]Shivmangal Singh Yadav [3]Ashu

[12]Student,[3]Assistant Professor, School of Computer Science & Engineering

Lovely Professional University, Punjab, India

Abstract: *The present online shopping experience can be enriched by following a natural conversational approach. Moreover, the number of customers is very large compared to the human resources of any organization, the pressure imposed on the executives is overwhelming. The stress often leads to degradation of customer's experience and decreases the quality of service. To minimize the waiting time of the enquiries and provide a hassle-free shopping experience to the customers, we need some effective way to deal with the problem. So, an AI Powered Conversational Shopping for Smart Cities along with the integration of Human plus Bot Hybrid Support would serve the purpose of streamlining the overall shopping experience of the customers.*

1. INTRODUCTION

A conversational solution approach for ordering and discovering products is a much better approach to understand the customer's requirements. Here chatbots come in very handy. Chatbots bridge the gap between online and offline experiences. Moreover, chatbots can deliver the same shopping experience to shoppers that they might expect from a brand either in-store or online. Therefore, making a chatbot would make it easier for the administrator to handle customer conversations. Next is gamifying the ordering experience so that customer is engaged right till the end and is never exhausted of options while ordering products. Additionally, providing a hassle-free checkout experience in the end would make conversational shopping much more reliable. A custom bot hybrid support is set up to establish a 24/7 connection between customer and end-user.

2. OBJECTIVES

We are trying to bridge the gap between online and offline experiences. The conversational solution helps in intensifying user engagement on any platform and keeps the user engaged providing a gamifying experience overall. The paper aims to fulfil the following objectives:

- AI Powered Conversational Shopping for users, a one-of-a-kind experience
- Provide Human Plus 24x7 Bot Hybrid Support
- Bridge the gap between online and offline shopping experience.
- Increasing Customer Engagement by gamifying ordering experience.
- Utilizing payments api to implement secure payments and faster checkout.
- Extending Dialogflow functionalities and making public consumable APIs.

3. LITERATURE REVIEW

In history, Conversational commerce started when apps such as Facebook Messenger, WhatsApp, Slack and others, decided to interact with customers simpler without needing to go to a particular website or phone app. These apps became popular because of their faster access and easy user interface, and they continue to have potential customers who want to use them. So, a need for AI Powered Conversational Shopping was felt with a bot hybrid support ready to respond to customer interactions.

Rajasshrie Pillaia, Brijesh Sivathanub and Yogesh K. Dwivedi in [1] have realized that the IT acquisition model used, incorporates a Technology adoption model and technological readiness to create an integrated model called Technology readiness and Acceptance Model (TRAM) to provide a better understanding of AI-powered automated retail stores (AIPARS) customer

acquisition objectives. The TRAM model has been used in various customer acceptance studies for new technologies. Customers with good technology access have the right attitude to use new technologies and are more comfortable accepting the technology.

4. THEORETICAL ANALYSIS

4.1 Architecture Overview of Conversational Solution

To analyze how the conversational solution works and the intents get called over functions, a systematic flowchart is shown in which the architectural overview of the conversational solution is represented. The way the intents are mapped and then called by the cloud functions is explained with the help of a logical diagram in Fig 1

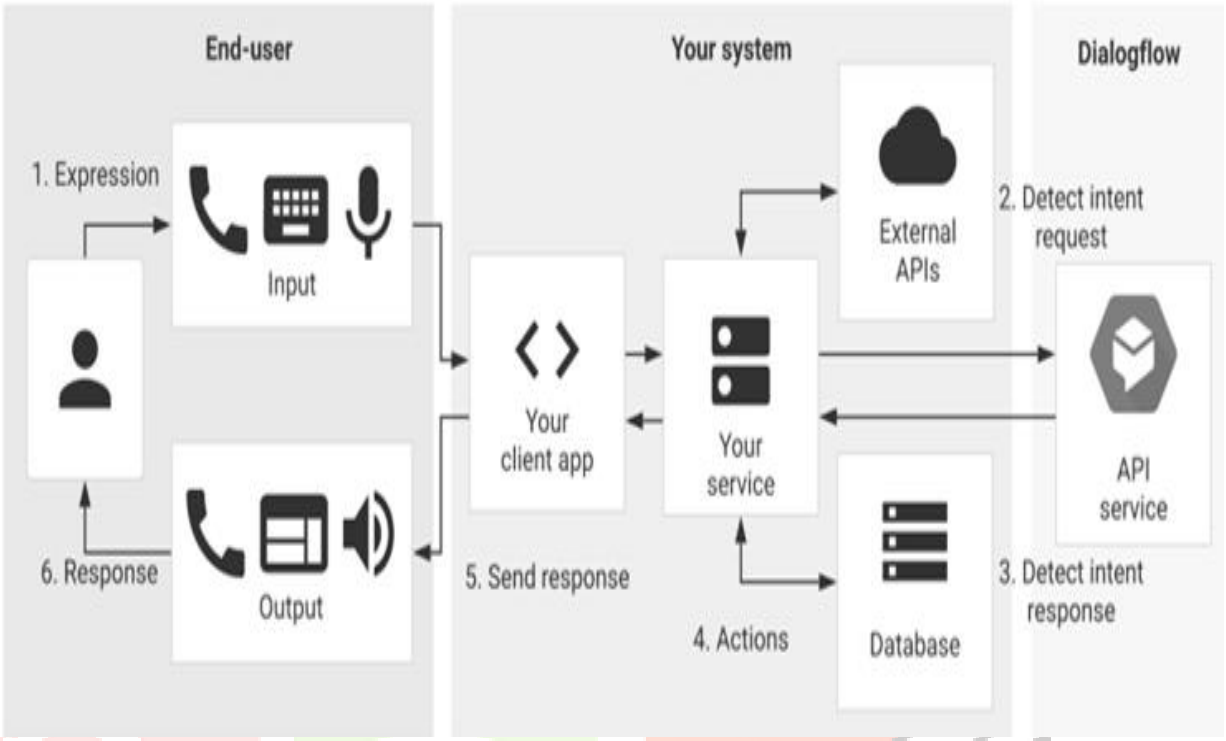

Fig 1

4.2 Dialogflow

Dialogflow is a program that concentrates on the significant part of the responses that users say. It's a characteristic language cognizance stage that can be utilized to make and fuse a conversational UI into versatile applications, web applications, coordinated voice reaction frameworks, and different applications. Dialogflow comprehends the plans and uses webhook calls to send them to the backend. Dialogflow is more like a front end that serves as a collector of user responses and then through its natural language understanding algorithm it maps the user's conversations to appropriate intents.

4.3 Firebase and Node.js

It is utilized as a BaaS (Backend as a Service) since it assists in making incredible client specific responses. You don't have to manually look upon the functions in intents and deploy a separate server. Firebase is the server, your API and your datastore, all composed so conventionally that you can adjust it to suit most requirements. Cloud capacities in firebase are answerable for producing reactions from the bot that you have found in the ser actions. At whatever point client give any assertion like "Continue to checkout" dialogflow parse the fitting goal for it and calls the cloud capacities. The cloud capacities are only JavaScript running at server side, more properly known as Node.js. At the backend side a controller is available for each plan characterized in the dialogflow support with which client can interact. So, when user says, "Continue to checkout", the capacity identified with that is conjured, and afterward with google sign in checkout is carried out. Using basic state management tactics to calculate price of order, an email is sent to user with SendGrid api. Similarly, other appropriate intents are invoked for responses. Talking about live stocks, the customers would be able to check the availability of the product using a simple voice command. If he/she wants to buy products in bulk, it would also tell maximum feasible quantity.

4.4 Integrations

Dialogflow incorporates with numerous famous discussion stages like Google Assistant, Facebook Messenger, Telegram, Slack, Line, etc. The bot can be coordinated into any site and surprisingly constant correspondence with the bot is conceivable through the communication administration. Direct end-client associations are naturally taken care of so the specialist could be worked by our necessities without agonizing over the adaptability of the mixes. Every combination handle end-client cooperation's in a stage explicit way so that there is no compelling reason to modify the code for each new mix.

We have integrated Kommunicate to give clients constant, proactive, and customized support directly from the bot window itself. Kommunicate is a human + bot cross breed client assistance programming for developing organizations. With Kommunicate, we can oversee client discussions, site talk, support specialists, group discussions, client viewpoints, etc. all at a similar spot without changing the conversational progression of the model. The conversational flow is represented with a diagram:

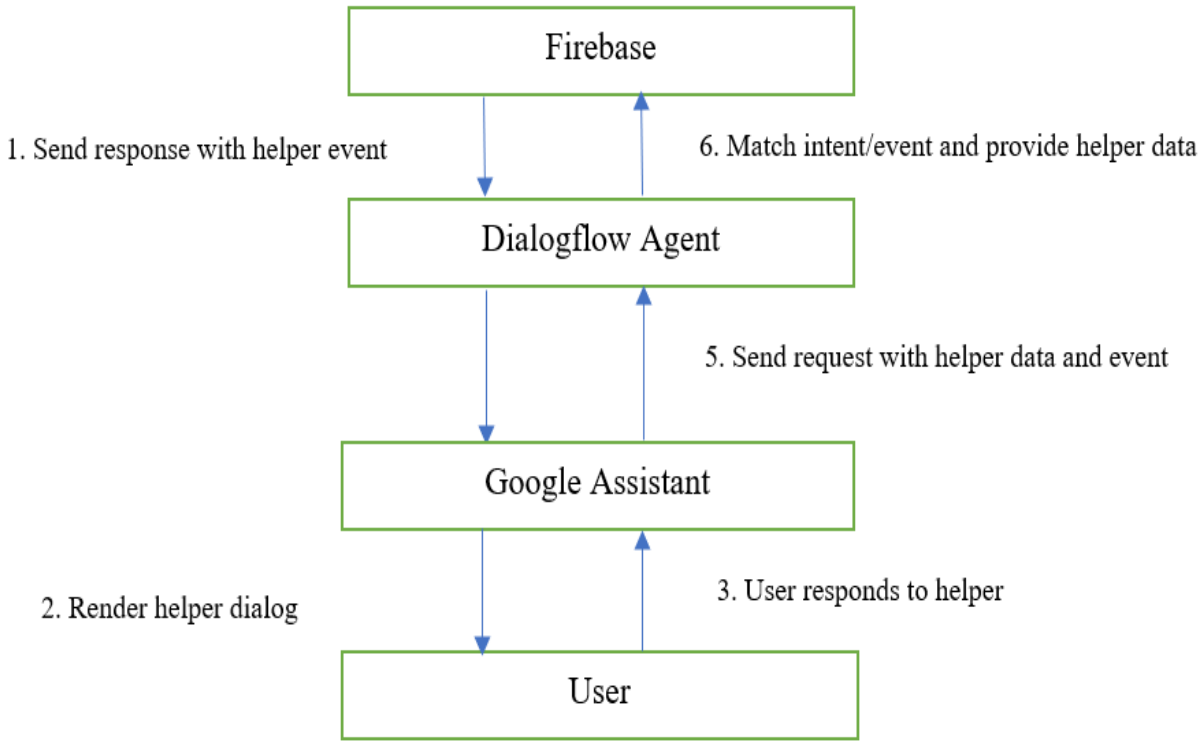

Fig 2

4.5 Potential Enhancements

- We are already storing user data on firebase. So, implementing order tracking system would not be difficult. The delivery person would update the status of the product and the user would get a push notification to his device. Talking about live stocks, the customers would be able to check the availability of the product using a simple voice command. If he/she wants to buy products in bulk, it would also tell maximum feasible quantity.
- A lot of filtering options would make it easier for the user to narrow down the results like selecting the price range according to their budget, and we are planning to incorporate the same into our product. We can do that by using the concept of indexing in NoSQL database designs. As we are using firebase's database which is a NoSQL database, we can create different types of composite index for efficient querying to database. Like if I need to filter some results in some specified price range and specific brand; filter of price category can help to do that. We are also planning to give recommendations of similar products to users based on the user's old conversations and interests in products. We simply keep track of users' preferences in the database, and whenever there is some sale or brand-new products added in that category firebase cloud functions can send an email and push notifications to users.
- The bot could also be used to book demos, schedule appointments and reservations and even open a live chat with the customer support agent or direct to a call according to the customers priority. Filtering and directing conversations based on rules and conditions of answers would be a further enhancement in providing telephonic support to the customer and would help in achieving 24-hour assistance 365 days a year. Once the chatbot would display a card showing the total amount to pay after checkout all they would have to do is click on the pay button and they would be redirected to the appropriate payment gateways. Integration of payment services like stripe and others would serve a great purpose in further streamlining customer's shopping experience.

4.6 Deploying Cloud Functions to Firebase:

For making the platform interface we will use Firebase CLI to deploy our cloud function. You must have node v6 installed on your machine since cloud functions are compatible with node v6. Now you need to install firebase tools. So, install nodejs and npm. $ npm install -g firebase-tools. After this initialize firebase cloud functions library. Login to firebase tools if you haven't logged in before. $ firebase login Once you're logged in to firebase, run the next command. $ firebase init functions

This command is a command-line wizard that will guide you through a process which associates your firebase function with your Google Cloud project that gets created when you created 'Actions on Google' project i.e. part of the deployment. Now chose a default Firebase project for this directory. Note that you should select the correct project ID. The project must be deployed to the same agent that is associated. When asked what language to use, select JavaScript using the arrow keys. The wizard will prompt: File functions/package.json already exists. Overwrite? File functions/index.js already exists. Overwrite? Enter No for both cases. The code will be overwritten in case you select 'yes' option. If this happens, clone the repository again. After that install dependencies with npm now. Choose Yes and let the wizard finish initialization. Once the association is done, next deploy the cloud function. Execute the following command by going to the root directory. $ cd functions && firebase deploy. After the deployment is done you will get an endpoint URL, it is the webhook endpoint that Dialogflow will use to make requests to. Paste the URL in the Fullfilment section of your Dialogflow agent. Now go ahead in actions console and test your application.

Fig 3

5. CONCLUSION

This paper is an attempt to take conversational shopping experience to a whole new level. The paper discusses the technology and conversational flow that needs to be followed to make an enriching conversational experience. The Bot Hybrid Support along with the AI powered conversational solution will help in increasing the engagement of users on any platform. An attempt is made to bridge the gap between shopping experiences in offline and online mode and gamify the order experience of the customers to provide a hassle-free and user-friendly experience. The paper tries to leverage the power of AI to build and create conversations that will imbibe the new era of conversational solutions and help in creating applications that will increase user engagement and can better cater to end-user needs.

REFERENCES

[1] Shopping intention at AI-powered automated retail stores (AIPARS) by Rajasshrie Pillaia, Brijesh Sivathanub and Yogesh K. Dwivedi DOI: 10.1016/j.jretconser.2020.102207

[2] A Dynamic Speaker Model for Conversational Interactions by Hao Cheng, Hao Fang and Mari Ostendorf DOI: 10.18653/v1/N19-1284

[3] Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems, by Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, Pascale Fung arXiv:1905.08743 [cs.CL]

[4] Do Neural Dialog Systems Use the Conversation History Effectively? An Empirical Study, by Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio arXiv:1906.01603 [cs.CL]

[5] Persuasion for Good: Towards a Personalized Persuasive Dialogue System for Social Good, by Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu arXiv:1906.06725 [cs.CL]

[6] Approximating Interactive Human Evaluation with Self-Play for Open-Domain Dialog Systems, by Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Agata Lapedriza, Rosalind Picard arXiv:1906.09308 [cs.CL]

[7] A Dynamic Speaker Model for Conversational Interactions by Hao Cheng, Hao Fang and Mari Ostendorf DOI: 10.18653/v1/N19-1284 Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems

[8] Approximating Interactive Human Evaluation with Self-Play for Open-Domain Dialog Systems, by Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Agata Lapedriza, Rosalind Picard arXiv:1906.09308 [cs.CL]