AN IMPLEMENTATION OF VOICE ASSISTANT FOR HOSPITALITY

Sagina Athikkal and John Jenq

Department of Computer Science, Montclair State University, NJ USA

ABSTRACT

Voice user interface has gained popularity in the recent years. A chatbot is a machine with the ability to answer automatically through a conversational interface. Instead of using mouse and keyboards as input and screen as output, a chatbot with extra voice user interface feature improve the system and enhance the user experience. A chatbot is considered as one of the most exceptional and promising expressions of human computer interaction. Voice-based chatbots or artificial intelligence (AI) devices transform human-computer bidirectional interactions that allow users to navigate an interactive voice response (IVR) system with their voice generally using natural language. In this paper, we focus on voice based chatbots for mediating interactions between hotels and guests from both the hospitality technology providers' and guests' perspectives. A hotel web application with voice user interface was implemented which provides voice input/output interface to enhance the user experience. Speech recognition component was used to dictate the user voice input to text. Speech synthesis API was used for text to voice conversion. A closed domain question answering (cdQA) Natural Language Processing (NLP) solution was used for processingof query and return the best answer possible.

KEYWORDS

Natural Language Processing, Chatbot, Voice Based Digital Assistants, Closed Domain Question Answering.

1. INTRODUCTION

In the advanced of computer technologies, computer systems are more and more like human. The interface between human and machine evolved from text oriented input to graphical user input to now a popular voice user input. Although all these interfaces coexist and are using today, voice user input/output systems are gaining popularity. It is very useful especially when using hands to interact with computer systems are inconvenient or impossible. Virtual assistants require implementation of voice user input/output interface subsystem to communicate with users. Popular big player of voice assistants on mobile phones and computers are Siri, Cortana, Alexa and Google Assistant. Siri was introduced by Apple as standalone in 2011 and iOS in 2011. It is the oldest and has been around the longest among these four popular assistants. Cortana was introduced by Microsoft in 2013. Amazon's Alexa home speaker was introduced in 2014, and Google's Assistant was introduced to its Pixel phones in 2016. Because natural language is a context sensitive languages, it poses great challenges to researchers and developers to create a system which can understand the meaning of the speaker and respond correctly. In addition to the language understanding research, other research areas include language translation from one language to the other, user voice data privacy and security, in-vehicle user voice commands implementation, voice authentication among many others.

DOI: 10.5121/sipij.2022.13401

A chatbot [1, 2, 3, 4, 5] is a programming interface that simulates the conversation or "chatter" of a human being through text or voice interactions. Nowadays, chatbots are available in almost many aspects of technology, such as mobile assistants, customer services, e-commerce, and smart devices. It is a type of software which can help the users by automating their conversations and interact with the customers through the messaging platforms. These chatbot-virtual assistants are found useful to handle simple, look-up tasks in business-to-consumer and business-to-business environments. Chatbot virtual assistants are helpful not only to make use of support staff time but also beneficial in providing a level of customer service when the supporting agents aren't available [6]. Chatbots interpret and process user's words or phrases giving them an instant pre-set answer [7]. The most important aspect of implementing a chatbot is selecting the right natural language processing (NLP) engine. If the user interacts with the bot through voice, for example, then the chatbot requires a speech recognition engine. Similar to regular apps, chatbots also have an application layer, a database, APIs, and Conversational User Interface (CUI)[7]. There are structured and unstructured conversations. Chatbots built for structured conversations are highly scripted, it simplifies programming but restricts the kinds of things that the users can ask. In most B2B environments, chatbots are commonly scripted and used to respond to frequently asked questions or perform simple, repetitive calls to action. In sales, a chatbot may be a quick way for sales reps to get phone numbers. For service departments, it assisting service agents in answering repetitive requests. Generally, once a conversation gets too complex for a chatbot, the call or text window will be transferred to a human service agent.

Although recent applications of chatbots are being used for various fields such as business, education, and healthcare, the first chatbots were mainly for entertainment purposes. Chatbots such as ELIZA and PARRY were early attempts at creating programs that could at least temporarily fool a real human being into thinking they were having a conversation with another person. ELIZA's method of imitating human conversation was to divert attention to participants by returning the user's responses in question format. Although ELIZA's ability to generate appropriate responses was not great, it did take spectators by surprise as this was the first of its kind in chatbots. ELIZA was so convincing to people at the time that they began to disclose personal information and secrets with the chatbot. The creation of the ELIZA starts the era of the development of more and better chatbots [8]. PARRY was developed in 1972, by a psychiatrist and computer scientist named Kenneth Colby, to imitate a patient with paranoid schizophrenia [9]. PARRY's effectiveness was benchmarked by using a version of a Turing test; testers only made the correct identification of a human versus a chatbot at a level consistent with making a random guess.

Chatbots have come a long way since then. They are built on artificial intelligence (AI) technologies, including deep learning, natural language processing and machine learning (ML) algorithms, and require massive amounts of data. The more an end user interacts with the bot, the better voice recognition becomes at predicting an appropriate response. There are different types of chatbots [10, 11, 12, 13, 14]. We can roughly classify chatbots into three categories: (a) Rule-based, this is the simplest type of chatbots. They require user to make a few selections, such as using drop downs or buttons, to give relevant answers. They are slow but is easy to implement. When many conditions or factors are involved in the knowledge base, this approach may not be the best solution [15]. (b) Intellectually independent chatbots: These chatbots learn from the user's inputs and requests by using Machine Learning. This kind of bots are trained in such a way to understand specific keywords and phrases that triggers bot's reply. They train themselves to understand more and more questions with practice and experience [15]. They spot keywords or phrases and provide predefined answer based on these spotted keywords or phrases. (c). AI-powered chatbots: It combines the best from the rule-based and intellectually independent chatbots. These bots understand free language and make sure they solve the user's problems with a predefined flow. They can switch the conversational scenario when needed and address random user requests at any moment. These chatbots use machine learning, AI, and Natural Language Processing (NLP) to understand and analyse human speech, find the right response and reply in understandable way in a human language.

Overall speaking, chatbots are considered as one of the most advanced and promising aspect of interaction between humans and machines. Chatbot applications helps in smoothening the interactions between the customers and the services [16]. They can enhance and engage customer interactions with less human intervention. More and more companies are moving to AI chatbots because of the advantage of time savings and efficiency which helps to increase sales or service productivity. Chatbots have the potential to enhance the customer experience by reducing customer waiting time and providing immediate answers, providing customers with 24/7 customer support, removing the threat of unpleasant human-to-human interactions that are dictated by the mood and emotions of both the service or sales representative and the customer. Chatbots are also useful in minimizing the stress and annoyance that some customers feel when having to contact customer support for example, by reducing wait time and streamlining the conversation, improving the redirection of customer queries, etc.

In [17], Dimitrios Buhalis and Iuliia Moldavska explained the importance of voice assistants in the hotel industry. They clearly mention the advantages of voice assistants in hotels outweigh the disadvantages for both hotels and guests. Their findings illustrate that voice-based human-computer interactions bring a range of benefits and voice assistants will be widely deployed in the future. Technology integrations are often complex and costly to set up but it provides significant benefits especially in hotel and tourism industry. As reported by them, guests appreciate the prospective benefits but are concerned with privacy and usability, although tech- savvy consumers are less concerned about privacy when using voice assistants. The findings indicated the direction for the future development of voice technology in hospitality towards multilingualism and modulated offers which can ultimately ensure the overall wider reach of the technology in the hotel industry.

Li, Bai et.al. [18], the authors reviewed a real-world conversational AI and NLP system for hotel booking. Their architecture design includes a frame-based dialogue management system that calls machine learning models for classification, named entity recognition, and information retrieval subtasks. Their chatbot has been deployed on a commercial scale, handling tens of thousands of hotels searches every day. They have also explained various machine learning models that they used for deployment and explained developing an e-commerce chatbot in the travel industry. Adam et.al. [19] describes the significance of chatbots in various fields. They explain how the use of ADCs (Identity, small talk, and empathy) as a common compliance technique, affect user compliance with a request for service feedback in a chatbot interaction. They have examined a randomized online experiment how verbal anthropomorphic design cues and the foot-in-the-door technique affect user request compliance. Their results show that both anthropomorphism as well as the need to stay consistent significantly increase the likelihood that users comply with a chatbot’s request for service feedback. They have commented that social presence mediates the effect of anthropomorphic design cues on user compliance. In [20], Hasan et.al. examine tourist chatbot usage intentions in service encounters within the context of a future international travel, assuming continued social distancing. Their results show that automation, habit, social presence, and health consciousness all contributed positively to chatbot usage intentions. Some variations were observed as a function of experiencing government-imposed lockdown. The role of social presence and human qualities in chatbots was weakened when controlling for lockdown and during the trip experience.

2. PROPOSED SYSTEM

Our proposed system recognizes speech on chatbot which uses NLP for interaction on any closed domain system. It uses latest technology scope forward and developed the application in internet.

Users can use voice to interact with the web application instead of searching and navigate the website. Any questions related to the hospitality or the hotel web application (like how to use, or where can I find it), or questions related to the business domain can get answered by the chatbot.

The system is trained to answer any question related to hospitality domain. The system is trained with any data set which has information related to the hotel website. It is implemented in a way so it can be easily retrained with any data set.

The proposed system requires following modules: a hotel web site which can host the voice chat bot. A database that contains tables that store the customers and administrators users. Since we are using chatbot in this application, we need a speech to text converting system in order to capture voice input and get the voice to convert into text. The text output from the speech recognition system will be served as the input to the question answering system. We design this question answering subsystem as an independent module so that we can implement it in different level of complexity as we need for the future research. In this paper, we implement it using cdQA approach.

For speech to text engine, there are various solutions for this. Here we will introduce two most used voice to text conversion approaches for web application.

2.1. Speech to Text Engine

Deep Speech is an open-source voice recognition and speech to text engine, which provide a trained model using Baidu’s Deep Speech research paper and the model implementation is under Mozilla Public license. The underlying implementation is using Google’s Tensor Flow. It come up with two models the acoustic model and the language model. Acoustic model is an end-to-end deep leaning system, and the language model is used to increase the accuracy of the transcription output which is included as separate model. The language model can be customized based on our domain.

To implementation it, we must download the model first. The .pbmm is the acoustic model whichis trained based on American English, in behind the scene it uses tensor flow. Scorer is the language model, which is useful for improving the accuracy of the predicted output. For example,using this, it will find out which word is grammatically right in a particular context.

The architecture of the engine was originally based up on Deep Speech: Scaling up end-to-end speech recognition. Currently it is different in many aspects and made it based on recurrent neural network (RNN) which is trained to ingest speech spectrogram and generate English text transcription. For the detail discussions and explanation of Deep Speech RNN model, please refer to [21]. Deep Speech RNN model use hybrid model for parallel optimization. Hybrid parallel optimization combines the benefit of asynchronous and synchronous optimization. It allows to use multiple GPUs but doesn’t have a problem of incorrect gradient present in asynchronous optimization.

In hybrid parallel optimization initially, it places the model in CPU memory. Then, as in asynchronous optimization, each of the G GPUs obtains a mini batch of data along with the current model parameters. Using the mini batch each of the GPUs then computes the gradients for all model parameters and sends these gradients back to the CPU. Now, in contrast to asynchronous optimization, the CPU waits until each GPU is finished with its mini batch then takes the mean of all the gradients from the G GPUs and updates the model with this mean gradient.

Hybrid parallel optimization has several advantages and few disadvantages. As in asynchronous parallel optimization, hybrid parallel optimization allows for one to use multiple GPUs in parallel. Furthermore, unlike asynchronous parallel optimization, the incorrect gradient problem is not present here. In fact, hybrid parallel optimization performs as if one is working with a single mini-batch which is GG times the size of a mini-batch handled by a single GPU. However, hybrid parallel optimization is not perfect. If one GPU is slower than all the others in completing its mini-batch, all other GPUs will have to sit idle until this straggler finishes with its mini-batch. This hurts throughput. But, if all GPUs are of the same make and model, this problem should be minimized.

So, relatively speaking, hybrid parallel optimization seems the have more advantages and fewer disadvantages as compared to both asynchronous and synchronous optimization. For discussion of DeepSpeech parallel optimization, see [22] for details. In this report, we use the hybrid model.

2.2. Speech Recognition API

In recent years, chrome version 25 came up with the Web Speech API embedded in browser which support conversion of voice to text conversion in web applications. It is getting popular and going to be the future for voice recognition [23]. Browser exposes the speech recognition feature via the Speech Recognition interface. This interface has an ability to recognize voice context from an audio input (normally via the device's default speech recognition service) and respond appropriately. We have to create Speech Recognition object using JavaScript that has a number of event handlers available for detecting when speech is input through the device's microphone. We can also check the browser’s compatibility using Webkit Speech Recognition present in browser window object. The Speech Grammar interface represents a container for a particular set of grammar that our app should recognize [24]. Grammar is defined using JSpeech Grammar Format.

2.3. Text to Voice Conversion

The output produced by the server will be in text format which need to be converted into voice. So, we need a solution to convert this text into voice in an efficient way. The voice response will provide an interactive feeling to the users. There are many options available in text to voice conversion, for example, many cloud hosted APIs, or standalone implementations using pre-trained models such as gTTS. Here we analysed two solutions such as gTTS and Web Speech API.

2.3.1. Text to Speech Engine Using Pre Trained Model

The text-to-speech (TTS) is the process of converting text data into a vocal audio form. The program takes an input text and using methods of natural language processing understands the linguistics of the language being used, and performs logical inference on the text. This processed text is passed into the next block where digital signal processing is performed on the processed text. Using many algorithms and transformations this processed text is finally converted into a speech format. This entire process involves the synthesizing of speech. We use Google’s Text To Speech (gTTS) library for text to speech conversion. gTTS is a very easy to use python library which convert text to audio file, which will be transferred to the client as blob data. This API support many languages include English, French, German, Hindi and many more. For how to get started with Google Text-To-Speech using Python see [25].

2.3.2. Speech Synthesis API

Speech synthesis is an interface coming up with the browser Web Speech library API. Speech synthesis is accessed via the Speech Synthesis interface, a text-to-speech component that allows programs to read out their text content. It comes up various voice types, rate and pitch that we can configure in the synthesis voice [11].

2.4. Closed Domain Question Answering (cdQA)

Closed Domain Question Answering (cdQA) is an NLP based Closed Domain Question Answering System. The mission of cdQA is to allow anyone to ask a question in natural language and get an answer without having to read the internal documents relevant to the question.

When we think about question answering systems, it appears as two different kinds of systems: open-domain QA (ODQA) systems and closed-domain QA (cdQA) systems. Open-domain systems deal with questions about nearly anything and can only rely on general ontologies and world knowledge. One example of such a system is DrQA, an ODQA developed by Facebook Research that uses a large base of articles from Wikipedia as its source of knowledge. As these documents are related to several different topics and subjects, we can understand why this system is considered an ODQA. Closed-domain systems deal with questions under a specific domain (for example, medicine or hospitality), and can exploit domain-specific knowledge by using a model that is fitted to a unique-domain database. The cdQA-suite was built to enable anyone who wantsto build a closed-domain QA system easily.

The cdQA architecture is based on two main components: the Retriever and the Reader. When a question is sent to the system, the Retriever selects a list of documents in the database that are most likely to contain the answer. It is based on the same retriever of DrQA, which creates TF-IDF (term frequency-inverse document frequency) features based on uni-grams and bi-grams and compute the cosine similarity between the question sentence and each document of the database [11].

After selecting the most probable documents, the system divides each document into paragraphs and send them with the question to the Reader, which is basically a pre-trained Deep Learning model. The model used was the Pytorch version of the well-known NLP model BERT which was made available by HuggingFace. Then, the Reader outputs the most probable answer it can find in each paragraph. After the Reader, there is a final layer in the system that compares the answers by using an internal score function and outputs the most likely one according to the scores.

Here the pretrained model contains Bert Stanford Question Answering Dataset (SQuAD) which have 100,000+ question-answer pairs on 500+ articles. We can further train this model based on our domain and make it a closed cdQA system.

Among the above-mentioned different solutions, by considering various aspects such as accuracy, ease of use, maintainability and cost, the proposed system will be implemented with solution below. We use apache server with PHP as server side scripting language. Voice to Text using Speech recognition API. For text to voice conversion we use Speech Synthesis API. The question answering model use cdQA. For dome discussions of cdQA see [22, 23] among many others. The cdQA in this report was hosted on Python Flask web server. The proposed system Architecture diagram is shown in Figure 1.

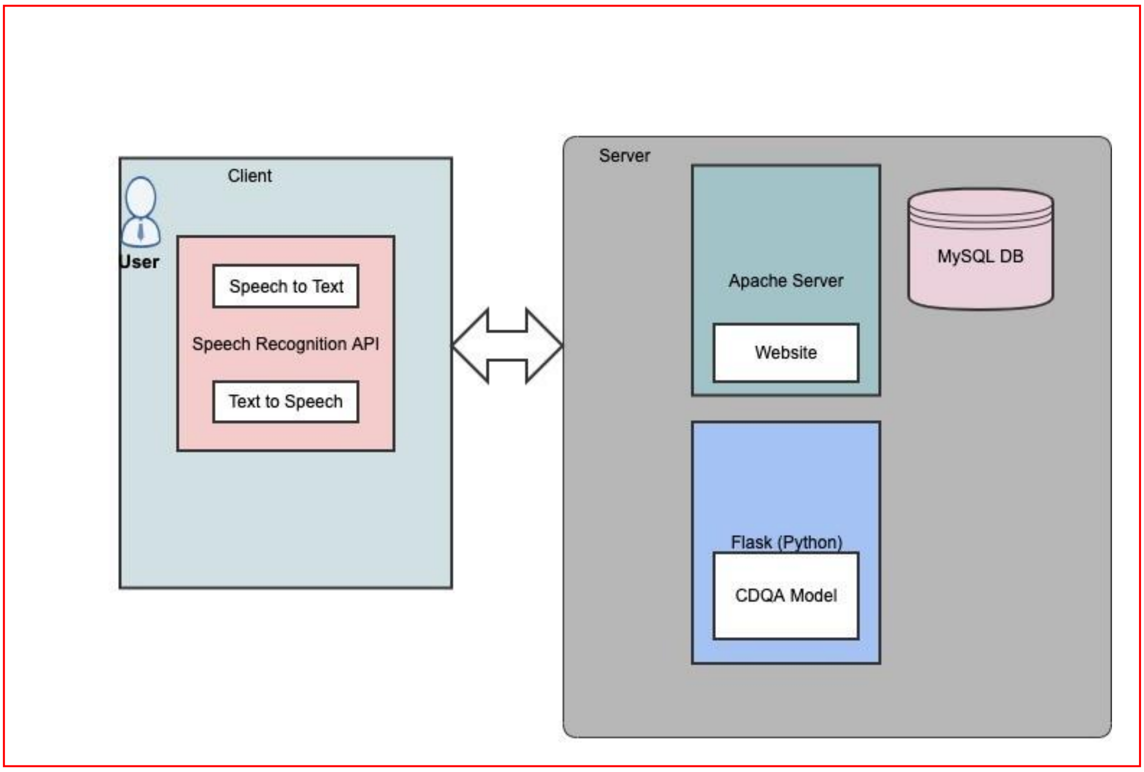


Figure 1. Proposed System Architecture

3. Implementation and Experimental Results

Based on the proposed system architecture the system has been implemented with the following. Web server was implemented using client–server architecture [28]. It divides an application into two parts, ‘client’ and ‘server’. Such an application is implemented on a computer network, which connects the client to the server. The server part of that architecture provides the central functionality: i.e., any number of clients can connect to the server and request that it performs a task. The server accepts these requests, performs the required task and returns any results to the client, as appropriate.

There are five views in this application. The UI shows main attractions nearby, major shopping areas, activities and details about the various kinds of rooms available. The website has been implemented using html, CSS, JavaScript, Bootstrap, FontAwesome for the client side, PHP and apache server for the server side and MySQL database as backend. Room availability is configured in database, it renders in UI when a user search for the rooms. At the top of the website, provided a button to initiate the voice chat. Home screen has been implemented as below, provided search feature at the left, where user can enter the date and guest number to search for the rooms. Click on search button will make rest API call to the server and load the available rooms from the database.

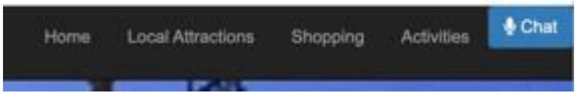
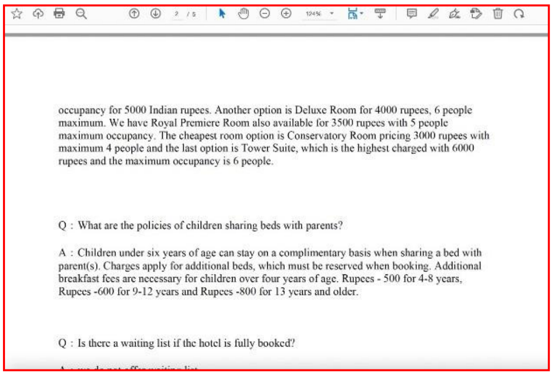


Figure 2. Home Page Menu

3.1. Closed Domain Question Answering (cdQA)

The cdQA subsystem is hosted by a Flask server [29]. Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries [29, 30]. To access the cdQA server from browser, it requires Cross-Origin Resource Sharing (CORS) [31] mechanism. We need to do *pip install flask-cors* module before using flask_cors module in our Python code of Flask server.

We retrained the model with hospitality pdf documents. Sample response for the question will show as below. The system not only shows outputs an answer, but also shows the paragraph where the answer was found and the title of the document / article.



occupancy for 5000 Indian rupees. Another option is Deluxe Room for 4000 rupees, 6 people maximum. We have Royal Premiere Room also available for 3500 rupees with 5 people maximum occupancy. The cheapest room option is Conservatory Room pricing 3000 rupees with maximum 4 people and the last option is Tower Suite, which is the highest charged with 6000 rupees and the maximum occupancy is 6 people.

Q : What are the policies of children sharing beds with parents?

A : Children under six years of age can stay on a complimentary basis when sharing a bed with parent(s). Charges apply for additional beds, which must be reserved when booking. Additional breakfast fees are necessary for children over four years of age. Rupees - 500 for 4-8 years, Rupees -600 for 9-12 years and Rupees -800 for 13 years and older.

Q : Is there a waiting list if the hotel is fully booked?

Figure 3. Sample cdQA model pdf

The model has been trained with 50-100 questions related to the hospitality. Each question-answer will be a paragraph. Once the model is trained, deployed, and exposed using the Flask server. It exposes an endpoint which accept a string input and query the model and return JSON data which include answer to the query, paragraph where it found the answer and document name. Figure 3 shows a sample cdQA model pdf document.

3.2. Voice Chat Bot (Emma)

The designed voice chat bot is named as “Emma”. Click on the “Chat (voice)” button at the top the navigation panel will show up a chat box as below with a welcome voice message as “My name is Emma, your voice assistance, how can I help you today”. This message is a configurable message.

When user clicks on “Talk” button, it will start listening of voice input till the user stop talking. Using the Web Recognition API, it will convert the input voice into text and will show up in the text area which appears at the top of the voice chat popup as in Figure 5. Click on “send” will initiate a cdQA model hosted server call. The processing status will show up as the loading icon on top left of the Talk button in Figure 5. Note the user’s voice was transcribed into text and showed on the screen as well.

Once the result is received, the audio response will render below the text area as shown in Figure 6. It also provides a transcript button, click on that will expand and show the text content.

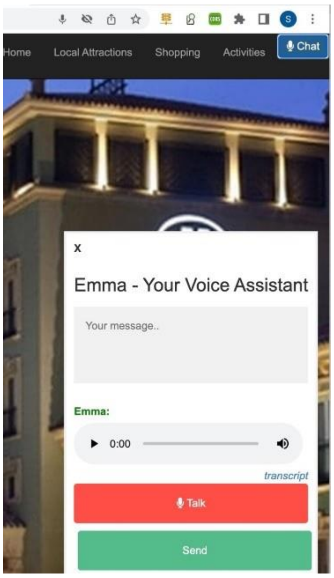
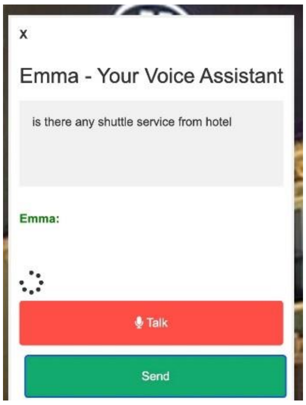
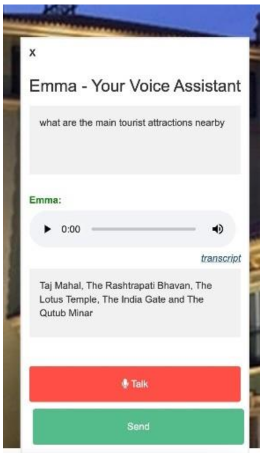


Figure 4. Voice chat bot screen



is there any shuttle service from hotel

Figure 5. After user’s voice input and the Send button clicked



what are the main tourist attractions nearby

Taj Mahal, The Rashtrapati Bhavan, The Lotus Temple, The India Gate and The Qutub Minar

Figure 6. The voice and text response of the Chat bot

4. Conclusion Remarks and Future Improvements

In this project focused on implementing a voice based chatbot as a feature of a hotel web application on an Apache server. Chatbots are software applications that use artificial intelligence and natural language processing to understand humans' need and guides them to their desired outcome. The voice questions are sent by the user, as a query string, to a separate Flask server. It will be parsed through an algorithm on a remote server that analyses the document for all possible relevant answers. The most relevant answer is sent back to the user by using JSON format so thatit can be easily processed by JavaScript. In addition to the best answer of the query, it also return with an approximate confidence from the model. Reusable generic module which captures voice input and convert it into text and convert the text back to voice once the result received from the server. It is implemented using html and JavaScript library.

Based on the research on various solutions to implement voice to text and text conversion like Google Text-to-Speech (gTTS) engine and speech to text engine, the downside is both require separate infrastructure to maintain it. So we decided to proceed with JavaScript utility wrap the implementation for speech recognition and speech synthesis API with event handling.

The question answering mechanism performed very well (based on the evaluation) and predictable for texts shorter than 3000 words. However, with longer texts it started to lose accuracy, losing track of details, and making significant mistakes. The summaries that were meant to help the user formulate questions also worked as intended, however, with the caveat that the summarization model exhibited unaccountable behaviour when supplied with longer texts. The actual usefulness provided are vary between documents. How to conquer this problem may require further research.

An incremental improvement on the system performance and accuracy can be achieved by retraining the model with more questionnaire and through a feedback loop. Training with the bigger dataset requires a higher end system with more GPUs. The improvement of the performance and accuracy will be further studied in the future.

References

[1] "What Is A Voicebot?" Genesys. https://www.genesys.com/en-sg/definitions/what-isa-voicebot

[2] Elupula, Vishnu. "How Do Chatbots Work? An Overview Of The Architecture Of Chatbots", 15May, 2019. https://bigdata-madesimple.com/how-dochatbots-%20work-an-overview-of-the- architectureof-a-chatbot/

[3] The Ultimate guide to chatbot. https://www.drift.com/learn/chatbot/

[4] Mazoor, Kashif. "What Are Chatbots? Beginners Guide To Chatbots", July, 2017, https://www.otechtalks.tv/wpcontent/uploads/2017/08/what-are-ChatBots.pdf

[5] Expert.ai.Team. "What Is A Chatbot? Why Are Chatbots Important?", 17 March, 2020.https://www.expert.ai/blog/chatbot/

[6] Brsuh, Kate and Scardina, Jesse. "Chatbot", Techtarget. https://searchcustomerexperience.techtarget.com/defi nition/chatbot

[7] Lishchynska, Daryna. "What Are Bots? How Do Chatbots Work?", https://botscrew.com/blog/what-are-bots/#:~:text=Chatbot%20or%20bot%20%E2%80%93%20is%20a,an%20instant%20pre%2Dset%20answer

[8] Adamopoulou, Eleni, and Lefteris Moussiades. "An overview of chatbot technology." In IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 373-383. Springer, Cham, 2020.

[9] ZEMČÍK, Mgr Tomáš. "A brief history of chatbots." DEStech Transactions on Computer Science and Engineering aicae (2019).

[10] Slesar, Mila. “Types Of Chatbots, An Overview For Business People”. https://onixsystems.com/blog/types-of-chatbots-overview-forbusiness-people

[11] Wouters, Joren. “3 Chatbot Types, What Is Best For Your Business?” 20 December, 2020. https://chatimize.com/chatbot-types/

[12] Lemaire, Adrein. “Hybrid Chatbot: How To Make Humans And Robots Work Together”. https://www.dimelo.com/en/blog/integration-agentschatbots-botmind/.

[13] Gouba, Rubinder. “What Are Contextual Chatbots? How They Can Make A World Of Difference In User Experience?”, 6 October, 2018. https://medium.com/makerobos/what-are- contextualchatbots-how-they-can-make-a-world-of-differencein-user-experience-e7446c96664e

[14] Engati, Team. Mapping The Growth Of Voice Enabled Chatbot!” https://www.engati.com/blog/growth-of-voiceenabled-chatbots

[15] Soeyuenmez, Anil. “Customer Service Chatbots”, https://www.messengerpeople.com/customer-service-chatbots-rule-based-vs-ai-what-you-need-to-know/

[16] Selig, Jay. “The Power Of Chatbots Explained”. https://www.messengerpeople.com/customer-service-chatbots-rule-based-vs-ai-what-you-need-to-know/

[17] Dimitrios Buhalis and Iuliia Moldavska (2021). In-room Voice-Based AI Digital Assistants Transforming On-Site Hotel Services and Guests’ Experiences. In: Wörndl, W., Koo, C., Stienmetz, J.L. (eds) Information and Communication Technologies in Tourism 2021. Springer, Cham. https://doi.org/10.1007/978-3-030-65785-7_3

[18] Bai Li, Nanyi Jiang, Joey Sham, Henry Shi, Hussein Fazal, “Real-World Conversational AI for the Hotel Bookings”. https://arxiv.org/pdf/1908.10001.pdf

[19] Adam, M., Wessel, M. & Benlian, A. AI-based chatbots in customer service and their effects on user compliance. Electron Markets 31, 427–445 (2021). https://doi.org/10.1007/s12525-020- 00414-7

[20] Rajibul Hasan, Bernadett Koles, Mustafeed Zaman, Justin Paul, "The potential of chatbots in travel & tourism services in the context of social distancing", April 2021 International Journal of Technology Intelligence and Planning 13(1) DOI:10.1504/IJTIP.2021.10041470

[21] “DeepSpeech Model”. https://deepspeech.readthedocs.io/en/v0.9.3/DeepSpeech.html

[22] "DeepSpeech Parallel Optimization", https://deepspeech.readthedocs.io/en/v0.9.3/ParallelOptimization.html

[23] Raun, Benson. “Voice to Text With Chrome Web Speech API”. https://towardsdatascience.com/voice-to-text-with-chrome-web-speech-api-d98462cb0849

[24] Web Speech API https://towardsdatascience.com/voice-to-text-with-chrome-web-speech-api-d98462cb0849

[25] Bharath, K. “How To Get Started With Google Text-To-Speech Using Python”, 26 August, 2020. https://towardsdatascience.com/how-to-get-startedwith-google-text-to-speech-using-python-485e43d1d544

[26] Farias, Andre. “How To Create Your Own Question Answering System Easily With Python”, 7 July, 2019. https://towardsdatascience.com/how-tocreate-your-own-question-answering-system-easilywith-python-2ef8abc8eb5

[27] “Question Answering Using CDQA(BERT)”. https://www.kaggle.com/guizmo2000/questionanswering-using-cdqa-bert-atos-big-data

[28] “Client-Server Architecture” https://www.open.edu/openlearn/sciencemaths-technology/introduction-webapplications-architecture/content-section1.1

[29] “Flask”, https://flask.palletsprojects.com/en/2.2.x/

[30] “Flask framework”, https://en.wikipedia.org/wiki/Flask_(web_framework)

[31] “CORS”, https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS