

AI and Innovation workshop

Virtual Assistant Chatbot

Sagaf Youssouf, Assia Chahidi, Lounaouci Mohamed Lamine,
Oukil Amira, Mahmoudi Faiza, and Adeyemi Abdulnafiu Olalekan

Université Paris-Est Créteil (UPEC)
International Master of Biometrics and Intelligent Vision
61 Avenue du Général de Gaulle, 94000 Créteil
{youssouf.sagaf, assia.chahidi, mohamed-lamine.lounaouci,
abdulnafiu.adeyemi, amira.oukil,
faiza.mahmoudi}@etu.u-pec.fr
<https://www.international-master-biometrics-intelligent-vision.org/>

Abstract. Any company or administration requires a virtual assistant to provide services, answers and online support remotely at any time. For this purpose, the main objective of our project is to work in a team to create a virtual assistant chatbot that will answer the most frequently asked questions related to Biometrics and intelligent vision master degree and help our real assistant: Xavier in dealing with bigger issues and paperwork. Working in group made us manage to create a 3D avatar using blender and a chatbot using Dialog-flow with different functionalities, all integrated in a graphical interface.

1 Introduction

Today, almost all companies have chatbot to engage their users and customer serve, by catering to their queries. A chatbot is an AI application software used to conduct an on-line chat conversation via text or text-to-speech and can simulate a chat with a user in natural language through messaging applications, websites, mobile apps or through the telephone. There exists many chatbot virtual assistant with different types and classifications but the challenge here is to work in a team to build and develop a well-functioning one with the addition of a 3D avatar to the AI application. The work collaboration was managed by a leader and 5 other members, where everyone had a specific task that must end after a certain period of time to check the results and help others if needed. In order to create the virtual assistant chatbot, Google Dialog-flow is used with an integration of different functionalities as speech recognition and emotion classification, all programmed using different python libraries and visualize it in a graphical interface previously programmed using PyQt5. In the following sections we will see the steps we talked about in more details and the flow of our work collaboration to end up with a functioning chatbot and a suitable 3D avatar.

2 Chatbot virtual assistant

Chatbot allows humans to communicate with digital devices just like they are communicating with a real person using Natural Language Processing (NLP). This technology focuses on understanding how humans communicate with each other and how we can get computers to understand and replicate that behaviour. The beauty of chatbot is its ability to shift through large amounts of information and provide relevant results. Figure 1 illustrate its flow diagram. It is one of the most advanced and promising expressions of interaction between humans and machines. They also enhance customer experience and offer companies new opportunities to improve the customers engagement process and operational efficiency by reducing the typical cost of customer service. The chat-bot actually analyses the request of human and identifies the relevant intents then returns a relevant response back to the user.



Fig. 1: Chatbot operation diagram

2.1 Dialogflow conversational bot-building platform

There are different chatbot platform tools available but in our project, we decided to use the Google Dialog-flow which allows a user to build conversational interfaces whilst providing a powerful natural language understanding engine to process and understand what users are looking for. It also handles the job of handling Natural Language into a machine readable data using the machine learning models trained by the user examples.

On the Dialog-flow platform, our first task was to create our agent which serves as our entire chatbot application. The experience of collecting what the user is saying or writing, and mapping it to an intent and then taking an action on it and providing the user with the response starts from the user input or utterance. After the user input has being received and the intent matching phase is completed then the Dialog-flow automatically train a machine learning model with a lot more similar phrases and then maps our users input phrase to the right intent we created. We created several intents with different questions and answers and some entities to serve as action parameters. This was exported as a JSON file and then integrated with our GUI.

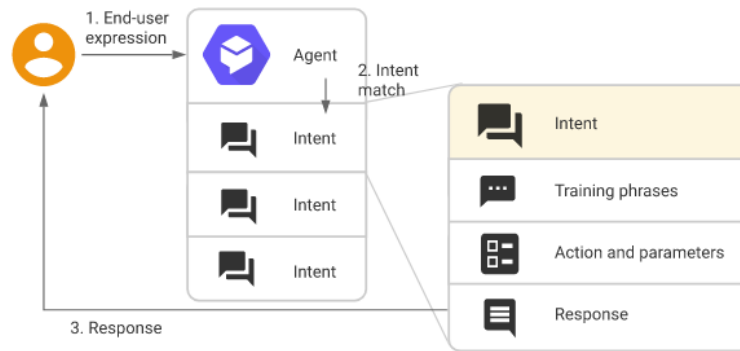


Fig. 2: Flow chart of the Dialogflow

2.2 Chatbot graphical interface

Our desktop application of the virtual assistant chatbot using GUI follow the flowchart in figure 3. The user will send a message in the chat bot as text or voice recording in the query, then we check the message with our intents in our agent. When intent with fulfilment enabled is matched, dialogflow sends a request to our webhook service with information about matched intent and since everything matched Dialogflow send response and display the message and the answer in the text room with speaking of the response message.

The integration of the Dialogflow with our Desktop application using our GUI is done by setting the extracted JSON file from the Google API Service in our program. This file contain all the training data from intent, entities and fulfilment to get our answer when we send question.

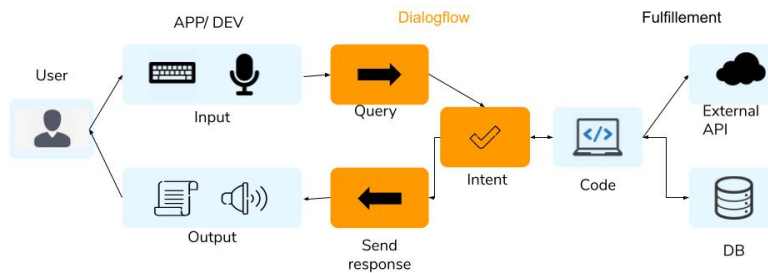


Fig. 3: Flowchart of chatbot

We used PyQt5 to implement the GUI where we create two sides in the application as shown in figure 4. The right side will be used for chat and contain two push buttons; one to send the text from the line text and the second one to record the voice of the question. We display the question and their correspondence answer in the text room while speaking the answer. The left side will be used to display the 3D avatar with different emotion.

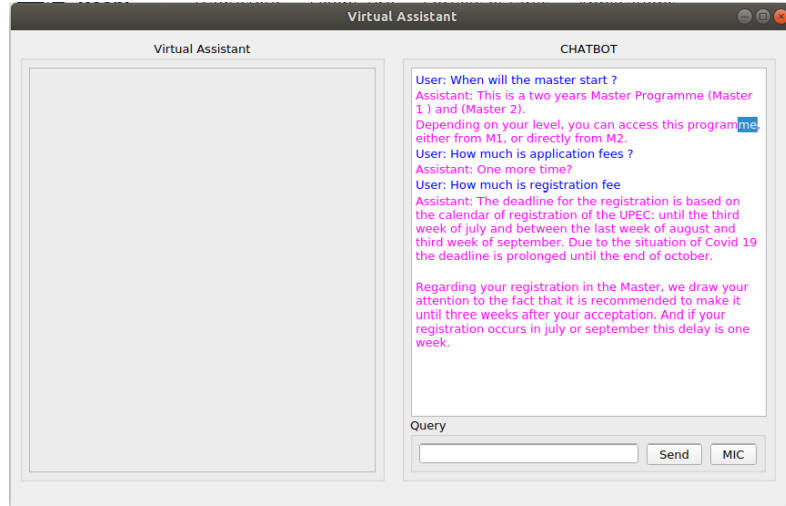


Fig. 4: GUI of Virtual assistant chatbot

2.3 Speech recognition integration

In our Application, we have integrated the Text To Speech(TTS) and Speech To Text(STT) and that's for a simpler communication. Speech recognition is simply the process of using a technology to translate text to a specific language using a specific API(TTS) and to translate a specific language(speech) to a text using a suitable library (STT).

For our case, we have used the speech recognition library in python to recognize the voice from the mic push button. The user will speak in English, as set in our code, the speech will be recognize, translated to text, then Analyzed it.

Finally and using the Google API, that helps generate human-like speech and converts text into audio formats, we translate the resulted text from the previous STT to display it with the correct answer from our Dialog-flow.

2.4 Emotion classification in short messages

To display the correct 3D Animation, emotion of the text given from the answer of the assistant should be extracted using a machine learning algorithm. Here we

are facing a multi-class sentiment analysis problem. We need to classify emotion in the following categories : Neutral, Confused, Smile, Angry, Sad, Surprised. The figure 5 illustrate the flow chart of this part of the system. A pre-trained model is used as a feature extractor. Given text as input with a max length of 500 words, the model is able to predict correct the emotion.

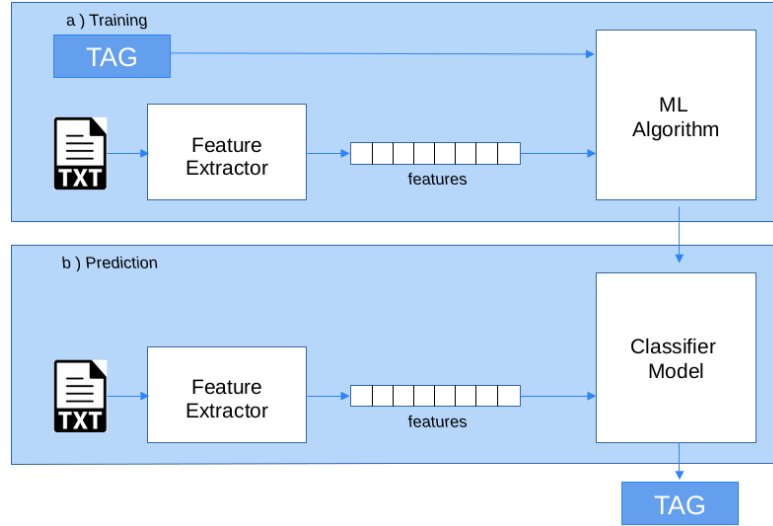


Fig. 5: Flow chart of sentiment analysis module

3 3D Animation

3D avatar is an important part in our program. Which add a life to our application. The user will have a 3D avatar, that can interact with. To attend that goal we used Blender software. First we create a 3D mesh of our real assistant: Xavier, as showing in figure 6a. We used face builder functionality present in Blender. But since we didn't have a satisfying results with 3D Xavier mesh. We moved to 3D cartoon avatar with a rigs all over his body, and since we are concerned just with the face. We manipulated the different rigs and we succeeded to create a different face expressions: smiley, sad, angry, chock, surprise, neutral, confuse as we can visualize in figure 6b. After having our avatar. We are ready now to use it in our application. From this point, we exported it as *fbx* and also as *obj* formats. But we didn't succeed to implement it in our GUI due to some libraries dependencies.

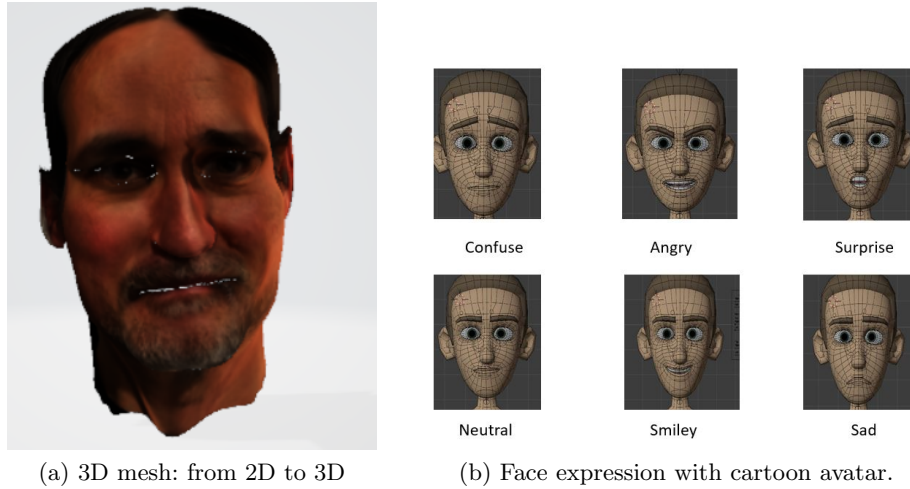


Fig. 6: Generated images by Blender

4 Team collaboration

We start working on this project on 22/11/20. We consecrate the first week for research and explore more the subject. Then, specific tasks were assigned to each member with specific deadlines as shown in the project timeline on the last page. Each week a meeting was held to review the progress of the project for each member of the group. The collaboration between the group members went very well and we learned a lot and helped each other.

5 Conclusion

Through this project, the objective was to work as team and to create a virtual assistant chatbot. We were able to integrate all the sub-modules: speech recognition and emotion analysis modules with the chatbot graphical interface. However, we didn't manage to integrate the 3D avatar. We tried different ways but unfortunately we couldn't integrate 3D avatar with the chatbot properly.

References

1. How to make 3D animated movies in blender, <https://inspirationtuts.com/how-to-make-3d-animated-movies-in-blender>
2. <https://github.com/lukasgarbas/nlp-text-emotion>
3. Emotion Classification in Short Messages, <https://github.com/tlkh/text-emotion-classification>
4. Create Chatbots using Dialogflow: Advanced Tutorial, <https://chatbotsjournal.com/create-chatbots-using-dialogflow-advanced-tutorial-536e6995617f>

Project Timeline

