Guani: Abstract

Kasa Guane is a hostel in Bucaramanga, Colombia, with more than ten years of experience. The hotel receives people worldwide and is famous for its excellent services and products.
Talking with the hostel owner, they receive more than 30 calls per day trying to book a room; some calls asking about the price and what services they have. However, other calls are trying to book a hotel in the prestigious hostel. The goal of Guani Chatbot is to decrease the latency between sending a customer's query and receiving the required response for booking a room in Kasa Guane. When this involves human interaction, the answer may take hours or even days, if required. Automation using Bot can essentially reduce the delay down to where it appears to be instantaneous. The main idea is to create a chatbot that can book hotel rooms for the customers of Kasa Guane. It will be accessible 24/7, help responds immediately to customer queries, and provide all necessary details about the hostel.

The process to work in this product was created first a MVP and defined scope, user persona, and stack technology.
The first scope was an English version that answered just ten questions; we didn't use any database to save the data; it worked locally for the demo. We used a question and answer chatbot (Q&A); the chatbot was focused on answering questions and saving the information in the booking process. Then we define the user persona to know what kind of person will use the Bot. In our case was Ali, Ali is 22 years old, and he graduated from college two years ago. He has no computer in India. The only access to the internet is at Internet Cafe. His main goal is to see the world, and Colombia is a great start. He likes to climb and meet like-minded people.

Then we define our stack technology. The environment that we used was Linux (Ubuntu 20.04), Python 3.8, and RASA. The first step was to create a virtual environment to avoid conflict between libraries and only install the libraries necessary for this project. Next, we install rasa x with the pip command in CMD, and we init a first chatbot template in RASA; when we init the chatbot, the system creates a template of intents, entities, slots and stories. The Intents are the phrases and dialogues that we expect

from the user. And Entities are critical pieces of information that the Bot should collect. It's essential to match entities' names and entities' values to get insight for the customer. The slots are the specific information that we would like to collect. Besides, we created responses: Just as we intend to abstract out what the user is trying to say, we have answers to represent what the Bot would say. Finally, we added stories; stories give our Bot an idea of how the conversation should flow. The main questions were:

1. I would like to make a reserve
2. What is your name?
3. During what period are you going to stay?
4. How many guests?
5. How do you hear about the hostel?

And the question that the chatbot could answer was:

1. Do the hotel has other activities?
2. Do you include the food?
3. *What is the price for the bed?*
4. What type of room do you have?
5. Is there an additional fee?

Then we tested the algorithm with more than 20 people trying simulate hostel booking. With all that information, we trained to improve the model's quality. As a result with getting that 90% of our customers can reserve a hostel without any problem and manifest that this algorithm can be applied to many other environments. Besides that, we get a loss of 14.9 in test a 90 % of F1, and inference accuracy of 97% in training.

The objective was achieved because a chatbot was generated with the ability to take the client's data, make a reservation and, in turn, answer general questions from clients. And in turn, it was tested with real clients, which allowed us to validate its operation in a real environment.

The next step could be to deploy the chatbot on a server and integrate it with the landing page of Kasa Guane and their most important social networks. Furthermore, We can add more language and, at the same time, add functions like canceling any reservations, making payments, and doing other tasks that can work in the future.