CHATBOT DEPLOYEMENT WITH IBM CLOUD WATSON ASSISTANT

Chatbot processes data

It interprets what users are saying at any given time and turns it into organized inputs that the system can process. The NLP engine uses advanced machine learning algorithms to determine the user's intent and then match it to the bot's supported intents list.

At the most basic level, a chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person.

**What are the different types of chatbot:**

* Menu or button-based chatbots
* Rules-based chatbots
* AI-powered chatbots
* Voice chatbots
* Generative AI chatbots

What are the two main chatbots processes?

Artificial intelligence chatbots employ AI and natural language processing (NLP) technology to recognize sentence structure, interpret the knowledge, and improve their ability to answer questions.

Best Practices and Strategies on how to gain a suitable Chatbot Data Collection

Chatbots are now an integral part of companies’ customer support services. They can offer speedy services around the clock without any human dependence. But, many companies still don’t have a proper understanding of what they need to get their chat solution up and running.

In other words, getting your chatbot solution off the ground requires adding data. You need to input data that will allow the chatbot to understand the questions and queries that customers ask properly. And that is a common misunderstanding that you can find among various companies.

Many businesses might have many “data” to include in the chatbots as answers. But, most don’t have the essential data to train their chatbot: examples of how people will express their intentions. Nevertheless, it all comes down to one question: what are the best Chatbot data collection strategies?

This article will give you a comprehensive idea about the data collection strategies you can use for your chatbots. But before that, let’s understand the purpose of chatbots and why you need training data for it.

## ****What Is the Purpose and the benefit of a Chatbot?****

A chatbot is a software or computer program that communicates and interacts with humans using natural language. Companies can use it for various purposes, such as customer support, marketing, etc. The best thing about chatbots is that it streamlines the communication process for companies.

They are exceptional tools for businesses to convert data and customize suggestions into actionable insights for their potential customers. The main reason chatbots are witnessing rapid growth in their popularity today is due to their 24/7 availability.

Customers can get instant services at any time of the day. Companies can now effectively reach their potential audience and streamline their customer support process. Moreover, they can also provide quick responses, reducing the users’ waiting time.

## ****Final Thoughts****

We hope you now have a clear idea of the best data collection strategies and practices. Remember that the chatbot training data plays a critical role in the overall development of this computer program. The correct data will allow the chatbots to understand human language and respond in a way that is helpful to the user.

Also, choosing relevant sources of information is important for training purposes. It would be best to look for client chat logs, email archives, website content, and other relevant data that will enable chatbots to resolve user requests effectively.

If you want to keep the process simple and smooth, then it is best to plan and set reasonable goals. Also, make sure the interface design doesn’t get too complicated. Think about the information you want to collect before designing your bot.

Lastly, organize everything to keep a check on the overall chatbot development process to see how much work is left. It will help you stay organized and ensure you complete all your tasks on time.

## Identifying Suitable Chatbot Workstreams

This is a very important step as it will dictate how and where the chatbot will be deployed. Also, something as simple as interviewing the agents to understand what sort tasks they’re performing most frequently, and also to help determine whether or not a human is needed to perform those A great place to start is to review call transcripts or recordings for calls that have a short tasks.

## Choosing the Correct Chatbot Model

**Autonomous/interviewer**

 Able to handle the entire interaction without agent assistance by not offering to speak to an agent. If customers need additional assistance, they’re directed to call, email or even request an agent to contact them via another method. This type of chatbot interviews the customer by asking some simple questions to guide the customer to make a purchase decision, lookup order status or basic troubleshooting steps.

**Live handoff**

 This mode has all the same features as the autonomous, but offers the ability for the customer to be connected to an agent if the chatbot cannot assist or doesn’t understand the request. Once the chatbot has gotten to the end of its capabilities, it will queue the customer until an agent can take over the conversation.

**Hybrid chat**

 This is the newest method of introducing chatbots, which includes the best of both worlds. With hybrid, a chatbot handles the conversation .

Chatbot coding

# Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer contains number of neurons

# equal to number of intents to predict output intent with softmax

model = Sequential()

model.add(Dense(128, input\_shape=(len(train\_x[0]),), activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(len(train\_y[0]), activation='softmax'))

# Compile model. Stochastic gradient descent with Nesterov accelerated gradient gives good results for this model

sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=**True**)

model.compile(loss='categorical\_crossentropy', optimizer=sgd, metrics=['accuracy'])

#fitting and saving the model

hist = model.fit(np.array(train\_x), np.array(train\_y), epochs=200, batch\_size=5, verbose=1)

model.save('chatbot\_model.h5', hist)

print("model created")

Thank you