

CHATBOT FOR CUSTOMER SERVICE

Talabathula Harsha Venkata Mukesh (Team lead)

Satyam Rai

Ismat Kazi

Sneha Agarwal

GitHub Link: <https://github.com/HarshavenkataMukesh/Rule-based-Chatbot>

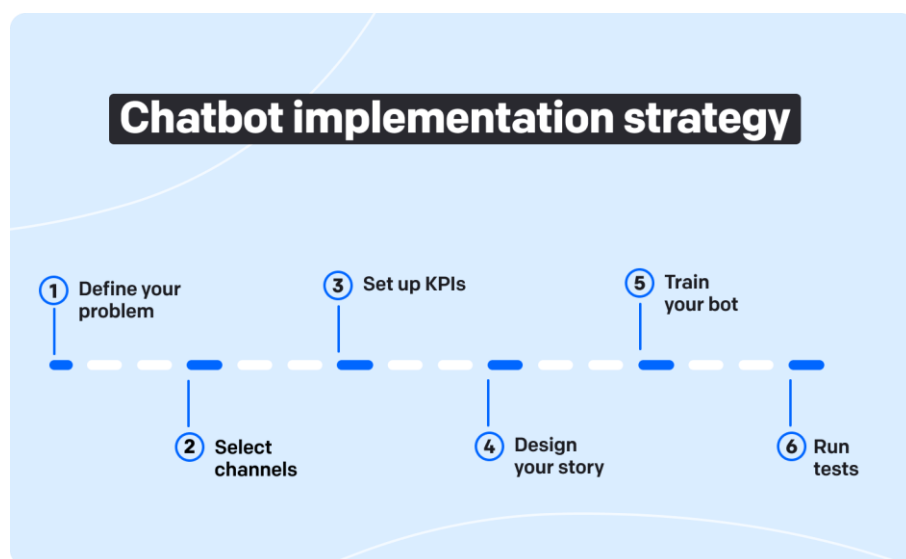
Date: 03-2-2024

Abstract:

In a world where convenience reigns supreme, customer service has significantly transformed. Excellent customer service is the lifeblood of any business, building trust and loyalty that drives repeat interactions and growth. However, managing manpower for this crucial area can be a headache – fluctuating demand, high training costs, and burnout from repetitive tasks are common challenges. Chatbots offer a solution, providing **24/7 support**, automating routine inquiries, and **freeing up human agents** for complex issues, ultimately improving **customer satisfaction** without overstressing resources. It's a win-win for both businesses and their customers.

Introduction:

Chatbots are computer software programs that can interact with humans. With the advancement in machine learning mainly natural language processing, everyone started to create intelligent chatbot systems. You can see different types of chatbots on different websites, chatbots for booking airline tickets on Airline company websites, customer support chatbots in different apps, etc... are such examples.



Problem Statement:

This project aims to develop a rule-based chatbot for small and medium businesses (SMBs) utilizing the Natural Language Toolkit (NLTK) based on **Natural Language Processing(NLP)** which is a machine learning technology that gives computers the ability to interpret, manipulate, and comprehend human language and **TensorFlow**.

The lack of accessible and personalized customer service leads to negative brand perception and lost opportunities. Our rule-based chatbot offers a **cost-effective** way for SMBs to deliver excellent customer service, improving satisfaction, and loyalty, and ultimately driving business success.

The chatbot will focus on core functionalities of intent recognition and response generation, aiming **to improve customer service efficiency** and **address common inquiries**.

Market and Customer need Assessment:

Small and medium enterprises often need good customer service at low cost, which can always be available. Acquiring people to meet the market needs is a very costly thing because there will be fluctuations in need from time to time and increased costs of work.

So, through the chatbot, they can fill these customer service gaps at a fraction of what is needed for people with improved efficiency.

1. **24/7 availability:** Immediate customer support regardless of time or location.
2. **Reduced wait times:** Efficient handling of routine inquiries to free up human agents for complex issues.
3. **Cost-effectiveness:** Affordable solutions that fit within budget constraints.
4. **Easily Scalable:** Adapting to changing customer volumes and business needs without significant additional investment.
5. **Omnichannel support:** Seamless integration with existing communication channels like websites, messaging apps, and social media.

By emphasizing these key points, our customer needs assessment will clearly demonstrate how chatbots are not just technological solutions, but strategic tools for improving a business's customer service. This will resonate with our target audience and strengthen the value proposition of this project.

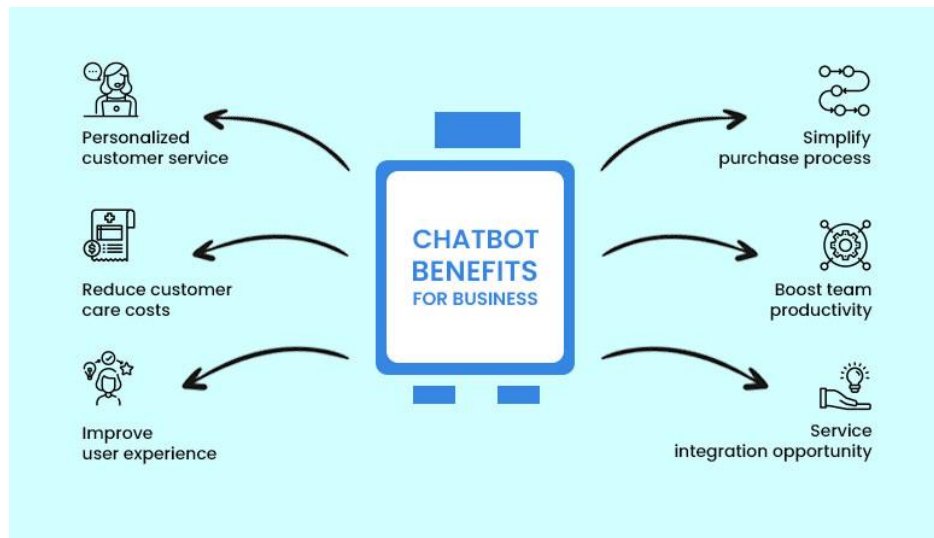


Figure 1: Benefits of using Chatbot for Businesses

Benchmarking Alternatives:

Chatbots are of three types of Rule-based, Hybrid and Machine learning-based.

- **Rule-based Chatbots:** These respond based on pre-defined rules and keywords. Easier to build but less flexible.

This is a good option for simple tasks like answering FAQs, scheduling appointments, or providing basic product information. It's easier to build and maintain, making it ideal for small and medium businesses with limited resources.

- **Hybrid Chatbots:** Combining rule-based with some AI capabilities like intent recognition can handle more complex queries and personalize

responses. This offers a balance between ease of development and advanced functionality.

- **Machine Learning Chatbots:** These learn from data and can adapt to new conversations. More complex but potentially more powerful.

This requires significant investment in data, training, and infrastructure. It's suitable for complex tasks like lead generation, customer service troubleshooting, or providing personalized recommendations. However, it might be too demanding for most small and medium businesses initially.

Applicable Regulations:

- **Data Privacy Regulations:** Depending on our location and target audience, data privacy regulations may apply. Ensure our data collection and usage practices comply with relevant regulations.
- **Accessibility Requirements:** Depending on our target audience, regulations like WCAG (Web Content Accessibility Guidelines) might require your chatbot to be accessible to users with disabilities.
- **Industry-Specific Regulations:** If our chatbot interacts with specific industries (e.g., healthcare, finance), consider industry-specific regulations for data security and compliance.

Business model (Monetisation ideas) for a Chatbot:

BaaS(Bots as a Service) – The future belongs to B2B bots that will empower entrepreneurs and companies to increase their productivity and handle difficult tasks, thereby replicating the business models of most B2B software.

Below mentioned ways are some ideas on how we can monetize a chatbot.

1. **Customization and Development Services:** We can offer personalized services, integration with specific platforms, and advanced feature development. This provides high-value opportunities for dedicated customers and expands your revenue potential along with customer satisfaction and retention.

2. **Pay-per-use model:** We can charge based on the number of interactions or conversations the chatbot is handling. This is a good revenue model for fluctuating customer interaction volume businesses.
3. **Subscription model:** We can offer tiered subscription plans based on features, number of concurrent users, and data storage needs. This provides predictable revenue and encourages long-term customer commitment.
4. **Data Analytics and Insights:** We can analyze chatbot interaction data and provide insights to businesses about customer preferences, pain points, and buying patterns. It is all about understanding and utilizing user psychology to our advantage.
5. **Native advertisement and Sponsored Content:** We can integrate contextual advertising within the chatbot interface, targeting relevant offers to users based on their interactions. This requires careful implementation to avoid compromising user experience (Native advertising has been performing far better than conventional banner ads because it works well for both parties publisher and the brand).

THE BEST CHATBOT MONETIZATION STRATEGIES

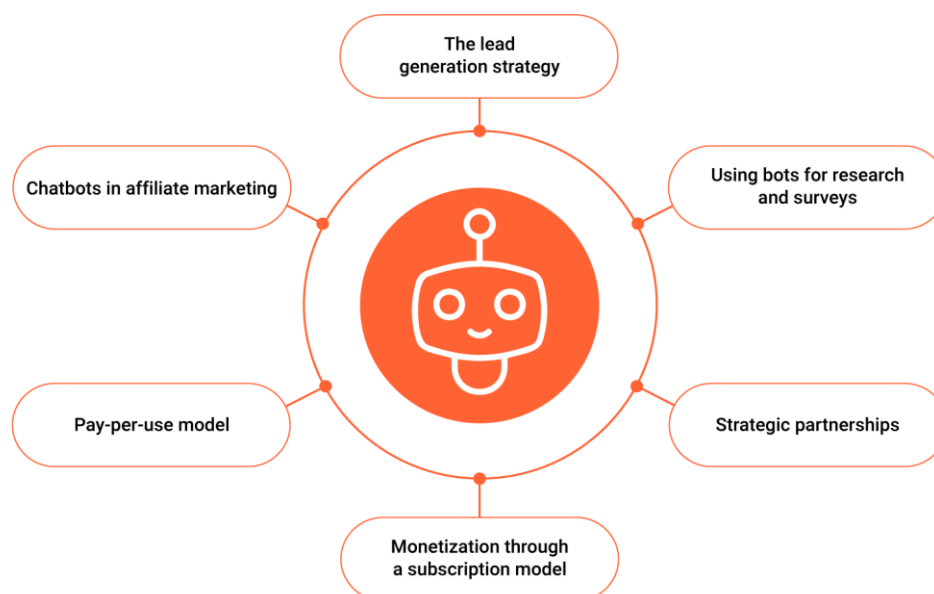


Figure 2: Ways to monetise a Chatbot.

Applicable constraints:

- Building and maintaining a chatbot might require additional space for data storage and processing. We need to train well on enough data to get efficient output.
- Requires expertise in Python, NLTK (Natural Language Processing), TensorFlow, and chatbot development best practices. Consider collaborating with developers or acquiring the necessary skills like deployment.

Concept Generation:

An efficient, 24/7 customer service assistant for your business. Powered by AI but simple and affordable. This rule-based chatbot is made especially for small and medium businesses.

Concept Development:

This chatbot "CHARM," (Customer-centric, Helpful Assistant for Responses & Management) will take care of your routine customer interactions, freeing up your valuable time and resources for what matters most - growing your business.

CHARM is more than just a chatbot; it's your customer's first impression, your 24/7 support lifeline, and your friendly AI companion, all rolled into one.

With CHARM, you can:

- Increase customer satisfaction and loyalty.
- Boost conversions and lead generation.
- Optimize agent productivity and reduce costs.
- Gather valuable customer insights and improve service.

CHARM is the smart, helpful, and always-on answer to your small and medium businesses' customer service needs. Start charming your customers today!

Final product proto type with a schematic diagram:

Schematic Diagram:

1. **User Interface:** Web-based dashboard for configuration, rule management, and performance monitoring. Mobile app for convenient access and interaction with CHARM on the go.
2. **Communication Channels:** Website integration for embedded chat functionality. Messaging platform integrations (e.g., Facebook Messenger, WhatsApp) for multi-channel reach. Social media integration for customer support inquiries and engagement.

Core Modules:

- **Intent Recognition:** Utilizes NLTK's Natural Language Toolkit to analyze user input and identify the underlying intent (e.g., booking appointment, product inquiry, feedback).
- **Rule Engine:** Matches recognized intents with pre-defined rules to trigger appropriate responses. Rules can be based on keywords, context, and user history.
- **Response Generation:** Generates text responses based on the triggered rule, incorporating simple templates and dynamic variables for personalization.
- **Knowledge Base:** Maintains a repository of relevant information (e.g., product details, FAQs, company policies) for accurate and consistent responses.
- **Data Analytics:** Tracks user interactions, measure performance metrics, and provides insights for optimizing rule sets and improving CHARM's effectiveness.

A cost-effective and accessible solution for small and medium businesses to embrace the power of AI and elevate their customer service experience. As a rule-based chatbot, it offers a balance between customization and ease of development, making it ideal for businesses of all sizes.

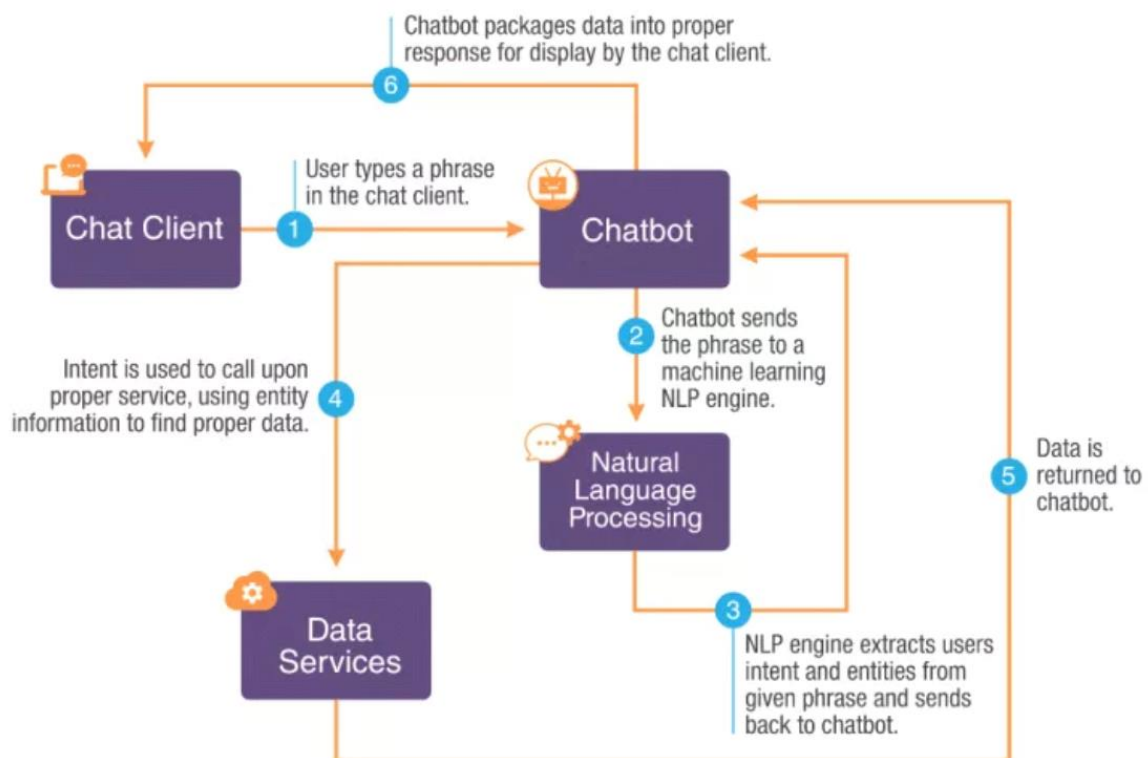


Figure 3: Schematics of Chatbot working.

1. User input.
2. Sending the phrase to the NLP engine we trained.
3. Finding and extracting user intent.
4. Finding proper data for intent recognised.
5. Data returned to chatbot.
6. Packaging the data for proper response and display.

References:

- <https://www.analyticsvidhya.com/blog/2022/05/a-complete-guide-on-chatbot-development-using-python/>
- <https://www.youtube.com/watch?v=t933Gh5fNrc>
- <https://data-flair.training/blogs/python-chatbot-project/>

Feasibility Analysis:

- Chatbots were made to be the alternative for support centers or inquiries-specific jobs.
- In the hope of utilizing the logical and thinking prowess of humans in more regions where necessary. But in reality, it turned out that even after 70 years of development an AI chatbot feels rather robotic. The feasibility of a chatbot is not as expected. Chatbot fails at creating an emotional connection with the user which destroys the conversation. The conversations don't last as long as fifteen minutes.
- Most of the chatbots still do this day are using poor old rule-based techniques. The complexity increases at an exponential level creating it more humanly.
- Artificial intelligence is increasing at the highest rate in the chatbot sector when compared to any other.

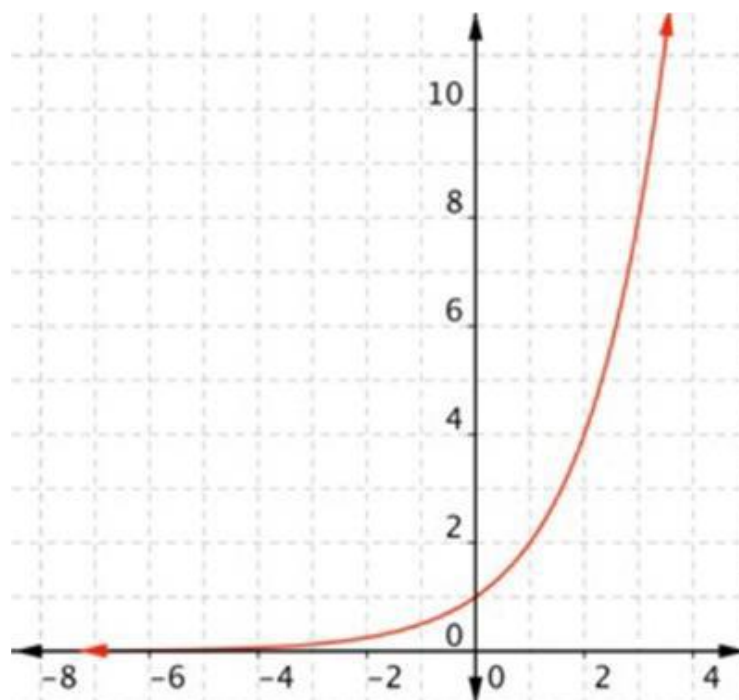


Figure 5: Complexity of neural networks vs technological improvement curve.

The Vertical line here represents the complexity of the Neural Network in comparison to the improvement as a more human-understandable chatbot.

Cyber Security:

Artificial Intelligence is the new electricity of the upcoming era. It would be integrated everywhere just like electricity nowadays. Artificial Intelligence is a big boon in the region of cyber security.

Some of the known problems resolved by Artificial Intelligence in the realm of cyber security are –

- **Increasing the number of cyber threats:** Updating the number of runs of checks by filtering possible places and presenting new and obscure dangers. The best way to defend against an attack is to know when and where it will take place. Updating the public beforehand to be cautious helps majorly.
- **Unsupervised Artificial intelligence** can easily point out some defect or abnormality in the network and warn before any potential damage is done. The chatbots collect data and analyze it in lookup for suspicious patterns or irregularities on the web.

Cost of building and maintaining a Chatbot:

Features/Types of work	Approx time (hours)	Approx cost ²
Conversation	122	6100
Payment System	81	4050
Geolocation	64	3200
Personal Approach	73	3650
Syncing With User Accounts	42	2100
Frameworks & libraries integration	32	1600
UI/UX design	140	7000
QA services	420	21000
DevOps services	110	5500
Project management	200	10000
Total	1284	64200

Initial expense division:

Cost Component	Typical Percentage of Total Cost
Development Cost	45%
Testing Cost	25%
Resources	20%
Maintenance Cost	10%

From here on, we can charge our client companies on a monthly subscription or on a yearly basis, in which we will provide them with maintenance service and regular updates which keeps up with the latest nlp trends.

We can sell the chatbot with different plans like:

Standard subscription plan	Between ₹0 and ₹1000/mo
Pro subscription plan	Between ₹1500 and ₹5000/mo
Enterprise subscription plan (custom made plans for businesses)	Between ₹600 and ₹5,000/mo
In-house chatbot development	About ₹10,000/mo
Hire an agency for chatbot development	Between ₹10,000 and ₹50,000/mo for development and maintenance of the chatbot
Consumption-based fee	Between ₹0.006 and ₹1 per text or audio request

Code Implementation:

Creating Intents File, this intents file contains the different patterns of the question that the user might enquire and the possible output for the specific question, and a tag for that type of question.

```
{
  "intents": [
    {
      "tag": "greetings",
      "patterns": ["Hello there", "Hey, How are you", "Hey", "Hi", "Hello", "Anybody", "Hey there"],
      "responses": ["Hello, I'm your helping bot", "Hey it's good to see you", "Hi there, how can I help you?"],
      "context": [""]
    },
    {
      "tag": "thanks",
      "patterns": ["Thanks for your quick response", "Thank you for providing the valuable information", "Awesome, thanks for helping"],
      "responses": ["Happy to help you", "Thanks for reaching out to me", "It's My pleasure to help you"],
      "context": [""]
    },
    {
      "tag": "no_answer",
      "patterns": [],
      "responses": ["Sorry, Could you repeat again", "provide me more info", "can't understand you"],
      "context": [""]
    },
    {
      "tag": "support",
      "patterns": ["What help you can do?", "What are the helps you provide?", "How you could help me", "What support is offered by you"],
      "responses": [ "ticket booking for airline", "I can help you to book flight tickets easily"],
      "context": [""]
    },
    {
      "tag": "goodbye",
      "patterns": ["bye bye", "Nice to chat with you", "Bye", "See you later buddy", "Goodbye"],
      "responses": [ "bye bye, thanks for reaching", "Have a nice day there", "See you later"],
      "context": [""]
    }
  ]
}
```

Importing some of the required libraries for our project.

```
import numpy as np
import nltk
import json
import pickle
import re
import random
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import SGD
from nltk.stem import WordNetLemmatizer
```

we are initializing some lists for saving the preprocessed and preprocessing data.

```
tokenized_words=[]
classes = []
doc = []
ignoring_words = ['?', '!']
data_file = open('intents_file.json').read()
intents = json.loads(data_file)
```

We loaded the JSON file. Now we have to perform some preprocessing, we are going to iterate through each of the pattern questions in the intents file and tokenize it. This tokenized text along with the tag is stored as a list. tokenized_words contain all the different words in the intents file which is tokenized using nltk.

```
for intent in intents['intents']:
    for pattern in intent['patterns']:
        w = nltk.word_tokenize(pattern) #tokenizing
        tokenized_words.extend(w)
        doc.append((w, intent['tag']))
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```

Now we have to perform lemmatization on the data and need to remove the question tag and other ignoring words

```
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(words.lower()) for words in
tokenized_words if w not in ignoring_words] #lemmatizatio
lemmatized_words = sorted(list(set(lemmatized_words)))
classes = sorted(list(set(classes)))
```

Now saving the lemmatized words and classes into a pickle file

```
pickle.dump(lemmatized_words,open('lem_words.pkl','wb'))
pickle.dump(classes,open('classes.pkl','wb'))
```

As the next step, we need to create our training data.

```
training_data = []
empty_array = [0] * len(classes)

for d in doc:

    bag_of_words = []

    pattern = d[0]

    pattern = [lemmatizer.lemmatize(word.lower()) for word in
pattern]

    for w in lemmatized_words:

        bag_of_words.append(1) if w in pattern else
bag_of_words.append(0)

    output_row = list(empty_array)

    output_row[classes.index(d[1])] = 1

    training_data.append([bag_of_words, output_row])

random.shuffle(training_data)

training = np.array(training_data)

train_x = list(training[:,0])

train_y = list(training[:,1])
```

Model Creation

Now we can create our Neural network model. With the help of Keras and TensorFlow library, we are creating the Model. So let's start the implementation.

```
bot_model = Sequential()
bot_model.add(Dense(128, input_shape=(len(x_train[0])),),
activation='relu'))
bot_model.add(Dropout(0.5))
```

```

bot_model.add(Dense(64, activation='relu'))
bot_model.add(Dropout(0.5))
bot_model.add(Dropout(0.25))
bot_model.add(Dense(len(y_train[0]), activation='softmax'))

```

We've created our model. Next is to compile our model with the stochastic gradient descent feature.

```

sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
bot_model.compile(loss='categorical_crossentropy', optimizer=sgd,
metrics=['accuracy'])
x_train = np.array(x_train)
y_train = np.array(y_train)
hist = bot_model.fit(x_train, y_train, epochs=200, batch_size=5,
verbose=1)
bot_model.save('chatbot_model.h5', hist)

```

Now let's take another python file for testing and creating our actual chatbot

```

intents_file = json.loads(open('intents.json').read())
lem_words = pickle.load(open('lem_words.pkl','rb'))
classes = pickle.load(open('classes.pkl','rb'))
bot_model = load_model('chatbot_model.h5')

def cleaning(text):
    words = nltk.word_tokenize(text)
    words = [lemmatizer.lemmatize(word.lower()) for word in words]
    return words

def bag_ow(text, words, show_details=True):
    sentence_words = cleaning(text)
    bag_of_words = [0]*len(words)
    for s in sentence_words:
        for i,w in enumerate(words):
            if w == s:
                bag_of_words[i] = 1
    return (np.array(bag_of_words))

```

Creating a prediction function for predicting the classes or tags of the question that are asked by the user.

```

def class_prediction(sentence, model):
    p = bag_ow(sentence, lem_words, show_details=False)
    result = bot_model.predict(np.array([p]))[0]

```



```

ER_THRESHOLD = 0.30
f_results = [[i,r] for i,r in enumerate(result) if r >
ER_THRESHOLD]
f_results.sort(key=lambda x: x[1], reverse=True)
intent_prob_list = []
for i in f_results:
    intent_prob_list.append({"intent": pred_class[i[0]],
"probability": str(i[1])})
return intent_prob_list

```

Now we are having the predicted classes or tags based on the inquiry of the user. As you can see in the intents file there are more than one response for each tag, so we are creating a function for selecting a random response from the predicted tag and sending it as a bot response.

```

def getbotResponse(ints, intents):
    tag = ints[0]['intent']
    intents_list = intents['intents']
    for intent in intents_list:
        if(intent['tag']== tag):
            result = random.choice(intent['responses'])
            break
    return result
def bot_response(text):
    ints = class_prediction(text, bot_model)
    response = getbotResponse(ints, intents)
    return response

```

We created several functions for the working of the chatbot. So let's talk to our chatbot.

```

for i in range(3):
    text = input("You : ")
    print("Bot : ",bot_response(text))

```

Output

```

You : hey
Bot :  Hi there, how can I help you?
You : what help can you do
Bot :  ticket booking for airline
You : bye
Bot :  See you later

```

Limitations:

Building a chatbot using NLP has some limitations, such as being complex to build and dependent on high-quality data, lack of understanding of context and standardization, limited ability to handle unstructured data, language barriers, and privacy concerns. It is important to carefully consider these limitations and take steps to mitigate any negative effects when implementing an NLP-based chatbot. They are designed to automate repetitive tasks, provide information, and offer personalized experiences to users. Using NLP in chatbots allows for more human-like interactions and natural communication.

Conclusion:

Through this, we demonstrated how AI can empower small and medium businesses to provide exceptional customer service, even with limited resources. Built on a foundation of powerful NLP tools and efficient rule-based automation, Chatbot tackles the challenges faced by SMBs head-on. It liberates human agents from routine tasks, ensuring 24/7 support and personalized interactions that foster customer satisfaction and loyalty.

Chatbot's flexibility and accessibility make it the ideal solution for businesses of all sizes. Its web-based dashboard and mobile app offer convenient control, while its seamless integration with existing platforms guarantees a smooth user experience. Moreover, Chatbot continuously learns and adapts, providing valuable data insights that enable businesses to further refine their service strategies.

The future of customer service lies in intelligent solutions that personalize and optimize interactions. A chatbot stands at the forefront of this revolution, offering SMBs a cost-effective and powerful tool to compete in today's digital landscape. By embracing chatbots, businesses can build stronger relationships with their customers, enhance brand reputation, and ultimately drive sustainable growth.