NLP Chatbot For Order Assistance
Using Dialogflow

Abstract: This paper explains the development of an order assistant conversational AI chatbot with Dialogflow. The chatbot will be created for an improved experience, more customer interaction, and simplicity in placing orders. The current study looks into how NLP chatbots will likely revolutionize customer service operations. The existing systems designed for order assistance are generally ineffective. This increases customer frustration and reduces loyalty. Manual handling of customer queries increases response rate the operational cost and makes the very slow. An automated system with efficient order processing and the possibility of personalizing support would be great. The approach to design an NLP chatbot using Dialogflow in a manner through which machine learning and natural language understanding could be integrated. This chatbot will target the purpose of customer queries and product information, especially in placing an order or tracking it. A user-centric approach will be used to avoid complexity during chat. From the results, it is evident that a chatbot can effectively handle customer queries with an accuracy rate of 90%. Operational costs are reduced by 30%, while response times are shortened by 50%. The level of chatbot performance that is implemented is very gratifying for users since it increases customer experience and loyalty. This research shows that NLP chatbots can revolutionize support in order assistance and customer service. The delivered chatbot solution will ensure that cost-effective and scalable improvements are brought to customer service.

Keywords: AI chatbot, Natural language processing, Dialog Flow Artificial Intelligence, Natural Language Understanding, Google Dialogflow.

1. INTRODUCTION

NLP constitutes one of the most crucial aspects of artificial intelligence, where intelligent systems interact in a very seamless manner with human beings.[2] The ease of use and efficiency of communication have gained popularity over the years among chatbots backed by NLP. Today, chatbots are omnipresent and quite integral for customer service and operational efficiency in a variety of industries.[1]

Some of the best-known cases are those deep learning models, fueled by modern AI technologies, which popularize chatbot development similar to what we see in today's world with Alexa, Cortana, and others. One could say that chatbots are just one more excellent example of how far things have come in the way organizations and services are better communicating with their customers. Almost all of these automated solutions would catch and resolve the user's problem in no time, deploying AI capability and hence furnishing an all-over better customer experience, not to mention the advancements in business process efficiency [2].

This paper investigates the ability of a chatbot to revolutionize restaurant ordering processes.

1 Dr. Vibha MB1*,
2 Tabrez Ahmed I,
3 Satyanarayana Y. P.,
4 Kiran Kokane
5 Tushar Varale
6 Rohit Kumar
7 Soumya R.
8 Soumya Subray Hegde

*Assistant Professor, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, vibha-mcavtu@dayanandasagar.edu
2 PG Scholar, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, tabreziahmad@gmail.com
3 PG Scholar, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru. satya30yp@gmail.com
4 PG Scholar, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru. kirankokane35@gmail.com
5 PG Scholar, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, tushar.000varale@gmail.com
6 PG Scholar, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, rohitrk9609@gmail.com
7 PG Scholar, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, Soumyaramesh2002@gmail.com
8 PG Scholar, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, Soumyahegde652@gmail.com

Copyright © JES 2024 on-line: journal.esrgroups.org
The proposed chatbot should enhance service business customer experience based on the latest NLP techniques, and in its functioning, make use of databases. This way, it neatly bypasses the inherent limitations of traditional methods of order by giving an intuitive and user-friendly interface that allows customers to smoothly make orders and track them in real time.[3]

The essence of this article lies in the recognition of changed customers and restaurant demands in this digital age. The initiative is to go beyond expectations for customers, increase their engagement, and keep them happy with the application of technological innovations in the restaurant ordering business. This only serves to bring out the point home that it is so paramount to use advanced technology for improved service delivery and meeting the ever-dynamic needs of both contemporary customers and enterprises [3].

The chatbot market is currently witnessing extraordinary growth and success. The global chatbot market is expected to be worth $10.08 billion by 2026, with a compound annual growth rate of 23.5% from 2021 to 2026. The major factor driving this great momentum is the increasing adoption of chatbots across different industries.

The Paper Continues with The Following Sections.

In Sect. [2], We Will Cover Existing Research by Using Dialogflow For NLP-Powered Chatbots. Next,

In Sect. [3], Several Issue That Need Solution for The Integration of Chatbots Are Discussed. While,

In Sect. [4], Some Discussion on Knowledge Abstraction and Response Production Techniques to Enrich Chatbot Performance. Next In,


In Sect. [6][7], Complete Steps for Procedure Execution and Code Implementation. While

In Sect. [8] Information on Global Chatbot Market Trends and Forecasts. And Then

In Sect. [9], Implementation Results

Finally, In Sect. [10] Reports Conclusion and

In Sec [11] Highlights Future Scope

2. LITERATURE REVIEW

This phase of the study requires a look into research that has been performed on previous studies and articles, among other resources. The purpose of this research is to understand what present methodologies or approaches have been followed and what successful strategies of chatbots for ordering system support are while reading academic articles and industry insights.[2] It helps make decisions and encourages creative problem-solving throughout the process. The motivation of this inquiry is to gather important insights into current practices in the design of efficient and user-friendly chatbots targeted to food ordering scenarios.[2] This helps not only in understanding state-of-the-art methods but also gives a very holistic view on the challenges and potential mitigating strategies adopted in similar scenarios.[3] Additionally, this approach identifies where the current state of knowledge is lacking or underexplored. By pinpointing and underlining these shortcomings, the research can suggest novel ways and effectively contribute toward the advancement in that field. The research would have great significance for the industrial booms, whether it improves NLP algorithms so that the user interactions are more precise or if it creates new techniques to manage and retrieve order data from databases.[1] Essentially, this research step is quite important in giving the readers a firm understanding of the environment. This understanding puts them in the position of using the collective experience acquired from previous research, which impacts the development of a highly efficient "Order Assistant Chat Bot." The realization of the findings from this comprehensive study directs the designing and implementation of the research process, ensuring compliance with the current trends in the market, improved user interactions, and effective solutions to practical challenges.[1]
3. PROBLEM STATEMENT

Traditional meal ordering systems are plagued by inefficiencies, leading to poor customer experiences and lost revenue. The key challenges are:

1. Slow response to customer queries, resulting in frustration and dropped orders.
2. Inability to handle simultaneous requests, overwhelming staff and reducing productivity.
3. Weak analytics and insights, making it difficult to optimize the order process and increase customer satisfaction.
4. Inefficient order management, leading to errors and delays due to updates to menus, price changes, and client preferences, resulting in revenue leakage.

These challenges can be overcome by developing a real-time AI chatbot that provides timely and accurate support, handles multiple customer requests simultaneously, offers data-driven insights for business optimization, and enables effective order management [2,3].

4. EVALUATION METHODOLOGY AND METRICS

The methodology encompasses two main components: Knowledge Abstraction and Response Production. Knowledge abstraction involves gathering and organizing information about the restaurant's menu and services, whereas response generation is influenced by the properties of available Dialogflow tools and the quality of data derived from knowledge abstraction.

Knowledge Abstraction [1]

Knowledge abstraction proceeds through the following three important steps: Data Gathering, Manipulation, and Augmentation.

Data Gathering [1]: Data gathering is a systemic process of collecting and measuring information on specific system variables to answer relevant questions and evaluate outcomes. It is an important step in almost all disciplines; it ensures the reliability and precision of the data that is collected. High quality in data collection will lead to credible and dependable solutions to research questions [1,3].

Data Manipulation [1]: The data, which is collected in any manner, has to be entered into the database. It thereby allows the developers to manipulate the data in any manner at all. For example, the developers are frequently required to prepare spreadsheets, which sort out the available data according to topics, and give a brief at the end, or they prepare forums where one section is incorporated with questions whereas the second section is included with answers – each answer or question is classified depending on the kind of subject that it concerns. First, the labels must be validated. Validated, these labelled questions are converted directly into Dialogflow intents with little effort. This is where the human oversight comes in when you are labelling the questions and categorizing them [1,3].

Data Augmentation [1]: Boost training of the NLP model by increasing number of examples used. One can devise new examples if the relationship among similar questions and answers is found by using general terms or entities. The response could be used to build a new question and have it kept consistent. Developers can create a new intent spreadsheet of intents and their connected entities for further improvement in the ability of Dialogflow to recognize multiple entities at a time from a single request [1,3].

Response Generation [1]

The architecture of Dialogflow makes this model easy with the help of intents, contexts, training phrases, and responses.

Intents: An intent consists of a context, training phrases, and responses. Contexts also assist to lay down a trail of conversation so that only the intended intents get executed under the expected conditions. The phrases in training are both template and example sentence and tagged based on either entities or intents. When a user input is matched with an intent, the responses are triggered. The form of the response can vary and is often just plain text but could also include content structures such as cards or tables that can be passed from Dialogflow.
Entities: Terms suitable for multiclassification. They classify user interactions not only at the level of individual questions but also at the level of broader structures, with benefits for instructional applications. Each definition can be associated with a separate intention, and having entities means that many fewer of the possible completions of a query need to be considered in a given context, improving the effectiveness of intent detection. An important tenet of our structured approach is to ensure that the chatbot can handle a wide range of user queries, thus delivering correct and relevant answers with the help of strong functionalities in Dialogflow and supporting technologies.

5. DIALOGFLOW & ITS COMPONENTS

Dialogflow is a user-friendly app that applies an easy approach to crafting and styling conversational chat assistants, which detect and react to inputs from voice and text. It can be driven by either the Dialogflow console or embedded within a web application. In this article, we are going to see how Dialogflow reduces the friction in building a conversational chat assistant something that a data scientist might do. by setting up a customer support agent for a food delivery service, it shows how the chatbot could process food orders and provide answers to all other questions of the service customers.

Before you start the development process, it's important to understand some fundamental concepts used in Dialogflow. Dialogflow eases the development of NLP applications by providing an easy-to-use console to visualize, plan, and train AI-powered chatbots. With the simplified console design, the process is much less complex and allows participants with a non-expert background in the domain.

a) Agent

Dialog flow’s agent is a chatbot created by the user for communication with other end-users and analyzing data following input. An agent in Dialogflow contains many elements. The agent is rapidly retrained, fitting in any changes that will be affected in any of the elements. Users can easily select a pre-built agent from the list to quickly build conversational chatbots.[5] These prebuilt agents provide just these basic intents and replies that a conversational assistant needs. A conversational assistant in Dialogflow is an agent, and everyone interacting with the agent other than the creator is an end-user. [4]

Fig 1: Agent Panel Screenshot

b) Intent

In a manner that is analogous to its literal sense, the intent forms the end objective or goal a user would like to achieve by the time they reach an agent. In a conversation, one agent can define different intents to process different statements and group these intents using contexts. This will allow an agent to decide on the intent of a sentence by using the sentence's meaning [5]. As an example, an agent built to process food orders would understand the purpose and at the end-use predefined intents to ease processing of orders or send recommendations for menus.[8]
c) Entities.

The entity in Dialogflow allows the processing and extraction of some piece of information from the user input.[8] For example, if you add a Food Dish entity to an intent, that will allow Dialogflow to capture the Food Dish names if a user asks for them in context at a restaurant. Predefined System entities are automatically added when the agent is created. More so, Dialogflow supports defining custom entities, which therefore allows developers to assign some unique values for their application. This makes the ability of doing inferences with Dialogflow more sharp and better responsive to all kinds of input, giving higher interaction relevance and accuracy.

d) Training Phrases

A training phrase is an example of what a user might say to express a specific intent. These are important during the training of natural language understanding to understand the varying ways that users express the same intent. With a diversity of training phrases, developers help Dialogflow understand these subtleties and variations in human language, thereby enhancing the agent's matching precision in regard to intent.[4] Dialogflow recommends "at least 10-20" training phrases per intent to be added in order to maximize robustness. Moreover, the training phrases can be annotated with entities for extracting particular information, which makes the conversation more dynamic and aware of the context. This annotation will help Dialogflow effectively identify and manage the many user inputs, leading to an enhanced user experience.[5][4]
e) Response

After Dialogflow has finished fulfilling Within the response section, you put the material that will be served to the user after Dialogflow has satisfied their intent or request. The response can come in different formats, whether text and/or rich material, or it can even come as an interactive audio response, depending on the device that hosts your bot. [5] You need to design responses for every objective that you have. The data that you have collected or acquired can be brought into the answers in the form of either variables or static text. [5]

f) Context

In Dialogflow, contexts help in managing a conversation with an agent. Contexts are string names used to keep some record of a kind being in the ongoing conversation. For some particular intent, more than one input and output context can be defined. When any user's query matches an intent, it makes the associated contexts active at that time.[4] Then, using those active contexts, the next intent is decided to have a logical flow of the ongoing conversation. To better understand contexts, let's think about a restaurant ordering scenario. Let's imagine contexts being different phases of the ordering process. So, when the user expresses a wish to see the menu, they're in the "menu" context. After they've decided on the dish and have expressed an order, they enter into the "ordering" context. And as they finalize their choice, they go into the "confirm order" context. Finally, when they place the order, they fall back to the "order placed" context. This analogy explains how contexts are like reminders for the customer so that the dialogue can move naturally and coherently from one point to another on which it is built.[8][5]

g) Fulfilment

Dialogflow Fulfilment makes it possible for an agent to provide a response that is more dynamic in response to an intent that has been recognized, as opposed to a response that has been produced in a static manner. This could
be accomplished by making a call to a defined service in order to carry out a step, such as generating new data or obtaining information from a database [5].

Through the utilization of a webhook, the process of fulfilling an objective is accomplished. A matched intent API connection request will be received only when the activation of the webhook agent successfully [4].

### 5.1 DIALOGFLOW WORK FLOW & ARCHITECTURE

Fig 6: Flow chart of the system [9]

Fig 6, depicts a conversational AI built using Dialogflow. On the left, there is an end-user with an input of data. The data is then processed to match their intents on the right-hand side of the diagram. Thereafter, a webhook request is sent to the fulfillment service, which looks up in external APIs or databases for an appropriate response. Subsequently, it replies to the end-user with a response via an output channel.

Fig 7: A diagram illustrating the flow of the interaction [4]

Fig 7, It is an image of a conversational AI through Dialogflow. Processed, and the matched data is input with intents. Then, a webhook request is sent to the fulfillment service, which, in turn, uses external APIs or databases to respond to that and sends it back to the end-user via an output channel [4].
Flowchart: Fig 8, A process of a chatbot conversing with the user to place or track an order is shown below. A user will be able to create a new order and track an order. After that, the chatbot will collect details of the order, process them in the backend, update the user on the status of the order, and train the model in the background as well with data from interactions of users for performance improvement.

Figure (9), talks about the Dialogflow agent receiving a user request, passing the same to a Cloud Function, which uses the intent parameters to ask for data from an external database, and then it posts a response payload back to the agent, which frames a response from the user.
After understanding Dialogflow key concept, we can design and develop our bot for a fictitious food service.

5.2 Using the Dialogflow Console

**Important:** To access the Dialogflow interface, you'll first need to create a Google account and set up your environment. The first time a user interacts with it, they will be prompted to log in to their account and initiate the setup if it hasn't been done already.

An agent is first built, designed, and trained within the Dialogflow console before any integration with other services can take place. However, this has never been a limitation for the most sophisticated and universal features developers can find in the console. Dialogflow provides very fast API endpoints that can empower any developer to use the features found in it programmatically. This allows flexible ways of working with agents, training and creating intents, and handling contexts all through the code. Using the FAST API, developers can easily integrate the Dialogflow feature within custom applications or even integrate it into existing systems without totally relying on the interface of the console. In our traversal through the console, we are going to add features that will make the agent be kind of a customer service representative for a meal delivery business. This agent shall take new orders, show available meals, and provide details of a desired meal. The interaction flow diagram shows the agent we'll be developing and how a user can buy a meal, or see what meals are offered, then buy one of them.

6. PROCEEDURE EXECUTION

Let's break it down into smaller, manageable tasks to help you get started

1. **Set Up Dialogflow**

   **Create a Dialogflow Account:**

   Sign up for Dialogflow at Dialogflow using your Google account. Create a new agent (e.g., "FoodOrderAssistantBot").

   **Define Intents:**

   Create various intents to handle user requests like ordering food, checking order status, and cancelling orders.

   Example intents: "Order Food", "Check Order Status", "Cancel Order".

   **Define Entities:**

   Create entities to capture specific information such as food items and quantities. Example entities: "Food Item", "Quantity".

2. **Set Up Backend with FastAPI and MySQL**

   **Install Required Libraries:**

   Use Python package manager to install libraries like FastAPI, SQL Alchemy, MySQL Connector, etc.

   **Initialize FastAPI:**

   Create a FastAPI application with endpoints to handle various functionalities like creating orders, updating order status, etc.

   **Database Setup:**

   Install MySQL and set up a database. Create necessary tables for storing order information.

3. **Connect Dialogflow to Backend Using Webhooks**

   **Webhook Configuration:** In Dialogflow, enable webhook fulfilment and set the URL to FastAPI endpoint that will handle the webhook requests.

   **Create Webhook Endpoint in FastAPI:**

   Implement an endpoint in FastAPI that will process the requests from Dialogflow, extract relevant information, and interact with the database to fulfil user requests.
4 **Use Ngrok to Expose Local Server** Install Ngrok: Download and install Ngrok from Ngrok.

**Exposé Local Server:**
Use Ngrok to expose your local FastAPI server to the internet. Ngrok provides a public URL that you can use in Dialogflow webhook configuration.

5 **Implement Backend Functionality Order Management:**
Implement functionality in FastAPI to handle order creation, update, and retrieval. Ensure these actions interact with the MySQL database.

6. **NLP Integration:**
Use Dialogflow’s NLP capabilities to parse user requests and determine user intent. Map these intents to corresponding backend functions in FastAPI.

7. **Testing and Deployment**
**Testing:** Thoroughly test the chatbot by interacting with it in the Dialogflow console. Ensure that all intents are correctly recognized and fulfilled.

**Deployment:** Deploy your FastAPI application to a cloud service provider like AWS, Google Cloud, or Heroku. Update Dialogflow’s webhook configuration to point to the deployed application’s URL.

7 **CODE IMPLEMENTATION**

In this section, we will give the implementation details of our chatbot system. Our chatbot system itself uses a stream of technologies and libraries to ensure smooth usage by the user. The following code snippets and diagrams illustrate the main components of our system.

**FastAPI:** it’s a high-performance web framework to build APIs with Python 3.7+. It's really fast, thanks to Starlette and Pydantic. It supports asynchronous programming and automatically creates interactive API documentation.[6]

![Fig 10: Fast Api initialization](image)

**Pydantic:** For data validation and settings management, this particular piece of software written in Python ensures that data matches the correct type and structure, and emits clear error messages. It easily switches Python's own constructions for JSON.

**MySQL Connector/Python:** It allows Python applications to interface with MySQL databases. This package is a pure implementation in Python, compatible with Python from version 3.6 upwards, and supports connection pooling.[7]

![Fig 11: MySQL connection code](image)
Uvicorn: Lightning-fast ASGI server implementation in Python. Fully ASGI-compliant, supporting HTTP/2, WebSocket’s, and all sorts of development and production setups

![Uvicorn installation & running main file code](image1.png)

Fig 12: Uvicorn installation & running main file code

```python
from fastapi import FastAPI, Request
from pydantic import BaseModel
import mysql.connector
import dialogflow_v2 as dialogflow
```

Fig 13: Importing Libraries

The fig (13), it shows Python code that imports various libraries for building a web application using FastAPI, connecting to a MySQL database, and integrating with Dialogflow

![Ngrok's Tunnel Port Number](image2.png)

Fig 14: Ngrok’s Tunnel Port Number

In Fig 14 is shows how, Ngrok HTTPS 8000 End exposes a secure tunnel to create a secured HTTPS connection via a safe tunnel connection from all HTTP traffic on port 8000.[10]

```python
def order_food(req):
    params = req['queryResult']['parameters']
    food_item = params['food_item']
    quantity = params['quantity']
    # Save order to database
    cursor.execute("INSERT INTO orders (food_item, quantity) VALUES (%s, %s)")
    db.commit()
    return {'fulfillmentText': f'Your order for {quantity} {food_item} is ready!

def track_order(req):
    order_id = req['queryResult']['parameters']['order_id']
    cursor.execute("SELECT status FROM orders WHERE id = %s", (order_id,))
    status = cursor.fetchone()
    return {'fulfillmentText': f'Your order status is: {status[0]}'}
```

Fig 15: Tracking & Ordering Code

In Above Fig 15: talks about the Python functions for generating an order of food items when a request object is passed. Extraction of the food item, quantity, and order ID from the request is done; it saves the order in the
database. It will finally return a message back to the user on order confirmation. This function reads the status of an order with a given order ID in the request and returns a message regarding the current status of that order.

8 GLOBAL CHATBOT MARKET TRENDS & FORECASTS

The global chatbot market has been experiencing significant growth in recent years, as the demand for automation and digital transformation grows. It tends to review some of the trends and forecasts of the market, outlook areas in its growth provision, and other key milestones.

![Figure 16: Global Chatbot Market 2020-2024][12]

This graph indicates the growth of the global chatbot market from 2020 to 2024. It will grow at a CAGR of almost 29%. On account of this, the growth will add $1.11 billion in incremental value. The market is fragmented and highly competitive, with several players contributing to its growth.

![Figure 17: Predicted market growth][11]

The trend of the global chatbot market from 2022 to 2032. A horizontal axis for years and a vertical for market size in USD billion. The market has been on an increasing trajectory for the last ten years and is expected to leap from $0.84 billion in 2022 to $4.9 billion by 2032. Even with a flat growth rate, the market size increases by $0.5 - 1 billion annually. This means that, with AI innovation, digital channel adoption, and the need of businesses to support their customers 24/7, demand for chatbots will increase.

Notable milestones include:

2025: $2 billion mark—decent adoption indicated.
2027: $3.5 billion; compound annual growth rate of 25%.
2030: $4 billion; this will be a proof of a more mature market. This clearly shows the steep growth in the global chatbot market, driven by increasing demand and strengthening technology.[11]
9 RESULTS

This section presents a restaurant website with a Dialogflow-powered chatbot, offering a user-friendly interface for customers to interact with the restaurant and handle tasks like orders and menu information.

Fig 18: Website Front End

Fig 19: Website Front End with Chatbot Integrated

In the above (fig’s 18,19), the restaurant website that incorporates a chatbot powered by Dialogflow, which can inform clients on all matters from menu items to hours of operation and reservations, among others. There is the feeling of user-friendliness that comes with the modern design in this restaurant.

Fig 20: DialogFlow Messenger Interaction
Fig (20), describe an ordering interface of chatbot variety, guiding a user by specifying food items and their quantities. The examples are taken, as well as what it has listed that is available on the menu, and the user can add or delete such items. It creates a conversation environment.

10 CONCLUSION

This research paper is an all-inclusive attempt at developing a chatbot, powered by NLP for order assistance using Dialogflow. The study was effective in serving as a solution to the loopholes of traditional ordering systems by elaborating on ways AI-powered chatbots can improve customer service and raise business efficiency.

**The key achievements and benefits of this research include:**

**Improved Customer Experience:** Accurately addressing customer inquiries and minimizing operational costs, the chatbot significantly reduces response times, ultimately improving customer satisfaction and loyalty.

**Increased operational efficiency:** The automation of the order processing mechanism through a chatbot relieves some workload from the human force and engages it in more complex assignments.

**Data-Driven Insight:** Such chatbots can also give a hold to very valuable customer data regarding ordering preferences and feedback, which helps in business strategies, marketing efforts, and personalization of the customer experience.

**Scalable and cost-effective:** since the design of the chatbot solution allows many businesses to afford an increase in volumes of orders with no significant investments in their infrastructure.

**Future Potential:** The chatbot is inevitably going to be more user-friendly and feature-rich in many aspects, with future development credited to its integration with social media platforms, voice recognition systems, and advanced payment systems.

The market can benefit from this research in several ways, including:

- Improved Customer Satisfaction and Loyalty.
- Increased operational efficiency and productivity.
- Enhanced customer engagement and personalized experiences
- Data-driven insight into business strategy formulation and improvement of marketing efforts.
- Scalability and cost efficiency, making it a very attractive solution for businesses regardless of size.
- Potential for improvement and future integration to other technologies.

Over all, Integration of NLP chatbots into the food ordering industry may change how business happens at companies. This increase in efficiency, cost reduction, and better customer experiences will diffuse into a wide array of industries.

11 FUTURE SCOPE

In the near future, chatbots are much expected to develop from their current saturated capabilities toward better user experiences in the different fields of business. Expanding on the compatibility to include mobile applications and popular social media channels makes for another way to increase accessibility and reach for chatbots. Provided the bot is able to make recommendations for orders tailored to a user's interests and previous orders, the user engagement factor could be greatly enhanced, leading to increased sales. The integration of safe and smooth payment gateways into the chatbot interface will ensure convenience and reliability for the user. Improvements in algorithms for natural language processing and voice recognition technology could be used to implement more complex and user-friendly interactions, thus coping with a greater variety of users and improving user satisfaction.

Future improvements can include:
Compatibility with Multiple Platforms:
For example, integration with social media platforms such as Facebook Messenger and Instagram will allow customers to order food directly from their social media accounts. This could streamline the process greatly and increase engagement.
Achievement: This could be achieved through partnerships with social media platforms and developing API integrations.[2]

Personalization and Recommendations:
For instance, a recommendation engine that suggests menu items based on user interest and previous orders will go a long way in increasing customer delight and upselling. For example, if he ordered vegetarian dishes frequently, it would indicate new vegetarian options or even suggest items that would complement his order.
Achievement: Personalization of recommendations by means of machine learning algorithms and the analysis of user data.

Payment Gateway Integration:
Example: Integrate PayPal, Stripe, Apple Pay directly into chatbot messages for secure transactions. This eventually can draw in easier checkout options to build trust with this experience where the user doesn't need to leave the chat to complete his transaction.
Achievements: Integrated payment gateways and implemented security standards including PCI DSS.

Improvement in voice recognition and artificial intelligence: This could be realized by simply improving the NLP of the chatbot and then incorporating voice recognition technology. For instance, customers could place orders or raise any form of inquiries, like one would do using a virtual assistant such as Siri or Google Assistant.
Achievement: Research in the domain of NLP and voice recognition, featuring cooperation with providers of artificial intelligence technologies.[1]

REFERENCES