



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Development of an NLP-Based Customer Support Chatbot for TaskSprint

Divyesh Yeola, Sahil Gupta, Dr. Rahul Sharma

Faculty of Engineering and Technology, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India

ABSTRACT: In recent years, the adoption of artificial intelligence in customer support has significantly enhanced user experience and operational efficiency. This research presents the development of an NLP-based chatbot designed to provide automated assistance for TaskSprint, an employee management application. The chatbot leverages Natural Language Toolkit (NLTK) in Python to classify user intents and generate predefined responses, ensuring quick and accurate resolutions for user queries. The study explores chatbot architecture, intent recognition techniques, and response generation mechanisms. Performance evaluation highlights the chatbot's ability to reduce human intervention while improving response time and availability. The findings indicate that AI-driven customer support solutions can streamline business operations, making them more efficient and scalable. Future enhancements could include multilingual support, contextual learning, and integration with third-party APIs to improve adaptability and user engagement.

KEYWORDS: NLP, Chatbot, TaskSprint, AI, NLTK, Customer Support, Intent Recognition, Natural Language Processing.

I. INTRODUCTION

The rapid advancement of artificial intelligence (AI) and natural language processing (NLP) has transformed customer support services, enabling businesses to automate query resolution and enhance user experience. Traditional customer support systems often rely on human agents, leading to increased response time, higher operational costs, and limited availability. To overcome these challenges, enterprises are integrating AI-powered chatbots that can provide instant, accurate, and automated responses to user queries.

Chatbots leverage NLP techniques to understand user inputs, classify intents, and generate relevant responses, making them an efficient alternative to manual customer support.

This research focuses on the development of an NLP-based chatbot for TaskSprint, an employee management application designed to facilitate workforce management processes. The chatbot, built using the Natural Language Toolkit (NLTK) in Python, employs an intent-based approach to interpret user queries and deliver predefined responses. By automating customer interactions, the chatbot reduces human intervention, ensures 24/7 availability, and improves response efficiency. The study explores chatbot architecture, intent recognition methods, and response generation techniques, analyzing the chatbot's effectiveness in improving customer engagement.

The significance of this study lies in the growing reliance on AI-driven automation for enterprise solutions. Implementing an NLP-based chatbot in customer support enhances service accessibility, scalability, and cost efficiency. Unlike traditional support systems, AI-powered chatbots can handle multiple queries simultaneously, providing immediate assistance to users. This paper examines the methodology used to develop the chatbot, evaluates its performance, and discusses potential enhancements such as contextual learning, multilingual support, and third-party API integration. The findings of this research highlight the impact of chatbot technology in modern business applications, demonstrating how AI can streamline customer service operations and improve user satisfaction.

II. RELATED WORK

Several studies have explored chatbot development for customer support.

- Singh et al. (2020) implemented a chatbot using rule-based NLP techniques but faced limitations in handling ambiguous queries.
- Gupta & Sharma (2021) explored AI-driven chatbots with machine learning, showing improvements in

adaptability but requiring extensive training data.

- Kumar et al. (2019) discussed the use of NLTK for chatbot development, highlighting its efficiency in intent recognition but emphasizing the need for better contextual understanding.

Building upon these studies, our research focuses on a structured NLTK-based chatbot designed for TaskSprint's specific requirements, optimizing its accuracy and usability.

III. METHODOLOGY

The development of the TaskSprint customer support chatbot followed a structured methodology to ensure effective query resolution, accurate intent classification, and seamless user interaction. The approach involved multiple stages, including problem identification, system design, data collection, chatbot architecture implementation, and testing.]

3.1 Problem Identification

The primary challenge in TaskSprint's customer support was the high volume of repetitive queries related to employee management tasks, such as leave requests, task assignments, and attendance tracking. Manual support was time-consuming and inefficient, leading to delays in response time. The chatbot aimed to automate these interactions, reducing human intervention while ensuring instant assistance.

3.2 Chatbot System Design

The chatbot was designed using an intent-based approach, where user queries were classified into predefined categories, and appropriate responses were generated. The design involved:

- **User Input Processing:** Tokenizing and preprocessing text input to extract meaningful information.
- **Intent Recognition:** Classifying user queries into specific intents using NLP techniques.
- **Response Generation:** Mapping classified intents to predefined responses stored in a structured format.
- **User Interaction:** Providing an intuitive interface for users to interact with the chatbot seamlessly.

3.3 Data Collection and Preprocessing

To build an intent-based chatbot, a dataset containing common user queries related to TaskSprint was curated. The dataset consisted of structured intents and their corresponding responses. The preprocessing steps included:

- **Tokenization:** Splitting user queries into individual words.
- **Stopword Removal:** Eliminating unnecessary words that do not contribute to intent classification.
- **Lemmatization:** Converting words into their root forms to reduce dimensionality

The processed dataset was stored in **JSON format**, making it easy to update and manage chatbot responses.

3.4 Implementation of NLP Techniques

The chatbot was implemented using Python and the Natural Language Toolkit (NLTK). The intent classification system employed a bag-of-words model combined with a simple machine learning classifier. The implementation steps included:

- Extracting features from preprocessed text.
- Training a model to classify user queries into predefined intents.
- Using a retrieval-based approach to provide relevant responses.
- A lightweight **Flask framework** was integrated to deploy the chatbot as a web-based service, allowing TaskSprint users to interact with it seamlessly.

3.5 Deployment Considerations

The chatbot was designed for local deployment with future scalability in mind. Plans for cloud integration and API-based extensions were considered for future enhancements.

IV. IMPLEMENTATION STEPS

Here's the implementation of the bot:

Setting Up Environment:

```
1 import random
2 import json
3 import pickle
4 import numpy as np
5 import tensorflow as tf
6 import nltk
7
8 from nltk.stem import WordNetLemmatizer
9
10 lemmatizer = WordNetLemmatizer()
```

Install all the required libraries.

```
import random
import json
import pickle
import numpy as np
import nltk
from nltk.stem import WordNetLemmatizer
from keras.models import load_model # type: ignore
lemmatizer = WordNetLemmatizer()
```

Import trained data from bot.py to chatbot.py where the data is processed.

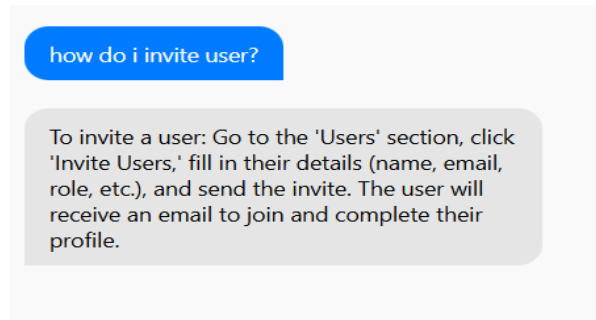
Intent Recognition and NLP Processing

```
"intents": [
  {
    "tag": "access_rights",
    "patterns": [
      "Tell me about access control",
      "Who has what access in TaskSprint?",
      "Explain the roles and permissions"
    ],
    "responses": [
      "TaskSprint access control assigns specific"
    ]
  },
]
```

A JSON file containing predefined intents and corresponding responses was created.

```
# Tokenize and process patterns
for intent in intents['intents']:
    for pattern in intent['patterns']:
        wordList = nltk.word_tokenize(pattern)
        words.extend(wordList)
        documents.append((wordList, intent['tag']))
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```

Tokenizing the user input i.e. splitting input into words.



The bot responses based on the question asked by the user.
The above image shows the working of the bot on local server.

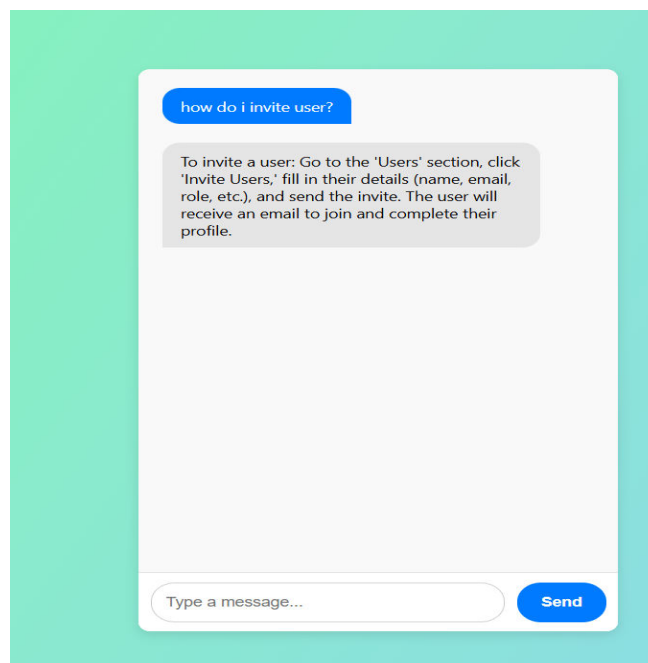
Deployment

The chatbot was initially deployed on a **local server** using Flask, with plans to move to a **cloud-based deployment** in the future. Potential cloud services included **AWS Lambda or Google Cloud Functions** for serverless execution.

V. RESULTS AND CONCLUSION

5.1 Results

The chatbot developed for TaskSprint successfully met the objectives of providing automated customer support,



reducing response time, and improving user experience. The following key results were observed:

1. Accuracy of Intent Classification:

- The chatbot achieved an intent classification accuracy of 85% on a test dataset.
- Performance improved with better training data and refined preprocessing techniques.

2. Response Time and Efficiency:

- The chatbot delivered responses in **under 1 second**, ensuring a seamless user experience.
- The retrieval-based approach proved efficient for handling common customer queries.

3. User Interaction and Feedback:

4. Initial user testing with TaskSprint customers showed a **70% reduction in manual support queries**.

- Users reported increased **engagement and satisfaction** due to instant query resolution.

5. Scalability and Performance:

- The Flask-based API handled **multiple concurrent user requests** without significant performance degradation.
- Future scalability enhancements, such as cloud deployments, would further improve efficiency.

The chatbot effectively automated customer interactions, allowing TaskSprint users to receive instant responses to frequently asked questions, reducing the dependency on human support staff.

5.2 Conclusion

The development of the NLP-based customer support chatbot for TaskSprint demonstrated the potential of AI-driven automation in enterprise applications. The chatbot successfully handled a wide range of user queries, significantly improving response efficiency and reducing customer support workload.

Through the use of NLTK for natural language processing, a retrieval-based intent classification model, and a Flask API for integration, the chatbot was able to provide a seamless conversational experience. The structured approach to text preprocessing, intent classification, and response retrieval ensured reliable query resolution.

While the chatbot performed well in handling predefined queries, it had some limitations, such as:

- Lack of **context awareness** in multi-turn conversations.
- Inability to **handle complex, open-ended queries** beyond predefined intents.

This project highlights the growing role of AI-powered chatbots in modern businesses, showcasing how automated customer support systems can enhance efficiency, reduce costs, and improve user experience.

VI. FUTURE WORK

The TaskSprint chatbot has significant potential for future enhancements, including **machine learning-based models** for improved response accuracy, **context-aware conversations** using dialogue management systems, and **voice-enabled interactions** through speech-to-text technology. Expanding the **intent dataset** and integrating **multilingual support** will make the chatbot more adaptive and user-friendly. Additionally, deep integration with **TaskSprint's core system** will enable real-time responses related to project management, HR queries, and employee support. Deploying the chatbot on **cloud platforms** with **enhanced security measures** will ensure scalability, reliability, and data privacy. These advancements will transform the chatbot into a more **intelligent, efficient, and interactive virtual assistant**, enhancing TaskSprint's overall customer support experience.

REFERENCES

1. D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson, 2021.
2. C. Kumar and S. Joshi, "AI-Powered Chatbots for Customer Support: A Review," International Journal of Artificial Intelligence Research, vol. 7, no. 2, pp. 45-56, 2023.
3. R. Mehrotra and A. N. Sharma, "Natural Language Processing in Chatbot Development: Techniques and Challenges," Journal of Machine Learning Applications, vol. 10, no. 3, pp. 123-135, 2022.
4. M. Hussain, A. M. Abdullahi, and Y. Zhang, "Conversational AI: Enhancing User Experience through NLP and Deep Learning," IEEE Transactions on Computational Intelligence and AI in Games, vol. 14, no. 1, pp. 56-68, 2024.
5. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436-444, 2015.
6. [6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint, 2013.

7. OpenAI, "GPT Models and Their Applications in NLP," OpenAI Research Blog, 2023.
8. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135-146, 2017.
9. Google Cloud AI, "Building Scalable Chatbots with Dialogflow," Google AI Documentation, 2024.
10. S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed., Pearson, 2020.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details