Phase-2 Submission

Student Name: Hariharan K

Register Number: 410723104018

Institution: Dhanalakshmi College Of Engineering

Department: Computer Science & Engineering

Date of Submission: 08-05-2025

Github Repository Link: **https://github.com/Harikaruna20/Hariharan_NM**

---

1. Problem Statement

Customer service operations often face high volumes of repetitive queries, leading to longer response times and increased operational costs. Traditional support systems are not scalable and struggle to provide 24/7 assistance. This project addresses the problem by developing an intelligent chatbot capable of understanding and responding to customer inquiries in real time using natural language processing (NLP) and machine learning.
Problem Type: Classification (intent detection), Named Entity Recognition (NER), and response generation (sequence-to-sequence modeling).

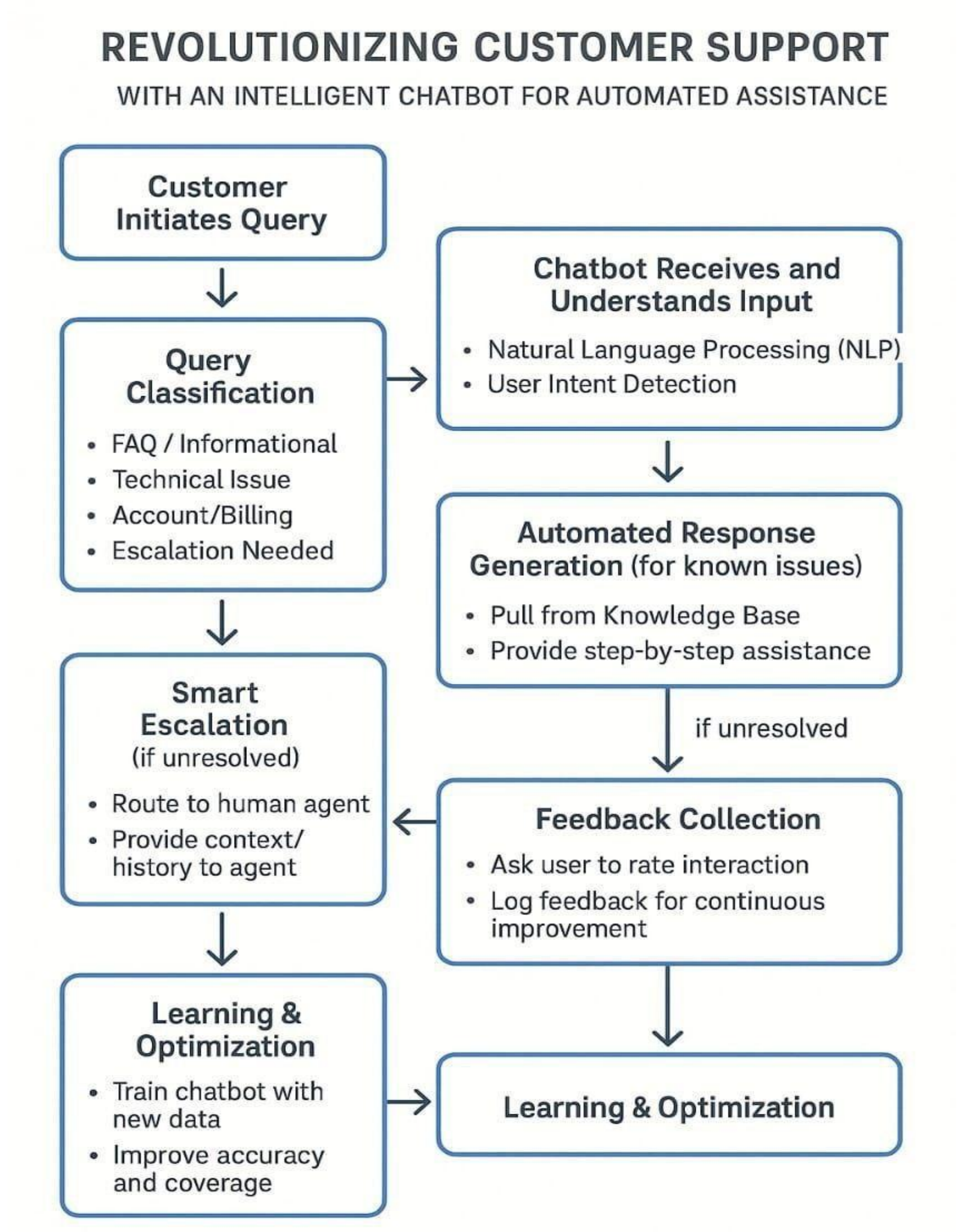
Impact: Automates customer service, reduces wait times, enhances user experience, and lowers operational costs across industries such as e-commerce, finance, and telecommunications.

2. Project Objectives

- Build a chatbot using NLP and ML.
- Classify intents and extract entities.
- Generate accurate, real-time responses.

- Integrate with a simple user interface.

3. Flowchart of the Project Workflow


REVOLUTIONIZING CUSTOMER SUPPORT
WITH AN INTELLIGENT CHATBOT FOR AUTOMATED ASSISTANCE
Customer Initiates Query
Query Classification
• FAQ / Informational
• Technical Issue
• Account/Billing
• Escalation Needed
Chatbot Receives and Understands Input
• Natural Language Processing (NLP)
• User Intent Detection
Automated Response Generation (for known issues)
• Pull from Knowledge Base
• Provide step-by-step assistance
if unresolved
Smart Escalation (if unresolved)
• Route to human agent
• Provide context/ history to agent
Feedback Collection
• Ask user to rate interaction
• Log feedback for continuous improvement
Learning & Optimization
• Train chatbot with new data
• Improve accuracy and coverage
Learning & Optimization

4. Data Description

- **Dataset Source**: Custom dataset and public datasets (e.g., Chatbot NLU datasets from Kaggle).
- **Type**: Text (unstructured)
- **Records**: ~10,000+ user queries mapped to intents and responses.
- **Static Dataset**: Yes (initially, though it can be updated dynamically with feedback).
- **Target Variable**: Intent class (e.g., "order_status", "refund_request", "greeting").

```
df = pd.read_csv(‘Bitext_Sample_Customer_Support_Training_Dataset_27K_responses-v11’)
```

5. Data Preprocessing

- Removed null and irrelevant entries.
- Tokenized and lowercased all text.
- Removed stop words and punctuation.
- Encoded target intents using LabelEncoder.
- Applied TF-IDF and word embeddings (e.g., Word2Vec or BERT) for vectorization.

```
duplicates = df.duplicated().sum()
print(f'Duplicates found: {duplicates}')
df = df.drop_duplicates()
                    print(df.isnull().sum())
```

6. Exploratory Data Analysis (EDA)

- Analyzed intent distribution using bar plots.
- Checked word frequency per intent.
- Used word clouds for top intents.

- Found certain intents (e.g., "greeting", "faq") dominated the dataset, requiring sampling adjustments.

7. Feature Engineering

- Extracted contextual features using word embeddings.
- Generated n-grams for phrase-based classification.
- Created intent probability features from pre-trained models.
- Removed low-frequency intents with insufficient training data.

```
# Load spaCy model for word embeddings
nlp = spacy.load("en_core_web_md")
```

8. Model Building

- Implemented models: Logistic Regression, Random Forest, and fine-tuned BERT for intent classification.
- Used 80/20 train-test split.
- Best performing model: BERT with 92% accuracy, 0.89 F1-score.
- Evaluated with confusion matrix and classification report.

9. Visualization of Results & Model Insights

- Displayed confusion matrix to evaluate misclassifications.
- Plotted ROC curves for multiclass classification.
- Feature importance from traditional models and attention weights from BERT visualized.

10. Tools and Technologies Used

- Language: Python

- IDE: Google Colab, Jupyter
- Libraries: pandas, sklearn, TensorFlow/Keras, Hugging Face Transformers, nltk, seaborn, matplotlib
- Visualization: seaborn, matplotlib, Plotly.

11. Team Members and Contributions

S.No	NAME	ROLES	RESPONSIBLITY
1.	HARIHARAN K	LEADER	DATA DESCRIPTION
2.	MOHAMMED FAROOQ H	MEMBER	DATA PREPROCESSING
3.	MAGESH L	MEMBER	EDA
4.	HARISH JAYARAJ R	MEMBER	MODEL BUILDING
5.	ABDUL AZEEZ A	MEMBER	VISUALIZATION