Final Report:
AI-POWERED CHATBOTS FOR CUSTOMER S ERVICE ANALYTICS

A document report was submitted to the School of Computer Science and Engineering in partial fulfilment of the requirement for the degree of
Big Data Analytics Course.

Instructor:
Assoc. Prof. Nguyen Thi Thuy Loan

In specialization by

Mr: Le Thanh Phuong Nam

ID (HCMIU) : ITITWE19025_ Student number (UWE) : 23083609

International University - Vietnam National University HCMC

University of the West of England

April 2025

Github link: https://github.com/NamBobby/BDA_WebDevelopment
Intent Classification Model link: https://colab.research.google.com/drive/14EGEIpFzVs_suTrZl7c37SGUKloouRdr?usp=sharing
Sentiment Analysis Model link: https://colab.research.google.com/drive/1AS4lvKaDEZ4-arObAuxESAbEr-EbnNEr?usp=sharing

Full Submission link: https://drive.google.com/file/d/14S-xWFXx2aey99RaeYWJddzNwlHiORIY/view?usp=sharing

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
LSTM	Long Short-Term Memory
API	Application Programming Interface
UI	User Interface
UX	User Experience
ML	Machine Learning
Flask	Python-based web framework
ReactJS	JavaScript library for UI development
GPU	Graphics Processing Unit
AdamW	Adaptive Moment Estimation with Weight Decay optimizer
Attention Mask	A mechanism in transformers to indicate which tokens should be processed
JWT	JSON Web Token
ORM	Object-Relational Mapping
SPA	Single Page Application
NLPaug	Natural Language Processing Augmentation
CPU	Central Processing Unit
MVC	Model-View-Controller
OTP	One-Time Password
SQL	Structured Query Language

Table 1. List of Abbreviations

Chapter One: INTRODUCTION

1. Project Overview

The project designs a system improvement for chatbot service efficiency employing NLP to interpret customer expressions and determine their intentions and feelings. Even today, there are a significant number of chatbots in existence that do not understand things when need be and as a result, they are unable to provide any reasonable responses to the customers (Dr. R. Ramki et al., 2024). The arresting and emotion-sensitive chatbot will be trained to develop and deploy an AI-driven system to build a better classification scheme for handling customer queries and recognition of moods, which will ultimately have a positive influence on responses. Although it will not match the original requirements provided by the instructor on this topic, I thought that instead of building a model and visualizing the results in Jupyter Notebook, Power BI or Google Colab, I decided that I should design a website to visualize the work results that could be more practical and interactable.

2. Problem Definition

Organizations experience challenges when processing large amounts of incoming customer inquiries through efficient channels. The use of chatbots helps decrease operational burdens yet they return generalized or incorrect information. The research goal entails developing a chatbot model that provides both correct customer intent analysis and sentiment interpretation to create more realistic and useful conversations. An improvement in chatbot accuracy enables better customer satisfaction along with less requirement for human intervention.

3. Background and Motivation

AI-powered chatbots have gained wide use in business customer service because organizations increasingly depend on automation. The current chatbots demonstrate limited capability to recognize both customer intent and emotional expressions resulting in inadequate customer satisfaction. The machine learning approach will transform chatbots into sophisticated systems that provide better efficiency and responsiveness to support user needs better. The sentiment-understanding capability of a chatbot enables it to modify its responses through user emotional detection leading to better client interaction. A model needs development to provide flexible adaptation within industries including e-commerce healthcare and finance.

4. Objectives

This project outcome will deliver a proficient chatbot system that efficiently classifies customer intents and modifies its responses based on sentiment detection capabilities. The development includes building a performance visualization dashboard for the chatbot. A final deliverable includes both a comprehensive report about the methodologies as well as a study of AI model efficiency and the measurement of customer service operational effectiveness. The implemented project functions to enhance business automated customer service systems while maintaining excellent customer interaction quality.

5. Approach and Tools

This project leverages Python and several NLP and ML libraries, including NLTK, SpaCy, TensorFlow, and Hugging Face Transformers, to build AI models for chatbot response optimization. Instead of using Jupyter Notebook, Power BI, or Google Colab for visualization, an interactive web-based dashboard is being developed using Flask (backend) and ReactJS (frontend) for real-time monitoring and analysis of chatbot interactions. The entire process of training models for the project was done entirely on Google Colab.

The project follows a structured approach:

1. Data Collection: Find some datasets including customer interactions, common support chatbot queries, and support responses. With these approaches to dataset diversity, I could improve model generalization.
2. Data Preprocessing: Apply text cleaning techniques, including stopword removal, tokenization, and lemmatization to format and structure data for compatibility with machine learning models.
3. Model Development:
    - Intent Classification Model: Train a BERT-based model to accurately categorize user queries into predefined intent categories.
    - Sentiment Analysis Model: Develop a DistilBERT-based model to detect customer sentiment and enhance chatbot responses accordingly.
4. Testing & Evaluation: Measure model accuracy using accuracy, precision, recall, and F1-score and improve performance through fine-tuning hyperparameters.
5. Web-Based Visualization & Deployment: Develop an interactive admin dashboard using ReactJS (frontend) and Flask (backend). The dashboard will display chatbot performance analytics, sentiment trends, and user interaction insights in real time. Enable administrators to track chatbot accuracy, response effectiveness, and overall user satisfaction.

6. Project Timeline

The project duration extends across 9 weeks (Table 1), which will be organized through specific task breakdowns (Figure 1) from the proposal report before. This mid-term progress report has rescheduled some specific tasks due to the revised requirements of this project but still ensures submission of the report in week 5. Finally, the final submission is still set for week 9 on 20/4/2025.

Task	Start Date	Due Date	Duration (Days)	Description
Proposal Writing & Submission	**09/02/2025**	**23/02/2025**	**15**	**Define project scope, objectives, and dataset approach.**
Research AI Chatbots	09/02/2025	16/02/2025	8	Study chatbot trends and existing models.

Define Scope & Objectives	09/02/2025	16/02/2025	8	Set project goals and expected outcomes.
Draft Proposal	16/02/2025	22/02/2025	7	Write and refine project proposal.
Submit Proposal	23/02/2025	23/02/2025	1	Submit proposal for approval.
Dataset Collection & Processing	**24/02/2025**	**01/03/2025**	**6**	**Collect and clean data for AI training.**
Identify Potential Datasets	24/02/2025	26/02/2025	3	Search for high-quality datasets.
Data Cleaning & Formatting	27/02/2025	01/03/2025	3	Process and structure data for training.
Model Development	**02/03/2025**	**13/03/2025**	**12**	**Train AI models for chatbot enhancement.**
Train Intent Classification Model	02/03/2025	07/03/2025	6	Develop model to categorize user queries.
Train Sentiment Analysis Model	08/03/2025	13/03/2025	6	Implement model to analyze emotions.
Evaluation the model	**14/03/2025**	**15/03/2025**	**2**	**Calculate accuracy.**
Midterm Progress Report	**16/03/2025**	**16/03/2025**	**1**	**Summarize progress and key findings.**
Testing & Optimization	17/03/2025	23/03/2025	7	Measure model performance and refine responses. Analyze accuracy and sentiment predictions.
Web Development	**24/03/2025**	**12/04/2025**	**21**	**Developing an web-application to visualize the result.**
Backend Development (Flask)	24/03/2025	30/03/2025	7	Developing APIs and integrating AI models into the backend.

Frontend Development (ReactJS)	31/03/2025	06/04/2025	7	Creating the user interface and integrating in with the backend.
Testing and Deployment	07/04/2025	12/04/2025	21	Conducting comprehensive testing and deploying the application.
Final Report & Submission	**13/04/2025**	**20/04/2025**	**8**	**Compile project results and submit final report.**

Table 2. Project Timeline

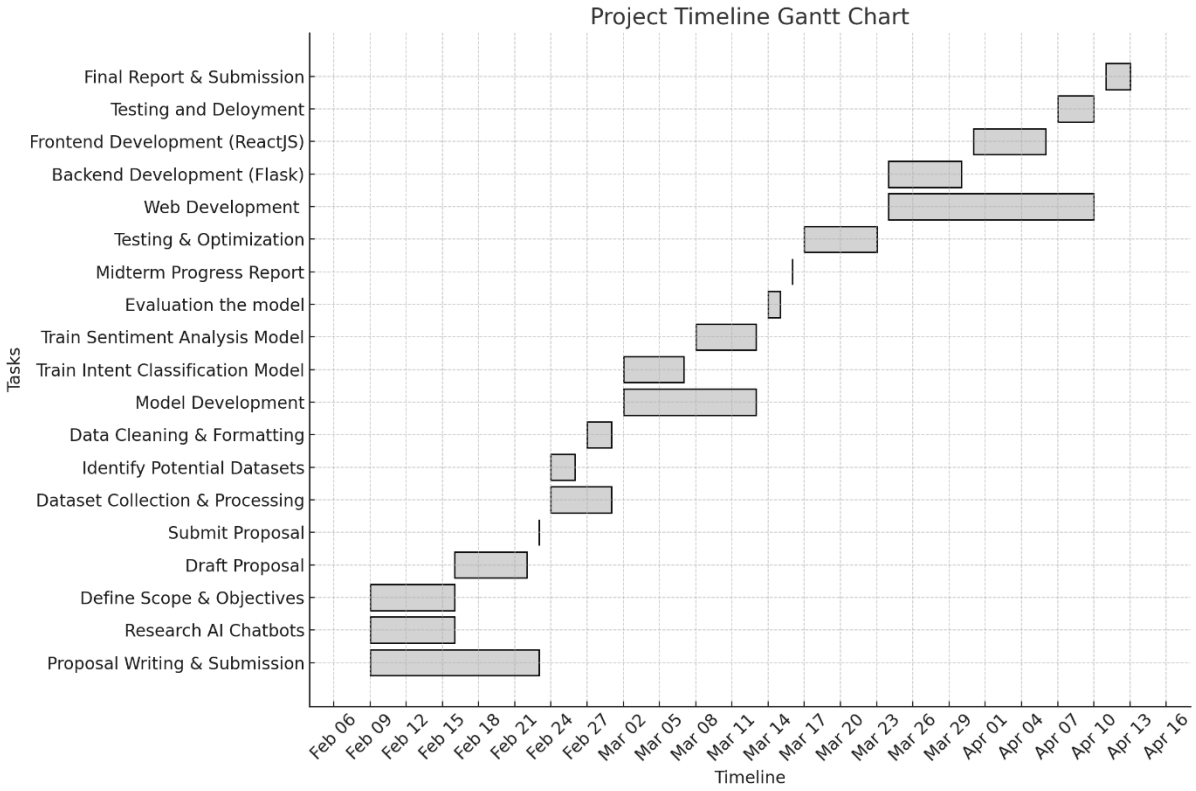


Figure 1. Project Timeline Gantt Chart

Chapter Two: LITERATURE REVIEW

1. Natural Language Processing Fundamentals

Modern NLP systems require text preprocessing as their initial foundational process. Research into stop word removal along with truncation or lemmatization methods conducted years ago showed how WordPiece tokenization in BERT and other advanced language model-based techniques increasingly take priority over traditional approaches (Chai, 2023). BERT's limited maximum token capacity of 64–128 tokens creates information loss in extended texts, thus, new solutions like Longformer and Reformer are applied. Additionally, filtering the English language with tools such as langdetect is a reasonable option to increase accuracy, especially when working on unprocessed multilingual data (Taufikin et al., 2024). In experimental systems, an organization must determine whether to use basic normalization techniques instead of complex methods such as NER and parsing due to their processing requirements and performance levels.

2. Deep Learning for NLP

Transformer models, particularly BERT and DistilBERT have replaced traditional RNN and LSTM models since they analyze text materials while processing simultaneously and detecting connections that exist beyond immediate proximity (Figure 2). BERT implements bi-directional self-attention to analyze context in a manner superior to ELMo or GloVe, which operate within one direction (Kora and Mohammed, 2023). Applying transfer learning through fine-tuning the pretrained BERT model significantly saves resources and training time (Han, Pang and Wu, 2021). The smaller DistilBERT model achieves performance equivalence to BERT through distillation methods that decrease its size, thus allowing effective usage in limited computing situations. Although these systems need GPUs for efficient training, they face limitations because Naive Bayes and SVM classical models can operate on traditional CPUs without any special requirements.

BERT Size	Transformer Layers	Hidden Size	Attention Heads	Parameters	Processing	Length of Train-ing
BERT Base	12	768	12	110M	4 TPUs	4 days
BERT Large	24	1024	16	340M	16 TPUs	4 days

Figure 2. Architecture of BERT model with multidimensional attention (Kora and Mohammed, 2023)

3. Text Data Augmentation Techniques

The method of data augmentation provides an effective solution to grow training datasets and enhance NLP model generalization performance when dealing with imbalances in data. Three data augmentation techniques are employed in practical tests, which include WordNet-based synonym replacement (SynonymAug), context-based word insertion with a language model (ContextualWordEmbsAug) and typo generation

(RandomCharAug). These methods (Figure 3) have much lower computational costs than popular techniques such as back-translation, while still generating reasonable data diversity (Shorten, Khoshgoftaar and Furht, 2021). The approach of replacing significant keywords from small classes through adversarial sampling leads to better model accuracy and decreased bias(Chen et al., 2023). However, data augmentation can reduce the naturalness of the text, especially if the context is not well controlled or the meaning of the word is changed.

Easy Data Augmentation	Short Example
Random Swap	I am jogging → I tiger jogging
Random Insertion	I am jogging → I am salad jogging
Random Deletion	I am jogging → I jogging
Random Synonym Replacement	I am jogging → I am running

Figure 3. Examples of easy data augmentation transformations (Shorten, Khoshgoftaar and Furht, 2021)

4. Model Evaluation and Validation

The evaluation of NLP models needs more than accuracy metrics because it handles imbalanced datasets. The evaluation system, made up of precision and recall combined with F1-score macro, proves best for analyzing multiclass classification when data distributions are uneven (Riyanto et al., 2023). Learning curves for bias–variance analysis provide a superior understanding of underfitting or overfitting than testing or validating only on the test set would deliver. The model gains consistency across different data partitions through the cross-validation method with special emphasis on Stratified K-Fold. Additionally, robustness testing is a way to assess the robustness of the system. The research conducted in 2024 demonstrates how random variation testing of models through typos enables the detection of weaknesses in actual implementation scenarios.

5. Related Work

A related study on “Application of AI-Driven Chatbots in E-Commerce Recommendation Systems for Targeted Customers in Product Selection” by Rahevar and Darji (2024) explores the application of AI-driven chatbots in e-commerce recommendation systems (Rahevar and Darji, 2024). The paper investigates how chatbots increase customer satisfaction levels during the purchasing process by enhancing product choices and user interactions and buying experiences.

Differences from This Project

The research above examines AI chatbots as shopping assistants for e-commerce yet this project intends to improve customer service interactions independent from product selection assistance (Rahevar and Darji, 2024)ii. The authors' work concentrates on recommendation engine algorithms to enhance e-commerce platform item selection. My project centers its strategy on processing customer demands and detecting their intents and analyzing their sentiment to enable chatbots with contextual emotional responses. The recommendation system they develop incorporates AI models that make recommendations by collecting user

activities and tracking purchasing patterns from their records. The project employs NLP models to enhance both chatbot responses and interactions during customer service support. The product recommendation function of their chatbot exists independently without sentiment detection capabilities or emotional response adjustment features. Sentiment-aware AI models form an essential part of this project because they enable the chatbot to adjust its responses according to customer emotions for providing better emotional support.

Chapter Three: METHODOLOGY

1. Data Collection and Preprocessing:

1.1. Dataset Selection

To develop an AI-powered chatbot that can understand user intent and emotion, I needed to prepare the Bitext Customer Support Chatbot Dataset [1] which was used for intent classification, while the Multiclass Sentiment Analysis Dataset [2] was selected to train the sentiment analysis model. The datasets form an essential basis for developing a chatbot which correctly understands user statements and modifies its responses by emotional content detection.

1.1.1. Bitext Customer Support Chatbot Dataset (for Intent Classification)

The dataset's structured format works best for intent classification because it helps the chatbot establish inquire nature signatures without delay. With a high structure, it is ideal for me to classify intents, ensuring that the chatbot can quickly and accurately determine the nature of the user's request:

- Instruction: The user's query.
- Intent: The specific action the user is requesting (e.g., "Track Order").
- Category: The general domain the request falls under (e.g., "Order Management").
- Response: The expected chatbot reply.

1.1.2. Multiclass Sentiment Analysis Dataset (for Sentiment Classification)

The chosen dataset served as emotion recognition training for the chatbot while being categorized into three sentiments among 41,643 text samples. The dataset was split into three distinct segments for model development: training (31,232 samples), validation (5,205 samples) together with test (5,206 samples) ensuring proper model development techniques. Detected changes in customer tone within customer interactions can be trained using this dataset to enable the chatbot proper emotion recognition.

1.2. Data Preprocessing

1.2.1. Preprocessing for Intent Classification Dataset

The raw dataset (Figure 4) needed cleaning together with a structural reorganization to reach its best model performance level. First, I removed placeholder values like {{Order Number}} as they did not contribute to intent recognition. Next, I converted all text to lowercase to standardize the input formats. An additional processing step involved column filtering (instructions, intents, categories and responses) to minimize memory usage and enhance processing speed. Finally, I encoded the "intent" and "category" labels into numeric values using LabelEncoder, making them compatible with the machine learning model's requirements (Figure 5).

[1] https://huggingface.co/datasets/bitext/Bitext-customer-support-llm-chatbot-training-dataset
[2] https://huggingface.co/datasets/Sp1786/multiclass-sentiment-analysis-dataset

```
<bound method NDFrame.head of       flags                                               instruction category  \
0         B   question about cancelling order {{Order Number}}    ORDER
1        BQZ  i have a question about cancelling oorder {{Or...    ORDER
2       BLQZ    i need help cancelling puchase {{Order Number}}    ORDER
3         BL         I need to cancel purchase {{Order Number}}    ORDER
4      BCELN  I cannot afford this order, cancel purchase {{...    ORDER
...      ...                                                ...      ...
26867     BL  I am waiting for a rebate of {{Refund Amount}}...   REFUND
26868    BIL  how to see if there is anything wrong with my ...   REFUND
26869   BLQZ  I'm waiting for a reimbjrsement of {{Currency ...   REFUND
26870     BL  I don't know what to do to see my reimbursemen...   REFUND
26871     BL  I need to know if there is anything new on the...   REFUND

            intent                                           response
0      cancel_order  I've understood you have a question regarding ...
1      cancel_order  I've been informed that you have a question ab...
2      cancel_order  I can sense that you're seeking assistance wit...
3      cancel_order  I understood that you need assistance with can...
4      cancel_order  I'm sensitive to the fact that you're facing f...
...             ...                                              ...
26867  track_refund  Thank you for sharing your situation regarding...
26868  track_refund  Ensuring the accuracy of your restitution is o...
26869  track_refund  Firstly, I genuinely understand the importance...
26870  track_refund  I've understood you're unsure about how to che...
26871  track_refund  It's completely understandable that you want t...

[26872 rows x 5 columns]>
```

Figure 4. Raw text samples for Intent Classification Dataset

```
                                instruction        intent category  \
0              question about cancelling order  cancel_order    ORDER
1    i have a question about cancelling oorder  cancel_order    ORDER
2                i need help cancelling puchase  cancel_order    ORDER
3                     I need to cancel purchase  cancel_order    ORDER
4  I cannot afford this order, cancel purchase  cancel_order    ORDER

                                          response  intent_label  \
0  I've understood you have a question regarding ...             0
1  I've been informed that you have a question ab...             0
2  I can sense that you're seeking assistance wit...             0
3  I understood that you need assistance with can...             0
4  I'm sensitive to the fact that you're facing f...             0

   category_label
0               6
1               6
2               6
3               6
4               6
Number of unique intent classes: 27
Number of unique category classes: 11
```

Figure 5. Cleaned text samples for Intent Classification Dataset

1.2.2. Preprocessing for Sentiment Analysis Dataset

My preprocessing system included various steps to get the data (Figure 6) ready for training operations. I began by replacing all missing or null values in the text field with empty strings to avoid errors during tokenization. The following processing

stage converted every text to lowercase and deleted special characters, which left only important punctuation marks such as periods, exclamation marks and question marks in place. A standard [URL] token took the place of all URLs while common contractions, including “won’t” and “I’ve”, both transformed into full forms “will not” and “I have” to preserve semantic meaning.

```
Sample data from training set:
      id                                               text  label sentiment
0   9536                  Cooking microwave pizzas, yummy      2  positive
1   6135  Any plans of allowing sub tasks to show up in ...      1   neutral
2  17697   I love the humor, I just reworded it. Like sa...      2  positive
3  14182                        naw idk what ur talkin about      1   neutral
4  17840         That sucks to hear. I hate days like that      0  negative
```

Figure 6. Raw text samples for Sentiment Analysis Dataset

A crucial step I added was language filtering. The text language identification function in langdetect automatically analyzed the inputs, which led to the removal of all samples written in languages other than English. The data quality showed major improvement because this measure standardized linguistic elements.

After applying the language filtering step using langdetect, I retained only English samples in each dataset partition. The analysis (Figure 7-8) shows how a major portion of texts that were not in English or did not show a language signature were excluded, resulting in 28,538 English texts for training and 4,758 for validation and 4,780 for testing.

Finally, the sentiment labels received a conversion through LabelEncoder into an integer format for model training readiness.

id	text	label	sentiment	language	text_cleaned	sentiment_label
9536	Cooking microwa	2	positive	en	cooking microwave pizzas yummy	2
6135	Any plans of allov	1	neutral	en	any plans of allowing sub tasks to sh	1
17697	I love the humor,	2	positive	en	i love the humor i just reworded it. lik	2
14182	naw idk what ur t	1	neutral	en	naw idk what ur talkin about	1
17840	That sucks to hea	0	negative	en	that sucks to hear. i hate days like th	0

Figure 7. Cleaned text samples for Sentiment Analysis Dataset

```
2. Preprocessing data...
Language distribution:
language
en         28538
unknown      500
af           306
so           289
cy           218
no           157
tl           122
nl           118
fr           109
ca            93
Name: count, dtype: int64
English samples: 28538 / 31232
Language distribution:
language
en         4758
unknown      89
af           58
so           43
cy           33
tl           30
no           26
nl           22
da           15
ca           14
Name: count, dtype: int64
English samples: 4758 / 5205
Language distribution:
language
en         4780
unknown      78
so           53
af           42
no           36
cy           27
tl           20
fr           20
nl           19
da           18
Name: count, dtype: int64
English samples: 4780 / 5206
Sentiment label mapping: {'negative': np.int64(0), 'neutral': np.int64(1), 'positive': np.int64(2)}
```

Figure 8. Language distribution before and after English filtering across train, validation, and test sets

1.2.3. Balanced Dataset

During data exploration, the dataset had an imbalance issue because the neutral sentiment class appeared sparse. To resolve this, I used a combination of text augmentation techniques through the nlpaug library. Specifically, the process consisted of alternating between WordNet synonym substitution and DistilBERT contextual word insertion, together with character-level spelling perturbation. These techniques were applied only to the underrepresented classes to synthetically expand their sample size. As a result, the dataset distribution between sentiment categories became equal, so the model gained exceptional skills in classifying neutral sentiment.

1.3. Data Splitting

A proper evaluation requires both datasets to be separated into training, validation, and test subsets through the 'train_test_split' method with 'random_state=42' parameter (Figure 9-10).

```
Training samples: 21497, Test samples: 5375
```

Figure 9.Intent Classification Dataset Splits

```
Training Samples: 25267, Test Samples: 6317
```

Figure 10.Sentiment Analysis Dataset Splits

1.4. Tokenization

Since both models were built using BERT, the text data needed to be converted into tokenized numerical sequences before training. The pre-trained tokenizer of BERT (BertTokenizer.from_pretrained("bert-base-uncased")) formatted input text into an understandable format for the model. The method required several significant procedure steps. The text got processed into subword tokens which BERT could understand while maintaining contextual meaning. Sequences longer than 128 tokens were truncated through this process to avoid both excessive input size and interruptions in training because of lengthy sentences. The input lengths for shorter sentences were normalized through padding to match the length of all sentences contained in the dataset. The generation of attention masks enabled BERT to recognize between genuine words and padding elements when it processed the data. The restructuring process turned unprocessed text into BERT-compatible embeddings to maximize efficiency during both intent classification and sentiment analysis operations (Tahmid et al., 2024).

1.5. Creating a PyTorch Dataset

To efficiently handle tokenized inputs during training, the data was converted into PyTorch datasets. The project contained two independent dataset types called IntentDataset for detecting intents and SentimentDataset for evaluating sentiment. The data storage contained both input IDs and attention masks as tokenized representations to assist the training process of the model. The model received identification labels that matched the tokenized sequences with their defined intent or sentiment classification categories. Through integration into PyTorch's DataLoader the datasets gained access to features such as mini batching alongside data shuffling and GPU acceleration benefits. The model gained the ability to process multiple samples consecutively thus both minimizing computational expenses and accelerating the training duration while preserving its precision.

2. AI Model Development

2.1. Intent Classification Model Architecture

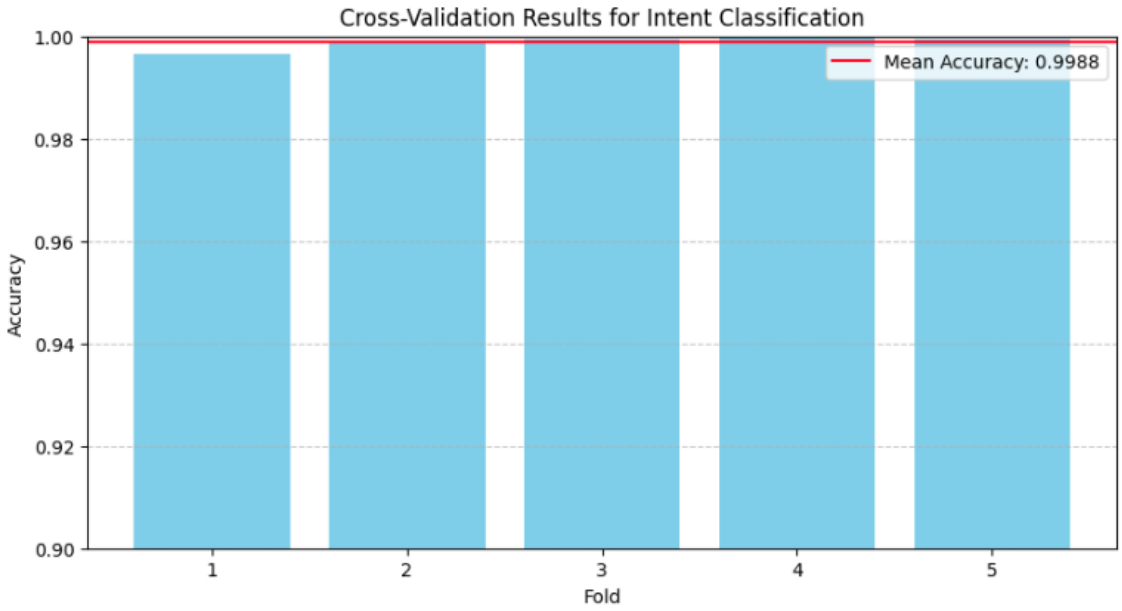
BERT supports the basis of the model structure while the custom classifier layer selects among 27 potential intent results (Mohammadi and Chapon, 2020). The

self-attention mechanism of the model enables its ability to understand how context affects customer queries.

2.1.1. Cross-Validation Strategy

I implemented 5-fold stratified cross-validation as an approach to validate model robustness and prevent overfitting from occurring when using one dataset split (Figure 11). The training dataset split contained five equivalent parts, maintaining exact intent label distributions between each segment.

During each iteration, I trained the model on four folds and validated it on the remaining one. By distributing the data into five folds, I ensured steady performance evaluation across each data subset. I monitored accuracy scores per fold before taking the mean value while calculating the standard deviation to understand the stability level. Through the process, I gained both increased generalization ability and confidence that my model would process different kinds of user queries.



	Fold	Accuracy
0	1	0.996512
1	2	0.999070
2	3	0.999302
3	4	0.999767
4	5	0.999535
5	Mean	0.998837
6	Std Dev	0.001186

Figure 11. Cross-validation accuracy across 5 folds

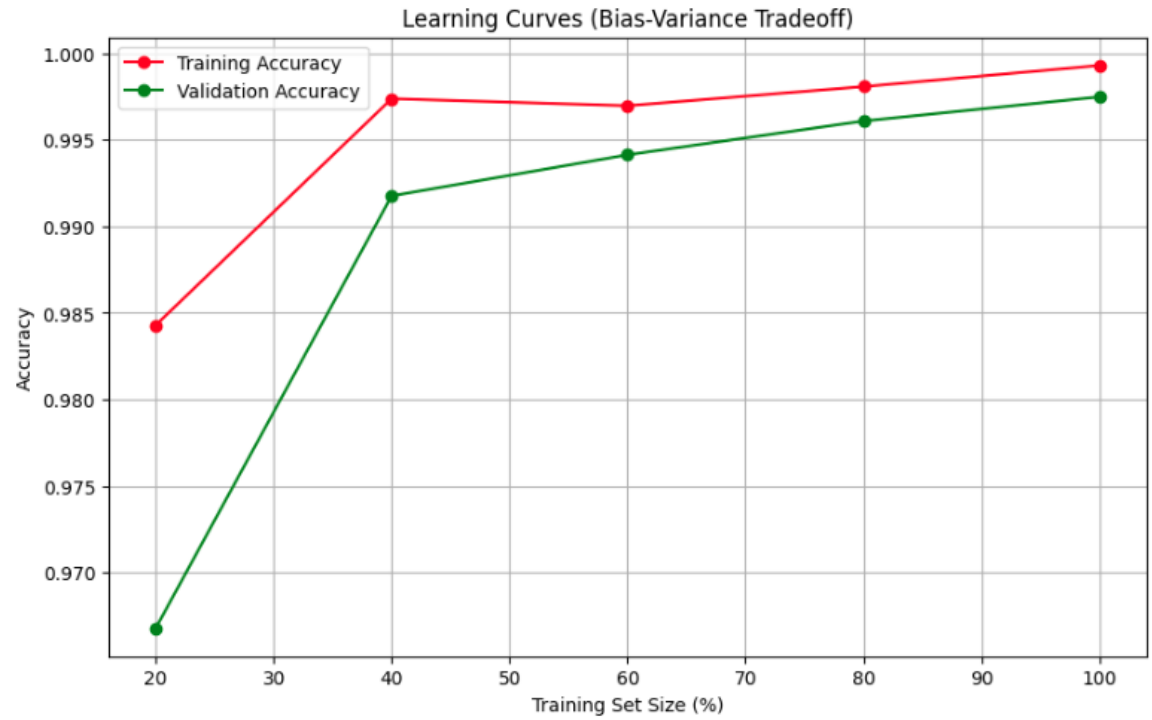
2.1.2. Learning Curve Analysis & Bias–Variance Tradeoff

The evaluation of model performance due to data quantity necessitated a learning curve analysis. The model training consisted of working with decreasing and increasing ranges of data from 20% to 100%. My measurements included training accuracy and validation accuracy at every validation stage.

The results helped me evaluate the bias–variance tradeoff (Figure 12). The model maintained acceptable performance because the training accuracy scores did not differ much from the validation accuracy scores. The model's performance

stability increased as the amount of data increased because the detected variance decreased. This made the model insensitive to changes in its input samples.

```
Train Accuracy: 0.9993, Validation Accuracy: 0.9975
```




```
Bias-Variance Analysis:
   Training Size (%)  Training Accuracy  Validation Accuracy  \
0               20.0           0.984299             0.966788
1               40.0           0.997383             0.991767
2               60.0           0.996976             0.994139
3               80.0           0.998081             0.996093
4              100.0           0.999302             0.997488

   Variance (Train-Val Gap)  Bias Approximation
0                  0.017511            0.015701
1                  0.005616            0.002617
2                  0.002837            0.003024
3                  0.001988            0.001919
4                  0.001814            0.000698
```

Figure 12. Training vs. Validation Accuracy

2.1.3. Training Configuration for Intent Model

Training was performed with the following configs:

- 3 epochs are enough to fine-tune BERT to avoid overfitting.
- Batch size 8 will be suitable for memory-constrained GPUs.
- AdamW optimizer.
- Weight reduction of 0.01 avoids overfitting by reducing the large number.
- Learning rate initialization (500 steps) helps to stabilize the initial training.

2.2. Sentiment Analysis Model Architecture

Sentiment analysis required DistilBERT modification into a model which distinguished between three sentiment categories (Mohammadi and Chapon, 2020). We added a dropout layer with a rate of 0.1 to prevent overfitting and applied weight decay of 0.01 because of different sentiment patterns.

2.2.1. Training Configuration for Sentiment Model

I trained DistilBERT for five epochs with a batch size of four and a learning rate of 2e-5. The training process depended on weight decay set to 0.01, combined with gradient checkpointing to prevent overfitting. To deal with label imbalance, I implemented customized weights for loss calculations. The model architecture was configured with a dropout rate of 0.1, and I saved the trained model and tokenizer for future deployment and testing.

2.3. Challenges in Model Training

The biggest challenge for me in this training process is that it takes too much time because the dataset used for both models is extensive, requiring a lot of time in the process of analyzing and preprocessing data. In addition, I trained entirely on Google Colab, where my friends recommended I use it with integrated training using T4-GPU to speed up but there is a time limit for use during the day. A model is trained in many epochs like that (Figure 13-15); if I do not use the above method, it will take even more time. This further hinders me in the process of debugging and deploying the model later.

Training on fold 1/5...

[4300/4300 06:15, Epoch 2/2]

Epoch	Training Loss	Validation Loss
1	0.025900	0.027450
2	0.007100	0.017172

Fold 1 Accuracy: 0.9965

Training on fold 2/5...

[4300/4300 06:10, Epoch 2/2]

Epoch	Training Loss	Validation Loss
1	0.030500	0.005967
2	0.004800	0.003447

Fold 2 Accuracy: 0.9991

Training on fold 3/5...

[4300/4300 06:15, Epoch 2/2]

Epoch	Training Loss	Validation Loss
1	0.015900	0.007998
2	0.002700	0.006443

Fold 3 Accuracy: 0.9993

Training on fold 4/5...

[4300/4300 06:20, Epoch 2/2]

Epoch	Training Loss	Validation Loss
1	0.014500	0.012577
2	0.000200	0.001514

Fold 4 Accuracy: 0.9998

Training on fold 5/5...

[4300/4300 06:11, Epoch 2/2]

Epoch	Training Loss	Validation Loss
1	0.006300	0.005084
2	0.006600	0.003239

Fold 5 Accuracy: 0.9995

Cross-Validation Results:
Fold accuracies: [0.9965116279069768, 0.9990697674418605, 0.9993021632937893, 0.9997673877645964, 0.9995347755291928]
Mean accuracy: 0.9988
Standard deviation: 0.0012

Figure 13. 5-Fold Cross-Validation (Intent Model)

Training final model on entire training set...

[8064/8064 11:24, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	0.008800	0.038959
2	0.005400	0.032174
3	0.000000	0.022048

Model completely trained!
['intent_model/category_encoder.pkl']

Figure 14. Final Training on Entire Dataset (Intent)

[7895/7895 22:08, Epoch 4/5]

Step	Training Loss
100	1.098600
200	1.084700
300	0.930100
400	0.753900
500	0.756400
600	0.701400
700	0.711500
800	0.669200
900	0.632400
1000	0.648600
6400	0.202000
6500	0.195400
6600	0.195400
6700	0.210500
6800	0.176600
6900	0.222900
7000	0.214000
7100	0.194900
7200	0.204600
7300	0.213100
7400	0.226500
7500	0.202300
7600	0.195800
7700	0.203500
7800	0.187800

```
Model completely Trained!
Evaluate the model on the test set...
Test Accuracy: 0.7578
```

Figure 15. Full Stepwise Loss + Accuracy (Sentiment – Final Epochs)

Chapter Four: IMPLEMENTATION

1. Frontend Implementation

To develop the frontend, I integrated React with Ant Design for building the user interface. The application exists as an SPA model that uses React Router along with a component-based architecture, which splits the application into reusable components and complete containers.

The authentication system was built by me based on AuthContext and JWT stored in localStorage. I customized Axios to display loading and handle errors through notifications automatically.

The main feature of the application is the ChatComponent, which allows users to interact with the AI chatbot and view information about emotions and intentions from each message. The DashboardComponent incorporates Recharts for visual chart presentation while showing statistics about chat history.

The user experience received my attention through proper notifications and animations, which led to the development of a current and simple-to-use web application interface.

2. Backend Implementation

In this project, the backend system consisted of two main components which included Node.js/Express for user management and authentication together with Flask/Python for AI processing. The MySQL database consists of three tables: user, otps, and chats, which are managed through Sequelize ORM.

Api access comes from the combination of JWT alongside bcrypt through the auth.js middleware implementation. Two-factor authentication is implemented via an OTP system that sends emails using Nodemailer; each code has a 10-minute expiration time and is securely encrypted.

Multer handles file uploads with a custom disk storage configuration, which includes random naming rules and checks for file size and type for security.

The AI part uses a BERT model through Flask with the functions get_chatbot_reply() and get_sentiment() to analyze the message content. Communication between Node.js and Flask is done via HTTP requests with axios in the processChatMessage() function.

The statistics feature in getUserChatStats() helps analyze user sentiment and intent trends through the developed statistics functionality. The entire system is enhanced with security best practices, from error handling, injection prevention, to environment configuration management with dotenv.

3. System Design

3.1. System Architecture

The system design implements the MVC pattern (Figure 16) for distinct classification between data, interface and processing logic operations. The user interface functions as the View through the React front-end application. Node.js functions as the Controller through its processing of client requests to handle and direct data types. The business logic together with data management falls under the responsibility of both Sequelize ORM and MySQL as the Model. Additionally, Flask AI server acts as a separate microservice that handles natural language parsing.

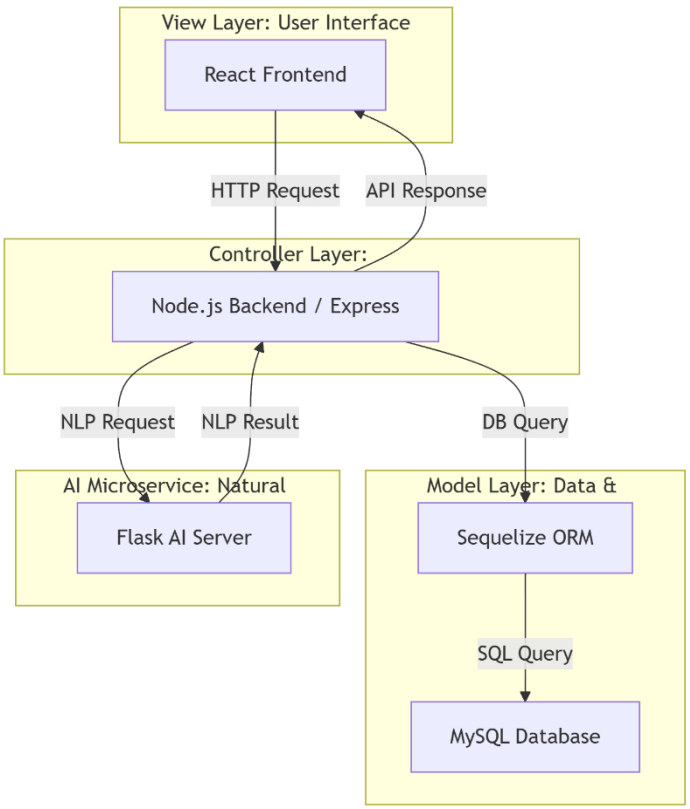


Figure 16. System Architecture Diagram

3.2. Use Case Diagram

The diagram (Figure 17) shows the interaction system between program users and the chatbot operating framework. There are two main actors: User (registered user) and Guest (unregistered user). Both can interact with the chatbot, but only the User can view chat statistics and manage accounts.

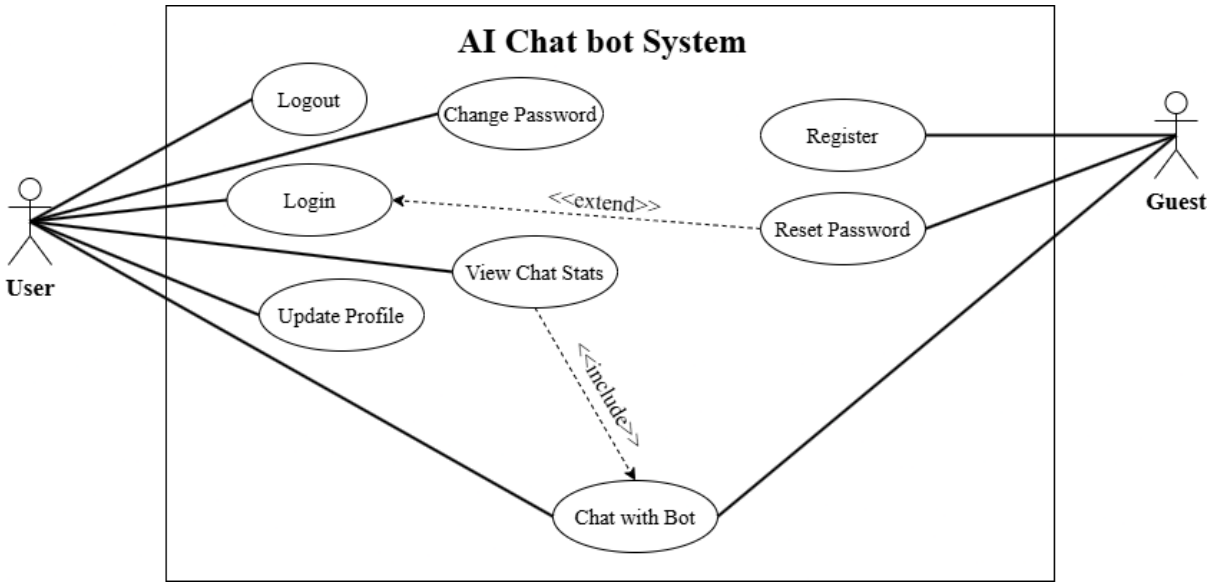


Figure 17. Use Case Diagram

3.3. Physical Database Diagram (Relational Database Diagram)

Three main database tables (Figure 18), consisting of users for storing user data exist alongside otps for storing one-time authentication codes and chats for saving chat interactions, are displayed in the diagram. The 1-n relationship between users and OTPs, users and chats ensures data integrity.

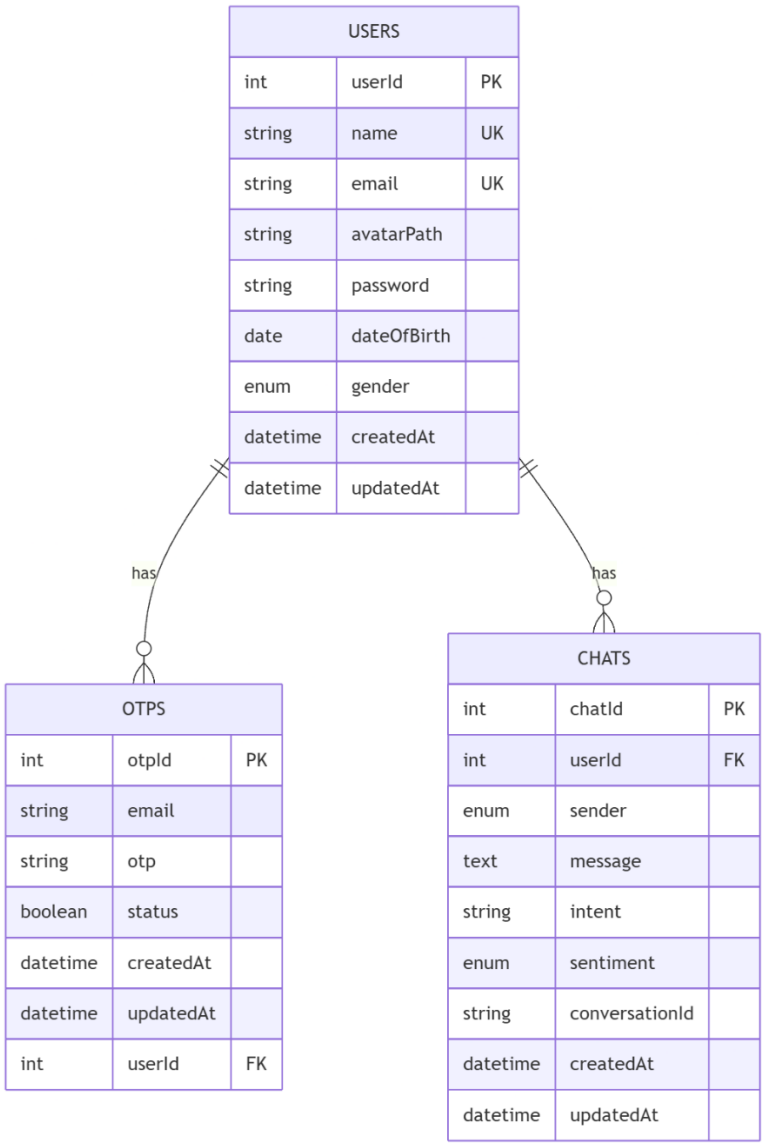


Figure 18. Relational Datebase Diagram

3.4. Class Diagram

The class diagram (Figure 19) shows the structure of the main objects in the system with three core classes. The User class manages user account information with

properties such as userId, name, email and methods such as create(), authenticate() and resetPassword(). The OTP class handles authentication codes with properties otpId, email, otp and methods generate(), verify() and expire(). The Chat class stores messages with properties chatId, message, intent, sentiment and methods saveMessage(), getHistory() and analyzeIntent(). The composition relationships are represented by the solid diamond at the beginning of "1" and "0..*" at the beginning of "many", indicating that a User can have zero or many OTPs and Chats.

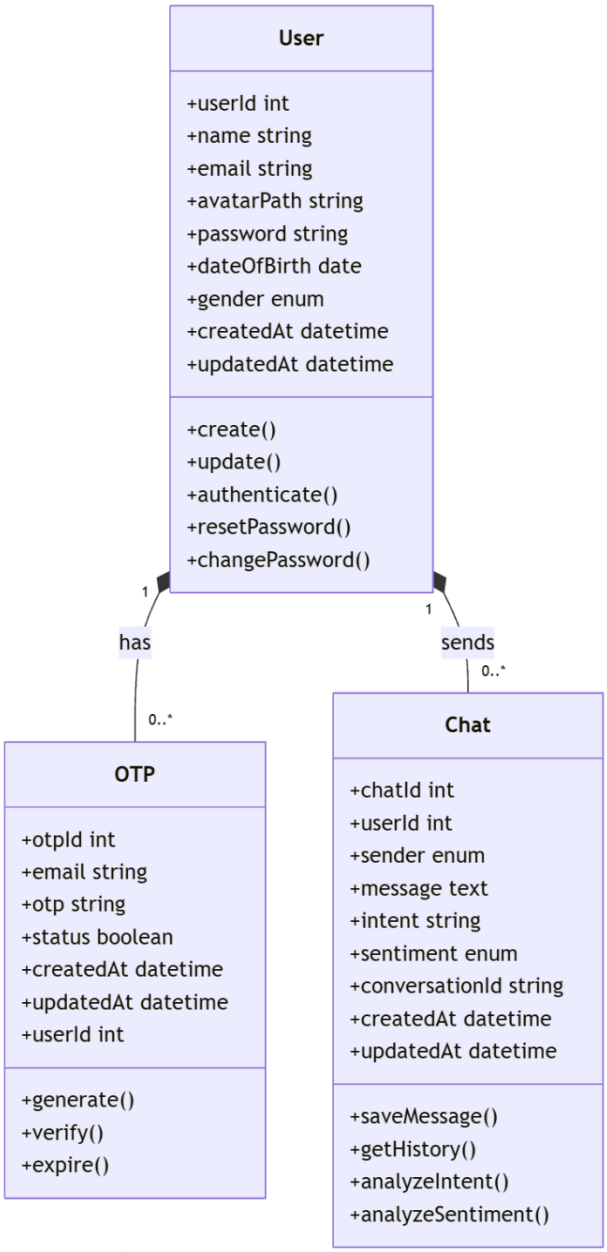


Figure 19. Class Diagram

Chapter Five: RESULT AND DISCUSSION

1. Intent Classification Model Results

1.1. Evaluation Metrics

The intent classification model evaluation utilized multiple performance metrics with accuracy, precision, recall, and F1-score among them. The model demonstrated outstanding performance at intent class distinction because of the dataset organization. The model displayed high efficiency for predicting customer intents by producing minimal error rates during its operation. The model achieved balanced performance results demonstrated by high F1-scores which indicated successfully predicted intents at a minimal false detection rate across all classes (Yacouby and Axman, 2020).

1.2. Robustness to Noisy Data

My model needed testing for its ability to understand noisy text since human typers frequently make mistakes. I tested various noise simulation levels between 10% and 40% on the test queries through the introduction of typo and character insertion and omission errors (Figure 20).

The model demonstrated excellent performance even under noisy conditions especially when the distortion level remained low. The evaluation demonstrated that the chatbot system managed to understand user goals successfully with imperfect input, which increased my confidence. Performance results included accuracy scores as well as macro F1-scores, which I observed degrading gradually rather than suffering an abrupt decline.

```
Testing with noise level: 0.0
Accuracy with noise level 0.0: 0.9980
F1 Score (macro): 0.9980

Testing with noise level: 0.1

Examples of original vs. noisy texts:
Original: need assistance to report problems with online payments
Noisy: need asistance hto report problems with onlnie ypaments

Original: I'm trying to see in what cases can i ask for my money back
Noisy: Im' tying to ese in what caes can i ask for my money bak

Original: i need assistance to check the allowed payment options
Noisy: ineed assistnce to check the allowd paymnt options

Accuracy with noise level 0.1: 0.8861
F1 Score (macro): 0.8873
```

```
Testing with noise level: 0.4

Examples of original vs. noisy texts:
Original: need assistance to report problems with online payments
Noisy: ee asistsxance toy erppot yprobeutkmsiith online paymetns

Original: I'm trying to see in what cases can i ask for my money back
Noisy: I'mr ysnig toyg se ni whadt ecyases cavn s sapk for m mony hbadc

Original: i need assistance to check the allowed payment options
Noisy: o ziened assistandct qcheck the awed pyament otions

Accuracy with noise level 0.4: 0.3287
F1 Score (macro): 0.3406
```

Figure 20. Accuracy under different noise levels

1.3. Outlier Detection and Handling

I ran multiple outlier tests on my model by feeding it with short or lengthy queries containing abnormal punctuation patterns and uncommon intent classifications. I identified these samples using distribution thresholds and special character ratios.

The model presented good overall performance but showed reduced accuracy in dealing with these unusual situations (Figure 21). Understanding the situation required me to inspect the confusion matrix for outliers and review multiple misclassified examples. The evaluation showed where the data might need expansion and how intent classification limits should be defined.

```
Analyzing model performance on outlier/edge cases...
Identified 626 potential outliers out of 5375 test samples

Overall test accuracy: 0.3287441860465116
Outlier test accuracy: 1.0000
Performance gap on outliers: -0.6713

Length outliers accuracy: 1.0000
```

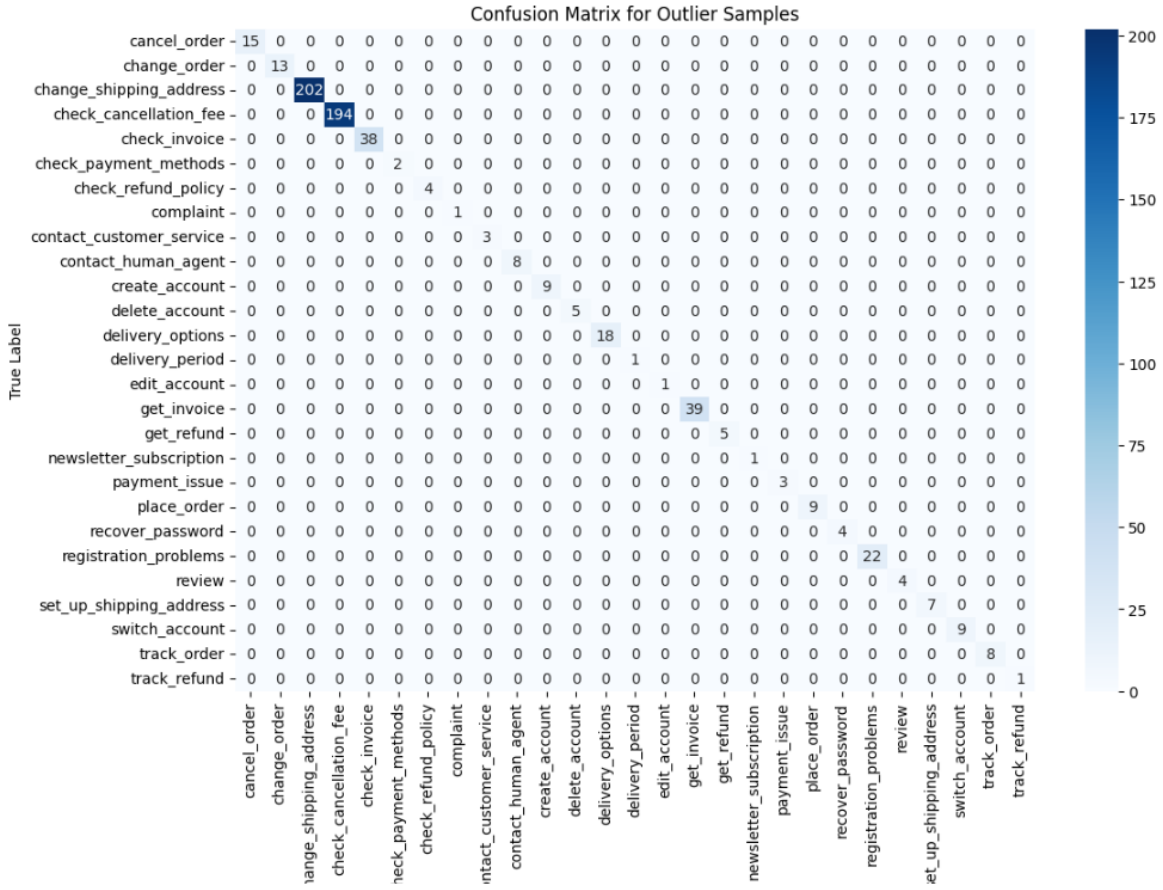


Figure 21. Confusion Matrix for Outlier Inputs

1.4. Results and Analytics

During testing the intent classification system generated outstanding results showing few classification mistakes among intents (Figure 22). During evaluation, few misclassifications occurred, mainly in intents that were closely related (e.g., “Track Order” vs. “Cancel Order”). Even though some misclassifications arose only they happened infrequently, so the model delivered accurate intent recognition as its main pattern.

```
Test Accuracy: 0.9979534883720931
                          precision    recall  f1-score   support

            cancel_order       1.00      1.00      1.00       200
            change_order       1.00      1.00      1.00       199
 change_shipping_address       0.99      1.00      1.00       195
  check_cancellation_fee       1.00      1.00      1.00       190
           check_invoice       0.99      0.99      0.99       200
    check_payment_methods       1.00      1.00      1.00       200
      check_refund_policy       0.99      0.99      0.99       199
                complaint       1.00      1.00      1.00       200
contact_customer_service       1.00      1.00      1.00       200
     contact_human_agent       1.00      1.00      1.00       200
          create_account       0.99      1.00      1.00       199
          delete_account       0.99      1.00      1.00       199
        delivery_options       0.99      1.00      1.00       199
         delivery_period       0.99      0.99      0.99       200
            edit_account       1.00      1.00      1.00       200
             get_invoice       0.99      0.99      0.99       200
              get_refund       0.99      0.99      0.99       199
 newsletter_subscription       1.00      1.00      1.00       200
           payment_issue       1.00      0.99      1.00       200
             place_order       1.00      1.00      1.00       200
       recover_password       1.00      0.99      1.00       199
  registration_problems       1.00      0.99      1.00       200
                  review       1.00      1.00      1.00       199
 set_up_shipping_address       1.00      0.99      1.00       199
          switch_account       1.00      1.00      1.00       200
             track_order       1.00      0.99      1.00       199
            track_refund       1.00      1.00      1.00       200

                accuracy                           1.00      5375
               macro avg       1.00      1.00      1.00      5375
            weighted avg       1.00      1.00      1.00      5375
```

Figure 22. Classification report for Intent Classification Model

2. Sentiment Analysis Model Results

2.1. Evaluation Metrics

After training, I tested the model on a final held-out test set that was not used during training or validation. I used a custom evaluation function that runs entirely on CPU to avoid GPU memory issues and computes predictions in batches. Once testing was complete, I exported the model, tokenizer, and label encoder as .pkl files for deployment. I also tested the trained model on real user-like inputs to assess how it would perform in practical chatbot scenarios, recording both the predicted sentiment and the confidence score.

The sentiment classification model evaluation utilized multiple performance metrics with accuracy, precision, recall, and F1-score among them. The model demonstrated outstanding performance at intent class distinction because of the dataset organization. The model displayed high efficiency for predicting customer

intents by producing minimal error rates during its operation. The model achieved balanced performance results demonstrated by high F1-scores which indicated successfully predicted intents at a minimal false detection rate across all classes. The negative class had high recall, indicating that most negative sentiments were identified, but some false positives occurred (Yacouby and Axman, 2020).

2.2. Results and Analytics

The sentiment analysis model demonstrated an acceptable 75.8% accuracy rate for detecting sentiment because of its demanding task. The model demonstrated strong classification abilities between positive and negative sentiments although it failed to accurately recognize neutral sentiment because of linguistic pattern similarities (Figure 23). The model struggles to detect neutral sentiment since most neutral expressions do not provide distinct sentiment indicators (Raghunathan and Saravanakumar, 2023). The dataset included sentences marked as neutral which contained words of minor positive or negative sentiment thus confusing matrix classifications evaluation (Figure 24).

```
Model evaluation in progress...
Evaluating model on test set...
Test Accuracy: 0.7578

Classification Report:
              precision    recall  f1-score   support

    negative       0.78      0.79      0.78      1009
     neutral       0.68      0.71      0.69      1106
    positive       0.83      0.78      0.81      1044

    accuracy                           0.76      3159
   macro avg       0.76      0.76      0.76      3159
weighted avg       0.76      0.76      0.76      3159
```

Figure 23.Classification report for Sentiment Analytics Model

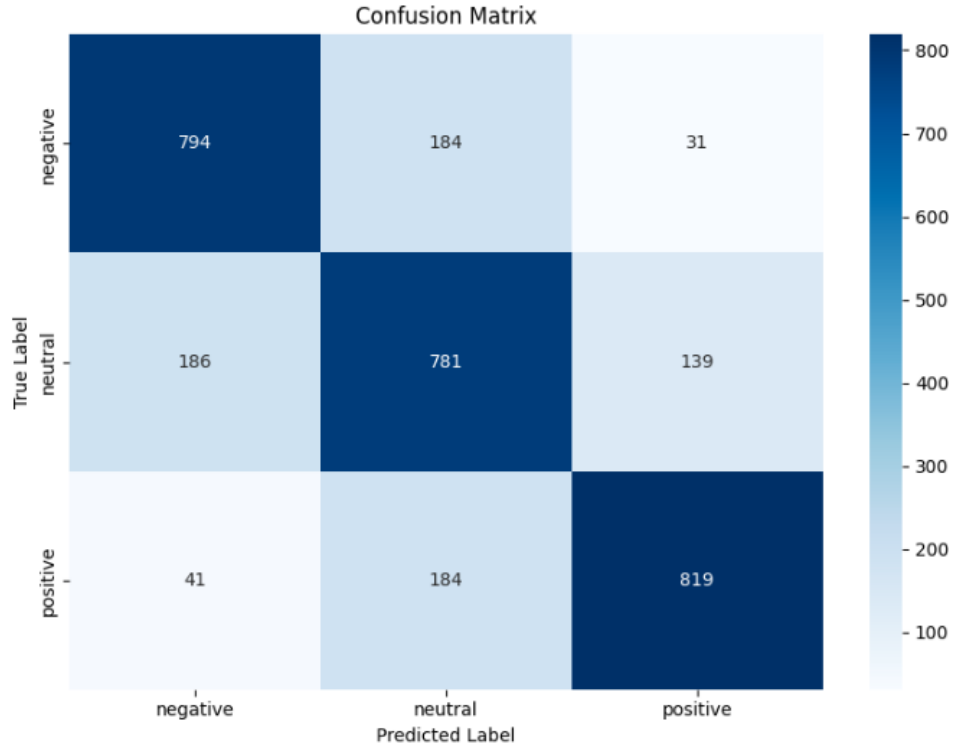


Figure 24. Confusion Matrix Evaluation

3. Discussion

The intent classification model delivered superior accuracy together with better overall consistency than the sentiment analysis model (Table 3). The structured nature of intent classification explains why its accuracy exceeded sentiment analysis since intent requires objective interpretation (Zhou, 2022).

Feature	Intent Classification	Sentiment Analysis
Accuracy	99.7%	75.8%
F1-score Range	0.99 - 1.00	0.69 - 0.81
Task Complexity	Lower (structured queries)	Higher (subjective expressions)
Training Epochs	3	4/5
Weight Decay	0.01	0.01
Dropout Rate	0.1	0.1
Performance Limiting Factor	Similar intent phrasing	Ambiguity in neutral sentiment

Table 3.Comparison Between Intent Classification and Sentiment Analysis Models

Chapter Six: CONCLUSION AND FUTURE WORK

1. Conclusion

The study presented information about building an AI-powered chatbot which performed intent classification and sentiment analysis tasks. The entire process of training models for the project was done entirely on Google Colab. The development of the chatbot used BERT-based models with training data obtained from Bitext Customer Support Chatbot Dataset to classify intents and Multiclass Sentiment Analysis Dataset to detect sentiments. The performance required execution of broad data processing and tokenization alongside model refinement procedures. The application achieved impressive accuracy of 99.7% with its intent classification model but struggled with sentiment analysis precision that reached 75.8% although it was effective. By unifying the models, the chatbot successfully managed to interpret user requests precisely and generate suitable responses tied to emotional messages leading to potential success as an automated customer service system.

2. Future work

In future development stages, I plan to connect visual data with audio signals, which will help me measure tone as well as facial expression and vocal expression for understanding sentiment. Such an approach provides expanded insight into the complete user emotional state, especially when dealing with multimedia communication. Additionally, my objective includes investigating RoBERTa and DeBERTa as transformer architectures beyond BERT because they will enhance neutral sentiment detection through enhanced contextual comprehension. The adaptation of evolving customer communication language trends can be achieved through domain-specific training and continual learning approaches applied to the chatbot system. Finally, the system will establish a feedback system to enable performance enhancement of the chatbot through user-end prediction corrections.

REFERENCES

Chai, C.P., 2023. Comparison of text preprocessing methods. *Natural Language Engineering*, 29(3), pp.509–553. https://doi.org/10.1017/S1351324922000213.

Chen, X., Zhang, W., Pan, S. and Chen, J., 2023. Solving Data Imbalance in Text Classification With Constructing Contrastive Samples. *IEEE Access*, 11, pp.90554–90562. https://doi.org/10.1109/ACCESS.2023.3306805.

Dr. R. Ramki, Dr. V. Gopi, Mrs Vidya. R, Dr. Rohit Markan, Dr Sundarapandiyan Natarajan and M.Rajalakshmi, 2024. AI-Powered Chatbots in Customer Service: Impact on Brand Loyalty and Conversion Rates. *Economic Sciences*, [online] 20(2), pp.190–203. https://doi.org/10.69889/vs5gtv52.

Han, W., Pang, B. and Wu, Y.N., 2021. Robust Transfer Learning with Pretrained Language Models through Adapters. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics. pp.854–861. https://doi.org/10.18653/v1/2021.acl-short.108.

Kora, R. and Mohammed, A., 2023. A Comprehensive Review on Transformers Models For Text Classification. In: *2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. IEEE. pp.1–7. https://doi.org/10.1109/MIUCC58832.2023.10278387.

Mohammadi, S. and Chapon, M., 2020. Investigating the Performance of Fine-tuned Text Classification Models Based-on Bert. In: *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE. pp.1252–1257. https://doi.org/10.1109/HPCC-SmartCity-DSS50907.2020.00162.

Raghunathan, N. and Saravanakumar, K., 2023. Challenges and Issues in Sentiment Analysis: A Comprehensive Survey. *IEEE Access*, 11, pp.69626–69642. https://doi.org/10.1109/ACCESS.2023.3293041.

Rahevar, M. and Darji, S., 2024. THE ADOPTION OF AI-DRIVEN CHATBOTS INTO A RECOMMENDATION FOR E-COMMERCE SYSTEMS TO TARGETED CUSTOMER IN THE SELECTION OF PRODUCT. *International Journal of Management, Economics and Commerce*, 1(2), pp.128–137. https://doi.org/10.62737/m1vpdq75.

Shorten, C., Khoshgoftaar, T.M. and Furht, B., 2021. Text Data Augmentation for Deep Learning. *Journal of Big Data*, 8(1), p.101. https://doi.org/10.1186/s40537-021-00492-0.

Tahmid, M.T., Shahgir, H.S., Mahbub, S., Dong, Y. and Bayzid, Md.S., 2024. *BiRNA-BERT allows efficient RNA language modeling with adaptive tokenization*. https://doi.org/10.1101/2024.07.02.601703.

Taufikin, Orosoo, M., Rengarajan, M., Fatma, G., Yuldashev, D. and Raj, I.I., 2024. Applying Transformer-Based Neural Networks to Corpus Linguistic Data for Predictive Text

Generation in Multilingual Environments. In: *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE. pp.1–6. https://doi.org/10.1109/ESCI59607.2024.10497447.

Yacouby, R. and Axman, D., 2020. Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models. In: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. Stroudsburg, PA, USA: Association for Computational Linguistics. pp.79–91. https://doi.org/10.18653/v1/2020.eval4nlp-1.9.

Zhou, L., 2022. Research on Quantitative Model of Brand Recognition Based on Sentiment Analysis of Big Data. *Frontiers in Psychology*, 13. https://doi.org/10.3389/fpsyg.2022.915443.