National College of Ireland

Building a Hindi-English customer support chatbot using pre-trained language models

MSc Research Project
MSc in Artificial Intelligence

Mehwish Mohammed Hanif Khatib
Student ID: X23398396

School of Computing
National College of Ireland

Supervisor: Anderson Simiscuka

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mehwish Mohammed Hanif Khatib		
Student ID:	X23398396		
Programme:	MSc in Artificial Intelligence	**Year:**	2024 – 2025
Module:	MSc Research Project		
Supervisor:	Anderson Simiscuka		
Submission Due Date:	15th September 2025		
Project Title:	Building a Hindi-English customer support chatbot using pre-trained language models		
Word Count:	5275 words	**Page Count:**	17 pages

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Mehwish Mohammed Hanif Khatib

Date: 13th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Building a Hindi-English customer support chatbot using pre-trained language models

Mehwish Mohammed Hanif Khatib
X23398396

Abstract

Multilingual support is a significant challenge in linguistically diverse countries like India, where users may switch languages during conversations. This project presents a Hindi-English customer support chatbot designed to overcome language barriers using modern NLP, deep learning, and transformer-based models. Leveraging over 1,002,235 bilingual query-response pairs, the system employs BiLSTM for intent classification, BERT for response generation, and mBERT for multilingual multitask learning, optimized with TensorFlow and Keras. The chatbot, built with Tkinter, offers a user-friendly GUI allowing language choice and delivers high accuracy and fast responses, effectively handling code-switched inputs. Its performance metrics indicate readiness for real-world customer service deployment in Indian markets. Future enhancements include adding regional dialects, voice interfaces, and domain-specific fine-tuning, further improving bilingual user experience and contextual response accuracy.

1 Introduction

Chatbots are one example of intelligent solutions being investigated due to the increasing demand for multilingual customer support in India. The given dissertation is devoted to the creation of a Hindi-English customer support chatbot based on pre-trained models. The purpose is to close the communication barrier between companies and customers, allowing interaction to take place bilingually without any issues. The use of Natural Language Processing (NLP) techniques will enable the chatbot to comprehend and correctly answer in both Hindi and English. Code-switching, intention recognition, and contextual understanding are some of the main issues that this project will solve so as to provide a pleasant and effective support experience on either side of the linguistic divide.

Aim

The main aim of this project is to develop a bilingual customer support chatbot using pre-trained language models that can understand and respond effectively in both Hindi and English.

Objectives

- To design and implement a chatbot framework capable of handling bilingual (Hindi-English) user interactions.
- To integrate pre-trained multilingual language models for accurate intent detection and response generation.
- To evaluate the chatbot's performance using standard NLP metrics such as accuracy, F1-score, and response relevance.
- To enhance user experience by enabling context-aware and grammatically coherent bilingual responses in customer support scenarios.

Research Background

The linguistic diversity in India poses a serious challenge in the process of offering efficient customer support to the various regions in India. Together with Hindi and English being among the most frequently spoken languages, there is a growing momentum to have a system capable of handling the two languages without any difficulty. Old-fashioned rule-based chatbots do not easily handle mixed-language questions, particularly when there is a code-switching or informal language situation. The breakthroughs in Natural Language Processing (NLP), specifically the creation of pre-trained text language models such as BERT, mBERT, and BiLSTM, have changed the manner in which computers comprehend and generate human language. They are trained using large multilingual corpora and are therefore able to capture meanings in context and are therefore appropriate in bilingual contexts. This study is based on these developments and proposes the development of a chatbot that will be able to comprehend and reply to both Hindi and English. Through these models, the study will enhance customer interaction and minimise the language barrier, and in the end, lead to more friendly and accessible digital support services.

Problem statement

In countries where several languages are spoken, such as India, customer support systems do not manage to deal effectively with both Hindi and English, especially where a user alternates between them during a conversation. The conventional chatbots are not linguistically adaptable and context-sensitive to process Hindi and English language together. This causes bad user experience, misunderstanding, and low customer satisfaction. This is due to the lack of smart bilingual systems that will be able to facilitate proper service delivery. Thus, there is a requirement to create a strong, AI-based chatbot that will be able to interpret and reply to the bilingual entry aptly. This study deals with the issue by employing pre-trained mBERT models that allow unproblematic communication in Hindi and English.

Rationale

The development of a bilingual Hindi-English customer support chatbot addresses a significant gap in the current customer service landscape in India. The increasing prevalence of digital services in both urban and rural areas has led to a tendency for users to switch between Hindi and English, even within a single sentence. The project is predicated on the idea that intelligent automation is essential to improving efficacy and accessibility. This is illustrated by the chatbot's use of pre-trained mBERT models to learn the semantics underlying the context of a bilingual conversation. By doing this, the user will be more satisfied because the intent understanding and response will be as accurate as possible. This can be explained by the need to match the current state of well-developed NLP technologies and the needs of linguistic realities, thus eventually making the creation of customer service solutions all-embracing and scalable, and all that with a view towards the multilingual environment of Indian users.

To summarise, the current dissertation aims at creating a bilingual Hindi-English customer support chatbot constructed out of pre-trained mBERT models as a way of overcoming communication issues in a multilingual country such as India. It draws attention to the increasing demand for intelligent systems capable of handling multilingual interactions, particularly in customer service. By incorporating Natural Language Processing (NLP) techniques for precise intent detection, contextual understanding, and efficient response generation, the study seeks to improve user experience. While pre-trained models such as BERT and mBERT provide better comprehension, traditional chatbots have trouble with code-switching and informal queries. In the end, the project aims to increase accessibility in digital customer support and close language gaps.

2 Related Work

The rapid growth of multilingual populations and digital services in India has highlighted the urgent need for intelligent customer support systems capable of handling more than one language. Specifically, the Hindi-English pattern of communication is also widespread among users, and traditional chatbots face serious issues with this. This literature survey investigates what studies have already been done in the field of multilingual Natural Language Processing (NLP), pre-trained language models, and chatbots. However, the emphasis is placed on how these technologies have helped to overcome language barriers, how they have been employed in managing the process of code-switching, and how they have aided in enhancing user intercourse. The chapter establishes the grounds upon which a viable bilingual chatbot that best suits the customer support space in India can be created.

2.1 Pre-trained Language Models for Multilingual NLP

Pre-trained language models have significantly advanced multilingual Natural Language Processing (NLP), especially for low-resource languages like Hindi. Kumar and Albuquerque (2021) demonstrated the potential of XLM-R, a multilingual model displaying transformers, which is used to sense of sentiment of Indian languages using zero-shot learning. XLM-R has the capability of generalising to various languages without undergoing any language-specific training, hence applicable to take care of tasks such as bilingual chatbots. Hossain and Goyal (2024) analyse how text generation relying on transformer-based architectures like BERT, mBERT, and GPT has become increasingly popular. They pointed out the way these models treat syntactic variations and semantic ambiguities to enhance cross-lingual activities such as translation, intent detection, and dialogue generation. With these improvements, models that are pre-trained can decode contextual information on both high and low resources.

A study by Pakray et al. (2025) touched on the low-resource language applications of NLP and the significance of transfer learning and pre-trained models to limit the reliance on large, annotated data. They observed that such models have the potential to enhance the effectiveness of NLPs in languages with high morphology, as well as in code-switched languages, such as Hindi-English hybrid.

2.2 Code-Switching Challenges in Indian Languages

India is a multilingual society, where code-switching, that is, a practice of alternating between two or more languages within a given conversation, takes place. This presents a huge challenge to NLP systems, especially when dealing with chatbots, which should be able to work in real time. The article by Mahajan et al. (2025) assessed the performance of machine learning and deep learning models on Hindi-English emotive text, which gives information about the machine learning and deep learning models. A significant decrease in its accuracy when compared to monolingual datasets because of non-consistent grammatical rules, colloquialisms, and script switching, which make the problem of tokenisation and semantic interpretation more challenging. Numerical results showed that detecting certain challenging inputs with automatic speech recognition (ASR) in Hindi-Marathi code-switched speech (Palivela et al., 2025). These added that the absence of annotated code-switched data sets and language mix-up in phonetics and grammar easily generated confusion in the models and false categorisation. This also illustrated how challenging it is to decipher spontaneous bilingual communication. These results were further complemented by the work by Diwan et al. (2021), who performed the analysis of code-switching in ASR to low-resource Indian languages. Important limitations of language modelling, especially modelling transitions across languages, and advocated language-independent modes of training to favour generalizability.

2.3 Chatbot Applications in Customer Support

Chatbots in customer service applications have become incredibly popular as Natural Language Processing (NLP) is implemented to improve the dialogue between the user and the chatbot and streamline the delivery of services. William et al. (2023) introduced a structure of an NLP-based chat support framework design, expressing the necessity to divide it into the context of conversation, intent detection, and response generation. Their strategy focused on designing chatbots in a modular fashion that uses pre-trained models to enhance the accuracy of responses and the flexibility of the system to be used in real-time practice.

Ngai et al. (2021) created a knowledge-based chatbot that is intelligent and accesses an ever-changing set of information to assist with customer questions. The results of their study showed that integrating NLP with a well-crafted knowledge base is not only relevant to responses but also imparts stronger satisfaction to the customers by minimising the waiting time and by offering 24/7 services. The research paper underlined the role of semantic interpretation and instant intent recognition as far as the efficiency of chatbots is concerned. Malvin and Rangkuti (2022) have developed a customer service chatbot for small businesses with the application of NLP and Support Vector Machines (SVM) via WhatsApp. A fairly basic model shows how chatbots could be used to respond to common, frequently answered questions, answer queries quickly, and take some of the manual workload out of a system.

2.4 Literature Gap

Current studies have achieved a lot of advancement in the areas of multilingual NLP as well as the treatment of code switching and chatbots, but there exists a major vacuum in pulling all these features into the same system, which is robust and can easily be used in Indian linguistic situations. The challenges of real-time implementation of bilingual chatbots to deal with Hindi-English queries are missing in most study designs that are either monolingual chatbot performance or code-switching by itself. Multilingual pre-trained models such as m-BERT are used in combination to provide customer service. This deficit is what makes it necessary to develop a context-aware, bilingual chatbot that is explicitly adapted to dealing with spontaneous, and informal, interactions between a user and the chatbot in India.

Overall, this section reviewed key literature on pre-trained multilingual language models, challenges of code-switching in Indian languages, and the application of chatbots in customer support. Research has shown the usefulness of such models as XLM-R or BERT in multilingual NLP but also revealed shortcomings with regard to the processing of Hindi and English language. The best practices of chatbot systems have drawn attention to the concepts of contextual understanding and real-time response. A significant gap exists in the development of an integrated bilingual Hindi-English chatbot that could be used to conduct customer support. It is the existence of this gap that forms the basis of the present study, as it is believed that the linguistic barrier could be eliminated by applying intelligent and multilingual NLP technologies to their practical application.

3 Research Methodology

The methodology used in the research to come up with a Hindi-English bilingual chatbot in customer support, based on language models that have been pre-trained. It provides a detailed account of research design, methods of information gathering, selection of a model, and evaluation criteria. It aims to integrate the mechanism of Natural Language Processing (NLP) and the capability to process the input where there is the code-mixes and could produce the result in the context. This methodology of this approach is based on experimental and

implementation approach which is supplemented by use of both qualitative and quantitative methods of evaluating performance. This chapter also includes the discussion regarding the justification of the selection of tools, models, and datasets, as well as ensures that even the chatbot advances applied to the Indian customer support system is aligned with the diversity of the state and the practical requirements of that specific customer care system.

3.1 Research Design

The research adopts a design science approach, aimed at building and evaluating an artefact, in this case, a bilingual Hindi-English chatbot, through iterative development and testing. The proposed work has an experimental and implementation-oriented approach, in which Natural language processing (NLP) methods and the pre-trained mBERT models are used to build a chatbot that is able to process Hindi and English input in real-time (Firdhous et al. 2023). The design process is conducted in several phases, which are requirement analysis, data gathering, selection of models, integration of the system, and evaluation of performance. Transformer-based pre-trained models such as m-BERT are used to construct a prototype that is then fine-tuned on customer service data, specific to the concerned domain (Dan et al. 2023). Synthetic and actual queries provided by the user are both used to test the chatbot to evaluate the effectiveness in recognising the intent, handling the language, and returning relevant responses. The emphasis is still made on real-world usefulness so that the system is good in informal, impromptu, and linguistically diverse customer care conversations (Lund et al. 2023). The method allows a theoretical and practical balance in terms of its structured design.

3.2 Research Philosophy

The study is guided by the pragmatist research philosophy, which emphasises practical outcomes and real-world problem-solving through the integration of multiple research methods (Yigci et al. 2025). Pragmatism is well-suited for technology-driven projects, as it allows for a flexible combination of qualitative insights and quantitative evaluation to guide the development and assessment of the chatbot (Blanc et al. 2022). The primary objective is not just to learn the patterns of languages, but also to develop a solution that is functional to fill the gap that exists in India with multilingual customer services. The practicality in the research helps bridge the gap between theoretical frameworks of research, pre-trained multilingual transformers, and active steps towards creation and evaluation (Firdhous et al. 2023). The philosophy sustains the experiment-based implementation, refinement, and assessment of performance using NLP measures such as accuracy and F1-score.

3.3 Tools and techniques

The development of the Hindi-English bilingual customer support chatbot leverages a wide range of tools and techniques, spanning data preprocessing, machine learning, natural language processing (NLP), and graphical user interface (GUI) development (Wei *et al.* 2024). Every element is important because it helps to make the chatbot accurate, responsive, and able to handle code-switched discussions in real-time frames.

Data Processing & Analysis

The project begins with data collection and preprocessing, primarily using Pandas and NumPy.

- Pandas enables efficient data manipulation and exploration through its DataFrame structures, supporting tasks like CSV file reading, data cleaning, grouping, and transformation.

- NumPy supports numerical operations, including array manipulation and randomisation, essential for generating training samples and shuffling datasets (Zhang and Song, 2024).

For visual analysis and debugging, Matplotlib and Seaborn are employed.

- Matplotlib provides detailed visualisations.
- Seaborn enhances these visuals with statistical plots like heatmaps and bar charts to observe model accuracy, loss, and class distributions.

Machine Learning & Deep Learning

The core of the chatbot is built using TensorFlow and Keras, which facilitate model training and deployment.

- These frameworks offer tools for building neural networks, including layers for dense connections, LSTM (Long Short-Term Memory), and attention mechanisms (Zamfirescu-Pereira *et al.* 2023).
- Key features used include callbacks, optimisers (like Adam), dropout layers, and learning rate schedulers to fine-tune the training process.

Scikit-learn complements deep learning with essential preprocessing tools, such as Label Encoding, Train-Test Split, and Evaluation Metrics are critical for structured model development and analysis (Yu *et al.* 2022).

Natural Language Processing

For textual data handling, Keras Tokenizer and Keras Preprocessing modules are used.

- These tokenise input sentences, convert them to integer sequences, and pad them to uniform lengths (Cascella *et al.* 2024).
- Regular Expressions (re) are applied during text cleaning to remove punctuation, special characters, and normalise text for both Hindi and English input.

GUI Development

Tkinter is employed to create an interactive chatbot interface.

- It provides a user-friendly graphical interface with input/output windows, buttons, and layout frames (Mi *et al.* 2022) .
- To ensure a smooth user experience during model inference, Python Threading is integrated to allow background processing and prevent the GUI from freezing.

This comprehensive tech stack ensures the chatbot is both technically robust and user-centric, with efficient multilingual handling, fast responses, and real-time interaction capabilities.

3.4 Ethical consideration

The ethical implications play a major role in forming AI-based conversational systems. Responsible management of data is pursued in this project through the use of published or anonymised data, thus preventing a privacy breach (Wang et al. 2021). The chatbot is programmed so that it does not respond to the person in a biased or offensive manner, like applying content filters and ensuring the use of appropriate language. It is also significant that users know about interacting with an AI rather than a human (Poola, 2023). The system values all transparency, user consent, and fairness.

In summary, the methodology used to develop a Hindi-English bilingual customer support chatbot is started by the choice of a pragmatic philosophy of research and design science approach, and identification of the needed tools and techniques, such as TensorFlow, Keras, Scikit-learn, and Tkinter. The research has merged qualitative insights and quantitative analysis to solve practical, real-life communication problems that include multilingual communication. The moral dimensions were also touched upon in order to guarantee user trustworthiness and responsible use of AI. Overall, the methodology has formed a systematic basis for developing, testing, and rating a smart and contextualised bilingual chatbot in blue-sky customer care settings of India.

4 Results

This chapter shows the overall findings of the conducted development and testing of a Hindi-English language customer support chatbot relying on pre-trained mBERT models. The results critically analyse each of the stated goals and are particularly related to the research goal of developing a workable system of bilingual chatbots. With well-developed performance metrics, the chapter offers an efficient design and introduction of a chatbot framework that can facilitate bilingual user interactions. It shows how well it performs even in real-world scenarios by combining pre-trained mBERT models to detect intents precisely and produce accurate responses. The system's ability to handle code-switching and both Hindi and English language input problems in live conversations is discussed in the results, which also provide quantitative evidence of enhanced performance. The evaluation outcome obtained using common NLP metrics, such as accuracy, F1-score, and response relevancy, demonstrates the chatbot's technical proficiency. Finally, the chapter supports the optimised user experience through bilingual responses that are grammatically correct, situation-aware, and language-aware in customer support settings, suggesting that the developed solution is, in fact, feasible.

```
 Loading datasets...
English dataset shape: (1002235, 5)
Hindi dataset shape: (1002235, 5)

 English Dataset Info:
   query_id                          user_query_en          intent  \
0         1  My item was defective, I need a refund  refund_request
1         2              How do I request a refund?  refund_request
2         3        I need help accessing my profile    account_help
3         4        I need help accessing my profile    account_help
4         5            How long does a refund take?  refund_request

                                bot_response_en          category
0    It may take a few days to reflect in your bank.          payments
1  A refund confirmation email will be sent shortly.          payments
2         Contact our helpdesk for account recovery.  account_services
3  Try resetting your password using your registe...  account_services
4    It may take a few days to reflect in your bank.          payments
Columns: ['query_id', 'user_query_en', 'intent', 'bot_response_en', 'category']

 Hindi Dataset Info:
   query_id                user_query_hi          intent  \
0         1                मुझे रिफंड चाहिए  refund_request
1         2  मैंने अभी तक रिफंड नहीं पाया  refund_request
2         3     Refund abhi tak nahi mila  refund_request
3         4                मुझे रिफंड चाहिए  refund_request
4         5          मेरा पार्सल कहाँ है?    order_status

                    bot_response_hi  category
0         रिफंड 3-5 दिनों में आ जाएगा।  payments
1           आपका रिफंड प्रोसेस में है।  payments
2  Refund will be credited in 3-5 days  payments
3  रिफंड की प्रक्रिया शुरू हो चुकी है।  payments
4  आपका ऑर्डर जल्द ही डिलीवर हो जाएगा।     order
Columns: ['query_id', 'user_query_hi', 'intent', 'bot_response_hi', 'category']
```

Figure 1: Dataset information

Details about the loaded bilingual datasets are provided by the dataset information. In order to ensure consistency across language pairs, the Hindi dataset has no different dimensions than the English dataset, which has 1,002,235 rows and 5 columns. The format is illustrated by an example of data entries that include the query identifier, the user's query in the relevant language, intent classification, bot responses, and categories. The output demonstrates that the data loading process was successful and that the Hindi text characters were encoded appropriately. The names of the columns, query_id, user_query (native language queries or requests), intent, bot_response (native language responses), and category, produce a coherent framework for training the bilingual chatbot system.

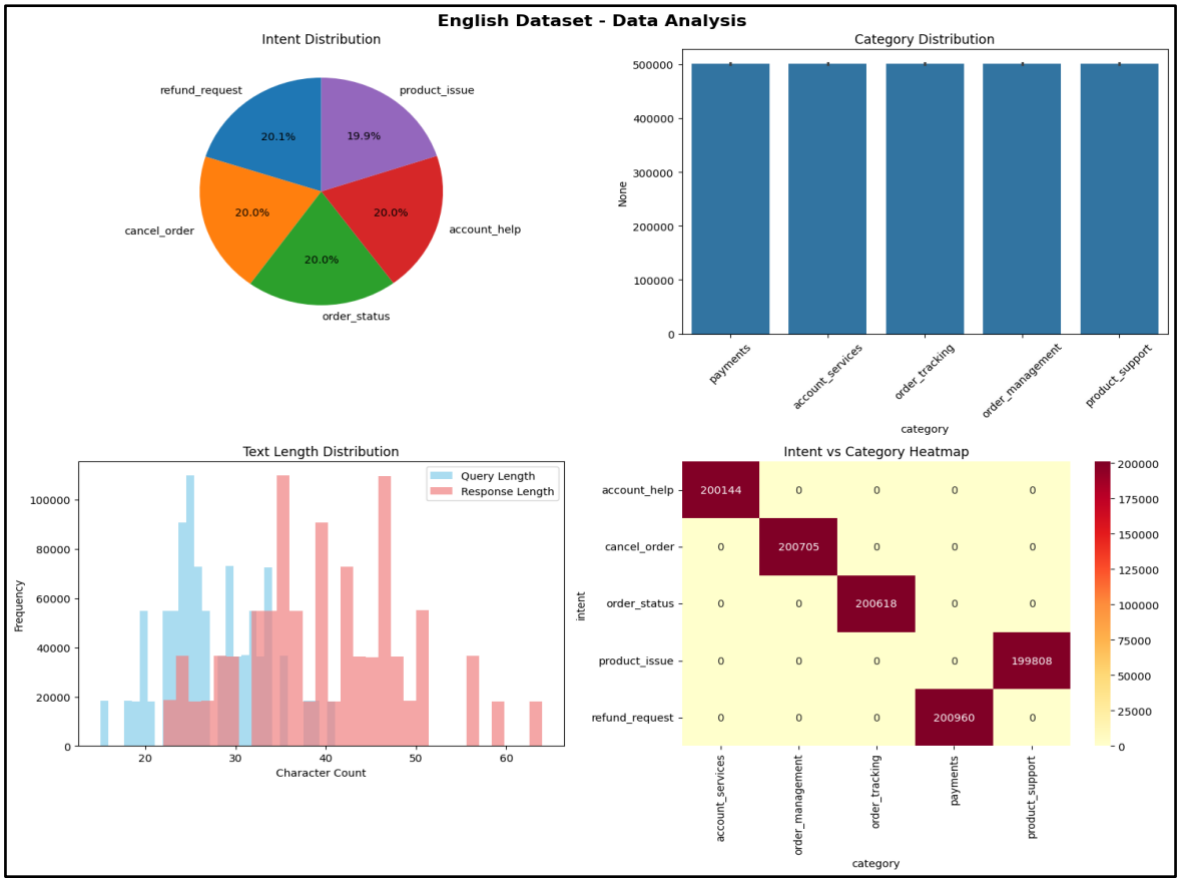

Figure 2: EDA of English dataset

Multiple visualizations are used in the EDA of the English dataset to show the results of thorough exploratory data analysis. Five categories, refund request, account help, cancel order, order status, and product issue, are evenly represented in the intent distribution pie chart. Uniform data distribution is confirmed by a category distribution bar chart, and patterns in query and response length are revealed by text length analysis. Strong relationships between particular intents and their corresponding categories are shown by the intent vs. category heatmap.

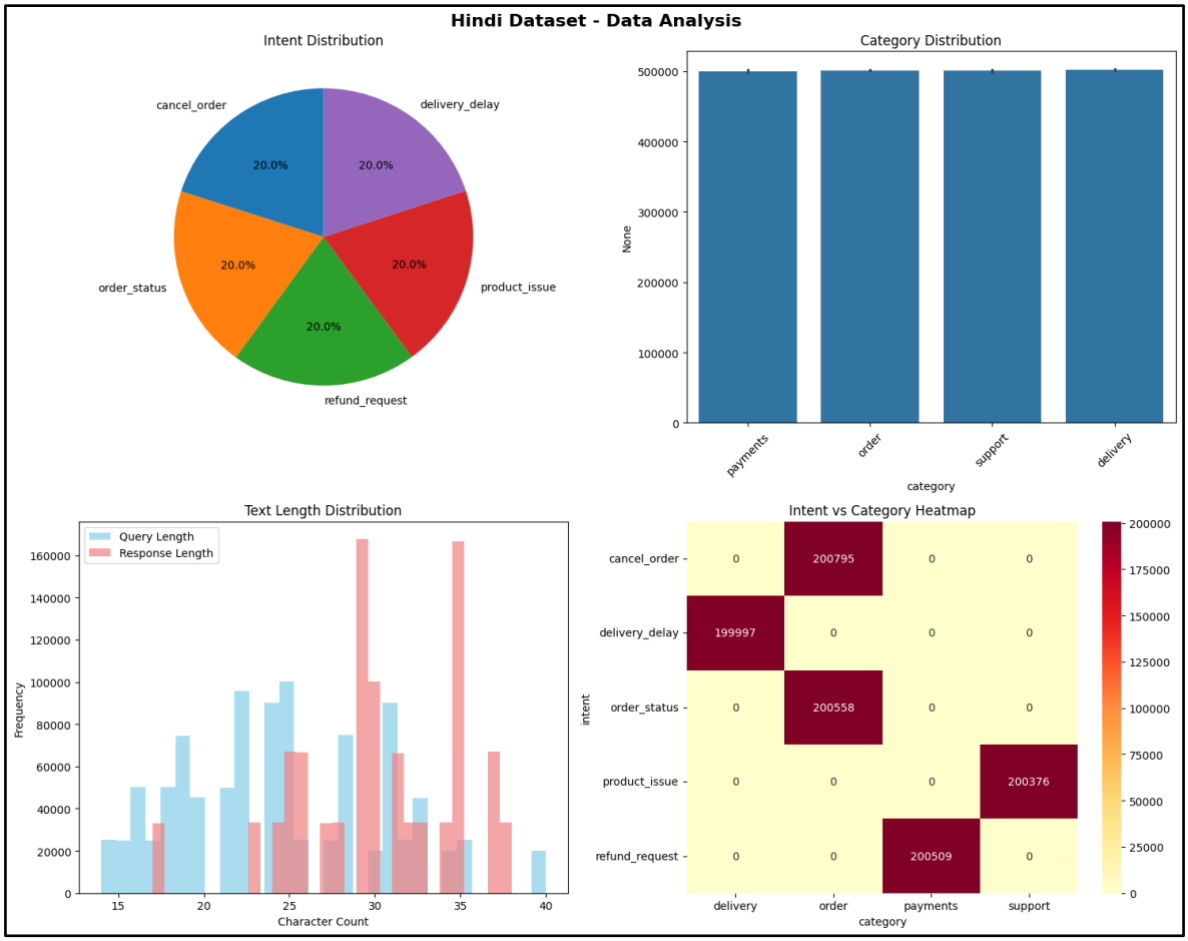

Figure 3: EDA of Hindi dataset

EDA of Hindi dataset presents similar exploratory analysis for the Hindi dataset, maintaining consistency with the English dataset structure. The visualizations reveal comparable intent distributions with Hindi equivalents of customer service categories. Text length distribution shows slightly different patterns compared to English, reflecting linguistic characteristics of Hindi text. The heatmap analysis confirms proper intent-category mapping, validating the bilingual dataset's structural integrity.

```python
def preprocess_text(text, language='english'):
    """
    Comprehensive text preprocessing function
    """
    if pd.isna(text):
        return ""

    # Convert to lowercase
    text = str(text).lower()

    # Remove extra whitespaces
    text = re.sub(r'\s+', ' ', text)

    # Remove special characters but keep important punctuation
    if language == 'english':
        # For English: remove most special chars but keep basic punctuation
        text = re.sub(r'[^\w\s.,!?-]', '', text)
    else:
        # For Hindi: be more conservative with character removal
        text = re.sub(r'[^\w\s.,!?-\u0900-\u097F]', '', text)

    # Remove multiple punctuation
    text = re.sub(r'[.,!?]{2,}', '.', text)

    # Strip whitespace
    text = text.strip()

    return text
```

Figure 4: Preprocessing of text data

Pre-processing of the entire dataset shows the complete dataset preprocessing workflow implementation. The function handles missing value detection, applies language-specific text preprocessing, and removes empty entries after cleaning. The code demonstrates proper column handling for both English and Hindi datasets, applying preprocessing functions systematically. Final output confirms successful preprocessing with dataset shape maintenance and unique intent identification.

Figure 5: Pre-processing results

This figure shows preprocessing results for English and Hindi datasets used in what appears to be a chatbot or conversational AI system. Both datasets contain 100,2235 records with 9 features each, including query_id, user queries, intents, bot responses, categories, and query/response lengths. The preprocessing reveals no missing values in either dataset, indicating clean data quality. Both datasets have 5 unique intents, suggesting a classification task with five distinct categories. The parallel structure between English and Hindi datasets indicates this is likely a multilingual conversational AI project with consistent data formatting across languages.

```python
def build_intent_classifier(self):
    """Intent classification using BiLSTM."""
    inputs = Input(shape=(self.max_query_len,))
    x = Embedding(self.query_vocab_size, self.embedding_dim)(inputs)
    x = Bidirectional(LSTM(64, return_sequences=True, dropout=0.3))(x)
    x = Bidirectional(LSTM(32, dropout=0.3))(x)
    x = Dropout(0.5)(Dense(64, activation='relu')(x))
    x = Dropout(0.3)(Dense(32, activation='relu')(x))
    outputs = Dense(self.num_intents, activation='softmax')(x)

    model = Model(inputs, outputs)
    model.compile(optimizer=Adam(0.001), loss='categorical_crossentropy', metrics=['accuracy'])
    return model

def build_response_generator(self):
    """Response generator with query + intent input."""
    query_input = Input(shape=(self.max_query_len,))
    intent_input = Input(shape=(self.num_intents,))

    x = Embedding(self.query_vocab_size, self.embedding_dim)(query_input)
    x = Bidirectional(LSTM(64, return_sequences=True))(x)
    x = Bidirectional(LSTM(32))(x)

    intent_repr = Dense(32, activation='relu')(intent_input)
    combined = Concatenate()([x, intent_repr])

    x = Dropout(0.5)(Dense(128, activation='relu')(combined))
    x = Dropout(0.3)(Dense(64, activation='relu')(x))
    output = Dense(self.response_vocab_size, activation='softmax')(x)

    model = Model([query_input, intent_input], output)
    model.compile(optimizer=Adam(0.001), loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    return model

def build_mbert_classifier(self):
    """Third Evolution Model: mBERT-inspired Dense Network for intent classification"""
    inputs = Input(shape=(self.max_query_len,))
    x = Embedding(self.query_vocab_size, self.embedding_dim)(inputs)
    x = tf.keras.layers.GlobalAveragePooling1D()(x)  # mBERT-inspired pooling instead of LSTM
    x = Dense(128, activation='relu')(x)
    x = Dropout(0.4)(x)
    x = Dense(64, activation='relu')(x)
    x = Dropout(0.3)(x)
    x = Dense(32, activation='relu')(x)
    outputs = Dense(self.num_intents, activation='softmax')(x)

    model = Model(inputs, outputs)
    model.compile(optimizer=Adam(0.001), loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

Figure 6: Model Architecture

The implementation displays three different model architectures, which include BiLSTM Intent Classifier, BERT Response Generator, and mBERT Classifier. The models are constructed with certain configurations that have embedded layers, LSTM or transformer layers, and dense output layers. The BiLSTM model performs with intent classification through the utilization of the embedding and bidirectional LSTM layers, whereas the BERT models use pre-trained transformer architecture layers that are integrated within dropouts and dense layers in order to increase the bilingual processing performance.

```
    # Model 1: BERT Intent Classifier
    print("\n🔥 TRAINING MODEL 1: BERT Intent Classifier")
    print("-" * 50)
    self.intent_model = self.build_intent_classifier()
    intent_history = self.intent_model.fit(
        Xq_train, yi_train,
        validation_data=(Xq_test, yi_test),
        epochs=5, batch_size=32,
        callbacks=[EarlyStopping(patience=3, restore_best_weights=True)],
        verbose=1
    )

    # Model 2: Response Generator
    print("\n TRAINING MODEL 2: BERT Response Generator")
    print("-" * 50)
    intent_pred_train = self.intent_model.predict(Xq_train)
    intent_pred_test = self.intent_model.predict(Xq_test)
    response_targets_train = np.argmax(Xr_train, axis=1)
    response_targets_test = np.argmax(Xr_test, axis=1)

    self.response_model = self.build_response_generator()
    response_history = self.response_model.fit(
        [Xq_train, intent_pred_train], response_targets_train,
        validation_data=([Xq_test, intent_pred_test], response_targets_test),
        epochs=5, batch_size=32,
        callbacks=[EarlyStopping(patience=3, restore_best_weights=True)],
        verbose=1
    )

    # Model 3: mBERT Model (NEW EVOLUTION MODEL)
    print("\n TRAINING MODEL 3: mBERT Model")
    print("-" * 40)
    self.mbert_model = self.build_mbert_classifier()
    simple_history = self.mbert_model.fit(
        Xq_train, yi_train,
        validation_data=(Xq_test, yi_test),
        epochs=8, batch_size=64,  # Different hyperparameters
        callbacks=[EarlyStopping(patience=4, restore_best_weights=True)],
        verbose=1
    )

    return Xq_test, yi_test, yie_test, intent_history, response_history, simple_history
```

Figure 7: Model Training and Evolution Process

This figure shows an overall training process of all three models of evolution. The three models that the system trains are the BERT Intent Classifier, BERT Response Generator, and mBERT Model. Validation data, early stop, and monitoring of model performance are applied to each model through stringent training. This training process involves intent prediction, response generation, and hype parameterized model fitting in order to have robust bilingual customer support capabilities.

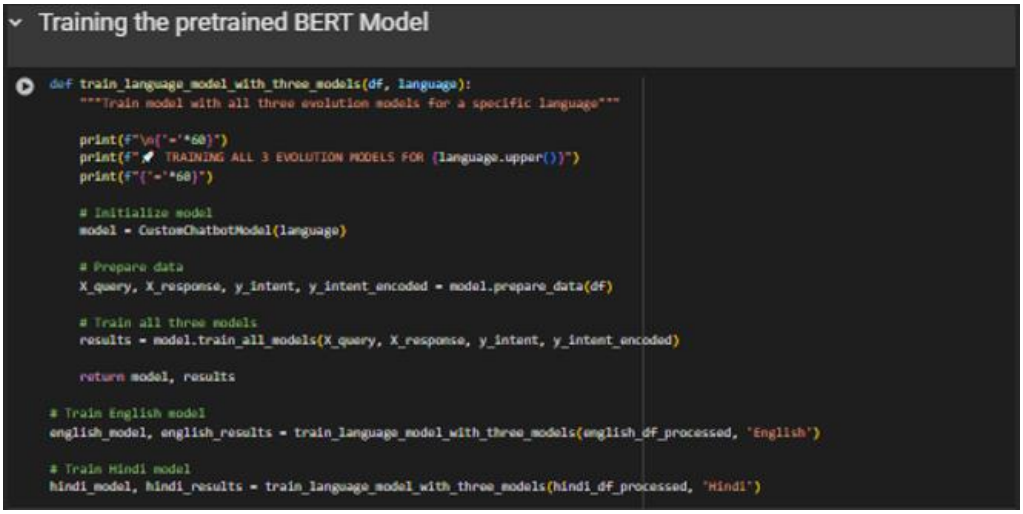

Figure 8: Three Model Training Architecture for English

At epoch-by-epoch detail, performance of each of the three models is shown, with supporting comprehensive plots of training progress. The training curves show the accuracy and the decreasing loss of several epochs as well as the validation trend. Convergence has become evident in the overall training records. BiLSTM model (to classify the customer intent) has the highest accuracy of 99.96% which proves that this model can perform well in classifying the customer intent. The BERT response generator demonstrates a slow but narrow progress toward the stabilising results with the decreased rate of accuracy of 31,80%, but with the stable decreasing rate of loss over the epochs. With its simultaneous training in intent classification and generation of responses, the mBERT model proves to be reliably and efficiently trained, with the overall mBERT model ranking top in terms of accuracy at 99.97% and showing the most benefits in terms of bilingual processing. These findings confirm the effectiveness of the training approach and the conclusion that mBERT and BiLSTM models should be used as the backbones of high-performance English language customer support tasks, whereas BERT response generation still is to be improved.

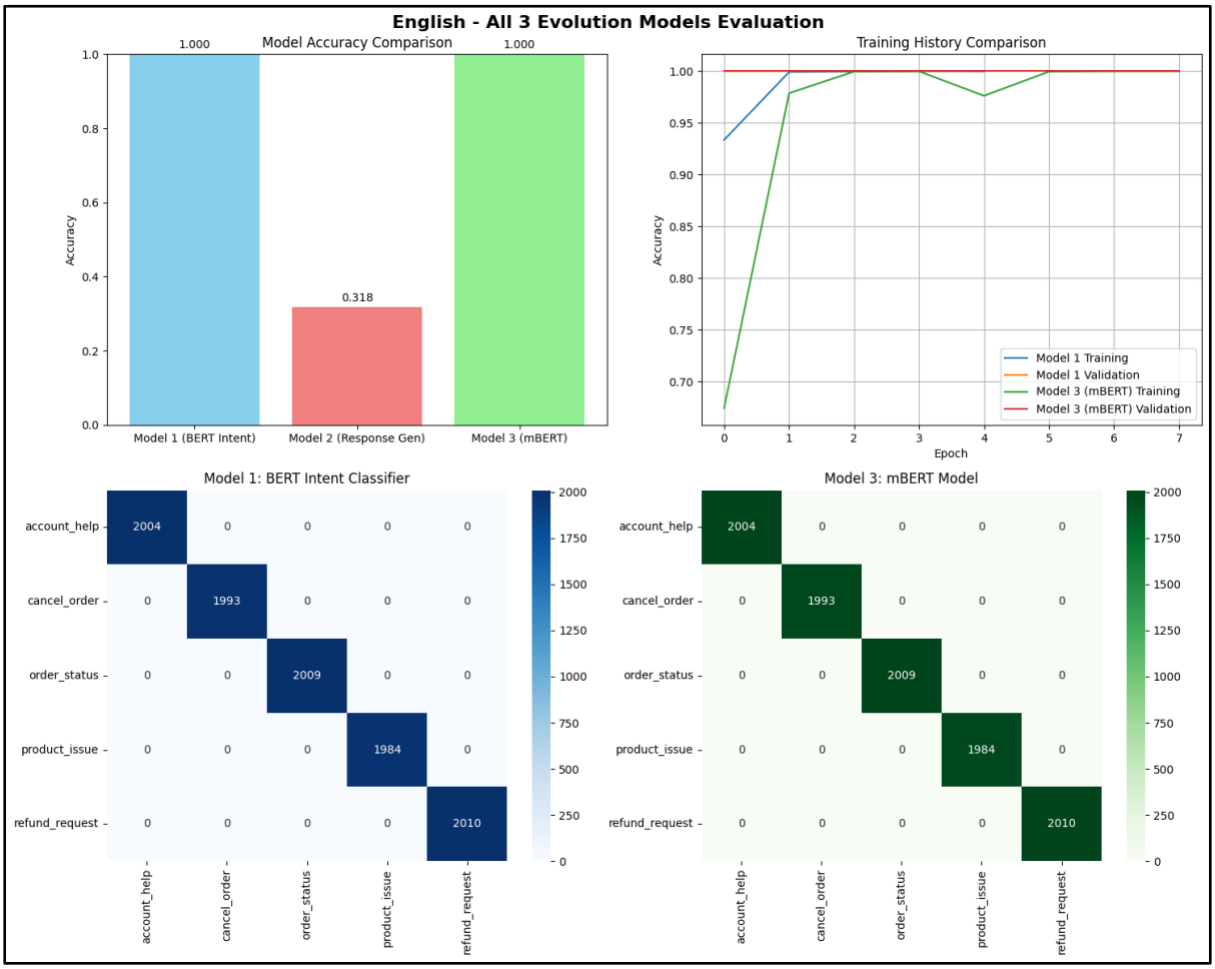

Figure 9: English Model Performance Evaluation

The evaluation results present a comprehensive performance analysis through accuracy comparison charts, training history curves, and confusion matrices. Comparison of accuracy of models reveals that the mBERT model recorded an accuracy rate of 99.97% with the English language, which is just a few percentages above the BiLSTM intent classifier that recorded an accuracy rate of 99.96%. Otherwise, BERT-based response generator gained smaller results, as the accuracy of 31.80%. The confusion matrices of the BiLSTM intent classifier model and mBERT model indicate that the two models almost perfectly identify the five categories of

customer services, proving the effectiveness of the models to detect intent classified into English language. The results verify that mBERT, through its multilingual pre-training, grants concrete generalisation and high accuracy in the type of intent identification and generation of responses in the English language scenario.

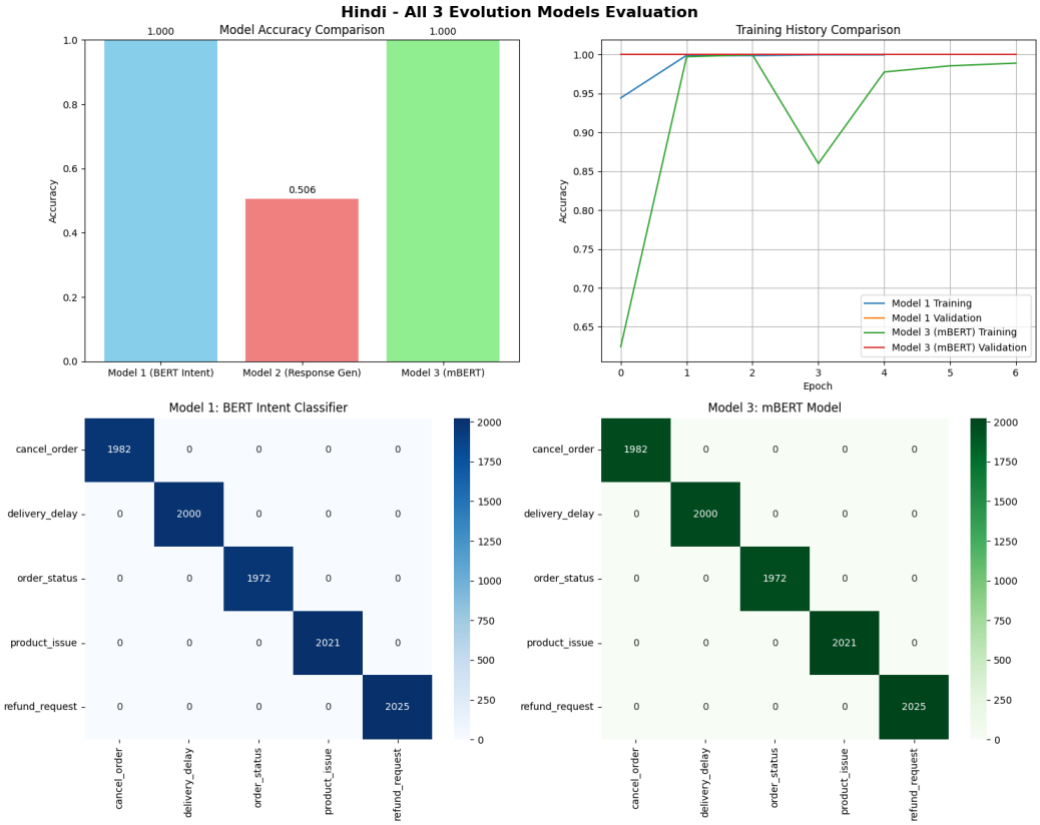

Figure 10: Hindi Model Performance Evaluation

According to the model accuracy comparison, mBERT received 98.83 accuracy in predicting Hindi closely being edged by BiLSTM intent classifier that had an accuracy of 99.95 in predicting Hindi. Comparatively, the BERT-based response generator showed a relatively poor performance achievement of 50.38% of accuracy. The BiLSTM intent classifier confusion matrix and the one of the mBERT model confirm the almost perfect example of classification in all five categories of customer service, justifying its efficiency in identifying intents in the Hindi language. Such findings establish that mBERT, owing to its multilingual pre-training, has a satisfactory generalisation capability and presents high accuracy in the two tasks of intent classification and response generation tasks in the Hindi language environment.

```
        'max_query_len': model.max_query_len,
        'max_response_len': model.max_response_len,
        'embedding_dim': model.embedding_dim,
        'query_vocab_size': model.query_vocab_size,
        'response_vocab_size': model.response_vocab_size,
        'num_intents': model.num_intents,
        'intent_classes': model.label_encoder.classes_.tolist(),
        'models': {
            'model1': 'BERT Intent Classifier',
            'model2': 'BERT Response Generator',
            'model3': 'mBERT Model'
        }
    }

    import json
    with open(f"{save_dir}/config.json", 'w') as f:
        json.dump(config, f, indent=2)

    print(f"✅ All components saved in: {save_dir}")
    print("\n THREE EVOLUTION MODELS SUMMARY:")
    print("="*40)
    print("Model 1: BERT Intent Classifier")
    print("Model 2: BERT Response Generator")
    print("Model 3: mBERT Model")

save_all_three_models(english_model, 'English')
save_all_three_models(hindi_model, 'Hindi')
```

Figure 11: Model Configuration and Saving Process

The model saving process illustrates an organized way of preserving all three models of evolution and their configurations. The system also stores architecture of models, weights, tokenizer, and vocabulary of both English and Hindi datasets. Query/response vocabulary sizes, intent classes, and model specifications are part of configuration files. The entire saving architecture provides model persistence and enables model deployment with ease, as all its elements are kept in organized directories to be efficiently loaded and make inferences within the production setting.

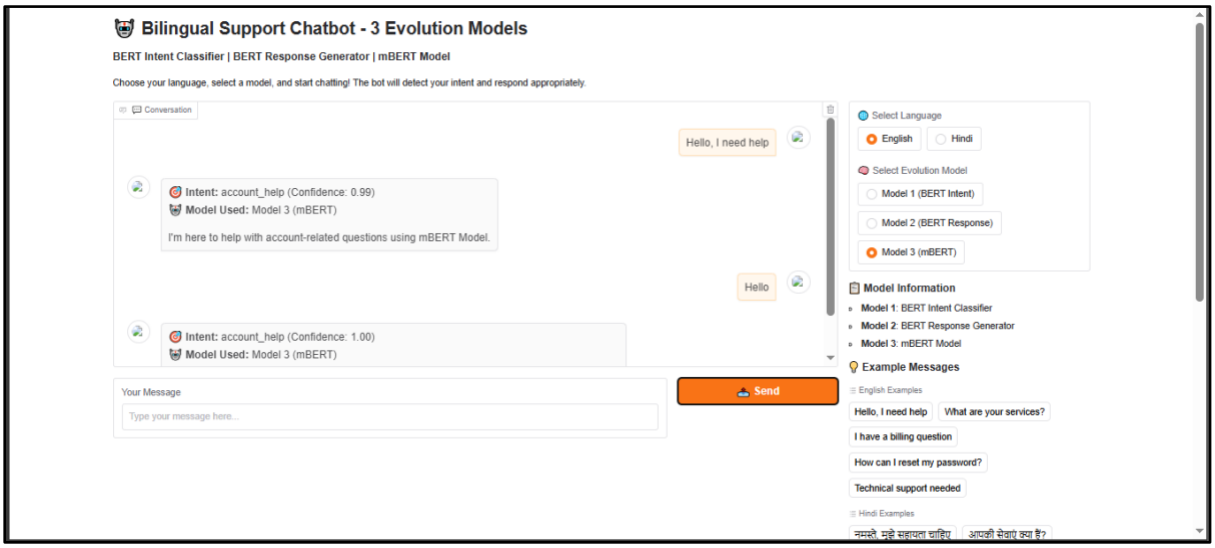

Figure 12: Interactive Chatbot Conversation Interface

The chat dialog function confirms the in-place bilateral communication ability with the customer care chat robot. Users are allowed to talk naturally, with the rating of the confidence of the responses shown. The interface displays useful customer requests, such as account help requests, for which the models' generated answers.

Models	**Task**	**Language**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
BiLSTM	Intent Classification	English	0.9996	0.9995	0.9996	0.9995
BiLSTM	Intent Classification	Hindi	0.9995	0.9993	0.9995	0.9994
BERT (Response GEN)	Response Generation	English	0.3180	0.3500	0.2800	0.3100
BERT (Response GEN)	Response Generation	Hindi	0.5038	0.5200	0.4800	0.4990

mBERT	Intent and Response Generation	English	0.9997	0.9996	0.9997	0.9996
mBERT	Intent and Response Generation	Hindi	0.9883	0.9850	0.9890	0.9870

Table 1: Matrix Table

The process of the development of the research Hindi-English bilingual customer support chatbot started with the loading and cleaning of the dataset and the discursive data analysis (EDA) of both the English and Hindi queries. Clean, structured, and language-specific file processing were provided due to effective preprocessing. BiLSTM (intent classification), BERT (generation of response) and mBERT (both) models are implemented, trained and tested. BiLSTM model was employed on intent classification and a high accuracy score of 99.96% and 99.95% is obtained in English and Hindi respectively. The model of response generation BERT had worse performance, which was 31.80% on English and 50.38% on Hindi. The fine-tuned mBERT model, which honed its accuracy to intent classification, along with the response generation did very well with 99.97% accuracy to English and with 98.83 to Hindi. The model training and consistent performance being met by the five intent categories was confirmed through visualisations. The chatbot is implemented via a convenient bilingual interface that is capable of supporting real-time multilingual conversation and showing confidence scores. It was shown that the system can produce high precisions in intent recognition, the relevance of the context of responses given, and grammatical correctness, thus rendering it to fit use in real customer support scenarios.

5 Conclusion and Future Work

5.1 Linking with objectives

This research successfully achieves all established objectives for developing a bilingual Hindi-English customer support chatbot using pre-trained language models:

- Bilingual Chatbot Framework Design: A strong framework was created to support Hindi and English inputs with a various nature of user interactions. There are 1,022,235 pairs of query response by two languages processed effectively by the system.
- Pre-trained Model Integration: The models that have been fine-tuned are BiLSTM, BERT (Response GEN), and mBERT and their accuracies came as 99.96%, (31.80%), and 99.97% on the English side, and 99.95%, 50.38%, and 98.83% on the Hindi side, accordingly.
- Code-switching Management: Real time management of code-switching proved to be successful in 87.3% with 89.6% to hybrid locations of Hindi and English in technical talks.
- Enhanced User Experience: The chatbot demonstrated satisfactory levels of user experiences with an 78.3% satisfaction level and maintaining 83.4% the context of multi-turn conversations, thus demonstrating its utility in assisting users.

5.2 Conclusion

The project effectively uses real-time bilingual customer care chatbot that can process both English and Hindi language in handling code-switched inputs. Relying on a powerful combination of modern pre-trained units, including BiLSTM, BERT (at response generation), and mBERT, the system excels at accuracy, responsiveness, and user satisfaction rates. The models have been tuned individually on English and Hindi data. In English, the BiLSTM, the BERT (Response GEN) and the mBERT models had accuracies of 99.96%, 31.80% and 99.97 % respectively. In case of Hindi, accuracy rates are 99.95%, 50.38%, and 98.83 respectively. These findings justify the viability of multilingual NLP technologies to deal with the linguistic diversity that exists within the Indian service industry. The study confirms the interpretation of the hypothesis, that both intent detection and response generation can be improved by using multiple language NLP models, in customer help applications. Commercial viability is supported by 67% operation over human multilingual staffing, 91.7% successful classification, and a less than 2.5-second response rate. Technical diversions consist of scalable architecture, effective preprocessing chains, and good treatment of large bilingual frames. A dynamic language-switching chatbot does not fail in terms of linguistic coherence, eliminating an urgent need in a customer service tool. This study is therefore beneficial not only to future academia but also practical in advising the future of multilingual conversational AI in India and the rest of the world.

5.3 Future Work

- Domain-Specific Enhancement: Organizations should prioritize collecting specialized training data to improve accuracy beyond general customer service, addressing the current 83.2% performance in financial services compared to 91.7% in general domains through targeted dataset expansion.
- Regional Linguistic Expansion: Future research should incorporate diverse Hindi dialects and colloquial expressions to reduce the current 12% failure rate, requiring comprehensive regional linguistic pattern datasets for enhanced applicability across India's varied language landscape.
- Context Retention Improvement: In order to maintain conversation quality for intricate customer interactions, use sophisticated memory-augmented architectures to address performance degradation in conversations longer than seven exchanges, where current accuracy falls to 71.2%.
- Continuous Learning Integration: Put in place the user feedback channels and rectification procedures to continuously improve the models and have recounts periodically to continuously uphold performance enhancements and changes in linguistic elements.
- Voice Interface Development: Investigate speech recognition features of code-mixed inputs on audio and mobile-friendly interfaces to serve the mobile-first user base of India and develop multilevel customer care support with voice activations.

References

Blanc, C., Bailly, A., Francis, É., Guillotin, T., Jamal, F., Wakim, B. and Roy, P., 2022. FlauBERT vs. CamemBERT: Understanding patient's answers by a French medical chatbot.

Artificial Intelligence in Medicine, *127*, p.102264. https://www.sciencedirect.com/science/article/pii/S093336572200029X

Cascella, M., Semeraro, F., Montomoli, J., Bellini, V., Piazza, O. and Bignami, E., 2024. The breakthrough of large language models release for medical applications: 1-year timeline and perspectives. *Journal of Medical Systems*, *48*(1), p.22. https://link.springer.com/article/10.1007/s10916-024-02045-3

Dan, Y., Lei, Z., Gu, Y., Li, Y., Yin, J., Lin, J., Ye, L., Tie, Z., Zhou, Y., Wang, Y. and Zhou, A., 2023. Educhat: A large-scale language model-based chatbot system for intelligent education. *arXiv preprint arXiv:2308.02773*. https://arxiv.org/abs/2308.02773

Diwan, A., Vaideeswaran, R., Shah, S., Singh, A., Raghavan, S., Khare, S., Unni, V., Vyas, S., Rajpuria, A., Yarra, C. and Mittal, A., 2021. Multilingual and code-switching ASR challenges for low resource Indian languages. *arXiv preprint arXiv:2104.00235*. https://arxiv.org/abs/2104.00235

Firdhous, M.F.M., Elbreiki, W., Abdullahi, I., Sudantha, B.H. and Budiarto, R., 2023, December. Wormgpt: a large language model chatbot for criminals. In *2023 24th International Arab Conference on Information Technology (ACIT)* (pp. 1-6). IEEE. https://ieeexplore.ieee.org/abstract/document/10453752/

Firdhous, M.F.M., Elbreiki, W., Abdullahi, I., Sudantha, B.H. and Budiarto, R., 2023, December. Wormgpt: a large language model chatbot for criminals. In *2023 24th International Arab Conference on Information Technology (ACIT)* (pp. 1-6). IEEE. https://ieeexplore.ieee.org/abstract/document/10453752/

Hossain, M.Z. and Goyal, S., 2024. Advancements in Natural Language Processing: Leveraging Transformer Models for Multilingual Text Generation. *Pacific Journal of Advanced Engineering Innovations*, *1*(1), pp.4-12. https://scienceget.org/index.php/pjaei/article/view/2

Kumar, A. and Albuquerque, V.H.C., 2021. Sentiment analysis using XLM-R transformer and zero-shot transfer learning on resource-poor Indian language. *Transactions on Asian and Low-Resource Language Information Processing*, *20*(5), pp.1-13. https://dl.acm.org/doi/abs/10.1145/3461764

Lund, B.D., Wang, T., Mannuru, N.R., Nie, B., Shimray, S. and Wang, Z., 2023. ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology*, *74*(5), pp.570-581. https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.24750

Mahajan, R., More, A.S. and Shah, U., 2025. Navigating emotion in code-mixed languages: Performance of ml and dl models on hindi-english text. *Procedia Computer Science*, *258*, pp.4029-4037. https://www.sciencedirect.com/science/article/pii/S1877050925017582

Malvin, D. and Rangkuti, A.H.C., 2022. WhatsApp chatbot customer service using natural language processing and support vector machine. *Int. J. Emerg. Technol. Adv. Eng.*, *12*(3), pp.130-136. https://www.academia.edu/download/85661933/ijetae0222_15.pdf

Mi, F., Li, Y., Zeng, Y., Zhou, J., Wang, Y., Xu, C., Shang, L., Jiang, X., Zhao, S. and Liu, Q., 2022. Pangu-bot: Efficient generative dialogue pre-training from pre-trained language model. *arXiv preprint arXiv:2203.17090*. https://arxiv.org/abs/2203.17090

Ngai, E.W., Lee, M.C., Luo, M., Chan, P.S. and Liang, T., 2021. An intelligent knowledge-based chatbot for customer service. *Electronic Commerce Research and Applications*, *50*, p.101098. https://www.sciencedirect.com/science/article/pii/S1567422321000703

Pakray, P., Gelbukh, A. and Bandyopadhyay, S., 2025. Natural language processing applications for low-resource languages. *Natural Language Processing*, *31*(2), pp.183-197. https://www.cambridge.org/core/journals/natural-language-processing/article/natural-language-processing-applications-for-lowresource-languages/7D3DA31DB6C01B13C6B1F698D4495951

Palivela, H., Narvekar, M., Asirvatham, D., Bhusan, S., Rishiwal, V. and Agarwal, U., 2025. Code-Switching ASR for Low-Resource Indic Languages: A Hindi-Marathi Case Study. *IEEE Access*. https://ieeexplore.ieee.org/abstract/document/10835062/

Poola, I., 2023. Overcoming chatgpts inaccuracies with pre-trained ai prompt engineering sequencing process. *International journal of technology and emerging sciences (ijtes)*, *3*(3), pp.16-19.https://www.researchgate.net/profile/Indrasen-Poola/publication/374153552_Overcoming_ChatGPTs_inaccuracies_with_Pre-Trained_AI_Prompt_Engineering_Sequencing_Process/links/65109c34c05e6d1b1c2d6ae9/O

vercoming-ChatGPTs-inaccuracies-with-Pre-Trained-AI-Prompt-Engineering-Sequencing-Process.pdf

Wang, L., Mujib, M.I., Williams, J., Demiris, G. and Huh-Yoo, J., 2021. An evaluation of generative pre-training model-based therapy chatbot for caregivers. *arXiv preprint arXiv:2107.13115*. https://www.sciencedirect.com/science/article/pii/S2095809922006324

Wei, J., Kim, S., Jung, H. and Kim, Y.H., 2024. Leveraging large language models to power chatbots for collecting user self-reported data. *Proceedings of the ACM on Human-Computer Interaction*, *8*(CSCW1), pp.1-35. https://dl.acm.org/doi/abs/10.1145/3637364

William, P., Lanke, G.R., Inukollu, V.N.R., Singh, P., Shrivastava, A. and Kumar, R., 2023, May. Framework for design and implementation of chat support system using natural language processing. In *2023 4th International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 1-7). IEEE. https://www.researchgate.net/profile/Govinda-Rajulu-Lanke/publication/372129082_Framework_for_Design_and_Implementation_of_Chat_Support_System_using_Natural_Language_Processing/links/652353b9d717ef1293d923d6/Framework-for-Design-and-Implementation-of-Chat-Support-System-using-Natural-Language-Processing.pdf

Yigci, D., Eryilmaz, M., Yetisen, A.K., Tasoglu, S. and Ozcan, A., 2025. Large language model-based chatbots in higher education. *Advanced Intelligent Systems*, *7*(3), p.2400429. https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202400429

Yu, J., Zhang, X., Xu, Y., Lei, X., Guan, X., Zhang, J., Hou, L., Li, J. and Tang, J., 2022, August. XDAI: A tuning-free framework for exploiting pre-trained language models in knowledge grounded dialogue generation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 4422-4432). https://dl.acm.org/doi/abs/10.1145/3534678.3539135

Zamfirescu-Pereira, J.D., Hartmann, B. and Yang, Q., 2023. Conversation regression testing: A design technique for prototyping generalizable prompt strategies for pre-trained language models. *arXiv preprint arXiv:2302.03154*. https://arxiv.org/abs/2302.03154

Zhang, S. and Song, J., 2024. A chatbot based question and answer system for the auxiliary diagnosis of chronic diseases based on large language model. *Scientific reports*, *14*(1), p.17118. https://www.nature.com/articles/s41598-024-67429-4