# Fine-Tuning a Technical Support Chatbot Using Transformer Models: A Comparative Analysis

Reham Faisal Alsubhi
s444003014@uqu.edu.sa

Lamar Waleed Fattah
s444006719@uqu.edu.sa

Layan Adel Babkour
s444002368@uqu.edu.sa

Rakha Matuq Nooh
s444001287@uqu.edu.sa

*Abstract*— **Given the growing importance of enhancing intelligent technical support systems ,this study explores the potential of Transformer-based models in improving chatbot accuracy and efficiency. We present a comparative analysis of BERT, ALBERT, RoBERTa, and GPT-2, optimizing them for technical query classification and response generation. A dataset of 2,000 technical queries* was used processed through various techniques to ensure its quality and suitability for the models. This dataset served as the foundation for fine-tuning each model incorporating optimized hyperparameters and structured architectures. The data was split into 85% for training and 15% for validation, and models were evaluated using Accuracy, Precision, Recall, and F1-score. The results showed that BERT achieved the highest accuracy (94%) outperforming the other models in classification. ALBERT followed with 93% accuracy, proving to be a more memory efficient alternative. GPT-2 while effective in response generation struggled with classification, achieving 80% accuracy whereas RoBERTa performed the worst, with only 3% accuracy indicating poor learning from the dataset. Key performance metrics emphasized the strength of BERT and ALBERT in handling structured technical queries, reinforcing their potential for AI-driven support automation. However, these findings highlight the need to consider dataset expansion model fine-tuning improvements and real-world deployment challenges. Future work should focus on expanding real-world datasets, optimizing RoBERTa training, and integrating hybrid models to enhance chatbot interactions.**

**Keywords— Chatbots, BERT, ALBERT, Fine-Tuning, GPT, Technical support, Roberta.**

## I. INTRODUCTION

The users typically face technical issues, and the slow response times of the support staff frustrate them, further delaying the solutions to the problems. The situation is made worse as the answers provided by human help may not be accurate or practical at all. Chatbots in technical help improve user experience. These systems can respond to whatever questions the user has on time and effectively, therefore saving time and effort. Secondly, chatbots are available anytime, so customers can get help whenever they want without having to wait for human support. Also, by using artificial intelligence technology, chatbots could minimize human mistakes by offering precise answers derived from a large knowledge base. This improves customer satisfaction and the overall quality of the service by making the support process more dependable and efficient. On top of that, by managing routine questions and easy tasks, chatbots relieve the pressure on human support teams, freeing up staff members to focus on more complex issues that require human assistance. This not only enhances the effectiveness of the support staff but also lowers operating expenses for businesses, improving service effectiveness and giving users a more comprehensive support experience.

## II . PAPER ORGANIZATION

The structure of this paper is organized as follows:

The structure of this paper is organized as follows: Section **I** Introduction: Highlights the study's motivation, problem, and objectives. Section **II** Related work: Reviews prior studies and identifies research gaps. Section **III** Methodology: Explains the data collection, data preprocessing, model training, evaluation metrics, and Justification of choice. Section **IV** Results and Discussion: Analyzes model performance using metrics and graphical results Section **V** Conclusion and Future Work: Summarizes findings and suggests future improvements and challenges.

## III. RELATED WORK

AI applications have gotten to this point with the arrival of sentiment analysis and text classification models into technical support apps. According to R. Sudheesh et al. in 2023, the highly efficient BERT model was used for analyzing users' opinions toward Chat GPT. As a result of its excellent efficacy in both text classification and sentiment analysis, it achieved a high accuracy of approximately 96.49% [1]. With the assistance of model-based algorithms like LDA and BERT topic, it analyzed 21,515 tweets for common topics in user input. Positive comments focused on performance efficiency and usability, while criticism mostly focused on mistakes and excessive dependence on AI. These results demonstrate that BERT can categorize user intent and sentiment, potentially enhancing AI-driven customer support systems.

Nguyen's 2019 study went through the application of chatbot technology to improve technical support services and customer satisfaction. It compared the findings of using a chatbot with more conventional approaches like e-support and a FAQ area using the DeLone and McLean IS success model [2]. In brief, the study revealed that while chatbots might respond to customer queries more quickly, they barely improve the quality of data as compared to more conventional approaches. Chats were suggested to improve the user experience since human employees may share complex information about customers to provide support. Human assistance is needed in case of complex issues, even if the chatbot can respond to simple questions. This

means that a chatbot requires continued resource utilization to remain functional.

Maaroufa et al. (2024) established a hybrid approach for developing chatbots that integrate BERT and GPT-3.5 Turbo with MySQL and Python technologies. While datasets with 11 predefined categories and an additional class for unclassified queries were applied to improve contextual awareness and classification accuracy[3]. The classification of intentions was done by BERT. GPT-3.5 Turbo has indeed been helpful in practice, for it is more natural and sensitive to context due to its increase in domain-specific data. MySQL handles user information and interaction records to personalize answers. For development, the team used libraries in Python such as PyTorch and Flask; enabling flexible resources for model creation and deployment could also be done via cloud computing platforms like AWS.

The methodology for developing the AI-based chatbot system involves identifying the purpose with BERT, generating responses using GPT-3.5 Turbo, and retrieving user queries to fetch highly relevant answers with MySQL. Performance studies show that in such scenarios, interactions are transparent and responsive. PyTorch with Flask was used to establish the Python environment, while model training was done on AWS. Although BERT was trained with a large number of datasets on intent categorization, GPT-3.5 Turbo in turn needed only fine-tuning based on special cases for improvements in answers' quality. With these results, the average generating time for one answer being approximately 1.2 seconds and 98.4% accuracy for correct intent classification of the chatbot in a practical application could thus be made effective and dependable.[3]

## IV. METHODOLOGY

This section describes the methodology utilized to achieve the study's goals, including data collection, preprocessing methods, and model implementation.

### *A. Data Collection*

ChatGPT was utilized to generate the artificial dataset used in this study. The dataset consists of 2,000 rows, structured into two columns: questions and answers. It was designed to simulate real-world technical support interactions by generating diverse technical support pairs.

### *B. Data Preprocessing*

The preprocessing of data will ensure that raw data is well-structured for training and evaluation. The dataset is loaded into a Pandas DataFrame, where missing values and unwanted characters are removed to maintain consistency. In the BERT, ALBERT, and RoBERTa classification tasks, text labels are converted into numerical values using LabelEncoder, so that these models can process categorical data, while GPT-2 does not need this since it is not a classifier but a generative model. Finally, the cleaned dataset is split into 80% for training and 20% for validation by train_test_split, ensuring that the data is well-balanced for learning and testing.

Each model is treated separately, tokenizing words independently: BERT uses WordPiece, and ALBERT employs SentencePiece. Special tokens were added at the start, such as [CLS] and for separating sentences, [SEP], and also padding up to a maximum of 128 tokens. While RoBERTa uses Byte-Pair Encoding (BPE) and does not rely on [CLS] and [SEP] during training, GPT-2 processes text as a continuous sequence without classification labels or sentence separation. The main differences between RoBERTa and BERT and ALBERT are related to some key aspects of preprocessing, namely the use of BPE instead of WordPiece, the absence of NSP, and the use of Dynamic Masking instead of static masking, allowing generalization across different datasets.

Moreover, RoBERTa is trained on more extensive datasets than BERT; thus, it is more powerful in capturing complicated linguistic patterns. After tokenization is done, the data gets formatted into PyTorch, structured as a dataset where every sample includes input_ids and labels that are used for classification models only. The dataset is then loaded into DataLoaders, which will efficiently handle batches: a batch size of 16 for BERT, ALBERT, and RoBERTa is a good trade-off between memory consumption and performance, while 8 is used for GPT-2 due to its higher computational requirements. These preprocessing steps ensure the dataset is optimally prepared for training BERT, ALBERT, RoBERTa, and GPT-2, improving model accuracy and efficiency across classification and generative tasks.

### *C. Model Training(Fine-tunning)*

The pre-trained "bert-base-uncased" version of BERT is used for fine-tuning on the sequence classification task. Train model for 10 epochs with a batch size of 16, using AdamW optimizer with the learning rate 5e-5, include computing the loss and updating model parameters with backpropagation and gradient descent. ALBERT (A Lite BERT) is pre-trained with the "Albert-base-v2" version and fine-tuned for sequence classification. It is fine-tuned with the AdamW optimizer and a learning rate of 1e-5 for 10 epochs, with reduced model size and greater efficiency at the same level of performance as BERT. RoBERTa (Robustly Optimized BERT Pretraining Approach) is fine-tuned with the "Roberta-base" version for text classification tasks. It is trained for 10 epochs with batch size 16 and learning rate 2e-5 with AdamW optimizer. This approach performs better than BERT on a wide range of tasks by incorporating dynamic masking in training. The (Generative Pre-Trained Transformer) fine-tunes on the model "gpt2" for question answering. It trains for 3 epochs with a batch size of 8 and a learning rate of 2e-5, providing text sequences and updating model parameters to reduce loss on the train.

### *D. Model Evaluation*

The performance of the models was evaluated using accuracy, which measures overall correctness, and loss, which indicates error during training and validation. Precision, recall, and F1-score were used for class-wise performance analysis.

### *E. Justification of Model Choice*

The four transformer-based models—BERT, ALBERT, RoBERTa, and GPT-2—were selected due to their ability to offer a technical assistance chatbot that is effective, reliable, and responsive. Each model significantly increases the efficacy of the chatbot's perception and reaction to user inquiries. BERT was selected because of its extensive bidirectional understanding of text, which enhances response accuracy by allowing it to comprehend a phrase fully. ALBERT was selected due to its optimized architecture, which significantly reduces memory usage while maintaining high performance, making it more suitable for large-scale deployment. RoBERTa was included due to its enhanced training methods, which include dynamic masking and greater batch sizes, which enable it to comprehend complex technical inquiries better. GPT-2 was used to create human-like responds, allowing the chatbot to answer open-ended queries and engage in more realistic, conversational exchanges. The combination of these models enables a balance between structured classification and flexible answer generation, improving the chatbot's capacity to provide quick, accurate, and contextually relevant technical assistance.

## V. RESULTS

This section presents the performance evaluation of the four models. The results are analyzed based on training, validation, and testing metrics, including accuracy and loss. A detailed comparison of the models is provided to identify the optimal architecture.

| Model | Training Loss | Validation Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|
| BERT | 0.15 | 0.94 | 0.94 | 0.94 | 0.94 |
| ALBERT | 0.55 | 0.93 | 0.94 | 0.93 | 0.92 |
| ROBERTA | 3.82 | 0.03 | 0.00 | 0.03 | 0.00 |
| GPT-2 | 0.04 | 0.80 | 1 | 0.00 | 0.00 |

Table I Confusion Matrix

Based on the comparison on Table I, BERT is the best-performing model, achieving a very high accuracy of 94%, indicating excellent performance on the dataset. ALBERT also performed well with an accuracy of 93%, which is very close to BERT. The main difference is that ALBERT trains faster and requires fewer resources, making it a more efficient choice in certain cases. On the other hand, RoBERTa performed the worst, with an extremely low accuracy of 3%, suggesting that the model did not learn well from the data and was unable to make accurate predictions. GPT-2 achieved a moderate accuracy of 80%, which is decent but still lower than BERT and ALBERT in classification tasks. Overall, BERT is the best model due to its high accuracy and strong classification performance, followed by ALBERT as a more efficient alternative.

Given that Precision, Recall, and F1 Score are unsuitable for evaluating a chatbot that generates open-ended responses, such as GPT, they all scored 0. These metrics are meant for classification tasks when there is a single right answer to look up to. However, as these metrics suppose a single right answer, they cannot effectively measure the quality of the generated text, resulting in zero scores. This is because there may be many legitimate replies in text generation. It is more appropriate to employ BLEU, ROUGE, and METEOR, which assess text similarity, phrase overlap, and semantic correctness, rather than classifying responses as literally accurate or incorrect.

| Model | Benefits | Limitations |
|---|---|---|
| BERT | Strong contextual understanding and high accuracy (94%). | High computational cost and slower inference speed. |
| ALBERT | Smaller and more efficient than BERT with lower memory consumption. | Slightly lower accuracy (93%) and requires more training epochs. |
| ROBERTA | Robust training approach with better handling of longer sequences. | High computational requirements and needs fine-tuning on large datasets. |
| GPT | Strong generative capabilities for detailed and context-aware responses. | Lower validation accuracy (80%) and may generate irrelevant responses. |

Table II Benefits and limitations

Table II shows the benefit and the lemation for each model.


Fig. 1. BERT

From Fig. 1, BERT answered correctly.


Fig. 2. ALBERT

From Fig. 2, BERT answered correctly.

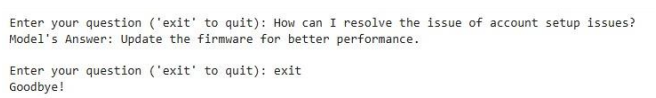


Fig. 3. ROBERTA

From Fig. 3, the model didn't answer correctly.

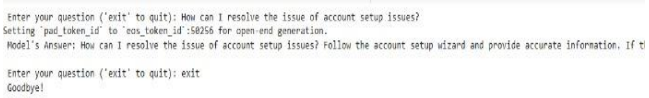


Fig. 4. GPT

From Fig. 4, the model answer correctly.

## VI. Conclusion and Future Work

The efficiency of BERT, ALBERT, RoBERTa, and GPT-2 in improving a technical support chatbot's performance was addressed in this study. According to the findings, BERT was the most accurate model for classifying technical questions, with an accuracy of 94%. With a 93% accuracy rate, ALBERT showed strong performance, providing a more effective option because of its faster training pace and less memory usage. Despite having excellent generating skills, GPT-2 had trouble classifying data; it achieved 80% accuracy but frequently generated irrelevant results. RoBERTa, on the other hand, did the worst, achieving an accuracy of just 3%, which suggests that it learned from the data poorly and was unable to provide reliable predictions.

A primary obstacle encountered was gathering information from various sources and confirming its quality to guarantee quality and reliability. We had to change the dataset size three times, first to 8,000 samples, then to 4,000, and finally to 2,000, which turned out to be suitable for the chosen models. Data size and capacity were also major obstacles. This made it challenging to balance data availability with model efficiency, especially for RoBERTa, which required larger sets of data to train well.

Future work should focus on growing the dataset with real technical help queries, improving RoBERTa's training with bigger datasets, and creating a combination approach that combines *GPT-2 for answer generation and BERT for classification* in order to boost the chatbot's performance. Exploring lightweight models like *DistilBERT* may also assist in maintaining accuracy while cutting down on computational costs. Additionally, we plan to enhance our evaluation methods by incorporating *BLEU, ROUGE, and METEOR* metrics, which will allow for a more effective assessment of the chatbot's response quality, text similarity, and semantic correctness. The chatbot's responses and overall efficacy will be further improved by implementing it in a real-world environment and gathering user input.

## References


[1] R. Sudheesh, M. Mujahid, F. Rustam, R. Shafique, V. Chunduri, M. G. Villar, J. B. Ballester, I. de la Torre Diez, and I. Ashraf, "Analyzing Sentiments Regarding ChatGPT Using Novel BERT: A Machine Learning Approach," *Information*, vol. 14, no. 474, Aug. 2023.

[2] T. S. Nguyen, "Potential Effects of Chatbot Technology on Customer Support: A Case Study," Master's thesis, Aalto University, Apr. 2019.

[3] O. Maaroufa, A. Maaroufa, M. Biniz, and R. El Ayachi, "Development of a Customer Service Chatbot Using BERT and GPT-3.5 Turbo: A Hybrid Approach," Preprint, SSRN, 2024