iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71007

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

BERT Enabled AI Chatbot for Enhanced Voice Interaction

Mrs.Elakia k[1], Dr.R.G.Suresh Kumar[2], Ms.Aksharapavithran[3], Ms.Sreenandana T M[4], Ms.Vismaya A K[5]
Computer Science and Engineering, RGCET, Pondicherry University

Abstract: AI is a set of technologies that enables computers to perform a variety of advanced functions, including the spoken and written language, analyse data, make suggestion. The existing system developed a chatbot that support quality education to give prospective user's with accurate information about universities and their unique courses. The chatbot give accurate information that can be available in any time. Traditional chatbot, which can affect in inaccuracies in information. Students face confusion due to disagreement in information Universities frequently fail to instantly answer user's queries. System cannot support voice interactions. To overcome these limitations, we utilizes Bidirectional Encoder Representations from Transformer (BERT), a more advanced technology that significantly improves accuracy. Also, the chatbot is integrated into a mobile operation built with React Native, enabling both text and voice. The voice feature enhances accessibility, offering a more accessible and user-friendly experience. This upgraded chatbot system ensures that user can receive reliable and precise information.
Key words: Artificial Intelligence ,NLP, chatbot, machine learning ,BERT , Reactive Native , voice interaction.

I. INTRODUCTION

In the digital age, educational institutions face numerous challenges in providing efficient services to students, faculty, staff, administrators, parents and other users. To address these challenges, chatbots have emerged as a viable solution, offering personalized assistance and streamlining communication. Chatbot have become essential tools for engaging people in the era of rising adoption of artificial intelligence(AI). This document outlines the chatbot needs of various users in educational institutions, highlighting the benefits of implementing AI-powered chatbot solutions. With the increasing prevalence of online learning and the diverse needs of modern students, the demand for efficient and responsive student services has never been greater. The integration of chatbots in student services has indeed gained significant traction in recent years. Chatbots have emerged as essential AI-powered tools that facilitate user interactions through web browsers, allowing individuals to ask questions on various topics and receive immediate responses. These conversational agents harness the capabilities of Artificial Intelligence (AI) and Natural Language Processing (NLP) [1] to deliver accurate answers, drawing from a predefined knowledge base. By processing and understanding human language, chatbots can interpret user queries effectively and provide relevant information or assistance. There are three primary types of chatbots: Rule-based, Retrieval based, and Generative-based models. Rule-based chatbots function according to a set of predefined rules [2]. They can only respond to a limited range of queries and typically rely on straightforward input patterns. This restricts their ability toengage in complex conversations, making them suitable for basic question-and-answer scenarios where user needs are straightforward. In contrast, Retrieval based chatbots utilize a more advanced approach by selecting the most appropriate response from a curated set of predefined answers. These chatbots are capable of understanding the context of a conversation, allowing for a more interactive experience. They can handle variations in user queries by matching input with corresponding responses, which enhances their effectiveness in real-world applications. The most sophisticated type is the Generative-based chatbot, which generates responses based on previous interactions and learned data patterns. These chatbots require complex computational models and extensive training data to operate effectively.

II. METHODOLOGIES AND APPROACHES

The research work on "Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis"[1] employs a multi-faceted methodology to develop and evaluate chatbots designed to provide information about technical universities. Initially, the study begins with a comprehensive literature review to identify existing NLP frameworks and their applications in educational contexts. The development phase involves selecting and finetuning various advanced NLP models, such as BERT, GPT, and other transformer-based architectures, to enhance understanding and generation of university-related queries. Data collection is crucial; the researchers gather diverse datasets from university websites, student forums, and FAQs to ensure a rich training corpus.

Subsequently, the work outlines the implementation of rigorous evaluation metrics, including precision, recall, F1score, and user satisfaction surveys, to compare the performance of each model. A/B testing is also utilized to assess real-time interactions with users. Finally, the study concludes with a comparative analysis of the models, highlighting strengths and weaknesses based on the metrics collected, and provides recommendations for future improvements in chatbot systems for educational institutions. This systematic approach ensures a thorough examination of both the technical and user experience aspects of the chatbots.

1) Retrieval-based: approach involves selecting the best response from a predefined database based on user input, using techniques like keyword matching, semantic similarity, and machine learning algorithms to rank and choose the most contextually relevant response. On the other hand
2) Generative-based: Thisapproach creates the responses dynamically, employing models such as sequence-to-sequence architectures or transformer-based models. This allows generative models to generate responses based on patterns learned from training data, offering flexibility and creativity beyond fixed replies. The study compares these methods, noting retrieval-based chatbots' strength in providing accurate and contextually relevant responses, while generative chatbots are praised for their adaptability and ability to personalize interactions[2].
3) Machine Learning Techniques: This involves using algorithms that learn from data. Models are trained on large datasets to decipher user intents and generate appropriate responses. This includes employing algorithms like decision trees, support vector machines, or neural networks, which can adapt to new inputs based on the training they receive.
4) Lexicon-Based Techniques: This method uses predefined rules and dictionaries to process user inputs. It relies on keyword matching and pattern recognition to generate responses, using a fixed lexicon of terms and phrases to understand and react to what the user says. The study compares the effectiveness, accuracy, and user satisfaction of these methodologies, evaluating each for their strengths and weaknesses in different chatbot applications[5].

III. LITERATURE OVERVIEW

A study byGirijaAttigeri, (member, IEEE), Sucheta v. Kolekar, (member, IEEE) and Ankit Agrawal, emphasized a chatbot comparison “Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis” explore the implementation of chatbots for university information dissemination, comparing five NLP models. Neural network-based models, particularly sequential modeling, demonstrated higher accuracy and effectiveness in providing consistent, real-time responses compared to TFIDF and pattern matching approaches[1]. “A comparative study of retrieval-based and generative-based chatbots using Deep Learning and Machine Learning “(2023) by Sumit Pandey, Srishti Sharma[2] compares retrieval based and generative-based chatbots for health-related applications, finding that generative based chatbots with encoder–decoder designs achieve 94.45% accuracy, outperforming retrieval-based chatbots like Bi-LSTM at 91.57%. Generative models excel in creating new text, while retrieval models rely on preexisting responses. ”Enough of the chit-chat: A comparative analysis of four AI chatbots for calculus and statistics” (2023) analyzed by David SantandreuCalongeA, Linda Smail ,Firuz KamalovC that compares ChatGPT, GPT-4, Bard, and LLaMA for their potential in mathematics and statistics education. The study finds GPT-4 excels in calculus and statistics learning compared to the other chatbots and suggests that AI chatbots can significantly enhance higher education [3]**.**

The study of “Comparative analysis of various chatbot framework”. (2023)[4] developed by Nachiket Kapure describe three chatbot models using different frameworks to evaluate their features and effectiveness. The analysis table helps users select the most suitable chatbot framework for enhancing customer service and automating tasks. A comparative analysis byKarthik Konar researched **“**A Comparative Study on Chatbot Based on Machine Learning and Lexicon Based technique”[5](2020).they compares lexicon-based and machine learning approaches for sentiment analysis in chatbots, using Python to develop two chatbots: one for classifying movie reviews and another (DocBot) for providing information on kidney disease. The study aims to determine which approach delivers more accurate results for chatbot implementations.

An Evaluation of General-Purpose AI Chatbots: A Comprehensive Comparative Analysis (2024)[6] analyse an in-depth evaluation of eight leading AI chatbots, using confusion matrices and pairwise comparisons across eight criteria to determine their efficiency. The research offers valuable insights and recommendations for developers and users, guiding them towards improving chatbot performance and ensuring they meet evolving needs and preferences in the AI industry. Another study that compares chatbots of -
“COMPARATIVE ANALYSIS OF CHATBOTS(2020)” [7] researched by Shivang Verma , Lakshay Sahni ,Moolchand Sharma evaluates and compares the accuracy of eight chatbots: they are Rose,GoogleAssistant,Siri, Machine Comprehension Chatbot, Mitsuku, Jabberwacky, ALICE, and Eliza—based on their responses to predefined questions. The analysis covers three main parameters: factual accuracy, conversational attributes, and handling of exceptional queries, leading to a ranked performance assessment of each chatbot.

"A comparative study of medical chatbots" [8] by Jitendra Chaudhary, Vaibhav Joshi, Atharv Khare, Rahul gawali, Asmita Manna proposed HEALTHBOT aims to streamline medical interactions by managing patient symptoms, test reports, and basic prescriptions in English and Marati. This chatbot will assist medical practitioners and enhance efficiency by reducing administrative and bridging the treatment gap (refer table 1).

IV. FINDINGS AND TRENDS

The findings and trends in "Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis" reveal significant advancements in the deployment of natural language processing (NLP) technologies within educational contexts. One notable trend is the increasing adoption of transformer-based models, such as BERT and GPT, which have demonstrated superior capabilities in understanding and generating human-like responses. BERT's bidirectional context comprehension allows chatbots to interpret user queries more accurately, thereby enhancing the relevance of the information provided. In contrast, GPT models excel in generating fluent and coherent text, fostering engaging interactions that can improve user satisfaction.

V. LITERATURE COMPARISON TABLE

Sl.no	Title	Technique used	Algorithm
01	Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis.	Conversational AI, Natural language processing , neural networks ,sequentialmodelling ,pattern matching , semantic analysis.	Neural network
02	A comparative study of retrieval-based and generative-based chatbots using Deep Learning and Machine Learning.	Deep learning, Machine learning, vanilla RNN, CNN, Bidirectional LSTM, GRU	Neural network
03	Enough of the chit-chat: A comparative analysis of four AI chatbots for calculus and statistics .	Natural Language Processing, Statistics, calculus, Bard, LLAMA,	Neural network
04	Comparative analysis of various chatbot framework.	NLP, chatbot frameworks, API, Webhook, AI,	Natural language processing
b 05	A Comparative Study on Chatbot Based on Machine Learning and Lexicon Based Technique	Chatbot , Lexicon, Machine learning ,Polarity, Subjectivity, Tokenization.	Natural language processing

Table 1

Another significant finding is the integration of reinforcement learning techniques, which allow chatbots to adapt and improve over time based on user interactions. This dynamic learning process enhances the chatbot's ability to address a wider range of queries effectively. Additionally, ethical considerations have gained prominence, with researchers emphasizing the need for fairness and transparency in chatbot responses, ensuring that diverse student populations are adequately represented.

Overall, these findings underscore a shift toward creating intelligent, user-centered chatbots that not only deliver accurate information but also contribute positively to the educational experience.

1) Model Performance: Evaluated different NLP models for their efficiency in processing and responding to queries specific to university information.
2) User Interaction: Found that advanced NLP models significantly enhance user experience by delivering more precise and context-aware responses.
3) Comparative Analysis: In this method, conducted a comparative evaluation, noting the relative strengths and weaknesses of various models in different applications.
4) Implementation Challenges: Here this method discussed challenges, such as ensuring data quality and integrating chatbots seamlessly with existing university information systems.
5) User Satisfaction: Users expressed higher satisfaction when using chatbots driven by sophisticated NLP models compared to conventional systems.
6) Future Research Directions: Identified future research avenues, including expanding model training datasets and improving support for multiple languages.

VI. CHALLENGES AND GAPS

Students encounter significant challenges when navigating the myriad of information sources available about universities, as discrepancies across websites, rankings, and brochures often lead to confusion. This lack of consistent information can hinder their ability to make informed decisions regarding their education. Furthermore, universities frequently struggle to respond promptly to student inquiries, which exacerbates this uncertainty and diminishes the overall student experience. Additionally, existing systems inadequately protect sensitive data, leaving it vulnerable to unauthorized access and tampering, raising concerns about privacy and security. Compounding these issues is the limited support for voice interaction in current chatbot implementations, which restricts accessibility and convenience for users who prefer or require voice-based communication. These challenges highlight significant gaps in the current landscape of university information dissemination, indicating the need for more robust and secure solutions that prioritize timely responses and user-friendly interfaces.

VII. IMPLEMENTATION DETAILS

1) *Input Layer / Data Collection:*

The Input Layer serves as the foundation of the chatbot system, handling both voice and text inputs from users. Voice queries are processed through tools like Google Speech-to-Text API or OpenAI's Whisper, which accurately convert spoken language into transcribed text. This conversion step is critical for ensuring that downstream NLP modules receive clean textual data, regardless of the input modality. Simultaneously, a web-based interface possibly built using stream-lit or Flask allows users to directly type their questions or requests, supporting accessibility and flexibility.

To support training and testing of the chatbot, the system integrates with publicly available conversational datasets. These datasets are typically sourced from Kaggle, which hosts a wide variety of labeled intent-response data collections (e.g., the "Intents JSON Dataset" or "Chatbot QnA Dataset"). Files are downloaded in CSV or JSON format and are structured to include intents, utterances, and sample responses. The goal is to ensure that the training data reflects the types of interactions the chatbot will encounter in real-world settings.

In practice, the data collection module includes scripts to automate the downloading and parsing of these datasets into formats compatible with machine learning pipelines. Once acquired, the datasets are visualized using tools like Pandas and Seaborn to inspect intent distribution and class balance.

2) *Pre-processing/NLP Processing Pipeline:*

This module transforms raw textual data into clean, usable input for the model. The process starts with noise removal, eliminating unnecessary punctuation, emojis, HTML tags, and nonASCII characters. Such artifacts can distort the meaning of queries and introduce inconsistencies in model training. This cleanup is especially important for voice inputs, which may include filler words or transcription errors. The cleaned text is then normalized by converting it to lowercase and removing repeated characters or contractions.

Tokenization follows, where sentences are split into tokens using tools like NLTK, spaCy, or tokenizers from the Hugging Face library. This helps break down complex queries into manageable units.

Next, stop words—commonly used but low-value words—are removed to retain only semantically rich tokens. In some designs, stemming or lemmatization may be added to reduce words to their base forms, which helps reduce vocabulary size and improve generalization.

3) Feature Extraction:

Once preprocessing is complete, the next critical step is to convert text into numerical representations the model can understand. This module uses BERT (Bidirectional Encoder Representations from Transformers) for feature extraction, leveraging its ability to capture deep semantic and contextual relationships in language. Each sentence or query is passed through a pretrained BERT model from Hugging Face's Transformers library, producing contextualized embeddings high-dimensional vectors that represent word meanings in context.

Unlike traditional word embeddings like Word2Vec or GloVe, BERT dynamically generates different vector representations depending on surrounding words, which greatly enhances the chatbot's understanding of user intent. The [CLS] token's embedding is typically extracted to represent the entire sentence during classification. These vectors are used as input to subsequent layers for intent classification or response generation. In terms of implementation, the Hugging Face Transformers library allows easy integration of pre-trained BERT models such as BERT-base-uncased.

4) Model Creation Using Bert:

This is the heart of the chatbot, where a fine-tuned BERT model is trained to classify intents or map inputs to response categories. Fine-tuning involves initializing BERT with pre-trained weights and then training it on the collected conversational dataset using supervised learning. The dataset should be split into training, validation, and test sets to monitor generalization. During training, only the final layers of BERT are often modified, while earlier layers retain pre-trained weights.

The model architecture typically consists of BERT's transformer layers followed by a dropout layer and a fully connected layer (often with soft-max) for classification. Metrics like accuracy, precision, recall, and F1-score are tracked during training using tools such as Tensor Board or Matplotlib plots. Cross-entropy loss is minimized using optimizers like Adam or AdamW, with early stopping or learning rate decay applied for better performance.

5) Validation Module:

Before proceeding to response generation, each user query is checked in this validation layer to ensure it is appropriate, complete, and falls within the trained domain. For example, if a user inputs a vague or malformed query (like "uhh..." or "asdf"), the system should prompt the user for clarification. Similarly, if the model's confidence in classifying an intent falls below a threshold (e.g., 60%), a fallback response or rephrasing prompt is triggered.

This is implemented using simple conditionals, regular expressions (for syntax checks), and soft-max confidence scores from the model. For instance, if the top two intent probabilities are close together, the chatbot might ask, "Did you mean X or Y?" This improves user trust and prevents invalid or misleading responses. Additionally, this module can route queries outside the trained domain to a secondary AI model or provide a generic "I'm not sure I understand" response.

6) Response Module:

Once validated, the system proceeds to generate a suitable response. This can be done in two main ways: through a response mapping system, where each intent is linked to a predefined reply, or via dynamic response generation, where transformer-based models (e.g., GPT-2, T5) generate context-aware replies. For real-time applications, static responses are typically faster and more reliable, while generative models offer more flexibility and personality.

The output text is then passed to a Text-to-Speech (TTS) engine such as Google TTS or pyttsx3, which converts it into audio that is streamed back to the user. This step enhances accessibility and user engagement, especially in voice-only environments. TTS can also adjust voice pitch, speed, and language, making the chatbot feel more personalized.The final chatbot response both in text and voice is displayed on the UI frontend. In platforms like Streamlit, this may include a chat bubble with the response and a playback button for audio.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

VIII. RESULT AND DISCUSSION

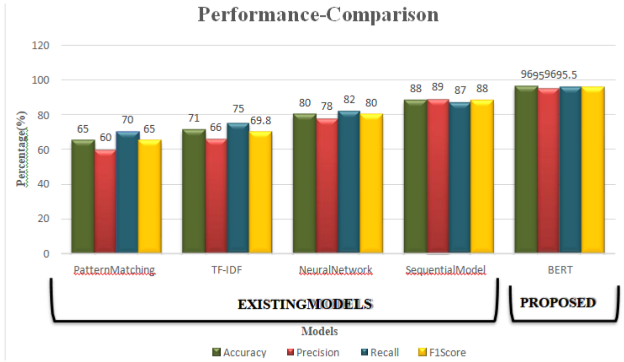


Fig: BERT Model Performance Metric

Across all metrics—accuracy, precision, recall, and F1 score—BERT delivers the strongest performance, achieving 96 % accuracy and recall, 95 % precision, and a 95.5 % F1 score. By comparison, the sequential model is the next best, with an accuracy of 88 %, precision of 89 %, recall of 87 %, and F1 of 88 %. Traditional neural networks follow, posting solid but lower results (80 % accuracy, 78 % precision, 82 % recall, and an 80 % F1 score), while TF-IDF and pattern-matching approaches lag behind—TF-IDF scoring in the low 70s across metrics and pattern matching in the mid-60s. This clear gradient highlights how context-aware, transformer-based embedding (BERT) far outstrips simpler statistical or rule-based techniques, delivering both more reliable intent detection and more balanced precision-recall trade-offs for conversational AI.

The proposed system successfully enhances information retrieval and user interaction through the integration of BERT for natural language understanding and React Native for cross-platform accessibility. The BERT model significantly improves response accuracy by understanding the context of user queries, ensuring that students receive relevant and precise answers. Additionally, the system’s voice access feature enhances usability, making it more inclusive for users who may face difficulties with traditional text input. The automated query response mechanism minimizes response time, offering real-time assistance to students.

Moreover,the React Native-based mobile-application provides a seamless and efficient user experience across Android and iOS devices. The ability to maintain a single codebase reduces development and maintenance efforts while ensuring optimal app performance. The system's combination of text-based and voice-enabled queries ensures that students can access information conveniently, fostering better engagement and decision-making. Overall, the results indicate that the proposed system provides a highly responsive, accessible, and efficient solution for educational support.

A. Input Embedding Layer:

The Input Embedding Layer in BERT is responsible for converting raw text into numerical representations that the model can process. Unlike traditional word embeddings that assign a single fixed vector to each word, BERT's embedding layer captures additional contextual information. It consists of three components: Token Embeddings, Segment Embeddings, and Positional Embeddings.

$$E_i = T_i + S_i + P_i$$

Where:

$E_i = Final\ embedding\ for\ token\ i$

$T_i = Token\ embedding\ of\ i$

$S_i = Segment\ embedding$

$P_i = Positional\ embedding$

Token Embeddings represent individual words or subwords using WordPiece tokenization, allowing the model to handle out-of-vocabulary words. Segment Embeddings distinguish between different sentences in input sequences, which is crucial for tasks like question-answering. Positional Embeddings provide information about word order, ensuring that the model understands the sequence of words in a sentence.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

B. Multi-Head Self-Attention Layer:

The **Multi-Head Self-Attention Layer** in BERT allows the model to capture relationships between words regardless of their position in a sentence. Unlike traditional sequence models, which process text sequentially, self-attention enables BERT to analyse all words in parallel, improving efficiency and context understanding. Each token attends to every other token in the sentence, determining how much importance should be given to different words using attention scores. This mechanism is computed as:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where **Q (Query), K (Key), and V (Value)** are linear transformations of the input embeddings. **Multi-head attention** extends this by computing multiple attention mechanisms in parallel, capturing diverse contextual meanings. The outputs of these attention heads are concatenated and transformed,allowing BERT to understand complex dependencies across different parts of a sentence. This makes BERT highly effective in tasks requiring deep contextual comprehension.

C. Feedforward Layer

The Feedforward Layer in BERT is responsible for transforming the output from the Multi-Head Self-Attention Layer and adding non-linearity to the model. It consists of two fully connected layers with an activation function in between. The first layer applies a linear transformation using learned weights, followed by a ReLU (Rectified Linear Unit) activation function to introduce non-linearity. The second layer then projects the transformed representation back to the original dimension. This process is mathematically represented as:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b2$$

Where ***W1*, W2** are weight matrices, and ***b1*, b2** are bias terms. This component allows BERT to learn complex representations by applying transformations beyond attention-based computations. The feedforward layer operates independently on each token, ensuring that the model captures both local and global patterns efficiently before passing data to the next transformer block.

D. Output Layer:

The Output Layer in BERT is the final stage that processes the transformed representations from the previous layers to generate task-specific predictions. Depending on the application (e.g., classification, question-answering, or text generation), this layer varies in structure. Typically, it consists of a fully connected (dense) layer followed by an activation function such as soft-max for classification tasks or sigmoid for binary outputs. Mathematically, it is represented as:

$$y = Activation(XW + b)$$

where *X* is the input from the last transformer layer, *W* is the weight matrix, and *b*is the bias term. For classification tasks, the soft-max function ensures that the output represents probability distributions over different classes. In **masked language modelling(MLM)**, the output layer predicts masked words based on learned contextual relationships. This final step allows BERT to provide meaningful predictions tailored to specific NLP tasks.

E. Precision

Precision is a performance metric used to assess the accuracy of a classification model, particularly focusing on how many of the instances that the model predicted as positive are actually true positives. It is especially useful when the cost of false positives is high, such as in medical diagnoses where misclassifying a healthy person as sick can lead to unnecessary treatments or interventions. Precision is calculated using the formula:

$$Precision = \frac{True\ Positives(TP)}{True\ Positives(TP) + False\ Positives(FP)}$$

Here, True Positives (TP) refers to the number of correctly predicted positive instances, while False Positives (FP) represents the number of instances that were incorrectly classified as positive when they were actually negative. A high precision value, close to 1 or 100%, indicates that the model is making very few errors in predicting positive cases. Conversely, low precision suggests that the model is frequently misclassifying negative cases as positive, which can lead to costly mistakes, particularly in sensitive areas like healthcare or fraud detection. Thus, precision helps in evaluating the reliability of a model when the focus is on minimizing false positive predictions.

F. Recall:

Recall, also known as sensitivity or true positive rate, is a metric used to evaluate how well a classification model identifies positive instances. It measures the proportion of actual positive instances that were correctly identified by the model.

In the context of the proposed system, recall is crucial for ensuring that the system doesn't miss important queries or information requests from students. For example, if a student asks a question, high recall means the system is effective at recognizing and providing an answer, reducing the chances of missing out on crucial inquiries that could affect a student's decision-making process.In the proposed system, recall is important because students may ask a wide variety of questions, and the system needs to correctly identify and respond to as many of these inquiries as possible. A high recall ensures that the application is thorough in addressing user needs, especially in educational settings where missing an important query could lead to confusion or incomplete understanding. Recall is calculated using the formula:

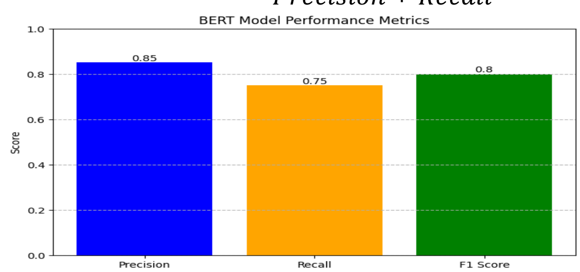
$$Recall = \frac{True\ Positives(TP)}{True\ Positives(TP) + False\ Negatives(FN)}$$

Where **True Positives (TP)** are the correctly identified positive cases, and **False Negatives (FN)** are the actual positive cases that were missed by the model. A higher recall indicates that the system is less likely to overlook relevant user queries, ensuring that students receive comprehensive and timely responses.

G. F1-score:

The F1 score is a crucial metric that combines both precision and recall into a single value, providing a balanced measure of a model's performance, especially when there is an uneven class distribution. In the proposed system, the F1 score is particularly important as it ensures that the application not only provides accurate answers (precision) but also effectively identifies all relevant queries from students (recall). By considering both precision and recall, the F1 score helps in evaluating the system's overall effectiveness in responding to student inquiries, balancing the trade-off between false positives and false negatives. A high F1 score indicates that the system is both precise in its responses and thorough in addressing as many queries as possible.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$



This formula ensures that both precision and recall contribute equally to the final score. In the context of the proposed system, a higher F1 score means that the application is effectively balancing accuracy and completeness in answering student queries, providing a more reliable and efficient support platform. A low F1 score, on the other hand, would indicate that the system is either too focused on accuracy but misses many relevant queries (low recall), or it casts a wide net and incorrectly classifies many responses as relevant (low precision), both of which could hinder the user experience.

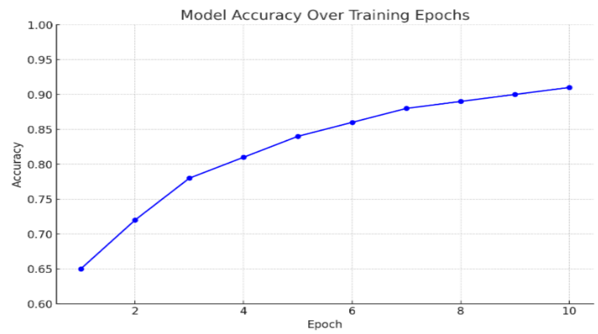
H. Accuracy

Accuracy is a key metric for assessing the overall performance of a classification model, as it calculates the ratio of correct predictions (both true positives and true negatives) to total predictions made by the model.It measures the proportion of correct predictions (both positive and negative) made by the model out of all the predictions it makes. In the context of the proposed system, accuracy reflects how well the model is performing in providing correct responses to student queries. A higher accuracy indicates that the system is generally effective in delivering the right answers, whether it is identifying specific course details, institutional information, or answering other queries accurately. However, while accuracy is useful, it might not always be sufficient in situations where the classes are imbalanced, as it doesn't distinguish between types of errors (false positives or false negatives).

$$Accuracy = \frac{True\ Positives(TP) + True\ Negatives(TN)}{True\ Positives(TP) + True\ Negatives(TN) + False\ Positives(FP) + False\ Negatives(FN)}$$

Where:

TP: Correctly predicted positive instances.
TN: Incorrectly predicted negative when they are negative.
FP: Incorrectly predicted as possible when they are negative
FN: Incorrectly predicted as a negative when they are positive.



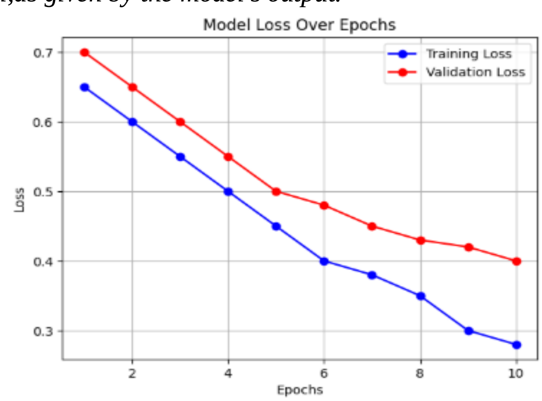
In the proposed system, a high accuracy indicates that the model is consistently making the correct predictions for the majority of queries, providing a reliable platform for students. However, it is essential to monitor other metrics like precision, recall, and F1 score, especially if the system needs to handle a wide variety of queries, some of which may be more critical than others.

I. Loss

The loss function is a critical component in any machine learning model, as it quantifies how far off a model's predictions are from the actual outcomes. In the context of the proposed system, the loss function helps the model assess the error when it provides incorrect answers to student queries. By minimizing this error during training, the model improves its ability to respond accurately and effectively to user input. A well-chosen loss function ensures that the model learns to prioritize the most important aspects of the query while adjusting its predictions to become more accurate over time.

$$Loss = -\sum_{i=1}^{c} \quad y_i log(\widehat{y_i})$$

where:
ŷiis the predicted portability for class i.
C is the number of classes(Eg: different types of queries like course info, faculty details, etc).
Yi is the predicted probability for class i,as given by the model's output.



This formula calculates the logarithmic loss for each class and sums it across all classes. The log function penalizes predictions that are further from the actual labels, meaning the model is more strongly corrected when it makes larger errors. By using categorical cross-entropy, the model can learn to make better predictions by minimizing the error between predicted and actual query classes. As the model trains, the loss value will decrease, indicating better alignment between predictions and real-world outcomes. This leads to an improved user experience where students receive more accurate and relevant information.

IX. CONCLUSION

In conclusion, the exploration of advanced NLP models for technical university information chatbots reveals a promising pathway toward enhancing student engagement and decision-making processes. The comparative analysis of various NLP architectures, such as BERT and GPT, underscores the strengths and limitations of each model in understanding and generating contextually relevant responses. While transformer-based models demonstrate significant improvements in accuracy and user experience, the integration of user feedback and engagement metrics is crucial for developing effective chatbot solutions. Additionally, addressing challenges such as data security and the need for voice interaction capabilities remains essential for creating more inclusive and user-friendly systems.

The findings of this survey highlight the necessity for a holistic approach that combines technical performance with ethical considerations, ensuring fairness and transparency in chatbot interactions. As universities continue to face challenges related to inconsistent information and delayed responses, the implementation of advanced NLP-driven chatbots could bridge these gaps, providing students with reliable support and information. Future research should focus on refining these models and incorporating adaptive learning mechanisms to enhance their responsiveness over time. Ultimately, the advancement of NLP technologies in educational chatbots holds the potential to significantly improve the overall student experience, fostering informed decision-making and engagement in the academic journey.

REFERENCES

[1] Girijaattigeri , (member, ieee), ankitagrawal, and sucheta v. Kolekar , (member, ieee) “advanced nlp models for technical universityinformationchatbots:development and comparative analysis” ,department of information and communication technology, digital object identifier 10.1109/access.2024.3368382,volume12,2024,https://creativecommons.org/licenses/bync-nd/4.0/

[2] Sumitpandey, srishtisharma “a comparative study of retrieval-based and generative-based chatbots using deep learning and machine learning”.the north cap university school of engineering & technology, gurugram, haryana, 122017,india ,https://doi.org/10.1016/j.health.2023.100198

[3] Davidsantandreucalonge, lindasmail, firuzkamalov,”enough of the chitchat: a comparative analysis of four ai chatbots for calculus and statistics”, vol.6 no.2(2023), issn : 2591-801x http://journals.sfu.ca/jalt/index.php/jalt/index

[4] Nachiketkapure “comparative analysis of various chatbot frameworks”, department of computer science & engineering, birla college of arts, science & commerce, kalyan, india.volume:04/issue:09/september-2022, e-issn: 2582-5208, www.irjmets.com

[5] Karthikkonar. Mca, nmimsmukeshpatel“a comparative study on chatbot based on machine learning and lexicon basedtechnique”,school of technology management & engineering, vileparle(west) mumbai. Volume 5, issue 5, may – 2020,issnno:-2456-2165. Www.ijisrt.com

[6] Oleksiichalyi, “an evaluation of generalpurpose ai chatbots:acomprehensive comparative analysis”,faculty of informatics, vytautas magnus university, kaunas, 44404, lithuani, https://www.isjtrend.com/

[7] Shivangverma, lakshaysahni, moolchandsharma, ”comparative analysis of chatbots”, department of computer science and engineering, maharaja agrasen institute of technology, ggsipu, delhi, india, department of electrical and electronics engineering, delhi technological university,delhi,india(icicc2020), https://ssrn.com/abstract=3563674

[8] Jitendrachaudhary,vaibhavjoshi,atharvkhare,rahulgawali,asmita manna, “a comparative study of medical chatbots”, volume: 08 issue: 02 | feb2021 , e-issn: 2395-0056, www.irjet.net

[9] A. Chandan, m. Chattopadhyay, and s. Sahoo, ‘‘implementing chat-bot in educational institutes,’’ ijrar j., vol. 6, no. 2, pp. 44–47, 2019.

[10] T.lalwani,s.bhalotia, a.pal, v.rathod, and s.bisen, ‘‘implementation of a chatbot system using ai and nlp,’’ int. J. Innov. Res. Comput. Sci. Technol. (ijircst), vol. 6, no. 3, pp. 26–30, 2018.

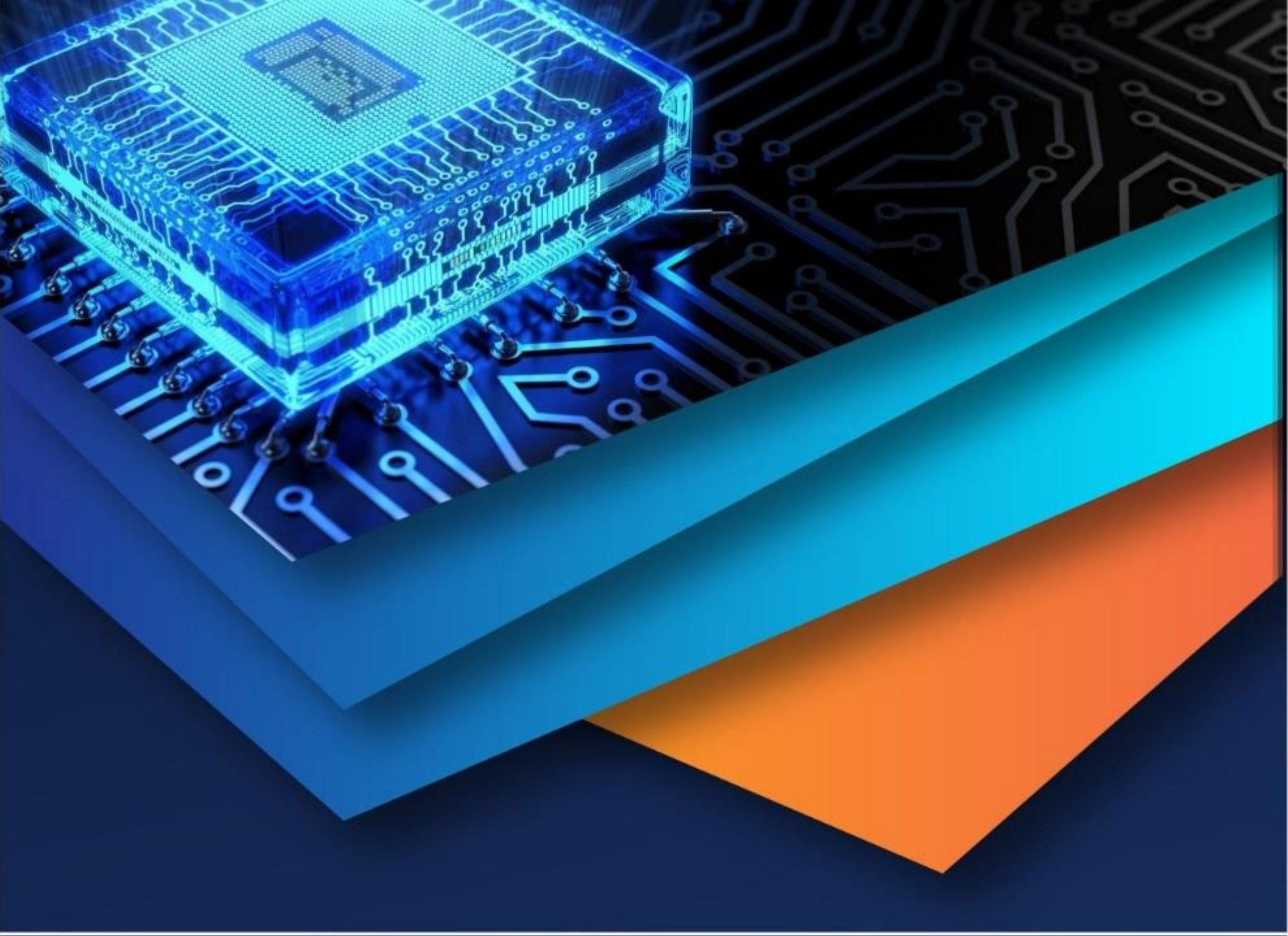
[11] J. Thukrul, A. Srivastava, and G. Thakkar, ‘‘Doctorbot—An informative and interactive chatbot for COVID-19,’’ Int. Res. J. Eng. Technol. (IRJET), vol. 7, no. 7, pp. 3033–3036, 2020.

[12] S. Maher, ‘‘Chatbots & its techniques using AI: A review,’’ Int. J. Res. Appl. Sci. Eng. Technol., vol. 8, no. 12, pp. 503–508, Dec. 2020.

[13] M. Aleedy, H. Shaiba, and M. Bezbradica, ‘‘Generating and analyzing chatbot responses using natural language processing,’’ Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 9, 2019.

[14] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, ‘‘Stanza: A Python naturallanguageprocessingtoolkit for many human languages,’’ 2020, arXiv:2003.07082.

[15] M. M. H. Dihyat and J. Hough, ‘‘Can rule-based chatbots outperform neural models without pre-training in small data situations? A preliminary comparison of AIML and Seq2Seq,’’ in Proc. 25th Workshop Semantics Pragmatics Dialogue, 2021, pp. 22–26.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY