BOHR International Journal of Smart Computing and Information Technology
2024, Vol. 5, No. 1, pp. 1–11
DOI: 10.54646/bijscit.2024.39
www.bohrpub.com

RESEARCH

Bridging the gap in college information access through natural language processing powered Lucy chatbot

Rupesh Gelal, Suraj Karki, Saman Shrestha and Krishna Bikram Shah[*†]

Department of Computer Science and Engineering, Nepal Engineering College, Affiliated to Pokhara University, Bhaktapur, Nepal

[*]**Correspondence:**
Krishna Bikram Shah,
krishnabs@nec.edu.np

[†]**ORCID:**
Krishna Bikram Shah
0000-0003-1763-511X

Received: 26 March 2024; **Accepted:** 11 June 2024; **Published:** 04 September 2024

In recent years, there has been a notable shift in how people interact with technology, marked by the growing popularity of voice assistants and chatbots. As these systems' capabilities improve, they are increasingly becoming the preferred method of contact between customers and technology. Chatbots are popular owing to their simplicity and efficiency. Chatbots may manage several users' inquiries simultaneously, removing the need for individual staff members to address them. Therefore, businesses that integrate chatbots into their operational systems save money. This study empirically investigated the efficacy of natural language processing (NLP) in creating a chatbot specifically tailored to handle college information inquiries. This paper introduces Lucy, a chatbot that utilizes a deep neural network in conjunction with complex language models including BERT, RoBERTa, and DistilBERT architectures. College students were surveyed to assess Lucy's performance in a user research study. The research showed that Lucy is effective as a teaching tool for imparting knowledge to students, with an accuracy rate above 85%. From the evidence, Lucy is capable of subsequently reducing in-person enquiries by acting as a centralized forum for student questions, which highlights the chatbots in the education sector and also demonstrates the capabilities to improve the quality of information shared to the students.

Keywords: Chatbot, natural language processing, college information inquiry, artificial intelligence, student experience

1. Introduction

This study proposes a web-based chatbot employing Natural Language Processing (NLP), a branch of artificial intelligence, to address college-related enquiries. The advent of natural language processing (NLP) has revolutionized human–computer interaction, particularly through the proliferation of chatbots, which serve as intuitive interfaces for accessing information and performing tasks. The swift and vast development of AI chatbots and assistants has transformed the way humans interact with digital systems. Being able to interact with the system most humanely and naturally, providing accessibility options for individuals with various types of disabilities, and profiting the business and organization by reducing expenses on customer support and services, chatbots and AI assistants have effectively benefited the world (1). Machine Learning (ML) and Natural Language Processing (NLP) technologies have allowed chatbots the ability to multitask, efficiently handle queries, and to adapt effectively in any field in which humans are engaged. Academic institutions' adoption of these chatbots provides real-time responses to inquiries in a cost-effective manner and makes use of ML and NLP technologies (2). With the increasing expertise of students in the IT field, respective institutions must also adapt to these changes by developing a new approach to optimize communication and provide precise real-time information. Chatbots can manage to handle concurrent inquiries also while reducing the workload of administrative staff in that field. Our research was focused on the use of NLP-powered chatbots and AI assistants to effectively respond to academic institution related inquiries. We introduce Lucy, an AI chatbot customized for college students. Lucy utilizes modern and sophisticated architectures like BERT, RoBERTa, and DistilBERT, incorporating NLP techniques and deep neural networks. Lucy has gone through extensive testing and performance ratings. On our extensive user research involving college students, we found out that Lucy performed with an accuracy exceeding 85%, demonstrating its ability to handle inquiries effectively. Along with Lucy's ability to handle these inquiries, Lucy has portrayed the ability to be a centralized hub for students' inquiries, reducing the reliance on traditional face-to-face communication and promoting a more streamlined approach to information exchange within academic institutions. This research contributes to the growing use of chatbots and NLP implementations, by Lucy's potential impact on academic institutions and students. By using the most out of NLP-powered chatbots like Lucy, academic institutions can improve and enhance information distribution, increase student engagement, and adapt to the dynamic digital learning environment.

1.1 Literature review

The utilization of Machine Learning (ML) and Natural Language Processing (NLP) has greatly improved computers' ability to understand, analyze, and generate human language in various AI applications. The significant growth of chatbots can be attributed to their ability to reduce operational costs for businesses. As AI and machine learning technologies continue to advance, chatbots are expected to become more efficient, leading to further market growth (3). However, the major challenges faced by these chatbots during the conversation are misunderstanding the context and user's requirement, which often leads to generating sentences that are irrelevant to the user's query. (4) have introduced a new chatbot with AI-based sentiment analysis to better understand the context and user's intentions often during long interactions.

Day by day, chatbots are growing in popularity and the use cases are increasing exponentially, prominently in service sectors, which has resulted in the growth of market capital for these technologies reaching around USD 27 Billion by 2030 (5). This uptrend is the result of the advancement in NLP technologies powered by Machine Learning and Artificial Intelligence. One such example is the implementation of virtual nurse agents named "Ellie" by (6) to provide emotional support to women with breast cancer. Encouragingly, the results showed significant reductions in depressive symptoms and anxiety levels among patients.

In financial institutions, chatbots are often integrated into their mobile applications and web applications to address various queries related to products and their services. One such system presented by (7) discusses the preparation of a dataset from the bank's FAQ section of the website which enables the chatbot to answer queries of customers which otherwise had to be attended by a real person, saving time and cost. A similar intelligent system was proposed by (8) to generate personalized responses to queries from banking customers.

The academic sector has also witnessed growth in the use of chatbots in the form of virtual teaching assistants for tailored learning experiences. This type of system has shown encouraging results and its efficacy. According to (9), the AI chatbot developed using a zero coding technique enabled students to perform better academically in comparison to the regular students who were just interacting with the instructor. Similarly, another study discusses chatbot systems designed as language learning mediums, where (10) emphasize their efficacy as aides for language practice and promoting self-directed learning and overall effectiveness. The study conducted by (11) investigated learners of English as a Foreign Language (EFL) who engaged in a group discussion and monitored how their discussion was affected by a proceeding conversation with a chatbot and the results show that the interaction with the chatbot increased their awareness of critical thinking and enabled them to form inquiring mindset.

In summary, the literature review reveals a notable pattern in the contemporary usage of Natural Language Processing supported by AI and ML, where voice assistants and chatbots have arisen as essential interfaces for human–machine interaction (HMI). Given their ease of use, productivity, and capacity to attend to numerous requests concurrently, chatbots have evolved into indispensable possessions for companies and organizations aiming to optimize customer interactions and minimize operational expenses. Even though there have been advancements in the efficiency and capability of chatbots due to the progress of NLP and ML, there are still challenges in accurately portraying the user needs and their context, which always at the end lead to irrelevant or even out-of-context responses. Although there has been a widespread implementation of chatbots across various industries, there is still a lack of research, use and effectiveness in addressing inquiries related to college knowledge. This proves that there is a need for more studies in such academic settings and environment. That is why we focused this research work on the use of NLP-powered chatbots and AI assistants to effectively and relevantly respond to academic institution related inquiries. The aim of this research work was to develop and evaluate Lucy, a chatbot that effectively handles requests for college information by using deep neural networks and advanced NLP methods. This will distribute the knowledge better and will reduce the case of a person intervening to respond to inquiries.

The objectives of this research were as follows:

1. To build, run, and evaluate Lucy, a chatbot that will handle and respond to queries related to college and academic institutions using the BERT, RoBERTa, and DistilBERT architectures.
2. To carry out a user research study in order to evaluate how well Lucy delivers relevant and accurate information to college students.
3. To assess how Lucy affects less in-person questions and how well information is distributed generally in a college environment.
4. To improve the appropriateness and accuracy of chatbot responses by identifying and addressing challenges related to understanding context and user needs.

2. Methodology

This section describes the methodology used to develop the college information enquiry chatbot system.

2.1 System design

The aim of the system design process in this project was to collect extensive data and comprehend the system and its components, as illustrated in **Figure 1**, following the architectural entities outlined in the system architecture models and views. System design is a crucial element influencing the application quality in this project. The system design, also known as the project methodology, was elucidated by examining the system's block diagram. The system design blueprint for our AI-integrated web-based chatbot application requires the use of various web tools, technologies, and NLP algorithms. As a widely used front-end framework for developing single-page web applications, React.js will be utilized to develop the chatbot interface. The infrastructure of the application is constructed using FastAPI and linked to a PostgreSQL database that possesses the capability to process both structured and unstructured data. The workflow of the chatbot system is initiated when the user inputs a query into the conversation interface. API calls are utilized by the messaging interface to transmit user input to the server. The server communicates with the database and applies text-processing techniques such as stemming, tokenization, and lemmatization to the input. To generate an appropriate response, the natural language processing (NLP) model examines the incoming data and processes it utilizing architectures including LSTM, GRU, and Transformer. The subsequently generated output is transmitted to the server, which routes it to the messaging interface. The messaging interface allows users to access and review their responses to enquiries. To train the chatbot model, we must obtain a balanced dataset to ensure its proper operation. A feedback mechanism is in place to facilitate user contribution to the chatbot. The feedback was stored in the database, verified by the administrator, and used to retrain the model. The system design diagram illustrates the use of online tools, technologies, and NLP algorithms to create a web-based chatbot application with an interactive interface. This application enables users to ask questions and receive responses using advanced NLP techniques. The system architecture allows the chatbot to be efficient, effective, and user-friendly.

Figure 2 shows a flow diagram starting with the user entering a query into the chat interface. The data were then sent to the server through an API call. The server interacts with the database to retrieve the necessary information. Input involves word processing techniques such as tokenization, lemmatization, and stemming. The input undergoes processing before being input into the NLP model, which is constructed using TensorFlow. The NLP model forecasts a suitable response to an enquiry. Subsequently, the response is sent back to the server, which then passes it to the chat interface. Finally, the user views the response to the query on the chat interface.

We obtained a balanced dataset to train the chatbot effectively. The feedback system enables users to submit inputs to chatbots. Feedback was subsequently documented in the database. The administrator assessed the feedback and adjusted the model through retraining.

2.1.1 Module design

The chatbot system is divided into two parts: the admin portion and the user section.

Admin: The functionalities permitted in the admin area of the college information enquiry chatbot system are as follows.

1. Creating an admin account
2. Logging in as an admin
3. Creating, reading, modifying, and removing responses
4. Reading, modifying, and removing feedback
5. Creating, reading, modifying, and removing questions
6. Creating, reading, modifying, and removing users

User: The approved functionalities in the chatbot system user section are as follows.

1. Asking questions through text
2. Asking questions through voice
3. Providing feedback about the chatbot to the admin

The module architecture of the chatbot system ensures a seamless experience for both users and administrators. The user interface is straightforward to use, whereas the admin interface contains all capabilities necessary to oversee the chatbot system.

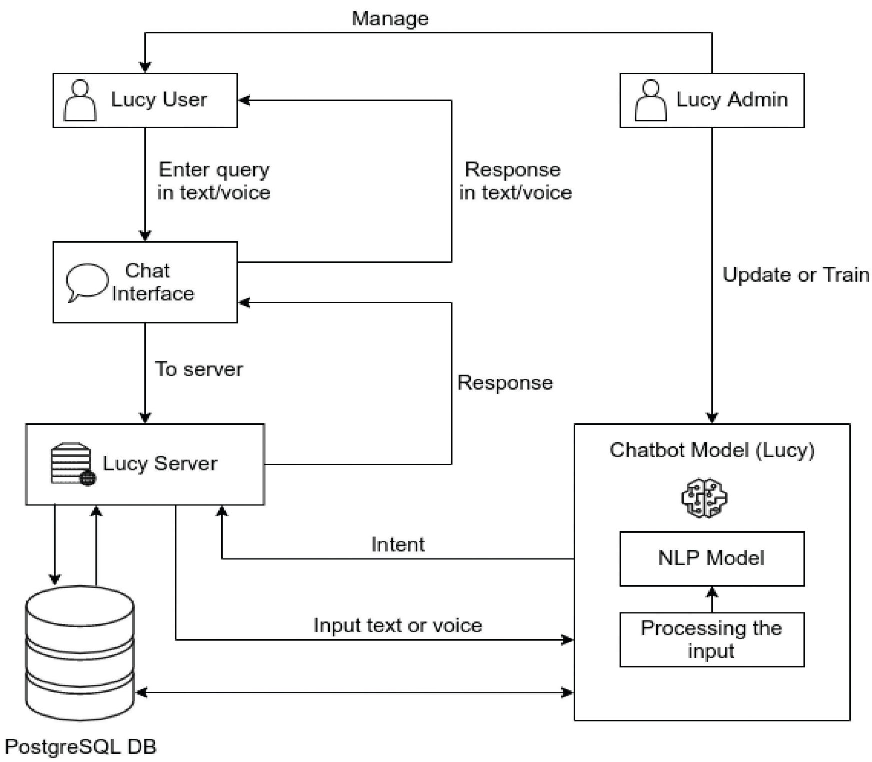

FIGURE 1 | The block diagram of the college information inquiry chatbot system.

2.2 Natural language processing (NLP) models

A neural network was used in these models. Three neural language models, BERT, RoBERTa, and DistilBERT, were considered in this study. Based on the accuracy and performance of each model, one can choose to employ it in this situation.

2.2.1 BERT

Google developed a machine learning method known as Bidirectional Encoder Representation from Transformers (BERT) to train natural language processing systems. It refers to the encoder component of the transformer design, which consists of an encoder and decoder. Decoders are frequently utilized in models such as generative pre-trained transformers (GPTs). BERT is meant to jointly condition the left and right contexts in all layers to pretrain deep bidirectional representations from unlabeled text, as shown in **Figure 3**. A single additional output layer can be added to the pre-trained BERT model to create cutting-edge models for a range of applications.

2.2.2 RoBERTa

RoBERTa builds on BERT's language-masking method. Here, BERT's next-sentence pre-training aim was removed, and the training was performed with significantly larger minibatches and learning rates. RoBERTa was trained on a larger volume of data over a longer period of time than BERT. Consequently, RoBERTa representations generalize to downstream tasks better than BERT (12) representations. RoBERTa shares the same architecture as BERT but employs a different pre-training strategy and a byte-level BPE as a tokenizer (similar to GPT-2).

2.2.3 DistilBERT

DistilBERT is a distilled version of BERT that uses half of the parameters, is quicker, and smaller, while still maintaining the performance characteristics of BERT. Using a process known as "distillation," DistilBERT replaces a portion of Google's massive neural network with a smaller one. The goal is to create a smaller and more efficient version of BERT (13). The distillation process is illustrated in **Figure 4**.

The input query was first tokenized by the DistilBERT tokenizer and then transformed into embeddings via token embedding, position embedding, and segment embedding. The transformer encoder then takes the embeddings, encodes them, and passes the output of the hidden states to the following layer.

2.2.4 Simple feed forward neural network

A feed-forward neural network, illustrated in **Figure 5**, is a type of artificial neural network in which the flow of information happens in one direction, from the input nodes through one or more hidden layers to the output nodes, without any loops or cycles in the connections between nodes. Given that the input is only processed in one direction, the feed-forward model is the most

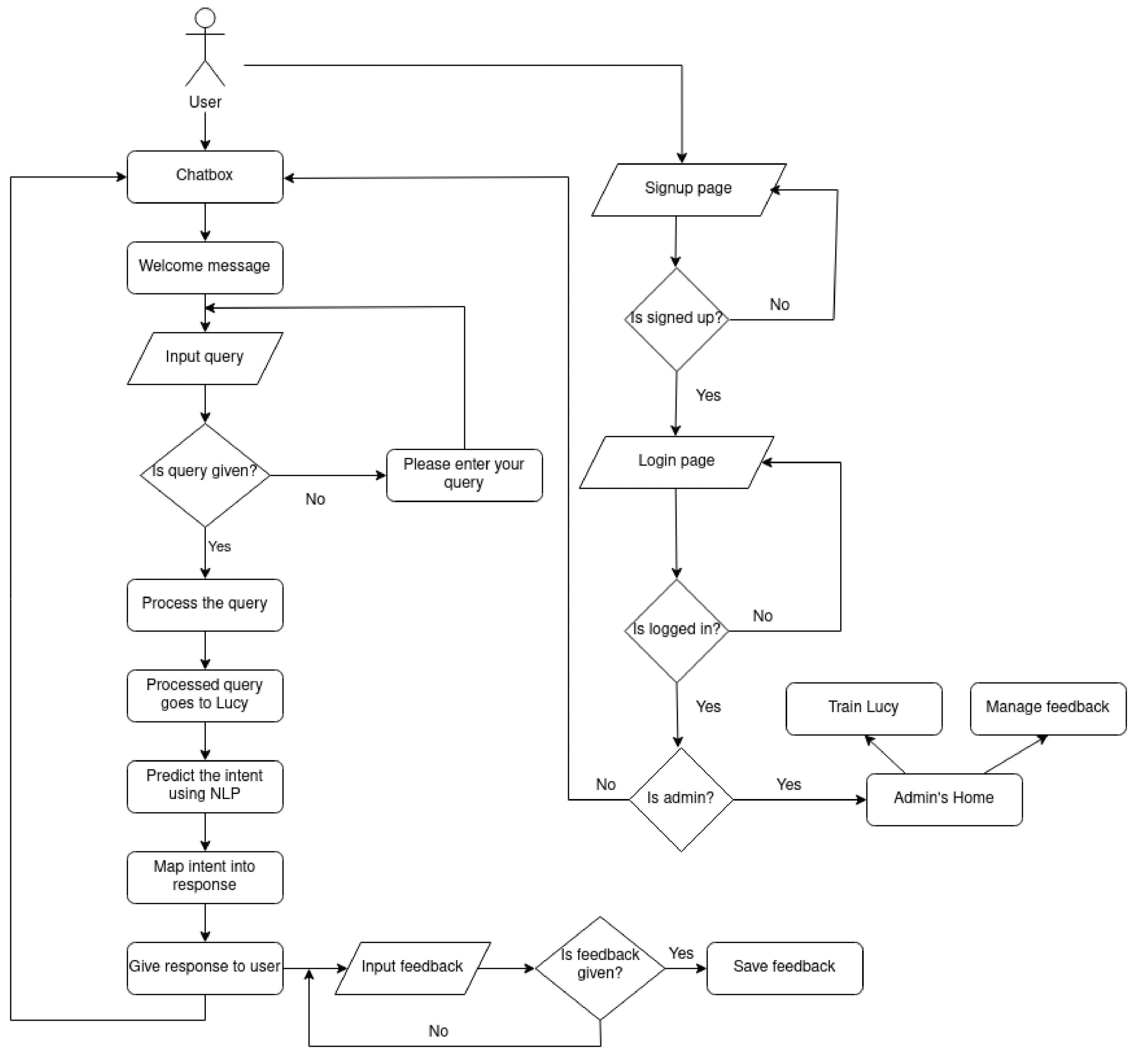

FIGURE 2 | The flow diagram of the college information inquiry chatbot system.

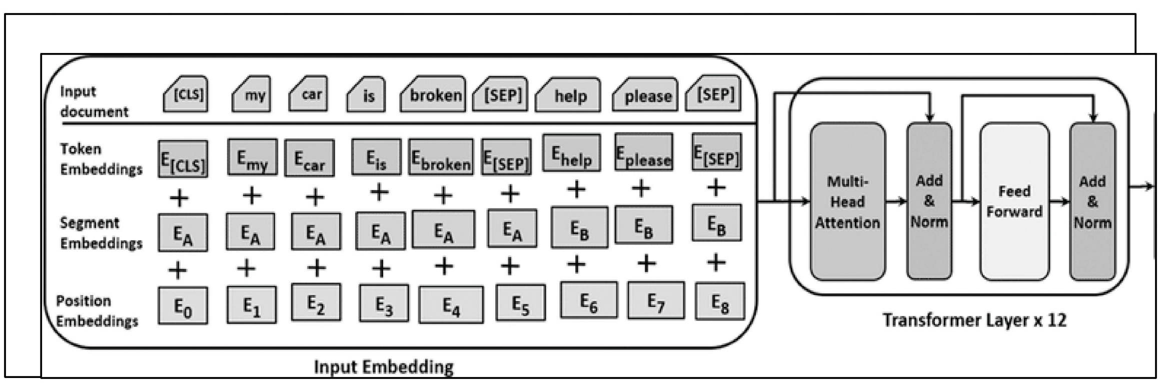

FIGURE 3 | Architecture of BERT base.

basic type of neural network. In this case, we employed one output layer and two concealed layers. The output of the 512 neurons in the first hidden layer was passed to a rectified linear unit (ReLU) activation function, and then each of the function's results was passed to each of the 256 neurons in the second hidden layer. The 42 outputs were passed through a ReLU activation function again in the output layer to introduce non-linearities and enable powerful learning in neural networks. The log-softmax activation function receives each output from the output layer neuron and outputs a probability value of 42 distinct intentions.

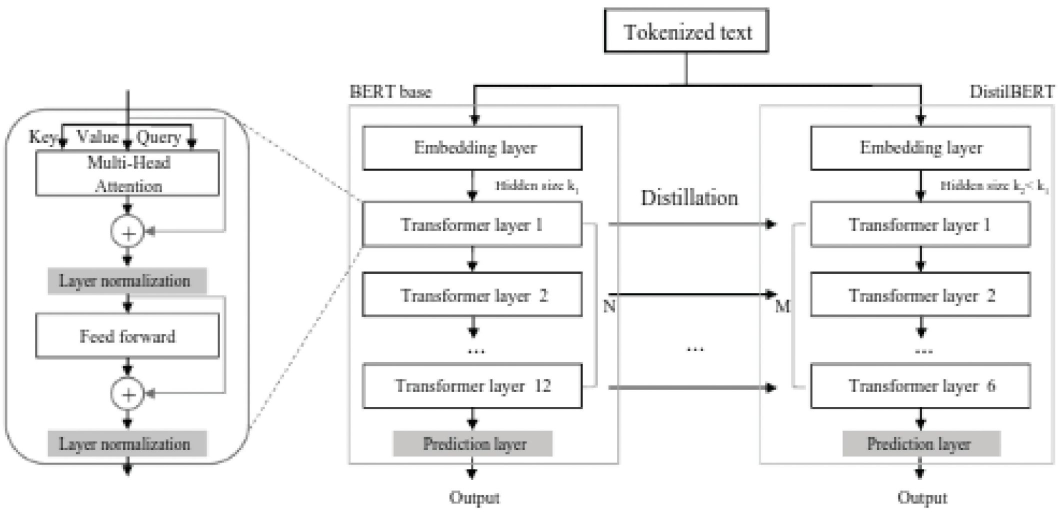

FIGURE 4 | Architecture of DistilBERT base.

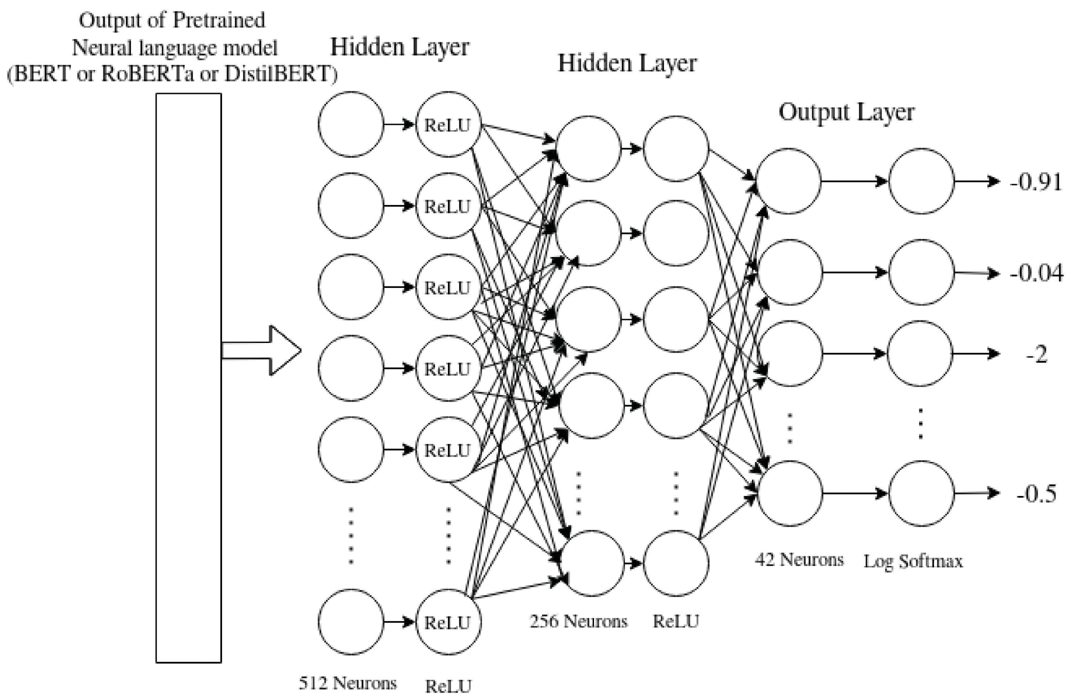

FIGURE 5 | Simple Feed-forward neural network.

2.3 Chatbot implementation

The chatbot system was implemented using the Flask Web framework in Python. The system was deployed on a cloud server to ensure that it is accessible to users anywhere in the world. The chatbot system was also integrated with the Google Cloud Speech-to-Text API to enable users to ask questions through voice input.

The college Inquiry chatbot, also known as Lucy's server, was built using the FastAPI framework for Python. Students can access the app using a web app built using the React Framework. FastAPI uses SqlAlchemy, an ORM tool, to connect to the PostgreSQL database backend. The NLP model was built using the BERT architecture with an extra feed-forward neural network. The complete codebase of the project was hosted on the GitHub.

2.3.1 Speech recognition

Speech recognition is the process of translating voices into a text. The Web Speech API SpeechRecognition (14) interface was used to implement speech recognition on the front end. The Web Speech API allows developers to incorporate speech data into web-based applications. The speech recognition interface is used to access speech recognition, which allows the device to interpret the spoken context from audio input and respond appropriately (usually using the device's default

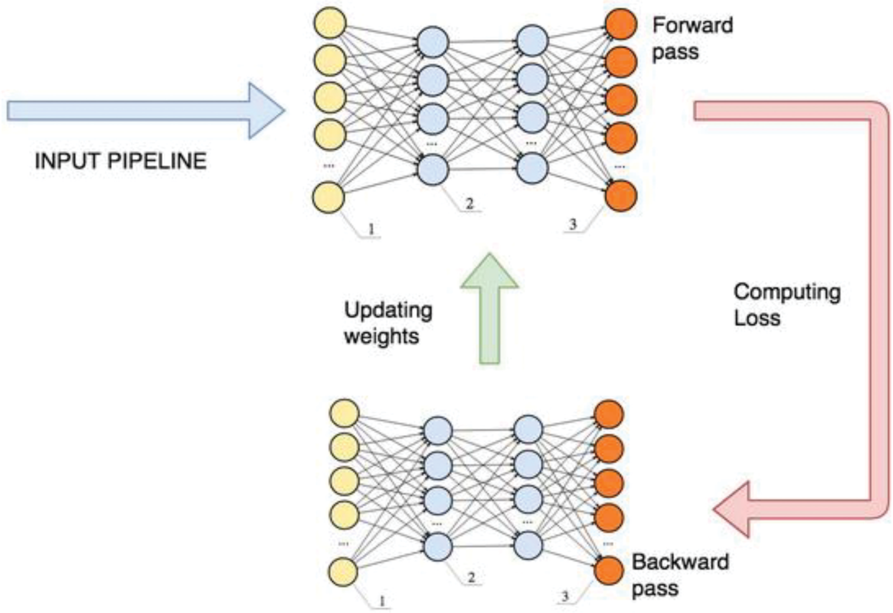

FIGURE 6 | Block diagram of model training.

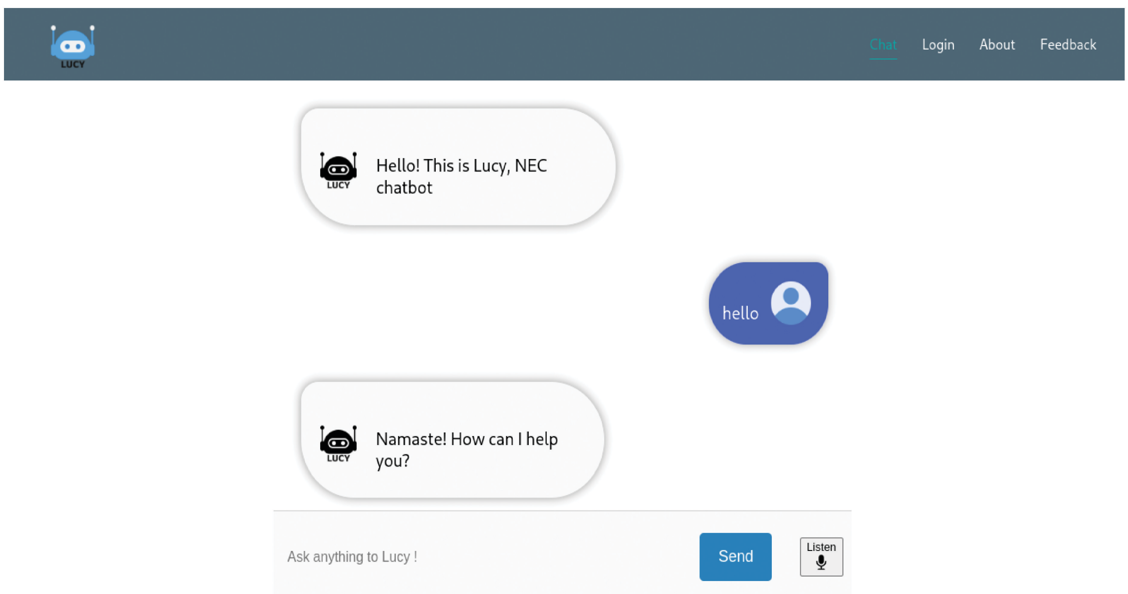

FIGURE 7 | Chat Interface.

speech recognition service). The React Speech Kit, which already has the API implemented, was used.

2.3.2 Text to speech

Text-to-speech refers to the voiceover of the text. The SpeechSynthesis (15) interface of the Web Speech API was used to implement text-to-speech at the front end. To start and halt speech and issue further commands, one can utilize the SpeechSynthesis interface to acquire details about the synthesis voices that are accessible to the device.

2.3.3 Frontend

React, a JavaScript package, was used to create the front-end web application because of its adaptability, superior performance compared to other well-known frameworks, and simplicity of usage. React makes it easier to create dynamic one-page applications. Each reaction component of the user interface was constructed using a Material-UI.

2.3.4 Backend

Python's FastAPI web framework, better known as Lucy's server, was used to create the College Inquiry chatbot.

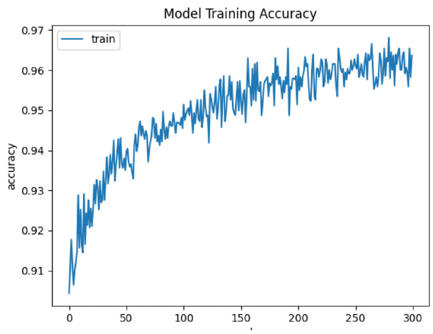

Figure 8(a) Training accuracy of BERT model

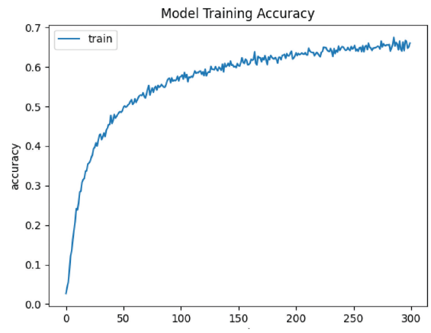

Figure 8(b) Training accuracy of RoBERTa model

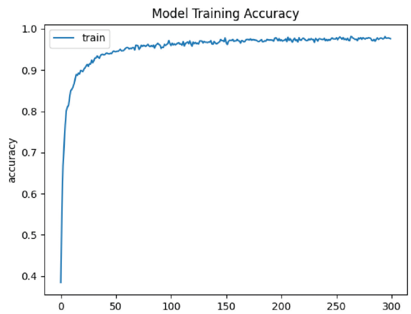

Figure 8(c) Training accuracy of DistilBERT

FIGURE 8 | (a) Training accuracy of BERT model. **(b)** Training accuracy of RoBERTa model. **(c)** Training accuracy of DistilBERT.

The ORM tool used to connect the database to the server was SqlAlchemy. It is written in Python and provides an application developer with the SQL's power and flexibility. The object model and database schema may be developed in a clear, decoupled manner from the start owing to SQLAlchemy's object-relational mapper (ORM), which allows classes to be mapped to the database. FastAPI assisted API construction by offering a wide range of HTTP utility methods and middleware. The chatbot model is also served by the backend servers and is reachable via an API endpoint.

Lucy backend architecture has four major parts

1. Routes: FastAPI routers generate unique URLs (routes) for each request.
2. Controller: Return a response based on the route accessed.
3. Model: A data template utilized by controller functions to communicate with PostgreSQL
4. PostgreSQL is a database that contains real data.

2.3.5 Dataset overview

The dataset used in this project is designed for developing an intent-based chatbot, formatted as a CSV file with two columns: one for text questions and another for corresponding intents (Table 1). The dataset, created by us, comprises 4,199 rows spanning 42 categories of intent, with each entry recorded manually in an Excel sheet.

TABLE 1 | Sample data from the dataset.

Text	Intent
Talk about the individuals who made you	creator
Who am I chatting to ?	name
What is the daily timetable of college?	hours
How can I reach the college by phone?	number
Can you tell me the courses available in NEC?	course

2.3.6 Model building

In this context, the chatbot model created using deep learning is referred to as the model. A complete model was created using a straightforward feed-forward neural network and pretrained BERT. Along with BERT, BERT variants RoBERTa and DistilBERT were used. The final hidden layer of the pretrained BERT output was linked to a feed-forward neural network. We simply need to train the feedforward network in this manner. Transfer learning is another term used for this method. Hugging Face, an American business that creates tools for creating applications utilizing machine learning, provided a pre-trained BERT model. Hugging Face offers a transformer Python module that allows importing the pretrained BERT model and its variations.

1. Pytorch is a free and open-source machine-learning framework that is used to create a feed-forward neural network. There are four key components of feedforward neural networks.

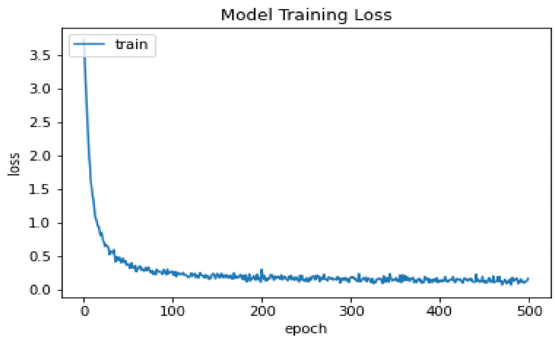

Figure 9(a) Training loss of BERT model

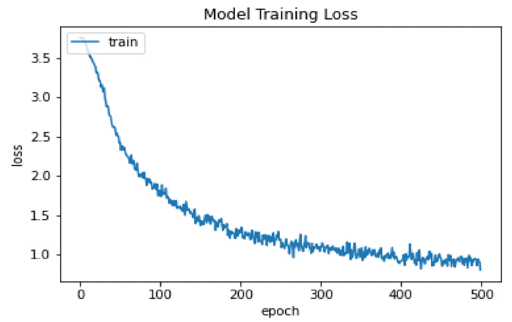

Figure 9(b) Training loss of RoBERTa model

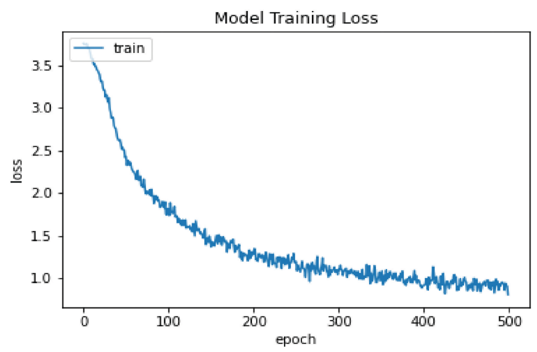

Figure 9(c) Training loss of DistilBERT

FIGURE 9 | (a) Training loss of BERT model. **(b)** Training loss of RoBERTa model. **(c)** Training loss of DistilBERT.

2. Neurons, which serve as the smallest building blocks of a neural network and function similarly to biological neurons, make up a neural network. Our neural network comprises 42 neurons in the output layer, 264 neurons in the second hidden layer, and 512 neurons in the first hidden layer.
3. The goal of activation is to inject non-linearities into neural networks, thereby allowing them to learn complex operations. In this model, two activation functions are employed: ReLU and log softmax.
4. Dropout: This contributes to lowering the squared norm of the weights, thereby reducing overfitting. Only two hidden layers were used for dropout.

2.3.7 Model training

Model training is an iterative process in machine learning, as depicted in **Figure 6**. Any machine learning algorithm's training time for a model depends on the model's architecture.

The general training procedures for Lucy are listed below:

1. At first, the pre-processed training data were fed into the model which was already built previously.
2. Then the model gives a prediction value as output which is passed to a loss function that measures the loss value by comparing the prediction with the actual value.
3. After that backward propagation takes place to calculate the gradients.
4. An appropriate optimizer is used to update the parameters of the model.

2.3.8 Model prediction

The model can now predict the outcome given the input once it has been trained. A query or question serves as the input data for the model we utilized, and the question's intent serves as its output.

Overall steps of the prediction of Lucy are given below:

1. Initial pre-processing, i.e., tokenization, is done on the input query.
2. The trained model is then given the tokenized input, and it produces an array of tensors containing the log probability of each intent.
3. The purpose with the highest log likelihood is then chosen, and it is mapped to an appropriate answer.

3. Results or finding

3.1 Output obtained

The following output was obtained as the result of completing the project.

1. Users can create their account and log into them and get admin privilege.
2. Users can ask questions to a bot via the chat interface through text and voice.
3. Users can give feedback about the chatbot.
4. Admin can create, view, update, and delete an intent.
5. Admin can create, view, update, and delete the dataset of the chatbot.
6. Admin can train the model and also can update the model in the updated dataset.
7. Chatbot model was successfully trained on the provided dataset and training accuracies of 96%, 66%, and 97% and testing accuracies of 90%, 67%, and 89% were obtained using BERT, RoBERTa, and DistilBERT, respectively.
8. These results indicate that BERT and DistilBERT offer high accuracy and good generalization, while RoBERTa's low performance shows underfitting and suggests a need for further improvement.

Different components have been used and created using React for the frontend web to make the chat interface interactive, as illustrated in **Figure 7**. NLP-powered chatbots analyze each event's output and provide results based on the input. Technically, none of the modules have any errors. We employed a variety of testing methodologies, including white box and black box testing.

Our ultimate goal was to offer a standard, user-friendly, and effective approach to get information about the college through a single interface. The output from the app's front end was tested using a variety of testing techniques, including white box testing and black box testing. The results of those tests were more than excellent. Unit testing was also used to test the various API developed for the front end to interface with the backend, and the results were fairly spectacular. For testing, many test scenarios are developed, and the majority of the scenarios pass with flying colors. The NLP-model, also known as Lucy, is the brains behind this project because it can handle any kind of query and provide an appropriate response. Following model training, we also went through a number of model testing stages.

3.2 Accuracy of the model

Figure 8a displays the accuracy of the BERT model in each epoch in the training phase. It reveals that the model's accuracy undergoes too many fluctuations from the start to the very end. **Figure 8c** illustrates that the DistilBERT model's accuracy undergoes an abrupt increase at the beginning, and after about 20 epochs, it starts to rise gradually. On the other hand, **Figure 8b** shows the accuracy of the RoBERTa model, which almost follows the trend of DistilBERT with a little bit of fluctuations in every epoch during training.

TABLE 2 | Different testing metrics applied on model.

Model	Precision	Recall	F1-Score	Accuracy
BERT	0.90	0.90	0.90	0.90
RoBERTa	0.71	0.67	0.68	0.68
DistilBERT	0.90	0.89	0.89	0.89

3.3 Loss of the model

Figure 9a displays the loss value of the BERT model in each epoch during the training phase. It suggests that the model's loss decreases with lots of fluctuations from the start to the end. Meanwhile, **Figure 9b** demonstrates that the loss of the RoBERTa model gradually decreases in every epoch with very small fluctuations during training. Similarly, **Figure 9c** illustrates that the DistilBERT model follows the same trend.

3.4 Testing metrics

Table 2 presents the outcomes of various testing metrics used to evaluate different versions of a trained model. Four key testing metrics, namely precision, recall, f1-score, and accuracy, are employed to compare the performance of different variants of the trained model. The results indicate that the precision of the DistilBERT and BERT models are the highest among all other models, standing at 0.9. In addition, the accuracy of BERT was higher than that of the DistilBERT and RoBERTa models. Furthermore, all other performance metrics of BERT surpass those of the other two models, making it a suitable choice for solving our problem on a given testing dataset.

4. Conclusion and future work

The primary objective of introducing Lucy, an NLP-based chatbot to college information requests, was to offer users a swift and efficient means of finding solutions to their inquiries. However, the initial implementation of the ANN model was hindered by its superficial design and insufficient training data, which prevented it from realizing its complete potential. We started using advanced language models such as BERT, RoBERTa, and DistilBERT architectures to train the model, which led us to achieve training accuracies of 96%, 66%, and 97% and testing accuracies of 90%, 67%, and 89%, respectively. While the BERT and DistilBERT models showed high accuracy, the RoBERTa model achieved a lower testing accuracy of 67%, indicating underfitting issues. The project demonstrated the potential to create a reliable, efficient, and useful chatbot specifically for handling college and academic institution related queries. The project's success has prompted consideration of expanding tasks, including improving the database to better meet end users' needs and preferences, gathering more datasets, and experimenting with different hyper-parameters to improve language model accuracy during testing. To address RoBERTa's underfitting, future efforts will focus on increasing the model's training data, increasing its training duration, reducing regularization, and refining its architecture. Moreover, there is a chance to submit a research paper utilizing the discoveries. Although the idea has demonstrated promise, additional endeavors are necessary. One weakness of the study is related to the availability of data. Enhancing the chatbot's accuracy can be achieved by implementing more relevant data collection strategies and improving data pre-processing technology. Moreover, combining voice recognition with sentiment analysis has greater potential to provide the user with accessibility and personalization. Lucy's implementation effectively utilized NLP models to construct chatbots that meet end users' requirements. This technology has the potential to be applied in several industries to enhance user experience and customer service through further research and development.

Conflict of interest

The authors declare that they conducted their research without any commercial or financial ties that could be seen as potential conflict of interest at present or in future.

Author contributions

In the article "Bridging the Gap in College Information Access Through Natural Language Processing Powered Lucy Chatbot" is a project output of B.E. Computer Science and Engineering students Rupesh Gelal, Suraj Karki, and Saman Shrestha at Nepal Engineering College. They conducted deep research and development work, focusing extensively on designing, implementing, and evaluating the Lucy Chatbot to enhance the information accessibility for college students using NLP techniques. Asst. Prof. Krishna Bikram Shah supervised the project and provided essential guidance on NLP methodologies, ensuring adherence to academic standards and critical feedback throughout the development process, thus significantly contributing to the project's successful completion and writing this research article.

Funding

No financial support was received for this research.

Acknowledgments

We thank our colleagues at Nepal Engineering College for their constructive feedback and our family for constant encouragement.

References

1. Asfoura E, Kassem G, Alhuthaifi B, Belhaj F. Developing Chatbot Conversational Systems & the Future Generation Enterprise Systems. *Int J Interact Mob Technol*. (2023) 17(10):155–75. doi: 10.3991/IJIM.V17I10.37851
2. Kesarwani S, Titiksha S, Juneja S. Student Chatbot System: A Review on Educational Chatbot. *7th International Conference on Trends in Electronics and Informatics, ICOEI 2023 - Proceedings*. London: (2023). doi: 10.1109/ICOEI56765.2023.10125876
3. Singh D, Manju, Jatain A. Artificial Intelligence and Natural Language Processing Inspired Chabot Technologies. *Recent Adv Comput Sci Commun* (2023): doi: 10.2174/2666255816666230712141148
4. Dongbo M, Miniaoui S, Fen L, Althubiti SA, Alsenani TR. Intelligent chatbot interaction system capable for sentimental analysis using hybrid machine learning algorithms. *Inf Proc Manage*. (2023) 60:103440. doi: 10.1016/J.IPM.2023.103440
5. Anon. *Chatbot Market Size Worth $27,297.2 Million By 2030*. (2023). https://www.grandviewresearch.com/press-release/global-chatbot-market [Accessed: 10 April 2023]
6. Bickmore TW, Pfeifer LM, Jack BW. Taking the time to care: Empowering low health literacy hospital patients with virtual nurse agents. *Conf Hum Fact Comput Syst Proc*. (2009) 1265:1274. doi: 10.1145/1518701.1518891
7. Mujeeb MS, Hassan U. An intelligent ML chatbot for banking systems. *Dogo Rangsang Res J UGC Care Group I J*. (2023) 13:2347–7180.
8. Dole A, Sansare H, Harekar R, Athalye S. *Intelligent Chat Bot for Banking System*. (2015).
9. Essel HB, Vlachopoulos D, Tachie-Menson A, Johnson EE, Baah PK. The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education. *Int J Educ Technol Higher Educ*. (2022) 19(1):1–19. doi: 10.1186/S41239-022-00362-6/TABLES/7
10. Windiatmoko Y, Rahmadi R, Chatbot H, Pradhan R, Shukla J, Bansal M, et al. Artificial Intelligence (AI) Chatbot as Language Learning Medium: An inquiry. *J Phys Conf Ser*. (2019) 1387(1):012020.
11. Goda Y, Yamada M, Matsukawa H, Hata K, Yasunami S. Conversation with a chatbot before an online EFL group discussion and the effects on critical thinking. *J Inf Syst Educ*. (2014) 13:1–7. doi: 10.12937/ejsise.13.1
12. Anon. *RoBERTa | PyTorch*. (2023). https://pytorch.org/hub/pytorch_fairseq_roberta/ [Accessed: 10 April 2023].
13. Sanh V, Debut L, Chaumond J, Wolf T. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. (n.d.). https://github.com/huggingface/transformers.
14. Anon. *SpeechRecognition PyPI*. (2023). https://pypi.org/project/SpeechRecognition/ [Accessed: 10 April 2023].
15. Anon. *speechSynthesis PyPI*. (2023). https://pypi.org/project/speechSynthesis/ [Accessed: 10 April 2023].