Grenze International Journal of Engineering and Technology, June Issue

A Deep Learning based Chatbot for Academics using NLP

Jai Chiranjeeva Dadi[1], Shaik Nazeer[2] and Bharathi Mohan G[3]
[1-3]Amrita Vishwa Vidyapeetham, Chennai, India Email: jaichiranjeeva91@gmail.com
Email: shaiknazeer297@gmail.com, g-bharathimohan@ch.amrita.edu

***Abstract*— Artificial Intelligence (AI) and Natural Language Processing (NLP) have significantly enhanced human-machine interaction, particularly in the education sector. This study presents the implementation of an AI-driven chatbot designed to assist learners on an e-learning platform using deep learning techniques. The chatbot employs a feedforward neural network to classify user intent, trained on a structured dataset categorized across specific learning domains. Preprocessing methods such as tokenization, stemming, and bag-of-words representation are applied to refine the data for classification. The model, built with PyTorch and optimized using the softmax function, selects the most relevant response with high precision. Performance metrics, including precision, recall, and F1-score, indicate a robust and efficient chatbot, achieving an accuracy of 99.20% in managing student inquiries. The chatbot successfully produced relevant responses for 98.7% of test queries, demonstrating its adaptability across multiple academic subjects. Additionally, the system integrates the Gemini API to enhance response accuracy for unfamiliar queries, ensuring a more dynamic and interactive learning experience. The chatbot offers real-time user engagement, making it an effective tool for personalized and context-aware education assistance.**

***Keywords*— Chatbot, Natural Language Processing, NLTK, Intent Recognition, Gemini API, Query Resolution, AI Chatbot, Deep Learning, PyTorch.**

I. INTRODUCTION

With the evolution of artificial intelligence, chatbots have emerged as an essential tool for improving user experience in many fields, such as customer service, healthcare, and education. Chatbots utilize NLP and deep learning algorithms to analyze human language and deliver contextually relevant responses. In the field of education, chat-bots act as virtual teaching assistants, allowing students to receive real-time answers to their questions, access learning resources, and enhance their learning experience [1]. The capacity of AI chatbots to offer immediate feedback and instruction heavily improves the learning process by rendering information easily accessible. With an increasing number of institutions embracing digital learning solutions, incorporating smart chatbots guarantees tailored and engaging learning experiences that address the needs of various learners [2].
Conventional rule-based chatbots are dependent on pre-established responses, restricting their capacity to manage dynamic questions. These chatbots have a rigid decision tree-based architecture, making them less versatile when dealing with user input variations. Machine learning-based chatbots, on the contrary, leverage massive datasets to learn and generalize user interaction, making them more flexible and effective [3].

Grenze ID: 01.GIJET.11.2.800_22

Through the use of deep learning models, chatbots can learn complicated linguistic patterns and give precise feedback, thus increasing user interaction and information sharing [4]. In contrast to rule-based models, deep learning chatbots become better with increased exposure to new data, hence improving their comprehension of user intention and giving responses that are pertinent to the context. This makes them valuable resources in settings where constant and diversified interactions, such as learning, are demanded. Some implementations utilize advanced NLP techniques, such as WordNet-based approaches, to refine chatbot responses through user feedback, allowing for continuous improvement [5]. Additionally, deep learning approaches like Seq2Seq models with an Attention Mechanism have shown promising results in handling complex queries in university chatbots, achieving competitive BLEU scores in multilingual contexts.

The intended chatbot aims to ensure smooth interaction by classifying user queries under predefined tags and matching them to suitable responses. This method makes user inputs effective and classified to produce effective and contextually aware responses. The major components are text preprocessing methodologies such as tok-enization, stemming, and bag-of-words model, followed by the training of a feedforward neural network utilizing PyTorch. These preprocessing operations transform text data into numerical forms that are processed efficiently by machine learning algorithms. The model estimates the probability distribution of potential responses and picks the most appropriate response. The use of softmax activation in the last layer ensures that the chatbot chooses the most appropriate response with high confidence [6]. Some educational institutions have already integrated chatbot frameworks like Dialogflow to automate student query resolution, improve accessibility, and ensure uninterrupted information availability [7].

This research paper explains the methods and implementation approach used in the creation of the chatbot. Perfor-mance testing proves the chatbot's competence in handling student queries and highlights areas where the chatbot needs further improvement. The results illustrate the capabilities of deep learning methods to drastically improve the performance of a chatbot through better comprehension and response generation. Future development will em-phasize increasing training data, improving response accuracy, and incorporating reinforcement learning methods to enhance contextual understanding. By integrating AI-powered academic assistance tools, institutions can also explore self-training mechanisms that enable chatbots to refine responses over time, ensuring continuous adapta-tion to evolving student queries [7]. Through the implementation of these developments, the chatbot can become a more advanced virtual assistant that can address a wider variety of student questions, engage more effectively, and provide a more interactive and effective learning environment.

II. Background Study

The integration of Artificial Intelligence (AI) in academic environments has led to the development of chatbots that assist students, instructors, and administrative personnel with multiple academic and operational tasks. These chatbots, driven by Machine Learning (ML) and Natural Language Processing (NLP), enhance communication, provide instant access to information, and improve the overall learning experience. Several studies have explored chatbot development methodologies and their effectiveness in educational settings. AI-driven chatbots are being designed to support students in both academic and extracurricular activities, reducing faculty workload and im-proving student interaction with college administrations [1]. Some implementations leverage NLP, WordNet, and ML algorithms to interpret student queries and refine responses through feedback mechanisms, enabling contin-uous improvement [2]. Deep learning approaches, such as RNN-based Sequence-to-Sequence (Seq2Seq) models with an Attention Mechanism, have been employed for university-related chatbots, demonstrating efficiency in handling complex language-specific queries, achieving a BLEU score of 0.41 in the Myanmar language [3]. In ad-dition to student assistance, chatbots have been integrated into college websites using frameworks like Dialogflow, automating query resolution, enhancing user interactions, and ensuring uninterrupted availability of information without human intervention [4]. These integrations improve accessibility, allowing students to receive prompt responses regarding admission processes, course details, and other academic inquiries.

Beyond academic interactions, chatbots are being extensively utilized for automating responses to frequently asked questions (FAQs). Rule-based chatbots combined with ML techniques optimize administrative communication by providing accurate and timely responses to inquiries about policies, events, and procedures, thus reducing the workload of college staff [5]. Some advanced chatbot systems extend beyond basic FAQ handling by incorporating NLP and ML techniques for answering both simple and complex college-related queries. These chatbots empha-size data preprocessing techniques such as tokenization, stemming, and lemmatization, significantly improving response accuracy and efficiency [6]. Additionally, chatbots with self-training capabilities refine their responses over time based on user interactions, ensuring continuous adaptation to new queries. AI-powered chatbots have also been employed to assist students in research topic validation. By integrating with web-based platforms and utilizing AI-driven validation mechanisms trained on Scopus-indexed articles, these chatbots provide automated academic advising and ensure research relevance [7]. Such implementations highlight the shift from traditional chatbots that simply retrieve information to intelligent assistants that actively engage with students, helping them navigate academic challenges and make informed decisions regarding their studies.

The increasing adoption of AI-driven chatbots in educational institutions underscores their potential to streamline administrative processes and facilitate academic assistance. As chatbot technology continues to evolve, future advancements are expected to focus on improving response accuracy for complex queries, expanding language support, and integrating voice-based interactions to create a more immersive user experience.

III. METHODOLOGY

The chatbot system is structured into three primary components: the Frontend Interface, the Backend Model, and External API Integration. These components work together to facilitate seamless interaction, process user queries efficiently, and provide accurate responses. The system workflow is illustrated in Fig. 1.

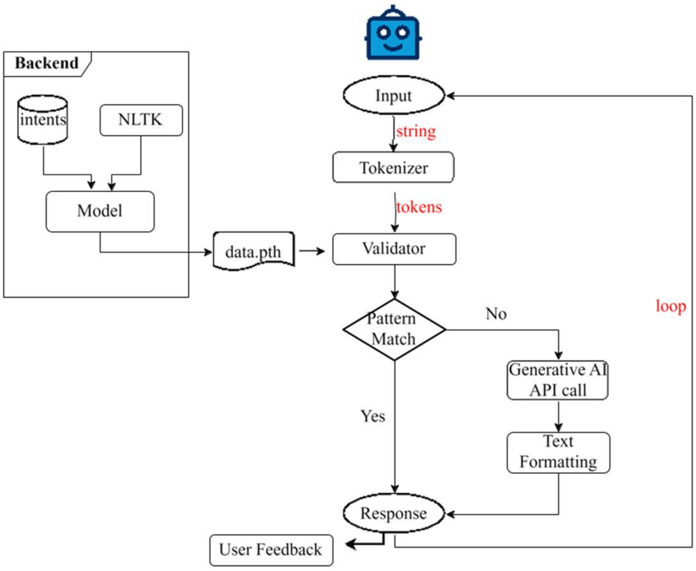

Fig. 1. System Architecture of the Chatbot.

A. Data Collection and Preprocessing

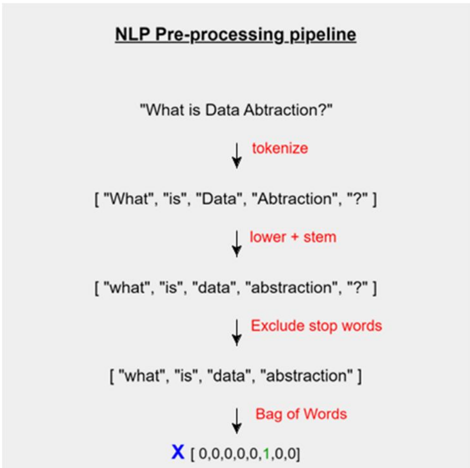

Fig. 2. Data Preprocessing

The process of training the chatbot starts with data gathering and preprocessing for proper learning. The dataset includes predefined text patterns along with their responses, which fall under particular tags. Tokenization is performed by utilizing the Natural Language Toolkit (NLTK) to divide text into small meaningful pieces of text like words or phrases. For ensuring consistency, stemming and lowercasing are utilized, lowering words to their base form and providing case insensitivity. Also, punctuation symbols and stop words are discarded to reduce noise and enhance the quality of textual data prior to its ingestion in the model. This is illustrated in Fig. 2.

B. Bag of Words Representation

For making the chatbot numerically process and understand the text, the bag-of-words (BoW) representation is utilized. The technique collects all of the distinct words within the dataset and represents each of the input sentences as a vector of numbers. Each of these vectors comprises binary values (1 or 0), stating the presence or absence of a vocabulary word within the input text. This ordered numerical representation enables the chatbot to compute text effectively and identify patterns while classifying. Although BoW fails to capture word order and semantic relationships, it is still a useful and computationally cheap method for text-based classification tasks.

C. Model Architecture

The fundamental of the chatbot is based on a feedforward neural network, which works to receive input text and output proper responses. The input layer is represented by the bag-of-words vector that is inputted into several hidden layers with ReLU (Rectified Linear Unit) activation functions to improve the ability to learn. The output layer utilizes the softmax activation function to output the probability distribution over different categories of response. The network is structured to generalize well from training data so that it can effectively deal with varied user queries. The architecture of the model is important in facilitating the chatbot to provide relevant and context-sensitive responses.

D. Training the Model

The chatbot model is trained on the PyTorch framework, which is flexible and efficient in deep model development. Categorical cross-entropy loss function is used to quantify classification errors and drive optimization. To optimize learning and convergence rate, the Adam optimizer is used, updating model weights dynamically from gradients. Training is done by providing the neural network with labeled input-output pairs, which enables it to learn the association between input patterns and related responses. In several iterations, the model improves its capacity to identify user queries and produce contextually relevant responses.

E. Frontend Interface

The frontend interface is the chief medium of communication between the users and the chatbot. The frontend interface has been developed in the form of a web platform, where the users can submit their queries using a text input field. The chatbot takes the queries real-time and renders responses dynamically. A minimalist and user-friendly interface provides easy accessibility and effective interaction between the users and the system. In addition, features like message history and response refinement make the interface more usable, enabling users to interact with the chatbot better. The frontend is responsible for ensuring that user interactions are processed efficiently and smoothly.

F. Backend Model

The backend model is responsible for processing user queries and generating responses based on the trained neural network. Implemented in Python, it incorporates NLTK for text preprocessing, including tokenization, stemming, and intent classification. The backend follows a structured workflow:

- Tokenizing and stemming user input to standardize text processing.
- Checking if the query matches a previously stored question in the dataset.
- If a match is found, the chatbot retrieves and returns the corresponding stored response.
- If no direct match is found, the chatbot forwards the query to the Gemini API for external processing. • Once the API provides a response, it is presented to the user for validation.
- If the user accepts the response, it is stored in the model's database for future use.

This approach allows the chatbot to continuously improve by learning from user interactions and refining its response accuracy over time.

G. External API Integration with Gemini API

To manage unknown questions and provide thorough coverage, the chatbot is integrated with the Gemini API. If the chatbot finds an unknown question that is not in its database, it sends the query to the external API. The API processes the input, gets the appropriate information, and returns a response. Users are provided with an option to ask for different responses if the first answer is not acceptable. This integration allows the chatbot to provide contextually relevant and current information, even for subjects outside of its pre-defined knowledge base.

IV. IMPLEMENTATION

A. Data Preparation

Training data is saved in JSON format to facilitate structured pattern organization, tags, and responses. Preprocess-ing is done by tokenizing text data through NLP methods, then stemming and lowercasing for uniform word form standardization. A bag-of-words representation is created to transform text into numerical feature vectors. Stop words and punctuation are also eliminated to eliminate noise and enhance model efficiency. These preprocessing techniques improve the chatbot’s capability to accurately recognize and classify user queries.

B. Neural Network Training

The model for the chatbot is trained with a feedforward neural network based on PyTorch. The training dataset is varied query-response pairs for thorough learning. The training steps include forward propagation, backprop-agation, and optimization of weights with the Adam optimizer. The loss function employed here is categorical cross-entropy to quantify the difference between the predicted and actual response classes. The model is trained across several epochs, iteratively adapting weights to minimize loss and enhance accuracy. When training is finished, the trained model is stored and ready for deployment. This is illustrated in Fig. 3.

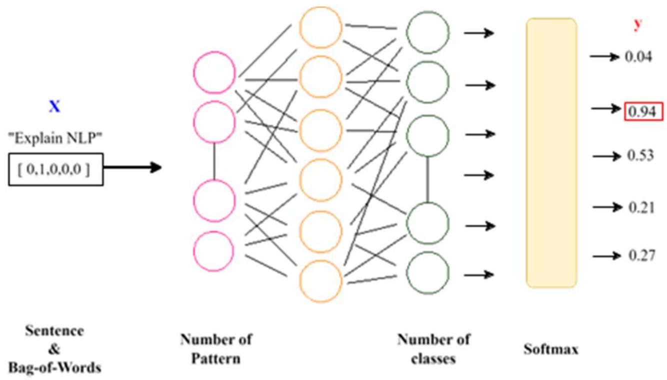

Fig. 3. Feed Forward Neural Network

C. Chatbot Deployment

The learned chatbot model is combined with a simple interface for query processing in real-time. When the user provides a question, the system tokenizes and transforms it into a numerical vector based on the bag-of-words approach. The vector is then given as input to the neural network, which outputs the most likely response category. On obtaining a match, the chatbot reads the stored response and presents it to the user. Where there is no direct correspondence, the chatbot sends the question to the Gemini API in order to create a corresponding response. This dual approach guarantees complete query processing, increasing user experience and chatbot effectiveness.

The following Fig. 4 & Fig. 5, illustrate the chatbot’s response process.

V. RESULTS

The suggested model was assessed based on an 80-20 data split, where 80% of the dataset was used for training and 20% for testing. This division provides a way that the model will learn from the majority of the data while holding another portion back to test the generalization of the model. The model’s performance was evaluated with the help of typical classification metrics, such as accuracy, precision, recall, and F1-score, to gain a complete insight into its performance in classification tasks.

The model showed a remarkable accuracy of 99.20% in the test set, which portrays its ability to classify instances appropriately with minimal discrepancies. The very high accuracy reflects that the model is well trained and can easily generalize to the unseen data. To assess it further, macro-averaged and weighted-averaged measures were computed for precision, recall, and F1-score as well.

The macro-averaged precision, recall, and F1-score were 98.84%, 99.02%, and 98.78%, respectively. These measures reflect that the model has balanced performance across all classes, irrespective of class distribution. The macro-averaged measures give equal weight to each class, so even a minority class contributes equally to the assessment, and thus it is an effective measure when class imbalance could be an issue.

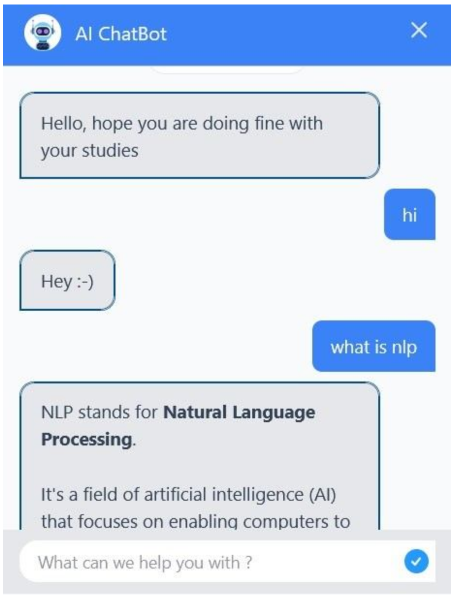

Fig. 4. Chatbot responding to a simple query

Fig. 5. Chatbot providing a response after neural network processing

Moreover, the precision, recall, and F1-score with weighted-averaging were 99.25%, 99.20%, and 99.10%, respectively. These measures are especially convenient for datasets with more frequent appearance of some classes over others. Weighted averages take into account the contribution of every class depending on its frequency to ensure that the ultimate evaluation indicates the actual influence of various class distributions.

The uniformly high values for all the metrics assure the reliability and strength of the model in classification problems. The high precision rate ensures that the model gives extremely few false positives, and the high recall rate ensures that it identifies most of the instances which are of relevance, giving low false negatives. The F1-score, being a balance of precision and recall, also confirms the model's effectiveness in prediction.

In general, the findings demonstrate the efficiency of the model in solving classification issues with great accuracy and minimal mistakes. The high evaluation measures indicate that the model is perfectly optimized for the task at hand and can be used confidently in real-world scenarios involving accurate and trustworthy classification.

TABLE I. CLASSIFICATION REPORT

Classification Report			
	Precision	Recall	F1 score
Accuracy			0.99204
macro avg	0.98841	0.99020	0.98788
weighted avg	0.99248	0.99204	0.99098

VI. CONCLUSION AND FUTURE WORK

Building a chatbot enabled by artificial intelligence for academic support shows remarkable scope in enhancing learning engagement and availability of educational materials. With natural language processing skills like tokenization, stemming, and representation with bag-of-words, the chatbot categorizes the input from the users into already designed intents with optimal accuracy. Combining the inclusion of a PyTorch-trained feedforward neural network brings much-needed accuracy and relevant results in the responses. Furthermore, integrating the Gemini API enables the chatbot to process never-before-seen queries, enhancing flexibility. This integration minimizes human intervention in answering repetitive student questions, making it a more effective and engaging learning environment.

Although as effective, the chatbot still has limitations in responding to highly ambiguous questions and deep con-textual understanding. As it largely depends on pre-defined patterns and machine learning-based classification, generating dynamic responses is limited in its capability. For its future enhancement, the training dataset of the chatbot will be further expanded with more varied sentence structures and domain-specific vocabulary. Further-more, reinforcement learning algorithms may be incorporated for the chatbot to learn from user responses with time. Employing cutting-edge NLP architectures like transformer models, BERT, and GPT will contribute to better contextual understanding and more accurate responses in the chatbot.
Improving user experience is another important area of future development. Adding voice-based interaction and multimodal support will increase accessibility and usability. A specific mobile app will also enhance availability, allowing students to interact with the chatbot at their convenience. Adding sentiment analysis will enable the chatbot to personalize responses according to user emotions, providing a more personalized learning experience. Incorporation of a user feedback system through the rating of responses will assist in gradually improving the performance of the chatbot. These improvements will lead to the evolution of an extremely intelligent and adaptive chatbot, revolutionizing the manner in which students can access educational material.

REFERENCES

[1] S. Kesarwani, Titiksha, and S. Juneja, "Student Chatbot System: A Review on Educational Chatbot," in Proceedings of the 7th International Conference on Trends in Electronics and Informatics (ICOEI 2023), 2023, pp. 1578–1583. IEEE. 10.1109/ICOEI56765.2023.10125876.

[2] Bafna, S. A., et al., Review on Study and Usage of MERN Stack for Web Development, International Jour-nal for Research in Applied Science Engineering Technology (IJRASET), Vol. 10, Issue II, pp. 178-186, February 2022. DOI: 10.22214/ijraset.2022.40209

[3] Naing Naing Khin and Khin Mar Soe, "Question Answering based University Chatbot using Sequence to Sequence Model," Proceedings of the 23rd Conference of the Oriental COCOSDA, pp. 5-7, November 2020, Yangon, Myanmar.

[4] Mishra, S. K., Chaudhary, S., Sanjay, R., Jain, P. K. (2024). Enhancing website through chatbot integra-tion. 2024 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC). IEEE. https://doi.org/10.1109/ICECCC61767.2024.10592872

[5] Huddar, A., Kolekar, U. D., Bysani, C., Upadhyaya, K., Suchak, C. (2020). Dexter the college FAQ chatbot. 2020 IEEE International Conference on Convergence to Digital World – Quo Vadis (ICCDW). https://doi.org/10.1109/ICCDW45521.2020.9318648

[6] Mahanan, W., Thanyaphongphat, J., Sangamuang, S., Sawadsitang, S. (2022). College Agent: The Machine Learning Chatbot for College Tasks. 2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT NCON). IEEE. https://doi.org/10.1109/ECTIDAMTNCON53731.2022.9720420

[7] Caballero Castellanos, J. R., Cardona, M., Ordonez-Avila, J. L. (2023). Chatbot for validation of research topics for engineering students. 2023 IEEE 41st Central America and Panama Convention (CONCAPAN XLI), 1-6. https://doi.org/10.1109/CONCAPANXLI59599.2023.10517536

[8] Javeed, A. (2019). Performance Optimization Techniques for ReactJS. In Proceedings of the 2019 IEEE International Conference on Software Engineering and Computer Systems (pp. 1-6). IEEE. DOI: 10.1109/SECS.2019.978-1-5386-8158-9.

[9] Liang, L., Zhu, L., Shang, W., Feng, D., Xiao, Z., Express Supervision System Based on NodeJS and MongoDB, Proceedings of the IEEE International Conference on Information Science, pp. 607-612, May 2017.

[10] Gu, Y., Wang, X., Shen, S., Wang, J., Kim, J.-U. (2015). Analysis of Data Storage Mechanism in NoSQL Database MongoDB. In 2015 International Conference on Consumer Electronics-Taiwan (ICCE-TW) (pp. 70-73). IEEE. DOI: 10.1109/ICCE-TW.2015.978-1-4799-8745-0.

[11] Jindal, A., Shrivastava, V., Pandey, A., Benefits of Use of MongoDB as Database: A Comprehensive Study, International Journal of Research Publication and Reviews, Vol. 5, Issue 4, pp. 2559-2562, April 2024.

[12] Soni, S., Ambavane, M., Ambre, S., Maitra, S., A Comparative Study: MongoDB vs MySQL, International Journal of Scientific Engineering Research, Vol. 8, Issue 5, pp. 120-123, May 2017.

[13] Min-Yuh Day. Chao-Yu Chen, Artificial Intelligence for Automatic Text Summarization, IEEE Inter-national Conference on Information Reuse and Integration for Data Science, pp. 478-484, 2018. DOI: 10.1109/IRI.2018.00076.

[14] Ranganathan, J., Abuka, G., Text Summarization using Transformer Model, 2022 International Conference on Social Networks Analysis, Management and Security, pp. 979-983, 2022. DOI: 10.1109/SNAMS52071.2022.00170.

[15] Ji, S., Yang, B., Abstractive Text Summarization Based on Long-Short Transformer, 2023 IEEE World AI IoT Congress (AIIoT), pp. 691-696, 2023. DOI: 10.1109/AIIoT58121.2023.10174260.

[16] Widyassari, A. P., Affandy, Noersasongko, E., Fanani, A. Z., Syukur, A., Basuki, R. S., Litera-ture Review of Automatic Text Summarization: Research Trend, Dataset and Method, 2019 Interna-tional Conference on Information and Communications Technology (ICOIACT), pp. 1-6, 2019. DOI: 10.1109/ICOIACT46717.2019.8976668.

[17] Kumari, A., Raviraj, A., Murari, A. K., Mukherjee, I., Real Life Implementation of Text Summarization Technique, 2023 International Conference on Data Science and Communication (IconDSC), pp. 1-5, 2023. DOI: 10.1109/IconDSC56688.2023.10123456.
[18] Lin, M.-H., Chen, H.-C., Liu, K.-S., ”A Study of the Effects of Digital Learning on Learning Motivation and Learning Outcome,” EURASIA Journal of Mathematics, Science and Technology Education, vol. 13, no. 7, pp. 3553-3564, 2017. DOI: 10.12973/eurasia.2017.00744a.
[19] Mohan, G. B., Kumar, R. P., Parathasarathy, S., Aravind, S., Hanish, K. B., Pavithria, G., ”Text Summariza-tion for Big Data Analytics: A Comprehensive Review of GPT-2 and BERT Approaches,” in Data Analytics for Internet of Things Infrastructure, R. Sharma et al. (Eds.), Springer Nature Switzerland AG, 2023. DOI: 10.1007/978-3-031-33808-3 14.
[20] Mohan, G. B., Kumar, R. P., ”A Comprehensive Survey on Topic Modeling in Text Summarization,” in Micro-Electronics and Telecommunication Engineering, D. K. Sharma et al. (Eds.), Lecture Notes in Net-works and Systems, vol. 373, Springer Nature Singapore Pte Ltd, 2022. DOI: 10.1007/978-981-16-8721-1 22.