

CHATBOT USING ARTIFICIAL INTELLIGENCE AND NATURAL LANGUAGE PROCESSING

Arivusudar S¹, Kalaivanan K², Vigneshwaran S³, Santhosh S⁴, Reeta joldrine A⁵

B.E, CSE, Sri Ramakrishna college of Engineering, Perambalur, India¹

B.E, CSE, Sri Ramakrishna college of Engineering, Perambalur, India²

B.E, CSE, Sri Ramakrishna college of Engineering, Perambalur, India³

B.E, CSE, Sri Ramakrishna college of Engineering, Perambalur, India⁴

Assistant Professor, CSE, Sri Ramakrishna college of Engineering, Perambalur, India⁵

Abstract: A chatbot is a computer program that makes conversations with humans using Artificial Intelligence (AI) in messaging platforms. Every time the chatbot gets input from the user, it saves input and response which helps the chatbot with little initial knowledge to evolve using gathered responses. We can implement an online chatbot system to assist website users. Using this tool, we can access files easily instead of going through different modules. Artificial Intelligence methods such as Natural Language Processing (NLP). In order to achieve quality education as a defined one of the sustainable goals, it is necessary to provide information about the education system according to the stakeholders' requirements. The process to obtain the information about university/institute is a critical stage in the academic journey of prospective Students who are seeking information about the specific courses which makes that university/institute unique. This process begins with exploration to general information about universities through websites, rankings, and brochures from various sources. Most of the time, information available on different sources leads to discrepancies and influences student's decisions. By addressing inquiries promptly and providing valuable information, universities can guide individuals in making informed choices about their academic future. To address this, the chatbot application is the most effective tool to be implemented and make it functional on university's functional website. A chatbot is an artificially intelligent tool which can interact with humans and can mimic a conversation. This tool can be implemented using advanced Natural Language Processing (NLP) models to provide the pre-defined answers to the student's queries. Chatbot is very helpful for query resolution during the counseling process of the institute as it will provide official/uniform information and can be accessed 24×7 . Therefore, the aim of this research work was to implement a chatbot using various NLP models and compare them to identify best one.

Keywords: AI bot, Virtual assistant, Customer support bot, Auto reply, Live chat, Message bot, Online assistant

I. INTRODUCTION

A chatbot using NLP (Natural Language Processing) is a smart program that can understand and respond to human language in a natural, conversational way.Unlike basic rule-based chatbots that only reply to specific commands, NLP chatbots can understand the meaning, context, and intent behind a message. A chatbot using Natural Language Processing (NLP) is an AI-powered program that can understand, interpret, and respond to human language in a natural, conversational way. It uses NLP techniques to analyze text or speech, recognize user intent, and generate relevant replies, enabling more human-like interactions. Chatbots are essential for counseling in engineering institutes for a number of reasons. The first benefit of these chatbots is that they make counseling services more accessible by eliminating time and location constraints and offering engineering students who could be facing difficulties in their personal, professional, or academic lives instant assistance.Due to chatbots' real-time functionality, kids may get help whenever it's convenient for them, which encourages a proactive approach to problem solving.Chatbot is an artificially intelligent entity that can interact with humans and mimic conversations. The input to the chatbot could be text-based or spoken (voice-based queries).Chatbots are majorly used for information retrieval.

II. LITERATURE SURVEY

The integration of chatbots into counseling services within engineering institutes has gained traction as a means to provide accessible and timely support to students facing academic, personal, or career-related challenges. Some of the chatbots

IJARCCE



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 $\,\,st\,$ Peer-reviewed & Refereed journal $\,\,st\,$ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14542

used in counseling process are briefed and comparative analysis is explained in this section. A study by Davis and Smith [1] emphasized the potential of chatbots in counseling to overcome geographical and time constraints. The implementation of chatbots allowed students to seek guidance beyond traditional office hours, leading to increased accessibility. The proposed chatbot emphasizes on the set of questions which are frequently getting asked during counselling process. Research by Johnson and Lee [2] explored the role of chatbots in providing emotional support to engineering students. The study found that chatbots equipped with sentiment analysis capabilities effectively identified and responded to students' emotional states, contributing to a supportive environment. The development proposed chatbot focuses on all types of questions which are generally need to be answered before taking decision to join any university for the course. This decision process is quite an emotional for the students and parents hence appropriate questions to be answered is a crucial task. Career-oriented chatbots were investigated by Patel et al. [3] for offering personalized career guidance to engineering students. Results indicated that students who engaged with career-focused chatbots demonstrated a clearer understanding of their career paths and increased confidence in their choices. The proposed chatbot mainly focuses on various preferences of students to decide the university for the admission. The work of Chang and Wang [4] delved into privacy concerns associated with counseling chatbots. The study highlighted the importance of secure communication

III. METHODOLOGY

The development of chatbots involves defining objectives, understanding user requirements, selecting a suitable platform, integrating with university systems, and deploying across various channels. The overview of an developed solution is explained step-wise as follows:

• **Preparation of Questions Related to Counseling Process:** The solution begins with a meticulous preparation of questions related to the counseling process. This involves understanding the varied needs and concerns of users, including prospective students, and parents. The question preparation phase includes input from counselling experts to ensure the chatbot is equipped to address a wide range of inquiries related to admissions, academic programs, career guidance, and support services.

• Handling Various Forms of the Same Query Using Semantic Analysis: To enhance the effectiveness of the chatbot, semantic analysis is employed to handle various forms of similar queries. Through natural language processing techniques, the chatbot is trained to recognize the semantic meaning behind different expressions of the same question. This enables the chatbot to provide consistent and accurate responses regardless of how users phrase their queries, ensuring a more user-friendly and efficient interaction.

• **Capability to Process All Types of Questions:** Simple to Complex: The developed solution ensures the chatbot's versatility in processing questions of varying complexity. Whether users have straightforward queries about admission deadlines or complex inquiries regarding academic policies, the chatbot is designed to comprehend and respond appropriately. The system is equipped with an extensive knowledge base and advanced algorithms to tackle a diverse set of questions, providing comprehensive support across the counselling spectrum.

• **Implementation and Analysis of Chatbots Using Various Technologies:** The implementation of the chatbot solution is characterized by the use of various technologies to optimize performance and user experience. Technologies such as natural language processing (NLP), machine learning algorithms, and possibly deep learning algorithms.

The pre-processing steps are explained below:

• **Converting to Lowercase**: The raw text is changed to lowercase to avoid numerous variants of the same word, and all the terms, regardless of their casing, are standardized/normalized to lowercase so they can be counted together.

• **Tokenization**: Tokenization is dividing a text stream into meaningful elements called tokens. Tokens can be words, sentences, or any other part of the sentence.

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering





FIGURE 1. Methodology for chatbot development.

Figure 1: depicts the methodology for developing a chatbot. The dataset is created by collecting all the questions asked about a particular technical university from various social media portals and the university's students and faculty. Answers are obtained for these questions from authorized sources from the university. The dataset has around 250 questions formed in differentways. The following methodology is used for using these questions and answers to design a chatbot. In the first step, raw data is pre-processed and converted into a format that is easier and more effective for further processing steps. It also normalizes the raw data in the dataset and reduces the number of features in the feature set. This leads to a decrease in the complexity of fitting the data to each classification model.

The pre-processing steps are explained below:

M

• **Converting to Lowercase**: The raw text is changed to lowercase to avoid numerous variants of the same word, and all the terms, regardless of their casing, are standardized/normalized to lowercase so they can be counted together.

• **Tokenization**: Tokenization is dividing a text stream into meaningful elements called tokens. Tokens can be words, sentences, or any other part of the sentence.



FIGURE 2. Structure of a neural network.

Figure 2: information in only one direction. Contrary to feed-forward neural networks, Long Short-Term Memory (LSTM) uses recurrent neural networks, where the information flow is nonlinear. While dealing with sequential data or data with a temporal link, LSTMs are favored. However, LSTMs have disadvantages: they are comparatively slow and require a sizeable high-quality dataset to get acceptable results. Once the model is ready, the next is to design a chatbot. There are many ways of creating a chatbot; all of them will have different performances. Even if the same query is fed to all the chatbots, their responses might be different.



Impact Factor 8.102 $\,\,st\,$ Peer-reviewed & Refereed journal $\,\,st\,$ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14542

the following five chatbots are explained:

- Smart Bot: In this chatbot, the neural network is created using TensorFlow (TfLearn)
- Sam: In this chatbot, the neural network is created using PyTorch
- Big Mouth: This chatbot is created using TF-IDF Vectorization
- Hercules: This chatbot is created using Sequential Modeling
- ALICE: This chatbot is created using AIML

SMART BOT: NEURAL NETWORK USING BOOSTING (TFLEARN) :

As the name suggests, TensorFlow runs computations based on tensors. In machine learning, a tensor is a generalization of vectors and matrices, represented as an n-dimensional array of a base datatype. All elements of a tensor always have the same data type. The explanation of Algorithm 1 is given in the following paragraphs. The data is stored in *intents. json* file, and it contains a list of goals. Each plan or class has a tag, a pattern, and a response. The "tag" defines the purpose or class. The "pattern" lists possible questions for the corresponding category.

The "response" is a list of possible answers to the questions of that "tag." The chatbot will take the message from the user, identify the "tag" of the message, and give the corresponding response. Every "pattern" of every "intent" is tokenized using nltk.word_tokenize() and is appended to the "words" list. All the tags are stored in the "labels" list. "words" is a list containing all the words in the database. Every word in "words" is converted to lowercase using the *lower()* function.

Algorithm 1: Algorithm for Chatbot Using Boosting

 $data \leftarrow load \ data \ from \ JSON \ file$ If the model has already been trained, load the variables from the pickle file Initialize lists words, labels, docsx, docsy while *intent* \in *data* do while *pattern* ∈ *intent* **do** wrds \leftarrow tokenize words in the pattern Apppend wrds to words Append wrds to docsx Append tag of the intent to docsy end while if tag \in labels then Append tag of the intent to labels end if end while remove punctuations from "words" Stemming and converting to lowercase of , "words" and store in "words"sort"words"and"labels" Initialize lists: training, output while sentence \in docsx **do** Initialize "bag" (bag of words) stem every word in "sentence" and store in "wrds" while word \in words do if word \in wrds then append 1 to bag else append 0 to bag end if end while Append "bag" to "training" outputrow[la bels.index(d ocsv[x]) = 1Append "bag" to "training" Append outputrow to output end while convert "training" and "output" to array and save

the variables in pickle file create a Deep Neural Network using tflearn. The size of the input layer is same as the size of the bag of words. Then add 2 hidden layers (fully connected) of 8 neurons each. The size of the output layer is equal to the number of tags set "number of epochs" to 1000



Impact Factor 8.102 $\,$ $\!$ $\!$ $\!$ Peer-reviewed & Refereed journal $\,$ $\!$ $\!$ $\!$ Vol. 14, Issue 5, May 2025 $\,$

DOI: 10.17148/IJARCCE.2025.14542

NEURAL NETWORK USING RANDOM FORESTING

The explanation of Algorithm 2 is given in the following paragraphs. The data is stored in *intents. json* file, and itcontains a list of intents. Each intent or class has a tag, a pattern, and a response. The "tag" defines the intent or class. The "pattern" is a list of possible questions for the corresponding class. The "response" is a list of possible answers to the questions of that "tag." The chatbot will take the message from the user, identify the "tag" of the message, and give the corresponding response. Pre-processing steps are applied to the data. Every question of every intent is tokenized using *nltk.word_tokenize()* and is appended to the "all_words" list. Every unique tag is stored in the "tags" list. Now, "all_words" is a list that contains all the tokenized words of the dataset, and "tags" is a list that contains all the tags of the database. All the punctuation tokens are removed; every word in the dataset is converted to lower case using the *lower()* function, and the words are stemmed using *PorterStemmer().stem()* function from nltk. "all_words" list is sorted using sorted(all_words) function and all the duplicate words are removed using *set(all_words)* function. "tags" list is also sorted. To create a bag of words,

Algorithm 1: Algorithm for Chatbot Using Randam Forest And Clustering

import nltk from nltk.chat.util import Chat, reflections pairs=[["(.*)hi|hello(.*)", ["Hi!, How can I assist you today?"]], ["(.*)how are you?(.*)"]["I'm doing well, thank you! How can I help you today?"]], ["(.*)what is your name?(.*)",["I am a chatbot created to assist you."]], ["(.*)what can you do?(.*)",["I can help answer your questions or assist with tasks."]], ["(.*)help|Can you help me?(.*)", ["Sure! How may i help you"]], ["(.*)what is your favorite color?(.*)", ["I don't have personal preferences, but I can talk about any color you like!"]], ["(.*)What time is it?(.*)", ["I don't have the ability to check the time. Please check your device's clock."]], cap = cv2.VideoCapture(0) nltk.download('punkt') while True: ret, frame = cap.read() cv2.imshow('frame', frame) if cv2.waitKey(1) & 0xFF == ord('q'): break cap.release() cv2.destroyAllWindows()

IV. SYSTEM MODULES

[1] Text Preprocessing Modules:

The Text Preprocessing Module is a part of the chatbot pipeline that cleans and prepares raw user input for Natural Language Processing (NLP). This is a critical step that enhances the accuracy of tasks like intent recognition, entity extraction, and response generation. Clean and normalize text. Chatbot common support of Lowercasing ,Removing punctuation,Tokenization The ChatbotTools in nltk, spaCy, re (Regex).

[2] **1. Tokenization:**

- Purpose: Tokenization splits a piece of text into smaller units, such as words, phrases, or sentences.
- Libraries: o NLTK: nltk.word_tokenize(), nltk.sent_tokenize() o spaCy: spacy.load('en_core_web_sm').tokenizer
 - Hugging Face Transformers: transformers.AutoTokenizer

Impact Factor 8.102 $\,\,st\,$ Peer-reviewed & Refereed journal $\,\,st\,$ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14542

- Example: "Hello, how are you?" → ["Hello", ",", "how", "are", "you", "?"]
- [3] 2. Lowercasing:

ΝM

- **Purpose:** Convert all the text to lowercase to ensure consistency (i.e., "Hello" and "hello" are treated the same).
- Libraries:
 - Python's built-in str.lower() method
 - **Example:** "HELLO" \rightarrow "hello"

[4] **3. Removing Punctuation:**

- Purpose: Removing unnecessary punctuation marks to simplify the text and avoid noise.
- Libraries: o Regex: Using Python's built-in re module to remove punctuation.
- **Example**: "Hello!!!!" \rightarrow "Hello".

V. RESULT ANALYSIS

The results of all the implemented chatbot are represented in accuracy and validation. The first section contains confusion matrices and accuracy, calculated based on a sample training dataset. In the second section, 150 queries were implemented on all the chatbots, and their responses were observed to check if they categorized the query correctly or not. Lastly, the last section contains screenshots of the conversation with the bots.

CONFUSION MATRICES AND ACCURACY BASED ON SAMPLE TEST DATASET

A test dataset having 144 queries is used to test the chatbot models. Confusion matrices and accuracies are calculated using sklearn metrics library. (using confusion_matrix and accuracy_score functions) The neural network was created using TensorFlow in this model, and multiple pre-processing steps were applied.

The Lancaster Stemming algorithm was used in the preprocessing phase, which is more accurate. Furthermore, the softmax activation function is applied to the output layer, increasing the neural network's performance.

In this model, the neural network is not created. Instead, TF-IDF Vectorization converts every sentence into a vector, and Cosine Similarity calculates the similarity between every sentence and the query. This model needs to understand the meaning of the query; it simply finds the most similar sentence. Table 1 is the confusion matrix of Big Mouth.

e 😜		C: \Users \SHARMA \Downloads \Chatbot using NLP 🔹 🗁 🛧			
± 8 8	ં 🗊 વ	c		≡	
Name 🔺	Туре	Size	Value		
chatbot	chat.util.Chat	1	Chat object of nltk.chat.util module		
pairs	list	31	[['(.*)hi hello(.*)', []], ['(.*)how are you?(.*)', []], ['(.*)w		
reflections	flections dict		{'i am':'you are', 'i was':'you were', 'i':'you', 'i'm':'you are', 'i'		
		l	Help Variable Explorer Plots Files		

This shows a Python IDE displaying variables for a chatbot program, including 'chatbot', 'pairs', and 'reflections' with their respective types and values. he image shows a Python IDE, likely Spyder, displaying variables in a chatbot program.

The variables and their values are:

chatbot: A chat object from the nltk.chat.util module.



Impact Factor 8.102 $\,\,st\,$ Peer-reviewed & Refereed journal $\,\,st\,$ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14542

pairs: A list of 31 items, likely containing pairs of patterns and responses used by the chatbot. The image shows examples such as ['(.)hi hello(.)', [...]] and ['(.) how are you?(.)', [...]].

reflections: A dictionary with 16 key-value pairs, used for reflecting user input. Examples shown are {'i am': 'you are', 'i was': 'you were', 'i':'you', 'i'm': 'you are'}.



An algorithm is a precise, step-by-step procedure or set of instructions designed to solve a problem or perform a computation. It's a finite sequence of instructions that, when executed, transforms input into output. Algorithms are essential in computer science, mathematics, and various other fields for automating tasks and problem-solving.

Console 3/A ×		Û		=
Suerb				
Sure! How may i help you				
>what is the weather like?				
I cannot check the weather. You m	ight want to use a weather app.			
>clustering alogrithms				
we have two clustering those are:	K_means Clustering and Agglomerative clus	tering		
>code for decisiontree regression				
from sklearn.tree import DecionTr	reeRegressor			
tree=DecisionTreeReg	ressor()			
<pre>tree.fit(X_train,y_t</pre>	rain)			
<pre>y_pred=tree.predict(</pre>	X_train)			
by using these we fi	rst install the sklearn by uisng pip (pip	install		
sklearn), In place of X train, and	y train, you can assign your own train data	Inplace	of	
decisiontree you can use the any	other alogrithms as well like svr,knn,rand	omforest,	boosti	ng
alogrithms				
> _	Activate Window	S		
	IPython Console History to Settings to activ	ate Windo)WS.	
conda: base (Python 3.10.9) 🛭 😵 Completi	ions: conda × LSP: Python : Line 58, Col 50 UTF-8	CRLF R	W Mer	n 67%

The text in the image is a console output showing interactions and code snippets related to machine learning in Python. It begins with a user asking "what is the weather like?" to which the system responds that it cannot check the weather and suggests using a weather app. Following this, there are explanations and code examples related to:

Clustering algorithms: Mentioning K-means and Agglomerative clustering as two types.

Decision tree regression: Providing a code snippet using the sklearn library in Python, demonstrating how to implement a decision tree regressor. The code includes importing the DecisionTreeRegressor, fitting the model with training data (X_train, y_train), and making predictions. It also mentions installing sklearn using pip and the possibility of substituting other algorithms like SVR, KNN, Random Forest, and boosting algorithms.

from sklearn.tree import DecisionTreeRegressor tree = DecisionTreeRegressor() tree.fit(X_train, y_train) y_pred = tree.predict(X_train)



Impact Factor 8.102 $\,\,st\,\,$ Peer-reviewed & Refereed journal $\,\,st\,\,$ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14542

It also explains that sklearn needs to be installed using pip (pip install sklearn). Additionally, it notes that other algorithms like SVR, KNN, Random Forest, and boosting algorithms can be used instead of the decision tree.

VI. CONCLUSION

Chatbots powered by **NLP and AI** have transformed the way humans interact with technology. They can understand natural language, respond intelligently, and continuously learn from conversations. These smart systems are now widely used in customer service, healthcare, education, and more, offering faster, more efficient, and more personalized communication. As NLP and AI technologies keep improving, chatbots will become even more human-like, making digital interactions smoother and more effective. eural networks, TF-IDF vectorization, and pattern matching. In neural network-related models, pre-processing steps like converting to lowercase, stemming, lemmatization, tokenization, removing stop words, and creating a "bag of words" are applied to the training data before passing it through the neural network. A query is taken from the user; preprocessing steps are used to it, and it is passed through the model, which returns the list of probabilities that the query belongs to a certain intent. Hercules performs best among the five chatbots discussed in the project because it has sequential modeling designed to prevent overfitting training data. Furthermore, it is the only chatbot with any optimizer applied to it, improving its performance. Therefore, it can be concluded that a chatbot similar to Hercules can be implemented in real-time for university/institute counseling. This will be very helpful for the students.

REFERENCES

- [1]. 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.
- [2]. 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.
- [3]. S. Maher, "Chatbots&its techniques using AI:Areview," Int. J. Res. Appl. Sci. Eng. Technol., vol. 8, no. 12, pp. 503–508, Dec. 2020.
- [4]. 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.
- [5]. P. Qi,Y. Zhang,Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: APython natural language processing toolkit for many human languages," 2020, *arXiv*: 2003.07082.
- [6]. 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.
- [7]. A. Chandan, M. Chattopadhyay, and S. Sahoo, "Implementing chat-bot in educational institutes," *IJRAR J.*, vol. 6, no. 2, pp. 44–47, 2019. [8] D. Davis and J. Smith, "The potential of chatbots in counseling," *J. Counsel.*, vol. 5, no. 2, pp. 123–136, Apr. 2019. [9] R. Johnson and S. Lee, "Chatbots providing emotional support to engineering students," in *Proc. IEEE Eng. Educ. Conf.*, Austin, TX, USA,2020, pp. 45–50.
- [8]. A. Patel, Personalized Career Guidance for Engineering Students. Springer, 2021.
- [9]. H. Yang and Q. Liu, "User acceptance of counseling chatbots," *J. Comput.*, vol. 8, no. 3, pp. 210–225, May 2019. [Online]. Available:https://www.example-url.com
- [10]. B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," in *Proc. Int. Conf. Adv. Comput., Commun. Informat.(ICACCI)*, Sep. 2017, pp. 1525–1530.
- [11]. V. Sharma, M. Goyal, and D. Malik, "An intelligent behaviour shown by chatbot system," *Int. J. New Technol. Res.*, vol. 3, no. 4, 2017, Art. no. 263312.
- [12]. E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Mach. Learn. Appl.*, vol. 2, Dec. 2020, Art. no. 100006.
- [13]. S. Khan and M. R. Rabbani, "Artificial intelligence and NLP-based chatbot for Islamic banking and finance," *Int. J. Inf. Retr. Res.*, vol. 11, no. 3, pp. 65–77, Jul. 2021.
- [14]. C. Curry and J. D. O'Shea, "The implementation of a story telling chatbot," Adv. Smart Syst. Res., vol. 1, no. 1, p. 45, 2012.
- [15]. V. Tiwari, L. K. Verma, P. Sharma, R. Jain, and P. Nagrath, "Neural network and NLP based chatbot for answering COVID-19 queries," *Int. J. Intell. Eng. Informat.*, vol. 9, no. 2, pp. 161–175, 2021.
- [16]. S. S. Ranavare and R. Kamath, "Artificial intelligence based Chatbot for placement activity at college using DialogFlow," *Our Heritage*