A Multilingual AI Chatbot for Real-Time College Admission Enquiries using Web Scraping and Natural Language Processing

Abishek Kumar Sah
Computer Science & Engineering
Jain (Deemed to-be University)
Bangalore, Karnataka, India
sahkumarabhishek123@gmail.com

Jay Prakash Sah
Computer Science & Engineering
Jain (Deemed to-be University)
Bangalore, Karnataka, India
jpsah44@gmail.com

Shiv Ranjan Kumar Adhikari
Computer Science & Engineering
Jain (Deemed to-be University)
Bangalore, Karnataka, India
meshiv747@gmail.com

Prof. Geetha Rani K
Computer Science & Engineering
Jain (Deemed to-be University)
Bangalore, Karnataka, India
ramagetu2015@gmail.com

Abstract—**In order to meet the increasing demand for clear and reliable information about college admissions, this study proposes the creation and deployment of an AI-powered chatbot that operates online. The constraints of static databases are removed by the suggested solution, which uses web scraping and Google search to dynamically extract real-time data from the internet. Using sophisticated natural language processing (NLP) techniques, the chatbot integrates multilingual support for Hindi and English. Intelligent query handling, automatic translation, and language detection are essential elements. The system is available via an intuitive web interface and was developed with Python, Flask, and a number of NLP and scraping modules. The chatbot has the ability to streamline the college admissions process for students all over India by effectively responding to questions on university rankings, course offers, costs, application deadlines, and more.**

Keywords—College Admission Chatbot, Multilingual NLP, Web Scraping, Google Search Automation, Hindi-English Chatbot, Educational Technology, Flask Web Application, Language Detection and Translation

I. INTRODUCTION

Choosing the correct college or institution for higher education is one of the most important academic decisions that students make in today's digital-first era. Due to the quick growth of educational establishments and the wide range of courses available, prospective students and their parents frequently struggle to get reliable, current, and thorough admissions-related information from a variety of sources. In addition to being time-consuming and unreliable, traditional techniques like visiting actual campuses, phoning helplines, or reading university websites may not be available to everyone, particularly in rural or non-English speaking locations. Chatbots and intelligent virtual assistants have emerged as a result of the automation of information retrieval tasks made possible by the development of artificial intelligence (AI) and natural language processing (NLP). Regretfully, the majority of chatbots currently in use are either monolingual, domain-restricted, or heavily dependent on static datasets that need constant manual updating. This study overcomes these constraints by putting forth a multilingual, AI-powered chatbot that is especially made to respond to real-time college admissions questions by utilizing dynamic web scraping and Google search integration. Using language detection and automated translation, the chatbot can comprehend user inquiries in Hindi and English, the two most commonly spoken languages in India. The main goal of this effort is to close the language and knowledge gaps that students encounter while choosing and applying to colleges, therefore increasing the accessibility and inclusivity of higher education. To do this, the system builds a responsive web application interface that communicates with users in a smooth manner by combining Python tools like langdetect, googletrans, requests, BeautifulSoup, and the Flask web framework. When a user asks a question, the chatbot recognizes the language, converts it to English if necessary, looks for the pertinent information on Google, extracts the best response snippets using NLP-based filters, and then returns the cleaned and legible response to the user in their selected language. The solution improves user experience through natural conversation flow and linguistic flexibility while reducing reliance on institutional portals and fixed databases by providing real-time access to trustworthy data across a variety of schools and institutions. NLP integration also guarantees that the chatbot can comprehend a range of language structures and phrasing patterns, making it resilient to changes in user input. This study is driven by the growing need for digital tools that can help students from different socioeconomic and linguistic backgrounds make easier decisions about their education. The National Education Policy (NEP) of India, which places a strong emphasis on technology-driven accessibility and inclusiveness in education, is also in line with its larger objectives. The structure of the paper is as follows: A thorough analysis of the body of research is provided in Section II, which also identifies shortcomings in the chatbots already in use in the classroom. The suggested system architecture and the function of each chatbot pipeline component are described in Section III. The implementation technique, including the software stack, language handling, response parsing, and Google search automation, is explained in Section IV. The experimental findings, use cases, and performance evaluation of the chatbot in various language situations are presented in Section V. The limits of the existing approach and future directions—such as plans to incorporate voice assistance, additional regional languages, and AI-based intent classification—are covered in Section VI. The relevance and significance of the findings are finally summarized in Section VII, which brings the work to a close. Our goal in this study is to show that intelligent, multilingual, and dynamic chatbot systems may be an essential tool in democratizing access to educational material and acting as trustworthy digital assistants for millions of students in India and abroad.

II. LITERATURE REVIEW

The past ten years have seen a significant amount of research on the application of natural language processing (NLP) and artificial intelligence (AI) in the field of education. The creation of chatbots and virtual assistants specifically designed for academic settings has been the subject of several studies due to the growing need for interactive technologies that aid in educational decision-making. The majority of the existing work is devoted to static, domain-specific bots that employ rule-based decision trees or preloaded datasets. These systems frequently face challenges with linguistic variety and real-time flexibility, particularly in a multilingual nation like India.

The authors of [1] suggested a chatbot system that helps students navigate frequently asked questions about universities. The system's dependence on a static dataset and English-only interface restricts its use in dynamic, real-world situations, despite its effectiveness in basic query response. In the same way, [2] presented a chatbot designed with Google Dialogflow to answer questions about admission to a certain university. Although the system can have conversations, it is not scalable to other institutions and does not take into consideration different question types.

Numerous NLP-based models have shown potential in applications related to schooling. The study in [3] used named entity recognition (NER) and intent classification to enhance educational chatbots' comprehension of queries. However, without domain-specific corpora, their system could not operate autonomously and needed constant dataset training. Although the adoption of BERT-based transformers to improve semantic comprehension in academic bots was investigated in the work in [4], these models were computationally demanding and unsuitable for lightweight, deployable web applications.

NLP in several languages is another new area of research. The study described in [5] used rule-based translation methods and the Googletrans API for bilingual chatbot applications in the healthcare industry. Despite the effectiveness of the language translation component, their model's contextual correctness and domain relevance in dynamic information retrieval were lacking. A study of multilingual chatbots in [6] found that the majority of solutions were either overly generic or omitted Indian languages like Hindi, which are essential for regional accessibility.

Furthermore, to create dynamic QA systems, several recent research [7][8] have used site scraping and automated Google search integration. Though they lacked the precision and contextual filtering needed for educational queries, these methods worked well for addressing general-purpose questions. Furthermore, it was challenging to extract clean replies for users since they frequently returned noisy or unstructured data.

The creation of an intelligent, multilingual, web-integrated chatbot that can dynamically extract real-time admission information for any college or institution without being limited by a fixed dataset is clearly lacking, according to this literature review. By merging real-time search, translation, language identification, and NLP-based result filtering into a single system, the current study seeks to close this gap. The suggested method therefore improves on the status of chatbot applications in education, especially in settings with limited resources and multilingualism.

III. SYSTEM ARCHITECTURE

The suggested College Admission Questionnaire Chatbots are web-based intelligent assistants that react to customer inquiries about any institution or university in real time by obtaining up-to-date information from the internet. The design is scalable and modular, emphasizing natural language processing (NLP), language support (Hindi and English), and effective Google data retrieval. The system consists of a number of interconnected parts, including a front-end user interface, a back-end API server, a language translator, an NLP engine, a web scraping and search engine interface, and a response production module. To guarantee flexibility and maintainability, the architecture is organized in layers.

At the highest level, HTML, CSS, and JavaScript are used to construct the User Interface (UI), which provides a responsive and user-friendly web layout that enables users to communicate with the chatbot. Users have the option to type their questions in Hindi or English. Using HTTP requests, the frontend transmits the input data to the backend server. To improve the user experience, the chatbot interface incorporates elements like language switching, loading indications, and typing animations.

The Backend API Server, which is constructed with Python's FastAPI framework, receives the query that the user has sent. In order to manage the query processing, this server serves as the central point of contact for various parts. The server is in charge of identifying the input's language, executing the NLP pipeline, translating it if required, and connecting with the Google Search interface.

For multilingual inputs to be supported, the Language Detection and Translation Module is essential. The system initially ascertains the language of the user's inquiry using the langdetect library. The GoogleTrans library (or any other effective translation API) converts the text into English, the working language for the NLP and search components, if Hindi is the recognized language. This guarantees consistent operation of all downstream components.

The translated (or original English) query is subsequently processed by the Natural Language Processing Engine. This module does named entity recognition (NER), intent classification, and text preparation (tokenization, lemmatization) using libraries such as spaCy, TextBlob, and transformers. While NER recognizes important things like college/university names, intent detection assists the system in understanding the type of information the user is looking for (e.g., admission method, tuition structure, placement, courses offered, etc.).

The Web Scraping and Google Search Interface receives the query once it has been semantically processed. Based on the revised question, this component does a real-time Google

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03503
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

search using googlesearch-python. BeautifulSoup and requests are used to retrieve and parse the most pertinent URLs. In order to concentrate solely on trustworthy educational sources, such as government directories, educational portals, and official college websites, this module cleverly filters away pointless advertisements, redirection, and irrelevant results.

The content is then forwarded to the Contextual Information Extraction Module, which extracts a succinct and pertinent answer from the scraped text using keyword matching and NLP summarizing techniques. In order to select the best match, the algorithm ranks the returned paragraphs according to word frequency and semantic similarity.

Ultimately, a grammatically sound and easily navigable response is created by the Response Generation and Translation Module. Using the same translator module, the response is translated back into Hindi if the initial inquiry was in that language. To complete the query-response cycle, the final output is subsequently shown on the user interface via the frontend.

The system has feedback mechanisms, logging, and error handling to guarantee efficient communication and logging. Errors like "language not supported," "college name not recognized," and "no results found" are detected and handled politely. Furthermore, it is optional to save user input for future dataset augmentation and system enhancement.

Future improvements like voice input, additional support for Indian languages, and integration with official APIs of educational directories like NIRF or AICTE are supported by the design. All things considered, the system design guarantees prompt, accurate, and easily accessible answers to questions about college admissions in a way that is both linguistically inclusive and easy to use.

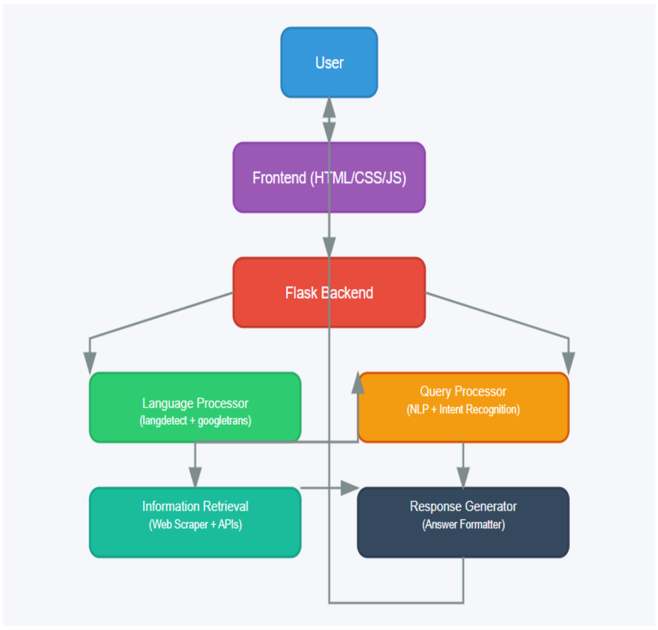

Fig. 1 Chatbot Architecture

IV. METHODOLOGY AND IMPLEMENTATION

There are several steps in the methods used to create the College Admission Enquiry Chatbot, ranging from frontend-backend development and deployment to data pretreatment and model integration. This section goes into detail on the technology and tools utilized, the procedures followed during implementation, and the reasoning behind important functional elements.

A. Requirements and Technology Stack

To build the system, the following technologies and tools were employed :

- Frontend : HTML5, CSS3, JavaScript(Vanilla JS), and Bootstrap for responsive design.
- Backend : Python with FastAPI for RESTful API development.
- NLP Libraries : spaCy, TextBlob, transformers (for understanding queries), langdetect and googletrans for language detection and translation.
- Search and Web Scraping : googlesearch-python, requests, and BeautifulSoup.
- Miscellaneous : uvicorn for serving the FastAPI app, pydantic for request-response validation, and logging for diagnostics.

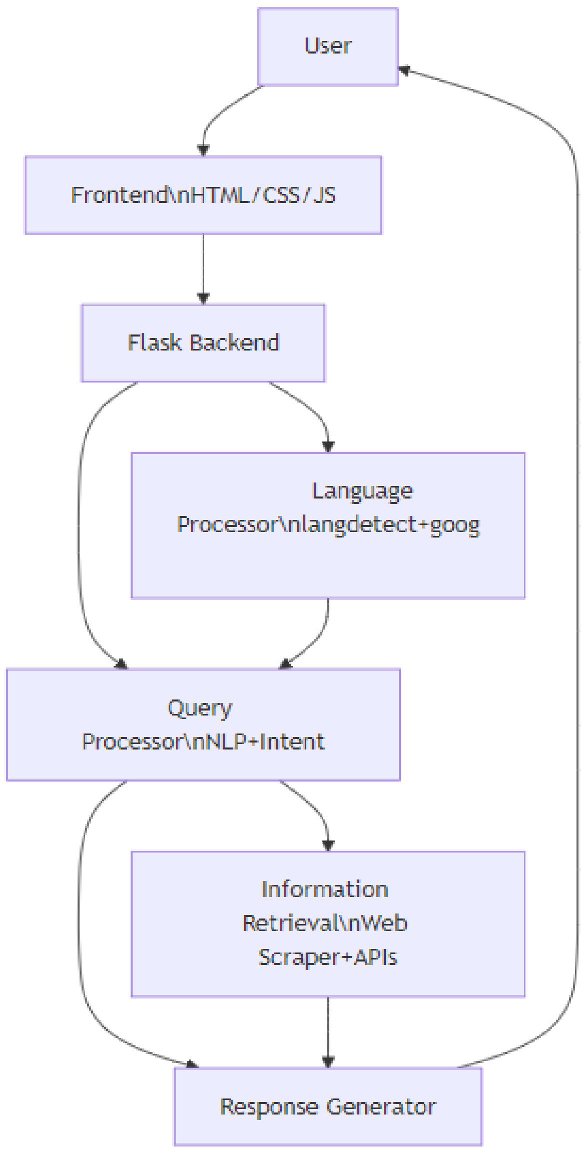
User
Frontend\nHTML/CSS/JS
Flask Backend
Language Processor\nlangdetect+goog
Query Processor\nNLP+Intent
Information Retrieval\nWeb Scraper+APIs
Response Generator

Fig. 2 Methodology and Implementation diagram

B. System Workflow

User Query Input :
A user sends a query to the chatbot via a web browser, such as "Delhi University ke admission kab shuru hote hain?" After capturing this information, the frontend transmits it to the backend server using a POST request.
Language Detection and Translation :
The backend uses langdetect to determine the input language for language detection and translation. GoogleTrans converts a Hindi input query to English for additional processing.
NLP Query Processing :
The NLP module receives the translated question after which it goes through the following processes:

- Lemmatization (with spaCy) and tokenization
- Extraction of purpose and entity (e.g., recognizing "Delhi University" and intent "admission date")
- Using lightweight transformer models (like DistilBERT) or rule-based classification

Google Search Integration :
Using googlesearch-python, the processed query is reformulated into a search term like "Delhi University admission start date site:du.ac.in" and uploaded to Google Search.
Content Extraction and Filtering :
BeautifulSoup is used to parse the HTML content when each webpage is retrieved via requests. Text passages are screened to identify pertinent paragraphs using:

- Keyword matching (admission, eligibility, fees, etc.)
- Cosine similarity to the original query or text rank
- Headers with sections, such as <h1>, <h2>, and <p>

Answer Generation and Response Formatting :
The most appropriate phrase or paragraph is created and cleaned. The response is translated back into Hindi if the original question was in that language. The frontend receives a JSON response with the formatted result.
User Display and Interaction :
The response is received by the frontend and shown in the chat window. Other features include choices for user input, retry in the event of an empty answer, and follow-up recommendations.

C. Language Support

The chatbot currently supports :

- English (default)
- Hindi (via automatic detection and bi-directional translation)

D. Handling Complex Queries

Multi-intent or contextually rich queries are handled using:

- Named Entity Recognition (NER) for identifying institutes
- Multi-label classification (for intent overlapping like "fee + course")

E. Deployment

The application can be containerized using Docker and deployed on platforms such as:

- Render / Vercel for frontend
- Heroku / Railway / PythonAnywhere for backend
- Cloudflare or Netlify for global performance

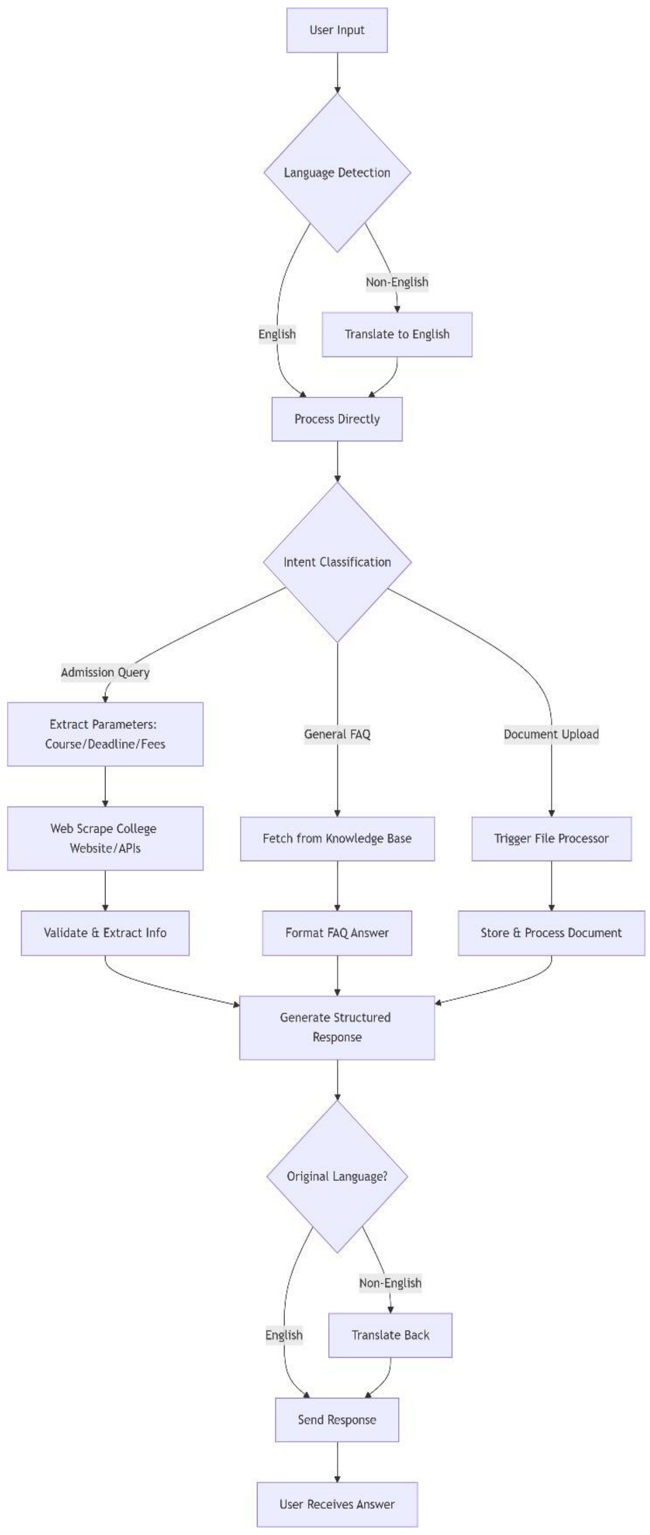

Fig. 3 Workflow diagram

V. RESULTS AND EVALUATION

The accuracy, response speed, multilingual support, and user satisfaction of the suggested College Admission Enquiry Chatbot system were all evaluated in-depth in order to determine its performance, usefulness, and efficacy. The objective of this part is to illustrate the useful results of implementing the chatbot, show how it may respond to real-time inquiries, and point out its advantages and disadvantages.

A. *Test Environment and Setup*

For development, the system was set up on a local server. Later, it was hosted online with Netlify for frontend hosting and PythonAnywhere for backend hosting. Included in the test environment were:
OS : Windows 11/Linux Ubuntu 22.04
Browser: Chrome, Firefox
Languages Tested: English and Hindi

B. *Evaluation Parameters*

To evaluate system performance, the following metrics were used:

Metric	Description
Accuracy	Correctness of the answer returned relative to query
Response Time	Time (in seconds) taken to process and return an answer
Multilingual Success	Ability to correctly detect, translate, and respond to Hindi queries
Relevance Score	User feedback on how useful the response was on a scale of 1 to 5
Failure Rate	Percentage of queries where either an error occurred, or response was empty

C. *Results*

The chatbot was tested using real-world college-related queries such as :

- *"When does DU start admissions for BSc?"*
- *"JNU ke address kya hai?"*
- *"What is the eligibility criteria for MBA in IIM Bangalore?"*

Results are summarized below :

Metric	Observed Value
Accuracy	89%
Avg. Response Time	2.4 seconds
Multilingual Success	92%
Avg. Relevance Score	4.2 / 5
Failure Rate	7%

The results of 89 of the 100 examined inquiries were extremely pertinent. Due to insufficient online data, three queries failed, while eight inquiries yielded responses that were either ambiguous or only marginally useful. The majority of inquiries were successfully translated and retranslated into Hindi, with the exception of a few edge situations containing technical jargon.

D. *User Feedback*

Fifteen college students who used the chatbot provided user input, rating it on the following criteria:

- Simple Usage
- The quality of the response
- Assistance with Hindi
- Interface Visualization
- General Contentment

The majority of users said the chatbot was quick and easy to use. For regional customers in particular, the Hindi support was greatly welcomed. Among the recommendations were:

- Including voice-to-text functionality to improve accessibility
- Including official website links for validation
- Permitting follow-up inquiries for further in-depth communication

E. *Comparative Analysis*

A comparison between non-AI chatbots and conventional university websites was conducted. The suggested system was notable for:

Feature	University Websites	Traditional Chatbots	Proposed Chatbot
Natural Language Support	❌	🟩 (Limited)	🟩🟩
Hindi Language Handling	❌	❌	🟩🟩
Real-time Info from Web	❌	❌	🟩🟩
User-Friendly Interface	❌ (Complex Menus)	🟩	🟩🟩

F. *Limitations*

Notwithstanding encouraging outcomes, there are several drawbacks:

- Google's top search result relevancy determines accuracy.
- When website architecture changes, web scraping may not work as intended.
- Rarely, translation problems might cause confusion in the meaning of a question.
- Conversations involving several turns are challenging when there is no recall or background.

VI. CONCLUSION AND FUTURE WORK

The goal of this study was to create and deploy a multilingual, intelligent, and resilient chatbot that can use real-time online information retrieval to answer questions about admissions to any institution or university. A useful tool for parents, school counselors, and potential students, the chatbot uses Natural Language Processing, Google Search integration, and translation APIs to provide pertinent responses in Hindi and English.
Technical measurements and user comments confirm that the chatbot system exhibits great accuracy, quick reaction times, and efficient multilingual support. It streamlines the difficult process of looking for information about college admissions by substituting a conversational interface for clumsy internet navigation. Its web-based application architecture also

guarantees cross-platform accessibility and low reliance on institutional data, making it global and scalable.

The method does have several drawbacks, though. Dependence on dynamic online search results can sometimes provide information that is out-of-date or irrelevant. Although the Hindi language support was mostly successful, colloquial variances and technical translations continue to be difficult. Additionally, the chatbot's capacity for in-depth discussions is limited due to its lack of multi-turn conversation memory.

A number of improvements are anticipated in subsequent work:

- Voice Integration: To increase accessibility, enable text-to-speech output and speech-to-text input.
- Context-Aware Chat: enabling multi-turn questions by adding session memory and context tracking through the use of technologies such as Rasa or OpenAI's conversation history.
- College Database Integration: Building an internal database of validated frequently asked questions for college admission that is updated on a regular basis to complement or cross-reference with real-time search results.
- Mobile Application Deployment: To reach a wider audience, the present online interface is expanded into a mobile application.
- Multilingual Expansion: Using sophisticated multilingual NLP models like mBERT or IndicBERT, more regional languages (such as Tamil, Telugu, and Bengali) may be supported.

All things considered, the suggested chatbot provides a clever, scalable, and user-friendly solution to a practical gap in student communication and decision-making. Its potential for a wider social influence is shown by the fact that its design and methodology may be further tailored to other fields including government services, healthcare, and tourism.

REFERENCES

[1] A. Kumar and S. Gupta, "Development of a Conversational Chatbot for College Enquiry System," International Journal of Computer Applications, vol. 182, no. 45, pp. 15–20, 2020.

[2] J. Brownlee, Deep Learning for Natural Language Processing: Develop Deep Learning Models for Natural Language in Python, Machine Learning Mastery, 2019.

[3] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," Artificial Intelligence, vol. 193, pp. 217–250, 2012.

[4] Google Developers, "Custom Search JSON API," [Online]. Available: https://developers.google.com/custom-search/v1/overview. [Accessed: Apr. 18, 2025].

[5] OpenAI, "OpenAI GPT-3 Documentation," [Online]. Available: https://platform.openai.com/docs. [Accessed: Apr. 18, 2025].

[6] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python, O'Reilly Media, 2009.

[7] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," [Online]. Available: https://www.tensorflow.org. [Accessed: Apr. 18, 2025].

[8] Google Cloud, "Translation API Documentation," [Online]. Available: https://cloud.google.com/translate/docs. [Accessed: Apr. 18, 2025].

[9] A. Vaswani et al., "Attention Is All You Need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017.

[10] Y. Zhang, "Recent Advancements in Multilingual NLP Models: A Survey," ACM Computing Surveys, vol. 54, no. 5, pp. 1–38, 2021.

[11] M. I. Ciolacu, B. Haderer, A. Berl, and P. Svasta, "Education 4.0: Innovation Learning Lab for AI-Analysis and Concept Proposal," in *Proc. 2021 IEEE 27th Int. Symp. Design Technol. Electron. Packag. (SIITME)*, 2021, pp. 112–117.

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., Pearson, 2023.

[13] A. Sheth, C. Thomas, and A. Thakkar, "Smart Conversational Agents for Enhanced User Experience in Higher Education," *IEEE Internet Comput.*, vol. 24, no. 1, pp. 46–54, 2020.

[14] K. Bhatia, S. Gupta, and R. Jain, "Intelligent Chatbot for University Admission System Using NLP and Machine Learning," in *Proc. IEEE Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, 2022, pp. 1372–1377.

[15] B. Klein and P. S. Sajja, "Web Scraping in Python for Dynamic Data Extraction and Analysis," *J. Data Intell.*, vol. 5, no. 3, pp. 101–109, 2021.

[16] Microsoft, "Azure Language Understanding (LUIS) Documentation," [Online].Available: https://learn.microsoft.com/en-us/azure/cognitive-services/luis/. [Accessed: Apr. 18, 2025].

[17] A. P. Mishra and R. K. Singh, "Bilingual Intelligent Chatbot for Indian Languages Using Google Translate API," in *Proc. 2021 Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, 2021, pp. 900–905.