International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Chatbot: An automated conversation system for the educational domain

*Dr. Prasanna Kumar M.[1], Amulya S [2], Meghana H R[3], Puneeth K S[4], Rakshitha C R[5]*

[1]Assistant Professor, Department of Information Science and Engineering, Shridevi Institute of Engineering and Technology, Tumakuru, Karnataka Email: prasanna.ise@shrideviengineering.org

[2]Department of Information Science and Engineering, Shridevi Institute of Engineering and Technology, Tumakuru, Karnataka Email: amulyas0910@gmail.com

[3]Department of Information Science and Engineering, Shridevi Institute of Engineering and Technology, Tumakuru, Karnataka Email: meghagowda3101@gmail.com

[4]Department of Information Science and Engineering, Shridevi Institute of Engineering and Technology, Tumakuru, Karnataka Email: punithkurbas12345@gmail.com

[5]Department of Information Science and Engineering, Shridevi Institute of Engineering and Technology, Tumakuru, Karnataka Email: rakshithacr1234@gmail.com

**ABSTRACT:**

Educational institutions receive a large volume of repetitive enquiries related to courses, admissions, placements, events and facilities. Traditional enquiry handling through phone calls, manual helpdesks and static websites is time-consuming, dependent on human availability and often fails to provide instant responses. This paper presents an AI-powered college enquiry chatbot designed specifically for the educational domain. The system integrates Natural Language Processing (NLP), fuzzy string matching and semantic search to understand user queries in natural language, even in the presence of spelling mistakes or variations in phrasing. The backend is implemented using the Flask framework in Python, with a JSON-based knowledge base containing frequently asked questions, placements and event information derived from the college website. A web based chatbot interface allows students and parents to interact with the system in real time. Experimental evaluation and user observations indicate that the proposed chatbot reduces manual workload, improves response time and offers a scalable and user friendly solution for automated college enquiry handling. Index Terms—Chatbot, Natural Language Processing, Fuzzy Matching, Semantic Search, Flask, College Enquiry, Educational Technology.

**Keywords**: Chatbot, Natural Language Processing (NLP), Fuzzy Matching, Semantic Search, Flask, College Enquiry System, Educational Technology, Web Application, Artificial Intelligence, Information Retrieval.

## INTRODUCTION

Colleges and universities receive continuous enquiries re garding programmes offered, admission procedures, fee structure, placements, events, hostel facilities and contact information. Traditionally, such enquiries are handled manually by administrative staff via phone calls, emails or in-person visits. This manual process is labour-intensive, restricted to office hours, and frequently leads to delays and inconsistent information delivery.

With the growth of Artificial Intelligence (AI) and Natural Language Processing (NLP), chatbots have emerged as an effective solution for automating routine question-answering tasks. An AI-powered chatbot can interact with users through a natural language interface and provide instant responses drawn from a structured knowledge base. In the educational domain, such a chatbot can act as a virtual assistant for students and parents seeking information about the institution.

This work focuses on designing and implementing a college enquiry chatbot for Shridevi Institute of Engineering and Tech nology. The proposed system uses a combination of keyword matching, fuzzy string matching and semantic similarity using sentence embeddings to interpret user queries. The backend is built using the Flask framework, and the knowledge base is stored using JSON files containing curated FAQ entries, events and placement information.

### A. Motivation

The primary motivation behind this project is to improve accessibility and responsiveness of college information ser vices. During admission seasons and examination periods, enquiry volume increases significantly, putting pressure on administrative staff. An automated chatbot can help in:

- Providing 24/7 access to essential information.
- Reducing repetitive manual tasks for staff.
- Improving user satisfaction through instant responses.

### B. Problem Statement

Students and parents face difficulty in accessing timely and accurate information about the college. Existing systems rely on manual communication channels and static web pages, which are not interactive and do not support conversational queries. There is a need for an automated, intelligent, and user friendly enquiry system that can understand natural language queries and provide relevant answers reliably.

### C. Objectives

The main objectives of the project are: • To design and develop an AI-based chatbot for college enquiry handling. • To integrate NLP, fuzzy matching and semantic search for robust query understanding. • To provide accurate and personalized responses using a structured knowl edge base. • To build a scalable and extensible system that can be updated with new FAQs, events and placement data.

## LITERATURE SURVEY

Chatbots have been widely adopted in domains such as e commerce, customer support and education. Rule-based chat bots rely on predefined patterns and simple keyword matching, which limit their ability to handle variations in user queries. Recent research has shifted towards data-driven and AI-based chatbots that leverage NLP and machine learning techniques.

Work in [1] highlights the usefulness of chatbots in supporting student queries in higher education. The authors emphasise that responsiveness and accuracy are critical for user satisfaction.

In [2], the role of NLP in understanding user intent and processing unstructured text is discussed, including techniques such as tokenization, stemming and semantic analysis. Fuzzy string matching methods such as Levenshtein distance and RapidFuzz [3] have been used to tackle typographical errors in user inputs.

Semantic search using sentence embeddings has gained popularity with the introduction of models like Sentence BERT [4]. These models map sentences to dense vectors such that semantically similar sentences are close in vector space. Combining fuzzy matching with semantic search enables chatbots to handle both spelling variations and paraphrased questions, making them suitable for real-world educational enquiry systems.

## PROPOSED SYSTEM

The proposed system is an AI-powered chatbot integrated with a web-based front end. The overall architecture is shown in Fig. ??. The system is composed of the following mod ules: user interface, Flask-based backend, NLP and matching engine, and JSON-based knowledge base.

### A. System Architecture

The system architecture consists of:

- **Frontend Chat Interface:** A web-based interface built with HTML, CSS and JavaScript which allows users to type queries and view responses in a conversational format.
- **Flask Backend API:** A Python Flask application exposes REST APIs (e.g., /api/query) to receive user queries, process them and return answers.
- **NLP and Matching Engine:** This module performs text preprocessing, fuzzy matching and semantic similarity computation.
- **Knowledge Base:** A structured set of JSON files such as faqs.json, events.json and placements.json, which store curated information about the
college.

### B. Workflow.

The workflow of the chatbot is as follows:

1. The user enters a query in the chatbot interface.
2. The frontend sends the query to the Flask backend via an HTTP POST request in JSON format.
3. The backend preprocesses the query, applies fuzzy spelling correction and constructs a normalized text form.
4. The system first attempts keyword-based and fuzzy matching against FAQ entries; if no strong match is found, semantic similarity is computed using sentence embeddings.
5. The best matching answer is retrieved from the knowledge base and returned as a JSON response.
6. The frontend displays the chatbot response in the chat window.

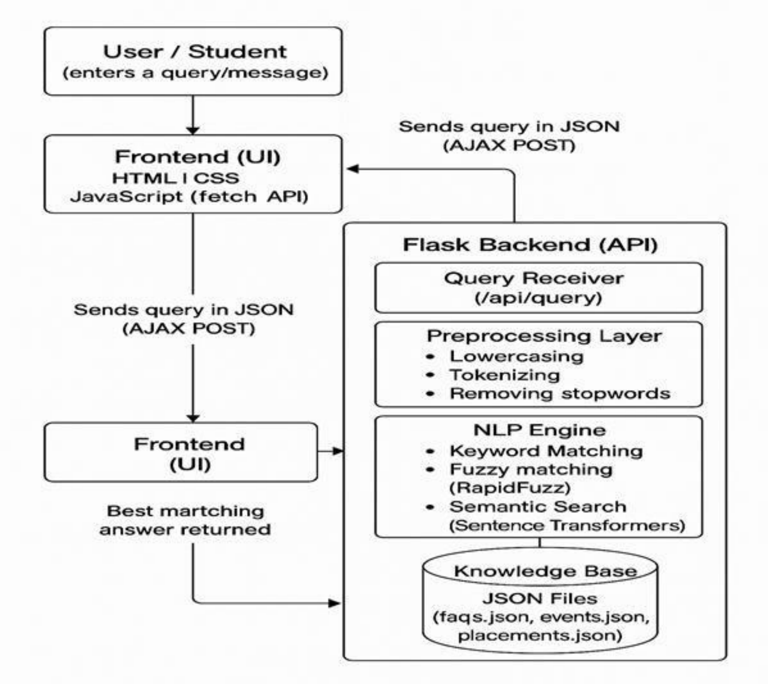

**Fig. 1. High-level system architecture of the college enquiry chatbot.**

## IMPLEMENTATION DETAILS

### A. Technology Stack

The implementation uses the following technologies:

- **Backend:** Python 3 and Flask framework.
- **NLP**: Sentence-Transformers for semantic search.
- **Fuzzy Matching**: RapidFuzz library for approximate string matching.
- **Data Storage**: JSON files for FAQs, placement and event data.
- **Frontend: HTML**, CSS and vanilla JavaScript for the chat interface.

### B. Knowledge Base Design

The FAQ knowledge base is stored in a JSON file with entries containing the following fields:

- question: canonical question text.
- answer: corresponding answer.
- keywords: list of keywords associated with the question.
- alt_questions: alternative phrasings of the same question. An example entry is shown below:

```
{
 "question": "What courses are available?",
"keywords": ["course", "program", "degree", "branch", "courses"],
"alt_questions": [ "Which programs do you offer?", "List the branches
available" ],
"answer": "The college offers B.E. programs in CSE, ISE, AI & ML, ECE, EEE, Civil and Mechanical Engineering."
}
```

### C. NLP and Matching Logic

The matching engine combines three approaches:

**1) Keyword and Fuzzy Matching:** The system maintains a map of keywords to answers. For each incoming query, simple keyword matching is performed. If the user input contains spelling mistakes, RapidFuzz is used to perform fuzzy matching between query tokens and known keywords.

**2) Semantic Search:** For robust handling of paraphrased questions, sentence embeddings are generated using a Sentence-Transformer model. Embeddings for all FAQ questions and alternative questions are precomputed and stored. During runtime, the query embedding is compared with stored embeddings using cosine similarity, and the answer corresponding to the most similar FAQ is selected

**3) Fallback Handling:** If both fuzzy and semantic scores are below predefined thresholds, the system returns a graceful fallback message suggesting the user to rephrase or ask about major categories such as courses, admissions, placements or contact

### D. Frontend Chat Interface

The frontend is implemented as a single-page chat in terface. User messages are appended as chat bubbles, and responses from the backend are shown in a conversational style. Quick-access buttons are provided for common cate gories like “Courses”, “Admissions”, “Placements”, “Events” and “Contact” to improve usability.

## RESULTS AND DISCUSSION

The chatbot was tested with a variety of queries related to the actual college website content, including questions on courses, admission procedures, contact details, hostel facilities and placements. The system was able to:
Correctly interpret common spelling mistakes such as “couses” for “courses” and “admisson” for “admission”.

- Match semantically equivalent queries that used different wording.
- Provide instant responses within a fraction of a second on a standard desktop system.

Informal user feedback from students indicated that the chatbot interface was intuitive and responses were helpful for general enquiry purposes. The system significantly reduces the need for manually answering repetitive questions and can be further extended to support additional domains such as exam timetables and personalized student services.

## CONCLUSION AND FUTURE WORK

This paper presented an AI-powered college enquiry chatbot that leverages NLP, fuzzy string matching and semantic search to provide automated responses to student and parent queries. The system is implemented using the Flask framework with a JSON-based knowledge base and a web-based frontend interface. The integration of fuzzy matching and sentence embeddings improves robustness to spelling errors and para phrasing.
Future work includes integrating the chatbot with messaging platforms such as WhatsApp and Telegram, adding multilin gual support (for example, Kannada and Hindi), incorporating live data sources for notices and timetables, and extending the system with authentication-based personalized services such as student-specific results and attendance.

### REFERENCES

1. *Author, B. Author, “Educational Chatbots for Student Support: A Survey,” International Journal of Educational Technology, vol. X, no. Y, pp. 1–10, Year.*
2. *Manning and H. Schutze, Foundations of Statistical Natural Language Processing. MIT Press, 1999.*
3. *M. Weigel, “RapidFuzz: A Rapid Fuzzy String Matching Library,” Online, Available: https://github.com/maxbachmann/RapidFuzz.*
4. *N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proceedings of the 2019 Conference on Empirical*
5. *Methods in Natural Language Processing, 2019.*
6. *Flask Documentation, “Flask Web Development Framework,” Online, Available: https://flask.palletsprojects.com.*
7. *Sentence-Transformers Documentation, “Semantic Textual Similarity and Applications,” Online, Available: https://www.sbert.net*