International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Based College Enquiry Chatbot Using Natural Language Processing

A Natural Language Processing Approach to Automating Student and Parent Enquiries

Mr. Irfan Amjad Shaikh

Ms. Chikhale Shivani Subodh

Assistant Professor, Department of Master of Computer Application,
JCEI's Jaihind Institute of Management and Research, Vadgaon Sahani
Email: shaikhirfanamjad56@gmail.com,

chikhaleshivani25@gmail.com

Abstract – Every year, college offices answer the same questions from students and parents about admissions, fees, courses, eligibility, scholarships, hostels, examinations and contact details. Handling these by hand takes up staff time, is possible only in office hours, and does not always give the same answer twice. In this work we build a college enquiry chatbot that reads a question written in ordinary language and returns the best matching answer from a college knowledge base. The system is written in Python with scikit-learn and a small rule-based text normalizer, and users reach it through a simple Flask chat page. Each question is cleaned, tokenized, lemmatized and turned into a TF-IDF vector. A Logistic Regression classifier predicts the topic of the question, and cosine similarity then picks the closest stored question within that topic. When the classifier is not confident enough, the bot gives a fallback reply instead of guessing. We tested the system on a dataset prepared for this study, with 270 questions in 9 intents and 40 out-of-scope questions, over five random splits. It reached an intent accuracy of 84.4% and a weighted F1-score of 84.3%, against 59.6% for a retrieval-only baseline. It declined 83.2% of the out-of-scope questions and needed about 1.4 ms per query. The whole system runs on an ordinary computer with no GPU or paid service, and it can later be extended with deep learning models, more languages, or voice and mobile interfaces.

Key Words: Chatbot, Natural Language Processing, College Enquiry System, Intent Classification, TF-IDF, Cosine Similarity, Python, scikit-learn.

1. INTRODUCTION

Whenever admissions open, the college office is flooded with the same set of questions. Who is eligible, what are the fees, when is the last date, which courses are offered, is there a scholarship, is hostel available, whom do I call? The staff can answer only during working hours, and two people may give two slightly different answers. Students and parents end up waiting for simple information, while the staff spend their day on routine replies instead of more important work.

A chatbot can answer such questions at any hour and give the same answer every time. The earliest chatbots, such as ELIZA, worked by matching keywords to fixed replies [1]. They are easy to build, but they break when a user words the question differently from the stored pattern. Natural Language Processing (NLP) helps here because the system works with the words in the question rather than an exact string. "What is the fee for MCA?", "MCA course charges" and "how much do I pay for MCA" can then lead to one and the same answer [2], [3].

We kept the technology deliberately plain. The system uses standard text pre-processing, TF-IDF vectors, an intent classifier and cosine similarity [4], [6]. Everything runs on a normal computer, so a college can deploy it without paid cloud services or special hardware. The knowledge base is an ordinary file that office staff can edit without writing any code.

1.1 Problem Statement

Manual handling of college enquiries is repetitive, depends on office timings and gives inconsistent answers, while keyword-based chatbots fail when questions are worded differently. What is needed is a simple NLP-based chatbot that understands natural questions, answers from official college information, and says so when it is not sure instead of giving a wrong answer.

1.2 Objectives

- To develop an NLP-based chatbot that answers college enquiries in natural language.
- To build a college knowledge base covering admissions, fees, courses, eligibility, scholarships, hostel, examinations, placement and contacts.
- To classify the intent of a query using TF-IDF features and a machine learning classifier.
- To retrieve the best answer by cosine similarity within the predicted intent.
- To deal with unknown or out-of-scope questions through a confidence threshold and a fallback reply.
- To measure the system with accuracy, precision, recall, F1-score and response time.
- To offer a low-cost solution that college staff can maintain without programming.

1.3 Contributions

- A two-stage design in which intent classification is followed by similarity-based answer retrieval.
- A confidence-threshold fallback that reduces wrong answers to unknown questions.
- A knowledge base in a simple editable format, so content can change without touching the code.
- An evaluation procedure that can be repeated, with standard metrics over five random splits.
- A framework that can later take deep learning, multilingual, voice and mobile support.

2. LITERATURE REVIEW

Chatbot research started with pattern-matching programs such as ELIZA, which imitated conversation through keyword rules [1]. Later surveys trace the move from rule-based designs to retrieval-based and generative models, and describe their use in education, customer service, health and business [2]. Shawar and Atwell looked at practical uses, including answering frequently asked questions, and found that a chatbot can act as an easy entry point to stored information [3].

FAQ-style chatbots rest on classical information retrieval. TF-IDF weighting and cosine similarity measure how close a query is to a stored text. They remain popular because they are fast and need little data [4]. Toolkits such as NLTK and scikit-learn put tokenization, lemmatization, feature extraction and classification within reach of students and small institutions [5], [6], and standard textbooks describe the language processing steps used in such pipelines [7].

Deep learning has improved language understanding a great deal. Pre-trained transformer models such as BERT perform well on many tasks [8], and frameworks such as Rasa offer complete dialogue management for building assistants [9]. The catch is that they usually ask for more training data, more computing power and more effort to deploy and maintain, which is a heavy load for a small college office.

This gap is what motivates our work: a chatbot that combines intent classification with similarity-based retrieval, runs on ordinary hardware, uses a knowledge base the staff can edit, and declines when it is unsure. Table 1 compares the approaches.

Table 1: Comparison of existing approaches and the proposed system

Approach	Strengths	Limitations
Manual office handling	Accurate when staff are available	Limited hours, slow, inconsistent replies
Rule-based chatbot [1]	Simple to build	Fails on differently worded questions
Retrieval-based (TF-IDF) [4]	Fast, needs little data	No intent understanding on its own
Transformer-based models [8]	Strong language understanding	Needs more data, computing power and maintenance
Dialogue frameworks [9]	Full conversation management	Heavier setup for a simple FAQ task
Proposed system	Intent classification plus retrieval, confidence fallback, low cost	Limited to topics in the knowledge base

3. METHODOLOGY

3.1 System Architecture

The system has six layers: user interface, pre-processing, feature extraction, intent classification, answer retrieval and knowledge base. The user interface is a web chat page where the student types a question and reads the reply. The pre-processing layer cleans and normalizes the text, and the feature extraction layer turns it into a TF-IDF vector. The classification layer predicts the topic (intent) of the question along with a confidence value. The retrieval layer then finds the most similar stored question inside that intent and returns its answer. The knowledge base is a CSV file with intents, example questions and answers, which staff can edit. If the confidence is below a threshold, the system does not guess; it asks the user to contact the college office, as shown in Fig. 1.

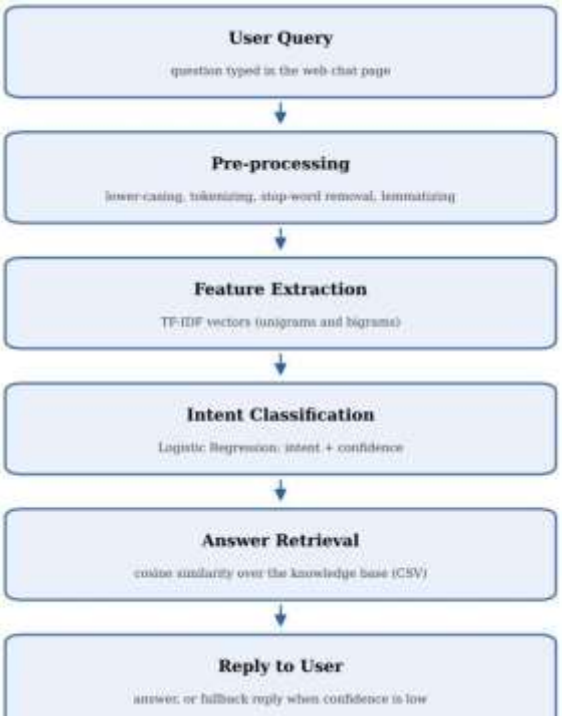


Fig. 1: Architecture of the proposed college enquiry chatbot

3.2 NLP Pipeline

Each query goes through five steps. First, the text is converted to lower case. Next it is split into tokens made of letters and digits, which also removes punctuation. Stop words are then removed using the English list from scikit-learn [6], but words such as “last”, “first” and “not” are kept because they change the meaning of an enquiry. A light rule-based lemmatizer strips plural and “-ing” endings so that “fees” and “fee” are treated alike. Finally, a small list of about 30 synonyms maps words like “tuition”, “charges” and “cost” to a single form.

The cleaned text is converted into TF-IDF vectors of unigrams and bigrams with scikit-learn [6], so that a phrase like “last date” stays together. Using the smoothed inverse document frequency of scikit-learn, the weight of a term t in a question d is given below, where N is the number of stored questions and df(t) is the number of questions that contain t [4]. Each vector is then normalized to unit length.

$$tfidf(t, d) = tf(t, d) \times (ln((1 + N) / (1 + df(t))) + 1)$$

3.3 Intent Classification and Answer Retrieval

A Logistic Regression classifier [6] with regularization parameter C = 10 is trained on the TF-IDF vectors of the example questions. It predicts one of nine intents: admission, fees, eligibility, courses, scholarship, hostel, examination, placement and contact. The probability of the predicted intent is used as the confidence score. Inside the predicted intent, the answer is chosen by the cosine similarity between the query vector q and each stored question vector d:

$$cos(q, d) = (q \cdot d) / (||q|| \times ||d||)$$

The answer attached to the highest-scoring question is returned. The fallback reply is used when the intent confidence

is below a threshold T or when the similarity is below a threshold T . c s

Both thresholds were chosen on the validation split by trying values from 0 to 0.60 in steps of 0.05 and keeping the pair that gave the highest mean of two numbers: the correct-answer rate on in-scope validation questions and the fallback rate on out-of-scope validation questions. The selected T ranged from c

0.25 to 0.45 across the five runs, while T came out as 0 every s time. In practice, then, the classifier confidence alone decided when to fall back.

3.4 Dataset

The dataset was written for this study and was not collected from real students. It holds 270 questions in 9 intents (admission, fees, eligibility, courses, scholarship, hostel, examination, placement and contact), with 30 questions per intent. Each intent has three sub-topics of ten questions, and every sub-topic has one answer. The fees intent, for example, covers the fee amount, the mode of payment, and refund or concession. In each sub-topic one question is in Marathi written with English letters (for example "Fees kiti ahe?"). The answers are illustrative template replies that point to the college website or office, and they contain no real college figures. A separate set of 40 out-of-scope questions, on topics such as weather, sports and general knowledge, is used to test the fallback. The data is stored as a CSV file with the columns intent, question and answer.

3.5 Evaluation Parameters

The in-scope questions are split by stratified sampling into training (70%, 189 questions), validation (10%, 27) and test (20%, 54) sets. The out-of-scope questions are split into 15 for validation and 25 for testing. The whole procedure is repeated with five different random seeds, and we report the mean and standard deviation (SD). Intent accuracy, precision, recall and F1-score are computed on the predicted intent without any threshold, as follows:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

$$F1 = 2 \times Precision \times Recall / (Precision + Recall)$$

Precision, recall and F1-score are weighted averages over the intents. Once the thresholds are applied, three more measures are reported on the in-scope test questions: the correct-answer rate (an answer from the correct intent), the wrong-answer rate (an answer from a wrong intent) and the false fallback rate (a refusal although the question was in scope). On the out-of-scope test questions we report the fallback rate, where higher is better. An answer is counted as correct if it comes from the right intent. The processing time per query, from raw text to selected answer, is measured in milliseconds. As a baseline we use a retrieval-only method that returns the intent of the most similar stored question over all intents, without the classifier.

3.6 Experimental Setup

The system is implemented in Python 3.13.16 [10] with scikit-learn 1.9.1 [6], pandas 3.0.5 and NumPy 2.5.3. The chat interface is a web page served by a small Flask application [11]. All experiments were run on a Linux machine with a two-core Intel Xeon processor at 2.10 GHz and 7 GB of RAM, without a GPU.

4. RESULTS

Over the five runs, the classifier reached a mean intent accuracy of 84.4% and a weighted F1-score of 84.3% on the test questions. This is 24.8 percentage points above the retrieval-only baseline (59.6%). With the confidence threshold applied, 78.1% of in-scope questions received a correct answer, 9.3% received a wrong answer and 12.6% were refused. The chatbot declined 83.2% of the out-of-scope questions. Table 2 gives the full results.

Table 2: Results on the college enquiry dataset (mean ± SD over five runs)

Analysis Parameter	Result Obtained
Total in-scope questions	270 (9 intents × 30)
Out-of-scope questions	40
Train / validation / test per run	189 / 27 / 54
Intent classification accuracy	84.4 ± 2.8 %
Weighted precision	87.2 ± 2.4 %
Weighted recall	84.4 ± 2.8 %
Weighted F1-score	84.3 ± 2.8 %
Retrieval-only baseline accuracy	59.6 ± 5.6 %
Correct-answer rate (with fallback)	78.1 ± 6.1 %
Wrong-answer rate (with fallback)	9.3 ± 2.3 %
False fallback rate (in scope)	12.6 ± 6.1 %
Fallback rate on out-of-scope questions	83.2 ± 5.2 %
Average processing time per query	1.39 ± 0.21 ms

The intent-wise F1-scores are shown in Fig. 2. Scholarship was classified best (F1 = 93.3%), while eligibility (73.9%) and courses (79.2%) were the weakest. Pooled over the five runs, which together contain 270 test predictions, the most frequent confusions were hostel questions predicted as fees (6 cases) and fees questions predicted as courses (4 cases).

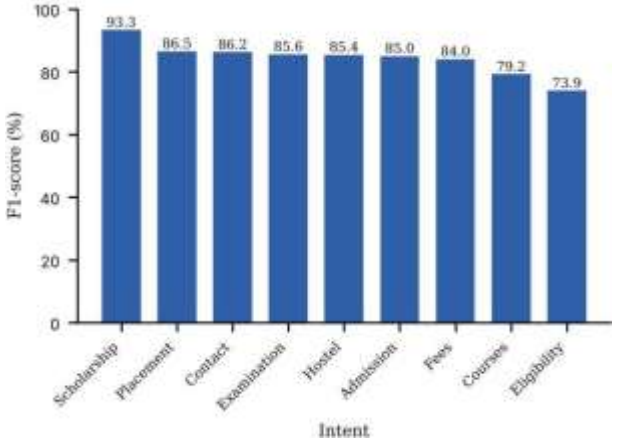


Fig. 2: Intent-wise F1-score (%), mean of five runs

5. DISCUSSION

The results show that a simple NLP pipeline can handle repetitive college enquiries without heavy infrastructure. Adding the intent classifier to similarity retrieval raised accuracy by 24.8 percentage points over retrieval alone. A likely reason is that the classifier learns which words point to each

topic, instead of depending on a single nearest question. The confidence threshold cut the share of wrong answers from 15.6% (when every question is answered) to 9.3%, but the price was that 12.6% of valid questions were refused. For information such as fees and last dates, being sent to the office is better than getting a wrong answer, so we consider the trade-off acceptable. Colleges can also tune the threshold to suit themselves.

The results need to be read with care. The dataset is small and was written for this study, with balanced intents and similar wording inside each sub-topic, so accuracy on real student queries is probably lower. Answers are judged at the intent level, and the answers in the dataset are templates, not official college data. The fallback is also imperfect: about 17% of out-of-scope questions still received an answer. In a manual test of the web application, for instance, the short request "tell me a joke" was matched to an admission reply. Eligibility and courses were the weakest intents, most likely because their wording overlaps with neighbouring intents such as admission. Only about 10% of the questions use Marathi in English letters, so mixed-language support has not been established.

The chatbot treats every question separately, and the reported processing time leaves out web and network delay. Testing with questions collected from real students and parents would give a more reliable measure. Compared with transformer-based models [8] and dialogue frameworks [9], our system is less powerful, but it is much easier to deploy, explain and maintain, which suits a small institution.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

This paper presented a college enquiry chatbot that answers common student and parent questions in natural language using text pre-processing, TF-IDF features, intent classification and cosine-similarity retrieval. On our study dataset it reached 84.4% intent accuracy and declined 83.2% of out-of-scope questions, and the confidence threshold let it refuse instead of answering wrongly. The system runs on ordinary hardware and uses a knowledge base that staff can edit, so it can reduce the repetitive workload of the college office and give users quick, consistent replies.

6.2 Future Work

Several improvements are possible: transformer-based sentence embeddings, support for Marathi and Hindi, memory for multi-turn conversations, voice input, and a mobile or WhatsApp interface. Evaluation with real student queries, integration with the college admission portal, and an admin dashboard for updating answers would make the system more accurate and more useful.

REFERENCES

[1] J. Weizenbaum, "ELIZA — A Computer Program for the Study ofNatural Language Communication Between Man and Machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[2] E. Adamopoulou and L. Moussiades, "Chatbots: History,Technology, and Applications," *Machine Learning with Applications*, vol. 2, Art. no. 100006, 2020.

[3] B. A. Shawar and E. Atwell, "Chatbots: Are They Really Useful?,"*LDV Forum*, vol. 22, no. 1, pp. 29–49, 2007.

[4] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge University Press, 2008.

[5] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA, USA: O'Reilly Media, 2009.

[6] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python,"*Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. (draft). Stanford University, 2023. [Online]. https://web.stanford.edu/~jurafsky/slp3/

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[9] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management," arXiv:1712.05181, 2017.

[10] Python Software Foundation, "Python Documentation." [Online]. https://docs.python.org/3/

[11] Pallets, "Flask Documentation." [Online]. https://flask.palletsprojects.com/