

Troy Assist: A Smart Gen-AI Student

Anil Khatiwada
Masters of computer Science
Troy University
akhatiwada227959@troy.edu

Shankar Bhattacharai
Masters of computer Science
Troy University
sbhattacharai225070@troy.edu

Bishal Awasthi
Masters of computer Science
Troy University
bawasthi@troy.edu

Abstract—*Troy Assist is designed to fill in the communication gaps between students and university administrative departments. Utilizing AI-based query resolving, a knowledge base, and response systems which operate in real time, the solution reduces waiting periods, support personnel load, and enhances support tailoring. The system, developed in React, Django and Gemini, demonstrates good results in accuracy, context retention, and user satisfaction.*

Keywords—AI, student support, chatbot, Django, Gemini, Troy University

Introduction

With education becoming more digital, students expect instant responses to their questions. Help desks which manage issues via emails and static Frequently Asked Question (FAQ) texts lack responsiveness, scalability, and personalization. This is exactly the problem Troy Assist is solving with its smart, generative AI student support system: intelligently personalized virtual assistants that can answer in context and care around the clock.

A. Problem Statement

Concerning admissions, housing, deadlines, and other academic-related matters, students express their concern, but the answer is not forthcoming. Contributing factors include support traffic jam of incoming calls, an assortment of disorganized information repositories, and the absence of streamlined user-centric educational interfaces. Students are not being served by these arrangements; rather, the administrative problems within the education institution are being compounded.

B. Objectives

- Develop a responsive generative AI chatbot that can real time Answer questions posed by students.
- Develop a knowledge base from scraped data from official university websites using a structured template.
- Build the frontend with React and implement responsive design on the user interface.

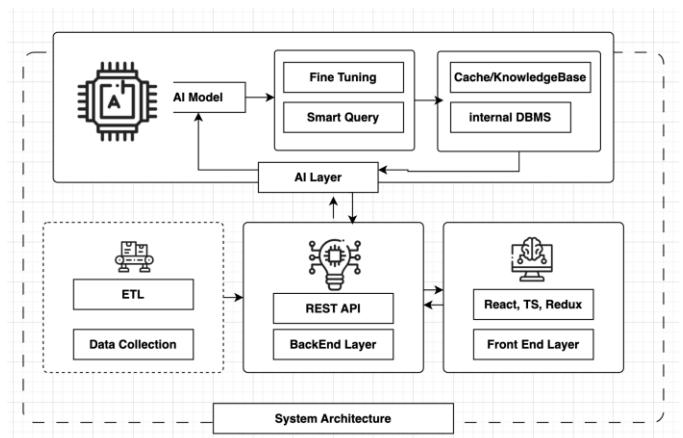
- Use Django coupled with Gemini AI for the workstation backend to formulate responses smartly in real-time upon receiving queries from users.

C. Literature Review

Support academic work AI chatbots like Jill Watson at Georgia Tech and Sunny at Arizona State University have systematically aided students in multiple institutions. These systems, however, are mostly closed-ended, restricting user-initiated interaction to predetermined responses. With Generative AI and language models such as Gemini and GPT, students are able to receive contextually more flexible and responsive assistance. The use of technology in the administration of higher learning institutions is still in its infancy but has immense potential.

D. Methodology

The project was developed using an agile model which supported periodic refinement and user-based design changes. Major steps of the project included requirement analysis, data collection, system design, AI architecture, and finally integration and deployment. There was constant active testing and validation cycle from the beginning through to the end of the work.



E. System Design and Architecture

The architecture is composed of three elements:

- Frontend: The interface for the end user utilizes React JS with Redux and TypeScript.
- Backend: A Django REST API handles service requests, session management and AI model calls.
- AI Layer: Neural Gemini handles intent recognition and response generation.

Data and file system management are executed within SQLite, web-scraped information fills the knowledge base.

F. System Design and Architecture

The architecture comprises the following three components:

- Frontend: The user interface is designed using React JS and managed by Redux, employing TypeScript.
- Backend: The request, session management, and interaction with the AI model is handled by a Django-based REST API.
- AI Layer: Intent detection and response generation is handled by Gemini.

Data is managed and stored using SQLite, and the knowledge base is populated from data scraped from the web.

G. Implementation Details

The application currently supports ancient search queries and other restrictions. The query is sent to the backend which performs all of the computations. The backend performs keyword and entity extraction, intent identification, and searches the knowledge base for information. If the data isn't readily available, a fallback semantic search is performed to ensure meaningful responses are generated. The chat user interface displays the answer alongside the follow-on suggestions.

H. Data Collection and ETL Pipeline

For the Assistant's foundational knowledge, the core ETL pipeline is structured as follows:

- Extraction: Troy university publicly available webpages were scraped and processed using the BeautifulSoup library.
- Transformation: Scrubbed data was cleaned of irrelevant text and non-anchored text data while using regular expressions.
- Loading: Information is stored in structured CSV files which are later saved in an SQLite database for quicker retrieval.

I. AI and Query Handling

The Gemini model takes care of:

- Intent Detection
- Keyword and Entity Identification
- Context Preservation
- Response Generation in Natural Language

Responses are consistent with the communication style of Troy University and can be completed in any language.

J. Results and Evaluation

The system was tested with more than 100 mocked queries from various domains, including admissions, housing, and courses. The automated accuracy checking returned over 92% qualitative accuracy in relation to the relevance and clarity given. Human evaluators confirmed the system's turnaround time is under 2 seconds.

K. Challenges and Limitations

- Completeness of the data for limited internet sources.
- Stable answers/single user session consistency.
- Undefined/general user query resolution.
- Knowledge base expansion requires cutting down the human interaction needed.

L. Future Improvements

- Installation of interfaces which respond to voice commands.
- Personal dashboards that allow data access for students.
- Updates from university databases in real-time through APIs.
- Future plans include extending to other universities.

M. Conclusion

The use of Generative AI in the administrative systems of higher education is demonstrated with Troy Assist Applications. It incorporates automation of common activities, construction data, and adaptive interfaces which facilitates and simplifies interaction and servicing to the system.

References

- [1] T. Mitchell, Machine Learning, 1st ed. New York: McGraw-Hill, 1997, pp. 45–56.
- [2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
- [3] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., draft version, Stanford Univ., 2023. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
- [4] OpenAI, “GPT-4 Technical Report,” 2023. [Online]. Available: <https://openai.com/research/gpt-4>
- [5] R. A. Calvo, S. D’Mello, J. Gratch, and A. Kappas, “Computing emotion-aware user interfaces,” in The Oxford Handbook of Affective Computing, R. A. Calvo et al., Eds. Oxford, U.K.: Oxford Univ. Press, 2015, pp. 377–393.
- [6] J. Guo, L. Fan, and X. Cheng, “A survey on knowledge base question answering,” ACM Trans. Inf. Syst., vol. 39, no. 2, pp. 1–40, Mar. 2021.
- [7] S. Das, “Beautiful Soup Documentation,” [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [8] Django Software Foundation, “Django 5.0 Documentation,” [Online]. Available: <https://docs.djangoproject.com/en/5.0/> G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)