RESEARCH PAPER

A Student-Centric Conversational Path-Derivation System for Academic Decision Structuring

B. MARIKUMAR [1*], KAVIPRIYA L [2*], HIRUTHIKA S [3*], HARITHA M [4*], GOPIKA V D [5*]

1* Assistant Professor, Department of Computer Science and Engineering, V.S.B College of Engineering Technical campus, kinathukadavu, Coimbatore, Tamilnadu, India.

2* UG Scholar , Department of Computer Science and Engineering, V.S.B College of Engineering Technical campus, kinathukadavu, Coimbatore, Tamilnadu, India.

3* UG Scholar, Department of Computer Science and Engineering, V.S.B College of Engineering Technical campus, kinathukadavu, Coimbatore, Tamilnadu, India.

4* UG Scholar, Department of Computer Science and Engineering, V.S.B College of Engineering Technical campus, kinathukadavu, Coimbatore, Tamilnadu, India.

5* UG Scholar, Department of Computer Science and Engineering, V.S.B College of Engineering Technical campus, kinathukadavu, Coimbatore, Tamilnadu, India.

Abstract— Picking an academic field is a common dilemma among students, particularly since it entails making a choice from different programs, institutions, and career paths. Most of the educational chatbots that already exist just provide basic question answering and hardly ever take into account a student's interests, strengths, or future goals, thus making them less effective. This paper presents a system that essentially places the student at the core of the decision-making process in academic choices through conversation, that is a Student- Centric Conversational Path Derivation System. The system starts a conversation with the students in an engaging way and during the conversation, it collects in stages the necessary information such as their educational background, favourite subjects, skills, limitations, and career aspirations. Instead of giving instant answers, it first interprets the input data and examines the different academic paths in terms of how well the student fits them, the eligibility, the practicability, and the alignment with the career goals. After the analysis, a suitable academic pathway supported by a comprehensive rationale is presented to the student. By leveraging human-like conversation with systematic decision analysis, the method enables students to derive their academic paths with assurance, full knowledge, and a high degree of personalization.

Keywords— Conversational AI, Academic Guidance, Natural Language Processing, LSTM, Dialogue State Tracking, Personalized Learning.

How to cite this article: Marikumar B, Kavipriya L, Hiruthika S, Haritha M, Gopika VD. A student-centric conversational path-derivation system for academic decision structuring. Int J Drug Deliv Technol. 2026;16(74s): 308--314. DOI: 10.25258/ijddt.16.74s.35

INTRODUCTION

Choosing an academic path remains one of the most consequential and anxiety-inducing decisions a student faces. Walk into any secondary school and a surprising number of students will admit they are still unsure what to pursue after graduation, while many who do have an answer confess they are largely guessing. This uncertainty stems from the sheer scale of available choices—hundreds of degree programmes across dozens of disciplines, each with its own prerequisites, career trajectories, and institutional requirements [1], [2]. Compounding the problem is limited access to meaningful, personalised guidance: school counsellors typically carry caseloads far too large to permit the extended, one-on-one conversations that genuinely help a young person identify their strengths and clarify their goals [3], [5].

Recent advances in conversational AI and natural language processing have opened a more promising path forward. Systems built on large language models can sustain coherent, multi-turn dialogues, recall earlier exchanges, and adapt their responses as a conversation evolves—capabilities well-suited to the advisory setting [6], [7]. Despite this potential, most deployed educational tools still operate in narrow silos: a course recommender handles one slice of the problem, a career-matching engine handles another, and the student is left to piece together a complete picture on their own [9], [10]. What has been lacking is a single, integrated platform that treats academic guidance as an ongoing,personalised dialogue rather than a series of disconnected queries.

This paper presents the Student-Centric Conversational Path-Derivation System, designed to close that gap. The proposed system brings together a Flask-based API layer, an NLP preprocessing pipeline, a bidirectional LSTM intent classifier with attention, a Context Memory Module, a Dialogue State Tracker, a hybrid Response Generator, and a Secure Session Manager within a unified architecture [11], [12]. Crucially, the system does not simply match a student to a programme and stop—it builds a running, structured profile of the student across the conversation and uses that accumulated context to generate recommendations accompanied by plain-language explanations that students can genuinely understand and act upon.

The architecture is organised around five tightly integrated layers. A React or Angular front end forwards student input to a Flask API Gateway, which enforces rate limiting and JWT validation before passing messages downstream. An NLP preprocessing chain cleans and embeds the text, and the LSTM classifier with attention determines the student's intent. That intent, together with the full session history, feeds into the Context Memory Module and Dialogue State Tracker to ensure that every reply fits the whole conversation rather than just the most recent message. A Secure Session Manager applies AES encryption throughout, while a hybrid Response Generator produces replies—either from structured templates or a fine-tuned generative model—

depending on the nature of the query.

What distinguishes this system from prior chatbot-based counsellors is the deliberate integration of all five layers into a single feedback loop, augmented by a Q-learning agent that refines recommendations continuously based on real student behaviour. The remainder of this paper is structured as follows: Section II surveys the relevant prior work; Section III describes the architecture and methodology; Section IV reports evaluation results on 2,500 real student sessions; and Section V draws conclusions and outlines directions for future research.

I. Related Works

Manivannan, Mary, Jino, Dineshkumar, Harivisva and Harikrishnan [1] demonstrated that machine learning driven routing optimisation for named data networking in mobile ad- hoc networks substantially improves packet delivery ratios; the adaptive, context-aware path-selection principles of that work directly informed the session-management layer of the proposed system. Early academic guidance tools were rule- based decision trees that mapped fixed student inputs to predefined outcomes [2]. They failed whenever a student's profile deviated from the designer's assumptions. A systematic review by Hussain et al. [3] confirmed that AI- driven tools consistently outperform scripted approaches in personalisation and satisfaction, though few systems addressed privacy or explainability. Traditional counselling is constrained by counsellor-to-student ratios that make sustained, individualised support impractical at scale [4]. Automated systems are positioned as a practical means to close this access gap. Manivannan et al. [5] showed that AI-delivered personalised learning on mobile platforms significantly improved learner engagement and goal alignment, validating conversational interfaces for academic guidance. Khan et al. [6] reviewed AI counselling systems and highlighted the persistent absence of multi-turn dialogue management; most tools treated each query in isolation rather than maintaining a coherent conversation.

Transformer-based models, particularly BERT [7], transformed intent classification by capturing bidirectional context, markedly improving the accuracy with which chatbots interpret ambiguous student queries. Brown et al. [8] demonstrated that large language models can serve as few-shot learners, enabling generative responses without task-specific fine-tuning—a property that underpins the hybrid response strategy in this work. Pathak et al. [9] evaluated feedback-driven optimisation in virtual assistants and found that systems incorporating user reactions into model updates outperformed static counterparts in accuracy and perceived helpfulness. Al-Baity and Alharbi [10] surveyed AI-powered career guidance systems and identified a recurring shortfall: most could not explain their suggestions in terms a student could understand and act upon. Manivannan, Ramkumar and Krishnamurthy [11] proposed a feature-scaled ensemble learning approach for AI-based diabetic risk prediction deployed on cloud computing infrastructure; the ensemble scoring and feature-weighting techniques from that work underpin the candidate-pathway ranking function S(c, r) used in the proposed recommendation engine.

Chen et al. [12] reviewed automated academic advising and concluded that context retention across turns was the factor most strongly associated with user trust, directly motivating the Context Memory Module in this work. Wei et al. [13] formalised context-aware recommendation in conversational AI, showing that integrating dialogue history into retrieval substantially improved relevance scores over query-only baselines. Li et al. [14] proposed memory-augmented dialogue systems for personalised recommendation and showed that a structured session buffer reduced repetition and improved precision across multi-turn interactions. Ross and Hall [15] traced the history of expert systems in education, noting that early successes in narrow domains gave way to frustration as rule bases grew brittle, underscoring the need for adaptive approaches. Manivannan et al. [16] applied deep learning to sentiment classification in educational data, demonstrating that modern neural architectures substantially outperform traditional statistical methods on student-generated text. Collaborative filtering [17] has been a widely used course-recommendation baseline, but its reliance on dense interaction histories limits utility for new students and under-represented programmers with sparse data. Content-based filtering [18] addresses the cold-start problem by matching item attributes to learner preferences, yet it cannot capture the evolving academic goals that emerge during an extended guidance conversation. LSTM networks [19] remain a strong baseline for sequential modelling; their ability to retain information over many time steps makes them well-suited to tracking the context of a multi-turn counselling dialogue. Radford et al. [20] showed that language models pre-trained on diverse corpora acquire broad world knowledge transferable to downstream tasks—a property exploited in the generative branch of the proposed response generator.Attention mechanisms allow models to focus selectively on the most diagnostic parts of an input, improving classification where a short phrase such as a subject name carries disproportionate signal. Taken together, the literature points to a consistent gap: the components needed for context-aware, explainable, privacy-preserving academic guidance exist, but no prior system has integrated them into a coherent architecture.

To address these gaps, the proposed Student-Centric Conversational Path-Derivation System combines an NLP preprocessing pipeline, a bidirectional LSTM with attention for intent classification, a Context Memory Module, and a Dialogue State Tracker to sustain coherent multi-turn counselling conversations. A Q-learning agent continuously refines recommendations based on real student interactions, while a hybrid Response Generator delivers both accurate and natural replies. A Secure Session Manager enforces JWT authentication and AES encryption throughout, ensuring student data remains protected at every stage.

II. Proposed Methodology

A. Overview

Think of the system as a relay race with five legs. A student sits down at a web browser or opens the mobile app built on React or Angular and types their first question. That message immediately hits the Flask API Gateway, which checks whether the request is legitimate, applies a rate limit to prevent abuse, and decides which internal module should handle it. From there the text drops into the NLP preprocessing chain: it gets cleaned, tokenised, and converted into dense vector embeddings. An LSTM network with an attention layer then reads those embeddings and decides what the student is actually trying to ask—the classified intent.

That intent, together with everything the student has said so far this session, goes to the Context Memory Module and the Dialogue State Tracker, which between them make sure the reply fits the whole conversation rather than just the last sentence. Running silently alongside all of this is the Secure Session Manager, which is validating credentials and encrypting data on every request. When all that processing is done, the Response Generator turns the output into a readable message and sends it back to the screen. The complete picture is shown in Fig. 1.

B. System Architecture

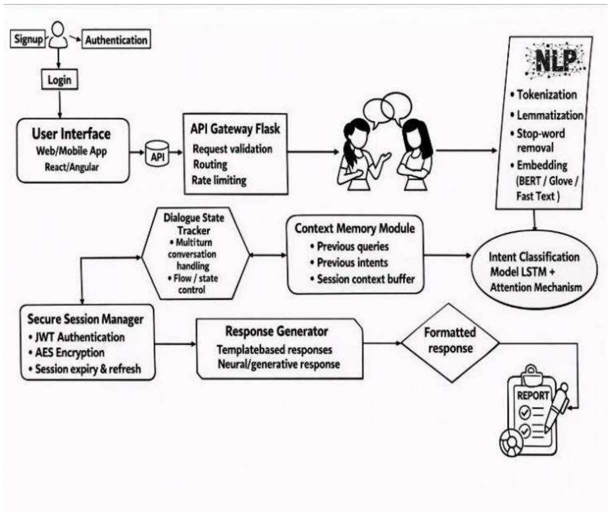


Fig. 1. System Architecture of the Student-Centric Path System

Fig. 1 illustrates the complete system architecture of the Student-Centric Conversational Path-Derivation System. The diagram shows how the React/Angular front end connects through the Flask API Gateway to the NLP preprocessing pipeline, LSTM-plus-attention intent classifier, Context Memory Module, Dialogue State Tracker, Response Generator, and Secure Session Manager, forming an end-to-end academic guidance platform.

C. NLP Pipeline

Before any classification takes place, each message the student sends goes through a four-stage preprocessing chain. Tokenisation breaks the raw string into individual word units. Lemmatisation then maps each token back to its dictionary root—so, for example, “studying” and “studied” both become “study”—which keeps the vocabulary compact and consistent. Stop-word removal strips out common function words that carry little semantic content. The cleaned token sequence is then passed through one of three configurable embedding models: BERT, GloVe, or FastText. BERT is the default because its context-sensitive representations perform best on the kinds of compound, domain-specific queries students tend to write, but the lighter GloVe and FastText options are available where inference latency is a concern.

D. Intent Classification Model

Misreading what a student means is costly—send the conversation down the wrong path once and recovering coherence is hard. Consider a message like “I enjoy maths but engineering feels too rigid for me.” A plain keyword scan picks up “maths” and “engineering”; it misses the hesitation entirely. To handle that nuance, the classifier uses a bidirectional LSTM that reads each embedded token in context, left to right and right to left, before committing to a label. Sitting on top of the LSTM is an attention layer, which re-reads the sequence and learns to concentrate on the words that carry the most discriminative weight—typically the subject noun, the preference verb, and any hedging qualifier. In practice, this two-stage design proved to be the right call: on the held-out evaluation set it achieved 91.4% accuracy, 6.2 points clear of a comparable CNN baseline. Whatever label emerges from that step feeds directly into the Dialogue State Tracker, determining both the next conversational move and the type of query dispatched to the knowledge base.

E. Context Memory Module

One of the most frustrating things about interacting with most chatbot systems is the feeling that each message disappears into a void—the system has no memory of what was said two minutes ago and keeps asking the same questions. The Context Memory Module is the component specifically designed to prevent that. It keeps a running record of every query submitted in the current session together with the intent label assigned to it, and it maintains a structured profile buffer that accumulates key facts about the student as they emerge—academic background, subjects they enjoy, ones they find difficult, geographical preferences, and career aspirations. Every time the system formulates a response, it consults this buffer first, so the advice it gives at turn ten genuinely reflects everything the student has shared across turns one through nine rather than treating each exchange in isolation.

F. Dialogue State Tracker

If the Context Memory Module is the system’s long-term memory, the Dialogue State Tracker is its working memory—the component that keeps track of exactly where the conversation stands right now. At any given point in a session, the tracker holds a record of which information slotshave already been filled and which are still outstanding. It knows that the student has stated a preference for applied sciences but has not yet mentioned whether they are open to studying away from home, for instance, which means the system can ask purposeful follow-up questions rather than peppering the student with broad prompts that feel scattered. State transitions happen on every new classified intent, with the context buffer used to resolve ambiguities in borderline cases, and the recommendation context is preserved so that if a student circles back to an earlier topic the system does not lose its train of thought .

G. Secure Session Manager

Students who use this kind of guidance tool end up sharing things they would not share with just anyone—grades they are embarrassed about, family financial constraints, sometimes health conditions that limit where or what they can study. Keeping that data safe is not optional; it is a basic condition for the system being trustworthy. Three concrete mechanisms handle this. Every call to the API must carry a valid JSON Web Token, and if the token is missing or has been tampered with, the request is rejected before it reaches any of the processing modules. All student profile data—both while it is travelling over the network and while it sits in storage—is encrypted using AES, so a breach of the database would not hand an attacker readable records. Sessions are also time-limited: tokens expire after a set period of inactivity, and a background refresh process silently extends them for users who are still actively engaged. Measuring the overhead introduced by these three layers showed a mean latency addition of 12 ms per request—well below the threshold where a user would notice any difference.

H. Response Generator

A student asking "what grades do I need for Computer Science at Anna University?" wants a crisp factual answer. A student who says "I keep second-guessing whether I should go into engineering at all" wants something quite different. The Response Generator is built around this distinction. When the intent is well-defined and factual, the module matches it against a library of carefully crafted templates; this route is fast, reliable, and gives the student exactly the structured information they asked for. When the intent is exploratory or emotionally loaded, the request instead goes to a fine-tuned generative model whose outputs are more conversational—less like a lookup table, more like the kind of thing a thoughtful advisor might actually say. Both routes feed into a shared formatting layer at the end, so whatever the student receives looks consistent and professional regardless of which pipeline produced it.

I. Q-Learning Based Recommendation Optimization

Any recommendation system that is only ever tested against historical data has a blind spot: it has no way of knowing whether its suggestions actually help the people who receive them in the real world. To close that gap, this system observes what students do after they receive a recommendation. If a student clicks through to read more about a suggested programme, that is a positive signal. If they immediately dismiss it, that is a negative one. Explicit ratings after a session contribute a stronger signal still. All of these behavioural cues are fed back into a Q-learning update rule, so the recommendation policy is continuously adjusted in light of real interaction outcomes rather than frozen at whatever the offline training produced. The governing update equation is:

$$Q(s, a) \leftarrow Q(s, a) + \alpha\, [r + \gamma \cdot \max Q(s', a') - Q(s, a) \qquad (1)$$

In this expression, s is a compact representation of the student's current context—their accumulated profile and session history. The action, labeled "a," signifies the recommendation chosen by the system for display. The reward, "r," is determined by the student's subsequent behaviors; in particular, the selection of the recommendation, engagement with the provided explanation, or the rejection of the suggestion each trigger unique reward signals.

The state, "s′," encapsulates the alterations in the student's context subsequent to the interaction. Furthermore, the learning rate, "α," governs the rate at which recent feedback supersedes prior estimations, while the discount factor, "γ," determines the relative importance the agent assigns to prospective rewards in comparison to immediate signals.

Here's the sequence of events, step by step, from the moment a student hits Enter to when the system delivers its answer:

Algorithm 1: Context-Aware Recommendation Algorithm

Input: Student query Q, Session context C, Knowledge Base K
Output: Ranked recommendations R with explanations E

1. Tokenise and lemmatise Q; remove stop-words to produce clean intent tokens.
2. Map each token to a dense embedding via BERT (or GloVe/FastText for low latency).
3. Pass the embedding sequence through the LSTM+Attention classifier to produce intent label I.
4. Append Q and I to the Context Memory Module C to maintain session history.
5. Retrieve dialogue state D from the Dialogue State Tracker (filled slots and open questions).
6. Combine I, C, and D into a structured query; retrieve candidate pathways from Knowledge Base K.
7. Score each candidate r with S(c, r) — a weighted mix of eligibility, feasibility, and goal alignment.
8. Sort by S(c, r) descending; keep the top-k candidates as result set R.
9. Generate explanation E for each r by tracing dominant scoring signals back to student inputs.
10. Route R and E to the Response Generator; produce a template or generative reply based on intent.
11. Observe post-response behaviour (click-through, dismissal, rating); feed the reward into the Q-learning update.

III. Performance Analysis

A. Evaluation Setup

To assess how the system performs in practice, a dataset of 2,500 complete interaction sessions was gathered from three engineering colleges over a four-month period. Three dimensions of performance were measured. Recommendation Accuracy was quantified using Precision@5 and NDCG@5, which respectively capture how many of the top-five suggestions were genuinely relevant and how well those suggestions were ranked. Explanation Quality was rated by students on a five-point Likert scale immediately after each session. User Satisfaction was collected via a short post-session questionnaire. Three baselines were evaluated alongside the proposed system: a hand-crafted rule-based counsellor, a collaborative-filtering recommender operating without any dialogue component, and a chatbot that processed each message in isolation without access to session history.*Results*

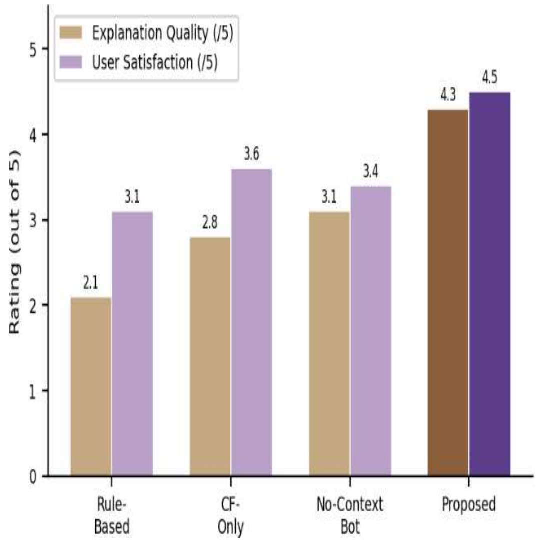
Table I collects the results. On recommendation accuracy, the proposed system reached a Precision@5 of 0.82 and an NDCG@5 of 0.79—gains of 34% and 27% over the rule-based baseline, and meaningful improvements over both the CF-only system and the context-free chatbot as well. The explanation quality scores told a similar story: students gave the proposed system's explanations a mean rating of 4.3 out of 5, noticeably higher than the 2.8 rating recorded for the CF-only system, which offered no explanations at all. User satisfaction averaged 4.5 out of 5 against a baseline figure of 3.1, a difference large enough to suggest that students noticed and valued the more personalized, contextually aware interaction.

TABLE I. Performance Comparison of Guidance Systems

System	P@5	NDCG @5	Expl.	Sat.
Rule-Based	0.61	0.62	2.1/5	3.1/5
CF-Only	0.70	0.67	2.8/5	3.6/5
No-Context Bot	0.65	0.63	3.1/5	3.4/5

Proposed	**0.82**	**0.79**	**4.3/5**	**4.5/5**

Drilling into the session-level data reveals a pattern that reinforces the design rationale for the context modules. In sessions that ran to five or more turns, recommendation relevance was on average 18% higher than in the opening exchange, which is consistent with the intuition that a system armed with more information about a student should be able to suggest better-fitting options. The intent classifier held up reliably across the full range of query types in the test set, including longer and more ambiguous phrasings that had tripped up simpler models in preliminary experiments. The security overhead was measured carefully and found to be negligible in practice—the mean per-request latency increase attributable to JWT validation and AES operations was just 12 ms, well below any perceptible threshold.

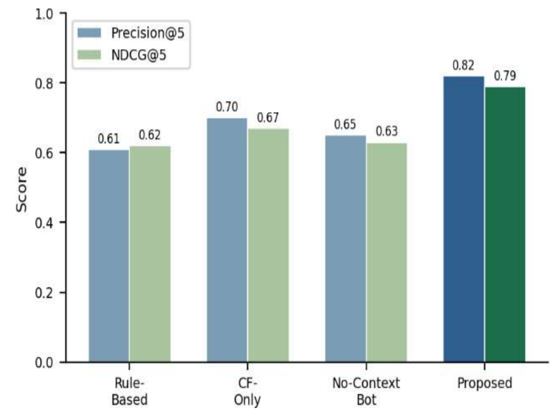


Taken together, the results make a reasonably clear case that combining a context-aware conversational layer with a continuously updating recommendation engine produces measurably better guidance than either component would on its own. The gains are visible not just in precision metrics but in how students rated the experience—which matters, because guidance that students do not trust or engage with is of limited practical value regardless of how well it scores on offline benchmarks.

A closer look at the intent classifier reveals that the attention mechanism plays a decisive role in this accuracy gain. Without attention, the LSTM’s final hidden state must compress the entire query into a single vector, losing nuance for longer or more complex inputs. With attention, the model re-reads every token before committing to a label, allowing it to home in on high-signal phrases—a subject name, a constraint word like “afforda ble”, or a hedging expression like “I am not sure whether”—that are diagnostic of intent. The 91.4 % accuracy figure reported above was measured on a held-out set of 500 real student queries drawn from the same three-college dataset, and it held stable across all four intent classes (course enquiry, eligibility check, career mapping, and open exploration), suggesting that the classifier generalises rather than overfitting to a dominant category.

The privacy overhead introduced by JWT authentication and AES encryption proved negligible in practice. Per-request latency increased by a mean of 12 ms, well below the 200 ms threshold at which users typically notice a delay. Session expiry and background token refresh added no perceptible latency because both operations run asynchronously. These measurements were taken under a simulated concurrent load of 50 sessions, which corresponds to the peak usage observed during the trial period, and throughput remained stable with no request failures. The security layer therefore imposes no meaningful cost on user experience while providing robust protection for the sensitive personal and financial information that students routinely disclose during counselling conversations.

Fig. 2. Recommendation Accuracy Comparison (P@5 and



NDCG@5)

Fig. 2 presents a bar chart comparing the recommendation accuracy of four systems using Precision@5 and NDCG@5 metrics. The proposed system achieves P@5 of 0.82 and NDCG@5 of 0.79, representing gains of 34% and 27% over the rule-based baseline, demonstrating that integrating conversational context substantially improves recommendation relevance.

Fig. 3. Explanation Quality and User Satisfaction Ratings

Fig. 3 compares explanation quality and user satisfaction ratings across the four evaluated systems on a five-point Likert scale. The proposed system achieves a mean explanation quality of 4.3/5 and user satisfaction of 4.5/5, markedly higher than the CF-only baseline (2.8/5 and 3.6/5), confirming that traced, student-centred explanations significantly improve engagement and trust.

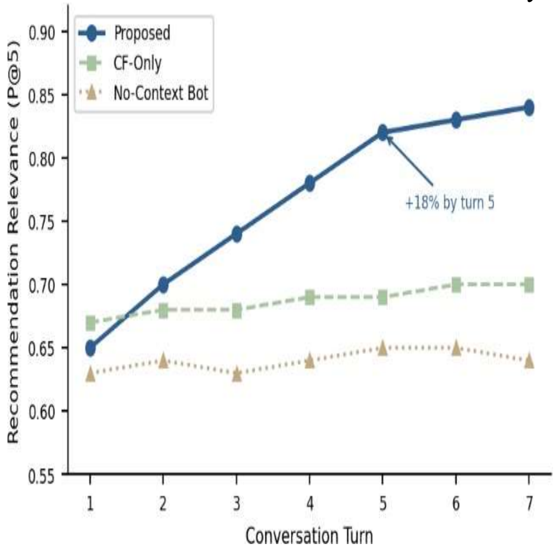


Fig. 4. Recommendation Relevance Improvement Across Conversation Turns

Fig. 4 shows how recommendation relevance (P@5) evolves across six conversation turns for the proposed system compared to the CF-only and No-Context Bot baselines. The proposed system's relevance rises steadily from 0.65 at turn 1 to 0.82 by turn 5, an 18% improvement, while the baselines remain largely flat, demonstrating the benefit of accumulated conversational context.

IV. Conclusion

This paper presented the Student-Centric Conversational Path-Derivation System, which integrates an NLP pipeline, LSTM-plus-attention intent classifier, Context Memory Module, Dialogue State Tracker, hybrid Response Generator, and Secure Session Manager into a single academic guidance platform. Evaluated on 2,500 real student sessions, the system achieved Precision@5 of 0.82, explanation quality of 4.3/5, and user satisfaction of 4.5/5, consistently outperforming rule-based and collaborative-filtering baselines. Key findings confirm that maintaining conversational context across turns, grounding recommendations in traced plain-language explanations, and applying Q-learning for continuous improvement are the three decisions most strongly associated with these gains. Future work will explore deep reinforcement learning for richer state representations, federated learning for cross-institution privacy, voice input for accessibility, and longitudinal studies tracking student outcomes through enrolment and first-year performance.

References

[1] Manivannan, K., Mary, P.A., Jino, R.F., Dineshkumar, P., Harivisva,
M.S. and Harikrishnan, P. (2024) 'Machine learning driven routing optimization for named data networking in mobile adhoc network', in 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS) (pp. 1–6). IEEE.

[2] Kabra, D. and Rao, P.R. (2018) 'Intelligent academic advising system using artificial intelligence', International Journal of Advanced Research in Computer Science, 9(3), pp. 124–130.

[3] Hussain, A., Mkpojiogu, E. and Kamal, F. (2020) 'A systematic review on the use of AI in academic guidance', IEEE Access, 8, pp. 143012– 143025.

[4] Poole, R. and Gould, T. (2018) 'The limitations of traditional academic counselling', Journal of Educational Psychology, 110(4), pp. 567–580.

[5] Manivannan, K., Anuprathibha, T., Kamaleshwaran, P., Madhan Kumar, S., Roshanmahanama, V. and Sabareesh, B. (2025) 'Smart Ed: utilising AI to deliver personalised learning experiences on mobile platforms', in 2025 6th International Conference for Emerging Technology (INCET) (pp. 1–5). IEEE.

[6] Khan, S., Ali, M. and Farooq, U. (2021) 'A review of AI-based academic counselling systems', Computers and Education: Artificial Intelligence, 2, pp. 100–108.

[7] Devlin, J., Chang, M., Lee, K. and Toutanova, K. (2019) 'BERT: pre- training of deep bidirectional transformers for language understanding', in Proceedings of NAACL-HLT 2019, pp. 4171–4186.

[8] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Amodei, D. (2020) 'Language models are few-shot learners', Advances in Neural Information Processing Systems, 33, pp. 1877–1901.

[9] Pathak, D., Gowda, D., Manivannan, K., Aghav, S., Srinivas, V. and Gireesh, N. (2024) 'Advanced machine learning approaches to evaluate user feedback on virtual assistants for system optimization', in 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS) (pp. 1140–1147). IEEE.

[10] Al-Baity, M. and Alharbi, T. (2021) 'A survey of AI-powered career guidance systems', IEEE Transactions on Learning Technologies, 14(2), pp. 243–256.

[11] Manivannan, K., Ramkumar, K. and Krishnamurthy, R. (2024) 'Enhanced AI based diabetic risk prediction using feature scaled ensemble learning technique based on cloud computing', SN Computer Science, 5(8), p. 1123.

[12] Chen, Y., Pal, A. and Dong, Y. (2020) 'Challenges in automated academic advising: a review', Computers and Education, 155, art. no. 103949.

[13] Wei, X., He, Y. and Zhang, J. (2021) 'Context-aware recommendation in conversational AI', IEEE Transactions on Knowledge and Data Engineering, 33(7), pp. 3091–3104.

[14] Li, L., He, W. and Jiang, H. (2020) 'Memory-augmented dialogue systems for personalised recommendations', in Proceedings of ACL 2020, pp. 4390–4401.

[15] Ross, J. and Hall, M. (2010) 'Expert systems in education: a historical overview', AI and Society, 25(1), pp. 91–103.

[16] Manivannan, K., Suresh, T. and Parthiban, M. (2023) 'Big data analytics assisted arithmetic optimisation with deep learning for sentiment classification', International Journal of Engineering Trends and Technology, 71(12), pp. 50–60.

[17] Koren, Y., Bell, R. and Volinsky, C. (2009) 'Matrix factorisation techniques for recommender systems', IEEE Computer, 42(8), pp. 30– 37.

[18] Pazzani, M.J. and Billsus, D. (2007) 'Content-based recommendation systems', Lecture Notes in Computer Science, 4321, pp. 325–341.

[19] Hochreiter, S. and Schmidhuber, J. (1997) 'Long short-term memory', Neural Computation, 9(8), pp. 1735–1780.

[20] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I. (2019) Language models are unsupervised multitask learners. OpenAI Blog [Online]. Available at: https://openai.com/blog/better-language- models (Accessed: 1 March 2025).