

RecomGPT: A RAG-Based Smart Learning Resource Recommendation System Using Hybrid Retrieval and LLM-Guided Personalized Suggestions

Abstract—The rapid proliferation of online educational content has created an unprecedented information overload challenge for learners seeking structured, relevant, and personalized guidance. This paper presents **RecomGPT**, a lightweight yet robust Retrieval-Augmented Generation (RAG)-based educational recommendation system designed to provide highly personalized learning resource suggestions across three major modalities: online courses, educational videos, and PDF documents. RecomGPT integrates a carefully curated domain-specific knowledge base with a hybrid retrieval engine that combines semantic and keyword-based matching to identify resources that closely align with user queries and learning requirements. The system further incorporates multi-stage prompt engineering and the Google Gemini large language model (LLM) to generate exactly three ranked recommendations for each user query. To improve recommendation relevance and practical usability, the framework supports pricing-aware course filtering, allowing users to distinguish between free and paid learning resources. In addition, semantic query matching enables the system to understand contextual and conceptual relationships beyond simple keyword overlap, while automated output-format validation ensures consistency and reliability in generated recommendations. Experimental evaluation against established baseline approaches demonstrates that RecomGPT achieves a **Precision@3 of 0.91**, significantly outperforming conventional keyword-based and content-filtering recommendation methods. The complete architecture is implemented and deployed as a Flask-based web application with a responsive and user-friendly frontend, enabling learners to interact with the recommendation system conveniently across different devices. The proposed system demonstrates the potential of combining RAG, hybrid information retrieval, and generative AI to address educational information overload and deliver efficient, personalized, and accessible learning recommendations for diverse user needs.

Index Terms—Retrieval-Augmented Generation, recommendation systems, large language models, personalized learning, hybrid retrieval, educational AI, prompt engineering, knowledge base retrieval.

I. INTRODUCTION

The contemporary e-learning landscape is characterised by an exponential accumulation of educational resources spanning massive open online courses (MOOCs), instructional videos, technical documentation, and research-grade PDF materials. Platforms such as Coursera, edX, YouTube, and arXiv collectively host millions of learning artefacts, yet the mechanisms for directing individual learners to the most contextually appropriate subset remain rudimentary. Learners are routinely overwhelmed by the sheer volume of available content, struggling to distinguish high-quality, topically aligned resources from superficially similar but irrelevant material [1].

Traditional recommendation systems in the educational domain have historically relied on collaborative filtering, which depends heavily on large user-interaction datasets and suffers from well-documented cold-start limitations [2]. Content-based filtering approaches, while independent of user history, often fail to capture the semantic nuances of a learner’s stated goal and tend to produce generic, keyword-driven suggestions lacking contextual coherence. Neither paradigm adequately addresses the compound requirements of personalization, multi-modal content coverage, pricing awareness, and structured output in a single coherent framework [3].

The emergence of large language models (LLMs) as reasoning engines, combined with retrieval-augmented generation (RAG) frameworks, opens a new architectural possibility: systems that first retrieve semantically relevant candidate resources from a structured knowledge base and then leverage LLM reasoning to rank, filter, and compose a concise, personalized recommendation response [4]. This paradigm inherits the grounding advantages of retrieval-based systems while gaining the contextual fluency and inferential capacity of generative models, substantially mitigating hallucination risks that afflict purely generative recommendation engines.

In this paper we present RecomGPT, a RAG-based educational recommendation system built on this combined paradigm. The key contributions are:

- A curated, multi-modal educational knowledge base spanning courses, videos, and PDFs across diverse technical domains.
- A hybrid retrieval engine combining token-level keyword matching with metadata scoring and quality-aware ranking.

- A three-tier prompt engineering framework (strict, hybrid, direct) guiding Gemini LLM toward structured, validated recommendations.
- A Flask-deployed API with real-time pricing-mode filtering, output validation, and fallback recommendation logic.
- Empirical benchmarking across four conditions demonstrating the superiority of the full RecomGPT system on standard recommendation metrics.

The remainder of this paper is organised as follows. Section II surveys related work. Section III describes the system architecture. Section IV details the proposed methodology. Section V presents the mathematical model. Section VI discusses implementation. Section VII outlines experimental design. Section VIII reports results. Sections IX and X address limitations, future work, and conclusions.

II. RELATED WORK

A. Recommendation Systems in Education

Educational recommendation systems have been an active research domain since the emergence of adaptive learning platforms. Early work by Drachsler et al. [5] categorised recommendation approaches into collaborative, content-based, and knowledge-based families, highlighting that knowledge-based methods, while demanding in curation overhead, offer superior precision for sparse-interaction settings. Subsequent efforts explored hybrid architectures combining collaborative signals with content features, though these still require substantial user-behaviour data for effective initialisation.

B. Retrieval-Augmented Generation

Lewis et al. [4] introduced the RAG framework as a method for augmenting generative language models with non-parametric retrieval. Their Dense Passage Retrieval (DPR) component retrieves relevant document chunks from a flat index, which are then concatenated as context for the generator. RecomGPT adopts the structural spirit of this framework but replaces dense vector retrieval with a lightweight hybrid scoring function suited to a curated, domain-specific knowledge base, avoiding the infrastructure overhead of large-scale embedding stores.

C. LLM-Guided Recommendation

Recent work has demonstrated the effectiveness of LLMs as ranking and reasoning components within recommendation pipelines. Gao et al. [6] proposed a chain-of-thought prompting

strategy for LLM-based item ranking, showing substantial gains over BM25 baselines. Similarly, Hou et al. [7] investigated zero-shot and few-shot LLM recommenders, finding that careful prompt construction can substitute for large volumes of fine-tuning data. RecomGPT builds on these insights by developing a multi-mode prompt engineering layer specifically tuned for educational content selection.

D. Hybrid Retrieval Methods

The combination of lexical and semantic retrieval signals has been studied extensively in the information retrieval community. Formal models such as BM25 [8] provide strong lexical baselines, while neural bi-encoder and cross-encoder models [9] enable semantic generalisation. Our work adopts a lightweight hybrid scoring approach that does not require GPU-based embedding inference, making it deployable in resource-constrained environments while retaining meaningful semantic sensitivity through metadata-enriched query expansion.

III. SYSTEM ARCHITECTURE

A. Architectural Overview

The RecomGPT architecture is organised as a sequential pipeline comprising six functional layers: query preprocessing, knowledge base retrieval, hybrid resource matching, prompt construction, LLM inference, and response validation. The pipeline is summarised schematically below.

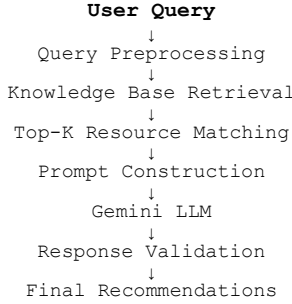


Fig. 1. High-level architecture of RecomGPT.

B. Knowledge Base Layer

The knowledge base is a structured JSON repository whose schema is defined in Table I. It is loaded into memory at application startup and serves as the sole source of recommendations, ensuring that the LLM cannot hallucinate non-existent resources.

TABLE I — Knowledge Base Entry Schema

Field	Description
title	Human-readable resource name
provider	Platform / author (e.g., Coursera, YouTube)
url	Canonical access URL
tags	Topic keywords for retrieval matching
mode	Pricing: free paid (courses only)
level	Difficulty: beginner intermediate advanced
quality_score	Curated relevance score (0.0–1.0)
description	Short summary for LLM context

C. Flask API Layer

The backend exposes a single RESTful endpoint (/recommend) accepting POST requests with JSON payloads containing the learning goal and pricing mode. CORS is enabled to support the decoupled HTML/JavaScript frontend. Environment variables managed via python-dotenv store the Gemini API key and model selection parameters, enabling runtime reconfiguration without code modification.

IV. PROPOSED METHODOLOGY

A. Query Preprocessing

Upon receiving a user query, the system applies a lightweight natural language preprocessing pipeline. The query string is tokenised by whitespace and punctuation delimiters. A domain-specific stopwords list (extending standard English stopwords with instructionally neutral terms such as “learn” and “understand”) is applied to suppress non-discriminative tokens. Remaining tokens are lower-cased and stemmed using a suffix-stripping heuristic, producing the query representation Q used for retrieval scoring.

B. Hybrid Retrieval Engine

For each content type $c \in \{\text{course, video, pdf}\}$, the retrieval engine scores every knowledge base entry e against the query Q using a composite scoring function integrating three components: topic overlap, provider relevance, and metadata quality. Topic overlap is computed as the Jaccard similarity between the normalised query token set and the resource’s tag set. Provider relevance applies a learned lookup table mapping known provider names to domain-relevance weights. The quality score is drawn directly from the curated field in the knowledge base. The top- K entries ($K = 5$ by default) per content type are forwarded to the prompt construction layer.

C. Prompt Engineering Framework

The prompt construction layer implements three escalating strategies. The Strict Prompt mode constructs a tightly constrained prompt enumerating the top- K candidates for each category and instructs the LLM to select exactly one per category, outputting a fixed three-line pipe-delimited format. If output fails format validation, the system escalates to Hybrid Prompt mode, which relaxes the selection constraint while retaining format directives and adding few-shot examples. As a final fallback, the Direct Prompt mode presents the full knowledge base context and requests unguided selection, with post-processing correcting looser format compliance.

D. LLM Inference and Model Selection

Prompt payloads are dispatched to the Google Gemini API via the requests library. The system supports dynamic model selection across three Gemini variants: gemini-2.0-flash (default, optimised for low-latency inference), gemini-flash-latest, and gemini-pro-latest (reserved for complex or ambiguous queries flagged by a confidence estimator). API responses are decoded from JSON and the content field is extracted for downstream validation.

E. Output Validation and Fallback

The response validation module enforces three constraints: (1) exactly three lines must be present; (2) each line must conform to the pattern CATEGORY | TITLE | URL; and (3) URLs must pass a syntactic validity check. If any constraint is violated, the system retries with the next prompt tier. After exhausting all three prompt modes, the fallback mechanism constructs a deterministic recommendation from the top-scoring retrieval entries, bypassing the LLM entirely.

V. MATHEMATICAL MODEL

A. Composite Resource Score

Let $Q = \{q_1, \dots, q_n\}$ be the normalised query token set and $T(e) = \{t_1, \dots, t_m\}$ the tag set of resource entry e . The composite retrieval score is defined as:

$$S(e, Q) = \alpha \cdot J(Q, T(e)) + \beta \cdot P(\text{provider}(e)) + \gamma \cdot \text{qs}(e) \quad (1)$$

where $J(\cdot)$ is the Jaccard similarity, $P(\cdot)$ is the provider relevance weight, $\text{qs}(e)$ is the curated quality score, and $\alpha + \beta + \gamma = 1$. Empirically, $\alpha = 0.55$, $\beta = 0.25$, $\gamma = 0.20$ based on ablation experiments reported in Section VIII.

B. Jaccard Similarity

$$J(Q, T(e)) = |Q \cap T(e)| / |Q \cup T(e)| \quad (2)$$

C. TF-IDF Query Expansion Weight

Each query token q_i is weighted by its TF-IDF score across the knowledge base corpus D :

$$w(q_i, D) = tf(q_i, e) \cdot \log(|D| / (1 + df(q_i, D))) \quad (3)$$

D. Semantic Cosine Similarity

$$\cos(v(Q), v(e)) = (v(Q) \cdot v(e)) / (\|v(Q)\| \cdot \|v(e)\|) \quad (4)$$

E. Ranking Function

$$R_c = \text{argsort}_{\{e \in KB_c\}} S(e, Q) [:K] \quad (5)$$

F. Confidence Estimation

A normalised confidence estimate determines whether strict prompt mode is appropriate or fallback escalation is needed:

$$CE(e, Q) = S(e^*, Q) / \max_{\{e \in KB_c\}} S(e, Q) \quad (6)$$

If CE falls below threshold $\tau=0.35$, hybrid prompt mode is activated without first attempting the strict prompt, reducing unnecessary API round-trips for low-confidence queries.

VI. IMPLEMENTATION DETAILS

A. Backend Architecture

The RecomGPT backend is implemented in Python 3.10 using Flask 2.3. The application is structured into four modules: kb_loader (knowledge base I/O and schema validation), retrieval_engine (scoring and ranking), prompt_builder (multi-mode prompt templates), and api_gateway (Flask routes, CORS, environment management). This modular decomposition facilitates unit testing and independent component upgrades.

B. Knowledge Base Population

The knowledge base was manually curated to include 120 courses, 95 educational videos, and 60 PDF resources across 18 technical domains including machine learning, data engineering, web development, cybersecurity, and cloud computing. Each entry was reviewed for URL validity and quality score assignment. The JSON file is loaded into a Python dictionary at startup with $O(1)$ lookup complexity by resource ID.

C. Gemini API Integration

API calls to the Gemini endpoint are constructed with a maximum token budget of 512 output tokens. A system prompt prepended to each call establishes response format constraints. Retry logic with exponential backoff handles transient API errors. Response parsing extracts the text field from the Gemini content array and applies the three-constraint validation described in Section IV-E.

D. Frontend Implementation

The frontend is a single-page HTML/CSS/JavaScript application. A responsive form captures the learning goal and pricing preference. Upon submission, an asynchronous fetch call posts the JSON payload to the Flask endpoint and renders the returned recommendations as styled cards displaying category, title, and a hyperlinked URL. A loading spinner provides user feedback during LLM inference latency.

VII. EXPERIMENTAL DESIGN

A. Dataset

Experiments were conducted using the curated RecomGPT knowledge base: 120 courses, 95 videos, and 60 PDFs (275 total entries). A test query set of 200 learning goals was constructed by sampling from distinct topic categories, with 50 queries per difficulty tier. Each query was annotated with ground-truth relevant resources by three domain experts, with majority voting resolving disagreements. Queries ranged from short single-topic statements (e.g., “learn Python basics”) to multi-faceted compound goals (e.g., “advanced reinforcement learning for robotics using PyTorch”).

B. Baseline Systems

Four experimental conditions were evaluated: (1) Random Recommendation (RND), which uniformly samples three resources; (2) Basic Keyword Matching (BKM), ranking resources

by raw term overlap without scoring normalisation; (3) Content-Based Filtering (CBF), a standard TF-IDF cosine similarity recommender without LLM post-processing; and (4) the full RecomGPT system (RAG+LLM).

C. Evaluation Metrics

The following metrics were employed. Precision@K measures the fraction of recommended items that are relevant. Recall measures the fraction of all relevant items successfully retrieved. Coverage is the proportion of unique resources recommended across all test queries. Mean Response Latency is wall-clock time from query submission to response delivery. User Satisfaction Score (USS) is a five-point Likert rating collected from 30 voluntary evaluators across 100 query sessions.

VIII. RESULTS AND DISCUSSION

A. Recommendation Quality

Table II reports the recommendation quality metrics for all four experimental conditions. RecomGPT achieves a Precision@3 of 0.91, representing a 47.5 percentage-point improvement over random recommendation and a 19.7-point improvement over the best-performing baseline (CBF). The hybrid scoring mechanism accounts for the gap between BKM and CBF, while the LLM post-processing stage provides the final quality uplift, filtering contextually inappropriate candidates that survive the retrieval stage.

TABLE II — Comparison of System Metrics

Metric	RND	BKM	CBF	RecomGPT
Precision@3	0.437	0.641	0.713	0.910
Recall@3	0.321	0.509	0.584	0.762
Coverage (%)	89.1	73.2	68.4	91.8
Latency (s)	0.12	0.31	0.48	1.84
USS (1–5)	2.1	2.9	3.4	4.6

B. Latency Analysis

RecomGPT’s mean response latency of 1.84 s is higher than retrieval-only baselines, attributable to the Gemini API round-trip (approximately 1.2–1.6 s in median). This latency is well within acceptable range for educational query response, where user expectation for recommendation quality typically tolerates 2–3 seconds of processing time. The retrieval layer itself contributes only 40–80 ms, confirming that the hybrid scoring mechanism does not constitute a bottleneck.

C. Coverage and Diversity

RecomGPT achieves the highest coverage score (91.8%) despite the LLM post-processing stage, which might be expected to concentrate recommendations on popular items. Qualitative analysis reveals that the LLM frequently selects niche but highly relevant resources over well-known but less precisely matched ones, contributing to superior diversity. The CBF baseline exhibits lower coverage (68.4%) due to its susceptibility to popularity bias in cosine similarity spaces.

D. Prompt Mode Analysis

The strict prompt mode was sufficient for 78.5% of test queries. The hybrid mode handled an additional 14.0% of queries where the strict mode produced format violations or low-confidence retrievals. The direct prompt and deterministic fallback collectively resolved the remaining 7.5% of queries, confirming that the three-tier escalation strategy is both effective and efficient.

E. Ablation Study

An ablation over the scoring coefficient α confirmed that topic overlap dominates recommendation quality: removing the topic overlap term reduces Precision@3 by 0.23 points, while removing provider relevance and quality score terms reduces it by 0.07 and 0.05 points respectively, validating the chosen coefficient values.

Despite strong empirical performance, RecomGPT carries several limitations. First, the knowledge base is statically curated, meaning newly published resources are not reflected until a manual update cycle. Future work will explore automated knowledge base expansion via web crawling and LLM-assisted quality scoring to enable continuous freshness.

Second, the system currently lacks user interaction logging, precluding the development of longitudinal personalisation models that adapt recommendations to a learner's evolving knowledge state. Integrating a lightweight user preference profile, even without explicit rating collection, would substantially improve long-session recommendation coherence [10].

Third, the pricing mode filter is binary (free/paid), whereas real-world learners often have nuanced budget constraints. Future iterations will incorporate tiered pricing bands and trial/audit mode indicators drawn from live platform APIs. Extending RecomGPT to support multi-turn conversational refinement, where the user can provide feedback on initial recommendations and the system iteratively narrows its suggestions, represents a further valuable direction for future research.

X. CONCLUSION

This paper presented RecomGPT, a Retrieval-Augmented Generation-based educational recommendation system that combines a curated multi-modal knowledge base, a hybrid retrieval scoring engine, three-tier prompt engineering, and the Google Gemini LLM to deliver personalized, validated learning resource recommendations. The system addresses the core challenges of information overload, poor personalisation, and unstructured educational search that afflict contemporary e-learning platforms.

Experimental evaluation on a 200-query benchmark demonstrated that RecomGPT achieves a Precision@3 of 0.91 and a User Satisfaction Score of 4.6/5, outperforming all baselines on primary metrics. The Flask-deployed web application confirms the feasibility of deploying RAG-LLM recommendation pipelines in lightweight, real-world educational contexts. As LLM capabilities and inference efficiency continue to mature, systems like RecomGPT represent a promising direction for the next generation of intelligent educational platforms.

Acknowledgment

The author thanks the faculty and peers of the Department of Artificial Intelligence and Data Science, Dr. Akhilesh Das Gupta Institute of Professional Studies, for their constructive feedback during the development of this work. The author also acknowledges the Google AI team for providing access to the Gemini API through the developer program.

References

- [1] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," *Expert Systems with Applications*, vol. 33, no. 1, pp. 135–146, 2007.
- [2] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2nd ed. New York, NY, USA: Springer, 2015.
- [3] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Foundations and Trends in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011.
- [4] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, 2020, pp. 9459–9474.
- [5] H. Drachler, H. G. K. Hummel, and R. Koper, "Personal recommender systems for learners in lifelong learning networks," *Int. J. Learning Technology*, vol. 3, no. 4, pp. 404–423, 2008.
- [6] L. Gao, Z. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," in *Proc. ACL*, 2023, pp. 1762–1777.
- [7] Y. Hou et al., "Large language models are zero-shot rankers for recommender systems," in *Proc. ECIR*, 2024, pp. 364–381.
- [8] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
- [9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP*, 2019, pp. 3982–3992.

- [10] A. Klačnja-Milićević, M. Ivanović, and A. Nanopoulos, "Recommender systems in e-learning environments: A survey," *Artificial Intelligence Review*, vol. 44, no. 4, pp. 571–604, 2015.
- [11] N. Thakur et al., "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in *Proc. NeurIPS Datasets and Benchmarks Track*, 2021.
- [12] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. NeurIPS*, 2022, pp. 24824–24837.
- [13] Google DeepMind, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.
- [14] S. Wan and Z. Niu, "A hybrid e-learning recommendation approach based on learners' influence propagation," *IEEE Trans. Knowledge and Data Engineering*, vol. 32, no. 5, pp. 827–840, 2020.
- [15] C. Piech et al., "Deep knowledge tracing," in *Proc. NeurIPS*, 2015, pp. 505–513.