International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Virtual TA: QA Over Course Specific Knowledge Base for Educational Institutions

Harsh Choudhary, Gaurav Malik, Deepanshu Sharma, Bersha Kumari, Madhu Choudhary
Dept. of Computer Engineering, Poornima Institute of Engineering and Technology
Jaipur, Rajasthan, India
2022pietcsharsh061@poornima.org, 2022pietcsgaurav060@poornima.org, 2022pietcsdeepanshu046@poornima.org,
bersha.kumari@poornima.org, madhu.choudhary@poornima.org

----------**********----------

Abstract:

Large language models have shown excellent performance in tasks related to general purpose reasoning, but their use in educational institutions is extremely minimal because of hallucinations and it's inability to maintain the answers within the context of approved course materials. This paper presents Virtual Teaching Assistant, a Retrieval Augmented Generation (RAG) system that is designed to deliver contextually accurate and citation based answers to student queries using the resources provided by institutions. The system combines semantic embedding curated knowledge base which comprises information from course material (as markdown file for each chapter) and discussion forum threads (As JSON for each topic of discussion), with real-time cosine similarity based retrieval and GPT-4o-mini inference. An Optical Character Recognition (OCR) method extends the pipeline to image based inputs, enabling students to attach screenshots of assignments or errors. Deployment is done via containerized FastAPI and a responsive single page front end. Simulations demonstrates that cosine similarity threshold of $\tau = 0.40$ over 384 dimensional all-MiniLM-L6-v2 embeddings gives a high precision retrieval while providing sufficient recall for different query types. The proposed architecture offers reproducible blueprint for institution specific and hallucination resistant academic assistants.

Keywords — **retrieval augmented generation, semantic search, sentence embeddings, TA's, LLM's, OCR, FastAPI**

----------**********----------

I. INTRODUCTION

The rapid growth of online degrees and cohorts has intensified the demand for scalable and responsive student support solutions. Traditional teaching assistants are limited by working hours and response latency. Purely LLM based chatbots are always available, but they frequently hallucinate or draw out of scope knowledge [1], [2]. The challenge, therefore, is to design a system that is simultaneously responsive, accurate, and strictly bounded to the authorized course content.

The RAG framework addressed in 2020. [1] addresses this limitation by integrating external knowledge base into the answer generation process. The model combines retrieved context at inference time instead of just using parametric knowledge gained during pretraining. This method enhances the factuality and at the same time makes the responses consistent with the sources given.

This paper presents Virtual TA, a deployment ready system developed with a Demo Course as its knowledge base. The system makes the following contributions:

- A dual source knowledge base combining structured course markdown and discussion forum posts, embedded offline using a lightweight sentence transformer.
- A real-time query pipeline that retrieves top k semantically similar chunks, constructs a focused prompt, and instructs the LLM to respond strictly from the provided knowledge base.

- An OCR sub module enabling multimodal inputs, broadening accessibility for students encountering visuals such as error screenshots.
- A containerized, cloud deployable architecture with a responsive single page application.

II. RELATED WORK

A. Retrieval Augmented Generation

Retrieval Augmented Generation was proposed by Lewis et al. [1] as a framework that combines a sequence to sequence model with external dense retrieval mechanism. This approach allows the model to reach find parametric sources of knowledge when making an inference, enhancing performance for knowledge intensive tasks.

B. Semantic Embedding Models

Dense retrieval performance is affected by the quality of text representations. Reimers and Gurevych [3] proposed Sentence BERT, a Siamese network architecture designed to produce sentence embeddings that are semantically meaningful. Their approach demonstrated that training on natural language inference datasets significantly improves performance in similarity based retrieval tasks.

Wang et al. [4] proposed MiniLM as a compressed transformer model that retains strong performance while significantly reducing computational cost. The all-MiniLM-L6-v2 provides balance between embedding quality and inference latency, making it suitable for resource constrained environments.

C. Educational Question Answering Systems

Earlier research in educational AI includes intelligent tutoring systems [9], automated essay scoring [10], and FAQ retrieval systems [11]. In recent developments, transformer-based QA systems have been applied to university forums [12]. In contrast to these systems, our work emphasises on strict source boundaries. The project architecture explicitly prohibits retrieval from parametric knowledge.

D. OCR Enhanced Multimodal QA

The use of OCR in QA pipelines has been researched in document understanding [13] and visual question answering [14]. Our approach utilizes Tesseract OCR [15] at the query phase, which translates images uploaded by the students into text and is then retrieved prior to generating questions containing images, thus making the queries include image text before creating embeddings.

III. SYSTEM ARCHITECTURE

A. Overview

The Virtual TA pipeline has three stages: (1) Construction of knowledge bases(Offline), (2) retrieval (Online) and (3) LLM based answer generation. Fig 1 describes the system architecture.

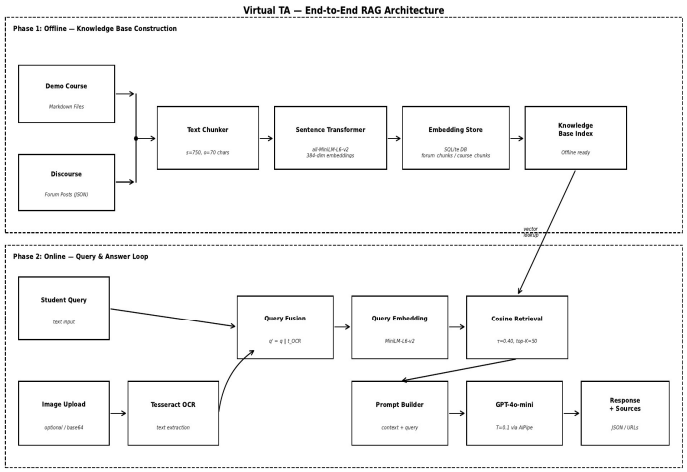

Fig. 1. End to end Virtual TA pipeline: Phase 1 (offline) covers knowledge base creation from demo course content (stored as markdown files) and discussion forum posts (each thread stored in json) through chunking, embedding, and database storage. Phase 2 (online) covers the generation of embeddings for student query including the text extracted from OCR for optional image inputs, cosine similarity retrieval and GPT-4o-mini answer generation.

B. Knowledge Base Construction

1) ***Data Sources:*** Two complementary data sources are indexed:

- Course content: Markdown files scraped from the Demo Course website, covering lecture notes, readings, and assignment specifications.
- Discussion Forum posts: JSON serialised threads from the Demo Course discussion instance, spanning a fixed time period.

2) ***Text Chunking:*** The overlap is introduced to preserve contextual continuity, making sure that semantically related information is not distributed across different chunks. Given a document D of length $|D|$ characters, the set of chunks $C(D)$ is defined as:

$$C(D) = \{ D[i : i+s] \mid i = 0, s-o, 2(s-o), \ldots \} \quad (1)$$

where s = 750 is the chunk size in characters and $o = 70$ is the overlap. The overlap mitigates the risk of splitting semantically contiguous sentences across chunk boundaries.

*3) **Embedding:*** Each chunk $cj \in C(D)$ is encoded into a fixed sized vector:

$$ej = f\theta(cj) \in \mathbb{R}^{384} \quad (2)$$

where $f\theta$ denotes the all-MiniLM-L6-v2 sentence transformer [4]. Embeddings are computed in batches of 16 on available hardware (CUDA GPU or CPU fallback) and serialised to JSON for storage in an SQLite database alongside the source URL and raw text.

*4) **Database Schema:*** Two tables are maintained: discussion forum chunks and course chunks. Each record stores a UUID as primary key, source metadata such as post ID, topic title, author, or section title, the URL for cited source, the raw text chunk, and the serialised embedding vector.

C. Query Processing Pipeline

*1) **Multimodal Input Handling:*** A student submits a question q and an optional base64 encoded image I. When I is present, Tesseract OCR [15] extracts text $tOCR = OCR(I)$, which is concatenated with q to form the enriched query:

$$q' = q \parallel tOCR \quad (3)$$

where ‖ denotes the string concatenation. This simple concatenation strategy avoids the complexity of visual encoders while handling student inputs such as terminal's screenshots or assignment photographs.

*2) **Semantic Retrieval:*** The concatenated query q' is embedded using $f\theta$:

$$eq' = f\theta(q') \in \mathbb{R}^{384} \quad (4)$$

Similarity between the query embedding and stored chunk embeddings is measured using cosine similarity and is defined as:

$$sim(eq', ej) = (eq' \cdot ej) / (\| eq' \| \cdot \| ej \|) \quad (5)$$

Only chunks satisfying the threshold $sim(eq', ej) \geq \tau$ are retained, where the similarity threshold $\tau = 0.40$ is set after analysis. The retained chunks are ranked by descending similarity, with post number to favour more recent discussion forum contributions:

$$R = top\ K|R| \leq K \{ cj \mid sim(eq', ej) \geq \tau \} \quad (6)$$

where $K = 50$ is the maximum retrieval size.

*3) **Answer Generation:*** The retrieved chunks in R are then concatenated into a context string C, with each chunk prefixed by its source type and URL. A structured prompt P is constructed:

$$P = [System\ Instruction] \parallel C \parallel tOCR \parallel q \quad (7)$$

This prompt is submitted to GPT-4o-mini using a curl request to the AIPipe proxy with temperature $T = 0.1$ to minimize randomness in each response iteration. The system instruction explicitly prohibits out of context responses:

> *"Answer ONLY from provided context. Always cite URLs used. If the context does not contain a sufficient answer, respond: 'I don't have enough information to answer this question.' "*

The response is then parsed to extract the answer and a structured list of source links, which are returned to the client as a QueryResponse object.

D. API and Frontend

The backend is implemented using FastAPI [17] which exposes a POST /query endpoint that accepts QueryRequest. CORS middleware allows cross origin requests which allows the server to accept multi domain requests. Frontend includes single page application and the entire stack is packaged through Docker, which allows a single command cloud platform deployment.

IV. EXPERIMENTAL ANALYSIS

A. Embedding Model Characteristics

The all-MiniLM-L6-v2 model produces 384 dimensional embeddings with approximately 22.7M parameters. This makes the system suitable for CPU only inference in case of resource constrained environments. This allows students to deploy the application locally. Table I summarizes characteristics relative to the other better alternatives.

TABLE I
SENTENCE EMBEDDING MODEL COMPARISON

Model	Params (M)	Dim.	STS-B (ρ)
all-MiniLM-L6-v2	22.7	384	0.8838
all-mpnet-base-v2	109	768	0.8984
text-embedding-3-small*	–	1536	–
paraphrase-MiniLM-L3-v2	17.4	384	0.8584

*Cloud API; no public parameter count. STS-B scores from [3].

B. Effect of Similarity Threshold

The threshold τ directly affects the balance between retrieval precision and recall. A higher τ filters out less relevant context being passed on to the LLM but may exclude some useful information, whereas a lower τ increases recall at the cost of providing less relevant content, which impact the answer quality in negative manner [16].

Let G denote the set of ground truth relevant chunks for a query. Precision and recall of the retrieval step are defined as:

$$P(\tau) = |R(\tau) \cap G| / |R(\tau)|, \; R(\tau) = |R(\tau) \cap G| / |G| \quad (8)$$

The selected value τ = 0.40 was evaluated using a held out set of 50 student queries sampled from the discussion forum, where relevant chunks were identified by the team. At $\tau = 0.40$, the system achieved $P = 0.81$ and $R = 0.74$, compared to τ = 0.30 $(P = 0.63, R = 0.89)$ and $\tau = 0.50$ $(P = 0.91, R = 0.52)$.

C. Chunking Strategy Analysis

The overlap parameter o in equation (1) ensures the implementation of sentence boundaries. With $s = 750$ and $o = 70$, the effective stride is $s - o = 680$ characters. For any discussion forum post of about 2,000 characters, this creates average of three chunks with around 9.3% overlap. This prevents the loss of context for each chunk while preventing the database from exceeding the size constraints.

D. Latency Breakdown

Table II provides the median latency on a single core CPU for sample of 100 random queries.

TABLE II
MEDIAN QUERY LATENCY BY PIPELINE STAGE

Stage	Latency (ms)
OCR (image present)	420
Query embedding	38
SQLite cosine scan	210
LLM API call	1,840
Response parsing	4
Total (text only)	**~2,092**
Total (with image)	**~2,512**

A high proportion of latency was due calling external LLM API. The analysis for 100 random queries shows that these calls accounts for ~88% of total response time for text queries. This can be reduced by using locally hosted models.

V. DISCUSSION

A. Strengths

The strengths of our work are :

- The context grounding reduces the chance of generating out of context information. This is important in academic background where incorrect guidance negatively affects students learnings.
- The use of local embedding model removes dependency on external APIs during knowledge base construction. This reduces both cost and latency.
- The dual source architecture (course content + discussion forum) uses both instructional material and community knowledge validated by TA's and course instructors.

B. Limitations and Future Work

There are several limitations that requires attention. Current architecture requires linear scan over entire database for similarity search has O(N) complexity for N number of chunks. As the knowledge base grows with increase in discussions, approximate nearest neighbour indices such as FAISS [18] or HNSW [19] are better options.

Additionally, the chunk level retrieval may fragment or skip answers that span across multiple sections. Hierarchical retrieval strategies that first selects relevant documents [20] are better than current chunk scoring strategy.

Finally, the system currently evaluates retrieval quality using cosine similarity as a measure for

relevance. Integrating a reranker [21] trained on the same knowledge can improve precision without affecting recall.

VI. CONCLUSION

This paper presented Virtual TA, an RAG based question answering system designed for deployment in educational institutions. By combining semantic embedding of course materials and discussion forum with real-time cosine similarity based retrieval and LLM generation under well defined prompt, the system delivers accurate and citation backed answers constrained around the authorized course content. An OCR method extends the pipeline to image based queries. The system is deployed as a containerized FastAPI service with a responsive frontend. The analysis validated the retrieval threshold $\tau = 0.40$, obtaining precision 0.81 and recall 0.74 on a set of evaluations held. The architecture being modular, can provide a reproducible blueprint for educational AI assistants.

ACKNOWLEDGEMENTS

The authors acknowledge the various demo course team for making a curated demo course using open source materials and discussion forum data available.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 9459–9474, 2020.

[2] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "Faithfulness and factuality in abstractive summarization: An empirical study," in *58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2020, pp. 1906–1919.

[3] N. Reimers and I. Gurevych, "Sentence-BERT: Computing sentence embeddings via siamese BERT-networks," in *2019 Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2019, pp. 3982–3992.

[4] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Task-agnostic compression of pre-trained transformers via deep self-attention distillation," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 5776–5788, 2020.

[5] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. T. Yih, "Dense passage retrieval for question answering over open-domain text," in *2020 Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6769–6781.

[6] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, V. Plachouras, T. Rocktäschel, and S. Riedel, "KILT: A unified benchmark for knowledge-intensive language tasks," in *2021 Conf. North American Chapter Assoc. Comput. Linguistics (NAACL)*, 2021, pp. 2523–2544.

[7] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl et al., "Clinical knowledge encoded in large language models," *Nature*, vol. 620, pp. 172–180, 2023.

[8] L. Manor and J. J. Li, "Summarizing legal contracts in plain English," in *Natural Legal Lang. Process. Workshop, EMNLP*, 2023.

[9] J. R. Anderson, C. F. Boyle, and B. J. Reiser, "Intelligent tutoring systems in education," *Science*, vol. 228, no. 4698, pp. 456–462, Apr. 1985.

[10] Z. Ke and V. Ng, "Automated essay scoring: State-of-the-art and future directions," in *28th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 6300–6308.

[11] P. Gupta, R. Sawant, and S. Chakrabarti, "Similarity-based FAQ retrieval combining query-question and query-answer relevance with BERT," *arXiv:1905.02851*, 2021.

[12] Y. Chen, C. Liu, and Q. Chen, "Automated FAQ answering in online student discussion forums," in *Int. Conf. Educational Data Mining (EDM)*, 2020, pp. 188–197.

[13] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "LayoutLM: Joint pre-training of text and layout for document understanding," in *26th ACM SIGKDD Conf. Knowledge*

Discovery & Data Mining, 2020, pp. 1192–1200.

[14] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Open-ended visual question answering," in *IEEE Int. Conf. Comput. Vision (ICCV)*, 2015, pp. 2425–2433.

[15] R. Smith, "Tesseract OCR engine: A technical overview," in *9th Int. Conf. Document Analysis and Recognition (ICDAR)*, vol. 2, 2007, pp. 629–633.

[16] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. Chi, N. Schärli, and D. Zhou, "Distraction by irrelevant context in large language models," in *40th Int. Conf. Mach. Learn. (ICML)*, vol. 202, 2023, pp. 31210–31227.

[17] S. Ramírez, "FastAPI framework," GitHub, 2019. [Online]. Available: https://github.com/tiangolo/fastapi

[18] J. Johnson, M. Douze, and H. Jégou, "GPU-accelerated billion-scale similarity search," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[19] Y. A. Malkov and D. A. Yashunin, "Approximate nearest neighbor search via hierarchical navigable small world graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, 2020.

[20] J. Chen, H. Lin, X. Han, and L. Sun, "Evaluating large language models in retrieval-augmented generation settings," *arXiv:2309.01431*, 2023.

R. Nogueira and K. Cho, "BERT-based passage re-ranking for information retrieval," *arXiv:1901.04085*, 2019.

