Towards an Efficient, Customizable, and Accessible AI Tutor

Juan Segundo Hevia* JH216@RICE.EDU
Facundo Arredondo FA32@RICE.EDU
Vishesh Kumar VK38@RICE.EDU
Rice University

Abstract

We propose a novel AI tutoring system that combines a Retrieval-Augmented Generation (RAG) pipeline with a lightweight language model to provide efficient, customizable, and accessible educational support. Designed to operate offline with minimal computational resources, the system addresses the challenges faced by resource-constrained communities. To develop its knowledge capabilities, we explore various retrieval strategies starting from a knowledge base of college textbooks. This work lays the foundation for developing adaptable and equitable AI tutoring solutions that bridge educational gaps and empower learners in under-resourced communities.

Keywords: RAG, LLMs, Customizable AI Tutor, Education, On-Device AI

1. Introduction

In recent years, the integration of large language models (LLMs) into education has shown great potential for enhancing accessibility and engagement in learning (Kasneci et al., 2023). However, LLMs that excel in engagement and knowledge metrics (Zhao et al., 2024) often require significant computational resources. This poses a challenge for low-income and remote communities, where access to reliable internet and high performance computing is limited, exacerbating educational inequities. For this reason, a primary focus in the field has been the development of Small Language Models (SLMs) designed for deployment on low-resource hardware. Models such as SmolLM (Allal et al., 2024b,a) and TinyLlama (Zhang et al., 2024) exemplify this trend by reducing parameter count at the expense of reduced knowledge representation capacity. In this work, we rely on Retrieval Augmented Generation (RAG) for grounding factuality in model responses. RAG has emerged as a powerful technique for integrating external knowledge corpora into LLM responses (Chen et al., 2023).

2. Pipeline Overview

The proposed closed system AI Tutor is an offline, computationally efficient system composed of two key components: a RAG pipeline and an SLM. The RAG pipeline ensures factuality by managing knowledge retrieval from a local corpus of the user's choice, using cosine similarity to provide contextually relevant information based on user queries (Wang et al., 2020). Meanwhile, the SLM delivers generative conversational capabilities, enabling engaging and human-like interactions. This design overcomes the limitations of SLMs (Li et al., 2024; Jin et al., 2024) by augmenting the input prompt with accurate, in-context knowledge, effectively balancing conversational fluency and factual precision.

* Corresponding Author

3. Evaluation

For our evaluation, we test a range of options for a language model: SmolLM 135M (Allal et al., 2024b), SmolLM2 135M and a larger SmolLM2 1.7B (Allal et al., 2024a). To focus on domain specific evaluation in Biology, we only consider *college* and *high school* Biology tasks in MMLU. We split the evaluation into two stages: *(i)* a comprehensive, full-pipeline performance evaluation with various experiments on retrieved-context and *(ii)* an initial step in addressing the SLM's struggle with large and noisy contexts.

3.1. Pipeline Performance

To construct a factual support for standard Biology coursework, we construct a database from the textbooks *Biology 2e* (Clark et al., 2018) *Biology AP* (Rye et al., 2016), *Concepts of Biology*(Fowler et al., 2013) from the OpenStax repository [1] We use Chroma[2] as our database of choice and split the knowledge corpora into blocks of size 300 tokens. We return the single most similar block and append it as context to the question ($k = 1$). To control for the effect of the added context, we compare the RAG pipeline version with a baseline comprising just the language model. A key challenge in using Retrieval-Augmented Generation (RAG) with small language models is ensuring that the retrieved context enhances, rather than obstructs, the model's ability to extract the correct answer. We hypothesized that long retrieved contexts could introduce noise in the form of extra tokens that distract the language model, making it harder to identify the informative pieces of context (Liu et al., 2023). To test this, we explicitly provided the correct answer (both as the multiple choice letter and the textual answer) to evaluate the model's ability to filter out noise and grasp the correct response.

	Pipeline Performance		**Option Answer**		**Text Answer**	
Model	RAG	Baseline	RAG	Baseline	RAG	Baseline
SmolLM 135M	20.5%	20.0%	26.6%	91.6%	20.7%	23.3%
SmolLM2 135M	21.8%	21.6%	40.3%	74.9%	22.9%	27.5%
SmolLM2 1.7B	33.0%	41.8%	73.1%	95.8%	29.5%	50.4%

Table 1: Performance results comparing setups with and without provided answers, using both letter-based and text-based formats. *Option answer* refers to providing one of `A, B, C,` or `D` hidden in the context; *Text answer* refers to including the actual text answer.

3.2. Improving the Retrieval Process

Semantic Chunking for RAG is a novel method designed to preserve the semantic coherence of retrieved contexts. The goal is to ensure that each chunk covers a single topic, maintaining internal consistency across blocks in a document database. We believe this approach is particularly beneficial for SLMs, as they are particularly sensitive to the prompts (Sinha et al., 2024). In our implementation, we embed each sentence in a corpus and group similar

1. openstax.org/k12/biology
2. github.com/chroma-core/chroma

sentences into cohesive chunks. One common recommendation is to use DBSCAN clustering (Qu et al., 2024), but this approach risks losing local context, particularly in structured texts like academic textbooks. To address this, we employ Transition-Based Dependency Parsing to analyze sentence structure, compare embeddings with neighboring sentences, and segment at an 80% similarity threshold (LangChain, 2024).

GraphRAG (Edge et al., 2024) improves retrieval by structuring knowledge as entities and relationships instead of isolated text blocks. Unlike traditional RAG, it enhances precision by matching queries with relevant concepts covered in the knowledge base rather than relying solely on semantic similarity between the raw text blocks, enabling more granular and knowledge-grounded results. We use GPT-4o-mini via the OpenAI API to extract structured representations of entities and relationships from textbook pages, limiting to two relationships per page to reduce hallucinations. Following MiniRAG (Fan et al., 2025), we embed nodes using text-embeddings-3-small for semantic similarity matching. At retrieval, user queries are matched to the graph's vectorized entities, returning the top-k most relevant nodes. This approach enhances factual accuracy and optimizes token efficiency for context augmentation.

Model	**Baseline**	**GraphRAG @ $k = 5$**	**Semantic @ $k = 2$**
SmolLM2 1.7B	41.85%	42.3%	35.7%

Table 2: Comparison of various retrieval methods.

Results evidence the potential of processing and curating the knowledge base during the creation of a retrieval pipeline. The GraphRAG retriever summarizes key concepts in the books and provides those that match the user query, enabling more efficient context augmentation.

4. Next Steps

Our experiments underscore the importance of a strategic approach to knowledge retrieval to ensure factuality. Building on this, we will continue refining the preprocessing of retrieval results using summarization and curation techniques. In particular, we plan to evaluate the retrieval component independently of the SLM, employing RAGAS (Es et al., 2023). This will enable us to objectively assess and optimize the retrieval segment of the pipeline. Subsequently, we will integrate the optimized retrieval system with an SLM, potentially distilling a LLM tailoring it for RAG. Our goal is to develop a compact, low-memory RAG pipeline capable of running on a Raspberry Pi or smartphone app, empowering individuals in resource-limited communities with accessible, on-device information through a comprehensive and adaptable tutoring system.

5. Code and Data Availability

All relevant code is available in Github[3]. The books used to build up the knowledge base were obtained from OpenStax[4]

3. github.com/JuanseHevia/accessible-ai-tutors
4. openstax.org/k12/biology

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Lewis Tunstall, Agustín Piqueres, Andres Marafioti, Cyril Zakka, Leandro von Werra, and Thomas Wolf. Smollm2 - with great data, comes great performance, 2024a.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. Smollm - blazingly fast and remarkably powerful, 2024b.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation, 2023. URL https://arxiv.org/abs/2309.01431.

Mary Ann Clark, Matthew Douglas, and Jung Choi. *Biology 2e*. OpenStax, Houston, Texas, 2018. URL https://openstax.org/books/biology-2e/pages/1-introduction. Accessed from: https://openstax.org/books/biology-2e/pages/1-introduction.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2024. URL https://arxiv.org/abs/2404.16130.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation, 2023. URL https://arxiv.org/abs/2309.15217.

Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. MiniRAG: Towards Extremely Simple Retrieval-Augmented Generation, January 2025.

Samantha Fowler, Rebecca Roush, and James Wise. *Concepts of Biology*. OpenStax, Houston, Texas, 2013. URL https://openstax.org/books/concepts-biology/pages/1-introduction. Accessed from: https://openstax.org/books/concepts-biology/pages/1-introduction.

Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, and Deyi Xiong. A comprehensive evaluation of quantization strategies for large language models, 2024. URL https://arxiv.org/abs/2402.16775.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103: 102274, 2023. ISSN 1041-6080. doi: https://doi.org/10.1016/j.lindif.2023.102274. URL https://www.sciencedirect.com/science/article/pii/S1041608023000195.

LangChain. How to split text based on semantic similarity, 2024. URL https://python.langchain.com/docs/how_to/semantic-chunker/. Accessed: 2025-02-19.

Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Evaluating quantized large language models, 2024. URL https://arxiv.org/abs/2402.18158.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Renyi Qu, Ruixuan Tu, and Forrest Bao. Is semantic chunking worth the computational cost?, 2024. URL https://arxiv.org/abs/2410.13070.

Connie Rye, Robert Wise, Vladimir Jurukovski, Jean DeSaix, Jung Choi, and Yael Avissar. *Biology*. OpenStax, Houston, Texas, 2016. URL https://openstax.org/books/biology/pages/1-introduction. Accessed from: https://openstax.org/books/biology/pages/1-introduction.

Neelabh Sinha, Vinija Jain, and Aman Chadha. Are small language models ready to compete with large language models for practical applications?, 2024. URL https://arxiv.org/abs/2406.11402.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020. URL https://arxiv.org/abs/2002.10957.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. TinyLlama: An Open-Source Small Language Model, 2024. URL https://arxiv.org/abs/2401.02385.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A Survey of Large Language Models, 2024. URL https://arxiv.org/abs/2303.18223.