RAG-Enhanced Large Language Model for Intelligent Assistance from Web-Scraped Data

Nikunj Kanataria
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research
Charotar University of Science and Technology (CHARUSAT)
Changa, Anand, India
nikunjthakkar2003@gmail.com

Kunj Pareshbhai Patel
Computer Science & Design Department
A.D. Patel Institute of Technology
Charutar Vidhya Mandal University (CVMU)
Vallabh vidyanagar, Anand, India
kunjpatel91012@gmail.com

Hetul Niteshbhai Patel
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research
Charotar University of Science and Technology (CHARUSAT)
Changa, Anand, India
hetulpatel1516@gmail.com

Parth Goel
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research
Charotar University of Science and Technology (CHARUSAT)
Changa, Anand, India
parthgoel.ce@charusat.ac.in

Krishna Patel
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research
Charotar University of Science and Technology (CHARUSAT)
Changa, Anand, India
krishnapatel.ce@charusat.ac.in

Dweepna Garg
Computer Engineering Department
Devang Patel Institute of Advance Technology and Research
Charotar University of Science and Technology (CHARUSAT)
Changa, Anand, India
dweepnagarg.ce@charusat.ac.in

***Abstract*— The explosion of online information necessitates efficient and accurate methods for retrieving and analyzing relevant data. This research proposes a novel framework that leverages Retrieval-Augmented Generation (RAG) techniques to address this challenge. The framework utilizes web scraping to extract information from various sources and employs advanced language models like Llama 3, Mistral, and Gemini to generate concise and informative summaries. Furthermore, the framework incorporates embedding models such as nomic-embed-text and Gemini embedding model text embedding 001 to create semantic representations of the retrieved data and utilize FAISS for efficient indexing and retrieval. The RAGAS framework was used to evaluate the performance of different LLMs, with Llama 3.1 demonstrating the highest accuracy at 86.67%. This framework has significant implications for various applications, including personalized education, where it can be used to provide students with tailored learning materials and personalized tutoring experiences.**

Keywords*— *Generative AI, Large Language models (LLMs), Retrieval augmented generation (RAG), Web scraping, Question-answering, RAGAS)

I. Introduction

As any sector becomes increasingly digitized, the volume of information available on the internet has grown significantly, transforming it into a valuable resource while also presenting significant challenges. Web platforms, including social media, news outlets, AI tools, and e-commerce sites, constantly generate and spread large amounts of data, ranging from text and images to videos [1]. Traditional information retrieval techniques often struggle to keep up with this rapid growth, frequently falling short in delivering relevant, timely, and user-specific insights [2]. In traditional data processing systems, models were limited by training data and were typically trained on rule-based NLP techniques [3]. Once trained, these systems couldn't easily incorporate new information, making it difficult to keep up with real-time changes. This created a gap, especially when analyzing recent events or shifts in public opinion. As a result, these systems often produced responses that were outdated or inappropriate for the specific context, limiting their ability to capture real-time updates in public sentiment. For instance, analyzing real-time public reactions to catastrophic events posed significant challenges that could not be quickly collected and analyzed [4]. In query-answering systems, web scraping enables the continuous collection of the most up-to-date data from various online sources, allowing the model to provide more timely and accurate responses [5]. This system can also act as a dynamic tutor, automatically retrieving, processing, and generating responses based on the latest available information. It continuously scrapes relevant data according to the user's needs, adapting to the user's query in real-time and providing accurate, live information. The integration of web scraping with large language models (LLMs) has revolutionized applications such as tutoring [6].

The RAG system, with scraped data, can be useful in a wide range of real-world applications. For example, it can be used in businesses to track customer opinions in real-time, analyze social media reactions to breaking news, or even assist in decision-making during rapidly unfolding events. By scraping web information for the RAG system, it provides a dynamic, real-time method for data preprocessing that overcomes the limitations of traditional models. This research framework could also support legal research by retrieving relevant case law and precedents, as well as real-time updates, helping lawyers and legal professionals build stronger cases [7].

This research addresses the shortcomings of traditional methods by offering a framework that combines web scraping

2024 9th International Conference on Communication and Electronics Systems (ICCES) | 979-8-3503-7797-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCES63552.2024.10859894

and LLMs to retrieve personalized, context-sensitive information. First, content will be scraped from the internet and converted into embeddings. It will then be stored in the FAISS vector database. When a user enters a query, the system performs a semantic search to find the top k contexts from the vector database and passes this content to the LLMs to generate personalized information. The goal is to demonstrate how these technologies can transform the way individuals and organizations access and leverage information for personalized insights. To evaluate the system, we leverage the RAGAS evaluation framework, creating a small dataset consisting of four columns: Question, Answer, Context, and Ground Truth. We utilized six different metrics that focus on various components of the system.

Main Contributions of our research work:

- This paper presents a RAG-powered digital tutor that can boost students' academic performance and general knowledge by delivering real-time, personalized information on a wide range of topics, continuously updated with current web data.
- This paper evaluates and compares the performance of three LLMs—Llama 3.1, Gemini-Pro, and Mistral—in RAG systems, with a focus on their application in the education domain.

Section II discusses relevant studies. Section III details the methodology. Section IV presents the results, along with a list of parameters, dataset descriptions, and evaluation metrics. Section V concludes the paper and discusses future work..

II. Literature Review

Traditional search structures often struggle to correctly interpret the context and semantics of consumer queries, resulting in inappropriate or imprecise outcomes. This issue has been significantly reduced with the introduction of transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers) [8] and GPT (Generative Pre-trained Transformer) [9]. BERT's ability to understand context bidirectionally and GPT's capability to generate coherent content have led to substantial improvements in search system performance.

Similarly, Retrieval-Augmented Generation (RAG) is a methodology that retrieves and generates information from large document datasets, ensuring contextually accurate outputs. This technique is particularly effective because the retrieved information is more coherent, and the generated text is contextually aware. In contrast to earlier techniques, which relied heavily on keyword matching and static data sources, modern systems using RAG and web scraping methods can dynamically retrieve information and generate contextually relevant responses in real-time. The author Singrodia et al. provide a comprehensive survey of web scraping applications, illustrating how web scraping has evolved, the different methods of scraping websites, and the advantages and disadvantages of web crawling [10]. A language model was developed that retrieves data from the web, integrates statistical semantic models, and supports multiple query levels. This model also facilitates the proximity analysis of specific terms in various datasets.

Kepping B et al. introduced a transformer-based embedding model for product personalization, designed to address the limitations of the zero-attention model (ZAM). The Transformer-based Embedding Model (TEM) actively adjusts the impact of personalization by incorporating usage questionnaires and purchase history [11]. Suhit Gupta et al. developed a content extraction system that uses DOM trees and W3C-defined interfaces, providing dynamic access to a document's structure rather than relying on simple HTML markup [12]. Qingyao Ai et al. provided a hierarchical embedding model capable of learning semantic representations for entities at various levels with associated language data [13]. Hema Yoga Narasimhan proposed a framework for search personalization that includes three modules: feature generation, normalization using the Lambda MART algorithm, and feature selection wrappers, which enhance scalability by optimizing accuracy with minimal computational overhead [14]. In a review paper, Chien-Chang Lin et al. discussed intelligent tutoring systems in education, focusing on issues such as data privacy concerns, potential biases in machine learning algorithms, and the challenges of developing reliable architectures [15]. Ashraf Alam highlighted the role of artificial intelligence in customizing educational content to meet individual student needs, emphasizing how AI algorithms provide immediate feedback and adjust content based on student performance [16]. Lastly, Fanqi Wan et al. proposed a knowledge fusion architecture that integrates the strengths of existing large language models (LLMs) into a unified model, which was tested on various publicly available benchmarks [17].

III. Methodology

This study aims to develop an AI system with the help of RAG architecture that incorporates real-time data from web scraping. This system is designed to answer questions and summarize information, primarily functioning as a tutor in education. The methodology comprises three primary components: web scraping, retrieval, and a generation module. These components work in harmony to provide accurate, contextually appropriate responses to the user's query. The following sections will demonstrate the proposed approach in detail.

A) Web Scraping

The process begins with web scraping, which automates the extraction of real-time data from websites as requested by users. For this study, the primary data source was one of the news websites that features all technology-related stories. As illustrated in Figure 1, the web scraping process involves three key steps:

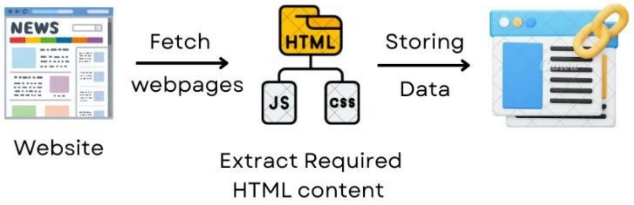

Fig.1 Web Scraping Process

Request Response:
Web scrapers initiate requests to the target website's server and wait for a response. The server then provides the requested data in HTML format.

Parse and Extract:
The HTML data received is parsed to extract the required information. This step involves identifying specific HTML elements—such as articles, comments, and metadata—that contain the relevant data.

Download Data:
Once extracted, the data is downloaded and stored in a structured format. This ensures that the information is organized and ready for further processing. Through this process, raw data is systematically gathered, cleaned, and organized into a comprehensive and up-to-date repository. Continuous content updates maintain the accuracy and relevance of the repository, providing a solid foundation for subsequent analysis and retrieval tasks.

The data retrieval component is the second major section of this study, as shown in Figure 2, following data collection and pre-processing. This component is integrated into the RAG-based question-answering architecture to efficiently identify and extract the most relevant information from the pre-processed data repository based on user queries. The retrieval component consists of several primary sub-components, such as Embedding Generation, Vector Indexing, and Query Optimization.

Embedding Creation:
Following embedding generation, the next step is vector indexing. This sub-component organizes the vector representations into a form that facilitates searching and retrieval. Techniques such as approximate nearest neighbour indexing are used to quickly access the most similar vectors in response to a query. This indexing method significantly reduces the search space and speeds up the retrieval process.

Vector Indexing:
Vector indexing follows embedding generation. This sub-component organizes the vector representations into a format that enables efficient searching and retrieval. Techniques such as approximate nearest neighbour indexing are used here to quickly access the most similar vectors in response to a query. This indexing method greatly reduces the search space and accelerates the retrieval process.

Query Optimization:
Query optimization is a critical sub-component that enhances both the performance and effectiveness of the retrieval process. It involves refining the user query to improve the relevance of the search results. Techniques such as query expansion, where additional relevant terms are appended to the query and relevance feedback, where user interactions inform refinements of future queries are employed. These techniques ensure that the system retrieves information that closely aligns with the user's intent.

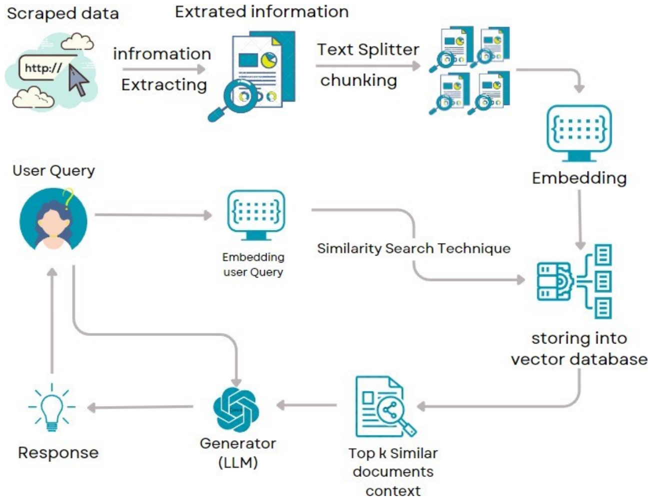

Fig.2 Workflow of System

1) Data Retrieval:

The final phase is the retrieval component. In this part, the optimized query is matched towards the listed vectors to identify and rank the most applicable chunks. Advanced retrieval algorithms, which includes those primarily based on cosine similarity or other distance metrics, are used to examine the query vector with the listed vectors. The chunks with the very best relevance ratings are then selected and forwarded to the generation module for in addition processing. Incorporating these advanced techniques, the data retrieval component ensures that the RAG system can efficiently and accurately extract the most relevant information from the data repository. This process complements the overall accuracy and reliability of the facts furnished, placing a stable basis forproducing coherent and contextually suitable responses in thesubsequent level.

C. Generator Phase:

The generation module represents the final phase of the RAG-based totally question-answering device. This aspect makes use of the document chunks retrieved inside the preceding section, which give relevant context, and inputs them into a generative language model along with the authentic consumer query. Transformer-based fashions, together with Mistral-7B, Gemini-Pro, and Llama3, that have been pre-trained on good sized datasets, are used on this section. To ensure optimal performance, a well-structured prompt template is used. this template helps connect the user's query with retrieved information, making it easier to generate more accurate and reliable responses. The input combined is the user's query, the prompt template, and the top retrieved document chunk that guides the model in producing specific relevant answers. It first generates to a summary created based on the context given by the retrieved document sections to provide a brief overview of the key information such that users can rapidly understand it and also in the right manner. Then, it will generate a relevant answer which can be accurate.

IV. Results & Discussion

A. Dataset Description

In this study, a custom dataset was developed to evaluate the RAG-based QA system specifically for handling data retrieved via web scraping. As shown in Table-I, the dataset consists of four columns: Question, RAG -Answer, Context, Ground-Truth. Question includes the user specific query related to content. Answer which is generated by RAG system, context is basically most relevant chunks which is converted during data preprocessing steps. The ground-truth is ideal answer for specific question. All the column except RAG-Answer and context were created very carefully to ensure transparent evaluation of Web scraping based RAG system.

TABLE I SAMPLE OF DATASET

Question	What are the primary concerns raised about using bytecode-based code generation in Python?
RAG Answer	The primary concerns raised about using bytecode-based code generation in Python are: Memory-unsafety: CPython is not memory-safe, and emitting bytecode directly can lead to memory-related issues. Stability: The bytecode API is not stable, and the availability and behavior of particular opcodes can vary between different CPython release versions.
Context	Some comments discussed Python bytecode generation and challenges, particularly focusing on maintenance and safety concerns.
Ground-Truth	Commenters pointed out the instability of Python bytecode across versions, making maintenance challenging. There were also concerns about memory safety when working directly with custom bytecode, which could introduce errors similar to those encountered inlanguages like Lua.

B. Hyperparamter Tuning

TABLE II HYPERPARAMETERS

Hyperparameters	Values
Document Chunk size	1000
Top k chunks	10
Sentence model	all-MiniLM-L6-v2
Maximum sentence length	1000

Hyperparameters are the pre define variable that one can control and that govern overall behavior of a RAG system. It is very crucial in determining the performance, accuracy, and efficiency of a RAG system. By controlling such as chunk size, overlapping, embedding model, retrieval parameters, directly influence how the system processes and retrieves information. Table-II presents all the hyperparameters with their correspond values.

C. Result Discussion:

TABLE III RESULTS

RAGAS Evaluation Framework	Llama3.1	Gemini-Pro	Mistral
Faithfulness	**91%**	85%	74%
Answer-relevancy	83%	**87%**	59%
Context-recall	**100%**	65%	92%
Context-precision	**100%**	69%	89%
Answer-correctness	71%	**78%**	69%
Answer-similarity	75%	**93%**	75%

This research utilizes RAGAS framework for evaluation. which Provides wide range of Evaluation metric which can assess overall RAG system. The metrics include Answer relevancy, Answer recall, Answer Similarity, Context recall, Context precision, Faithfulness. Each metric design to evaluate a specific component of the RAG systems. During the evaluation process it found that each model exhibited particular shortcomings in different components. Table III above present the overall results of each LLMs and embedding model, highlighting which model perform best in specific area.

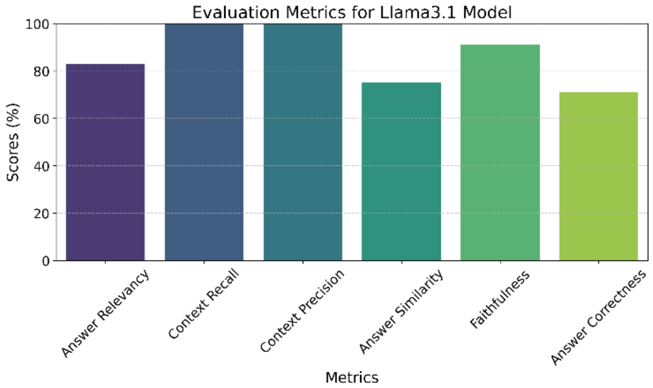

Fig.3 Results of Llama3.1

In the Figure 3, the Llama3.1 model is evaluated across six metrics. "Context Recall" and "Context Precision" score highly, indicating that the model effectively retains and retrieves information from the given context. However, "Answer Correctness" scores the lowest, suggesting that even though the model holds and processes contextual details well,it cannot always deliver fully accurate answers. This highlights a potential need for improvement in response accuracy. Llama3.1 (Figure 3), although more balanced, has a score of above 80% and also performs very well in ContextRecall, thus making it an all-rounder. However, the low Answer Correctness score at 65% indicates that most of the model responses are relevant but factually incorrect.

As shown in the Figure 4, has good results in understanding and remembering the context, as it scores high in Context Recalland Context Precision of about 90%,

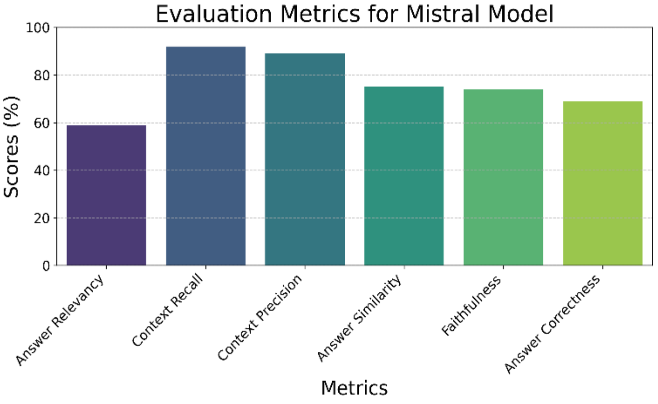

Fig.4 Results of Mistral

while yielding low scores in Answer Relevancy of 60% and Answer Correctnessof 70%. This indeed shows that though it is able to retrieve theproper context, it lacks either relevancy or correctness in an answer.

Figure 5 shows that Gemini-pro scores over 80% in both Answer Relevancy and Answer Similarity, indicating consistent and expected responses, it scores below 70%, reflecting challenges in handling tasks, which really need deep contextual understanding.

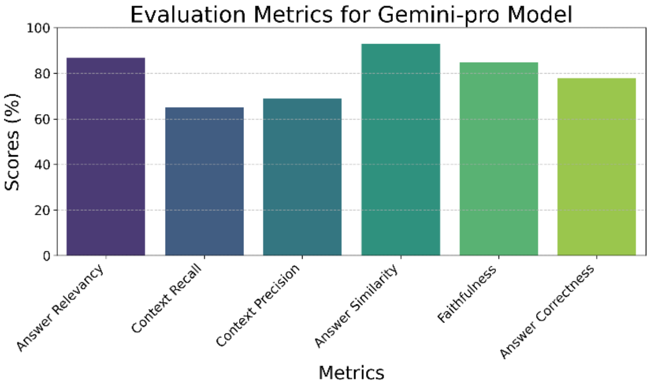

Fig.5 Result of Gemini-pro

Figure 6 plot the performance of AI models, Mistral, Llama3.1, and Gemini-pro, over six metrics: Answer Relevancy, Context Recall, Context Precision, Answer Similarity, Faithfulness, and Answer Correctness.

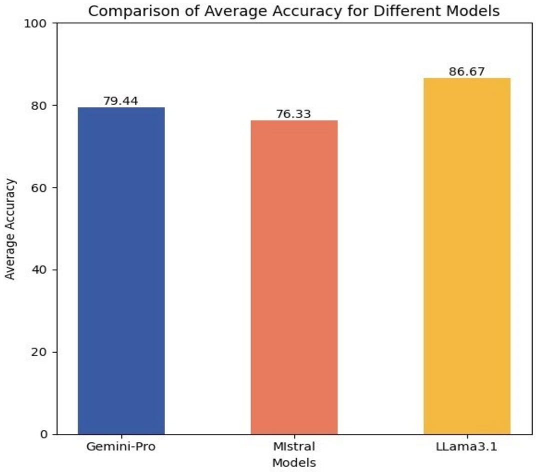

Fig.6 Overall performance of RAG system for web scraped data

Figure 6 presents the comparison of the three models—Gemini-pro, Mistral, and Llama3.1—it indicates that Llama3.1 stands out in overall performance. In particular, Llama3.1 performs exceptionally well when addressing matching responses with the original prompts, recalling relevant information, and staying consistent in terms of the accuracy of the answers provided. Accuracy scores, particularly in these categories, remain consistently high, which makes it the most dependable model for delivering responses that not only align with what is being asked but also remain reliable in content. In contrast, Mistral,while good at remembering and retrieving information, struggles somewhat when to ensure that answers directly address the questions being asked, as indicated by relatively lower score in relevancy. Meanwhile, Gemini-pro manages to provide answers that aresimilar in form but falls short in maintaining the same level of accuracy and recall as Llama3.1, especially when the context becomes more complex. In summary, Llama3.1 achieves a more consistent performance across all the evaluated metrics, making it the strongest model when looking at both the relevance and accuracy of responses.

V. Conclusion and Future Scope

In conclusion, Llama3.1 provides outstanding performance, making it appropriate for various applications such as tutor in education domain, though it is lacking in answer correctness and answer similarity compare to other models. Gemini-Pro is most effective in structured tasks where consistent and relevant outcomes are essential, but it faces challenges when dealing with intricate contexts. Mistral demonstrates outstanding results in tasks requiring comprehensive context comprehension, although there are opportunities for enhancement in producing more accurate and pertinent results. Overall Llama3.1 outperform Gemini-pro by 7.23% and 10.34%. This method may face problem with real-time retrieval of data in dynamic changing data due to the latency introduced by the model inference and embedding generation. Future work could focus on integrating advanced models to enhance context comprehension and answer relevancy, leveraging external knowledge bases to improve Llama3.1's factual reliability, and refining Gemini-Pro's ability to handle complex contexts. In addition, to explore various other real-world application in other domains such as in medical, e-commerce etc.

REFERENCES

[1] S. Mishra and A. Misra, “Structured and unstructured big data analyt- ics,” in 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC). IEEE, 2017, pp.740–746.

[2] M. Kobayashi and K. Takeda, “Information retrieval on the web,” ACM computing surveys (CSUR), vol. 32, no. 2, pp. 144–173, 2000.

[3] P. Goel and A. Ganatra, “A survey on chatbot: Futuristic conversational agent for user interaction,” in 2021 3rd international conference on signal processing and communication (ICPSC). IEEE, 2021, pp. 736– 740

[4] Y. Huang and J. Huang, “A survey on retrieval-augmented text generation for large language models,” arXiv preprint arXiv:2404.10981, 2024.

[5] A. Ahluwalia and S. Wani, “Leveraging large language models for webscraping,” arXiv preprint arXiv:2406.08246, 2024.

[6] C. Dong, “How to build an ai tutor that can adapt to any course and provide accurate answers using large language model and retrieval augmented generation,” arXiv preprint arXiv:2311.17696, 2023.

[7] V. Singrodia, A. Mitra, and S. Paul, “A review on web scrapping and its applications,” in 2019 international conference on computer communication and informatics (ICCCI). IEEE, 2019, pp. 1–6.

[8] J. D. M.-W. C. Kenton and L. K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in Proceedings ofnaacL-HLT, vol. 1. Minneapolis, Minnesota, 2019, p. 2

[9] T. B. Brown, “Language models are few-shot learners,” arXiv preprintarXiv:2005.14165, 2020.

[10] D. Hiemstra, “Using language models for information retrieval,” 2001.

[11] K. Bi, Q. Ai, and W. B. Croft, “A transformer-based embedding modelfor personalized product search,” in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1521–1524

[12] S. Gupta, G. E. Kaiser, P. Grimm, M. F. Chiang, and J. Starren, “Automating content extraction of html documents,” World Wide Web, vol. 8, pp. 179–224, 2005.

[13] Q. Ai, Y. Zhang, K. Bi, X. Chen, and W. B. Croft, “Learning a hierarchical embedding model for personalized product search,” in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 645–654.

[14] H. Yoganarasimhan, “Search personalization using machine learning,”Management Science, vol. 66, no. 3, pp. 1045–1070, 2020.

[15] C.-C. Lin, A. Y. Huang, and O. H. Lu, “Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematicreview,” Smart Learning Environments, vol. 10, no. 1, p. 41, 2023.

[16] A. Alam, “Harnessing the power of ai to create intelligent tutoring systems for enhanced classroom experience and improved learning outcomes,” in Intelligent Communication Technologies and Virtual Mobile Networks. Springer, 2023, pp. 571–591.

[17] F. Wan, X. Huang, D. Cai, X. Quan, W. Bi, and S. Shi, “Knowledge fusion of large language models,” arXiv preprint arXiv:2401.10491, 2024.