MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 5181-5184

LLM-Powered Textbook Answer Generation with Qdrant Vector Search

[1] **Ms. R. Saraswathi** Assistant Professor
Department of CST SNS COLLEGE OF TECHNOLOGY, Coimbatore-35, Tamilnadu, India, sarasrajagopal2@gmail.com
[2] **Parasuraman P**
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore-35, Tamilnadu, India velpazhani049@gmail.com
[3] **Manoj Boopathy K**
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore-35, Tamilnadu, India manojboopathy07@gmail.com
[4] **Dhanush Kumar M**
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore-35, Tamilnadu, India mathavandhanush341@gmail.com
[5] **Raja C**
Department of CST, SNS COLLEGE OF TECHNOLOGY, Coimbatore-35, Tamilnadu, India rajarrb8681@gmail.com

***Abstract*—** The rapid advancement of large language models (LLMs) has opened new avenues for building intelligent question-answering systems capable of delivering contextually accurate and pedagogically relevant responses. Traditional information retrieval systems in academic settings rely on keyword-based search and rigid database queries, which often fail to comprehend the semantic intent of learner queries and retrieve contextually irrelevant content. This results in fragmented knowledge delivery, reduced learner engagement, and limited educational effectiveness. This research proposes an LLM-powered textbook answer generation system that integrates Qdrant vector search with retrieval-augmented generation (RAG) to deliver accurate, context-aware responses from textbook content. The system encodes textbook chapters into dense vector embeddings using sentence transformer models and stores them in a Qdrant vector database for efficient semantic retrieval. When a learner poses a question, the system identifies the most semantically relevant textbook passages and feeds them to an LLM for coherent answer synthesis. The platform provides two core functionalities: semantic document retrieval for precise passage identification and LLM-based answer generation for natural language response synthesis. The proposed system was implemented using a scalable architecture consisting of a React.js frontend, FastAPI backend, and Qdrant vector database. Experimental evaluation demonstrates that the integration of vector search with LLM reasoning significantly enhances answer relevance, factual grounding, and educational usability, supporting students and educators in efficient knowledge retrieval.

Keywords— Large Language Models, Retrieval-Augmented Generation, Qdrant Vector Search, Textbook QA, Semantic Retrieval, Dense Embeddings, Educational AI, Natural Language Processing

I. INTRODUCTION

In the contemporary educational ecosystem, academic institutions generate vast volumes of structured and semi-structured learning content spanning textbooks, lecture notes, research papers, and digital course materials. Despite this abundance of knowledge resources, learners frequently encounter difficulties in locating precise answers to their queries, particularly when information is spread across multiple chapters or presented in complex technical language. Traditional search mechanisms embedded within learning management systems primarily rely on keyword matching and Boolean retrieval, which lack the ability to comprehend the semantic context of a question and retrieve genuinely relevant content.Conventional academic question-answering approaches often depend on manually curated FAQs, static answer keys, or simple lexical search tools. While such methods serve basic informational needs, they are incapable of handling the diversity and complexity of learner queries in real-time educational settings. As a result, students frequently receive incomplete or contextually misaligned answers, which hampers their ability to build comprehensive conceptual understanding. These limitations highlight the critical necessity for intelligent answer generation systems capable of understanding learner intent and synthesizing accurate responses from authoritative textbook sources. Recent advances in natural language processing, particularly the development of transformer-based large language models (LLMs) such as GPT-4, LLaMA, and Mistral, have fundamentally transformed automated text understanding and generation. These models possess remarkable capacity to comprehend complex linguistic structures, infer semantic relationships, and generate fluent, coherent natural language text. However, LLMs in isolation suffer from a critical limitation: they generate responses based on pre-trained parametric knowledge without access to current or domain-specific source documents. This results in hallucinated answers that may appear fluent but are factually incorrect or unsupported by authoritative material.Retrieval-Augmented Generation (RAG) addresses this fundamental limitation by coupling a retrieval component with the generative capability of LLMs. In a RAG framework, relevant document passages are first retrieved from an external knowledge base and provided as context to the LLM, which then synthesizes an answer grounded in the retrieved content. The effectiveness of RAG systems depends critically on the quality of the retrieval component. Vector search databases such as Qdrant enable high-performance semantic retrieval by encoding documents into dense numerical embeddings and supporting approximate nearest-neighbour search, thereby retrieving passages that are semantically similar to the learner query rather than merely lexically matching.This paper proposes an LLM-powered textbook answer generation system that integrates Qdrant vector search with state-of-the-art language model generation to deliver pedagogically grounded, semantically precise answers to learner queries. The system encodes textbook content into sentence-level embeddings, stores them in a Qdrant collection, retrieves the most relevant passages at query time, and synthesizes a comprehensive answer using an LLM. The proposed architecture supports scalable, real-time answer generation while ensuring factual grounding through retrieval-augmented context injection.

II. RELATED WORKS

Vaswani et al. [1] introduced the transformer architecture, which fundamentally redefined natural language processing by replacing recurrent structures with self-attention mechanisms. Their work established the theoretical foundation upon which modern large language models are built. The transformer's ability to model long-range dependencies and contextual relationships within text enabled subsequent models such as BERT and GPT to achieve unprecedented performance in text comprehension and generation tasks. This architectural innovation forms the backbone of the LLM component in the proposed system.Devlin et al. [2] proposed BERT, a bidirectional transformer pre-trained on large corpora using masked language modeling and next-sentence prediction objectives. BERT demonstrated that contextual embeddings derived from pre-trained models significantly outperform static word embeddings on downstream NLP tasks including question answering, named entity recognition, and textual entailment. Their findings established that pre-trained sentence representations carry rich semantic information suitable for passage-level retrieval and answer extraction.Lewis et al. [3] introduced Retrieval-Augmented Generation (RAG), a framework that combines parametric knowledge stored in LLM weights with non-parametric knowledge retrieved from external document collections. The RAG model conditions its generation on retrieved passages, reducing hallucination and improving factual accuracy. Their experiments on open-domain question answering benchmarks showed that RAG significantly outperforms closed-book LLMs on knowledge-intensive tasks. This work directly motivates the retrieval-first architecture adopted in the proposed system.Karpukhin et al. [4] proposed Dense Passage Retrieval (DPR), demonstrating that dense vector representations of passages outperform traditional BM25 sparse retrieval on open-domain question answering tasks. DPR trains dual-encoder models to project questions and passages into a shared dense embedding space where semantic similarity can be measured via dot product. Their results established that dense retrieval is more effective than keyword-based methods for capturing semantic intent, supporting the use of sentence transformer embeddings in the proposed architecture. Reimers and Gurevych [5] developed Sentence-BERT (SBERT), a modification of BERT that produces semantically meaningful sentence embeddings suitable for efficient semantic search and paraphrase detection. SBERT uses siamese network structures and contrastive training objectives to generate fixed-size embeddings where cosine similarity reflects semantic equivalence. The computational efficiency and semantic richness of SBERT embeddings make them highly suitable for large-scale document encoding in vector databases such as Qdrant.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 5181-5184

Johnson et al. [6] introduced FAISS, a library for efficient similarity search and clustering of dense vectors. Their work demonstrated that approximate nearest-neighbour algorithms based on product quantization and inverted file structures enable sub-linear retrieval from billion-scale vector collections. FAISS established the foundational principles of vector-based retrieval that modern purpose-built vector databases such as Qdrant extend with additional filtering, payload storage, and distributed indexing capabilities.Izacard and Grave [7] proposed Fusion-in-Decoder (FiD), a retrieval-augmented generation approach that encodes multiple retrieved passages independently and fuses their representations in the decoder. FiD demonstrated that conditioning generation on multiple retrieved contexts improves answer quality by providing complementary evidence. This multi-passage retrieval strategy informs the top-k retrieval configuration used in the proposed system, where multiple relevant textbook passages are concatenated as LLM context.Brown et al. [8] introduced GPT-3 and demonstrated that few-shot prompting of large language models enables competitive performance on diverse NLP tasks without task-specific fine-tuning. Their findings revealed that model scale correlates with emergent reasoning and in-context learning abilities. GPT-3's instruction-following capacity is central to the proposed system's answer synthesis pipeline, where retrieved textbook passages are provided as in-context evidence for grounded answer generation.akano et al. [9] proposed WebGPT, a system that trains language models to search the web and synthesize answers from retrieved web pages. WebGPT demonstrated the feasibility of grounding LLM generation in externally retrieved sources and showed that retrieval-grounded models produce more accurate and verifiable answers. This reinforces the importance of retrieval grounding in educational answer generation systems where factual accuracy and source attribution are essential.

Shi et al. [10] investigated the effect of irrelevant context on LLM reasoning and showed that language models can be distracted by semantically similar but factually incorrect passages. Their study highlights the importance of retrieval precision in RAG systems, motivating the use of high-quality semantic similarity scoring in Qdrant to ensure that only genuinely relevant textbook passages are provided to the LLM. Retrieval quality directly determines the factual grounding of generated answers.

Xiong et al. [11] proposed ANCE, an approach that uses the retrieval model itself to mine hard negatives for dense retriever training. ANCE demonstrated that iteratively updating the document index with the current retrieval model substantially improves retrieval quality. Their findings support the importance of using well-trained embedding models for initial document encoding, which the proposed system achieves through the use of pre-trained sentence transformer models.

Borgeaud et al. [12] introduced RETRO, a large language model augmented with retrieval from a trillion-token database. RETRO demonstrated that retrieval augmentation at inference time allows smaller models to achieve performance comparable to much larger parametric models. This finding reinforces the efficiency motivation of the proposed system: by grounding a moderately sized LLM with precise textbook retrieval, the system achieves high answer quality without requiring extremely large model parameters.

Mao et al. [13] proposed Generation-Augmented Retrieval (GAR), demonstrating that expanding queries with LLM-generated context before retrieval improves passage recall on knowledge-intensive tasks. GAR showed that enriched query representations capture additional semantic dimensions present in relevant documents but absent from the original query. This motivates future enhancements to the proposed system through query expansion before Qdrant retrieval.Ram et al. [14] proposed In-Context RALM, showing that retrieval-augmented LM prompting using off-the-shelf retrievers substantially improves language model perplexity and factual question answering without any fine-tuning. Their plug-and-play framework validates the design philosophy of the proposed system, which integrates Qdrant retrieval with a pre-trained LLM through prompt engineering rather than end-to-end training.Guu et al. [15] introduced REALM, a retrieval-augmented pre-training approach that jointly trains the language model and retriever using an unsupervised masked language modeling objective. REALM showed that integrating retrieval into pre-training enables the model to learn to retrieve relevant documents for factual queries. While the proposed system uses retrieval only at inference time, REALM's results confirm that retrieval-augmented generation produces answers with substantially higher factual accuracy than purely parametric generation.

III. ARCHITECTURE AND DESIGN

The architecture of the LLM-powered textbook answer generation system is designed to support seamless transformation of raw textbook content into contextually accurate answers through automated semantic retrieval and language model inference. The system follows a modular pipeline consisting of five core stages: document ingestion, embedding generation, vector storage, semantic retrieval, and answer generation. Each stage is carefully designed to balance retrieval precision, generation quality, system scalability, and user accessibility, ensuring that both students and educators can effectively leverage the platform.

A. Raw Textbook Document Corpus

he system initially receives raw textbook content collected from PDF documents, EPUB files, or plain-text academic resources. The corpus typically contains chapter text, section headings, definitions, theorems, and explanatory passages organized by subject domain. This content serves as the authoritative knowledge source for answering learner queries. The ingestion interface is designed to be intuitive, enabling educators to upload course materials without requiring technical expertise in natural language processing or database administration.

B. Input (Query / Subject Selection)

The platform allows learners to enter natural language questions and optionally select specific textbook subjects or chapter ranges for targeted retrieval. This user interaction enables focused answer generation and ensures that retrieved passages originate from pedagogically relevant content sections. The query interface supports free-form natural language input without requiring structured query syntax, making the system accessible to learners at all technical skill levels.

C. Document Chunking and Embedding Generation

Uploaded textbook documents are first segmented into overlapping text chunks of approximately 512 tokens, with a stride of 128 tokens to preserve cross-boundary semantic coherence. Each chunk is encoded into a 768-dimensional dense vector embedding using a pre-trained sentence transformer model. The embedding model maps semantically similar passages to proximate positions in the embedding space, enabling cosine similarity-based retrieval. This stage transforms raw textbook text into a structured numerical representation suitable for high-performance vector search.

D. Qdrant Vector Storage and Indexing

The generated embeddings are stored in a Qdrant vector database alongside their corresponding text payloads and metadata including chapter identifiers, page numbers, and subject tags. Qdrant organizes the vectors into HNSW (Hierarchical Navigable Small World) graph-based indices that enable approximate nearest-neighbour search with sub-millisecond latency at scale. The metadata filtering capability of Qdrant allows the system to restrict retrieval to specific subjects or chapters, improving retrieval precision for targeted learner queries.

E. Semantic Retrieval and Context Assembly

When a learner submits a question, the query text is encoded into a dense embedding using the same sentence transformer model applied during indexing. Qdrant performs an approximate nearest-neighbour search over the stored embeddings and returns the top-k most semantically similar passages ranked by cosine similarity score. The retrieved passages are assembled into a structured context prompt that provides the LLM with

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 5181-5184

authoritative textbook evidence for answer synthesis. The system defaults to retrieving the top five passages to balance contextual coverage with prompt length constraints.

F. LLM-Based Answer Generation

The assembled context prompt, consisting of retrieved textbook passages and the original learner query, is submitted to a large language model via API or local inference. The LLM is instructed through a system prompt to synthesize a comprehensive, pedagogically appropriate answer grounded exclusively in the provided context. The generated answer is streamed back to the user interface in real time, enabling responsive interaction. The LLM is configured with a low temperature setting to minimize hallucination and maximize factual fidelity to the retrieved textbook content.

Table 1: System Architecture Workflow

Stage	Component
Document Ingestion	PDF/EPUB Parser, Text Extractor
Chunking	Sliding Window Tokenizer (512 tokens)
Embedding	Sentence Transformer (all-mpnet-base-v2)
Vector Storage	Qdrant HNSW Index
Retrieval	Top-k Cosine Similarity Search
Generation	LLM (GPT-4 / Mistral / LLaMA)
Interface	React.js Frontend + FastAPI Backend

IV. METHODOLOGY

The proposed LLM-powered textbook answer generation system integrates document preprocessing, dense vector embedding, semantic retrieval, and language model generation to provide an end-to-end educational question-answering solution. The methodology focuses on converting raw textbook documents into a searchable semantic knowledge base that enables accurate, grounded answer synthesis in response to learner queries. Unlike traditional keyword-based retrieval tools that only locate documents containing query terms, the proposed approach emphasizes semantic similarity and LLM-based contextual reasoning. The overall methodology is organized into five major stages.

A. Dataset and Document Corpus Handling

The system accepts structured and unstructured textbook documents in PDF, EPUB, and plain-text formats. Documents are parsed using PyMuPDF and pdfplumber libraries to extract clean textual content while preserving chapter and section boundaries. For model evaluation, a corpus of 12 undergraduate textbooks across computer science, mathematics, and physics domains was assembled, comprising approximately 8,400 pages of academic content. The textbook corpus was divided such that chapters from eight books were used for system indexing while content from the remaining four books served as the source for evaluation query generation.

B. Text Chunking and Preprocessing

Before embedding generation, extracted textbook text undergoes preprocessing to ensure retrieval quality. This stage includes removal of headers, footers, page numbers, and figure captions, correction of hyphenation artifacts introduced during PDF extraction, and normalization of whitespace and special characters. The cleaned text is then segmented into overlapping chunks using a sliding window approach with a window size of 512 tokens and a stride of 128 tokens. Overlapping chunks ensure that answer-relevant content spanning section boundaries is not lost during segmentation. To ensure embedding quality, chunks shorter than 50 tokens or consisting primarily of numerical or tabular content are filtered out. The final processed corpus yielded approximately 94,000 text chunks from the 12-textbook collection.

C. Embedding Generation and Vector Indexing

Each preprocessed text chunk is encoded into a 768-dimensional dense vector using the all-mpnet-base-v2 sentence transformer model, selected for its strong performance on semantic similarity benchmarks. Batch encoding is performed using the SentenceTransformers library with GPU acceleration, processing approximately 2,400 chunks per minute. The generated embeddings are stored in a Qdrant collection configured with HNSW indexing parameters of m=16 and ef_construction=100, balancing retrieval accuracy and index build time. Each stored vector point includes a JSON payload containing the source chunk text, chapter title, book identifier, and page number to enable metadata-filtered retrieval.

D. Query Processing and Semantic Retrieval

At query time, the learner's natural language question is encoded using the same sentence transformer model to generate a query embedding. Qdrant performs an approximate nearest-neighbour search using the HNSW index with ef=128 to retrieve the top-k=5 most semantically similar chunks ranked by cosine similarity. Optional metadata filters can restrict retrieval to specific textbook subjects or chapter ranges based on learner-specified context. The retrieved passages are ranked by similarity score and concatenated in descending relevance order to form the retrieval context for LLM prompting.

E. Answer Generation and Response Synthesis

The assembled retrieval context and learner query are formatted into a structured prompt following a retrieval-augmented generation template. The system prompt instructs the LLM to generate a comprehensive, well-structured answer based solely on the provided textbook passages, explicitly discouraging speculation beyond the retrieved context. The answer generation employs a temperature of
0.2 and a maximum output length of 512 tokens to ensure concise, factually grounded responses. Generated answers are streamed to the frontend interface in real time using server-sent events. Each generated answer is accompanied by source citations indicating the specific textbook, chapter, and page number from which supporting passages were retrieved.

V. RESULTS AND DISCUSSION

A. Experimental Setup

The proposed system was evaluated using a curated benchmark of 500 natural language questions derived from the four held-out textbooks in the evaluation corpus. Questions were manually formulated by subject matter experts across difficulty levels: 35% conceptual, 40% applied, and 25% analytical. Each question was accompanied by a reference answer extracted directly from the corresponding textbook passage to enable objective evaluation. System performance was measured across three dimensions: retrieval quality, answer accuracy, and user experience.

B. Retrieval Performance

The Qdrant-based semantic retrieval component achieved a Precision@5 of 0.87 and a Recall@5 of 0.83 across all evaluation questions, demonstrating that the dense embedding approach consistently retrieves relevant textbook passages within the top-5 results. Retrieval performance was notably higher for conceptual questions (Precision@5: 0.91) compared to analytical questions (Precision@5: 0.81), reflecting the greater semantic complexity of multi-step reasoning queries. The HNSW-based approximate nearest-neighbour search achieved an average query latency of 12 milliseconds on the 94,000-chunk index, confirming real-time operational feasibility.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 5181-5184

C. Answer Generation Quality

Generated answers were evaluated using ROUGE-L, BERTScore, and human relevance ratings on a five-point Likert scale. The system achieved a mean ROUGE-L score of 0.61, a BERTScore F1 of 0.84, and a mean human relevance rating of 4.2 out of 5.0. Comparative analysis against a keyword-based retrieval baseline and a standalone LLM without retrieval demonstrated that the proposed RAG architecture substantially outperforms both alternatives. The standalone LLM produced factually incorrect answers on 31% of domain-specific questions, while the proposed system reduced this error rate to 8%, confirming the effectiveness of retrieval grounding in educational answer generation.

Table II. Model Performance Comparison

System Type	ROUGE-L	BERTScore F1	Human Rating (/5)
Keyword Search + LLM	0.42	0.71	3.1
BM25 Retrieval + LLM	0.51	0.76	3.5
Standalone LLM (No Retrieval)	0.38	0.68	2.9
Qdrant RAG (Proposed System)	**0.61**	**0.84**	**4.2**

D. Usability and System Effectiveness

Beyond generation quality metrics, usability was evaluated through a structured user study involving 45 undergraduate students and 12 faculty members. Participants rated the system on ease of use, answer clarity, source trustworthiness, and overall satisfaction. The system received a mean usability score of

4.3 out of 5.0, with particularly high ratings for answer clarity (4.5) and source attribution (4.4). Faculty participants highlighted the citation mechanism as a significant trust-building feature, noting that the ability to trace generated answers to specific textbook pages enhanced academic credibility. Students appreciated the conversational interface and the ability to refine queries without reformulating them in structured query syntax. The system processed queries with an end-to-end response latency of 1.8 seconds on average, demonstrating practical responsiveness for educational deployment.

E. Limitations

Although the system produced promising results, certain limitations were observed. First, the answer generation quality is bounded by the content coverage of the indexed textbook corpus. Queries seeking information absent from the uploaded documents cannot be answered accurately. Second, the current chunking strategy is purely token-count-based and does not account for semantic paragraph boundaries or hierarchical document structure. Third, the system presently supports only English-language textbooks. Extending the embedding and generation pipeline to multilingual academic content would substantially broaden the platform's applicability in diverse educational contexts.

VI. CONCLUSION AND FUTURE WORK

The proposed LLM-powered textbook answer generation system presents an intelligent educational question-answering framework that integrates Qdrant vector search with retrieval-augmented generation to deliver accurate, grounded, and pedagogically relevant answers from authoritative textbook sources. By encoding textbook content into dense semantic embeddings, storing them in a high-performance Qdrant vector database, and conditioning LLM generation on retrieved passages, the system effectively bridges the gap between large-scale language model capabilities and domain-specific factual grounding.The evaluation demonstrated that the proposed RAG architecture substantially outperforms keyword-based retrieval and standalone LLM baselines on answer accuracy, factual grounding, and user satisfaction metrics. A key contribution of this work is the demonstration that purpose-built vector search databases such as Qdrant can serve as highly effective retrieval backends for educational RAG systems, providing sub-millisecond semantic retrieval from large textbook corpora with flexible metadata filtering. The web-based deployment architecture enables institutional adoption without requiring specialist infrastructure expertise, supporting scalable rollout across academic departments and institutions.Future work will focus on three primary directions. First, the integration of query expansion and hypothetical document embedding (HyDE) techniques will be investigated to improve retrieval recall for analytically complex queries. Second, a multi-hop retrieval mechanism will be developed to handle questions requiring synthesis of evidence from multiple textbook chapters. Third, the system will be extended to support multilingual textbook corpora through multilingual sentence transformer models, broadening its applicability to non-English academic

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. NeurIPS, vol. 30, 2017.
[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, pp. 4171–4186, 2019.
[3] 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," in Proc. NeurIPS, vol. 33, pp. 9459–9474, 2020.
[4] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, pp. 6769–6781, 2020.
[5] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, pp. 3982–3992, 2019.
[6] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, 2021.
[7] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in Proc. EACL, pp. 874–880, 2021.
[8] T. B. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, vol. 33, pp. 1877–1901, 2020.
[9] R. Nakano et al., "WebGPT: Browser-assisted question-answering with human feedback," arXiv:2112.09332, 2021.
[10] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. H. Chi, N. Schärli, and D. Zhou, "Large language models can be easily distracted by irrelevant context," in Proc. ICML, pp. 31210–31227, 2023.
[11] L. Xiong et al., "Approximate nearest neighbor negative contrastive estimation for dense text retrieval," in Proc. ICLR, 2021.
[12] S. Borgeaud et al., "Improving language models by retrieving from trillions of tokens," in Proc. ICML, pp. 2206–2240, 2022.
[13] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen, "Generation-augmented retrieval for open-domain question answering," in Proc. ACL-IJCNLP, pp. 4089–4100, 2021.
[14] O. Ram et al., "In-context retrieval-augmented language modeling," Transactions of the Association for Computational Linguistics, vol. 11, pp. 1316–1331, 2023.
[15] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "REALM: Retrieval-augmented language model pre-training," in Proc. ICML, pp. 3929–3938, 2020.
[16] S. 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.
[17] M. E. Peters et al., "Deep contextualized word representations," in Proc. NAACL-HLT, pp. 2227–2237, 2018.
[18] T. Kwiatkowski et al., "Natural Questions: A benchmark for question answering research," Transactions of the Association for Computational Linguistics, vol. 7, pp. 452–466, 2019.