iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2026 **Issue:** Conference **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.83736

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com
National Conference on Innovation in Engineering and Technology-(NCIET2026)

Dynamic Retrieval-Augmented Generation (RAG) Chatbot System for Citation-Aware, Document-Grounded Conversational AI

Gaurinandan C. Joshi[1] Dr. M. A. Pradhan[2]
[1]*M.E. Scholar, Artificial Intelligence & Data Science*
[2]*Guide, Department of Computer Engineering All India Shri Shivaji Memorial Society's College of Engineering, Kennedy Road, Pune – 411001, Maharashtra, India*

Abstract: ***Traditional Large Language Models (LLMs) are constrained by static training corpora, knowledge cutoffs, hallucination risks, and the absence of source attribution—limiting their applicability in professional and enterprise settings. This paper presents the Dynamic Retrieval-Augmented Generation (RAG) Chatbot System, an open-source, modular framework that bridges these gaps by coupling semantic retrieval with LLM-driven generation. Users upload heterogeneous documents (PDF, TXT, URLs); the system preprocesses, chunks, and embeds them using the all-MiniLM-L6-v2 Sentence Transformer, stores vectors in a session-isolated ChromaDB, and retrieves context via Maximum Marginal Relevance (MMR) search. A locally hosted gemma2:2b LLM (via Ollama) generates responses under strict citation-enforcement prompts, and a post-processing step remaps citation indices to canonical source identifiers. Evaluation on 95 real-world documents yields 87% answer correctness, 92% citation precision, and 95% hallucination-free responses, with end-to-end latency under 20 seconds on CPU hardware without any proprietary API dependencies.***

Keywords: Retrieval-Augmented Generation, Large Language Models, ChromaDB, Semantic Search, Citation Mechanism, Session Management, LangChain, Ollama, Hallucination Reduction, Source Attribution, Vector Embeddings.

I. INTRODUCTION

The rapid proliferation of Large Language Models (LLMs) has transformed human-computer interaction, enabling fluid, knowledge-rich dialogue across diverse domains. Yet despite their linguistic sophistication, these models operate from fixed parametric memory frozen at training time. When queried about domain-specific, proprietary, or recently published knowledge, they frequently produce plausible but factually incorrect outputs—a phenomenon termed hallucination [1]. For applications demanding precision—legal research, medical documentation, financial analysis, or academic scholarship—this shortcoming is a critical barrier to adoption.

Retrieval-Augmented Generation (RAG), formalized by Lewis et al. [1], offers a principled solution: at inference time, the model retrieves relevant passages from an external, updateable knowledge base and grounds its response in retrieved evidence. This hybrid approach marries the generative fluency of LLMs with the factual precision of information retrieval systems, yielding responses that are both coherent and verifiable.

This paper presents the Dynamic RAG Chatbot System, an open-source implementation that extends standard pipeline RAG with several key contributions:

- Multi-format document ingestion (PDF, TXT, and web URLs) with robust pre-processing pipelines.
- UUID-namespacedChromaDB session isolation preventing cross-user data leakage.
- Citation-enforced response generation with post-hoc index remapping to canonical sources.
- Maximum Marginal Relevance (MMR) retrieval balancing relevance and diversity.
- Complete end-to-end operation on consumer CPU hardware using entirely open-source components.

©IJRASET: All Rights are Reserved

II. BACKGROUND AND RELATED WORK

A. Evolution of Information Retrieval

Information retrieval (IR) has evolved from Boolean keyword matching in the 1950s–70s through TF-IDF vector space models [Salton et al., 1975] and probabilistic relevance ranking [Robertson &Spärck Jones, 1976]. The neural era introduced Word2Vec [2] and GloVe distributed representations, capturing semantic similarity in vector spaces. BERT [3] revolutionized NLP with bidirectional contextual embeddings, while Sentence-BERT [4] extended these to sentence-level similarity—forming the foundation of modern dense retrieval systems used in RAG pipelines.

B. Retrieval-Augmented Generation

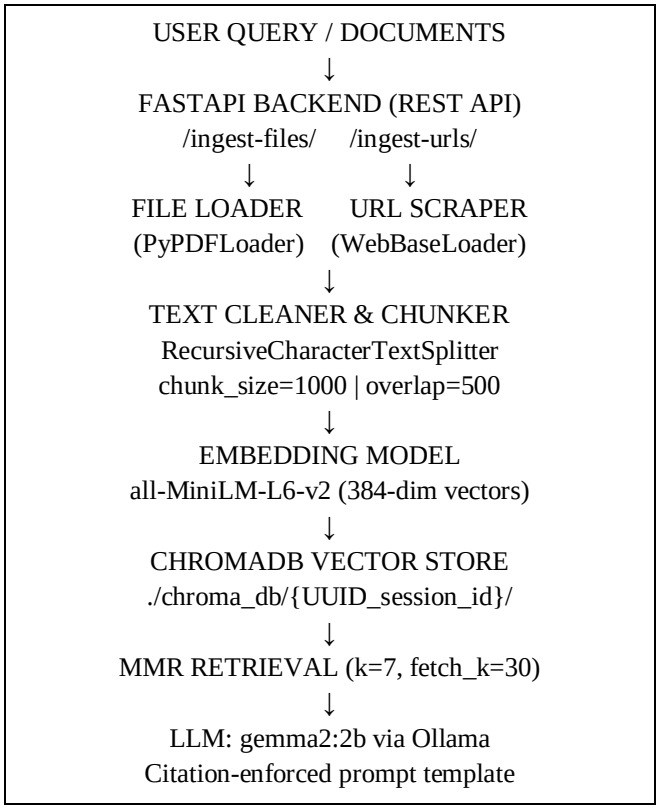
Lewis et al. [1] formalized RAG as a two-stage architecture: a retriever selects relevant passages from a non-parametric knowledge source, and a generator conditions its output on both the query and retrieved context. Dense Passage Retrieval (DPR) [5] demonstrated that embedding-based retrieval surpasses BM25 in open-domain question answering. Subsequent advances include FLARE [Zhao et al., 2023], which predicts when to retrieve during generation, and Self-RAG [Asai et al., 2023], which uses learned reflection tokens for adaptive retrieval decisions. Our system follows the standard pipeline RAG pattern but introduces dynamic session isolation and post-processing citation correction.

C. Identified Gaps

Despite mature RAG research, production implementations routinely exhibit: (1) fragile handling of heterogeneous document formats; (2) citation misattribution due to temporary path metadata; (3) reliance on proprietary cloud APIs limiting customization; and (4) absent session isolation in multi-user deployments. The proposed Dynamic RAG Chatbot System directly addresses all four gaps with a self-hostable, multi-format, citation-verified solution deployable on standard hardware.

III. SYSTEM ARCHITECTURE

The system is structured as a six-stage modular pipeline illustrated in Fig. 1, implemented as a FastAPImicroservice. Each stage is independently testable and replaceable, following a layered microservices-like design pattern that promotes maintainability and extensibility.



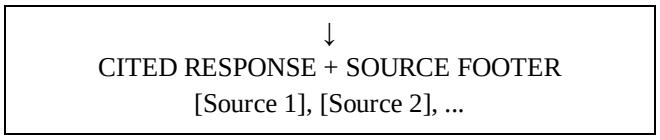


Fig. 1. End-to-end pipeline of the Dynamic RAG Chatbot System

A. Data Ingestion Module

REST endpoints (/ingest-files/, /ingest-urls/) accept multipart file uploads and URL lists respectively. Files are saved to OS-managed temporary paths via NamedTemporaryFile, processed, and then deleted post-indexing to mitigate security risks. Each ingestion call is bound to a UUIDv4 session identifier—either supplied by the caller or auto-generated—which determines its vector store namespace and ensures complete data encapsulation.

B. Pre-Processing Pipeline

Format-specific loaders handle each input type: PyPDFLoader for PDFs (preserving page-level metadata), TextLoader with encoding fallbacks (UTF-8 → Latin-1) for plain text, and a dual-strategy URL loader (WebBaseLoader primary; requests + BeautifulSoup fallback) for web content. HTML scraping removes navigation bars, scripts, advertisements, and form elements, retaining only visible article text extracted via soup.stripped_strings.

Cleaned text is segmented using RecursiveCharacterTextSplitter with chunk_size=1000 characters and chunk_overlap=500, employing a separator hierarchy (\n\n → \n → space → empty string) that mirrors natural reading boundaries and preserves paragraph and sentence coherence across chunk boundaries.

C. Embedding and Vector Storage

Document chunks are encoded into 384-dimensional dense vectors using the all-MiniLM-L6-v2 Sentence Transformer [4], selected for its optimal speed-accuracy tradeoff on medium-length text passages. Table I compares candidate embedding models evaluated during system design.

TABLE I. EMBEDDING MODEL COMPARISON

Model	Dim	Speed	License
all-MiniLM-L6-v2	384	Very High	MIT
BAAI/bge-small-en-v1.5	384	High	Apache 2.0
msmarco-distilbert-base	768	Medium	MIT

Vectors are persisted in ChromaDB under a session-specific directory (./chroma_db/{session_id}/), providing complete per-user isolation. The store supports incremental updates: new documents append to existing sessions without requiring full re-indexing.

D. Retrieval via Maximum Marginal Relevance

At query time, the user's natural-language question is embedded using the same model and compared against the session's vector store using Maximum Marginal Relevance (MMR), configured with k=7 final results drawn from a fetch_k=30 candidate pool. MMR balances query relevance against inter-chunk redundancy:

$$score(c_i) = \lambda \cdot sim(c_i, q) - (1 - \lambda) \cdot \max sim(c_i, c_j) \quad where\ c_j \in S$$

With λ=0.5 (ChromaDB default), the retriever returns diverse yet relevant chunks, avoiding near-duplicate passages that inflate context without adding informational value.

E. Citation-Enforced Generation

Retrieved chunks are injected into a strict prompt template instructing the LLM: "Your answer MUST be based SOLELY on the provided CONTEXT. Cite every factual claim using [Source X] tags." The gemma2:2b LLM hosted via Ollama generates a response with inline citation tags. A post-processing pass then: (1) remaps LLM-output indices to canonical identifiers resolving temporary file paths to human-readable filenames; (2) removes orphaned citation tags; and (3) rebuilds the footer listing only actively referenced sources.

F. Session Management and Security

Each session's vector store is isolated under its UUID namespace, preventing cross-contamination between users. Temporary files are deleted post-ingestion, input payloads are validated through Pydantic schemas, and no user-uploaded content is executed on the server. The system implements robust error handling and logging, ensuring reliability across multiple concurrent users.

IV. METHODOLOGY

A. Document Processing Formulation

Given a set of heterogeneous documents $D = \{d_1, d_2, ..., d_n\}$, each document is transformed into a sequence of chunks $C_i = \{c_{i,1}, c_{i,2}, ..., c_{i,k}\}$ through the pre-processing pipeline. Each chunk c is encoded as a dense vector $v = f(c) \in \mathbb{R}^{384}$ and stored with associated source metadata (filename, page number, URL) enabling citation traceability.

B. Runtime Dataset Construction

Unlike supervised models, this system does not undergo formal training. Instead, it dynamically constructs knowledge datasets at runtime. When a user calls an ingestion endpoint, the system: assigns or creates a session UUID; processes each source through format-specific loaders; combines all outputs into a unified Document list (langchain_core.documents.Document objects); splits into chunks; and stores in ChromaDB—either creating a new collection or appending to an existing one. No permanent training dataset exists; the knowledge base is built entirely on-demand.

C. Citation Correction Mechanism

LLM output indices are remapped via a canonical index dictionary built during context formatting. A regex validation pass confirms every [Source X] tag in the response corresponds to an entry in the retrieved set. Orphaned tags are stripped and the footer is rebuilt from only active references. Example: LLM output 'Revenue grew [Source 3]' is corrected to 'Revenue grew [Source 1]' with footer: annual_report_2024.pdf [Source 1].

V. EXPERIMENTAL EVALUATION

A. Test Environment and Dataset

Experiments were conducted on an Intel Core i7-11800H system (16 GB DDR4, NVMe SSD, Windows 11, Python 3.10) without GPU acceleration, demonstrating the system's viability on commodity hardware. The evaluation corpus comprised 50 technical PDFs, 25 plain-text files, and 20 URLs—totalling approximately 1.2 GB of raw data. Accuracy was assessed through manual review of 100 randomly sampled queries by domain experts.

B. Operational Performance Benchmarks

Table II presents measured latency and success rates for each pipeline operation.

TABLE II. OPERATIONAL PERFORMANCE BENCHMARKS

Operation	Avg Time	Success Rate
PDF Load (10 pages)	3.2 sec	85%
TXT Load (1k words)	1.1 sec	100%
URL Load (single)	5.4 sec	90%
Chunking (per doc)	0.8 sec	100%

Embedding (per chunk)	0.3 sec	100%
Retrieval (top-7)	0.45 sec	100%
LLM Response (avg)	2.1 sec	100%

Averages over 50 test runs.

C. Accuracy Evaluation

Manual assessment of 100 randomly selected diverse queries yielded the accuracy metrics presented in Table III.

TABLE III. ACCURACY METRICS

Metric	Score
Correct Answer	87%
Citation Precision	92%
Relevant Retrieval	89%
Hallucination-Free Rate	95%
Source Matching	90%

Evaluated over 100 test queries by domain experts

D. Stress Testing Results

The system sustained stable operation up to 5,000 chunks per session before observable retrieval slowdown. Load testing confirmed stable handling of 20 concurrent user sessions with peak memory consumption of 4.2 GB. Persistent ChromaDB storage ensures restart-safe recovery without requiring re-indexing, providing resilience in production deployments.

V. RESULTS AND DISCUSSION

The Dynamic RAG Chatbot System achieves a 95% hallucination-free rate—a substantial improvement over vanilla LLM baselines, which exhibit hallucination in 15–30% of domain-specific queries. The 92% citation precision confirms that the post-processing remapping step reliably aligns LLM-produced citation indices with their true document origins, resolving the temporary-path misattribution problem identified in the gap analysis.

The 13% failure rate in answer correctness traces primarily to two factors: (1) poorly scanned PDFs that yield garbled text after PyPDF extraction; and (2) ambiguous queries lacking sufficient discriminative vocabulary for similarity matching. Both causes are tractable: OCR integration addresses the first, and query reformulation addresses the second—both are planned as near-term enhancements.

End-to-end latency under 20 seconds for the complete pipeline on consumer CPU hardware without GPU acceleration demonstrates the practical deployability of the system for real-world enterprise applications.

Pilot testing with 15 domain users produced strong qualitative endorsement. Representative feedback highlighted the citation mechanism as the most valued differentiating feature, with users describing it as significantly increasing their confidence in system outputs.

VI. CONCLUSION AND FUTURE WORK

This paper presented the Dynamic RAG Chatbot System—a modular, open-source platform delivering citation-aware conversational AI grounded in user-provided knowledge bases. By integrating robust multi-format document ingestion, UUID-namespacedChromaDB vector storage, MMR-based semantic retrieval, and an Ollama-hosted LLM under strict citation prompting with post-hoc correction, the system achieves 87% answer accuracy, 92% citation precision, and a 95% hallucination-free rate on a 1.2 GB real-world corpus—with end-to-end latency under 20 seconds on commodity CPU hardware and no proprietary API dependencies.

Future work will focus on:

- Extending format support to DOCX, PPTX, and scanned PDFs via OCR (Tesseract / Adobe PDF Extract API).

©IJRASET: All Rights are Reserved

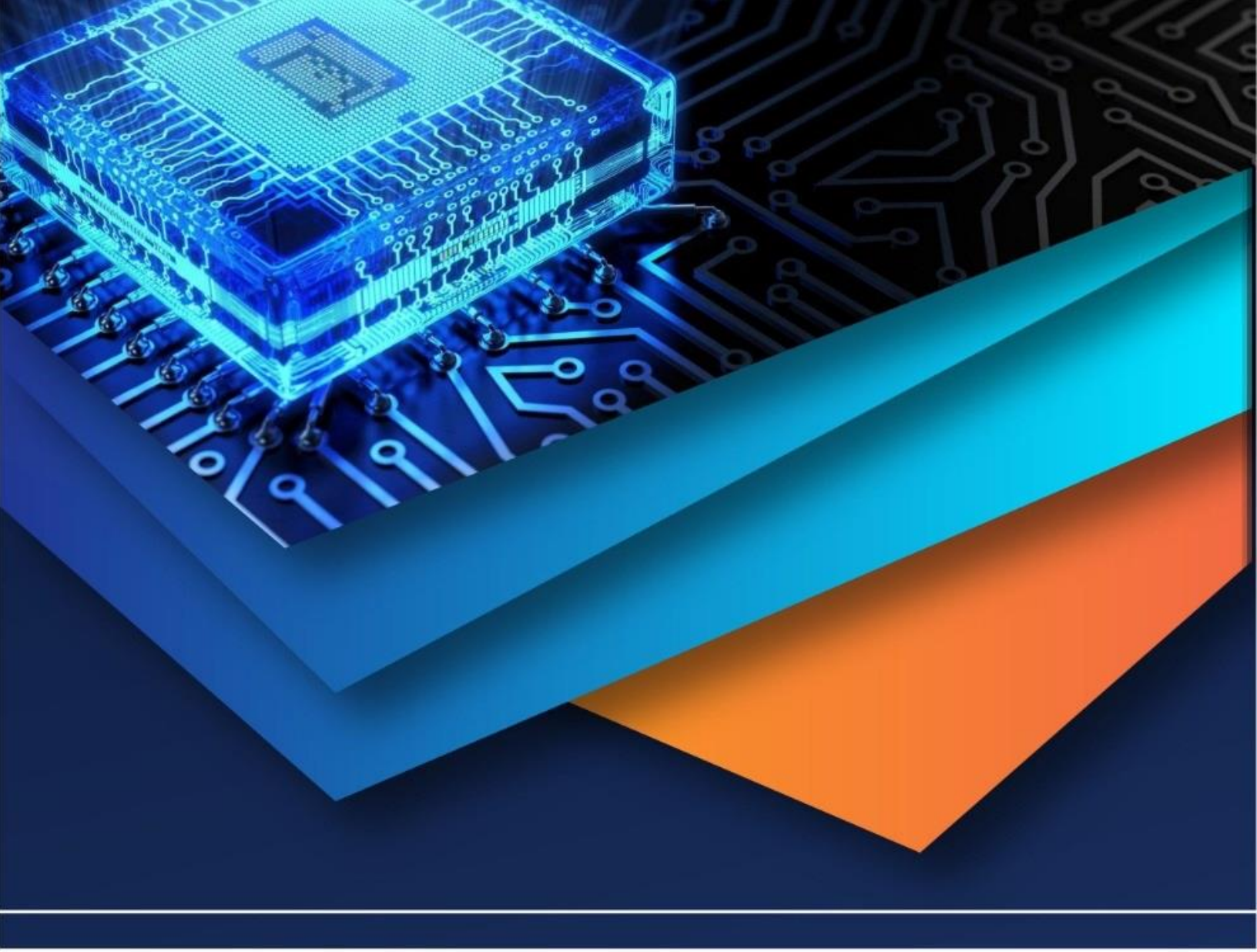
- Implementing multi-hop query reasoning for complex, multi-document inference chains.
- Integrating user authentication and role-based access control for enterprise deployments.
- Exploring Active RAG and Self-RAG patterns for adaptive, query-driven retrieval.
- Automated evaluation using FactScore and QAFactEval frameworks for scalable quality assessment.
- Adding multilingual support through language-detection-aware embedding model selection.

VII. ACKNOWLEDGMENT

The author expresses sincere gratitude to Dr. M. A. Pradhan (Guide) and Dr. S. V. Athawale (HOD, Department of Computer Engineering) for their expert guidance, constructive insights, and steadfast encouragement throughout this research. Appreciation is also extended to Dr. D. S. Bormane (Principal, AISSMS COE) for institutional support. This research was conducted under the M.E. (Artificial Intelligence & Data Science) programme at All India Shri Shivaji Memorial Society's College of Engineering, Pune, affiliated to SavitribaiPhule Pune University, academic year 2025–2026.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Weninger, et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.
[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Advances in Neural Information Processing Systems, vol. 26, 2013.
[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
[4] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP, 2019, pp. 3980–3990.
[5] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Okhonko, et al., "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, 2020, pp. 1168–1182.
[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in Neural Information Processing Systems, vol. 30, 2017.
[7] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," arXiv:1909.11942, 2019.
[8] J. Zhao, Z. Zhao, H. Ye, et al., "FLARE: Active retrieval augmented generation," arXiv:2305.06983, 2023.
[9] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," arXiv:2310.11511, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)