

Overcoming the Static Knowledge Bottleneck in AI Chatbots Using a Dual-Path Hybrid Architecture: A Review

Doke Sakshi Vitthal¹, Prof. Bangar Abhishek P.², Prof. Karad Vandana A.³

^{1,2,3}Department of Computer Engineering / Jaihind College Of Engineering / SPPU, Pune, India

Abstract- AI chatbots built on large language models (LLMs) answer from parametric knowledge that is fixed when training ends, so they cannot reliably reflect updated, institution-specific or access-restricted information unless the model is retrained or connected to external sources. This paper presents a structured narrative review of 17 sources published between 2019 and 2026, centred on five peer-reviewed IEEE Access studies from 2024–2026 and 12 foundational works, organised into seven themes: the static knowledge bottleneck and knowledge injection; retrieval foundations and pipeline design; fine-tuning, retrieval-augmented generation (RAG) and hybrid comparisons; retrieval-bypassing and question-indexed alternatives; corrective and adaptive retrieval; permission-aware retrieval; and local deployment, evaluation and reliability. Reported evidence shows that retrieval substantially improves on closed-book generation, that fine-tuning and RAG have complementary cost and quality profiles, that a combined approach attained the highest semantic scores in the one study that tested it, that similarity-based answer validation and question-indexed retrieval improve grounding on domain corpora, and that per-user access control can be enforced at retrieval time with modest overhead. However, none of the five core studies evaluates a mechanism that routes queries between a parametric path and a retrieval path, none measures answer quality after the underlying knowledge changes, and most validation is limited to one corpus or benchmark. The paper contributes a thematic synthesis, a comparison of twelve representative studies, a proposed conceptual seven-layer dual-path architecture and a prioritised research agenda.

Keywords— Static Knowledge Bottleneck; AI Chatbots; Retrieval-Augmented Generation; Fine-Tuning; Hybrid Architecture; Large Language Models; Access Control

I. INTRODUCTION

Chatbots built on large language models (LLMs) are now used for question answering in fields such as healthcare, law, finance and education [1], [2], but the knowledge they draw on is largely frozen at training time. Facts that appear rarely in pretraining are less likely to be learned reliably, and changing a specific fact usually requires further training or model-editing methods, which can be costly and may introduce unwanted side effects [3], [4]. This static knowledge bottleneck makes a chatbot unreliable for content that changes, belongs to one institution, or is visible only to some users.

Two strategies are commonly used to adapt LLMs to such content: fine-tuning on domain data and retrieval-augmented generation (RAG) [5], [6]. Fine-tuning is computationally demanding, costly and

risks forgetting previously learned information [6], whereas RAG retrieves relevant documents at inference time and conditions the generator on them [7]. RAG, however, introduces its own bottlenecks: answer quality depends heavily on the retriever and on the precision of the retrieved context [2], the most accurate retriever may be far slower than a compact one [4], retrieved content may be

exposed to users who should not see it [1], and local deployments can fail to produce an answer at all [4]. A dual-path hybrid architecture treats the two mechanisms as parallel paths: a parametric path, in which a domain-adapted model supplies stable and fast knowledge, and a retrieval path, which supplies fresh, permissioned and verifiable evidence, with a coordinating layer that decides how the two are used. The term is used in this review as a design concept; the reviewed studies evaluate components of such a design rather than the complete system.

This review aims to (i) synthesise approaches that address the static knowledge bottleneck in chatbots, (ii) compare their methods, reported performance and limitations, (iii) propose a literature-grounded conceptual dual-path architecture, and (iv) identify research gaps that must be closed before deployment.

II. REVIEW METHODOLOGY

The review is centred on five peer-reviewed IEEE Access articles published between 2024 and 2026 that were selected by the authors as core evidence on retrieval-augmented, fine-tuned and hybrid chatbot knowledge systems. Each core article was read in full, and its objective, method, dataset, metrics, results and stated limitations were extracted. Twelve further sources, comprising foundational retrieval and RAG papers, evaluation methods and related work, were drawn from the reference lists of the core articles; their bibliographic details are taken from those lists, and the findings attributed to them are those reported in the core articles unless stated otherwise. Publications from 2019 onwards were retained where they established a method or metric still in use.

Numerical results are reproduced as reported by the original authors. They are not directly comparable across studies because tasks, datasets, models and evaluation protocols differ. Because no database searches with logged queries were performed, the review is structured and narrative rather than systematic, and its coverage is limited to the selected core studies and the works they cite.

III. LITERATURE REVIEW

1. The Static Knowledge Bottleneck and Knowledge Injection

Pretrained LLMs store factual knowledge in distributed parameters, but this knowledge is uneven: long-tail facts are learned less reliably, and accuracy tends to scale with the amount of relevant evidence seen during pretraining [3], as summarised in [4]. Because the knowledge is not stored explicitly, it is difficult to inspect or justify where an answer came from, and knowledge-intensive prompting can

still yield confident hallucinations [4]. Lewis et al. proposed RAG, in which a retriever fetches passages at inference time and the generator is conditioned on them [7]; because the corpus can change without retraining the generator, retrieval addresses staleness directly. Ovadia et al. compared knowledge injection by fine-tuning and by retrieval; as summarised by Budakoglu and Emekci, RAG was stronger at injecting real-time knowledge, particularly where the knowledge base is updated frequently, while fine-tuning was much weaker at encoding new information [8], [5]. The wider RAG landscape is surveyed by Gao et al. [9].

2. Retrieval Foundations and Pipeline Design

A typical RAG pipeline chunks a corpus into passages, embeds them in a dense vector space, indexes them for approximate nearest-neighbour search and inserts the top-k passages into the generator prompt [4]. Dense Passage Retrieval, a dual-encoder approach, established dense retrieval as a strong practical method for open-domain question answering [10], and KILT reframed evaluation around a shared Wikipedia snapshot with explicit provenance, so that answer correctness can be separated from evidence-grounded correctness [11].

Retriever choice matters in practice. In a reduced-corpus KILT-NQ study, Qwen3-Embedding-4B reached 96.8 % Recall@5 and 98.0 % Recall@10 but needed about 2.28 s per query, whereas all-MiniLM-L6-v2 reached 94.4 % and 96.6 % in about 0.015 s, roughly 150 times faster; nomic-embed-text-v1.5 reached only 60.0 % and 65.2 % [4]. The strongest and the fastest retriever were therefore different systems, and the authors caution that a corpus containing all gold passages plus 100,000 distractors makes retrieval easier than at full scale [4]. Query rewriting is a further lever: the Rewrite–Retrieve–Read framework trains a rewriter with reinforcement learning to produce retrieval-oriented queries and reported an Exact Match of 45.72 on PopQA [12], as reported in [2]. By contrast, same-model query rewriting with the tested prompt did not outperform standard RAG for any of 12 local generators and added latency and failures [4].

3. Fine-Tuning, RAG and Hybrid Comparisons

Budakoglu and Emekci compared fine-tuning, RAG and a combination of both on SQuAD, MS MARCO and SQL CREATE TABLE statements, using a 12,500-triplet subset with 10,000 pairs for training or the vector database and 2,500 for testing [5]. The retrieval base was Chroma DB, answers were generated with GPT-3.5 Turbo, and fine-tuning was applied to the BAAI/bge-small-en embedding model [5]. On SQuAD and MS MARCO, fine-tuning obtained higher ROUGE than RAG (0.25 versus 0.17 and 0.20 versus 0.14), while BLEU was low for all methods. The hybrid gave the highest cosine similarity on every dataset (0.54 on SQuAD, 0.58 on MS MARCO and 0.89 on SQL statements, against 0.47–0.49, 0.55–0.56 and 0.83–0.87 for the single methods) and the highest BERTScore recall (0.90, 0.86 and 0.94). On SQuAD, answer correctness was 0.82 for the hybrid against 0.74 for fine-tuning and 0.50 for RAG, and faithfulness was 0.89, 0.88 and 0.82 respectively [5].

Qualitative error analysis showed that the RAG model sometimes replied that it did not have the information, or gave a wrong answer although the context was clear, whereas fine-tuned and hybrid models avoided such replies but produced longer answers that risk hallucination. In cost terms, RAG needed minimal initial CPU and memory, low-cost related documents and no training, but incurred a substantial vector-database cost; fine-tuning needed high-cost labelled data and training but no vector database, and the hybrid bore the combined costs. Two cautions apply. The abstract states that the hybrid may not surpass fine-tuning, and lexical results were mixed (on SQuAD, METEOR fell from 0.44 for fine-tuning to 0.16 for the hybrid), so the benefit of combination is clearest on semantic and context-based metrics. In addition, fine-tuning is described mainly at the embedding-model level rather than as adaptation of the answer-generating LLM.

4. Retrieval-Bypassing and Question-Indexed Alternatives

Collini et al. proposed Context-Aware RAG (CA-RAG), which removes the retrieval filter: documents are split into semantic chunks (a new chunk starts

when the cosine similarity between consecutive sentence embeddings falls below $\theta = 0.3$), every chunk is given to the generator (Zephyr-7B-Beta) together with the question, and each resulting answer is filtered by its similarity to its chunk using a threshold $\tau = 0.6$, with 'no context provided' returned for unsupported cases [2]. On benchmark datasets (TriviaQA, Natural Questions, SQuAD and AmbigQA; about 10,000 question-answer pairs), the best CA-RAG configurations reached BERTScore F1 of 0.8629 and 0.9089 on TriviaQA, above the RAG and RAG-Rerank baselines, but a reranked RAG baseline was strongest on the shorter-context Natural Questions and SQuAD [2]. On a custom set of full scientific papers of roughly 6,700–11,200 words, CA-RAG with post-processing reached F1 of 0.93–0.99 under semantic chunking, against about 0.78 for the best RAG variant and for Llama-3.3-70B given the whole document [2]. The cost was latency: about 262 s per question with semantic chunking and about 2,289 s with fixed-size chunking, against about 86 s for RAG-Rerank. The authors argue this is acceptable where accuracy matters more than speed and that the method suits moderate-size corpora; the thresholds were chosen empirically, and the custom evaluation was manual with a small question set.

Saha et al. keep retrieval but change what is indexed. An LLM generates questions for each chunk, the questions are embedded and quantized to prototype vectors in an inverted index, and a user query is matched against stored questions (top three) to locate the chunk that answers it [6]. Using 687 scraped pages from two university websites (22,609 generated questions, reviewed manually by two researchers) and Llama-3-8B-instruct as the generator, QuIM-RAG with the custom corpus reached faithfulness 1.00, answer relevancy 0.99, context precision 0.92, context recall 0.74 and BERTScore F1 0.67, against 0.72, 0.82, 0.69, 0.45 and 0.36 for traditional RAG on the same corpus [6]. The system is prompted to state when a query falls outside its knowledge, and the authors plan to refresh the corpus every four months because university content changes each semester [6].

5. Corrective and Adaptive Retrieval

Several approaches add a decision layer around retrieval. Corrective RAG (CRAG) scores retrieved documents with a T5-large evaluator, labels them correct, incorrect or ambiguous, refines correct documents into knowledge strips and falls back to web search for incorrect cases; its drawbacks are dependence on the evaluator and possible bias in web search [13], as described in [2]. Self-CRAG, which adds the ideas of Self-RAG [14], reached an accuracy of 69.1 on the Biography dataset [2]. REAR, as described by Collini et al., scores each query-document pair with a lightweight LLM and lets the LLM rely on its internal knowledge when the score is low [2]; this is the closest precedent among the reviewed sources to selecting between a parametric and a retrieval path. The similarity filter in CA-RAG is a lighter-weight relative of the same idea. Extra stages carry a cost, however: in the local study, the rewriting stage contributed 82 of the 242 recorded failures [4].

6. Permission-Aware Retrieval

Most RAG research assumes that retrieved content is equally accessible to every user [1]. Jeong and Lee argue that this conflicts with least-privilege principles when data sit in independently governed sources, and that merging RBAC and ABAC policies into one layer is error-prone [1]. Their Permission-Aware RAG keeps a central vector store but checks each retrieved document at query time against the owning provider's native IAM (GCP IAM, AWS IAM or Keycloak) through a credential manager and a multithreaded IAM adapter [1]. On HotpotQA (13,783 documents and 7,405 questions, with 1,000 questions in the main test) all eight user profiles achieved a perfect policy-match result, while a control group using a manually merged policy was granted a document that the correctly configured user was denied, because one attribute was omitted in the merge [1].

Answer quality followed permission coverage: a user with 0 % coverage scored EM 0.24 and F1 0.31, a user with 69.84 % coverage scored EM 0.38 and F1 0.50, and the administrator scored EM 0.47 and F1 0.61. The authors suggest that the non-trivial score at zero coverage reflects the model's internal knowledge or

correct guesses on binary questions. With caching disabled, IAM latency grew by about 140 ms as top-k rose from 1 to 20, and mean IAM latency fell by about 70 % (420.71 ms to 127.33 ms) as workers rose from 1 to 15, while end-to-end time was dominated by the LLM. Acknowledged limitations include dependence on IAM availability, static credentials in the experiments and a benchmark that does not capture enterprise scale.

7. Local Deployment, Evaluation and Reliability

Oliveira and Cerqueira evaluated 12 quantized local generators, three embedding models and four pipeline conditions (retrieval only, closed-book, standard RAG and same-model rewrite-RAG) on a 500-question reduced KILT-NQ setup using a single 24 GB GPU [4]. RAG improved every generator over closed-book: Qwen3:0.6B rose from 7.5 % to 22.1 % F1, and the best result, Qwen3:30B, reached 58.4 % F1 and 46.0 % EM. Performance was not monotonic in model size, since Qwen3:4B (51.0 % F1) outperformed Qwen3:8B and Qwen3:14B, and semantic answer similarity exposed outliers such as LFM2.5:1.2B, which reached 66.1 % SAS at 17.1 % F1. Failures were counted rather than discarded: 242 failed trials, of which 51 occurred in closed-book generation, 51 in standard RAG generation, and 82 in rewriting and 58 in answer generation under rewrite-RAG, most often because no first token was emitted before the watchdog threshold [4]. Mean time per question for the best standard-RAG settings ranged from 2.74 s to 18.15 s.

Evaluation practice is also uneven. RAGAS offers metrics such as faithfulness, answer relevance and context relevance and is designed to avoid human annotation [15], and BERTScore compares contextual embeddings of generated and reference answers [16]; the core studies use them in different combinations with EM, F1, ROUGE, BLEU and METEOR. Barnett et al. catalogue seven failure points in engineering RAG systems [17], and Oliveira and Cerqueira note that reliability problems appear in real operation and change as the design changes [4].

8. Synthesis

Three patterns emerge. First, the reviewed work attacks the static knowledge bottleneck from

separate angles: fine-tuning for domain adaptation, retrieval for freshness, chunk-level generation and question indexing for grounding, and IAM delegation for governance. Second, validation is narrow: one set of university websites, one multi-hop benchmark, a reduced corpus, or a small manually judged question set, and in at least one core study the dataset was deliberately static; none of the five core studies measures what happens to answer quality after the knowledge source changes. Third, metrics, cost and failure reporting are inconsistent, so that lexical, semantic, reference-free and

operational results cannot be combined into a single ranking.

IV. COMPARATIVE ANALYSIS

Table 1 compares twelve representative studies in chronological order. Reported metrics are not directly comparable because tasks, datasets, class definitions and evaluation protocols differ.

Table 1. Comparison of representative studies on knowledge-augmented chatbot architectures.

Author, year	Objective	Method / technology	Major reported findings	Limitations
Lewis et al., 2020 [7]	Knowledge-intensive NLP with external knowledge	Retrieve-then-generate; generator conditioned on retrieved passages	Retrieval treated as part of the model's access to knowledge	Answer quality depends on the retriever
Karpukhin et al., 2020 [10]	Open-domain QA retrieval	Dual-encoder dense passage retrieval	Established as a strong practical method	Embedder choice affects recall and cost
Petroni et al., 2021 [11]	Provenance-aware evaluation	Shared Wikipedia snapshot; KILT tasks	Separates correctness from grounded correctness	Benchmark, not a system
Kandpal et al., 2023 [3]	Long-tail knowledge in LLMs	Analysis of pretraining evidence vs. accuracy	Accuracy tends to scale with supporting evidence seen in pretraining	Reported here as summarised in a core study
Ovadia et al., 2023 [8]	Compare knowledge injection	Fine-tuning vs. retrieval	RAG stronger for real-time knowledge	Reported here as summarised in a core study
Ma et al., 2023 [12]	Better retrieval queries	Rewrite–Retrieve–Read; T5-large rewriter; RL	Highest EM 45.72 (PopQA)	Generalisation vs. specialisation
Yan et al., 2024 [13]	Robust retrieval	T5-large evaluator; knowledge strips; web search	Self-CRAG accuracy 69.1 (Biography)	Evaluator dependence; web bias
Saha et al., 2024 [6]	Domain QA; less dilution	Question-to-question inverted index; Llama-3-8B-instruct	Faithfulness 1.00; BERTScore F1 0.67 (custom corpus)	One university site; no user study
Budakoglu & Emekci, 2025 [5]	Compare fine-tuning, RAG, hybrid	Fine-tuned embedder; Chroma DB; GPT-3.5 Turbo	Hybrid highest cosine similarity (0.54 / 0.58 / 0.89)	12,500-triplet subset; mixed lexical scores
Collini et al., 2025 [2]	Context-aware answering without retrieval	Semantic chunking; all chunks to Zephyr-7B; similarity filter	F1 0.93–0.99 vs. about 0.78 for RAG variants (custom papers)	About 262 s per question; manual

				small-set evaluation
Jeong & Lee, 2025 [1]	Per-user access control in RAG	IAM-delegated checks (GCP, AWS, Keycloak)	Perfect match for 8 users; merged-policy control leaked	HotpotQA only; IAM availability
Oliveira & Cerqueira, 2026 [4]	Local quantized RAG evaluation	12 generators, 3 embedders, 4 pipelines; one GPU	RAG > closed-book; best F1 58.4 % (Qwen3:30B)	Reduced corpus; single seed; one stack

**V. PROPOSED CONCEPTUAL
ARCHITECTURE**

Fig. 1 presents a proposed dual-path architecture derived from the reviewed literature; it has not been implemented or validated. Layer 1 is the user and query environment, including the user’s identity and access scope [1]. Layer 2 analyses the query (intent and domain, need for fresh information, confidence) and decides whether to use the parametric path, the retrieval path or both; this routing step is a design proposal, with only partial precedents in relevance-aware and corrective retrieval [2], [13].

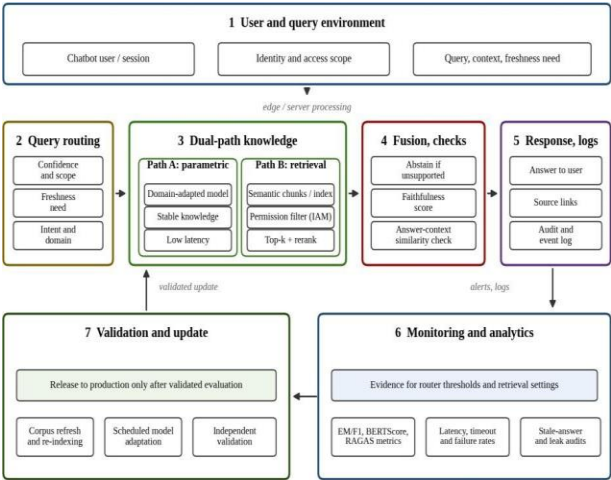


Fig. 1. Proposed conceptual dual-path architecture for a chatbot that overcomes the static knowledge bottleneck (conceptual, not experimentally validated).

Layer 3 holds the two paths. Path A is a domain-adapted model that supplies stable knowledge at low cost and latency, chosen on the quality–latency trade-off rather than on parameter count [5], [4]. Path B retrieves current evidence using semantic

chunking [2] and question-indexed matching [6], with an embedder chosen on recall and latency together [4] and with each candidate document checked against the owning system’s native IAM [1]. Because parametric weights cannot be filtered per user, this review suggests, as a design consideration rather than a reported finding, that Path A should be adapted only on content cleared for all users and that restricted content should be reachable only through Path B.

Layer 4 verifies the candidate answers by checking their similarity to the supporting context [2], scoring faithfulness [15] and abstaining when evidence is insufficient [2], [6]. When Path B returns no authorised context, the system should state that fact or answer from Path A only for non-restricted knowledge, since a user with no accessible documents still obtained a non-trivial score from the model’s internal knowledge [1]. Layer 5 returns the answer with source links [6] and records an audit log. Layers 2–5 form the request path and should run where protection and answering continue to work if connectivity is lost.

Layer 6 monitors quality (EM, F1, BERTScore and RAGAS-style metrics), latency, timeouts and failure rates [4], [1], and supplies evidence for router thresholds and retrieval settings.

Layer 7 refreshes and re-indexes the corpus on a schedule, as planned in [6], adapts the parametric model periodically, and releases updates only after independent validation. Cached permission results need a time-to-live and forced invalidation after permission changes [1].

VI. RESEARCH GAPS AND FUTURE DIRECTIONS

Table 2 summarises the gaps supported by the reviewed evidence and the corresponding research directions.

Table 2. Research gaps and future directions.

Gap supported by evidence	Supporting studies	Future direction
Answer quality after the knowledge source changes is not measured	[5], [6], [1]	Longitudinal tests on a changing corpus; measure update-to-answer delay and stale-answer rate
No evaluated routing between a parametric path and a retrieval path	[5], [2], [4]	Build and test routers using freshness, confidence and access signals
Validation on one corpus, benchmark or reduced corpus	[6], [1], [4]	Multi-domain and enterprise trials with full-scale corpora
Lexical, semantic and grounding metrics reported inconsistently	[4], [5]	Joint reporting of EM/F1, semantic similarity, grounded and faithfulness metrics
Cost, latency and failure reported unevenly	[5], [2], [4], [1]	Report latency distributions, failure rates and resource cost beside quality
Access control studied apart from answer-quality design	[1]	Design and test fallback behaviour when authorised context is empty, without leakage through trained weights
Abstention and filtering thresholds set empirically	[2]	Calibrate and validate thresholds across domains and corpus sizes

Directions that still require investigation include human-factors studies of how users respond to abstentions and source links, scalable indexing for

large corpora when every chunk is processed by an LLM, and cost-benefit analysis of the parametric and retrieval paths in low-resource settings.

VII. CONCLUSION

Research on overcoming the static knowledge bottleneck has progressed from comparing fine-tuning with retrieval towards hybrid designs, similarity-validated and question-indexed retrieval, permission-aware retrieval and resource-constrained local evaluation. Retrieval clearly improves over closed-book generation, and combining adaptation with retrieval gave the best semantic results in the one study that tested it. However, evidence is dominated by single-corpus and static-knowledge experiments, query routing between a parametric and a retrieval path has not been evaluated, and cost, latency and failure are reported unevenly. Reliable deployment will require freshness-aware evaluation, transparent quality and operational metrics, access control that covers both paths, and independently validated updates, as reflected in the proposed architecture.

REFERENCES

1. Jeong, J., Lee, S.-G. "Permission-aware RAG: Identity and access management (IAM)-based access filtering in multi-resource environments". IEEE Access 13, 192819–192835 (2025). <https://doi.org/10.1109/ACCESS.2025.3628960>
2. Collini, E., Kurniadi, F.I., Nesi, P., Pantaleo, G. "Context-aware retrieval augmented generation using similarity validation to handle context inconsistencies in large language models". IEEE Access 13, 170065–170080 (2025). <https://doi.org/10.1109/ACCESS.2025.3614553>
3. Kandpal, N., Deng, H., Roberts, A., Wallace, E., Raffel, C. "Large language models struggle to learn long-tail knowledge". In: Proc. Int. Conf. Mach. Learn., pp. 15696–15707 (2023).
4. Oliveira, M.L.L. de, Cerqueira, J.D.J.F. "Resource-constrained evaluation of quantized local LLMs for retrieval-augmented generation on a reduced-corpus setting". IEEE Access 14, 70402–

- 70422 (2026).
<https://doi.org/10.1109/ACCESS.2026.3691267>
5. Budakoglu, G., Emekci, H. "Unveiling the power of large language models: A comparative study of retrieval-augmented generation, fine-tuning, and their synergistic fusion for enhanced performance". IEEE Access 13, 30936–30951 (2025).
<https://doi.org/10.1109/ACCESS.2025.3542334>
6. Saha, B., Saha, U., Malik, M.Z. "QuIM-RAG: Advancing retrieval-augmented generation with inverted question matching for enhanced QA performance". IEEE Access 12, 185401–185410 (2024).
<https://doi.org/10.1109/ACCESS.2024.3513155>
7. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., Kiela, D. "Retrieval-augmented generation for knowledge-intensive NLP tasks". In: Proc. Adv. Neural Inf. Process. Syst. 33, pp. 9459–9474 (2020).
8. Ovadia, O., Brief, M., Mishaali, M., Elisha, O. "Fine-tuning or retrieval? Comparing knowledge injection in LLMs". arXiv:2312.05934 (2023).
9. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H. "Retrieval-augmented generation for large language models: A survey". arXiv:2312.10997 (2023).
10. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.-T. "Dense passage retrieval for open-domain question answering". In: Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), pp. 6769–6781 (2020).
11. Petroni, F., Piktus, A., Fan, A., Lewis, P., Yazdani, M., De Cao, N., Thorne, J., Jernite, Y., Karpukhin, V., Maillard, J., Plachouras, V., Rocktäschel, T., Riedel, S. "KILT: A benchmark for knowledge intensive language tasks". In: Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol., pp. 2523–2544 (2021).
12. Ma, X., Gong, Y., He, P., Zhao, H., Duan, N. "Query rewriting in retrieval-augmented large language models". In: Proc. Conf. Empirical Methods Natural Lang. Process., pp. 5303–5315 (2023).
13. Yan, S.-Q., Gu, J.-C., Zhu, Y., Ling, Z.-H. "Corrective retrieval augmented generation". arXiv:2401.15884 (2024).
14. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H. "Self-RAG: Learning to retrieve, generate, and critique through self-reflection". arXiv:2310.11511 (2023).
15. Es, S., James, J., Espinosa-Anke, L., Schockaert, S. "RAGAS: Automated evaluation of retrieval augmented generation". arXiv:2309.15217 (2023).
16. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y. "BERTScore: Evaluating text generation with BERT". arXiv:1904.09675 (2019).
17. Barnett, S., Kurniawan, S., Thudumu, S., Brannelly, Z., Abdelrazek, M. "Seven failure points when engineering a retrieval augmented generation system". In: Proc. IEEE/ACM 3rd Int. Conf. AI Eng. – Softw. Eng. AI, pp. 194–199 (2024).