Mathematical Modeling and Algorithm Application
ISSN: 3006-0842 | Vol. 10, No. 1, 2026

A Review of Retrieval, Context Compression, and Evidence Grounding Strategies for Efficient Locally Deployed PDF-Based RAG Systems

Shaoyuan Wu*

Department of Computing and Software, McMaster University, Hamilton, Ontario, L8S 4L7, Canada

* **Corresponding author**: Shaoyuan Wu (Email: mikewu2004@163.com)

Abstract: Retrieval-Augmented Generation (RAG) is widely used to connect large language models (LLMs) with external documents. RAG enables language models to retrieve evidence from external documents based on the content of those files. In local deployment, documents such as textbooks, research papers and technical reports in PDF format can be processed without relying on external services, which may protect users' privacy. However, RAG systems based on local PDFs still face many challenges, including difficulties in efficiently and accurately extracting metadata from PDFs, noisy text produced by PDF parsing or OCR, inference efficiency, incomplete evidence selection, and the potential for "hallucinations" in generated answers. This paper looks at three core strategies for boosting local PDF-RAG performance: better retrieval, smarter context compression, and more rigorous evidence grounding. On the retrieval side, the goal is to pull out relevant paragraphs and refine how passages are ranked across large document sets. Context compression, meanwhile, steps in before generation to cut through redundant content, which helps keep inference costs in check. Evidence grounding takes a different angle—it checks whether the answers actually align with the retrieved material and, if not, defaults to a "no answer" response when the evidence falls short. After walking through and comparing these approaches, the paper argues that a well-functioning local RAG system isn't about optimizing any single piece in isolation. It needs to juggle retrieval precision, computational expense, citation reliability, and response faithfulness all at once—and that balancing act is where the real challenge lies.

Keywords: Retrieval-Augmented Generation (RAG); Locally Deployed RAG; Context compression; Large Language Model (LLM); Evidence Grounding

1. Introduction

Retrieval-Augmented Generation, or RAG, hooks a large language model up to outside evidence during inference. In systems built around PDFs, that evidence tends to come from all sorts of places—research papers, technical reports, user manuals, textbooks, even scanned archival material. Running everything locally means the whole workflow—parsing, indexing, retrieval, generation—stays inside a controlled setting. That matters for two reasons: it keeps sensitive documents from being pushed to some external server where they might leak, and it leaves users in charge of updates and model choices.

The literature tends to treat RAG as an end-to-end pipeline, where the final output hinges on how you handle data prep, retrieval, augmentation, and generation [1]. LongLLMLingua, for instance, shows that if you make compression aware of the query, you can cut down on long-context costs without losing too much useful signal [2]. Fan et al. make a similar point—they argue that retrieval-augmented LLMs need coordinated design, not just piecemeal tuning [3]. Taken together, these works underscore why you have to look at the whole chain.

Most of the work out there tends to zoom in on a single piece of the puzzle. What's rarer is a side-by-side comparison of different optimization strategies, especially in local deployments where PDFs dominate. A few comparative studies do hint that decisions around chunking, retrieval settings, or ranking can push both accuracy and response times in one direction or the other [5]. Compression looks appealing for efficiency, no doubt—but you're always running the risk of dropping something that actually matters. Grounding might boost trustworthiness, but that only works if the retrieval and evidence selection aren't shaky from the get-go.

So this review steps in at three different junctures—retrieval optimization, context compression, and evidence grounding—and compares them across four axes: how accurate the retrieval is, how efficient the inference runs, how well citations hold up, and whether the answers are actually faithful to the source. The aim is to offer a structured way of thinking about designing local PDF-based RAG systems that are both efficient and reliable.

2. Background and scope

2.1. Definition of PDF-Based RAG Systems

Unlike plain text, PDFs aren't just about words on a page—they often pack in charts, equations, and images that can themselves carry textual cues. Parsing tools might grab the obvious text, but they frequently drop things like the intended reading flow, how tables are laid out, or even the captions sitting right under figures. So PDF-RAG systems really ought to do more than just pull strings of text; they need to hang onto source-level markers—page numbers, section headings, bounding boxes, that sort of thing [4]. With that metadata in place, retrieved chunks and final citations can be pinned back to exact spots in the original document, which makes it a whole lot easier for someone to actually go check the source material themselves.

2.2. Scope of Locally Deployed RAG Systems

As we use the term here, local deployment means the main

pipeline—PDF parsing, index building, retrieval, and answer generation—all runs on the user's own machine. We're talking about ordinary consumer-grade computers, not big GPU clusters. Keeping everything local cuts reliance on outside servers, avoids shipping documents off to third-party services, and gives users a tighter grip on their files, indexes, and model choices. That said, local setups are bound by whatever storage, RAM, and compute power the hardware happens to have. A retrieval approach might look great on paper in terms of accuracy, but if it chews through too much memory or drags response times way up, it's probably not going to fly in practice [5].

2.3. Basic Retrieval Pipeline

Once a PDF is fed into the system, it usually goes through parsing, chunking, index building, and retrieval before any relevant bits actually get handed over to the LLM for generating an answer. These stages don't operate in isolation—they're strung together, which means a slip-up early on can ripple through and mess with whatever comes later. Take a multi-column page: if the parser gets the reading order wrong, two unrelated paragraphs might end up welded into the same chunk. Then even if the retriever latches onto the right keywords, the evidence it pulls back could be incomplete or disjointed. To head that off, it's better to attach page numbers, section markers, and source IDs to chunks right when they're being created during chunking and indexing, rather than tacking them on after the answer's already been produced [4].

2.4. Review Scope and Comparison Criteria

We selected 20 representative studies from 2022 to 2025 that touch on retrieval optimization, context compression, evidence grounding, PDF processing, and citation evaluation within PDF-RAG systems. The trouble is, these papers vary wildly in terms of datasets, model architectures, system configurations, and hardware setups, so stacking their reported numbers side-by-side for any kind of direct comparison or overall ranking would be pretty meaningless. Rather than cramming everything into one numeric score, this review takes a step back and asks what each method is really trying to accomplish—and what added computational, storage, or evidence-processing burdens might come with running it locally. We're evaluating them along four lines: retrieval precision, inference speed, citation soundness, and how faithfully the answers reflect the source material.

3. Taxonomy of Optimization Strategies

3.1. Retrieval Optimization

Retrieval quality depends on quite a few things—parsing choices, whether metadata survives the process, the first-stage retrieval step, and then reranking after that. Shorter chunks might give you tighter matching precision, though they tend to shed useful context; longer ones hold onto more of the overall picture but bring in additional noise as a trade-off. With PDFs in particular, chunking should really hang onto page numbers and section headings, so that any evidence retrieved can still be pinned down to a specific spot.

ColBERTv2 adopts a token-level late-interaction mechanism and throws in residual compression to improve ranking while cutting down on the storage demands that earlier late-interaction systems struggled with [6]. DSE, meanwhile, sidesteps all that by grabbing page screenshots directly, preserving layout, text, and visual elements that OCR-heavy pipelines tend to lose [7]. The trade-off? ColBERTv2 relies on a multi-vector index, whereas DSE requires image storage plus a vision-language encoder. And because they've been assessed on entirely different benchmarks, neither one can really claim a decisive edge for local deployment [9].

3.2. Context Compression

Context compression is all about trimming down what actually gets fed into the generator. LongLLMLingua uses query-aware compression and document reordering to surface question-relevant information [2]. RECOMP, by contrast, decides whether to keep retrieved content at all and can even suppress augmentation when it's unlikely to add value [10]. Evidence-oriented extraction goes a step further with a more targeted kind of reduction. SEER learns to pull concise evidence from retrieved passages while juggling faithfulness, helpfulness, and brevity [11]. EXIT employs context-aware extractive compression to shorten input length and cut inference time, all while preserving connections between neighboring sentences [12]. Compared to abstractive summaries, extractive approaches keep a more direct tie to the original PDF—though they do add an extra extraction step to the pipeline. In local setups, compressed context can feed into generation, with the original passages and source markers still on hand for verification purposes.

3.3. Evidence Grounding

Evidence grounding works by tying generated claims back to retrieved PDF evidence and then checking whether that evidence actually holds up. ALCE breaks answer quality apart from citation correctness and citation completeness, making the point that even plausible answers can lean on unsupported or poorly cited claims [13]. MiniCheck, meanwhile, shows that smaller models can assess claim-document consistency at a fraction of the cost of big proprietary evaluators [14].

More recent approaches have pushed attribution to a finer level of detail. VISA pinpoints supporting regions in document screenshots via bounding boxes [15]. AGREE adapts language models to strengthen grounding and citation generation [16], while ReClaim weaves references and claims together to offer sentence-level attribution in longer responses [17]. These make verification easier, though visual attribution brings in multimodal inference costs and model adaptation may add training overhead.

Grounding can also be looked at from a system-wide perspective. RAGAs offer reference-free metrics covering context quality, answer relevance, and faithfulness [18]. RAGTruth contributes a human-annotated corpus for studying hallucinations in retrieval-augmented output [19], whereas SynCheck keeps an eye on unfaithful sentences during generation rather than just evaluating the final answer [20]. Together, they cover different points in the evaluation pipeline, though how reliable they are still depends on evaluator quality, domain fit, and whether generation-level signals are accessible.

4. Comparative Analysis

The Qualitative Comparison of Representative Studies for Local PDF-Based RAG is shown in Table 1.

Table 1. Qualitative Comparison of Representative Studies for Local PDF-Based RAG

Study/category	Evaluation Focus	Main reported finding	Local deployment implication
ColBERTv2 [6] Retrieval optimization	Retrieval quality and index footprint	Improves fine-grained ranking while reducing late-interaction storage requirements	Higher-quality retrieval, but heavier than a single-vector index
DSE [7] Retrieval optimization	Visually complex document retrieval	Preserves page layout and visual evidence beyond OCR-only text extraction	Useful for complex PDFs, but requires image storage and visual encoding
LongLLMLingua [2] Context compression	Tokens, cost, latency, and task performance	Reduces prompt length while prioritizing query-relevant content	Improves efficiency, but adds a compression pass
RECOMP [10] Context compression	Open-domain QA and language-modeling tasks	Suppresses redundant or unhelpful retrieved context	Controls noise, but task-specific compressors may transfer poorly
ALCE [13] Evidence grounding	Answer and citation quality	Separates answer quality from citation correctness and completeness	Provides clear citation targets. But evaluation may add model cost
MiniCheck [14] Evidence grounding	Claim-document consistency	Provides relatively lightweight claim-evidence verification	Locally feasible, but domain shift remains a risk

Note. The numerical results from these six studies are based on different configuration environments, datasets, models, evaluation metrics, and hardware environments. Therefore, they should not be regarded as directly comparable performance scores. Local deployment implications represent a qualitative synthesis of computational, storage, and evidential requirements.

4.1. Retrieval Accuracy

Preserving document structure through chunking and enabling fine-grained alignment during retrieval—that's really what pushes retrieval accuracy forward. ColBERTv2 with late interaction and DSE with visual embeddings both improve precision across textual and visual evidence, while VisDoM and LongRAG extend those ideas to multi-document and long-context scenarios [6–9]. The trade-off, though, is that local deployment tends to require larger indexes, visual encoders, and a fair bit more compute.

4.2. Inference Efficiency

Inference efficiency depends on the whole pipeline, not just the generation step by itself. RECOMP, SEER, and EXIT each offer a different take on context reduction—selective augmentation, evidence extraction, and context-aware extractive compression, respectively [10–12]. They can definitely cut down on what the generator has to chew through, but that extra compression or extraction stage can add its own delay. So a sensible local pipeline really ought to balance things out: keep the initial retrieval light, rerank only a small candidate pool, and compress just the passages that actually get selected.

4.3. Citation Quality

Reliable citation attribution relies on persistent source identifiers and decoupled verification of both correctness and completeness. To improve traceability and reduce hallucination, the generator should cite original passages or explicit page regions rather than compressed paraphrases. While retrieval identifies candidate evidence, grounding verifies whether each citation genuinely backs its associated claim. Recent frameworks operationalize these requirements across distinct levels: ALCE and VISA focus on general citation evaluation and region-level attribution, whereas AGREE and ReClaim enable grounded generation and sentence-level reference-claim alignment [13, 15–17].

4.4. Answer Faithfulness

Answer faithfulness cannot be inferred from retrieval relevance or citation presence alone. MiniCheck indicates that compact verification models can support claim-level verification in local systems. However, verification cannot reconstruct missing or incorrectly compressed evidence. When adequate support is unavailable, abstention is more reliable than generating a plausible answer from weak context. MiniCheck, RAGAs, RAGTruth, and SynCheck address complementary aspects of answer faithfulness through claim verification, reference-free evaluation, hallucination annotation, and generation-time monitoring [14,18–20].

5. Challenges and Future Directions

5.1. PDF Parsing and Metadata Quality

Parsing errors can lead to inaccurate retrieval results, loss of context, loss of structural information, and incorrect reading order even before the retrieval begins. Future benchmarks should include scanned and complex-layout PDF files and preserve metadata at the page, section, and bounding box levels for each retrievable unit. In addition, the evaluation of parsing quality needs to be improved, as errors arising during document ingestion may affect retrieval, citation, and answer generation.

5.2. Trade-offs under Local Resource Constraints

Locally deployed systems must strike a balance between retrieval quality and memory usage, index size, latency, and model requirements. While more powerful retrievers, rerankers, compressors, and verification models may improve performance at individual stages, they also increase hardware requirements and system complexity. Future research should report peak memory usage, index update time, and end-to-end latency on fixed hardware, rather than merely presenting retrieval accuracy or token reduction.

5.3. Joint Optimization and Unified Evaluation

A sensible system design ought to hold onto metadata from the moment documents are ingested, pair a lightweight retriever with an optional reranking step, compress copies of whatever evidence gets pulled, and check claims before the final answer is shown. That said, these pieces really need to be assessed as a whole pipeline rather than as separate modules working in isolation. Going forward, evaluations should be run on the same PDF corpus and hardware setup, covering retrieval accuracy, token count, latency, memory footprint, citation quality, and whether the answers actually stay true to the evidence.

6. Conclusion

This review examined how retrieval optimization, context compression, and evidence grounding collectively improve locally deployed PDF-RAG systems. Rather than operating in isolation, these components address complementary stages of a unified processing pipeline: retrieval identifies candidate evidence, compression removes redundant context to reduce generation overhead, and grounding checks whether the final responses and their citations are supported by the source material. However, trade-offs between these stages are unavoidable. Large multi-vector indexes and complex retrieval models may exceed local computational limits, whereas aggressive context compression may remove structural details needed for accurate citation and verification.

A practical local system isn't about picking one best technique—it's about coordinating pipeline stages within tight hardware limits. On constrained devices, metadata-preserving chunking, lightweight retrieval, source-linked citations, and evidence-based refusal should come first; reranking, compression, multimodal retrieval, and standalone verifiers are optional. In practice, that means attaching page and structural metadata during indexing, reranking only a small candidate set, keeping original text alongside compressed context, and checking key claims before final output. At a minimum, the device needs enough memory to run the embedder, index, and generator at acceptable speed. If evidence falls short, the system should plainly say so.

References

[1] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, M., & Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv preprint arXiv:2312.10997. https://doi.org/10.48550/arXiv.2312.10997

[2] Jiang, H., Wu, Q., Luo, X., Li, D., Lin, C.-Y., Yang, Y., & Qiu, L. (2024). LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression. In Proceedings of ACL 2024 (pp. 1658–1677). https://doi.org/10.18653/v1/2024.acl-long.91

[3] Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., & Li, Q. (2024). A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 6491–6501). https://doi.org/10.1145/3637528.3671470

[4] Khan, A. A., Hasan, M. T., Kemell, K. K., Rasku, J., & Abrahamsson, P. (2024). Developing Retrieval Augmented Generation (RAG) Based LLM Systems from PDFs: An Experience Report. arXiv preprint arXiv:2410.15944. https://doi.org/10.48550/arXiv.2410.15944

[5] Wang, X., Wang, Z., Gao, X., Zhang, F., Wu, Y., Xu, Z., Shi, T., Wang, Z., Li, S., Qian, Q., Yin, R., Lv, C., Zheng, X., & Huang, X. (2024). Searching for Best Practices in Retrieval-Augmented Generation. In Proceedings of EMNLP 2024 (pp. 17716–17736). https://doi.org/10.18653/v1/2024.emnlp-main.981

[6] Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., & Zaharia, M. (2022). ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In Proceedings of NAACL 2022 (pp. 3715–3734). https://doi.org/10.18653/v1/2022.naacl-main.272

[7] Ma, X., Lin, S.-C., Li, M., Chen, W., & Lin, J. (2024). Unifying Multimodal Retrieval via Document Screenshot Embedding. In Proceedings of EMNLP 2024 (pp. 6492–6505). https://doi.org/10.18653/v1/2024.emnlp-main.373

[8] Suri, M., Mathur, P., Dernoncourt, F., Goswami, K., Rossi, R. A., & Manocha, D. (2025). VisDoM: Multi-Document QA with Visually Rich Elements Using Multimodal Retrieval-Augmented Generation. In Proceedings of NAACL 2025 (pp. 6088–6109). https://doi.org/10.18653/v1/2025.naacl-long.310

[9] Zhao, Q., Wang, R., Cen, Y., Zha, D., Tan, S., Dong, Y., & Tang, J. (2024). LongRAG: A Dual-Perspective Retrieval-Augmented Generation Paradigm for Long-Context Question Answering. In Proceedings of EMNLP 2024 (pp. 22600–22632). https://doi.org/10.18653/v1/2024.emnlp-main.1259

[10] Xu, F., Shi, W., & Choi, E. (2024). RECOMP: Improving Retrieval-Augmented LMs with Context Compression and Selective Augmentation. In Proceedings of the International Conference on Learning Representations. https://openreview.net/forum?id=mlJLVigNHp

[11] Zhao, X., Li, D., Zhong, Y., Hu, B., Chen, Y., Hu, B., & Zhang, M. (2024). SEER: Self-Aligned Evidence Extraction for Retrieval-Augmented Generation. In Proceedings of EMNLP 2024 (pp. 3027–3041). https://doi.org/10.18653/v1/2024.emnlp-main.178

[12] Hwang, T., Cho, S., Jeong, S., Song, H., Han, S., & Park, J. C. (2025). EXIT: Context-Aware Extractive Compression for Enhancing Retrieval-Augmented Generation. In Findings of ACL 2025 (pp. 4895–4924). https://doi.org/10.18653/v1/2025.findings-acl.253

[13] Gao, T., Yen, H., Yu, J., & Chen, D. (2023). Enabling Large Language Models to Generate Text with Citations. In Proceedings of EMNLP 2023 (pp. 6465–6488). https://doi.org/10.18653/v1/2023.emnlp-main.398

[14] Tang, L., Laban, P., & Durrett, G. (2024). MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents. In Proceedings of EMNLP 2024 (pp. 8818–8847). https://doi.org/10.18653/v1/2024.emnlp-main.499

[15] Ma, X., Zhuang, S., Koopman, B., Zuccon, G., Chen, W., & Lin, J. (2025). VISA: Retrieval Augmented Generation with Visual Source Attribution. In Proceedings of ACL 2025 (pp. 30154–30169). https://doi.org/10.18653/v1/2025.acl-long.1456

[16] Ye, X., Sun, R., Arik, S. O., & Pfister, T. (2024). Effective Large Language Model Adaptation for Improved Grounding and Citation Generation. In Proceedings of NAACL 2024 (pp. 6237–6251). https://doi.org/10.18653/v1/2024.naacl-long.346

[17] Xia, S., Wang, X., Liang, J., Zhang, Y., Zhou, W., Deng, J., Yu, F., & Xiao, Y. (2025). Ground Every Sentence: Improving Retrieval-Augmented LLMs with Interleaved Reference-Claim Generation. In Findings of NAACL 2025 (pp. 969–988). https://doi.org/10.18653/v1/2025.findings-naacl.55

[18] Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAs: Automated Evaluation of Retrieval Augmented Generation. In Proceedings of EACL 2024: System

Demonstrations (pp. 150–158). https://doi.org/10.18653/v1/2024.eacl-demo.16

[19] Niu, C., Wu, Y., Zhu, J., Xu, S., Shum, K., Zhong, R., Song, J., & Zhang, T. (2024). RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. In Proceedings of ACL 2024 (pp. 10862–10878). https://doi.org/10.18653/v1/2024.acl-long.585

[20] Wu, D., Gu, J.-C., Yin, F., Peng, N., & Chang, K.-W. (2024). Synchronous Faithfulness Monitoring for Trustworthy Retrieval-Augmented Generation. In Proceedings of EMNLP 2024 (pp. 9390–9406). https://doi.org/10.18653/v1/2024.emnlp-main.527