Saad Ahmad

Retrieval-Augmented Generation: State-of-the-Art and Use Cases

Supervisor: Sadegh Keshtkar

Agenda

- Motivation & Definition
- Architecture & Retriever Types
- Key RAG Models
- Advanced RAG Variants
- Benchmarks & Results
- Applications & Deployment
- Challenges & Future Work

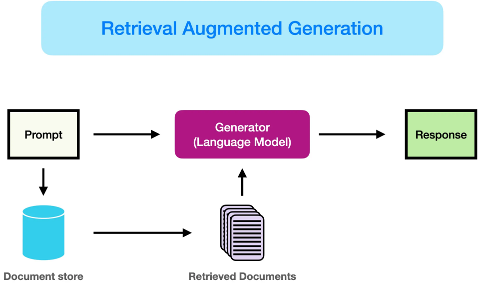
Motivation

Why Retrieval-Augmented Generation?

- Addresses factual errors and hallucinations (Lewis et al., 2020)
- Accesses external knowledge dynamically
- Useful in domains with evolving data

What is RAG?

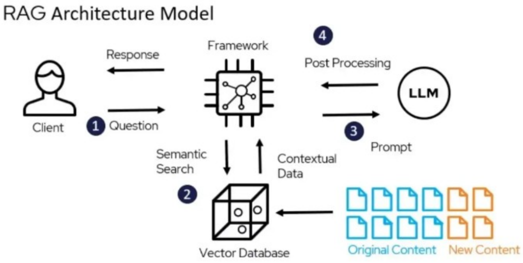
- Combines retriever and generator modules
- Generator is conditioned on retrieved documents
- Enables grounded, knowledge-rich responses



Architecture

RAG System Architecture

- Query processed by retriever to fetch relevant docs
- Generator combines query and docs to answer
- Often built with dense retrievers + seq2seq transformers

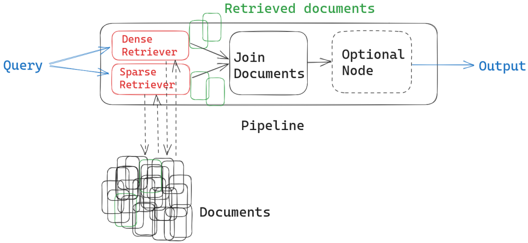


Building RAG Systems: Tools and Infrastructure

- **Vector Databases:** Fast similarity search over embeddings.
  - Examples: FAISS, Pinecone
- **LLM Integration Frameworks:** Combine retrieval and generation steps.
  - Example: LangChain simplifies orchestration
- **Indexing Pipelines:** Manage document chunking, embeddings, updates.
  - Example: LlamaIndex for document indexing
- **APIs/Platforms:** RAG-as-a-service platforms
  - Examples: Azure Cognitive Search + OpenAI, Databricks RAG tools

Dense vs Sparse vs Hybrid Retrieval

- Dense: semantic similarity (Karpukhin et al., 2020)
- Sparse: term-based (e.g., BM25)
- Hybrid: combines both (Guu et al., 2020)



Real-World Example: Slack AI

- Slack AI uses vector DB + OpenAI API
- Query $\rightarrow$ embedding $\rightarrow$ search $\rightarrow$ inject into prompt
- Final response generated with context from matching docs

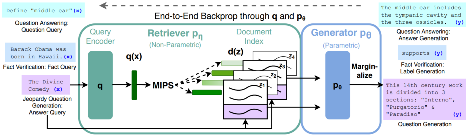
RAG vs Other Approaches

- **Prompt Engineering:**
  - Uses existing model with no training
  - Quick to implement, no additional data required
  - Limited in injecting new facts – reframes query but does not change the model’s internal knowledge or parameters
- **Retrieval-Augmented Generation (RAG):**
  - Requires external knowledge base (e.g., documents + vector DB)
  - Enables dynamic updates and domain-specific grounding
  - Increased system complexity and inference cost
- **Fine-Tuning:**
  - Needs labeled domain-specific data
  - Model internalizes knowledge and can specialize
  - High cost, risk of overfitting, model becomes static again

Key Models

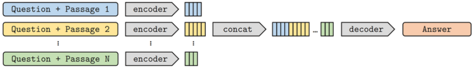
Facebook RAG (2020)

- Combines DPR retriever + BART generator
- End-to-end trainable (Lewis et al., 2020)
- Strong performance in QA tasks



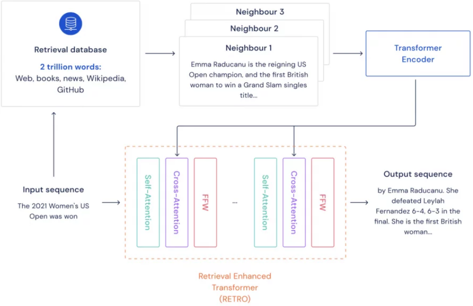
Fusion-in-Decoder (FiD)

- Uses T5; fuses multiple retrieved docs inside decoder
- Allows evidence aggregation across documents
- Outperforms RAG on multi-hop QA tasks



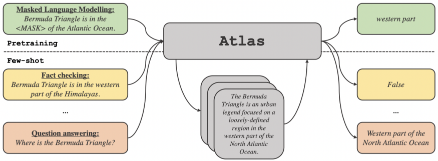
RETRO (DeepMind)

- Uses frozen LMs + external memory lookup
- Retrieves similar chunks using local context
- Efficient for very large-scale retrieval



Atlas (Meta AI)

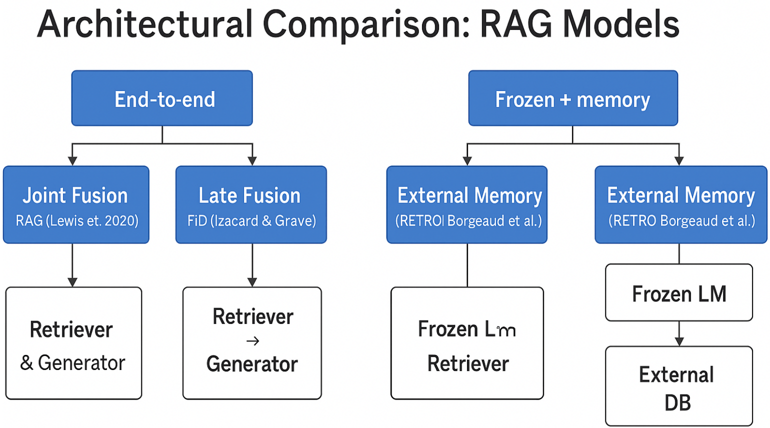
- Unified multitask RAG model (Izacard et al., 2022)
- Strong on QA, summarization, dialogue
- Combines dense retriever + T5



Comparison of RAG Models

- **RAG**: DPR + BART; end-to-end trainable (Lewis et al., 2020)
- **FiD**: Late fusion; T5 decoder integrates evidence (Izacard & Grave, 2020)
- **RETRO**: Frozen LM + external memory; scalable and modular (Borgeaud et al., 2022)
- **Atlas**: Unified multitask; flexible retriever-generator setup (Izacard et al., 2022)

Architectural Comparison: RAG Models

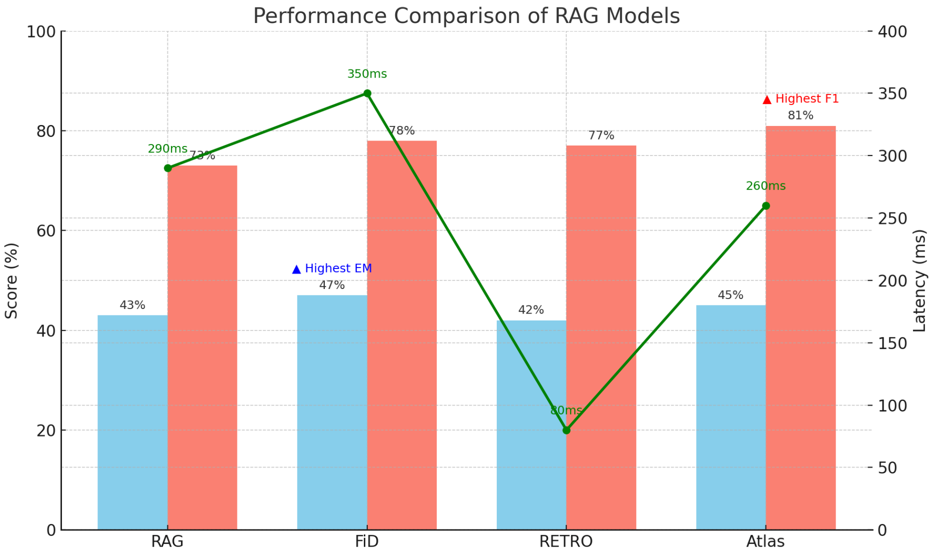


Benchmarks

Evaluation Metrics

- Exact Match (EM), F1 Score
- Latency (ms), Retrieval Accuracy
- Datasets: NQ, TriviaQA, HotpotQA, KILT
- Note: The benchmark results were calculated using HotpotQA

Performance Overview

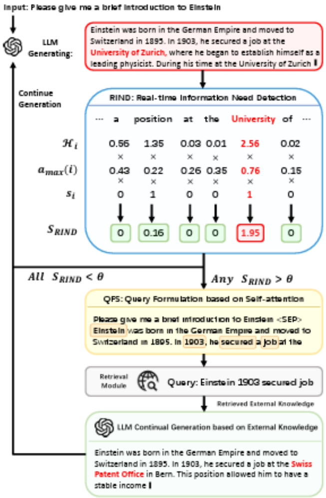


Advanced RAG Variants

DRAGON: Uncertainty-Aware RAG

- Dynamically triggers retrieval only when model is uncertain.
- Uses entropy threshold to reduce unnecessary lookups.
- Balances generation confidence and retrieval cost.

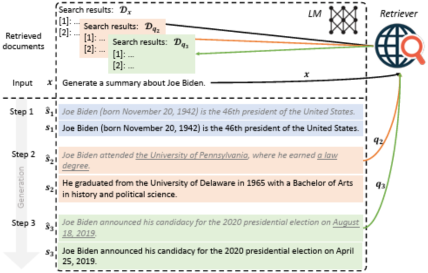
Source: Lin et al. (2024). *https://arxiv.org/abs/2403.10081*



FLARE: Forward-Looking Active Retrieval

- Performs retrieval mid-generation when needed.
- Uses entropy of output tokens to decide retrieval time.
- Improves factual grounding while reducing latency.

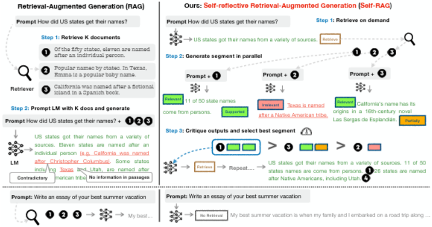
Source: Nakano et al. (2023). *https://arxiv.org/abs/2305.06983*



Self-RAG: Retrieval with Self-Critique

- Initial answer generated, then critiqued by the same model.
- Low confidence triggers re-retrieval and regeneration.
- Mitigates hallucinations using self-feedback loop.

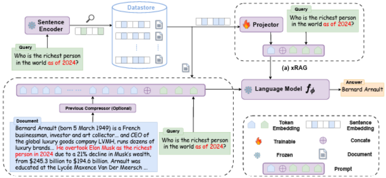
Source: Asai et al. (2023). *https://arxiv.org/abs/2310.11511*



xRAG: Cross-Context Retrieval

- Retrieves from multiple memory types (search, internal, external).
- Ranks results across diverse retrieval streams.
- Strong results on multi-hop and hybrid domain queries.

Source: Zhang et al. (2024).
https://arxiv.org/abs/2405.13792



Evaluation Datasets and Metrics

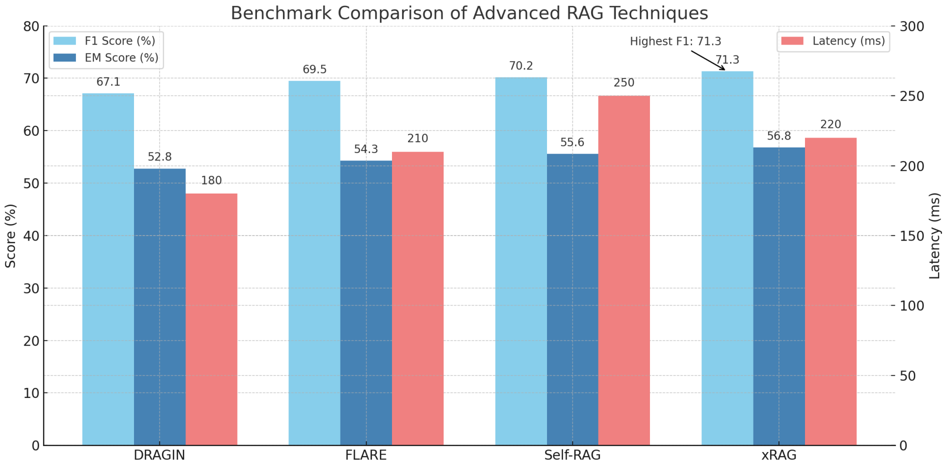
Datasets Used for Evaluation:

- **Natural Questions (NQ):** Open-domain QA dataset with real user queries.
- **TriviaQA:** Question-answer pairs with high lexical diversity.
- **HotpotQA:** Multi-hop reasoning required across documents.
- **KILT Benchmark:** Standardized format across 5+ QA datasets.
- **Note:** The benchmark comparison in the upcoming was conducted using HotpotQA

Metrics Evaluated:

- **F1 Score:** Measures overlap between predicted and ground truth spans.
- **Exact Match (EM):** Binary metric for exact span match.
- **Latency:** Average response time per query (ms).

Benchmark: Advanced RAG Techniques



Sources: Lin et al. (2024), Nakano et al. (2023), Asai et al. (2023), Zhang et al. (2024)

Applications

Use Cases in Practice

- Enterprise search (e.g., Slack AI)
- Chatbots (e.g., Bing Copilot)
- Scientific/biomedical QA (BioRAG)
- Legal & financial document assistants

Adoption in Industry

- Perplexity.ai uses hybrid RAG for live web answers
- Bing Chat leverages RAG over search index
- OpenAssistant uses fine-tuned RAG for dialogue

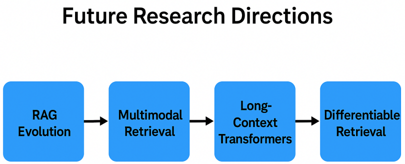
Discussion

Challenges in RAG

- Retrieval noise and relevance mismatch
- Latency from document fetching
- Domain adaptation and generalization

Future Research Directions

- **Multimodal retrieval (text + image):** *Enable queries across images, audio, and tables alongside text.*
- **Long-context transformers:** *Use models like Claude or GPT-4-128K to reduce need for retrieval.*
- **Differentiable retrieval:** *Train the retriever via backpropagation with the generator.*



Conclusion

- **RAG** significantly enhances factual accuracy by grounding responses in external knowledge.
- Multiple architectures (e.g., RAG, FiD, Atlas) balance trade-offs between accuracy, latency, and scalability.
- Real-world adoption across search, chat, legal, and scientific domains confirms RAG's practical value.
- Continued research in differentiable retrieval and long-context handling will shape the next generation of RAG systems.
- RAG balances flexibility and freshness of knowledge, unlike static fine-tuning or prompt-only tweaks.

References

- Lewis, P., et al. (2020). Retrieval-Augmented Generation. arXiv:2005.11401
- Guu, K., et al. (2020). REALM. arXiv:2002.08909
- Izacard, G., & Grave, E. (2020). FiD. arXiv:2007.01282
- Borgeaud, S., et al. (2022). RETRO. Nature, 610(7930), 754–761
- Izacard, G., et al. (2022). Atlas. arXiv:2208.03299
- Lin et al. (2024). DRAGON. arXiv:2403.10081
- Nakano et al. (2023). FLARE. arXiv:2305.06983
- Asai et al. (2023). Self-RAG. arxiv.org/abs/2310.11511
- Zhang et al. (2024). xRAG. arXiv:2405.13792