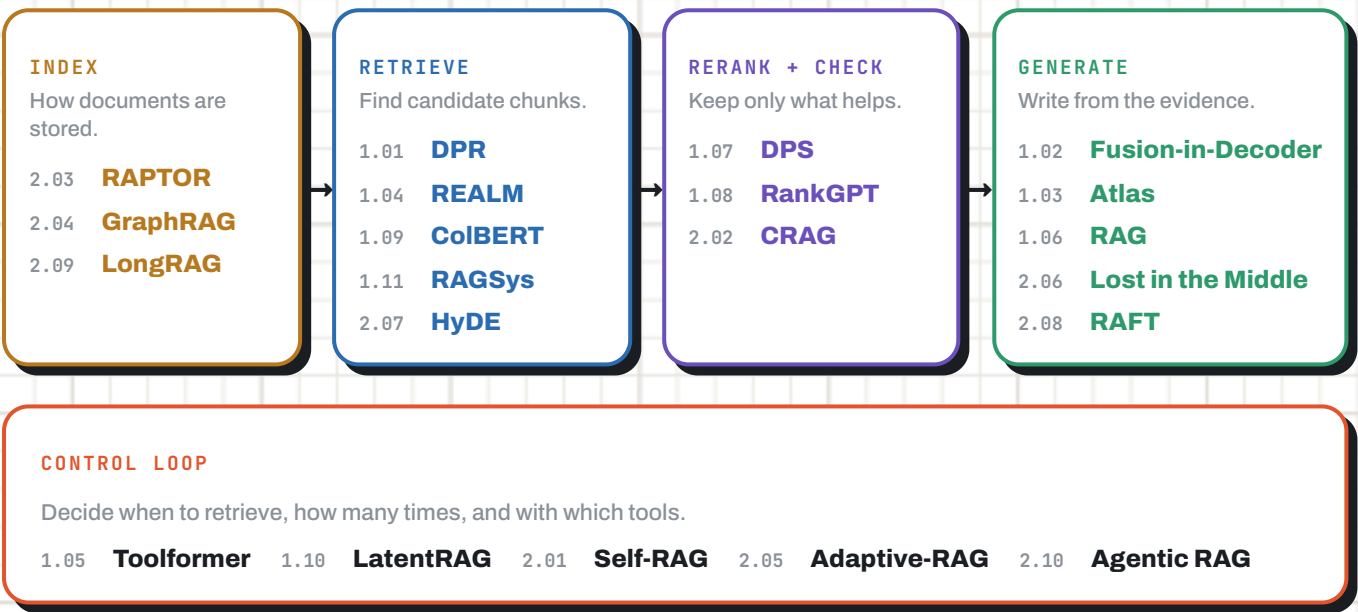


21 RAG Papers

MUST READ FOR AI ENGINEERS

Every paper from both RAG reels: what it changed, where it fits in your pipeline, and which RAG problem it fixes.



PART 1 = the "11 RAG research papers" reel, numbered 01-11 as in the reel
PART 2 = the "10 RAG papers" reel, numbered 00-10 as in the reel

SUMIT KUMAR DASH
SKDNEURON.COM • @SKDNEURON

SAVE FOR YOUR NEXT INTERVIEW

The cheat sheet.

ONE LINE EACH

PART 1 · THE "11 RAG RESEARCH PAPERS" REEL

#	PAPER	YEAR	STAGE	THE IDEA IN ONE LINE
1.01	DPR	2020	RETRIEVE	Two encoders turn questions and passages into vectors.
1.02	Fusion-in-Decoder	2020	GENERATE	Encode passages separately, read them all at once.
1.03	Atlas	2022	GENERATE	A small retrieval model that learns from a few examples.
1.04	REALM	2020	RETRIEVE	The retriever is learned during pre-training itself.
1.05	Toolformer	2023	CONTROL	The model teaches itself when to call search and other tools.
1.06	RAG	2020	GENERATE	The paper that named it: retrieve first, then generate.
1.07	DPS	2025	RERANK	Pick the right set of passages, not a fixed top-k.
1.08	RankGPT	2023	RERANK	An LLM reorders the retrieved passages by relevance.
1.09	ColBERT	2020	RETRIEVE	Match query and document token by token, cheaply.
1.10	LatentRAG	2026	CONTROL	Agentic RAG's thinking in hidden vectors: about 90% faster.
1.11	RAGSys	2024	RETRIEVE	Pick few-shot examples like a recommender: diverse, not similar.

PART 2 · THE "10 RAG PAPERS" REEL

#	PAPER	YEAR	STAGE	THE IDEA IN ONE LINE
2.00	RAG	2020	GENERATE	Same paper as 1.06.
2.01	Self-RAG	2023	CONTROL	Decides when to retrieve, then critiques its own answer.
2.02	Corrective RAG	2024	CHECK	Grades the retrieved docs and fixes bad ones with web search.
2.03	RAPTOR	2024	INDEX	A tree of summaries to retrieve from, at every level.
2.04	GraphRAG	2024	INDEX	A knowledge graph for questions about the whole corpus.
2.05	Adaptive-RAG	2024	CONTROL	Routes each question by how hard it is.
2.06	Lost in the Middle	2023	GENERATE	Facts in the middle of a long context get missed.
2.07	HyDE	2022	RETRIEVE	Search with a hypothetical answer instead of the question.
2.08	RAFT	2024	GENERATE	Fine-tune the model to ignore distractor documents.
2.09	LongRAG	2024	INDEX	Fewer, much longer retrieval units for long-context LLMs.
2.10	Agentic RAG	2025	CONTROL	Agents plan, retrieve and iterate (a survey).

1.01 DPR Dense Passage Retrieval

RETRIEVE

WHAT IT CHANGED

Two encoders, one for questions and one for passages, trained so matching pairs end up close together. Retrieval becomes a vector search.

WHY IT MATTERS

The dense retriever pattern behind every vector-database RAG system, trained from a fairly small set of question-passage pairs.

Beats BM25 by 9–19 points on top-20 retrieval

SAY IT **"Embed questions and passages separately, then retrieve by vector similarity."**

Karpukhin et al., 2020 • Dense Passage Retrieval for Open-Domain Question Answering • arxiv.org/abs/2004.04906

1.02 Fusion-in-Decoder FiD

GENERATE

WHAT IT CHANGED

Each retrieved passage is encoded separately with the question, then the decoder reads all of them together to write one answer.

WHY IT MATTERS

Showed generators are good at combining evidence: accuracy kept rising as more passages were retrieved. Top results on Natural Questions and TriviaQA.

Split encoding, joint decoding

SAY IT **"Encode passages independently, fuse them in the decoder: more evidence, better answers."**

Izacard & Grave, 2020 • Leveraging Passage Retrieval with Generative Models for Open Domain QA • arxiv.org/abs/2007.01282

1.03 Atlas Few-shot RAG

GENERATE

WHAT IT CHANGED

A retriever and a language model pre-trained together, so the system learns knowledge-heavy tasks from very few examples.

WHY IT MATTERS

Knowledge can live in the index instead of the weights, so a small model can beat a huge one, and the index can be updated without retraining.

42% on NQ from 64 examples: beats a 540B model, 50× smaller

SAY IT **"Put knowledge in the index, not the parameters: a small model plus retrieval beats a giant."**

Izacard et al., 2022 • Atlas: Few-shot Learning with Retrieval Augmented Language Models • arxiv.org/abs/2208.03299

1.04 REALM Retrieval-Augmented Language Model Pre-Training

RETRIEVE

WHAT IT CHANGED

Adds a retriever during pre-training and trains it without labels: masked-word prediction is the signal, back-propagated through retrieval.

WHY IT MATTERS

The first to show a retriever can be learned end to end over millions of documents, with knowledge you can inspect and swap.

+4–16 points on open-domain QA

SAY IT **"REALM learns what to retrieve while it learns language."**

Guu et al., 2020 • REALM: Retrieval-Augmented Language Model Pre-Training • arxiv.org/abs/2002.08909

1.05 Toolformer Language Models Teach Themselves to Use Tools

CONTROL

WHAT IT CHANGED

The model learns by itself which API to call, when, and with what arguments: calculator, Q&A, two search engines, translation and a calendar.

WHY IT MATTERS

Retrieval becomes one tool the model chooses to use, not a fixed first step. That's the idea behind tool-calling agents.

Self-supervised: a handful of demos per API

SAY IT "Search becomes a tool the model decides to call, not a step it's forced to take."

Schick et al., 2023 · Toolformer: Language Models Can Teach Themselves to Use Tools · arxiv.org/abs/2302.04761

1.06 RAG Retrieval-Augmented Generation

GENERATE

WHAT IT CHANGED

Joins a generator to a neural retriever over a dense Wikipedia index. Two versions: the same passages for the whole answer (RAG-Sequence), or new ones per token (RAG-Token).

WHY IT MATTERS

Named the pattern. Answers became more specific and factual, with sources, and knowledge could change without retraining.

State of the art on three open-domain QA tasks

SAY IT "Parametric memory plus a searchable index: retrieve evidence, then write from it."

Lewis et al., 2020 · Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks · arxiv.org/abs/2005.11401

1.07 DPS From Ranking to Selection

RERANK

WHAT IT CHANGED

A reranker trained to choose the best set of passages together, and how many, instead of scoring each one alone and keeping a fixed top-k.

WHY IT MATTERS

Small k misses evidence, large k adds noise. Choosing the set fixes multi-hop questions, and it plugs into an existing pipeline.

MuSiQue F1: +30% over Qwen3-reranker

SAY IT "Don't keep a fixed top-k: select the passages that work together."

Meng et al., 2025 · From Ranking to Selection: A Simple but Efficient Dynamic Passage Selector for RAG · arxiv.org/abs/2508.09497

1.08 RankGPT LLMs as Re-Ranking Agents

RERANK

WHAT IT CHANGED

Gives an LLM the retrieved passages and asks it to output them in order of relevance, then distills that skill into a small reranker.

WHY IT MATTERS

Properly prompted LLMs matched or beat supervised rerankers, and the distilled model is cheap enough for production.

Distilled 440M beats a 3B supervised model on BEIR

SAY IT "Let an LLM reorder the candidates, then distil it into a small, fast reranker."

Sun et al., 2023 · Is ChatGPT Good at Search? Investigating LLMs as Re-Ranking Agents · arxiv.org/abs/2304.09542

1.09 ColBERT Contextualized Late Interaction over BERT

RETRIEVE

WHAT IT CHANGED

Encodes query and document separately into one vector per token, then scores each query token against its best-matching document token.

WHY IT MATTERS

Close to BERT-reranker quality, but documents are encoded ahead of time, so it's fast enough to retrieve straight from a large collection.

100× faster, 10,000× fewer FLOPs per query

SAY IT "Token-level matching with pre-computed documents: reranker quality at retriever speed."

Khattab & Zaharia, 2020 · ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT · arxiv.org/abs/2004.12832

1.10 LatentRAG Latent Reasoning and Retrieval

CONTROL

WHAT IT CHANGED

Agentic RAG writes out thoughts and sub-queries token by token, which is slow. LatentRAG produces them as hidden vectors in one forward pass and retrieves with those.

WHY IT MATTERS

Multi-step agentic RAG at close to single-step speed, while a parallel decoder can still turn the hidden steps back into text.

About 90% lower latency, similar accuracy

SAY IT "Think and search in vectors instead of words, and agentic RAG gets about 10× faster."

Zheng & Worring, 2026 · LatentRAG: Latent Reasoning and Retrieval for Efficient Agentic RAG · arxiv.org/abs/2605.06285

1.11 RAGSys Item-Cold-Start Recommender as RAG System

RETRIEVE

WHAT IT CHANGED

Treats retrieving few-shot examples for a prompt as a cold-start recommendation problem: aim for discovery and new information, not only similarity.

WHY IT MATTERS

Ten near-identical examples teach the model little. The paper shows diversity and quality in the retrieved examples matter for in-context learning.

Diverse, high-quality examples over most-similar ones

SAY IT "When you retrieve examples for a prompt, optimise for what the model learns, not just similarity."

Contal & McGoldrick, 2024 · RAGSys: Item-Cold-Start Recommender as RAG System · arxiv.org/abs/2405.17587

Part 1 in one line: retrievers learned to find (1.01, 1.04, 1.09), generators learned to combine (1.02, 1.03, 1.06), rerankers learned to choose (1.07, 1.08), and the model learned to decide when to search (1.05, 1.10).

2.01 Self-RAG Retrieve, Generate, Critique

CONTROL

WHAT IT CHANGED

One model trained to emit special "reflection tokens":
should I retrieve? Is this passage relevant? Is my answer
supported by it?

WHY IT MATTERS

Retrieves only when needed and checks its own work,
which improves factuality and citations. Behaviour can be
tuned at inference time.

7B/13B beat ChatGPT on open QA and fact checks

SAY IT **"Self-RAG decides when to retrieve, then grades its own answer against the evidence."**

Asai et al., 2023 • Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection • arxiv.org/abs/2310.11511

2.02 Corrective RAG CRAG

CHECK

WHAT IT CHANGED

A light evaluator scores the retrieved documents.
Depending on its confidence, the system uses them,
replaces them with a web search, or combines both.

WHY IT MATTERS

Plain RAG trusts whatever it retrieves. CRAG catches
bad retrieval before it becomes a bad answer, and plugs
into any RAG pipeline.

Grade docs → keep, filter, or fall back to web search

SAY IT **"Check the retrieval before you trust it, and have a fallback when it's wrong."**

Yan et al., 2024 • Corrective Retrieval Augmented Generation • arxiv.org/abs/2401.15884

2.03 RAPTOR Tree-Organized Retrieval

INDEX

WHAT IT CHANGED

Embeds chunks, clusters them, summarises each
cluster, and repeats, building a tree from detail up to
overview. Search covers every level.

WHY IT MATTERS

Short chunks lose the big picture. With a tree, questions
that need a whole long document can still be answered.

+20 points on QuALITY with GPT-4

SAY IT **"Retrieve from a tree of summaries, so you can answer at any level of detail."**

Sarathi et al., 2024 • RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval • arxiv.org/abs/2401.18059

2.04 GraphRAG From Local to Global

INDEX

WHAT IT CHANGED

An LLM builds an entity knowledge graph from the
documents and pre-writes a summary for each group of
related entities. Answers combine those summaries.

WHY IT MATTERS

Chunk RAG fails on questions like "What are the main
themes?" GraphRAG answers them across a whole
corpus, around a million tokens in the paper.

Whole-corpus questions that chunk RAG can't answer

SAY IT **"For global questions, retrieve from a graph of summaries, not individual chunks."**

Edge et al., 2024 • From Local to Global: A Graph RAG Approach to Query-Focused Summarization • arxiv.org/abs/2404.16130

2.05 Adaptive-RAG Routing by Question Complexity

CONTROL

WHAT IT CHANGED

A small classifier predicts how hard each question is, then picks a strategy: answer directly, retrieve once, or retrieve over several steps.

WHY IT MATTERS

Simple questions stop paying for multi-step retrieval, and hard ones get it. Better accuracy and lower cost than one fixed strategy.

3 routes: no retrieval · one step · multi-step

SAY IT "Route by difficulty: don't run a multi-step pipeline for a one-hop question."

Jeong et al., 2024 · Adaptive-RAG: Learning to Adapt Retrieval-Augmented LLMs through Question Complexity · arxiv.org/abs/2403.14403

2.06 Lost in the Middle How LLMs Use Long Contexts

GENERATE

WHAT IT CHANGED

Moved the relevant passage to different positions in a long input and measured accuracy. Performance drops when the answer sits in the middle.

WHY IT MATTERS

Stuffing more chunks into the prompt can make answers worse. Put the best chunks first or last, and retrieve fewer, better ones.

Best "at the beginning or end of the input context"

SAY IT "Position matters: models use the start and end of the context best, the middle worst."

Liu et al., 2023 · Lost in the Middle: How Language Models Use Long Contexts · arxiv.org/abs/2307.03172

2.07 HyDE Hypothetical Document Embeddings

RETRIEVE

WHAT IT CHANGED

An LLM first writes a fake answer to the question. That answer is embedded, and the search finds real documents that look like it.

WHY IT MATTERS

Questions and answers are phrased differently, so answer-to-document search matches better. Works with no labelled training data.

Zero-shot: no relevance labels needed

SAY IT "Search with what the answer would look like, not with the question."

Gao et al., 2022 · Precise Zero-Shot Dense Retrieval without Relevance Labels · arxiv.org/abs/2212.10496

2.08 RAFT Retrieval-Augmented Fine-Tuning

GENERATE

WHAT IT CHANGED

Fine-tunes on questions paired with retrieved documents that include distractors. The model learns to quote the right document and reason step by step.

WHY IT MATTERS

Real retrieval is noisy. RAFT teaches a model to ignore the noise in your own domain, improving results on PubMed, HotpotQA and Gorilla.

Train with distractors + verbatim quotes

SAY IT "RAFT is open-book exam practice: learn to find the useful page and ignore the rest."

Zhang et al., 2024 · RAFT: Adapting Language Model to Domain Specific RAG · arxiv.org/abs/2403.10131

2.09 LongRAG RAG with Long-context LLMs

INDEX

WHAT IT CHANGED

Groups related documents into 4K-token retrieval units instead of 100-word chunks, so there are far fewer units to search, and a long-context LLM reads them.

WHY IT MATTERS

Tiny chunks make the retriever hunt for a needle and lose context. Long units move work to the reader, which modern LLMs handle well.

NQ: 62.7% exact match, no training, under 8 units

SAY IT "Long-context models let you retrieve fewer, bigger units, so the retriever's job gets easier."

Jiang et al., 2024 · LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs · arxiv.org/abs/2406.15319

2.10 Agentic RAG A Survey

CONTROL

WHAT IT CHANGED

Maps the field where AI agents run the RAG pipeline, using reflection, planning, tool use and multi-agent teamwork to decide what to retrieve next.

WHY IT MATTERS

Fixed retrieve-then-generate pipelines can't handle multi-step tasks. This survey sorts the agentic designs and their trade-offs.

Reflection · planning · tool use · multi-agent

SAY IT "Agentic RAG puts an agent in charge: it plans, retrieves, checks and retrieves again."

Singh et al., 2025 · Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG · arxiv.org/abs/2501.09136

Part 2 in one line: make the index smarter (2.03, 2.04, 2.09), the search sharper (2.07), the evidence cleaner (2.02, 2.08), the context better placed (2.06), and the pipeline adaptive (2.01, 2.05, 2.10).

Start here if you're new: 1.06 RAG → 1.01 DPR → 2.06 Lost in the Middle → 2.02 Corrective RAG → 2.04 GraphRAG → 2.10 Agentic RAG.

Which paper fixes your problem?

START FROM THE SYMPTOM

The **right chunk** is never retrieved.

DPR [1.01](#) **ColBERT** [1.09](#) **HyDE** [2.07](#)

It's retrieved, but **ranked too low** or top-k is wrong.

RankGPT [1.08](#) **DPS** [1.07](#)

Retrieved docs are **wrong or noisy**, and the model trusts them.

CRAG [2.02](#) **Self-RAG** [2.01](#) **RAFT** [2.08](#)

The answer is in the context, but **the model misses it**.

Lost in the Middle [2.06](#) **FiD** [1.02](#)

Users ask about **the whole collection**: themes, trends.

GraphRAG [2.04](#) **RAPTOR** [2.03](#)

Chunks are **too small** and lose their context.

LongRAG [2.09](#) **RAPTOR** [2.03](#)

Easy and hard questions get the same expensive pipeline.

Adaptive-RAG [2.05](#)

Questions need **several steps or tools**.

Agentic RAG [2.10](#) **Toolformer** [1.05](#)

Multi-step RAG works, but it's **too slow**.

LatentRAG [1.10](#)

Few labelled examples, or a niche domain.

Atlas [1.03](#) **RAFT** [2.08](#)

Retrieving **few-shot examples** for prompts.

RAGSys [1.11](#)

Found this useful?

Follow SKD Neuron for more explainers on RAG, AI agents and AI system design. Save this for your next interview.

@SKDNEURON
SKDNEURON.COM