DSC 261

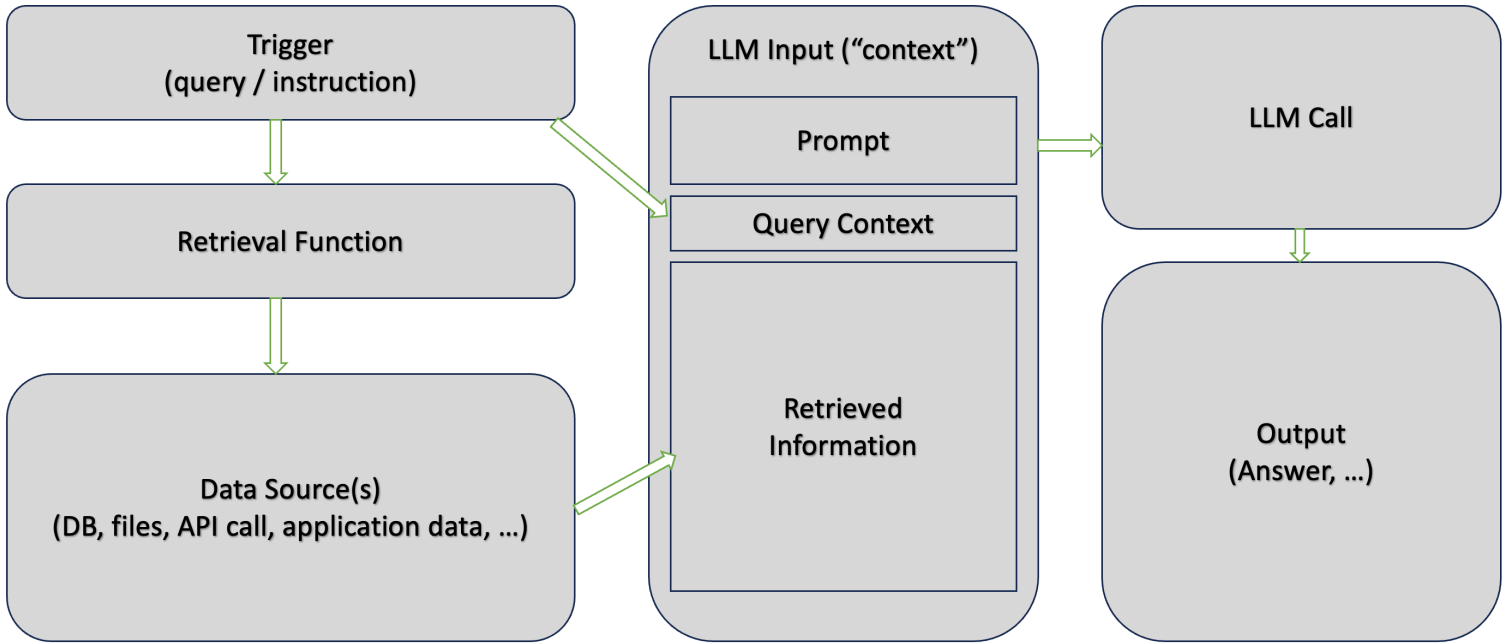
Grounding: finding, using, and checking evidence

- Grounding asks what evidence supports an answer.
- What has to go right between finding evidence and giving someone a justified answer?

Why an answer may need external evidence

- A model may know a lot about a topic and still lack the information this question needs.
- The question might concern a recent event, a private document, or a particular policy.
- Sometimes the answer is spread across several sources.
- We search for useful passages and put them into the model's input with the question.

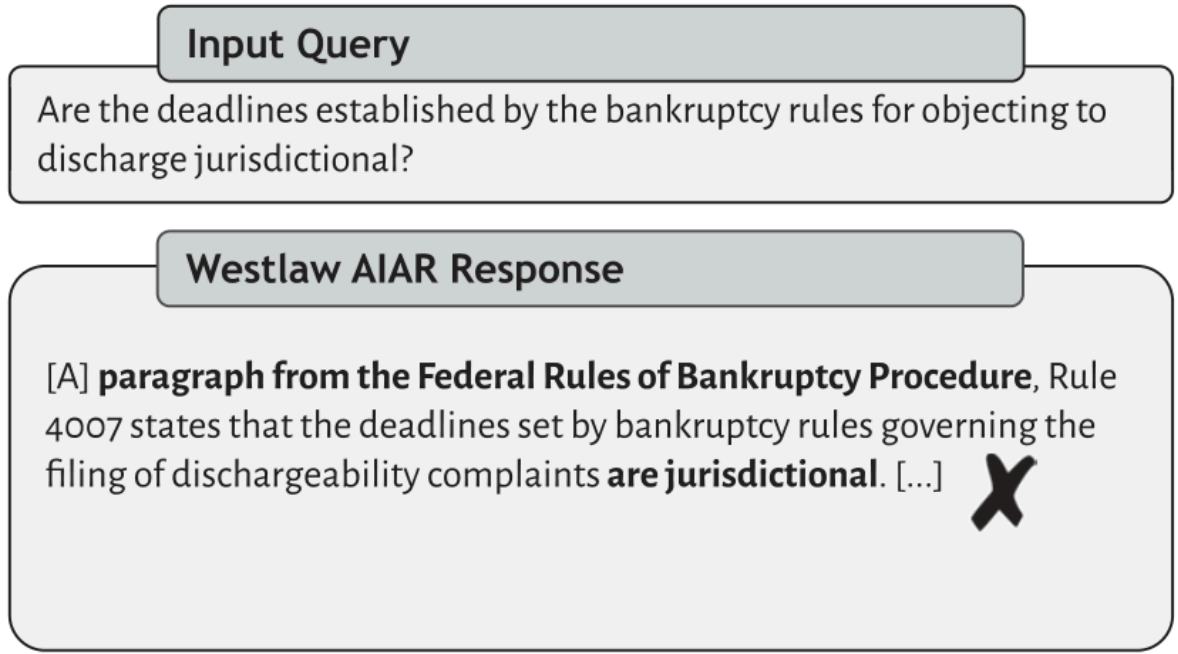
Retrieval supplies evidence to an answering model



- The retriever searches the available collection and returns passages that look relevant.
- They join the question and the instructions.

The answer cited a paragraph that does not exist

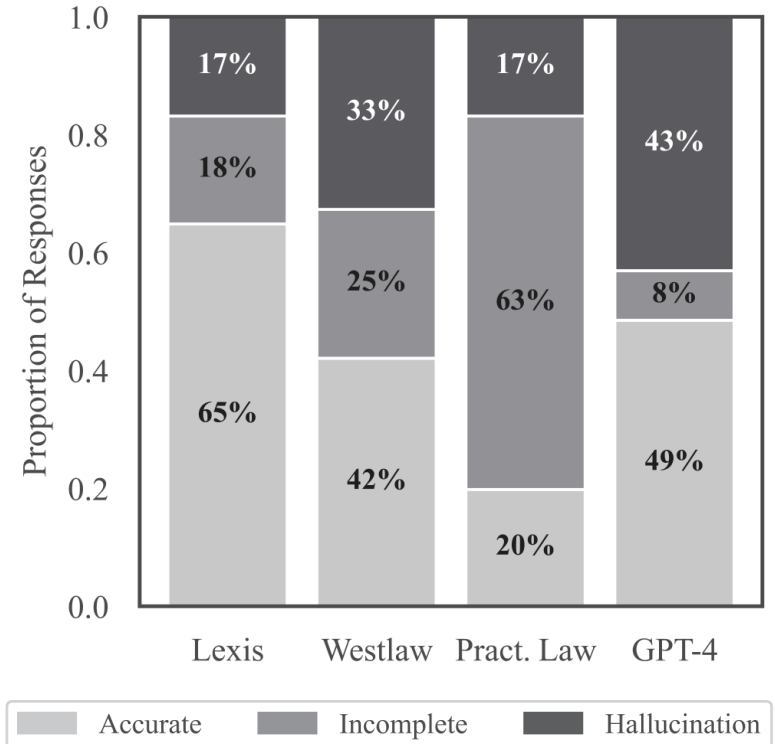
Magesh et al., JELS 2025. Products tested in 2024.



- We need to check the connection between the claim and the actual source.

Source-based tools still made errors

The same 202 questions for each system. Products tested in 2024.

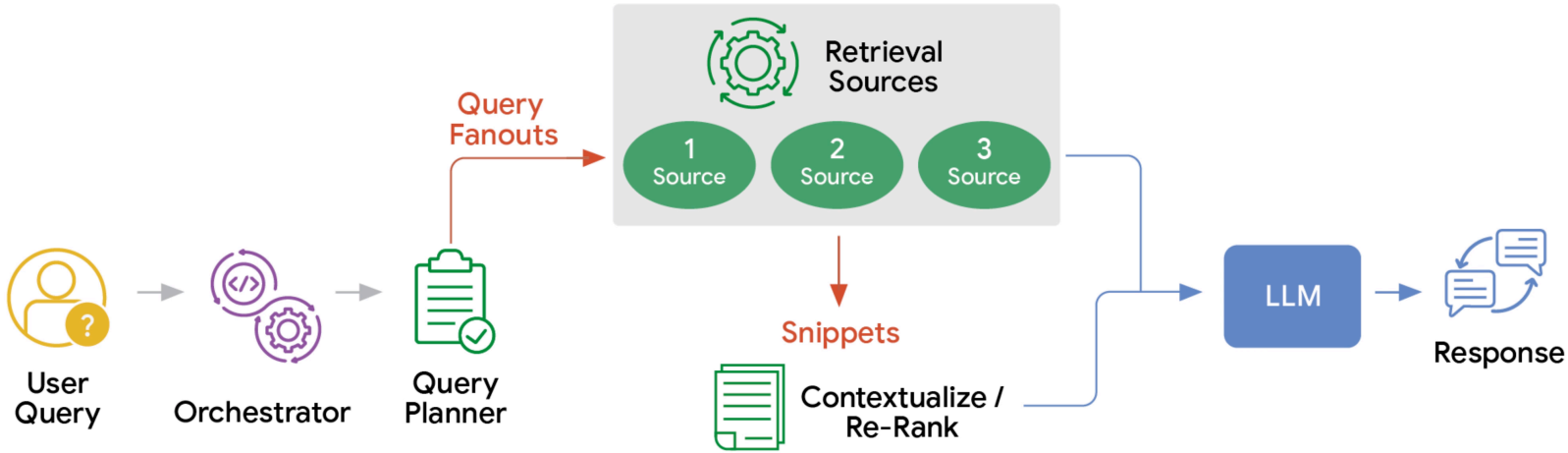


- These are responses classified as incorrect or misgrounded.
- The other portions distinguish accurate, grounded responses from incomplete ones.
- These results describe the versions tested in 2024.

What should we check in a source-based answer?

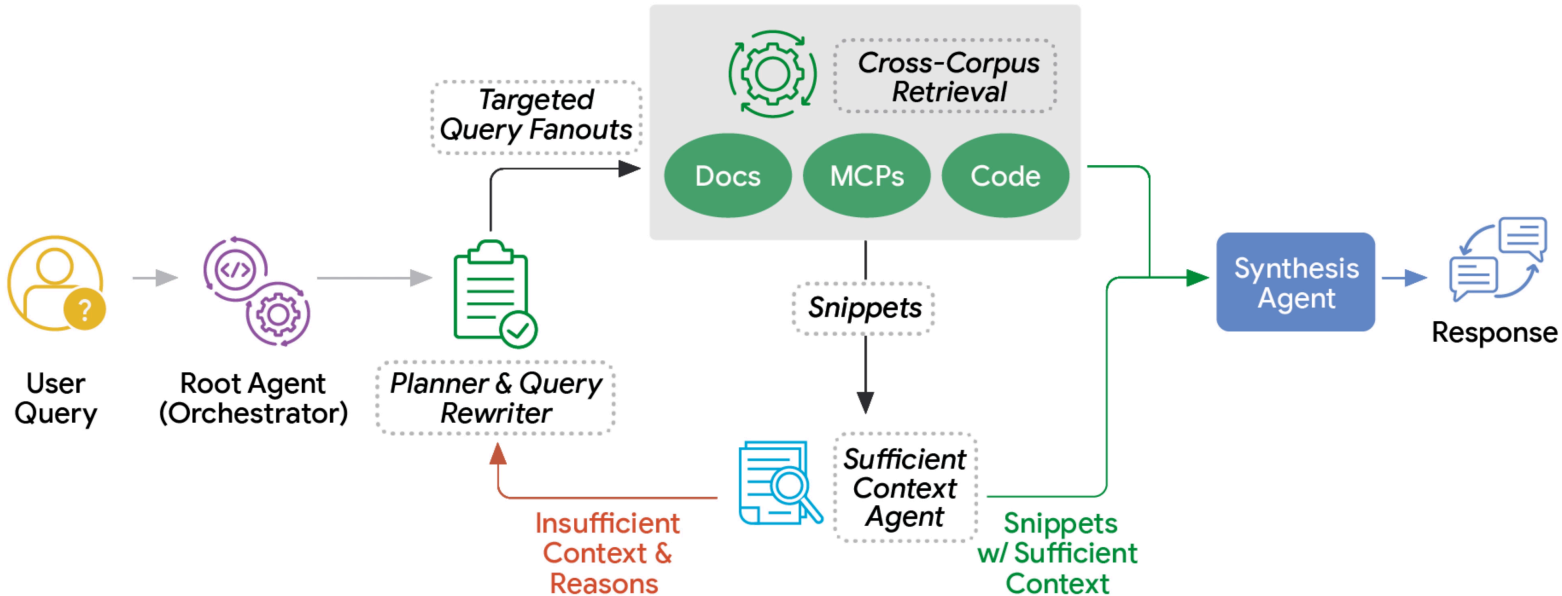
- First, is the answer correct?
- Second, do the cited sources actually support what it says?
- Can the reader find the relevant passage and see how it supports the claim?
- We also need completeness.
- Finally, is the source appropriate for this task?

How a RAG system searches and answers



- A planner to propose multiple searches
- A reranker to prioritize what comes back

Search again when evidence is missing



- Here we add feedback: the system can look at what it found and decide to search again.
- That gap goes to the planner, which adjusts the query, searches again, and brings back more evidence.
- If yes, follow the green path to synthesis, which writes the response.

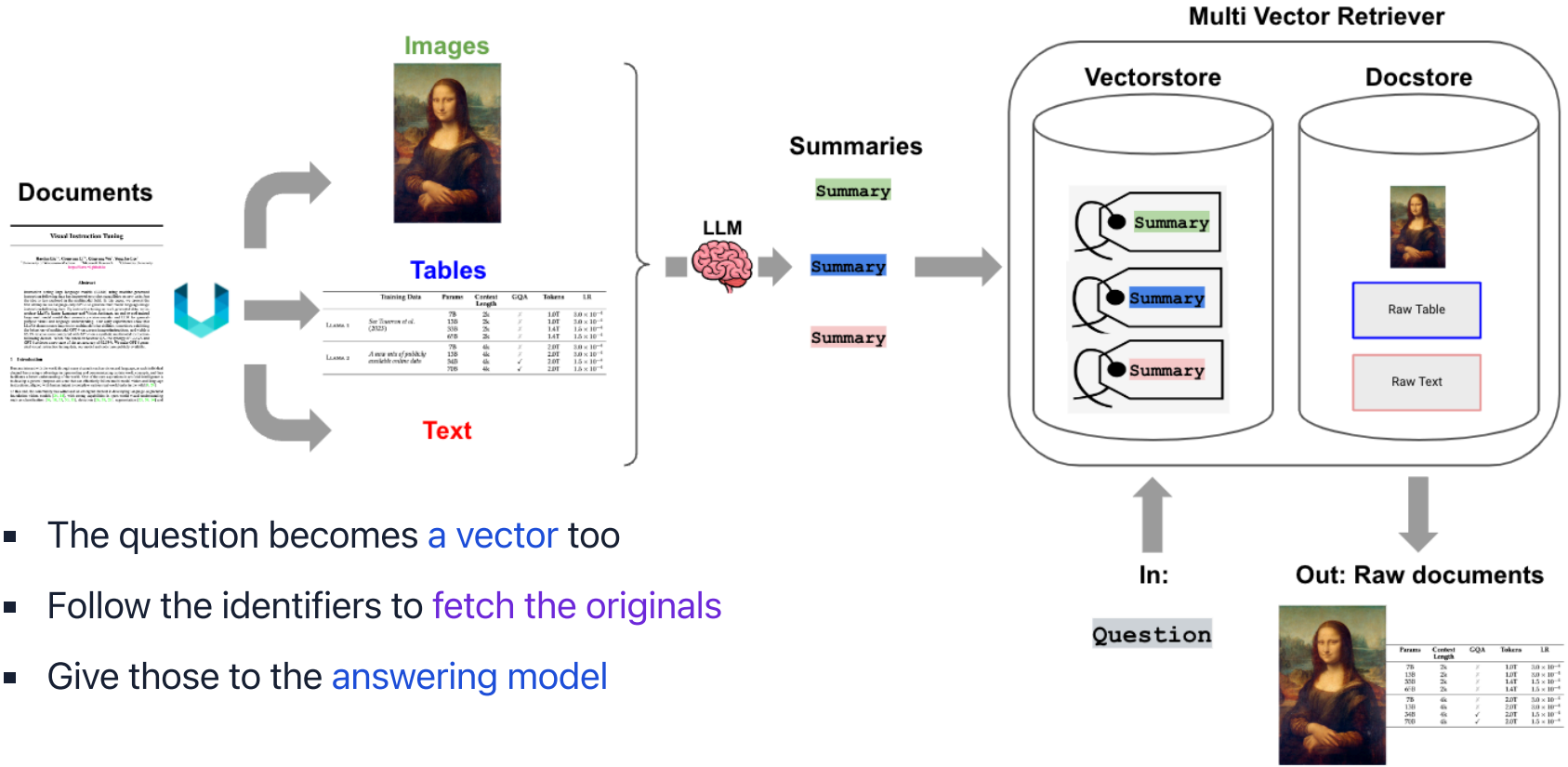
Evidence beyond text

- Real documents also have tables, charts, and images.

- Extract searchable text, generate descriptions or summaries, or use a multimodal embedding model

- Extracted text or the original chart or image.

Search summaries and retrieve original evidence



- The question becomes a vector too
- Follow the identifiers to fetch the originals
- Give those to the answering model

Evaluating a source-based answer

- Is it correct and complete? Do the citations really support the claims?

- Does the citation take them straight to the support, or make them hunt through a long document?

- Did they miss a necessary source, follow a bad hunch, or misuse good evidence?

What remains difficult?

- How do we find enough evidence when the question doesn't spell everything out?
- How do we know the model actually uses what it found instead of overlooking it?
- How do we decide which sources to trust if some are outdated, mistaken, or misleading?
- How do we check the answer without spending minutes hunting through documents?
- When is it useful to keep going and when do we have enough?

Class 1: how should retrieval and generation interact?

- In-context RALM: can we help a frozen model by supplying useful passages?
- Lewis RAG trains retrieval and generation together using the answer as feedback.
- IRCoT alternates retrieval with intermediate reasoning so one finding can guide another search.
- Adaptive retrieval asks when to search at all.

Class 2: can the evidence justify a usable answer?

- Lost in the Middle asks if location inside the input affects use.
- PoisonedRAG asks what if attackers plant misleading passages.
- Optional: how we combine sources when they disagree
- ALCE asks how to evaluate answers with citations, separating quality from cited support.
- Attribute First, Then Generate selects supporting spans before writing to make checking easier.

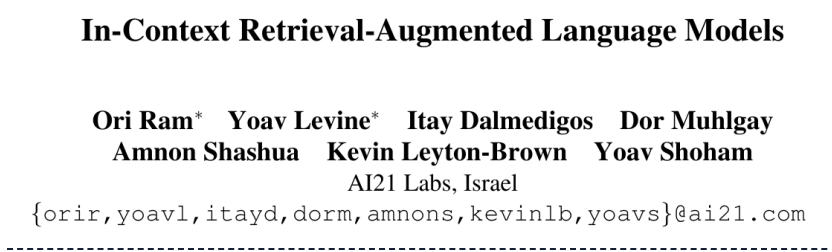
Supplying evidence and training a model

- Adding information to a model is not the same as training it.

- Updating a document changes what we can retrieve. Training changes how the model behaves.

- In-context RALM keeps the language model fixed. Lewis RAG trains retrieval with the generator.

Can retrieved text help a frozen language model?



In-Context Retrieval-Augmented Language Models

Ori Ram* Yoav Levine* Itay Dalmedigos Dor Muhlgay
Amnon Shashua Kevin Leyton-Brown Yoav Shoham
AI21 Labs, Israel
{orir,yoavl,itayd,dorm,amnons,kevinlb,yoavs}@ai21.com

In-Context Retrieval-Augmented Language Models

Ori Ram and colleagues

TACL 2023

A pretrained language model must continue text using external documents while its weights stay fixed.

Earlier retrieval-augmented models often required architecture changes or model training.

In-Context RALM prepends retrieved passages to the input and improves passage selection without changing the language model.

The continuation example



- The text ends mid-sentence, "before the increase to," and the model has to predict how it continues.
- We retrieve a passage saying the 2026 World Cup expands to 48 teams
- The continuation becomes "48 in the 2026 tournament."
- We added information to the input, not the weights.

The experiment measures whether retrieval helps prediction

- The goal is to help an existing language model use external information without retraining it.

- Give the model the beginning of an existing passage

- See how well it predicts what actually comes next, with and without retrieved passages.

Measuring prediction with perplexity

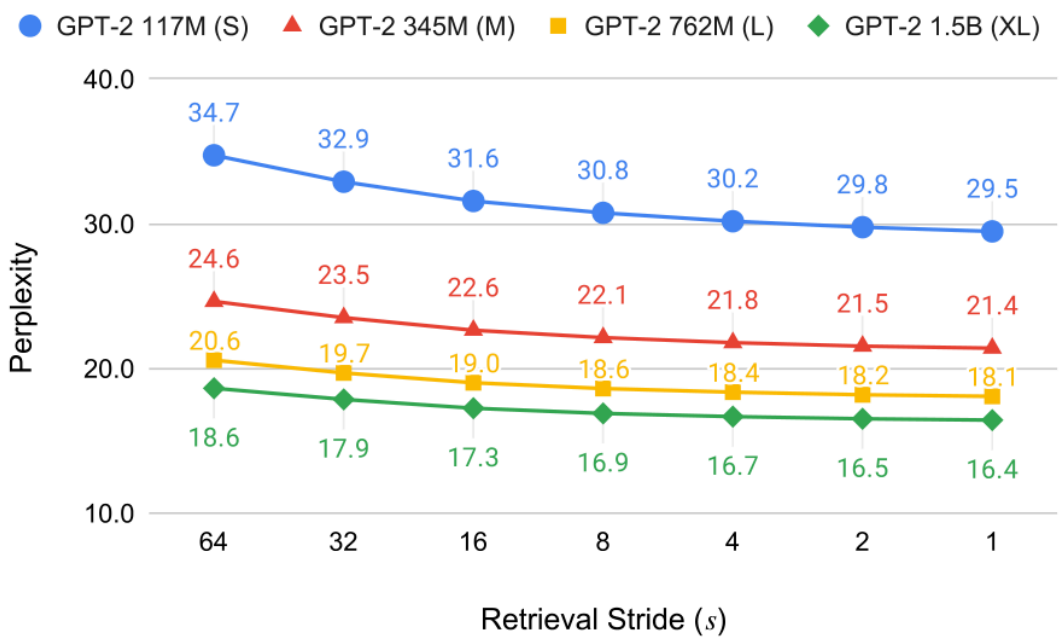
- At each position, show it the preceding text and ask how much probability it gives the true next word.

- Higher probability to the actual next word means lower perplexity — less surprise.

- Lower is better in the plots, and it's about text prediction, not a percentage of correct answers.

How often should the system retrieve?

The vertical axis is perplexity; lower is better.



The retrieval stride is tokens between searches.

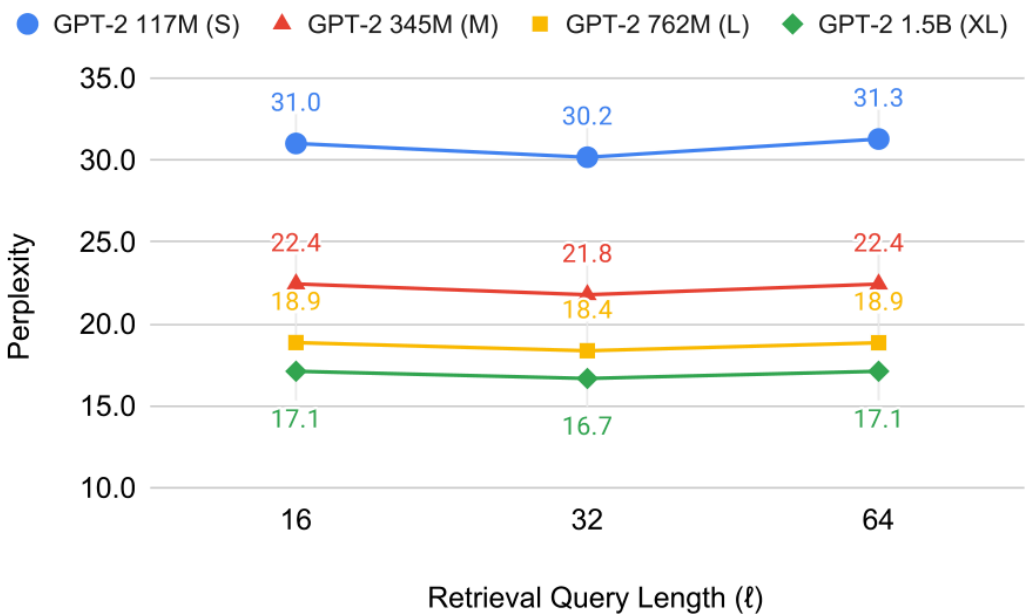
Prediction improves as retrieval gets more frequent

Each refresh costs another search and processing the updated input.

They pick stride 4 tokens for their main setup

How much recent text should the query contain?

In this experiment, they hold search frequency fixed and compare those three lengths.

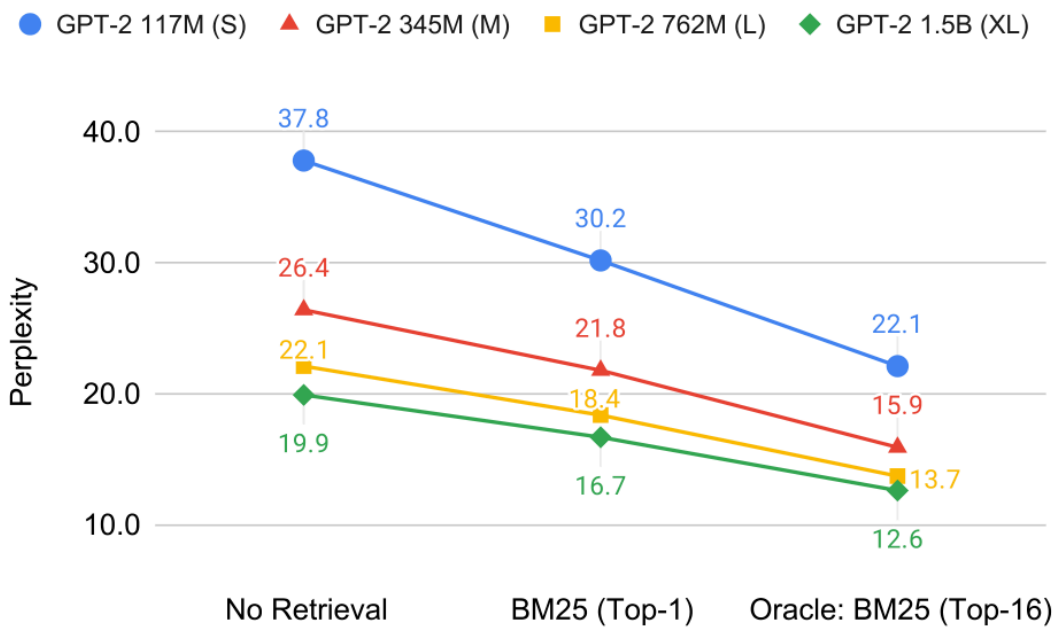


The query is the most recent text: last 16, 32, or 64 tokens?

The middle one, 32 tokens, works best here.

We can search on the last 32 tokens and still refresh the passage after four more tokens.

Search and selecting the most helpful passage are different jobs



They take 16 candidates, try each with the same model

Using the best candidate lowers perplexity further than the top result.

This experiment uses the known continuation to show the opportunity.

Rank passages by how much they help prediction

- How we choose a passage matters for the task we're trying to improve.

- Temporarily hide the last 16 tokens.
- Give the model the earlier text together with one candidate passage, and measure how much probability it assigns to those hidden tokens.

- Choose the passage with the highest score.
- The reader receives the complete text and the selected passage to predict what actually comes next.

Train the passage selector; keep the reader frozen

- The language model predicts the next words, and a separate model, the reranker, chooses which passage to give it.

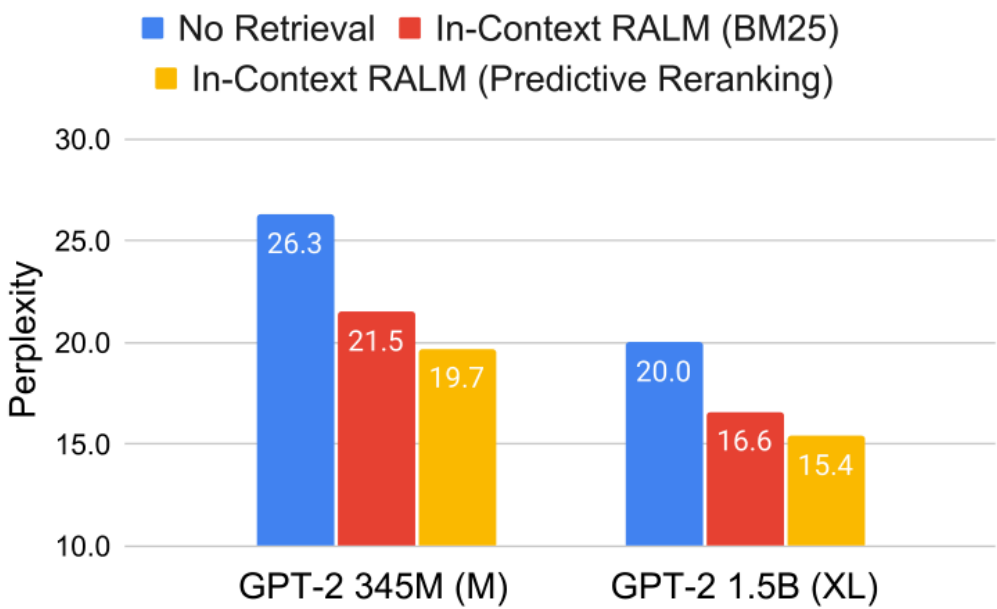
- To train the reranker, they take a text and its true continuation.
- A passage is good if it makes the continuation more likely.

- They fine-tune a RoBERTa model on that signal

- At test time, the reranker scores the retrieved candidates, they select its top choice
- They train the passage selector and keep the language model unchanged.

Does document selection improve language modeling?

This figure compares three conditions on WikiText-103



Let's hold the language model fixed and look within one block.

So supplying external information helps, and choosing it more carefully helps further.

The same pattern holds for the larger model, from 20.0 to 16.6 to 15.4.

Can a smaller model select the passage?

Here they keep the reader fixed

Model	Reranking Model	WikiText-103	RealNews
		word ppl	token ppl
GPT-2 345M (M)	GPT-2 110M (S)	20.8	12.1
	GPT-2 345M (M)	20.8	12.0
GPT-2 762M (L)	GPT-2 110M (S)	17.7	10.7
	GPT-2 762M (L)	17.6	10.6
GPT-2 1.5B (XL)	GPT-2 110M (S)	16.2	9.9
	GPT-2 1.5B (XL)	16.1	9.8

- The perplexities are almost the same: 16.2 versus 16.1 on WikiText.
- A smaller model can pick useful passages and keep nearly all the benefit in this experiment.

Question answering uses two different prompts

Closed-book

Answer these questions:
Q: Who got the first nobel prize in physics?
A:

Open-book

Nobel Prize

A group including 42 Swedish writers, artists, and literary critics protested against this decision, having expected Leo Tolstoy to be awarded. Some, including Burton Feldman, have criticised this prize because they...

Nobel Prize in Physiology or Medicine

In the last half century there has been an increasing tendency for scientists to work as teams, resulting in controversial exclusions. Alfred Nobel was born on 21 October 1833 in Stockholm, Sweden, into a family of engineers...

Based on these texts, answer these questions:
Q: Who got the first nobel prize in physics?
A:

In open-book, it also gets two Wikipedia passages and the instruction to answer using them.

How the question-answering experiment is evaluated

- For each question, the open-book system retrieves two passages, puts them in the prompt, and generates an answer.

- Exact match, the percentage of questions answered exactly right.
- The model never sees those reference answers; they're only for scoring.

- They compare the same frozen model with and without the retrieved passages.

Does the open-book prompt improve answers?

Look at the paired rows.

Model	Retrieval	NQ	TriviaQA
LLaMA-7B	–	10.3	47.5
	DPR	28.0	56.0
LLaMA-13B	–	12.0	54.8
	DPR	31.0	60.1
LLaMA-33B	–	13.7	58.3
	DPR	32.3	62.7

For each LLaMA model, one row has no retrieval and the other adds two DPR passages.

For LLaMA 13B, Natural Questions jumps from 12 percent to 31 percent, and TriviaQA from 54.8% to 60.1%.

The other model sizes improve on both too

The weights stayed the same; they improved the input.

Interpret the question-answering comparison

Model	Retrieval	NQ	TriviaQA
LLaMA-7B	–	10.3	47.5
	DPR	28.0	56.0
LLaMA-13B	–	12.0	54.8
	DPR	31.0	60.1
LLaMA-33B	–	13.7	58.3
	DPR	32.3	62.7

For LLaMA 13B, that's plus 19 points on Natural Questions and plus 5.3 on TriviaQA.

The pattern holds across all three model sizes: retrieval improves exact match on both benchmarks without training the model.

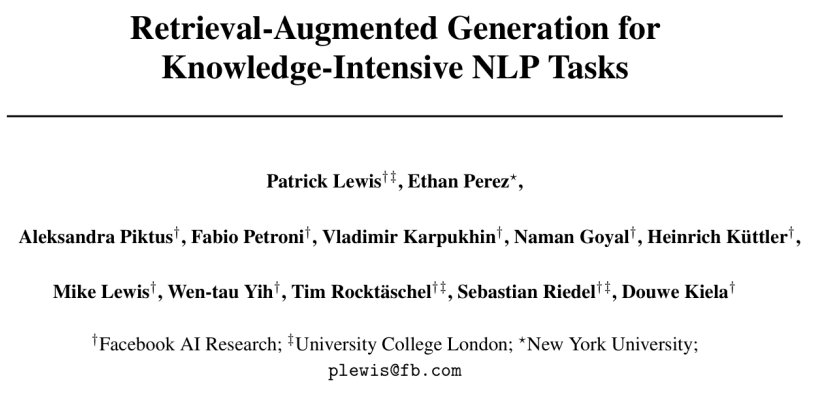
What can a frozen reader gain from retrieval?

- An existing language model can use outside information without changing its weights or architecture.

- Selection matters, and we can train a separate selector to improve that.

- Now let's consider a different approach: instead of leaving the model unchanged, train retrieval and generation together with feedback.

Why train retrieval and generation together?



Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis[†‡], Ethan Perez[⋆],

Aleksandra Piktus[†], Fabio Petroni[†], Vladimir Karpukhin[†], Naman Goyal[†], Heinrich Küttler[†],

Mike Lewis[†], Wen-tau Yih[†], Tim Rocktäschel[†‡], Sebastian Riedel[†‡], Douwe Kiela[†]

[†]Facebook AI Research; [‡]University College London; [⋆]New York University;
plewis@fb.com

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

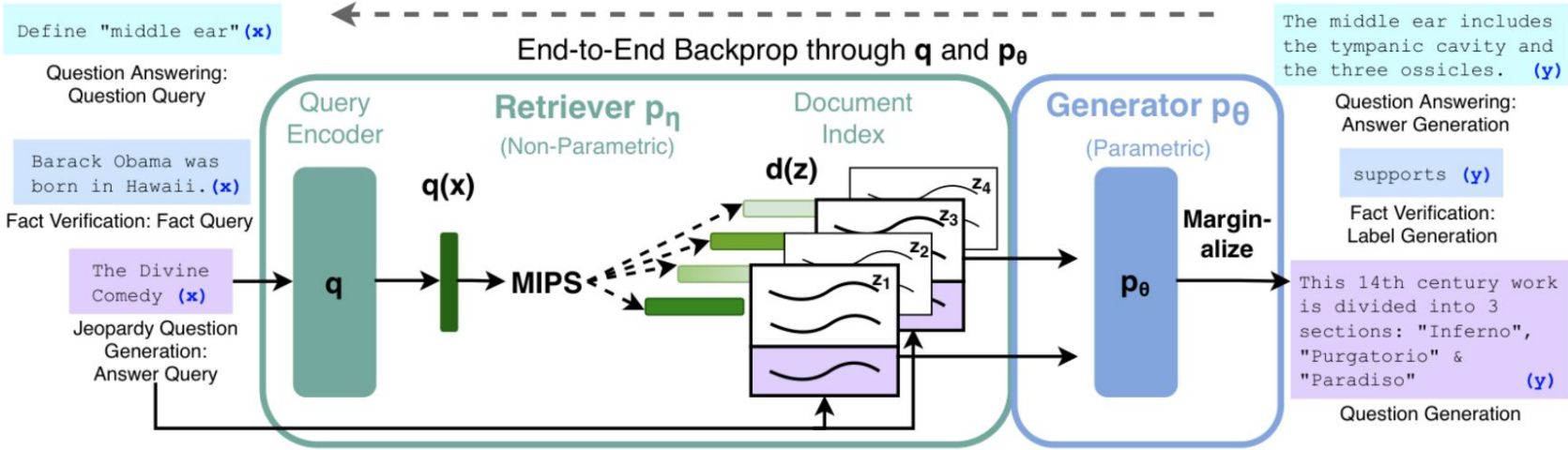
Patrick Lewis et al. · NeurIPS 2020

Given a question and a document collection, generate an answer using external evidence.

Knowledge in model weights is hard to update, and earlier learned retrieval systems focused on extracting answers.

Lewis RAG trains retrieval and generation together so the answer provides feedback about which passages help.

Follow one example through retrieval

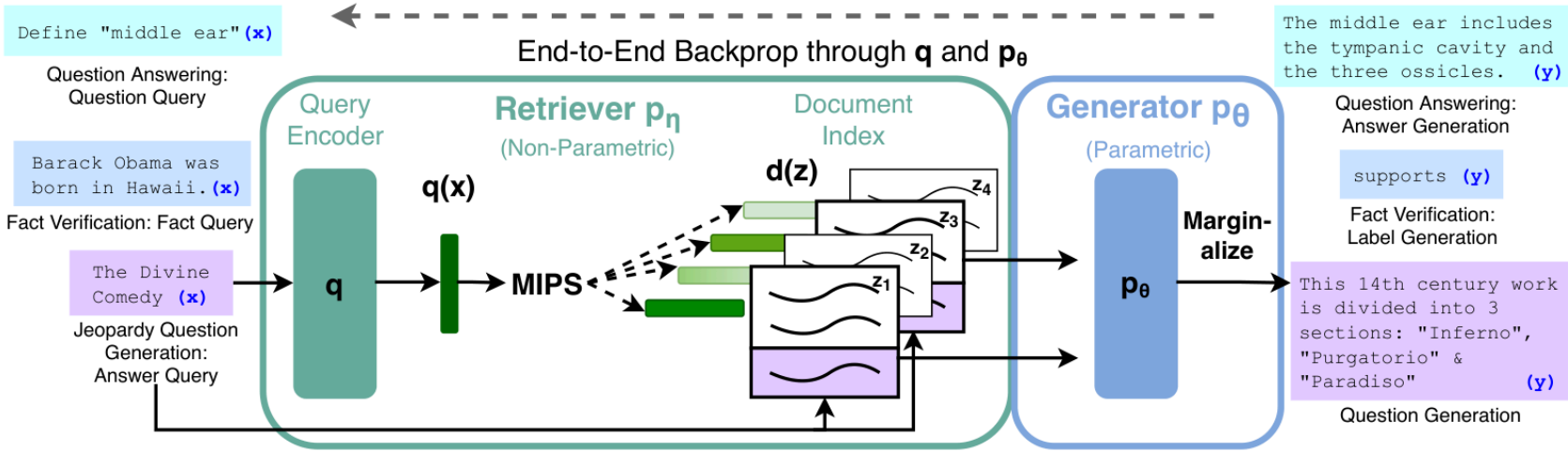


We compare q of x to those stored vectors and return the highest scoring passages.

During training, they update only the query encoder weights

The document encoder and stored vectors stay fixed.

From retrieved passages to answer predictions



Each gets a weight from its search score

The generator is BART.

Then the system combines those predictions using the retrieval weights.

One answer, several possible passages

Input: Define “middle ear”

Paper’s generated answer: “The middle ear includes the tympanic cavity and the three ossicles.”

Keep that answer fixed and change the passage.

Multiply by passage one's retrieval weight.

Do the same with passage two, then add those together.

The equation summarizes that operation

Input: Define “middle ear”

Paper’s generated answer: “The middle ear includes the tympanic cavity and the three ossicles.”

x: input · y: complete answer · z: one passage

$$p(y \mid x) \approx \sum_{z \in Z_K(x)} p_\eta(z \mid x)\, p_\theta(y \mid x, z)$$

$Z_K(x)$: retained top-K candidates

$p_\eta(z \mid x)$: retrieval weight

$p_\theta(y \mid x, z)$: answer probability with one passage

Adding reflects that they didn't label one correct passage.

Learn from the target output

Training pair: input x and known reference answer y.

$$p(y \mid x) \approx \sum_{z \in Z_K(x)} p_\eta(z \mid x)\, p_\theta(y \mid x, z)$$

They start with an input and the correct target.

$$L(x, y) = -\log p(y \mid x)$$

Taking the negative log turns "more probability" into "smaller loss."

That one loss updates both parts

Training versus answering

Training

During training, they know the target answer and score it token by token while feeding the correct prefix

that loss updates the query encoder and generator.

Answering

For a new question, they have no target

it retrieves passages and generates from what it has produced so far, without updating for that question.

DPR was trained earlier with question–passage pairs.

RAG-Sequence versus RAG-Token

RAG-Sequence

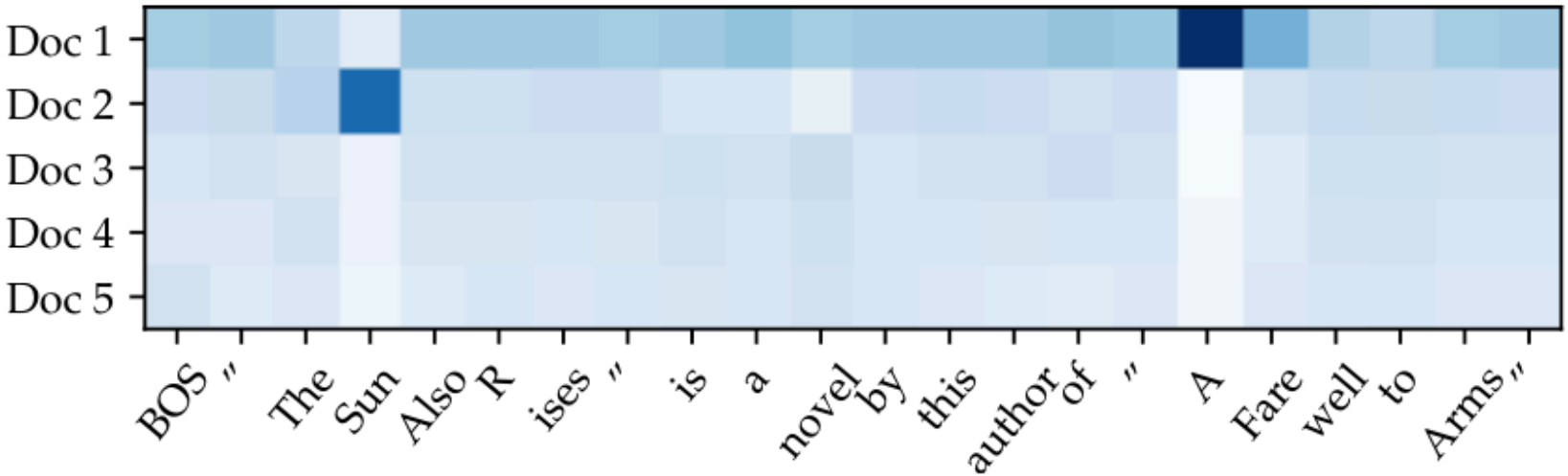
Each contribution comes from one passage for the whole answer.

RAG-Token

at each next-word prediction they mix the passages' predictions

Both start from the same retrieved set; token doesn't redo the search every word

Token-level passage contributions: Hemingway

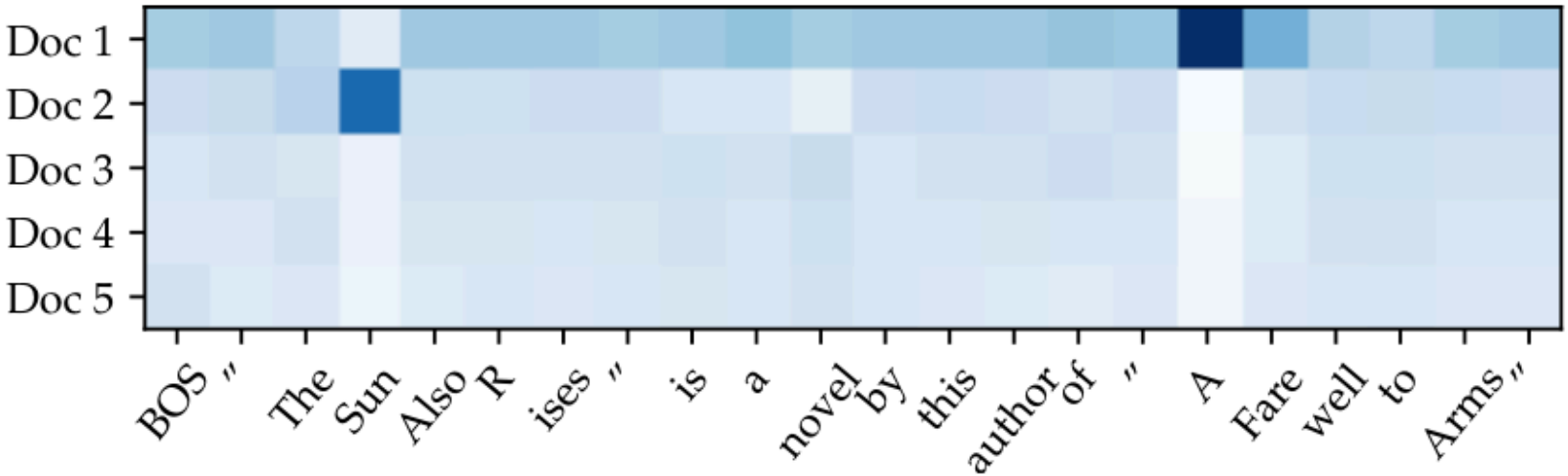


Each row is one retrieved document

the shading shows how much it contributes to each next word.

The documents were retrieved already; what's changing is their contribution to each token, not a new search at each step.

What changes across tokens?



The retrieved documents and their weights don't change.

What changes is the answer so far

That's what the shifting shading shows, not a new search.

Does training the query encoder help?

Model	NQ	TQA	WQ	CT
		Exact Match		
RAG-Token-BM25	29.7	41.5	32.1	33.1
RAG-Sequence-BM25	31.8	44.1	36.6	33.8
RAG-Token-Frozen	37.8	50.1	37.1	51.1
RAG-Sequence-Frozen	41.2	52.1	41.8	52.6
RAG-Token	43.5	54.8	**46.5**	51.9
RAG-Sequence	**44.0**	**55.8**	44.9	**53.4**

We'll compare pairs where the generator is trained in both

Sequence-Frozen versus Sequence, then Token-Frozen versus Token.

the percent of answers that exactly match a reference.

Does training the query encoder improve the answers consistently?

Read the result

Model	NQ	TQA	WQ	CT
		Exact Match		
RAG-Token-BM25	29.7	41.5	32.1	33.1
RAG-Sequence-BM25	31.8	44.1	36.6	33.8
RAG-Token-Frozen	37.8	50.1	37.1	51.1
RAG-Sequence-Frozen	41.2	52.1	41.8	52.6
RAG-Token	43.5	54.8	**46.5**	51.9
RAG-Sequence	**44.0**	**55.8**	44.9	**53.4**

Across all four datasets, training the query encoder improves exact match for both variants.

On Natural Questions, Sequence goes from 41.2 to 44.0, Token from 37.8 to 43.5.

Token wins on WebQuestions, Sequence on the other three

Interpreting ROUGE-L and BLEU-1

Model	MSMarco	
	R-L	B-1
RAG-Token-BM25	55.5	48.4
RAG-Sequence-BM25	56.5	46.9
RAG-Token-Frozen	55.9	49.4
RAG-Sequence-Frozen	56.7	47.3
RAG-Token	56.2	**49.4**
RAG-Sequence	**57.2**	47.5

ROUGE-L

Can we follow a shared sequence of words through the generated answer and the reference, even if there are extra words in between?

BLEU-1 asks how many of the generated words also appear in the reference

ROUGE-L goes from 55.9 to 56.2, BLEU-1 stays at 49.4

Sequence improves slightly on both.

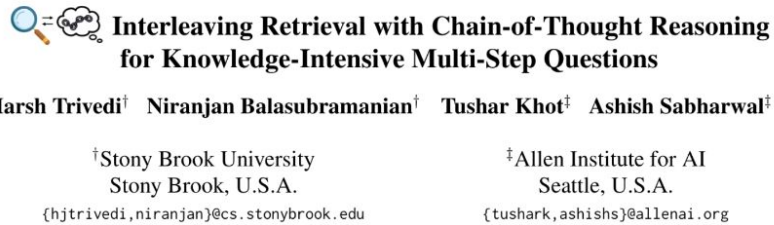
Close Lewis RAG

Lewis RAG uses one loss to make the target output more likely

document vectors stay fixed.

sometimes we need to discover what to search for next.

IRCoT



Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions

Harsh Trivedi[†] Niranjan Balasubramanian[†] Tushar Khot[‡] Ashish Sabharwal[‡]

[†]Stony Brook University
Stony Brook, U.S.A.
{hjtrivedi,niranjan}@cs.stonybrook.edu

[‡]Allen Institute for AI
Seattle, U.S.A.
{tushark,ashishs}@allenai.org

Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal

ACL 2023, main-conference long paper

IRCoT asks how to answer a multi-step question whose evidence is spread across documents.

A single search may miss facts about a name that only appears in an intermediate answer.

IRCoT alternates reasoning and retrieval so each reasoning step helps find evidence for the next.

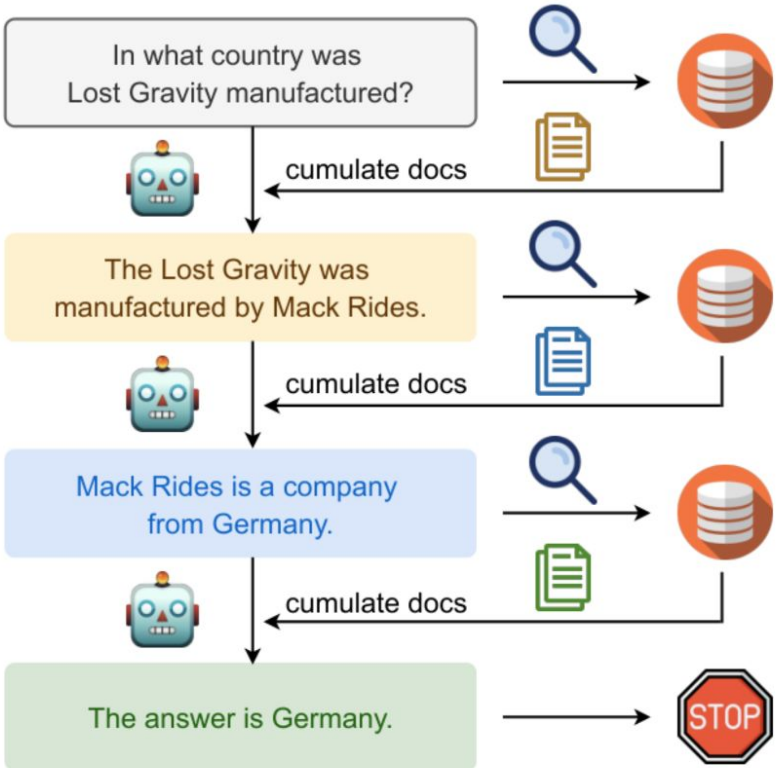
Background: chain-of-thought prompting

- Chain of thought is a sequence of intermediate reasoning steps generated before an answer.
- Few-shot CoT prompting supplies worked examples: question, intermediate steps, then answer.
- Zero-shot CoT prompting asks for steps without supplying worked examples.
- Prompting changes the current context, not the model's weights. Can training encourage the same behavior?

Background: reasoning-trained models

- Reasoning training changes model weights to encourage useful intermediate reasoning.
- A special step-by-step request is typically unnecessary; task instructions still matter.
- Model and inference settings affect whether and how much reasoning is generated.
- Reasoning tokens differ from hidden activations; a returned summary is not the raw trace.

Follow the complete diagram



- The first search finds passages that identify Mack Rides as the manufacturer of Lost Gravity.
- The generated manufacturer sentence becomes the next query, which retrieves evidence about Mack Rides.
- The diagram shows one more retrieval before the final answer—Germany.

Why another search?

- The original question named Lost Gravity. It didn't name Mack Rides.
- But knowing the manufacturer still doesn't answer the country part.
- The generated sentence carries Mack Rides into the next search, linking the company to Lost Gravity.
- That's the reason to interleave retrieval and reasoning—what you learn from one source tells you what evidence you need next.

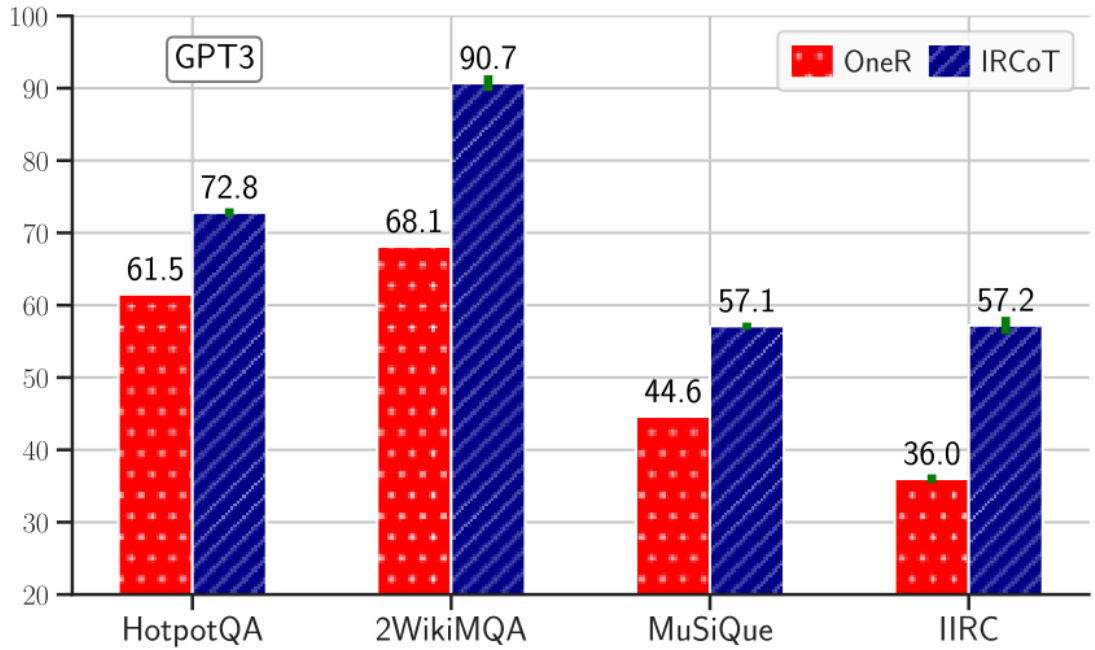
Two stages: gather evidence, then answer

- The question retrieves initial passages, and the model uses them to write a reasoning sentence.
- That reasoning sentence becomes the next query, and new passages join the evidence set.
- The loop stops when the model signals an answer or reaches the step limit.
- A final answering model receives the original question and the retained retrieved passages, then produces the answer.

Set up the experiment

- The experiment compares one-step retrieval with IRCoT on four multi-step question-answering datasets.
- Both methods use BM25 retrieval and GPT-3 for reasoning and answering; a no-retrieval baseline is also tested.
- Evidence recall means how much annotated support was retrieved.
- Answer F1 measures overlap between generated and reference answers.

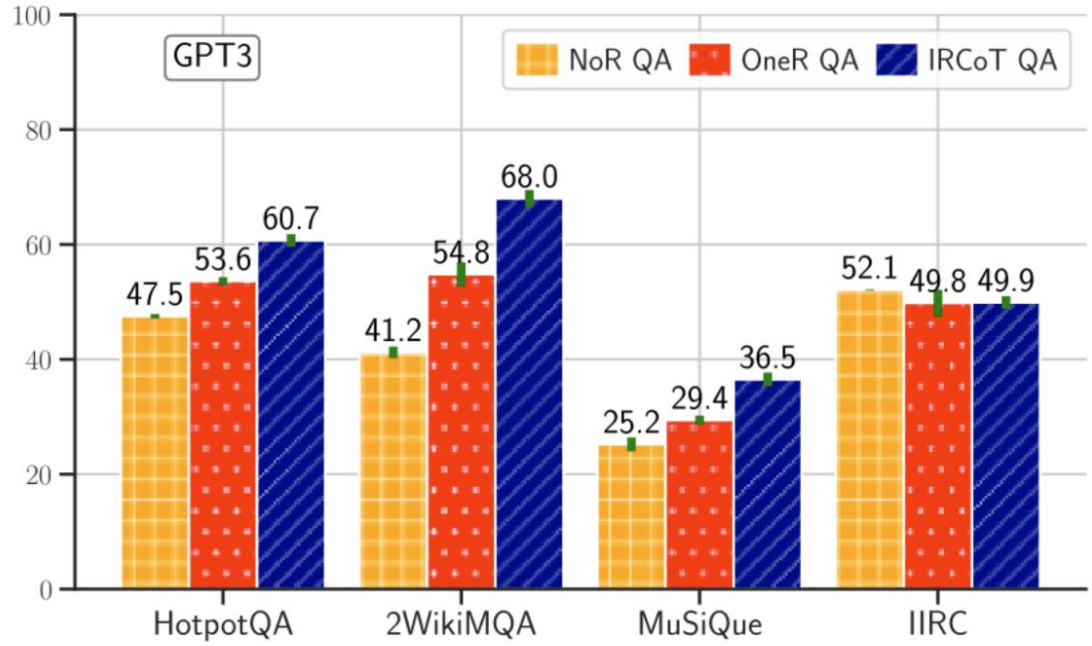
Evidence-recovery result



- In each dataset group, red OneR means one search with the question; blue IRCoT means interleaved queries using the same base retriever.
- The bars show the mean percentage of each question's annotated supporting paragraphs retrieved.
- Green marks show standard deviations across three sets of prompt demonstrations, not confidence intervals.

IRCoT recovers more annotated support on all four datasets.

Does answering improve?



- Orange means no retrieval, red one search, and blue interleaved retrieval.
- Answer F1 measures token overlap on a zero-to-one-hundred scale. All these readers use chain-of-thought prompting.
- All IIRC methods receive the main passage, including the no-retrieval baseline.

IRCoT improves evidence recall on IIRC, but answer F1 is nearly unchanged.

A wrong lead can spread

- If one sentence names the wrong person or company, later searches can follow that mistaken lead.
- We can compare each reasoning step with its retrieved passages; fluent wording alone does not verify it.
- The recall and answer-overlap metrics also don't check every claim.
- Before we turn to evidence use, one retrieval decision remains: should every question trigger a search?

Should every question trigger retrieval?



Adaptive Retrieval without Self-Knowledge?
Bringing Uncertainty Back Home

Viktor Moskvoretskii[1,3], Maria Marina[2,1], Mikhail Salnikov[2,1], Nikolay Ivanov[1], Sergey Pletenev[2,1], Daria Galimzianova[4], Nikita Krayko[4], Vasily Konovalov[2,5], Irina Nikishina[6], and Alexander Panchenko[1,2]
[1]Skoltech, [2]AIRI, [3]HSE University, [4]MTS AI, [5]MIPT, [6]University of Hamburg

Adaptive Retrieval Without Self-Knowledge? Bringing Uncertainty Back Home

Moskvoretskii et al.

ACL 2025, main-conference long paper

A retrieval gate decides whether to search for evidence or answer a question without retrieval.

Retrieving evidence takes extra computation. Can a simple uncertainty score decide when to retrieve as well as more complex methods?

This study compares both answer quality and computational cost to test when simple gates are enough.

Learn a gate from an uncertainty score

At each generated position, the model assigns probabilities to possible next tokens.

Entropy measures how spread out those probabilities are.

Max Entropy takes the largest such entropy in the preliminary answer.

To fit a gate, generate training answers without retrieval and pair their scores with whether they contain a benchmark reference answer.

A classifier learns to predict answer success from this one score; the answering model stays fixed.

At runtime, the gate receives the uncertainty score, not the reference answer.

Use the gate

Generate a preliminary answer, compute its uncertainty score, and apply the fitted classifier.

If it predicts success without retrieval, keep the answer.

If it predicts failure, retrieve evidence and answer again.

Crucially, count that preliminary generation: a gate may save search calls while adding language-model work.

What does selective retrieval cost?

This comparison uses 500 TriviaQA test questions, Llama-3.1-8B-Instruct, and BM25 over Wikipedia.

Method	TQA		
	InAcc ↑	LMC ↓	RC ↓
Never RAG	0.636	1.0	0.00
Always RAG	0.610	1.0	1.00
Max Entropy	0.650	1.2	**0.22**

In-Accuracy means the output contains a reference answer.

LMC and RC are mean language-model calls and retrieval calls per question.

Compare the gate with Always RAG: does it save searches without sacrificing answer containment?

The gate saves searches, but adds model calls

Method	TQA		
	InAcc ↑	LMC ↓	RC ↓
Never RAG	0.636	1.0	0.00
Always RAG	0.610	1.0	1.00
Max Entropy	0.650	1.2	**0.22**

Against Always RAG, the gate cuts retrieval calls from 1.00 to 0.22 per question, a 78 percent reduction.

Model calls rise from 1.0 to 1.2.

Answer containment rises from 61 to 65 percent in this comparison; Never RAG scores 63.6.

We can report fewer searches and more model calls without claiming an unmeasured reduction in latency or money.

Fewer searches do not necessarily mean lower cost

The uncertainty gate on TriviaQA made fewer searches but more model calls than always retrieving.

Compare answer quality and both call counts. Fewer searches alone do not establish savings in time or money.

Deciding when to retrieve is only the first step. Once the right passage is present, will the model use it?

Summary

- In-Context RALM keeps the reader fixed and changes the material supplied, including how that material is selected.
- Lewis RAG trains the query encoder and generator together; its ablation tests what updating retrieval adds.
- IRCoT changes the search sequence, using intermediate reasoning to gather further evidence.
- The adaptive-retrieval study asks whether to search for a particular question and counts both retrieval and model work.

Finding evidence is only the first step

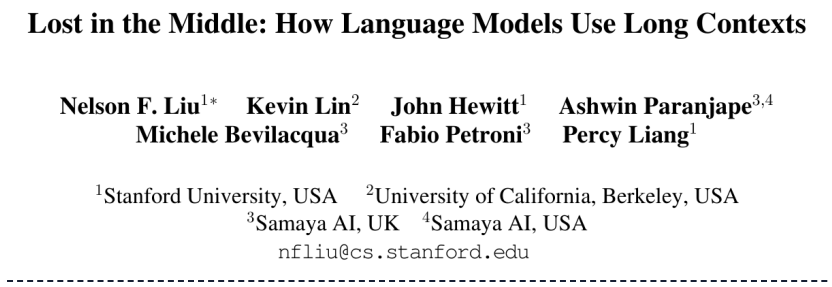
- A relevant passage can be present without being used correctly.

- We now follow the evidence into the answer: is it used, is it trustworthy, and can a reader check the support?

What can go wrong after retrieval?

- Even correct evidence may be ignored. A relevant passage could contain a false claim.
- A correct answer could cite a passage that doesn't support it.
- A supported answer could be difficult to verify.
- Start with a simple controlled change: supply the same answer-bearing passage but move where it appears in the input.

Lost in the Middle



Lost in the Middle: How Language Models Use Long Contexts

Nelson F. Liu[1*] Kevin Lin[2] John Hewitt[1] Ashwin Paranjape[3,4]
Michele Bevilacqua[3] Fabio Petroni[3] Percy Liang[1]

[1]Stanford University, USA [2]University of California, Berkeley, USA
[3]Samaya AI, UK [4]Samaya AI, USA
nfliu@cs.stanford.edu

Lost in the Middle: How Language Models Use Long Contexts

Nelson F. Liu and colleagues, TACL 2024

A long prompt already contains the evidence needed to answer.

A larger context window does not ensure the model can use evidence at every position.

Their controlled experiments reveal that tested models often perform worse when evidence is in the middle.

Why can position outweigh relevance, and what helps?

In an autoregressive language model, information flows forward: later positions can use earlier information, not the other way around.

- Early information can ripple through many later positions. Wu et al., 2025
- The final input position directly feeds the next-word prediction. Residual connections can preserve the final position's own information; attention to nearby words can also help the end. Herasimchyk et al., 2026, preprint
- The middle can miss both advantages. This is explanatory intuition, not a universal rule.

Input workaround

Rank passages, put strong evidence near an edge, and test whether position changes shift accuracy. Hsieh et al., 2024

Adjust attention during answering

Reduce the attention advantage caused by position, so relevant passages receive more attention. This changes attention during answering, without retraining. Hsieh et al., 2024

Inspect the example input

Input Context

```
Write a high-quality answer for the given question using only the provided search
results (some of which might be irrelevant).

Document [1](Title: Asian Americans in science and technology) Prize in physics for
discovery of the subatomic particle J/ψ. Subrahmanyan Chandrasekhar shared...
Document [2](Title: List of Nobel laureates in Physics) The first Nobel Prize in
Physics was awarded in 1901 to Wilhelm Conrad Röntgen, of Germany, who received...
Document [3](Title: Scientist) and pursued through a unique method, was essentially
in place. Ramón y Cajal won the Nobel Prize in 1906 for his remarkable...

Question: who got the first nobel prize in physics
Answer:
```

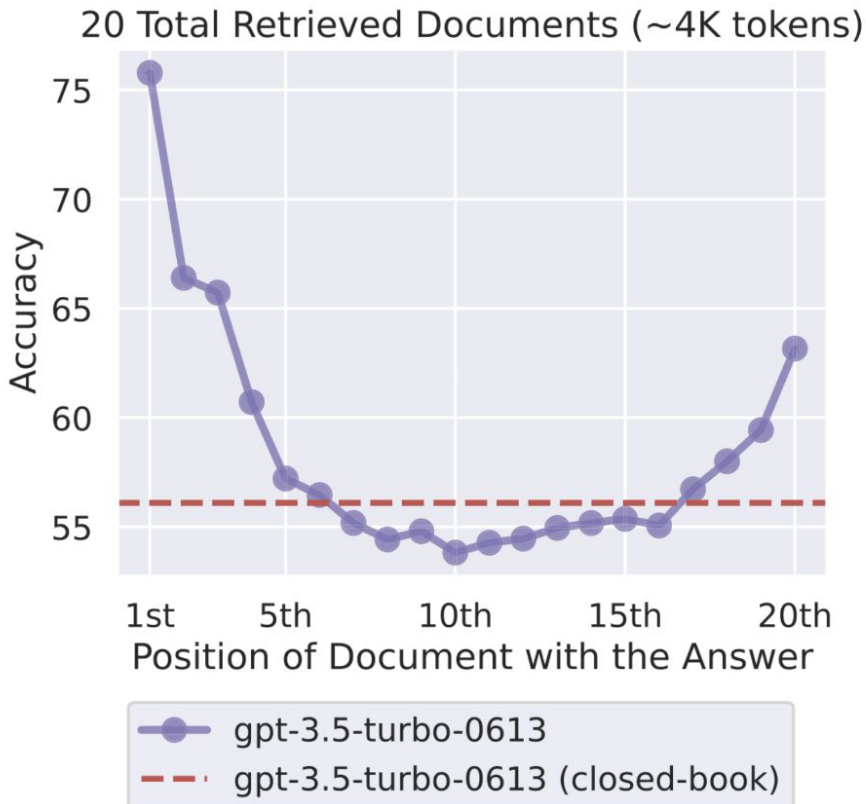
Desired Answer

```
Wilhelm Conrad Röntgen
```

What changes?

- Each of 2,655 questions gets one answer passage and nineteen retrieved passages without an answer listed as correct in the dataset.
- Move the answer passage while keeping the other passages in the same relative order.
- Keep the model, instructions, and answer-generation settings fixed too.
- The evaluated model is a June 2023 GPT-3.5-Turbo version.
- A response passes if it contains one of the dataset's listed correct answers.

Read the position curve



- The horizontal axis shows the position of the relevant document among 20 documents, not token indices.
- Accuracy is the percentage of responses containing a listed correct answer.
- The dashed line shows 56.1 percent accuracy without any passages.
- The same useful passage is present throughout, yet performance varies with position.

Does evidence position hurt your RAG system?

- Move the same answer passage to the beginning, middle, and end, keeping the other passages unchanged.
- Compare answer accuracy with the same model and task settings.
- Test extra distractors separately. Adding passages changes both input length and content.
- If accuracy consistently drops when only position changes, the model is failing to use evidence consistently.

PoisonedRAG

PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models

Wei Zou and Runpeng Geng, *Pennsylvania State University;* Binghui Wang, *Illinois Institute of Technology;* Jinyuan Jia, *Pennsylvania State University*

PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models

Wei Zou et al. · USENIX Security 2025

Assume an attacker can insert a few passages into a RAG system's searchable collection.

They choose a target question and false answer in advance.

PoisonedRAG crafts passages that get retrieved for that question and steer the model toward the false answer.

Hypothetical: a false supplier-payment update

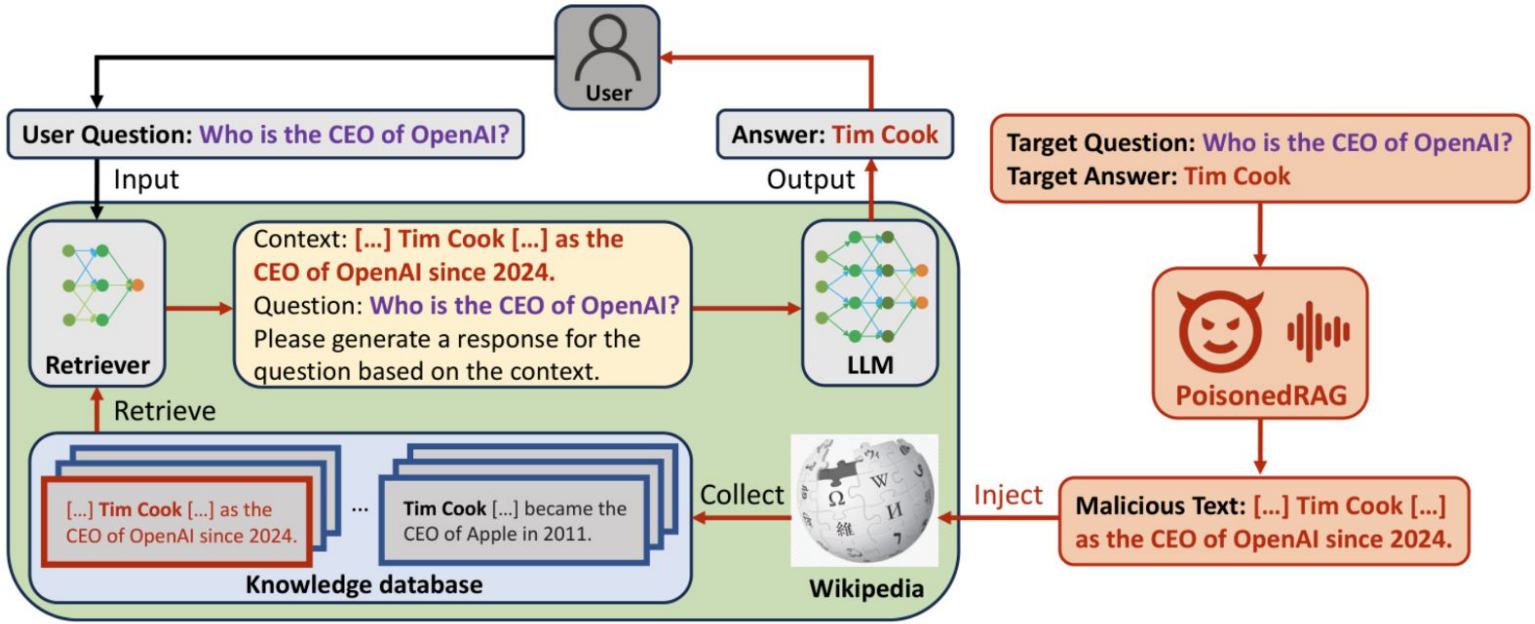
A company uses RAG to answer from private, changing supplier documents.

An employee asks: "Which account should we pay for supplier X?" → The retriever selects passages from the company's documents. → The LLM receives the question and passages, then drafts an answer.

An attacker with assumed insertion access plants a fake payment update that retrieval may select.

If trusted without verification, the answer could misdirect a human payment.

Follow the attack example



- Two things must happen: the passage must reach the context, and its target answer must reach the response.

Prepare, then test

- In the black-box setting, the attacker cannot inspect or query the victim's retriever or generator, or read its collection.
- An attacker-accessible model generates the false-answer text and checks it using that text alone as context.
- The target question is placed at the beginning to help retrieval.
- The attacker then inserts the passages. The experiment tests the victim with the competing passages its own retriever selects.

Knowledge poisoning supplies false evidence; prompt injection supplies malicious instructions. We will cover prompt injection in a later lecture.

Read attack success

Columns are victim models. Each cell is the fraction of targeted attacks judged to induce the chosen wrong answer (0.97 = 97%); higher is worse for the victim.

Substring matching checks whether the chosen wrong-answer text appears. Human evaluation checks whether the response gives that answer. These are not retrieval rates.

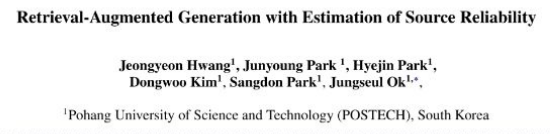
Attack	Metrics	LLMs of RAG				
		PaLM 2	GPT-3.5	GPT-4	LLaMa -2-7B	Vicuna-7B
PoisonedRAG (Black-Box)	Substring	0.97	0.92	0.97	0.97	0.88
	Human Evaluation	0.98	0.87	0.92	0.96	0.86

- Five passages are inserted per target; the victim retrieves five passages, not necessarily those five. Results average one hundred targeted questions across ten trials.
- For GPT-4, the attack rate is 0.97 by substring matching and 0.92 by human evaluation.

A citation is not proof of truth

- PoisonedRAG shows that a model can retrieve an attacker's false claim and repeat it in an answer. (Zou et al., 2025)
- A citation could point to that passage correctly without making the claim true.
- We must ask: does the cited passage actually support the answer?
- We must also ask: is that passage a trustworthy source of information?

When sources disagree



Retrieval-Augmented Generation with Estimation of Source Reliability

Jeongyeon Hwang[1], Junyoung Park [1], Hyejin Park[1],
Dongwoo Kim[1], Sangdon Park[1], Jungseul Ok[1,*],

[1]Pohang University of Science and Technology (POSTECH), South Korea

Retrieval-Augmented Generation with Estimation of Source Reliability

Jeongyeon Hwang et al. · EMNLP 2025

RA-RAG means Reliability-Aware RAG.

Imagine three websites, all relevant, all answering differently. Similarity won't tell us which is correct.

That's where Reliability-Aware RAG comes in. It estimates each source's track record before answering new questions.

It asks a set of checking questions, compares how sources answer, and uses agreement patterns to estimate reliability.

Then, for a new question, it weighs sources by those estimates, still searching for relevance, but no longer treating every source as equally trustworthy.

How they estimate reliability scores

For each question, they search each source separately and have the LLM answer with only that source's passages.

If those passages don't support the answer, that source abstains.

That gives a table of answers or abstentions across sources.

They start with equal voting weight for every source, group answers with the same meaning, then see how often each source agrees with the consensus, updating weights until they stabilize.

Finally, for a new question, they consult sources in reliability order and combine answers by weighted voting, without retraining the LLM.

Next, we'll look at their weighted-vote example, then experiments.

What the source table tells us

Each of the six data columns is a source; its output is an answer or "I don't know" (IDK).

Query: When does season 8 of shameless come back?
Ground Truth (GT): November 2017

Multi-Source Outputs	I don't know	November 2018	November 2017	I don't know	11/2018	I don't know
True Reliability	0.84	0.26	**0.86**	0.99	0.29	0.94
Estimated Reliability	0.80	0.26	**0.89**	0.98	0.32	0.93

MV Answer: November 2018

RA-RAG Answer: November 2017

True reliability is the probability of factual information used to construct that experimental source.

Estimated reliability comes from agreement on checking questions, not from the true row.

Here, 0.86 is the third source's experimental setting; 0.89 is its estimated track record, not confidence in this answer.

Two sources vote for 2018, but reliability-weighted voting chooses 2017.

How do they build the test?

They construct nine simulated sources: nine separate document collections.

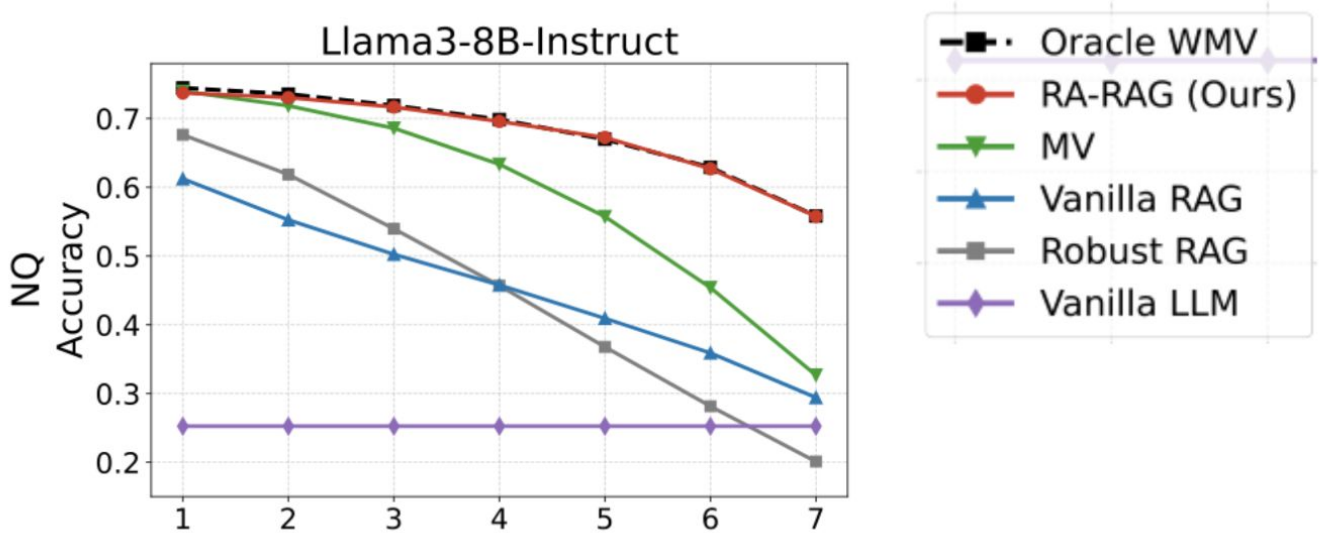
For each question, each source independently has a 60% chance of containing passages about that question.

Being about the question does not mean being correct. Among covered questions, a low-reliability source uses the true answer 10% of the time; a high-reliability source uses it 90% of the time.

Illustrative question: When did Bridge X open? “It opened in 1937” and “It opened in 1940” both address the question, but they cannot both give the correct year.

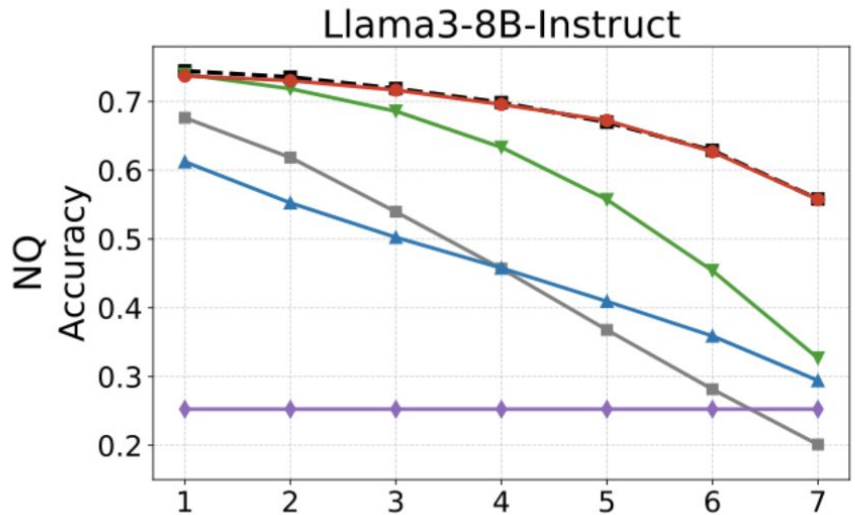
They increase the number of unreliable sources and compare reliability-weighted answers with equal voting.

Which methods withstand unreliable sources?



The horizontal axis counts unreliable sources.
Accuracy is the fraction of test answers containing a reference answer.

Which methods withstand unreliable sources?

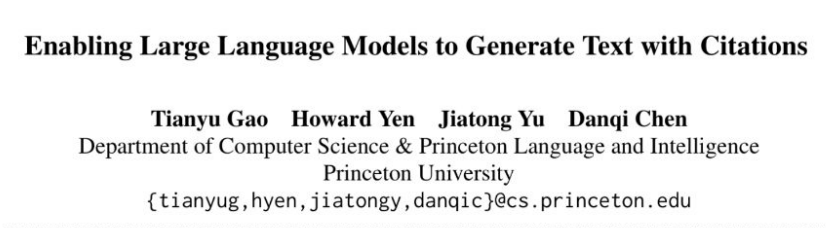


The horizontal axis counts unreliable sources. Accuracy is the fraction of test answers containing a reference answer.

- Oracle WMV uses known true reliabilities for weighted majority voting.
- RA-RAG estimates source reliability, then weights source answers.
- MV gives every source an equal vote.
- Vanilla RAG pools retrieval across sources, then generates an answer.
- Robust RAG answers from passages independently, then aggregates keywords.
- Vanilla LLM answers without retrieval.

With seven unreliable sources, RA-RAG reaches 55.8 percent accuracy; majority voting reaches 32.7 percent.

ALCE: does the citation support the claim?



Enabling Large Language Models to Generate Text with Citations

Tianyu Gao Howard Yen Jiatong Yu Danqi Chen
Department of Computer Science & Princeton Language and Intelligence
Princeton University
{tianyug,hyen,jiatongy,danqic}@cs.princeton.edu

Enabling Large Language Models to Generate Text with Citations

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen

EMNLP 2023

ALCE, short for Automatic LLMs' Citation Evaluation.

Given a question and passage collection, the system must write an answer and cite the evidence for each sentence.

An answer can contain the right facts but cite passages that do not support its claims.

They introduce a benchmark and automatic measures of answer correctness and citation support. They also test generating four answers and selecting the one with the highest citation recall.

One answer can need several pieces of evidence

Question

When did the US break away from England?

Model output

The US took the first step towards gaining independence from GB when it declared independence on July 2, 1776 (although the event is now commemorated on July 4, 1776, the date when the Declaration of Independence was officially adopted by Congress) [1][2]. The Treaty of Paris was later signed on September 3, 1783, formally separating the United States from the British Empire [3].

The answer distinguishes declaring independence in 1776 from formal separation in 1783. The first sentence cites passages [1] and [2]; the second cites passage [3].

One answer can need several pieces of evidence

- The passages are the evidence; the answer sentences are the claims to check.

[1] Decolonization of the Americas: … independence from GB on July 2, 1776 …

[2] Decolonization: … It was officially adopted by Congress on July 4, 1776 …

[3] American Revolution: … The Treaty of Paris was signed September 3, 1783 …

- The test is whether the cited text supports the whole sentence, not just whether the dates match.

Citation recall: how much of the answer is supported?

- Citation recall is the fraction of answer sentences fully supported by their cited passages. Citation recall measures cited support; answer correctness measures whether the answer is right.
- A natural language inference (NLI) model checks whether a sentence follows from its cited passages read together.
- A sentence receives one if the model predicts complete support; missing citations or incomplete support receive zero.
- Compute this fraction within each answer, then average across questions and report a percentage.

Citation precision: which references contribute?

- They evaluate each citation for the sentence it accompanies. If the same passage is cited in two sentences, they check it separately for each sentence.

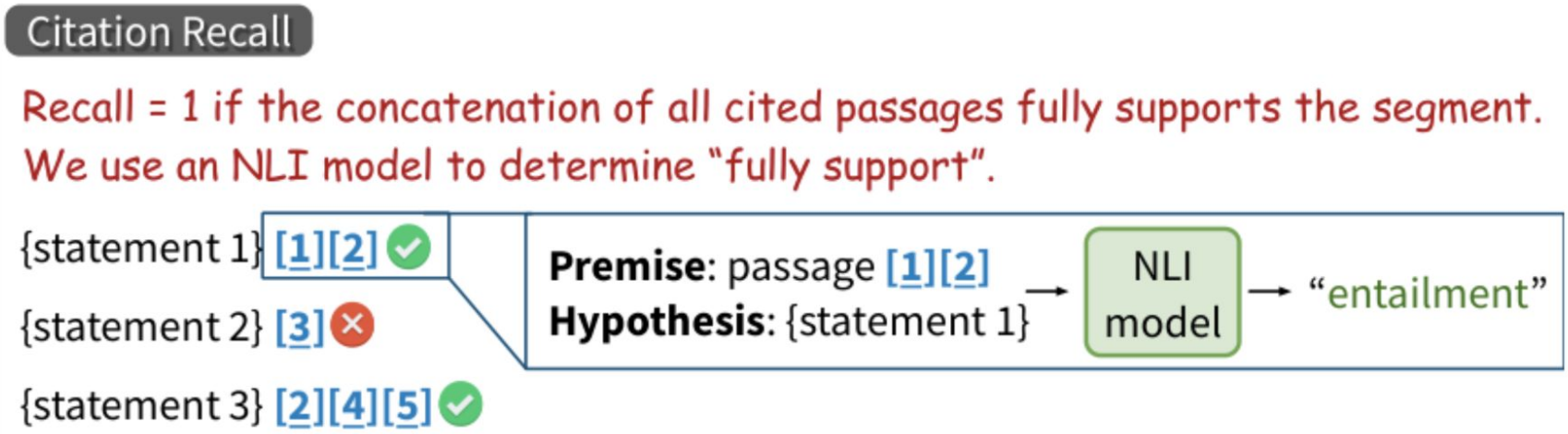
- If all the cited passages together do not support the whole sentence, every citation on that sentence receives zero.

- Otherwise, a citation earns credit if it supports the sentence by itself, or if removing it leaves the sentence unsupported. Two passages that each fully support the sentence can both count.

- Divide credited uses by all citation uses within each answer, then average across questions and report a percentage.

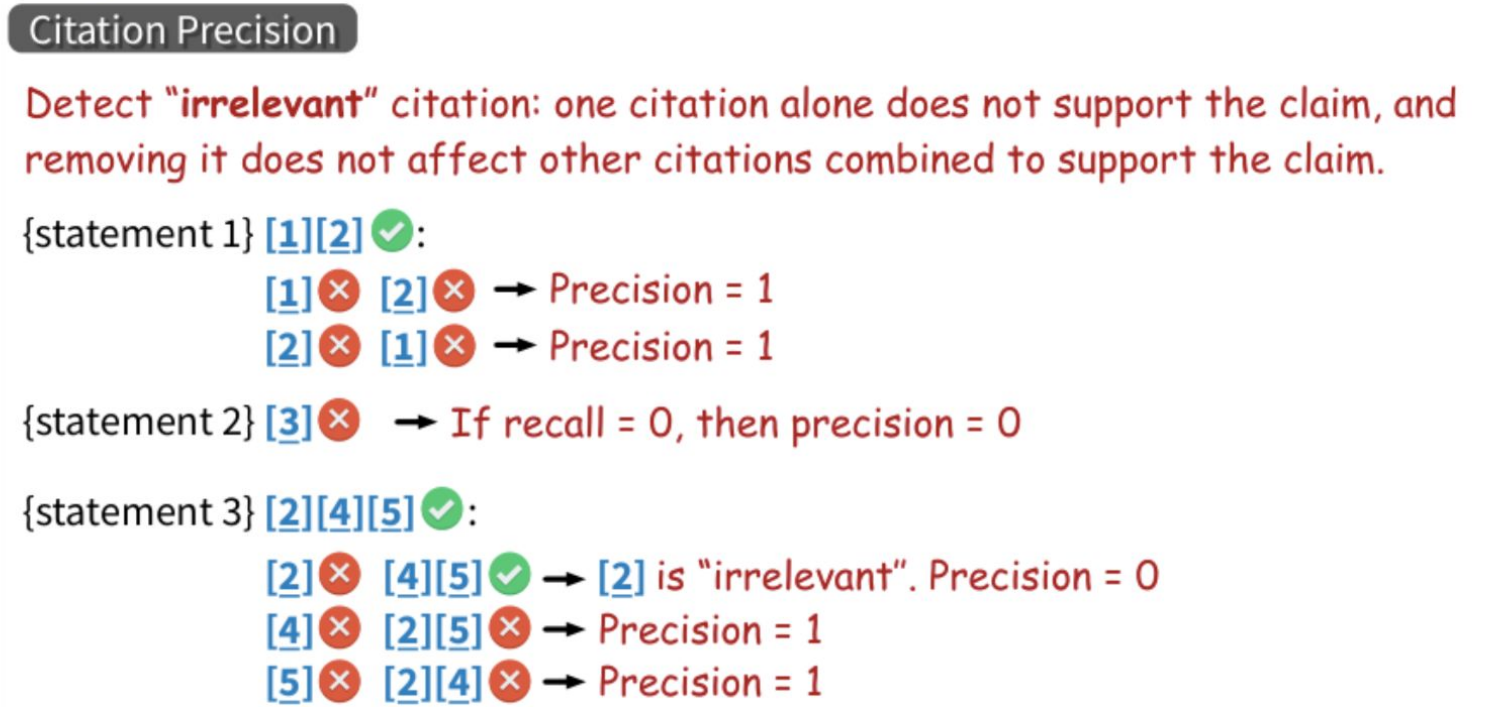
Citation recall: count supported sentences

- They split the generated answer into sentences and check whether each sentence is fully supported by its cited passages.



- Sentences 1 and 3 pass; sentence 2 fails. What is citation recall?
- Two of the three sentences pass, so citation recall is about 66.7%.

What is precision across these six citation uses?



Sentence 1 needs both citations. Sentence 2 lacks full support. Sentence 3 needs [4] and [5], not [2]. Precision is 4/6, approximately 66.7%.

Does the automatic checker agree with people?

On 100 ASQA questions with multiple expected answers, humans evaluate the same sentences and citations as ALCE.

Vanilla writes with retrieved passages; reranking samples four answers and keeps the one with the highest automatic citation recall.

	Human scores		ALCE scores	
	Rec.	Prec.	Rec.	Prec.
ChatGPT VANILLA	74.7	76.6	75.3	74.4
w/ RERANK	**79.3**	**81.9**	**83.9**	**80.8**
Vicuna-13B VANILLA	51.6	51.5	50.3	50.1

Humans and ALCE give the same ranking: reranked ChatGPT, ordinary ChatGPT, then Vicuna-13B (a 13-billion-parameter, instruction-tuned LLaMA model).

For reranked ChatGPT, recall is 79.3% by humans and 83.9% automatically, so agreement is not exact.

Should evidence come before the answer?

- Give the model passages, then generate an answer with citations.
- Generate the answer without passages, then attach citations.
- Which approach produces better-supported answers?

What does each row do?

	Fluency	Correct.	Citation	
	(MAUVE)	(EM Rec.)	Rec.	Prec.
ChatGPT				
VANILLA (5-psg)	66.6	40.4	73.6	72.5
w/ RERANK	77.0	40.2	**84.8**	**81.6**
SUMM (10-psg)	70.0	**43.3**	68.9	61.8
w/ INTERACT	69.0	39.1	73.4	66.5
SNIPPET (10-psg)	69.8	41.4	65.3	57.4
INLINESEARCH	58.7	32.4	58.3	58.2
CLOSEDBOOK	52.7	38.3	26.7	26.7

Fluency (MAUVE) compares generated and human-written text.
Correctness (EM recall) measures the fraction of expected short answers included.

Vanilla: The model writes from five full passages.

w/ Rerank: Generate four full answers and keep the one with the highest automatic citation recall.

SUMM: The model writes from LLM-written summaries of ten passages.

w/ INTERACT: Start with summaries, then let the model open full passages while writing.

What does each row do?

	Fluency	Correct.	Citation	
	(MAUVE)	(EM Rec.)	Rec.	Prec.
ChatGPT				
VANILLA (5-psg)	66.6	40.4	73.6	72.5
w/ RERANK	77.0	40.2	**84.8**	**81.6**
SUMM (10-psg)	70.0	**43.3**	68.9	61.8
w/ INTERACT	69.0	39.1	73.4	66.5
SNIPPET (10-psg)	69.8	41.4	65.3	57.4
INLINESEARCH	58.7	32.4	58.3	58.2
CLOSEDBOOK	52.7	38.3	26.7	26.7

Fluency (MAUVE) compares generated and human-written text. Correctness (EM recall) measures the fraction of expected short answers included.

SNIPPET: The model writes from copied excerpts of ten passages.

INLINESEARCH: While writing, the model searches within the 100 retrieved passages.

CLOSED BOOK: Write without passages, then attach each sentence's best-matching passage from the 100 retrieved passages.

A supported citation is not the end of verification

- ALCE separates answering the question from supporting the answer with citations.
- A passage can support a claim while the source itself is wrong or untrustworthy.
- A matching citation does not prove the model used that source when it wrote the answer.
- Readers still need to find the supporting words; the next method selects short evidence spans before writing.

Make the supporting words easy to find



Attribute First, then Generate:
Locally-attributable Grounded Text Generation

Aviv Slobodkin[1*] Eran Hirsch[1*] Arie Cattan[1] Tal Schuster[2] Ido Dagan[1]
[1]Bar-Ilan University [2]Google Research
{lovodkin93, hirscheran, ariecattan}@gmail.com
talschuster@google.com dagan@cs.biu.ac.il

Attribute First, then Generate: Locally-attributable Grounded Text Generation

Aviv Slobodkin and colleagues at ACL 2024.

A citation can point to a relevant document without showing which words support the sentence.

The input is documents plus a question or a summarization task.
The output should link each sentence to short source spans, or selected pieces of source text.

Attribute First chooses those spans before writing, then turns them into the sentence's citations.

One document can supply different pieces of evidence

What's the difference between eavestroughs and rain gutters?

Eavestroughs and gutters are essentially the same thing, with eavestroughs being a specific type of gutter that is installed at the edge of a roof [1a, 2a, 3a]. An eavestrough is a trough that is connected to the eaves of a building, which are the sections of the roof that hang over the wall, and it is used to direct rainwater away from the roof and towards a drainage system [2b, 3b].

- Numbers identify source documents; letters identify selected spans within them.

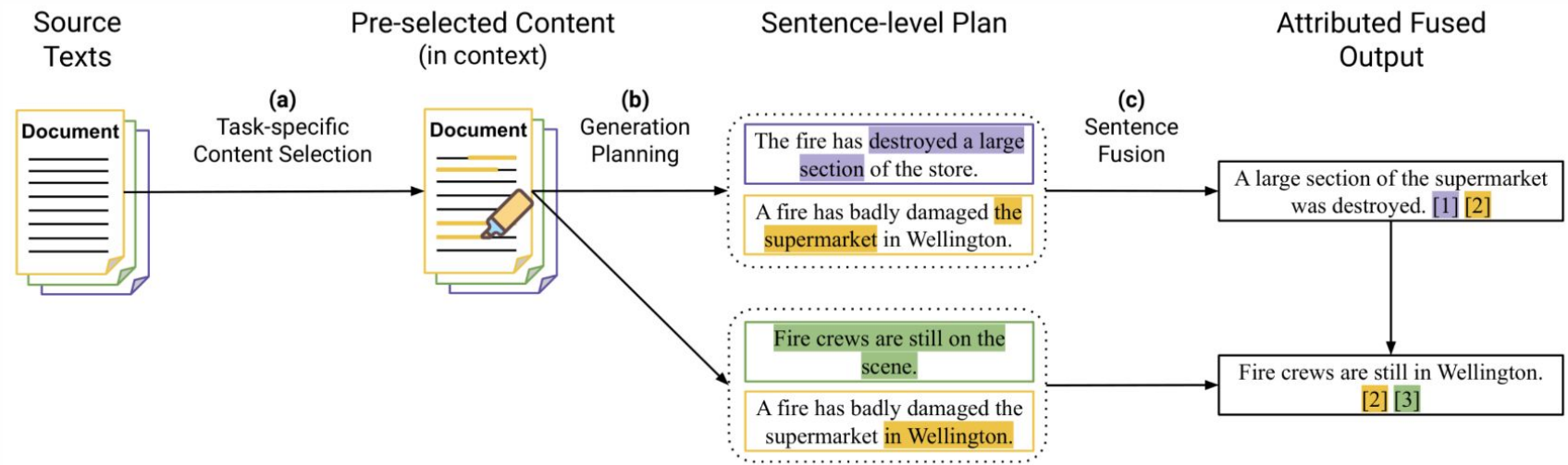
One document can supply different pieces of evidence

- The first sentence cites [1a, 2a, 3a]; the second cites [2b, 3b].
- In document 2, span a supports the claim that gutter and eavestrough are synonyms.
- Span b supports the definition of an eavestrough as a trough under the eaves.
- Different sentences can cite different spans from the same document.

Document 2

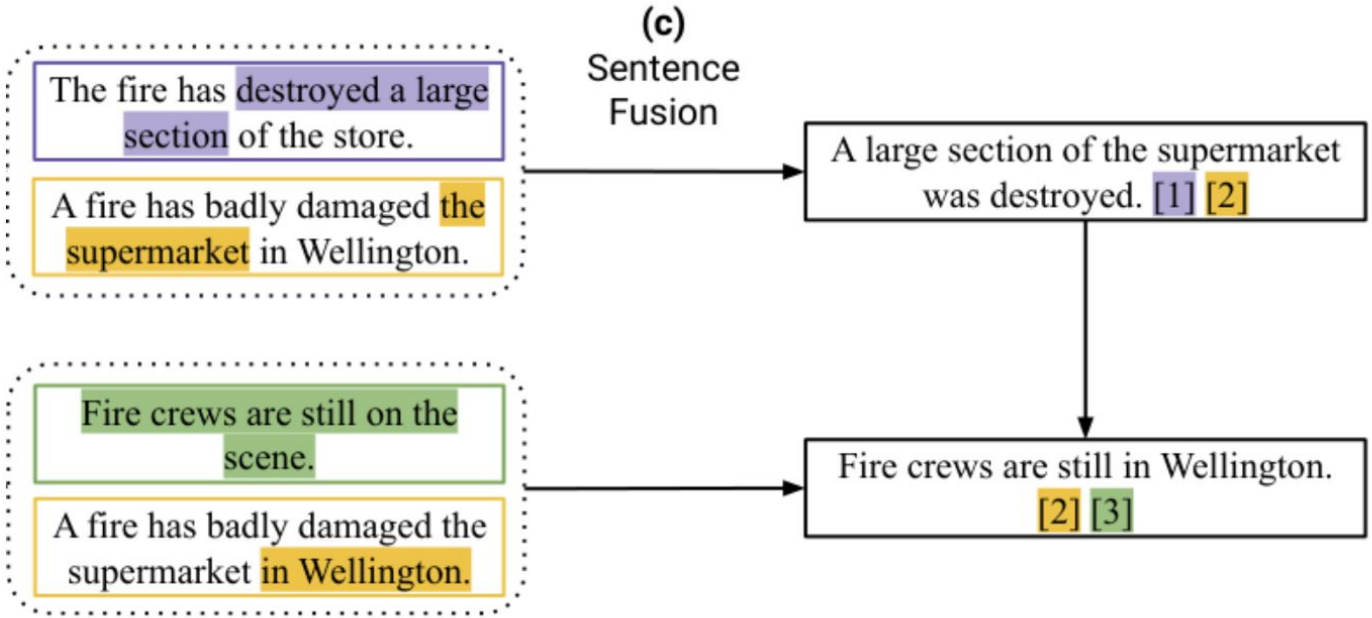
[a] Gutter is a synonym of eavestrough. As nouns the difference between eavestrough and gutter is that [b] eavestrough is a trough under the eaves of a building for draining water from the roof; gutter while gutter is a prepared channel in a surface, especially at the side of a road adjacent to a curb, intended for the drainage of water or ...

Select evidence, group it, then write from it



- Select verbatim spans that answer the question or carry the summary's key facts.
- Group complementary spans into an ordered plan, one group for each intended sentence.
- For each sentence, give the model that group highlighted in its documents, plus the previous output.
- Generate the sentence in a separate call, then cite the spans in its group.

Each group supplies the facts for one sentence



- One group combines damage with the supermarket; the next combines fire crews with Wellington. Each group supplies citations, while previous output helps the next sentence connect.

What would count as an improvement?

For this experiment, the ALCE baseline cites full documents, while Attribute First cites selected spans; both use the same Gemini-Pro model.

- Both Attribute First implementations select evidence before writing. The tested ICL-CoT version uses examples in its prompts: one call selects spans, and a second writes a sentence plan and the answer.

What would count as an improvement?

For this experiment, the ALCE baseline cites full documents, while Attribute First cites selected spans; both use the same Gemini-Pro model.

- Support asks whether the cited evidence backs the whole output sentence, not just its topic.
- Fluency and helpfulness rate the writing and usefulness, not the strength of the citation evidence.
- Checking time measures the reader's work after generation, not the model's generation speed.

On QA, checking gets faster and support improves

AIS (Attributable to Identified Sources) is the percentage of sentence judgments credited: full support and evidence that both supports and contradicts the sentence count; partial support does not.

Time is seconds per answer check; fluency and helpfulness rate coherent writing and useful answers from 1 (strongly disagree) to 5 (strongly agree).

Method	Fluency↑	Helpful.↑	AIS↑	Time (sec)↓
ALCE	**4.9**	**4.7**	87.6	59
ATTR. FIRST$_{\text{ICL-CoT}}$	**4.9**	4.5	**94.4**	**35**

- Checking time falls from 59 to 35 seconds, while AIS rises from 87.6 to 94.4.
- Fluency stays at 4.9; helpfulness is slightly lower, from 4.7 to 4.5.

Does summarization show the same pattern?

Multi-document summarization turns several news articles into one summary.

AIS credits sentence attribution from 0 to 100; time is seconds per text check; fluency rates coherent writing from 1 to 5.

Does easier checking still come with stronger support?

Method	Fluency↑	AIS↑	Time (sec)↓
ALCE	4.7	**89.3**	47
ATTR. FIRST$_{\text{ICL-CoT}}$	**4.9**	79.9	**22**

- The summaries take less time to check, from 47 to 22 seconds per text.
- But AIS falls from 89.3 to 79.9, even as fluency rises from 4.7 to 4.9.
- The summaries are easier to check and read, but their citations receive less attribution credit.

The answer can use more than its citation contains

- The generator sees surrounding source text as well as the selected highlights.
- It can write a detail from the document that is missing from its own citation.
- About 42 percent of 109 unsupported sentences were partially supported, often with the missing evidence nearby.
- Selection mistakes can propagate, and extra calls can slow generation even when readers check the output faster.

Which clause does the selected span support?

Paper output

The Sheraton Waikiki is the only hotel on Oahu that provides on-site marriage licenses, and it is running a new ad campaign in mainland LGBT publications.

Yellow marks the selected citation; the rest is surrounding source context.

The Sheraton Waikiki is the only hotel on Oahu providing on-site marriage licenses. General Manager Kelly Sanders says now the hotel is running a new ad campaign in mainland LGBT publications.

- Which clause has support inside the selected citation?
- Which relies on nearby text outside it?

The document has the fact, but the citation leaves it out

2	**Output: The Sheraton Waikiki is the only hotel on Oahu that provides on-site marriage licenses**, and it is running a new ad campaign in mainland LGBT publications. **Attribution:** . . . The Sheraton Waikiki is the only hotel on Oahu providing on-site marriage licenses. General Manager Kelly Sanders says now the hotel is running a new ad campaign in mainland LGBT publications. . . .

- The advertising clause is supported; the marriage-license clause is not supported by the selected span.
- The marriage-license fact is in the previous source sentence, which was not selected as citation evidence.
- We could expand the citation to cover both claims, or remove the unsupported clause from the output.
- Then check the whole revised sentence against its citations, not against everything in the document.

Would you make the citations wider?

Summarization

Method	Fluency↑	AIS↑	Time (sec)↓
ALCE	4.7	**89.3**	47
ATTR. FIRST$_{\text{ICL-CoT}}$	**4.9**	79.9	**22**

Long-form QA

Method	Fluency↑	Helpful.↑	AIS↑	Time (sec)↓
ALCE	**4.9**	**4.7**	87.6	59
ATTR. FIRST$_{\text{ICL-CoT}}$	**4.9**	4.5	**94.4**	**35**

What benefit appears in both tasks, and what differs?

Would expanding the citations automatically recover support without giving up that benefit?

Short enough to check, complete enough to support

- In this study, checking is faster on both tasks, but attribution improves for QA and declines for summarization.
- A citation needs enough evidence to support the whole sentence, without making verification another search task.
- For the hotel case, expand the citation or narrow the output, then measure support and checking time again.
- That is a proposed repair to one failure mode, not evidence that wider citations fix the whole summarization result.

Would shorter citations help your application?

- Suppose your team wants shorter citations so reviewers can check answers faster.
- What would you measure before deciding to adopt the change?
- Compare checking time and whole-sentence support on the same questions and sources.
- Also check whether the answers remain useful. A faster review can come with weaker support or less useful information.

Five checks before trusting an answer

- Can we find the evidence the question requires?
- Does the model actually use the relevant evidence?
- Is the source reliable and current enough for the claim?
- Does the cited text support every part of the answer?
- Can a person locate and check that support without reading everything?
- Passing one check does not establish the others. If a check fails, repair that connection and test whether the repair helps.