Retrieval-Augmented Generation

Lars Schimmelpfennig, Levi Kaster, and Zilong Wang

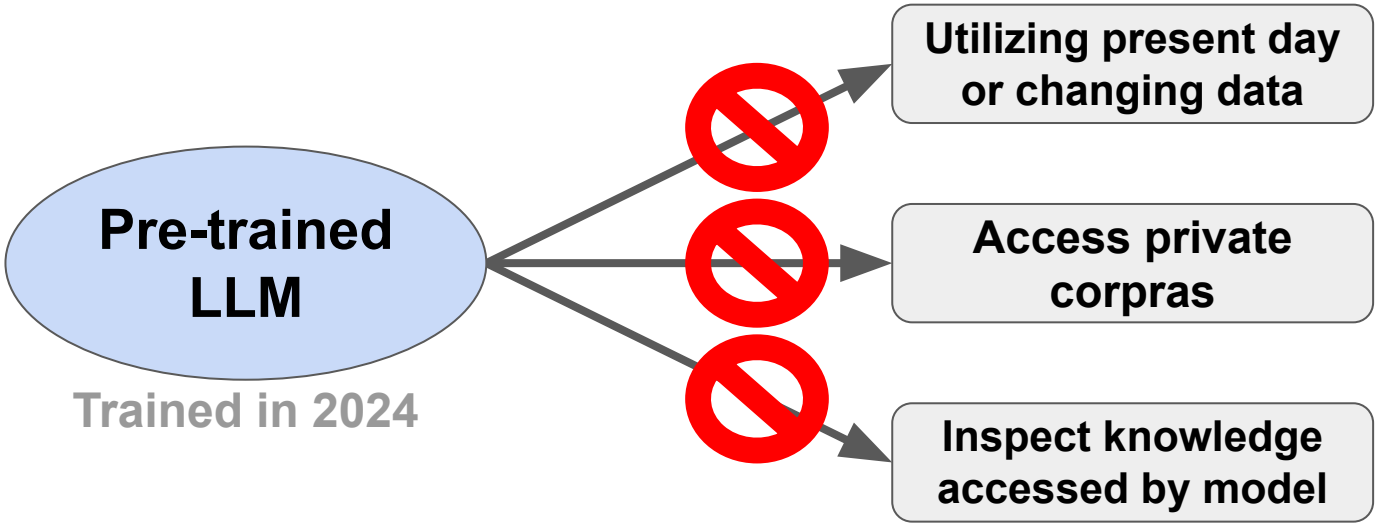
Paper 1: Retrieval Augmented Generation for Knowledge-Intensive NLP Tasks

P Lewis, E Perez, A Piktus, F Petroni, V Karpukhin, N Goyal, H Küttler, M Lewis, W Yih, T Rocktäschel, S Riedel, D Kiel

Presenter: ***Levi Kaster***

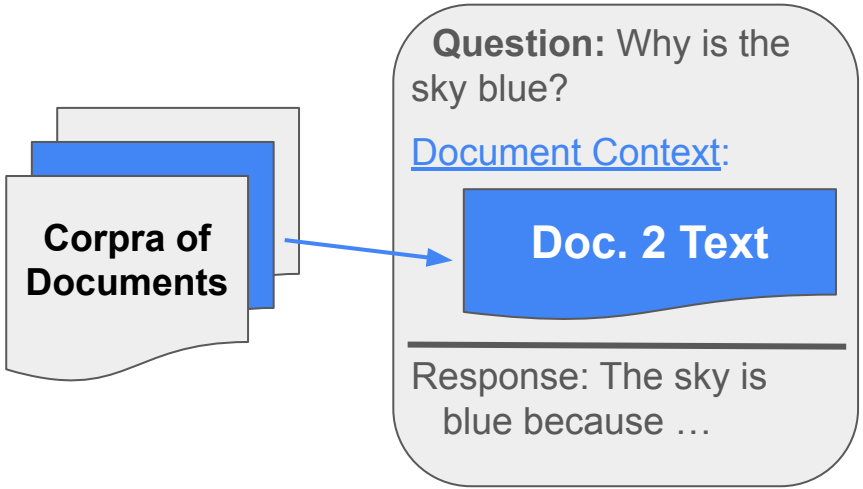
Introduction

Pre-trained models have an parametric knowledge base that is not easily expanded or revised.



Hybrid models that combine the parametric knowledge with retrieval-based memory allow knowledge to be revised and expanded.

Simplified Retrieval-based Systems

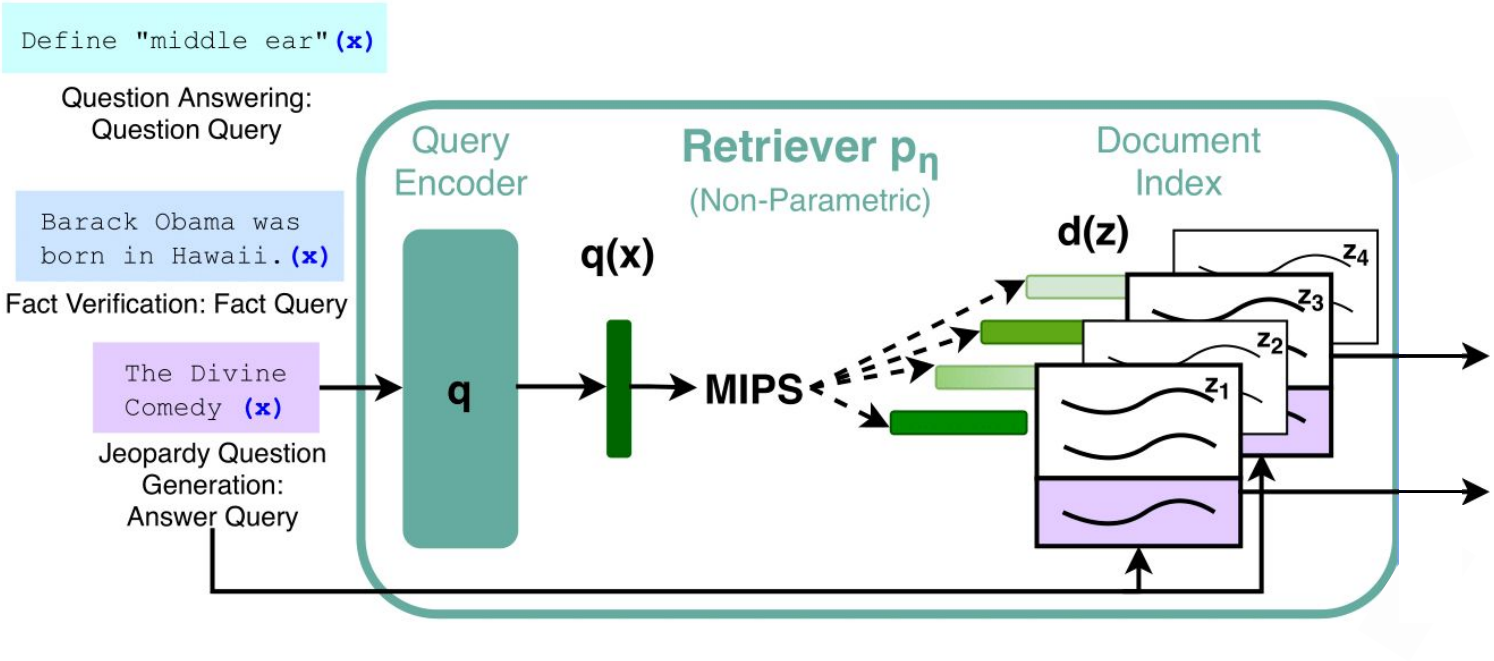


Components

- Retriever
- Generator

Methodology: Retrieval Augmented Generation Approach

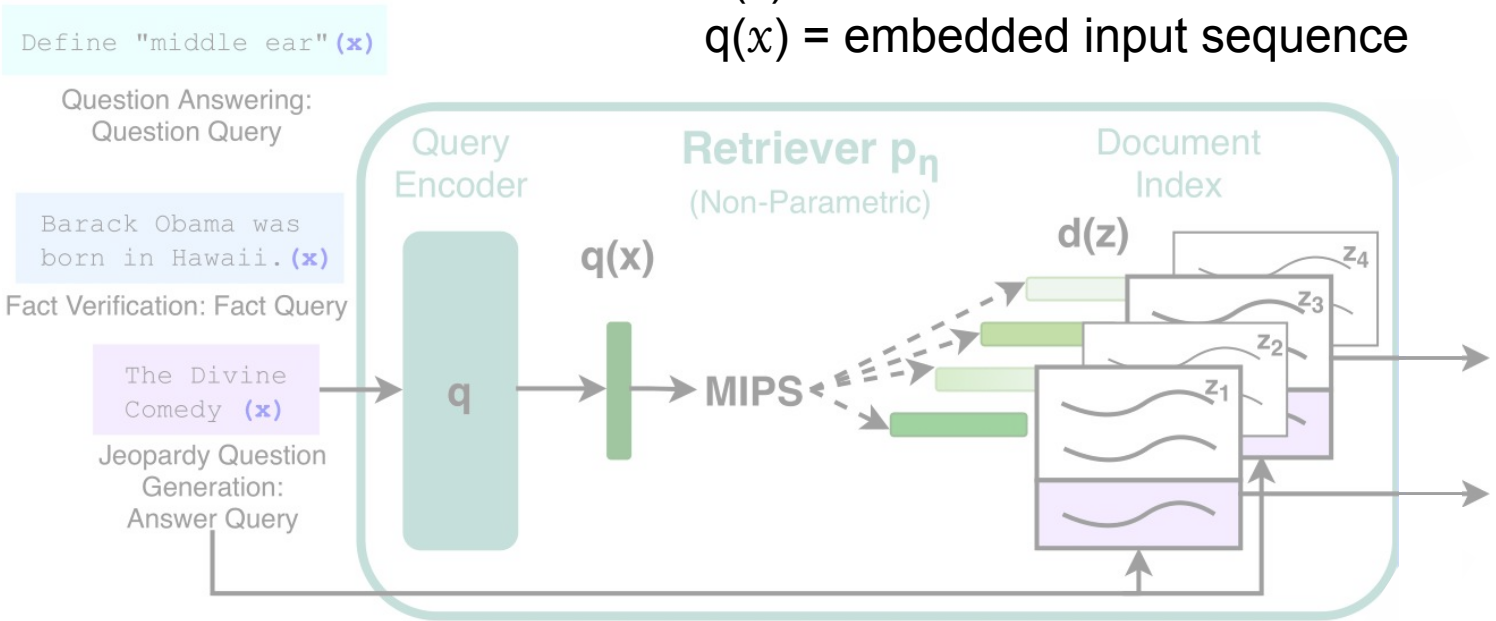
RAG Retriever (p_η)



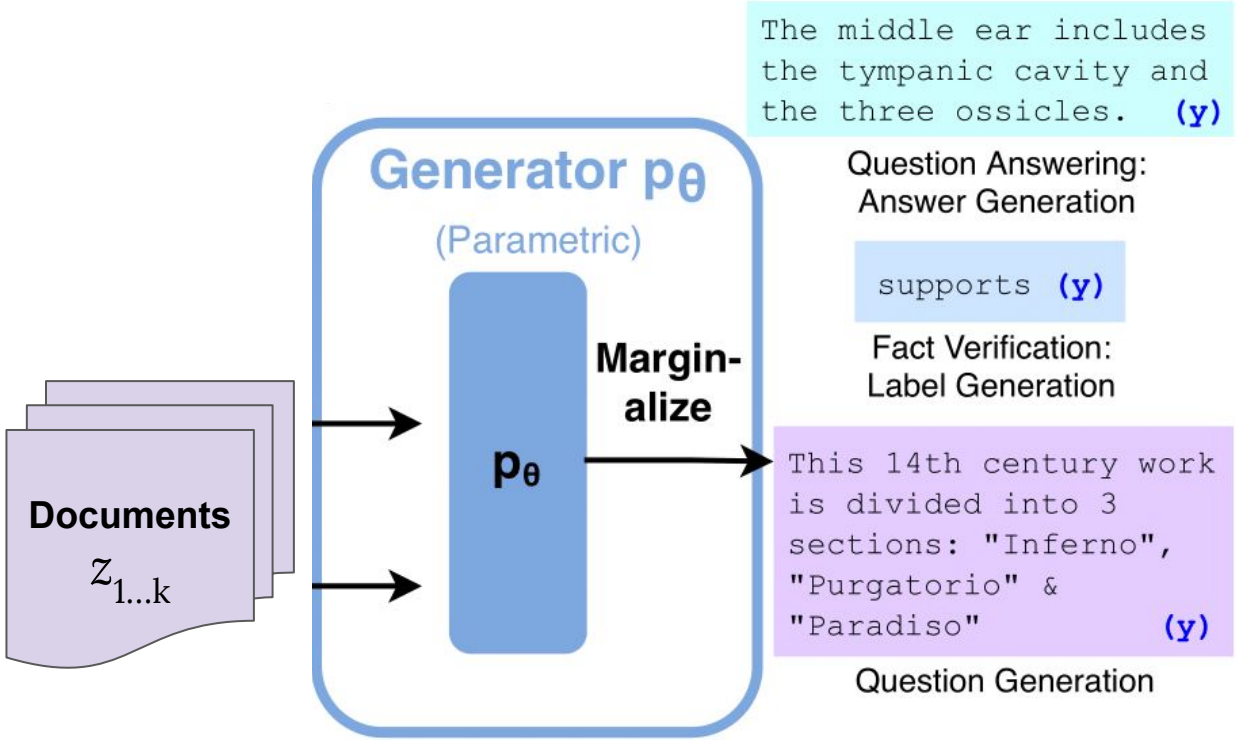
RAG Retriever (p_η)

$$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right)$$

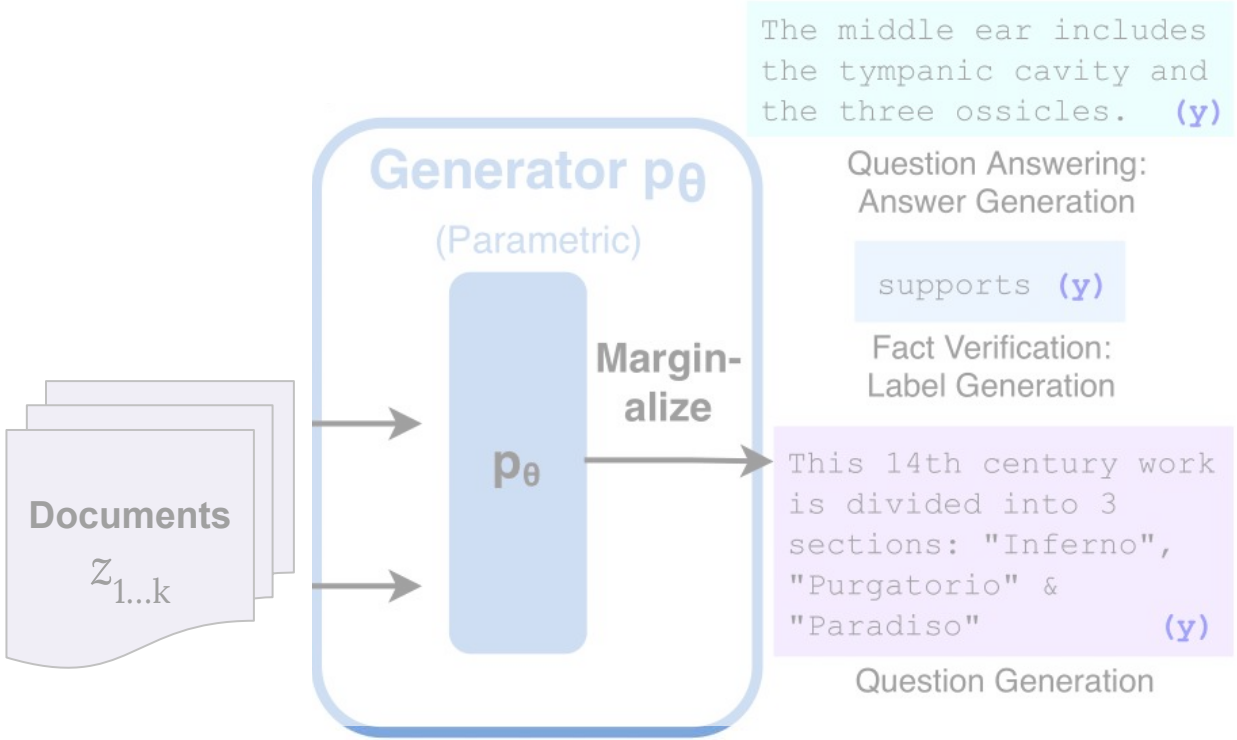
x = input sequence
d(z) = embedded documents
q(x) = embedded input sequence



RAG Generator (p_θ)



RAG Generator (p_θ)



$$p_\theta(y_i|x, z, y_{1:i-1})$$

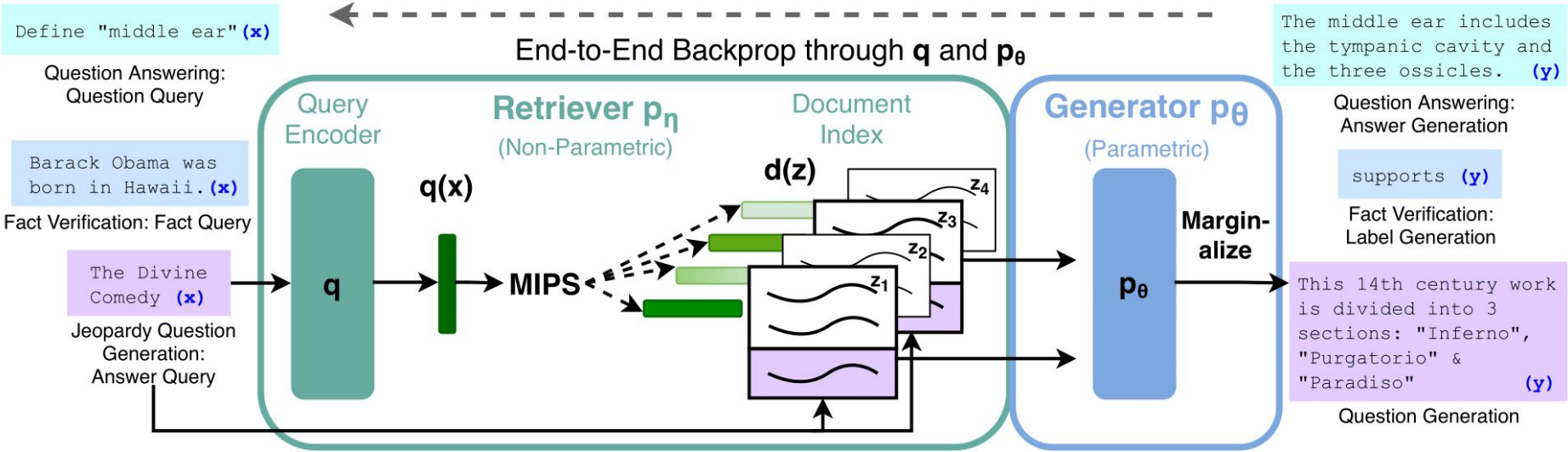
y_i = next token in output

x = input sequence

z = corpora documents

$y_{1:i-1}$ = previous output tokens

Retrieval-Augmented Generation (RAG) Approach



RAG-Sequence and RAG-Token Model

RAG-Sequence

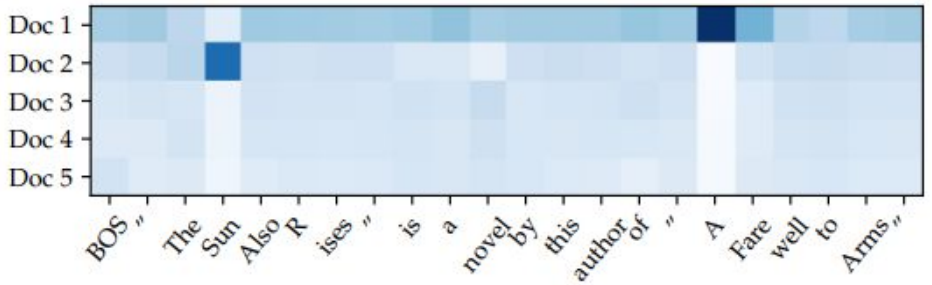
- The same retrieved document(s) is used to generate the complete generator sequence

RAG-Token

- Separate latent documents can be used for the generation of each target token by the generator

Document 1: his works are considered classics of American literature ... His wartime experiences formed the basis for his novel **"A Farewell to Arms"** (1929) ...

Document 2: ... artists of the 1920s "Lost Generation" expatriate community. His debut novel, **"The Sun Also Rises"**, was published in 1926.



Experiments

RAG Implementation in Paper

Encoding models within Retriever

- $\text{BERT}_{\text{BASE}}$ *document encoder*
- $\text{BERT}_{\text{BASE}}$ *query encoder*

Document Corpora

- Vector index of Wikipedia articles from 2018 (when document database not provided by dataset)
  - Each article split into 100 word-chunks
- Retrieve $k \in \{5, 10\}$ documents for each task

Generator Model

- BART Model utilized as encoder-decoder generator

Datasets

Open Domain Question Answering

- Four Separate Datasets:
  - Natural Questions (NQ)
  - TriviaQA (TQA)
  - WebQuestions (WQ)
  - CuratedTrec (CT)
- Each task includes a question, an answer, and excerpt(s) containing that answer.

Abstractive Question Answering

- MSMARCO Dataset
- Question-answer task, but excerpts associated with each question are excluded

Jeopardy Question Answering

- SearchQA Dataset
- Example:
  - Input: “The World Cup”
  - Correct Output: “In 1986 Mexico scored as the first country to host this international sports competition twice.”

Fact Verification

- FEVER Dataset
- Given wikipedia text, model must determine factuality of a claim
  - Binary: supports/refutes
  - Multiclass: supports/refutes/not enough info

Open Domain Question Answering

Evaluation:
Accuracy of Exact String Match

	Model	NQ	TQA	WQ	CT
Closed Book	T5-11B [52]	34.5	- /50.1	37.4	-
	T5-11B+SSM[52]	36.6	- /60.5	44.7	-
Open Book	REALM [20]	40.4	- / -	40.7	46.8
	DPR [26]	41.5	**57.9**/ -	41.1	50.6
	RAG-Token	44.1	55.2/66.1	**45.5**	50.0
	RAG-Seq.	**44.5**	56.8/**68.0**	45.2	**52.2**

Abstractive QA, Jeopardy and Fact Verification

Evaluation Metrics

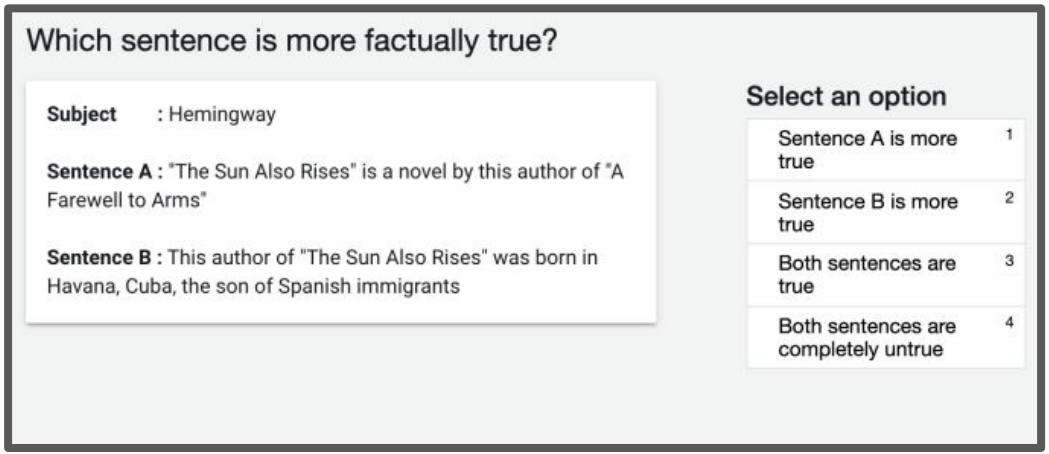
Jeopardy
B-1: BLEU-1
QB-1: Q-BLUE-1

MSMARCO (QA)
R-L: Rouge-L
B-1: BLEU-1

Fact Verification
Fact verification label accuracy (2/3 Classes)

Model	Jeopardy		MSMARCO		FVR3	FVR2
	B-1	QB-1	R-L	B-1	Label Acc.	
SotA	-	-	**49.8***	**49.9***	**76.8**	**92.2***
BART	15.1	19.7	38.2	41.6	64.0	81.1
RAG-Tok.	**17.3**	**22.2**	40.1	41.5	72.5	89.5
RAG-Seq.	14.7	21.4	40.8	44.2		

Jeopardy Task Human Evaluation



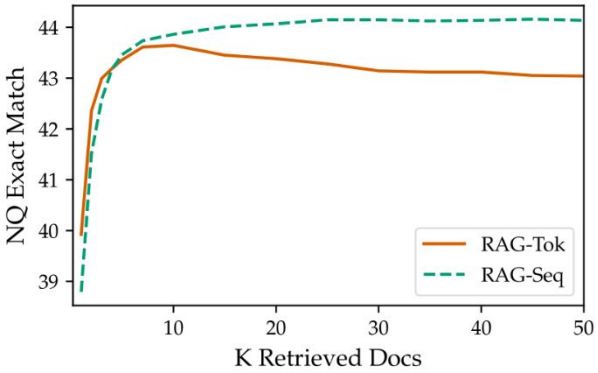
	Factuality	Specificity
BART better	7.1%	16.8%
RAG better	**42.7%**	**37.4%**
Both good	11.7%	11.8%
Both poor	17.7%	6.9%
No majority	20.8%	20.1%

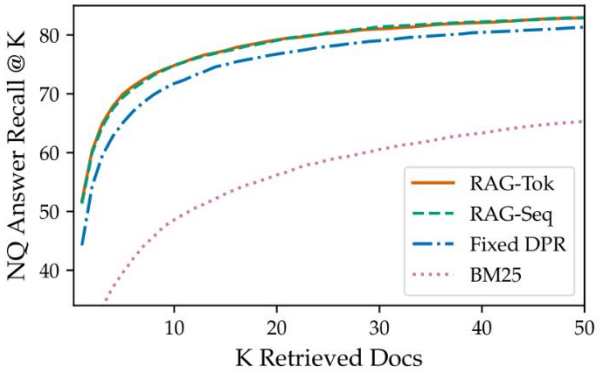
Additional Benefits of RAG

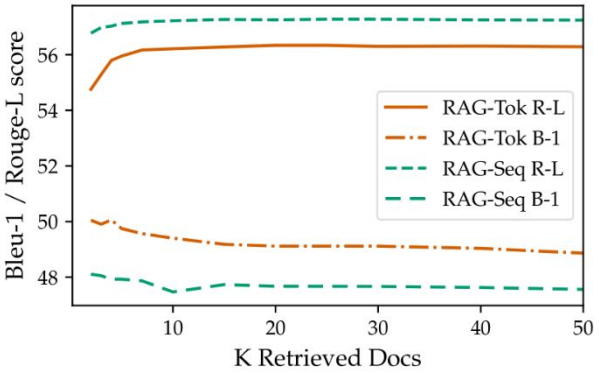
- RAG responses are more diverse than BART alone (parametric only)
- Retrieving more documents may improve performance.
  - Dependent on *task*, *performance metric*, and *RAG-methodology*

Distinct Tri-grams for Generation Tasks

	MSMARCO	Jeopardy QGen
Gold	89.6%	90.0%
BART	70.7%	32.4%
RAG-Token	77.8%	46.8%
RAG-Seq.	83.5%	53.8%







Major Contributions / Discussion

- RAG models obtain imporved performance compared to base pre-trained language models across various tasks

- RAG allows previously unattainable tasks to be performed, allowing for the incorporation of data from external documents during response generation.
  - Expanded Knowledge

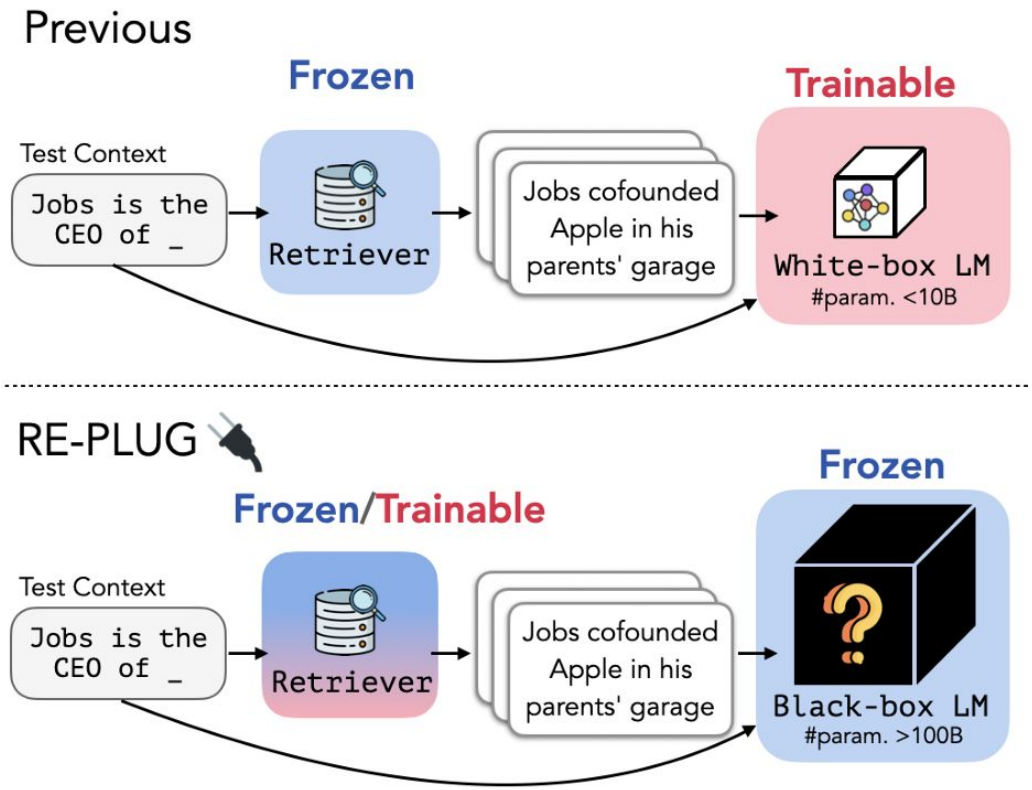
- RAG produced more factual and more specific responses compared to parametric-only models

Paper 2: REPLUG: Retrieval-Augmented Black-Box Language Models

W Shi, S Min, M Yasunaga, M Seo, R James, M Lewis, L Zettlemoyer, W Yih

Presenter: ***Lars Schimmelpfenning***

REPLUG: Retrieval-Augmented Black-Box Language Models

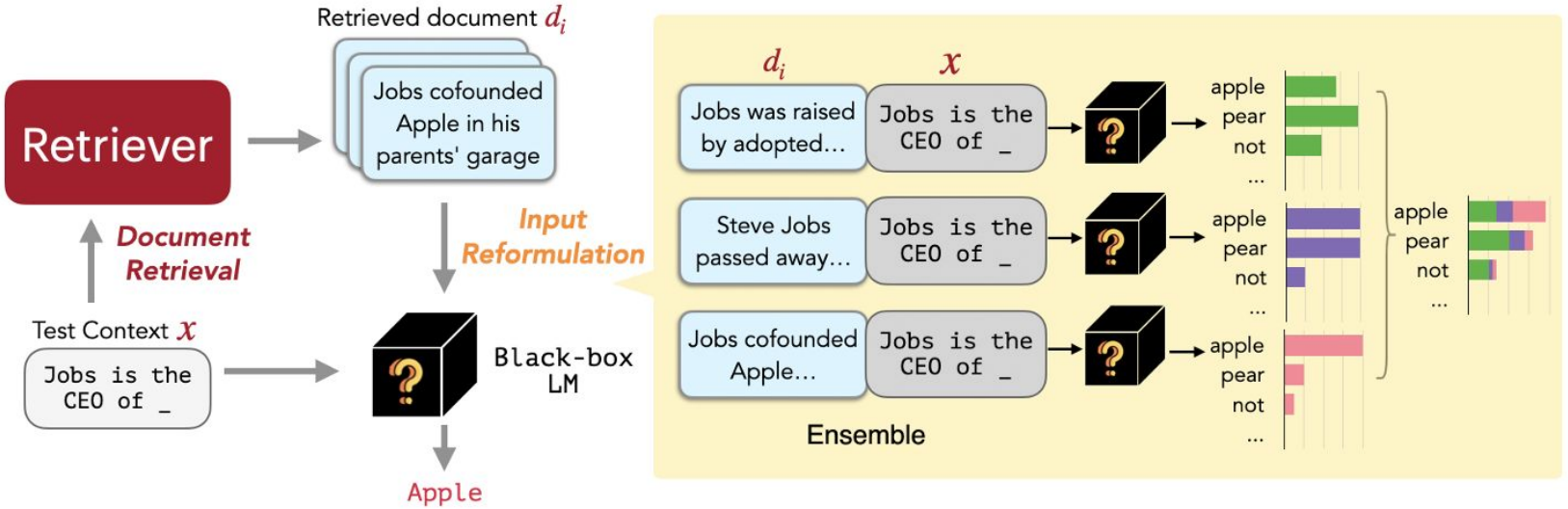


Motivation

Fine-tuning an LLM is expensive and infeasible for large API accessed models.

The authors propose only training the retriever and prepending to the input.

Downside: adding extra text to the prompt takes attention away from the prompt and answer. They mitigate this with an ensemble approach.



Answer token distribution is taken as the weighted average of each sample.

Weight is assigned by document similarity/retriever score

In reality each document only contributes one answer*

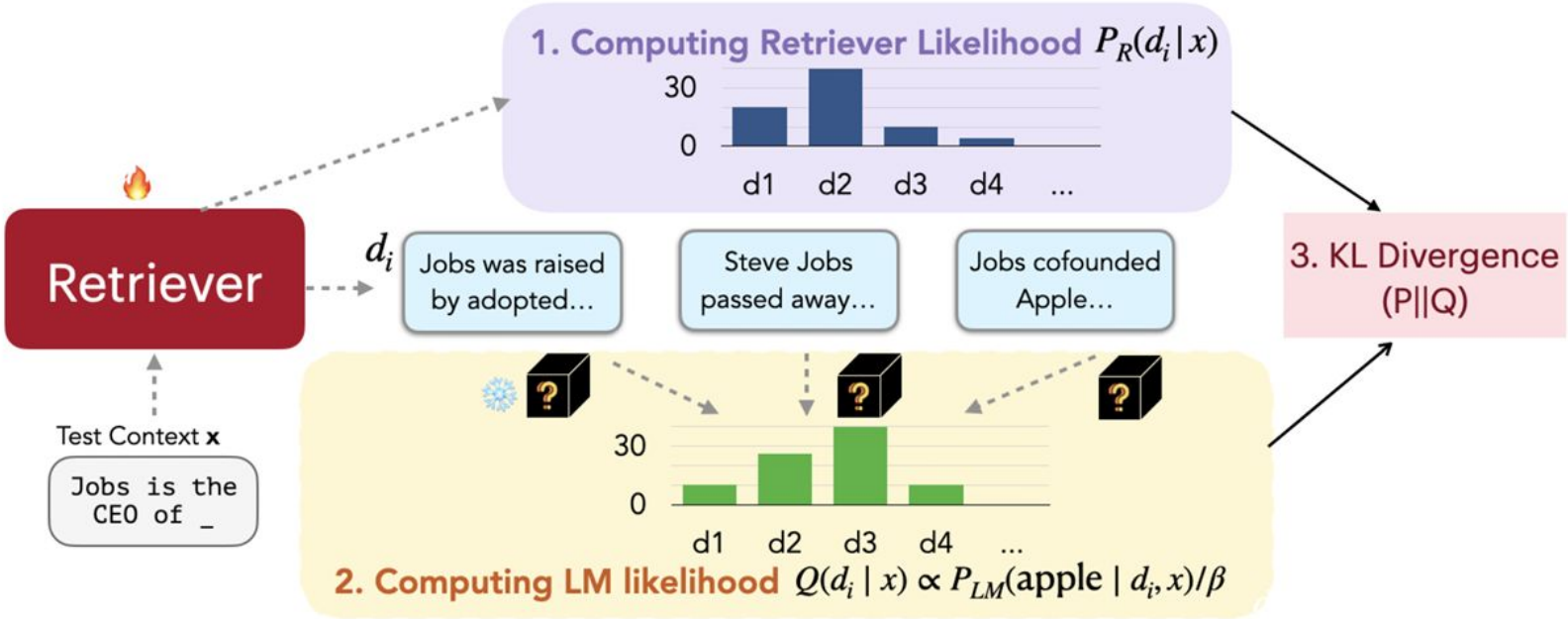
Note on computational cost

“Therefore, compared with the method of prepending all the retrieved documents, our ensemble methods do not incur additional computational cost overhead.”

They still need to run multiple samples to create the ensemble.

API output tokens are more expensive than input tokens.

Retriever training (REPLUG LSR on 'The Pile') Gao et al. 2020



"LM likelihood" measures how much a document improves confidence. Measured as perplexity: the length normalized -log likelihood.

Evaluation

Massive Multi-task Language Understanding (MMLU): exam MC questions from 57 subjects, grouped into 4 categories (humanities, STEM, social sciences, and other)

Natural Questions (NQ) and TriviaQA (TQA)

Retriever is allowed to get 10 Wikipedia documents for both.

MMLU (Multiple Choice Accuracy)

Model	# Parameters	Humanities	Social.	STEM	Other	All
Codex	175B	74.2	76.9	57.8	70.1	68.3
PaLM	540B	77.0	81.0	55.6	69.6	69.3
Flan-PaLM	540B	-	-	-	-	72.2
Atlas	11B	46.1	54.6	38.8	52.8	47.9
Codex + REPLUG	175B	76.0	79.7	58.8	72.1	71.4
Codex + REPLUG LSR	175B	76.5	79.9	58.9	73.2	71.8

(5-shot) All models get 5 examples of Q:A pairs.

Atlas (lit. result) is fine-tuned to perform RAG with a different document corpus: Wikipedia + CCNet (common crawl)

NQ and TQA Results (Exact Match Between Synonyms)

RAG Retriever Fine-Tuned Jointly
Results taken from lit.

Model	NQ		TQA	
	Few-shot	Full	Few-shot	Full
Chinchilla	35.5	-	64.6	-
PaLM	39.6	-	-	-
Codex	40.6	-	73.6	-
RETRO†	-	45.5	-	-
R2-D2†	-	55.9	-	69.9
Atlas†	42.4	**60.4**	74.5	**79.8**
Codex + Contriever$_{cc}$ [2]	44.2	-	76.0	-
Codex + REPLUG	44.7	-	76.8	-
Codex + REPLUG LSR	**45.5**	-	**77.3**	-

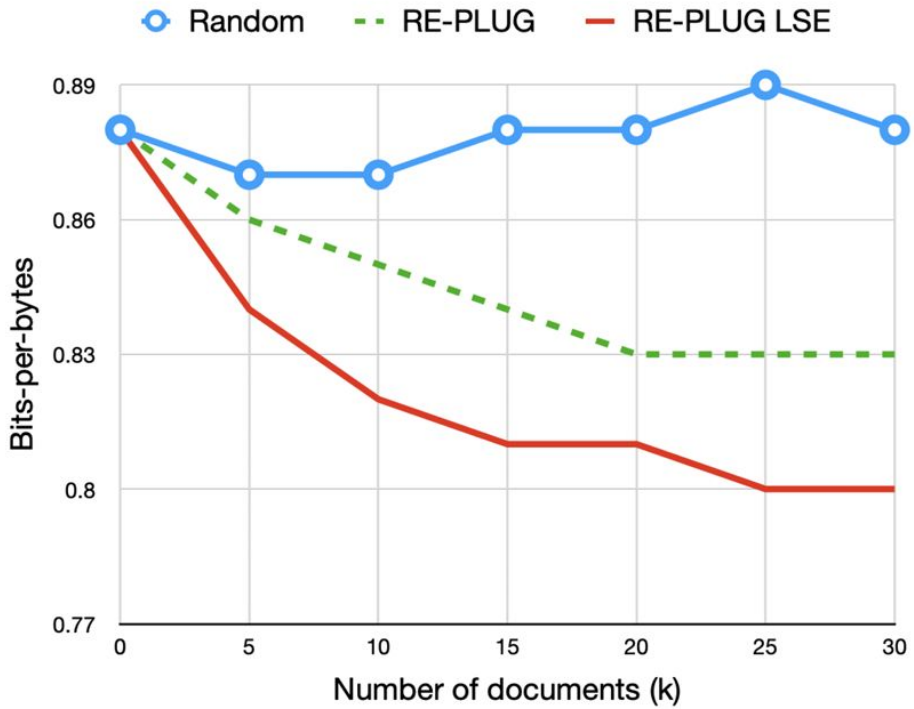
Not a completely fair comparison as the other RAG models did not have the same document corpus.

RETRO uses MassiveText (similar to The Pile). Again, atlas uses wikipedia + CCNET. R2-D2 and Codex + Contriver likely use different subsets of Wikipedia.

The Pile

Bits-per-bytes: Bits measures uncertainty per UTF-8 character (byte) instead of token.

The ensemble approach with random documents does not help the model



Conclusions + Limitations

REPLUG offers an approach for fine-tuning only the retriever by predicting which documents will improve model confidence.

The ensemble approach alone performed well without fine tuning.

Other RAG approaches were not re-implemented by the authors with the same Wikipedia corpus making the results less comparable.

Q&A

Q: If the retriever is trained to minimize LM perplexity, how do you ensure this also improves task accuracy rather than just “perplexity gaming”?

A: Great question, accuracy could also be optimized but with some complications

- Need a supervised dataset
- Accuracy is non-differentiable → need RL or reward modeling
- Some newer works (DRAGONt) explore this direction

Investigating the Factual Knowledge Boundary of Large Language Models with Retrieval Augmentation

Ruiyang Ren†, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao‡, Jing Liu‡, Hao Tian, Hua Wu, Ji-Rong Wen. Haifeng Wang

Presenter: *Zilong Wang*

Introduction & Motivation

- Limited flexibility in knowledge-intensive tasks

  Rely on the incorporation of retrieval augmentation

- Lack of understandings of LLMs' factual knowledge boundaries

  Does LLMs 'overconfident' or 'underconfident'?

  "Do I even need supporting documents?"

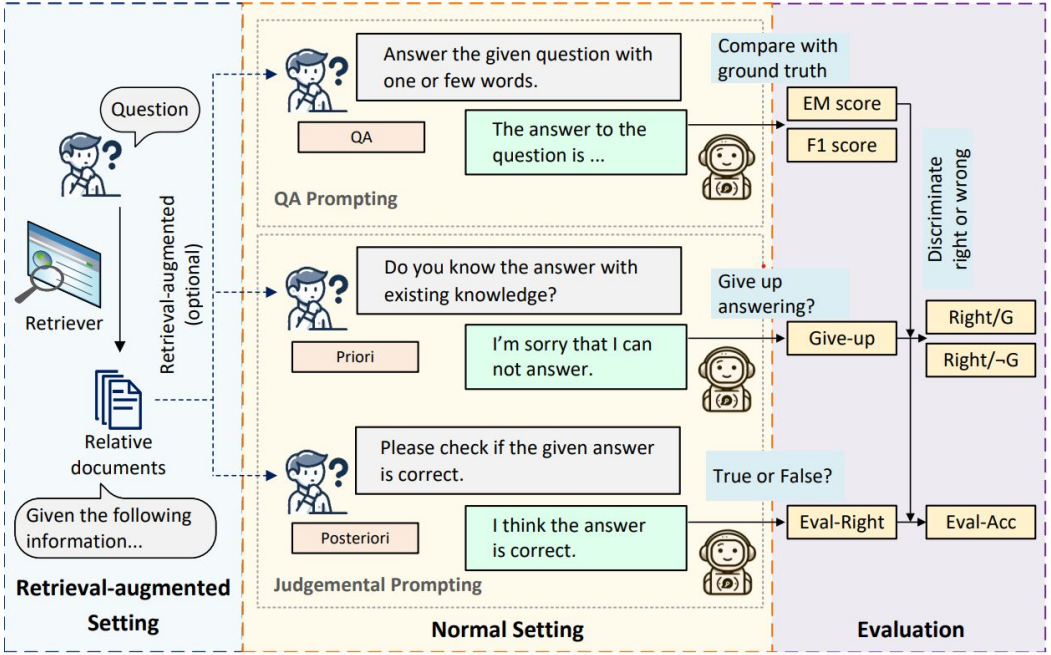
Introduction & Motivation

(i) To what extent can LLMs perceive their factual knowledge boundaries?

(ii) What effect does retrieval augmentation have on LLMs?

(iii) How do supporting documents with different characteristics affect LLMs?

Methodology



QA Prompting:

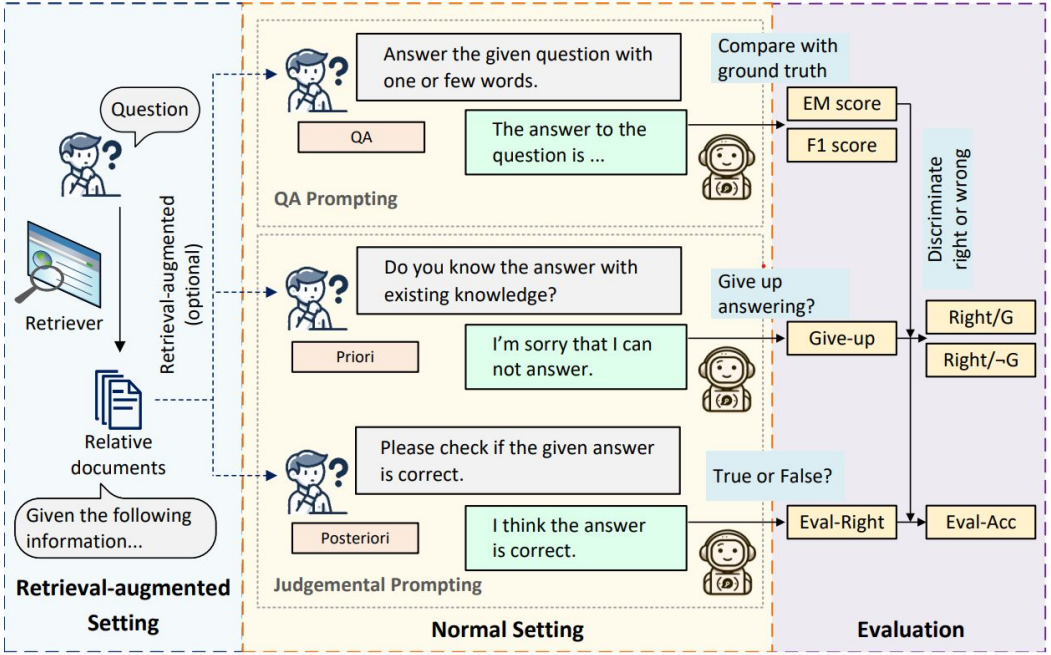
Exact Match (EM) Score:

Percentage of questions with exactly correct answers.

F1 Score:

The word overlap between prediction and ground truth using the harmonic mean of precision and recall.

Methodology



Judgemental Prompting (Priori):

Give-up:

The percentage of questions that LLMs give up answering

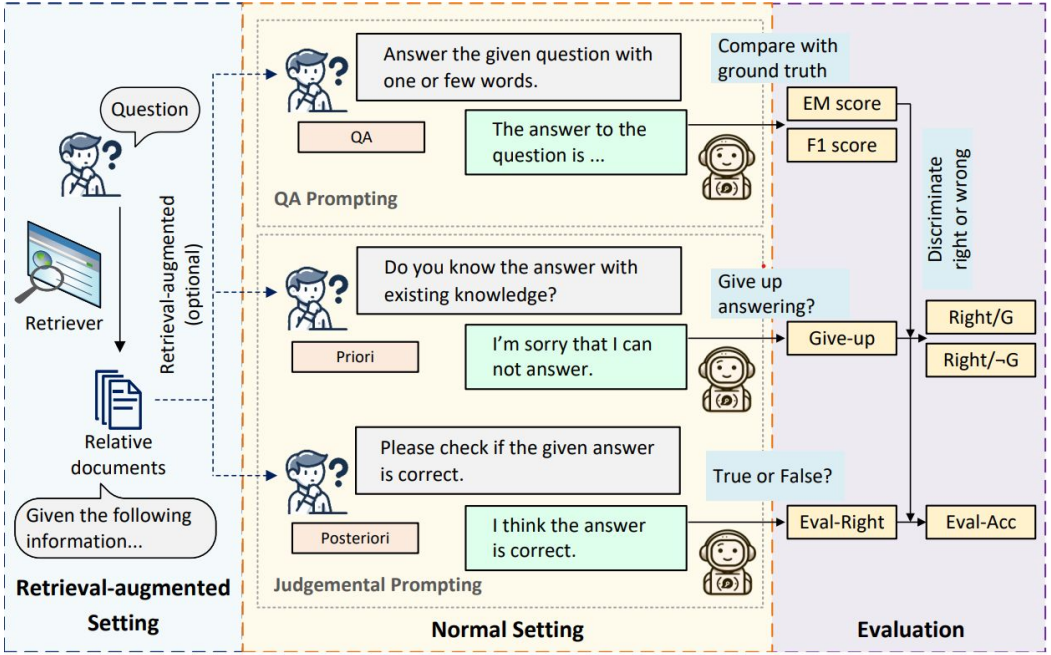
Right / G:

P (give up answering **|** can answer correctly**)**

Right / ¬G:

P (Not give up answering **|** can answer correctly**)**

Methodology



Judgemental Prompting (Posteriori):

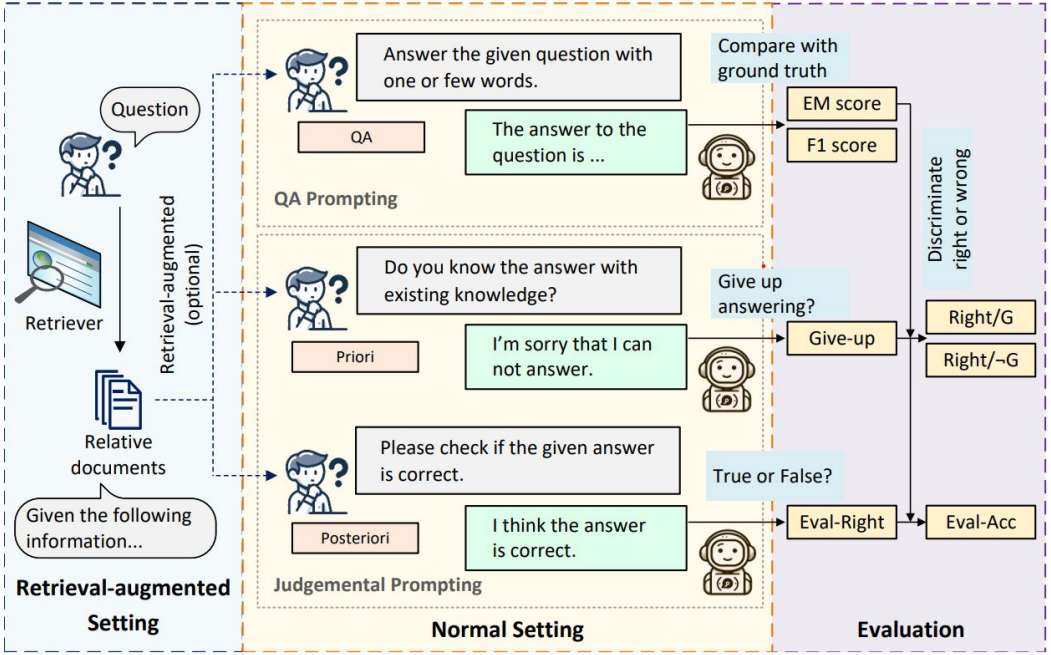
Eval-Right:

Proportions of questions assessed as 'correct'

Eval-Acc:

Share of answers where the self-assessment aligns with the fact

Methodology



Retrievers:

(i): Dense Retrieval (RocketQAv2 + FAISS)

Find semantically relevant documents for questions

(ii): Sparse Retrieval (BM25)

Find lexical relevant documents for questions

(iii): ChatGPT

Instructed to produce relevant documents in response to a given question

Q1: To What Extent LLMs' Know Their Knowledge Boundaries?

Before Answering (Prior):

***High Right/¬G*: Often answer even when unsure.**

***Very low Give-up rate*: Rarely admit "I don't know".**

After Answering (Posterior):

Eval-Right ≫ EM: LLMs think they're correct more often than they really are.

Eval-Acc misaligned with actual **QA ability (*EM*).**

Main Finding:

LLMs struggle to perceive their factual knowledge boundary, and tend to be overconfident.

	LLM	QA		Priori Judgement		Posteriori Judgement	
		EM	F1	Give-up	Right/¬G	Eval-Right	Eval-Acc
NQ	Davinci003	27.20	36.20	29.20%	32.77%	72.80%	45.01%
	ChatGPT	33.40	45.32	57.40%	42.72%	84.40%	43.40%
	GPT-4	34.60	48.72	15.20%	39.15%	90.20%	38.87%
	LLaMA2	16.60	24.26	6.60%	17.56%	58.40%	46.74%
	Mistral	11.20	19.30	49.80%	15.94%	68.00%	37.90%
TriviaQA	Davinci003	65.20	69.57	7.40%	67.17%	87.00%	69.82%
	ChatGPT	69.00	75.29	25.00%	75.73%	88.80%	71.95%
	GPT-4	75.80	84.52	8.80%	77.85%	93.00%	76.57%
	LLaMA2	48.80	53.40	4.80%	50.21%	75.60%	57.60%
	Mistral	36.20	42.09	34.80%	46.63%	86.00%	48.10%
HotpotQA	Davinci003	18.40	26.78	35.40%	24.15%	70.60%	43.99%
	ChatGPT	20.80	29.27	78.40%	31.48%	66.80%	43.12%
	GPT-4	28.60	40.33	54.80%	42.92%	72.40%	45.74%
	LLaMA2	11.40	16.88	25.60%	12.63%	49.80%	54.88%
	Mistral	10.80	17.86	64.00%	19.44%	81.80%	27.40%

Q2: What Effect Does Retrieval Augmentation Have on LLMs?

Main Findings:

Dense Retriever Performs the Best

Even Wikipedia helps → pre-trained knowledge not fully utilized.

Open-source models gain more; closed-source show smaller benefit.

Noisy docs can mislead (performance drop on TriviaQA).

While LLMs cannot sufficiently utilize their internal knowledge, RAG can serve as a valuable knowledge supplement for LLMs

Datasets	LLMs	Retrieval Source	QA		Priori Judgement			Posteriori Judgement	
			EM	F1	Give-up	Right/G	Right/¬G	Eval-Right	Eval-Acc
NQ	Davinci003	Sparse	27.80	38.29	21.20%	12.26%	31.98%	39.40%	67.94%
		Dense	39.00	51.27	12.80%	14.06%	42.66%	46.40%	71.43%
		ChatGPT	34.00	47.36	6.20%	6.45%	35.82%	46.00%	71.54%
	ChatGPT	Sparse	28.40	41.10	42.40%	17.92%	36.11%	67.00%	48.77%
		Dense	39.40	52.65	26.60%	18.05%	47.14%	68.80%	53.56%
		ChatGPT	32.20	47.37	7.40%	2.70%	34.56%	78.80%	49.90%
	GPT-4	Sparse	34.20	45.81	28.20%	14.18%	42.06%	57.20%	48.48%
		Dense	43.60	56.36	12.60%	15.87%	47.60%	66.40%	50.86%
		ChatGPT	34.40	48.56	4.20%	4.76%	35.70%	69.80%	48.69%
	LLaMA2	Sparse	23.00	34.14	32.80%	15.85%	26.49%	6.00%	75.00%
		Dense	33.40	45.39	24.80%	20.16%	37.77%	5.20%	73.08%
		ChatGPT	33.40	48.19	5.20%	15.38%	34.39%	5.00%	87.88%
	Mistral	Sparse	23.20	33.21	59.00%	13.22%	37.56%	48.60%	54.71%
		Dense	35.20	45.82	40.00%	21.50%	44.33%	50.20%	56.39%
		ChatGPT	32.60	47.49	14.40%	16.67%	35.28%	41.00%	64.24%
TriviaQA	Davinci003	Sparse	64.60	70.19	15.60%	19.23%	72.99%	69.00%	77.15%
		Dense	69.60	75.31	10.00%	30.00%	74.00%	74.40%	81.49%
		ChatGPT	67.40	75.43	2.00%	10.00%	68.57%	72.20%	81.00%
	ChatGPT	Sparse	62.60	69.98	23.00%	34.78%	70.91%	79.80%	73.29%
		Dense	66.20	74.75	18.20%	39.56%	72.13%	82.40%	75.73%
		ChatGPT	65.00	74.44	3.00%	13.33%	66.60%	90.40%	74.34%
	GPT-4	Sparse	66.20	75.99	12.40%	35.48%	70.55%	83.40%	76.79%
		Dense	69.00	78.01	7.20%	30.56%	71.98%	85.80%	76.51%
		ChatGPT	66.40	76.33	2.60%	15.38%	67.76%	83.40%	73.39%
	LLaMA2	Sparse	51.00	59.51	35.60%	40.45%	56.83%	13.20%	70.19%
		Dense	58.60	66.57	33.40%	40.72%	67.57%	11.40%	75.00%
		ChatGPT	63.00	71.76	2.80%	28.57%	63.99%	18.20%	79.35%
	Mistral	Sparse	52.20	59.55	30.40%	20.39%	66.09%	59.20%	68.75%
		Dense	57.40	65.59	24.20%	26.45%	67.28%	59.80%	72.53%
		ChatGPT	62.20	71.72	3.60%	16.67%	63.90%	55.20%	77.76%
HotpotQA	Davinci003	Sparse	31.20	40.95	27.20%	14.71%	37.36%	31.20%	76.84%
		Dense	26.80	35.89	37.00%	13.51%	34.60%	35.20%	76.89%
		ChatGPT	28.20	39.34	8.20%	12.20%	29.63%	33.40%	77.37%
	ChatGPT	Sparse	29.60	41.28	50.60%	17.39%	42.11%	51.80%	54.90%
		Dense	26.40	35.75	58.40%	14.38%	43.27%	48.20%	56.10%
		ChatGPT	26.40	38.30	11.20%	7.14%	28.83%	68.20%	48.24%
	GPT-4	Sparse	36.00	47.71	25.60%	14.84%	43.28%	43.40%	64.90%
		Dense	31.80	43.92	37.00%	17.30%	40.32%	46.00%	60.34%
		ChatGPT	29.80	41.67	8.80%	6.82%	32.02%	48.40%	68.55%
	LLaMA2	Sparse	24.00	33.45	45.60%	16.67%	30.15%	8.60%	58.89%
		Dense	21.60	31.17	57.00%	13.68%	32.09%	7.60%	66.99%
		ChatGPT	25.80	37.56	11.80%	22.03%	26.30%	7.20%	82.53%
	Mistral	Sparse	25.00	35.49	52.40%	13.74%	37.39%	42.40%	62.42%
		Dense	23.60	32.70	59.80%	13.38%	38.81%	45.60%	59.75%
		ChatGPT	26.20	37.93	14.00%	12.86%	28.37%	37.60%	70.04%

Q2: What Effect Does Retrieval Augmentation Have on LLMs?

Main Findings:

Accuracy of LLMs' self-assessment improves after incorporating supporting documents

Right/¬G rises significantly as ***Right/G*** declines, showing improved accuracy in prior judgement.

Eval-Acc rises and ***Eval-Right*** decreases and is more consistent with EM metric

Retrieval augmentation improves LLM's ability to perceive their factual knowledge boundaries

Datasets	LLMs	Retrieval Source	QA		Priori Judgement			Posteriori Judgement	
			EM	F1	Give-up	Right/G	Right/¬G	Eval-Right	Eval-Acc
NQ	Davinci003	Sparse	27.80	38.29	21.20%	12.26%	31.98%	39.40%	67.94%
		Dense	39.00	51.27	12.80%	14.06%	42.66%	46.40%	71.43%
		ChatGPT	34.00	47.36	6.20%	6.45%	35.82%	46.00%	71.54%
	ChatGPT	Sparse	28.40	41.10	42.40%	17.92%	36.11%	67.00%	48.77%
		Dense	39.40	52.65	26.60%	18.05%	47.14%	68.80%	53.56%
		ChatGPT	32.20	47.37	7.40%	2.70%	34.56%	78.80%	49.90%
	GPT-4	Sparse	34.20	45.81	28.20%	14.18%	42.06%	57.20%	48.48%
		Dense	43.60	56.36	12.60%	15.87%	47.60%	66.40%	50.86%
		ChatGPT	34.40	48.56	4.20%	4.76%	35.70%	69.80%	48.69%
	LLaMA2	Sparse	23.00	34.14	32.80%	15.85%	26.49%	6.00%	75.00%
		Dense	33.40	45.39	24.80%	20.16%	37.77%	5.20%	73.08%
		ChatGPT	33.40	48.19	5.20%	15.38%	34.39%	5.00%	87.88%
	Mistral	Sparse	23.20	33.21	59.00%	13.22%	37.56%	48.60%	54.71%
		Dense	35.20	45.82	40.00%	21.50%	44.33%	50.20%	56.39%
		ChatGPT	32.60	47.49	14.40%	16.67%	35.28%	41.00%	64.24%
TriviaQA	Davinci003	Sparse	64.60	70.19	15.60%	19.23%	72.99%	69.00%	77.15%
		Dense	69.60	75.31	10.00%	30.00%	74.00%	74.40%	81.49%
		ChatGPT	67.40	75.43	2.00%	10.00%	68.57%	72.20%	81.00%
	ChatGPT	Sparse	62.60	69.98	23.00%	34.78%	70.91%	79.80%	73.29%
		Dense	66.20	74.75	18.20%	39.56%	72.13%	82.40%	75.73%
		ChatGPT	65.00	74.44	3.00%	13.33%	66.60%	90.40%	74.34%
	GPT-4	Sparse	66.20	75.99	12.40%	35.48%	70.55%	83.40%	76.79%
		Dense	69.00	78.01	7.20%	30.56%	71.98%	85.80%	76.51%
		ChatGPT	66.40	76.33	2.60%	15.38%	67.76%	83.40%	73.39%
	LLaMA2	Sparse	51.00	59.51	35.60%	40.45%	56.83%	13.20%	70.19%
		Dense	58.60	66.57	33.40%	40.72%	67.57%	11.40%	75.00%
		ChatGPT	63.00	71.76	2.80%	28.57%	63.99%	18.20%	79.35%
	Mistral	Sparse	52.20	59.55	30.40%	20.39%	66.09%	59.20%	68.75%
		Dense	57.40	65.59	24.20%	26.45%	67.28%	59.80%	72.53%
		ChatGPT	62.20	71.72	3.60%	16.67%	63.90%	55.20%	77.76%
HotpotQA	Davinci003	Sparse	31.20	40.95	27.20%	14.71%	37.36%	31.20%	76.84%
		Dense	26.80	35.89	37.00%	13.51%	34.60%	35.20%	76.89%
		ChatGPT	28.20	39.34	8.20%	12.20%	29.63%	33.40%	77.37%
	ChatGPT	Sparse	29.60	41.28	50.60%	17.39%	42.11%	51.80%	54.90%
		Dense	26.40	35.75	58.40%	14.38%	43.27%	48.20%	56.10%
		ChatGPT	26.40	38.30	11.20%	7.14%	28.83%	68.20%	48.24%
	GPT-4	Sparse	36.00	47.71	25.60%	14.84%	43.28%	43.40%	64.90%
		Dense	31.80	43.92	37.00%	17.30%	40.32%	46.00%	60.34%
		ChatGPT	29.80	41.67	8.80%	6.82%	32.02%	48.40%	68.55%
	LLaMA2	Sparse	24.00	33.45	45.60%	16.67%	30.15%	8.60%	58.89%
		Dense	21.60	31.17	57.00%	13.68%	32.09%	7.60%	66.99%
		ChatGPT	25.80	37.56	11.80%	22.03%	26.30%	7.20%	82.53%
	Mistral	Sparse	25.00	35.49	52.40%	13.74%	37.39%	42.40%	62.42%
		Dense	23.60	32.70	59.80%	13.38%	38.81%	45.60%	59.75%
		ChatGPT	26.20	37.93	14.00%	12.86%	28.37%	37.60%	70.04%

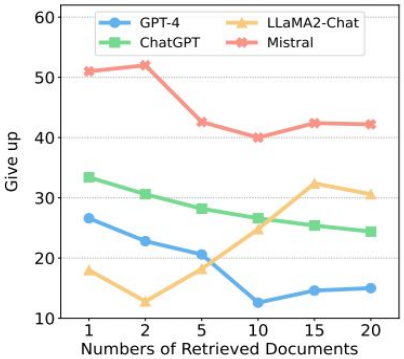
Q2: What Effect Does Retrieval Augmentation Have on LLMs?

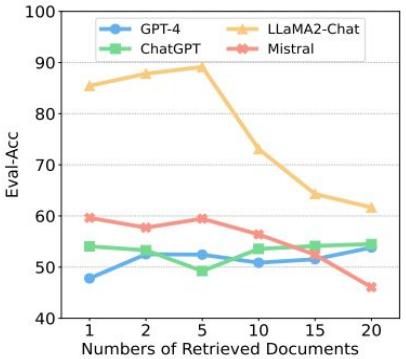
Main Findings:

As the number increases, the **QA performance (EM)** gradually increases until reaching a certain threshold.

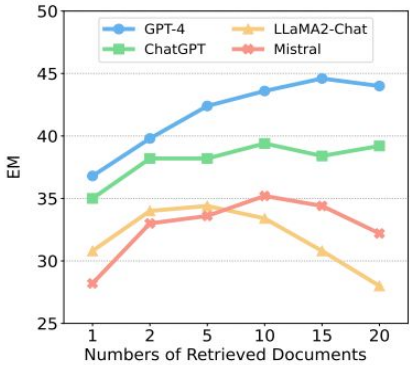
Performance improvement is not solely attributed to the **recall rate.**

LLMs seem to be insensitive to the **ordering** of retrieved documents.





Increasing the number of supporting documents improves the performance of LLMs below a model-specific threshold.



Q3: How do Different Relevance Supporting Document Affect LLMs?

Retrieved documents vary in relevance and correctness The study examines how different types of supporting documents affect LLM performance and confidence.

Four Document Types:

Type	Definition	Sampling Source	Contains Correct Answer
Golden	**Highly relevant** to the question and contain at least one **correct answer.**	Sampled **top-down** from the **top 100** results	Yes
High-related incorrect	**Highly relevant** to the question but contain **no correct answer**.	Sampled **top-down** from the **top 100** results	No
Weak-related incorrect	**Weakly relevant** to the question and contain **no correct answer**.	**Randomly** sampled from the **top 100** results	No
Random incorrect	**Irrelevant** to the question and contain **no correct answer**.	Randomly sampled from the **entire Wikipedia corpus**.	No

Q3: How do Different Relevance Supporting Document Affect LLMs?

Main Findings

LLMs exhibit higher confidence when equipped with high-quality supporting documents.

LLMs heavily rely on the given supporting documents

Relevance ≠ Reliability: LLMs trust relevance signals more than factual correctness.

The relevance of supporting documents significantly influences LLMs' reliance on supporting documents.

LLMs	Supporting Doc	EM	F1	Give-up	Right/G	Right/¬G	Eval-Right	Eval-Acc
Davinci-003	None	27.20	36.20	29.20%	13.70%	32.77%	72.80%	45.01%
	Golden	50.60	62.93	15.80%	15.19%	57.24%	52.00%	71.08%
	Retrieved	39.00	51.27	12.80%	14.06%	42.66%	46.40%	71.43%
	High-related	10.20	20.66	18.00%	8.89%	10.49%	28.40%	57.89%
	Weak-related	11.80	19.69	41.40%	10.63%	12.63%	20.80%	61.71%
	Random	23.00	30.82	88.40%	20.59%	41.38%	19.40%	66.26%
ChatGPT	None	33.40	45.32	57.40%	26.48%	42.72%	84.40%	43.40%
	Golden	50.00	64.28	22.60%	23.01%	57.88%	75.20%	53.24%
	Retrieved	39.40	52.65	26.60%	18.05%	47.14%	68.80%	53.56%
	High-related	16.20	28.20	42.00%	13.81%	17.93%	56.20%	47.82%
	Weak-related	18.40	29.86	60.20%	16.61%	21.11%	49.80%	46.21%
	Random	24.80	35.35	91.00%	23.30%	40.00%	29.80%	48.80%
GPT-4	None	34.60	48.72	15.20%	9.21%	39.15%	90.20%	38.87%
	Golden	53.60	67.36	15.60%	20.51%	59.72%	73.00%	53.58%
	Retrieved	43.60	56.36	12.60%	15.87%	47.60%	66.40%	50.86%
	High-related	21.60	35.13	39.20%	24.49%	19.74%	62.40%	47.21%
	Weak-related	24.40	34.83	61.20%	24.18%	24.74%	60.00%	44.96%
	Random	34.40	45.43	71.40%	25.49%	56.64%	54.00%	42.50%
LLaMA2	None	16.60	24.26	6.60%	3.03%	17.56%	58.40%	46.74%
	Golden	48.60	61.33	18.40%	29.35%	52.94%	7.60%	72.12%
	Retrieved	33.40	45.39	24.80%	20.16%	37.77%	5.20%	73.08%
	High-related	9.40	19.07	30.60%	8.50%	9.80%	4.80%	67.86%
	Weak-related	8.80	16.00	50.20%	9.16%	8.43%	6.20%	59.09%
	Random	13.20	19.34	93.40%	13.28%	12.12%	5.80%	58.97%
Mistral	None	11.20	19.30	69.80%	8.02%	18.54%	78.20%	31.65%
	Golden	47.80	60.93	39.60%	28.28%	60.60%	50.20%	58.67%
	Retrieved	35.20	45.82	40.00%	21.50%	44.33%	50.20%	56.39%
	High-related	5.60	14.23	58.40%	4.11%	7.69%	47.60%	53.35%
	Weak-related	5.40	11.21	76.80%	3.91%	10.34%	46.60%	53.48%
	Random	12.40	18.40	98.40%	11.79%	50.00%	64.80%	33.57%

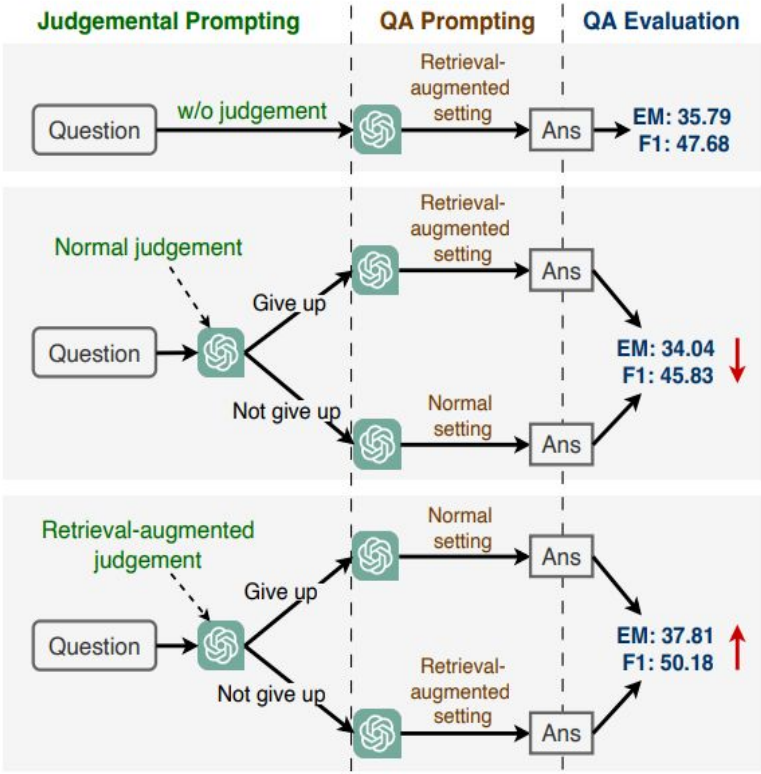
Dynamic Retrieval Augmentation

Key Idea: Let LLMs decide when to retrieve based on their own confidence (priori judgement).

w/o Judgement: Always uses retrieval (fixed RAG).
→ Stable, but not adaptive.

Normal Judgement: Model decides w/o external info.
→ Poor decision-making → performance dropped

Retrieval-augmented Judgement: Model decides with retrieval context.
→ More accurate judgement → performance improved



Conclusion

LLMs' Knowledge Boundary Awareness is Weak → They are often overconfident or misled by retrieved info.

Retrieval Augmentation Helps → Best results depend on retrieval model type, doc number, and LLM scale.

Document Relevance Matters Most → LLMs trust relevance > truth — high-related wrong docs are most misleading.

Dynamic RAG is Promising → LLMs can "know when to look up information."

Limitations

API dependency: Closed-source models (e.g., GPT-4, ChatGPT) change over time, affecting reproducibility.

Method robustness: The evaluation framework remains applicable to future LLMs — so the risk of API does not affect the significance of the contribution.

Paper 4: Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, Hannaneh Hajishirzi

Presenters: *Levi Kaster, Lars S., Zilong Wang*

Standard RAG Methods have pitfalls that must be addressed.

- Documents often retrieved according to document similarity, not according helpfulness of documents towards addressing input prompt
  - Retrieved documents may include off-topic information degrading quality of responses

- Models not explicitly trained to leverage facts from the extracted documents
  - Model may completely ignore retrieved information from document
  - Extracted documents may be leveraged in way that does not address user questions

Alternative Methods that incorporate RAG

1) Adaptive Document Retrieval

2) Filter out and/or summarize retrieved documents before passing to language model

3) RLHF (Reinforcement Learning from Human Feedback) to align responses with human preference through reward models

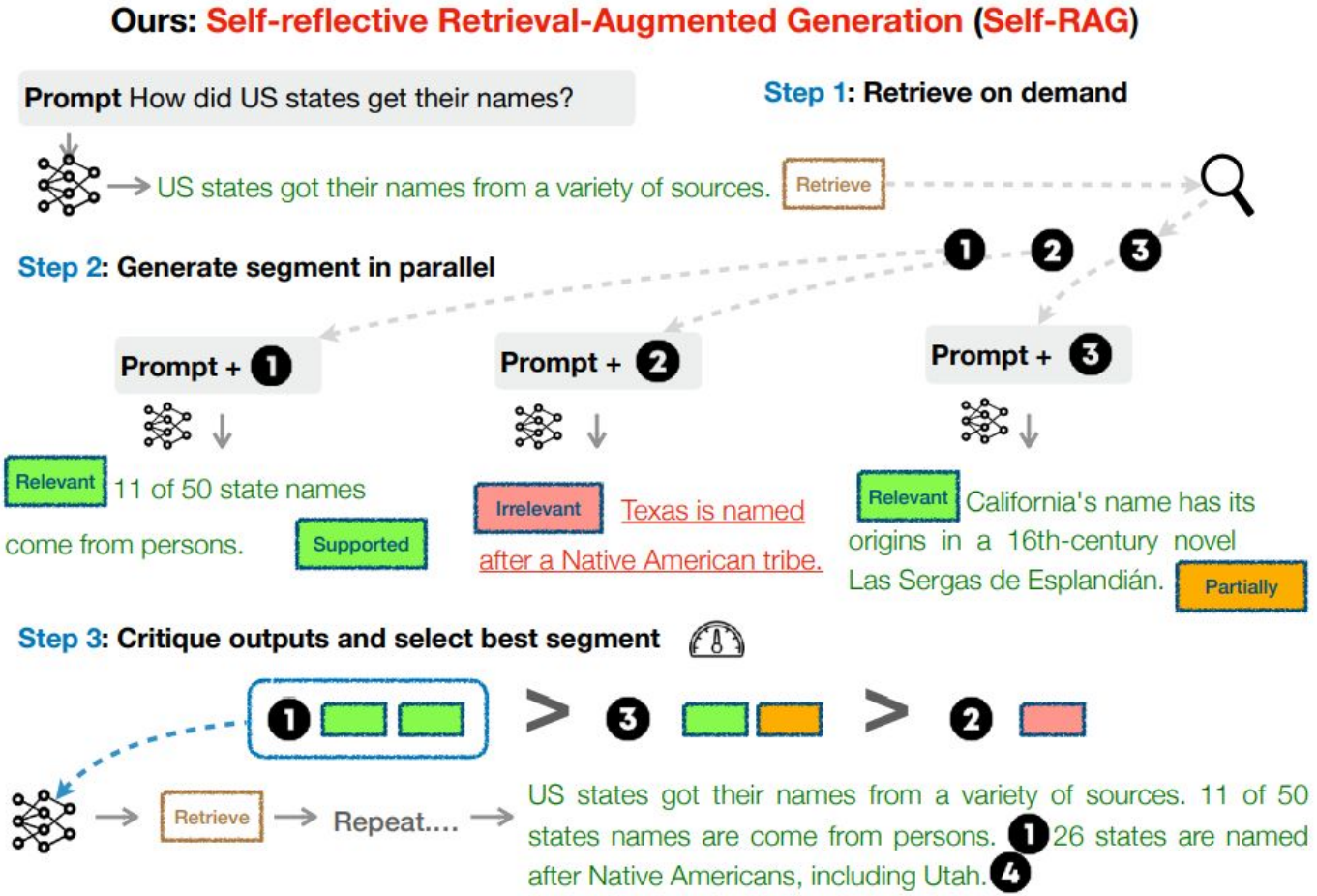
Self-RAG Incorporates Aspects of Each!

1) **On-demand Retrieval**

2) **Relevance of retrieved text is determined**

3) **Critic tokens to determine usefulness of documents, and how much retrieved passage 'supports' generated response.**

Self-RAG Outline



Methods

Type	Input	Output	Definitions
Retrieve	x / x, y	{yes, no, continue}	Decides when to retrieve with $\mathcal{R}$
IsRel	x, d	{**relevant**, irrelevant}	d provides useful information to solve x.
IsSup	x, d, y	{**fully supported**, partially supported, no support}	All of the verification-worthy statement in y is supported by d.
IsUse	x, y	{**5**, 4, 3, 2, 1}	y is a useful response to x.

Algorithm 1 SELF-RAG Inference

Require: Generator LM $\mathcal{M}$, Retriever $\mathcal{R}$, Large-scale passage collections $\{d_1, \ldots, d_N\}$

Input: input prompt x and preceding generation $y_{<t}$, **Output:** next output segment y_t
$\mathcal{M}$ predicts Retrieve given $(x, y_{<t})$
if Retrieve == Yes **then**
    Retrieve relevant text passages $\mathbf{D}$ using $\mathcal{R}$ given (x, y_{t-1}) ▷ Retrieve
    $\mathcal{M}$ predicts IsRel given x, d and y_t given $x, d, y_{<t}$ for each $d \in \mathbf{D}$ ▷ Generate
    $\mathcal{M}$ predicts IsSup and IsUse given x, y_t, d for each $d \in \mathbf{D}$ ▷ Critique
    Rank y_t based on IsRel , IsSup , IsUse ▷ Detailed in Section 3.3
else if Retrieve == No **then**
    $\mathcal{M}_{gen}$ predicts y_t given x ▷ Generate
    $\mathcal{M}_{gen}$ predicts IsUse given x, y_t ▷ Critique

Self-RAG training

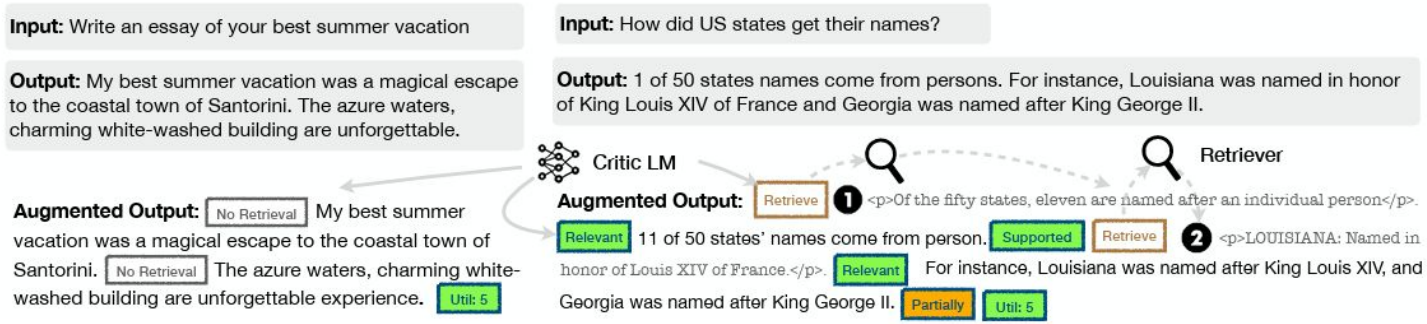


Figure 2: SELF-RAG training examples. The left example does not require retrieval while the right one requires retrieval; thus, passages are inserted. More examples are in Appendix Table 4.

GPT-4 was used to generate reflection tokens for a supervised dataset for distilling their own Critic model. (Open-Instruct: includes many instruct datasets)

Data Collection For Generator

The distilled Critic model creates data to train the generator.

- If retrieval is required adds **Retrieve** and K docs are retrieved
- For each retrieved doc predicts is relevant **ISREL**
- If relevant predicts if supporting **ISSUP**
  - Relevant and supporting tokens are added to the end of each doc
- Finally predicts utility of the segment. **ISUSE**

Generator learning

Uses the standard next token objective

$$\max_{\mathcal{M}} \mathbb{E}_{(x,y,r)\sim\mathcal{D}_{gen}} \log p_{\mathcal{M}}(y, r|x).$$

x: input, y: correct output, r: correct reflection tokens

Unlike the critic model, the generator learns both the target output and the reflection tokens. Retrieved text chunks are masked out during training

Tree decoding with critique tokens

After creating a reflection token they perform beam search with critic score (S). Branches at location t are selected with:

$$f(y_t, d, \boxed{\text{Critique}}) = p(y_t|x, d, y_{<t})) + \mathcal{S}(\boxed{\text{Critique}}), \text{where}$$

$$\mathcal{S}(\boxed{\text{Critique}}) = \sum_{G \in \mathcal{G}} w^G s_t^G \text{ for } \mathcal{G} = \{\boxed{\text{IsRel}}, \boxed{\text{IsSup}}, \boxed{\text{IsUse}}\},$$

w: weight of each critique token, s is the critic score assigned by the **Critic model.**

Results & Analysis (Zilong)

Baselines with retrieval generally performs better than ones without retrieval

SELF-RAG has strong overall gains across 6 benchmarks

On PubHealth and ARC, retrieval baselines do not improve performance notably from their noretrieval counterparts

SELF-RAG 7B occasionally outperforms 13B

	Short-form		Closed-set		Long-form generations (with citations)					
LM	PopQA (acc)	TQA (acc)	Pub (acc)	ARC (acc)	Bio (FS)	ASQA (em)	ASQA (rg)	ASQA (mau)	(pre)	(rec)
LMs with proprietary data										
Llama2-c$_{13B}$	20.0	59.3	49.4	38.4	55.9	22.4	29.6	28.6	–	–
Ret-Llama2-c$_{13B}$	51.8	59.8	52.1	37.9	79.9	32.8	34.8	43.8	19.8	36.1
ChatGPT	29.3	74.3	70.1	75.3	71.8	35.3	36.2	68.8	–	–
Ret-ChatGPT	50.8	65.7	54.7	75.3	–	40.7	39.9	79.7	65.1	76.6
Perplexity.ai	–	–	–	–	71.2	–	–	–	–	–
Baselines without retrieval										
Llama2$_{7B}$	14.7	30.5	34.2	21.8	44.5	7.9	15.3	19.0	–	–
Alpaca$_{7B}$	23.6	54.5	49.8	45.0	45.8	18.8	29.4	61.7	–	–
Llama2$_{13B}$	14.7	38.5	29.4	29.4	53.4	7.2	12.4	16.0	–	–
Alpaca$_{13B}$	24.4	61.3	55.5	54.9	50.2	22.9	32.0	70.6	–	–
CoVE$_{65B}$ *	–	–	–	–	71.2	–	–	–	–	–
Baselines with retrieval										
Toolformer*$_{6B}$	–	48.8	–	–	–	–	–	–	–	–
Llama2$_{7B}$	38.2	42.5	30.0	48.0	78.0	15.2	22.1	32.0	2.9	4.0
Alpaca$_{7B}$	46.7	64.1	40.2	48.0	76.6	30.9	33.3	57.9	5.5	7.2
Llama2-FT$_{7B}$	48.7	57.3	64.3	65.8	78.2	31.0	35.8	51.2	5.0	7.5
SAIL*$_{7B}$	–	–	69.2	48.4	–	–	–	–	–	–
Llama2$_{13B}$	45.7	47.0	30.2	26.0	77.5	16.3	20.5	24.7	2.3	3.6
Alpaca$_{13B}$	46.1	66.9	51.1	57.6	77.7	**34.8**	36.7	56.6	2.0	3.8
Our SELF-RAG $_{7B}$	54.9	66.4	72.4	67.3	**81.2**	30.0	35.7	**74.3**	66.9	67.8
Our SELF-RAG $_{13B}$	**55.8**	**69.3**	**74.5**	**73.1**	80.2	31.7	**37.0**	71.6	**70.3**	**71.3**

Results & Analysis

Training:

No Retriever R: Trains an LM using the standard instruction-following method without retrieved passages

No Critic C: Trains an LM that are always augmented with the Top 1 retrieved document without reflection tokens.

Inference-time:

No retrieval: Disables retrieval during inference

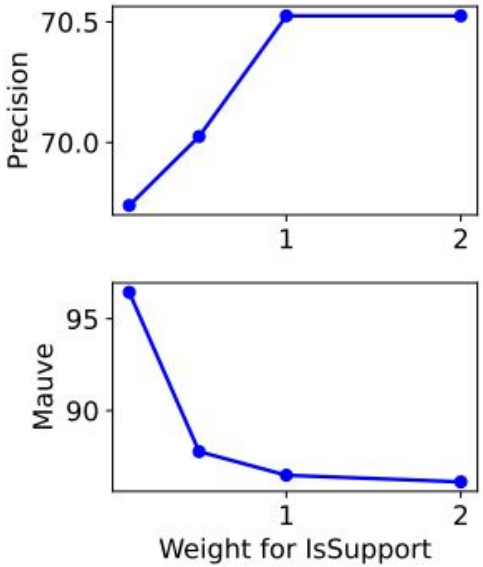
Hard constraints: Always retrieve when Retrieve = Yes

Retrieve top 1: Always retrieves and uses the top one document only

Remove IsSup: SELF-RAG without IsSup reflection token

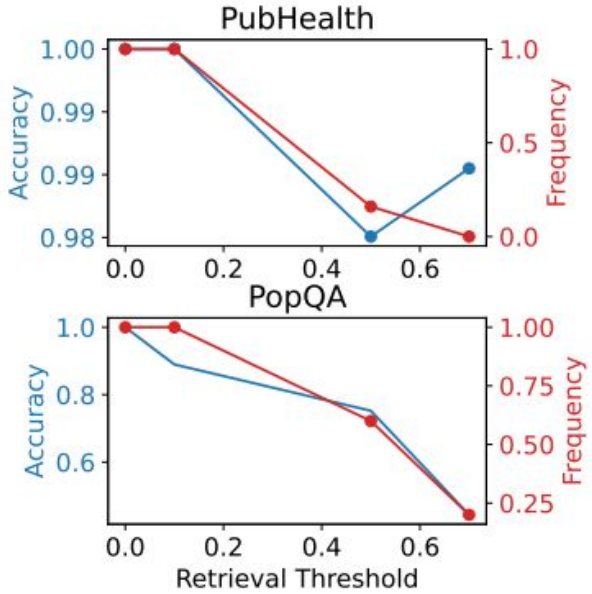
	PQA (acc)	Med (acc)	AS (em)
SELF-RAG (50k)	45.5	73.5	32.1
Training			
No Retriever $\mathcal{R}$	43.6	67.8	31.0
No Critic $\mathcal{C}$	42.6	72.0	18.1
Test			
No retrieval	24.7	73.0	–
Hard constraints	28.3	72.6	–
Retrieve top1	41.8	73.1	28.6
Remove IsSup	44.1	73.2	30.6

Results & Analysis



Adjusting weights of critique tokens can change the generation style.

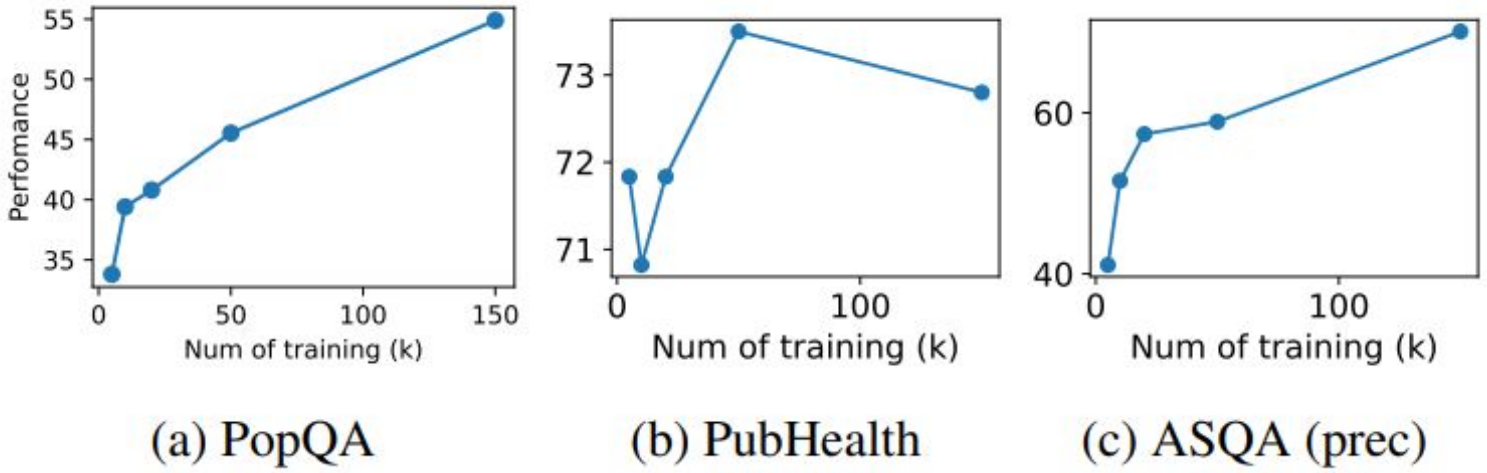
Higher IsSup *weight* → citation precision ↑
But lower fluency *(MAUVE ↓).*



Adaptive threshold controls retrieval frequency–accuracy trade-off.

Model's retrieval frequencies dramatically change on both datasets.

Results & Analysis



(a) PopQA (b) PubHealth (c) ASQA (prec)

Scaling up training data improves performance

Training SELF-RAG 7B on larger subsets (5k → 150k) yields steady improvements, especially on *PopQA* and *ASQA*

Results & Analysis

50 samples from *PopQA* and *Bio* evaluated by human annotators.

Metrics: S&P = Plausible (reasonable response) + Supported

Model-predicted IsRel & IsSup tokens largely align with human judgments.

SELF-RAG's reflection tokens reliably capture human notions of relevance and support.

	Pop	Bio.
S & P	92.5	70.0
ISREL	95.0	90.0
ISSUP	90.0	85.0

(d) Human evaluation on PopQA and Bio generation.

Conclusions

New framework:

The model learns to retrieve, generate, and critique using both normal tokens and reflection tokens.

Improved factuality and controllability:

Dynamically decides when to retrieve and evaluates its own outputs, enabling controllable generation without retraining.

Potential improvements:

The framework is scalable with data, and reflection tokens make reasoning steps transparent and verifiable.