Retrieval-augmented generation (RAG) & Tool-use models

CS 4804: Introduction to AI

Fall 2025

https://tuvllms.github.io/ai-fall-2025/

Tu Vu

VIRGINIA TECH

Logistics

- HW 2 due 11/18
- Final presentations: 12/4 & 12/9
  - Sign-up form available on Piazza later today

AI processing to space

Sundar Pichai @sundarpichai · Nov 4

Our TPUs are headed to space!

Inspired by our history of moonshots, from quantum computing to autonomous driving, Project Suncatcher is exploring how we could one day build scalable ML compute systems in space, harnessing more of the sun's power (which emits more power than 100

Show more

828 3.1K 17K 5.4M

NVIDIA @nvidia · 20h

Congratulations to @Starcloud_Inc1 on a successful launch. 🚀

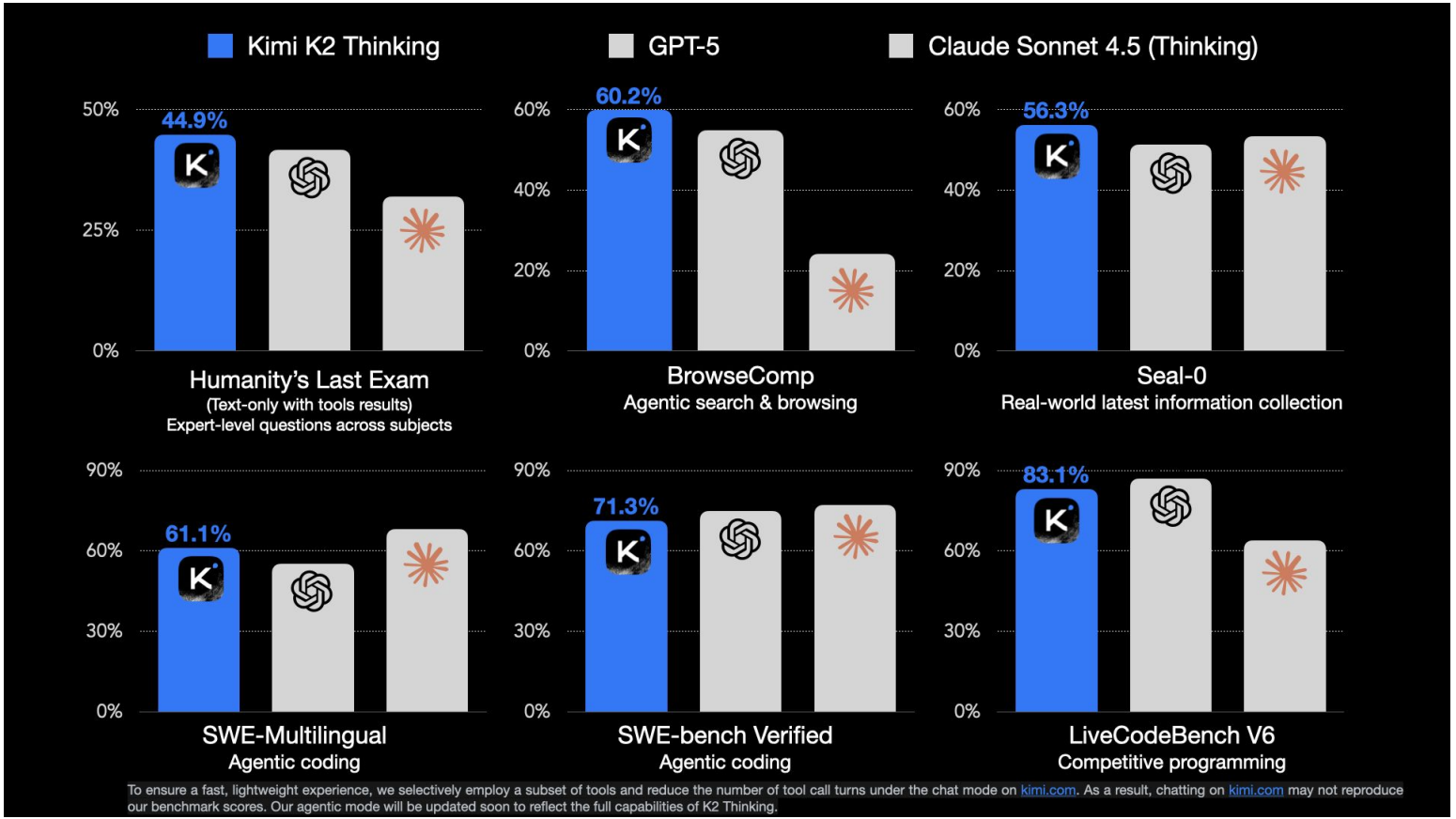
🛰️ Running inference in **space**, where the data is collected, allows insights to be delivered nearly instantaneously

⚡ Reduced response time delivers immense benefits for applications in wildfire detection and

Show more

45 154 872 111K

Kimi K2 Thinking



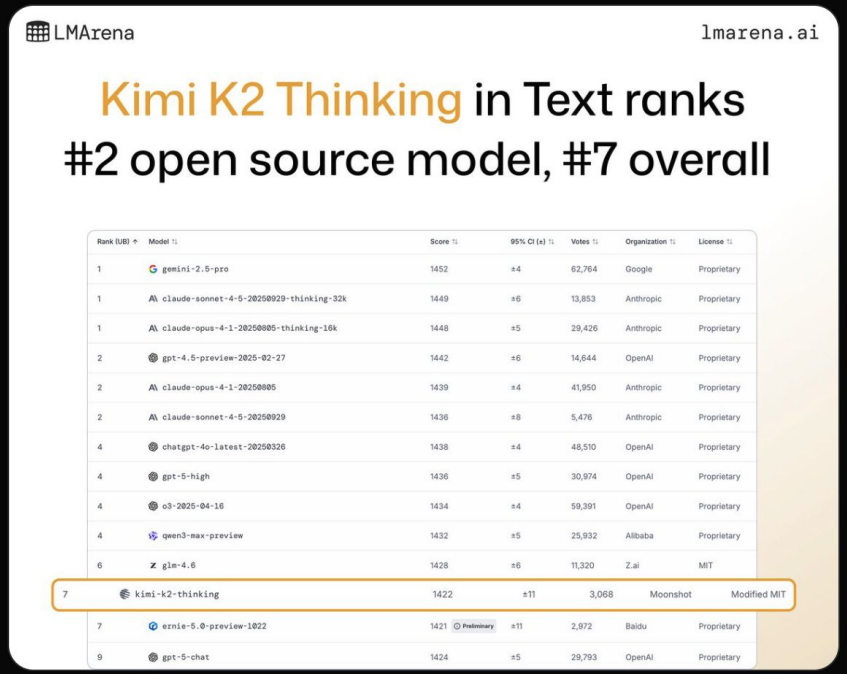
lmarena.ai @arena · 22h

🚨 Leaderboard Update!

Kimi K2 Thinking by @Kimi_Moonshot has landed on the Text leaderboard as the #2 open source model (MIT modified), tied for #7 overall. These are real-world results. With only a six-point difference with @Zai_org 's GLM 4.6, the competition is tight.

Kimi

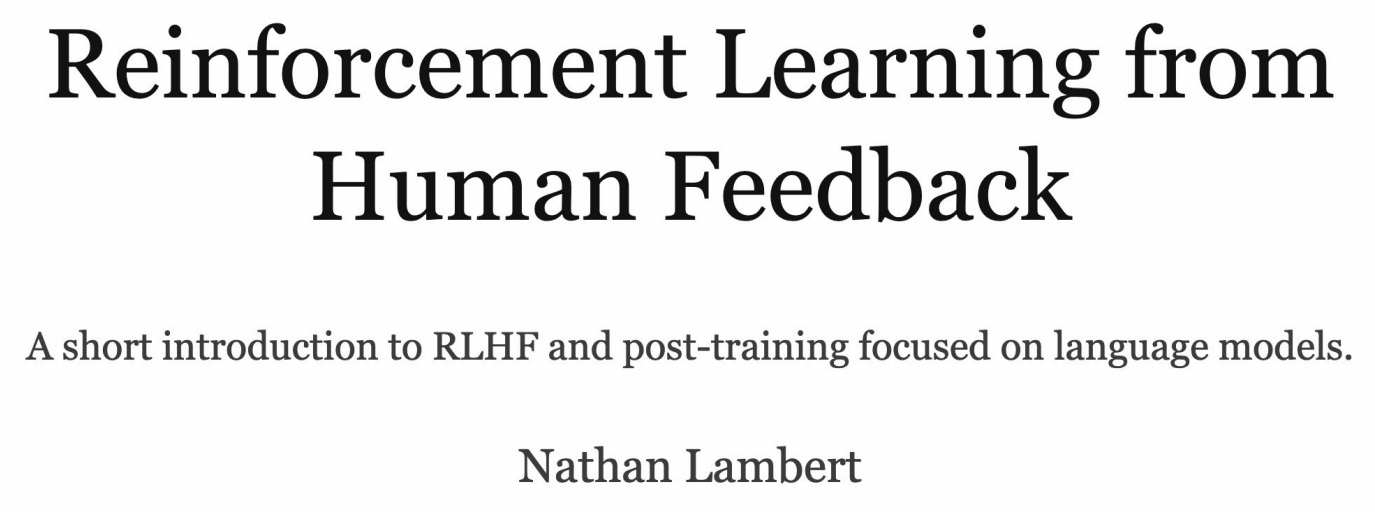
Show more



Rank (UB)	Model	Score	95% CI (±)	Votes	Organization	License
1	gemini-2.5-pro	1452	±4	62,764	Google	Proprietary
1	claude-sonnet-4-5-20250929-thinking-32k	1449	±6	13,853	Anthropic	Proprietary
1	claude-opus-4-1-20250805-thinking-16k	1448	±5	29,426	Anthropic	Proprietary
2	gpt-4.5-preview-2025-02-27	1442	±6	14,644	OpenAI	Proprietary
2	claude-opus-4-1-20250805	1439	±4	41,950	Anthropic	Proprietary
2	claude-sonnet-4-5-20250929	1436	±8	5,476	Anthropic	Proprietary
4	chatgpt-4o-latest-20250326	1438	±4	48,510	OpenAI	Proprietary
4	gpt-5-high	1436	±5	30,974	OpenAI	Proprietary
4	o3-2025-04-16	1434	±4	59,391	OpenAI	Proprietary
4	qwen3-max-preview	1432	±5	25,932	Alibaba	Proprietary
6	glm-4.6	1428	±6	11,320	Z.ai	MIT
7	kimi-k2-thinking	1422	±11	3,068	Moonshot	Modified MIT
7	ernie-5.0-preview-1022	1421 Preliminary	±11	2,972	Baidu	Proprietary
9	gpt-5-chat	1424	±5	29,793	OpenAI	Proprietary

Reinforcement learning material

- https://rlhfbook.com/
  - math/implementation descriptions of all the latest RL algorithms



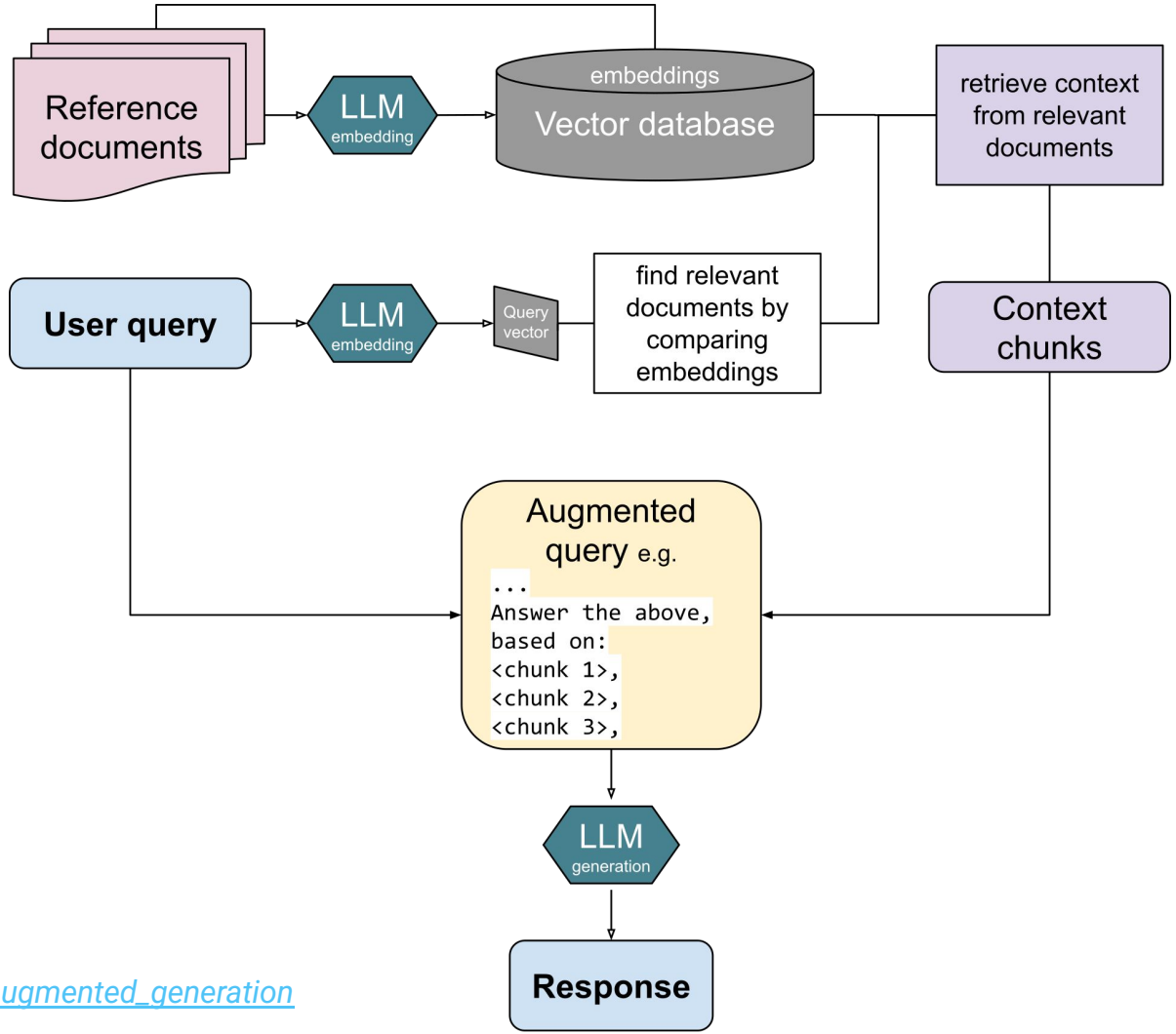
The Secrets to Building World-Class LLMs

- https://huggingface.co/spaces/HuggingFaceTB/smol-training-playbook

Retrieval-Augmented Generation (RAG)

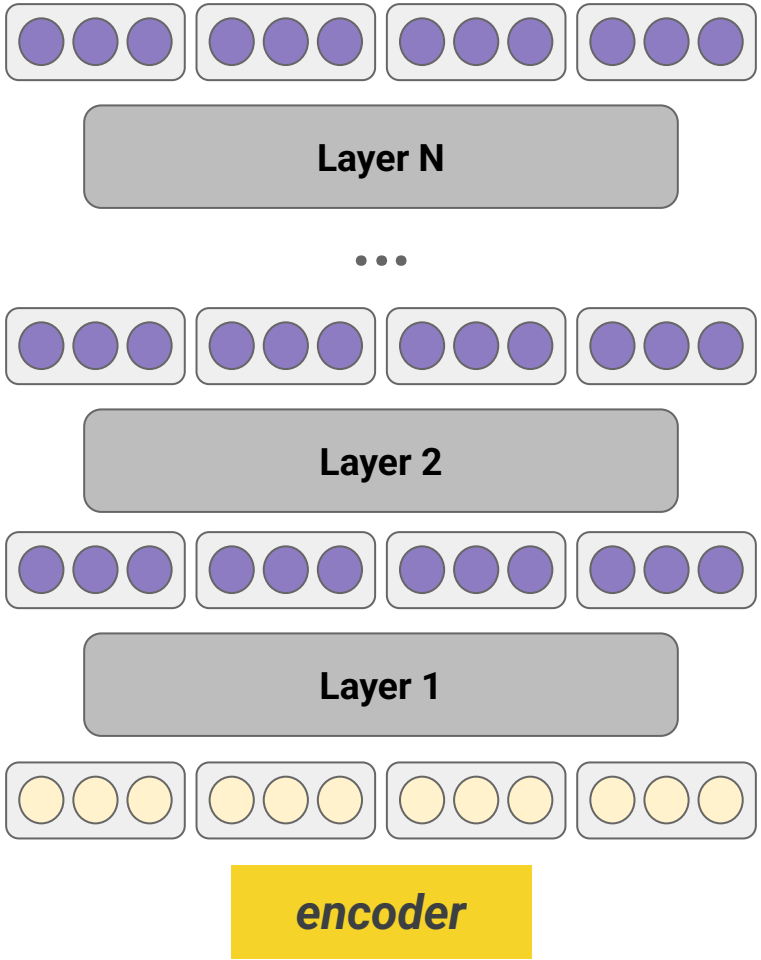
- Vanilla RAG
  - E.g., RAG, REALM
- RAG++
  - E.g., FreshLLMs, ReAct, Toolformer
- RAG + reasoning, agentic RAG, agentic memory
  - E.g., Self-RAG, Search-R1, Deep Research, A-Mem

Vanilla RAG

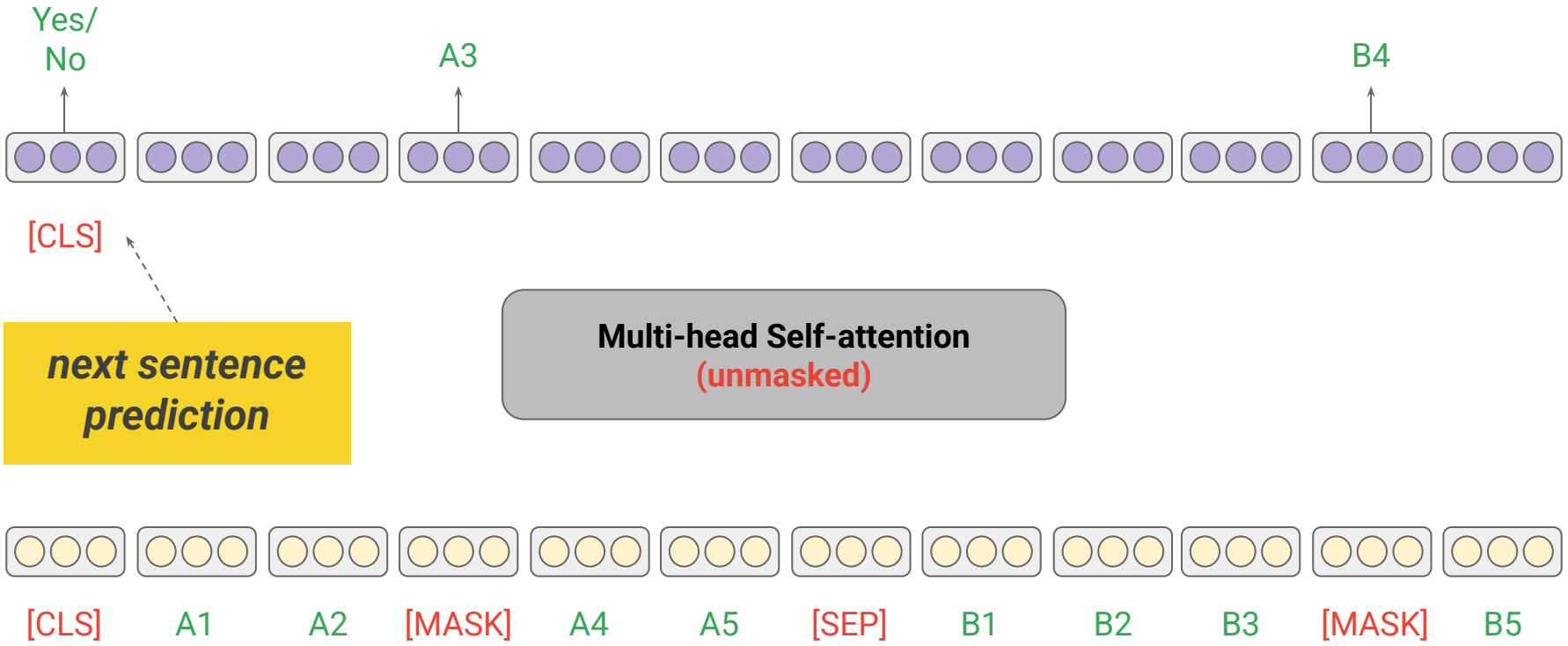


https://en.wikipedia.org/wiki/Retrieval-augmented_generation

Encoder (N layers)



BERT encoder



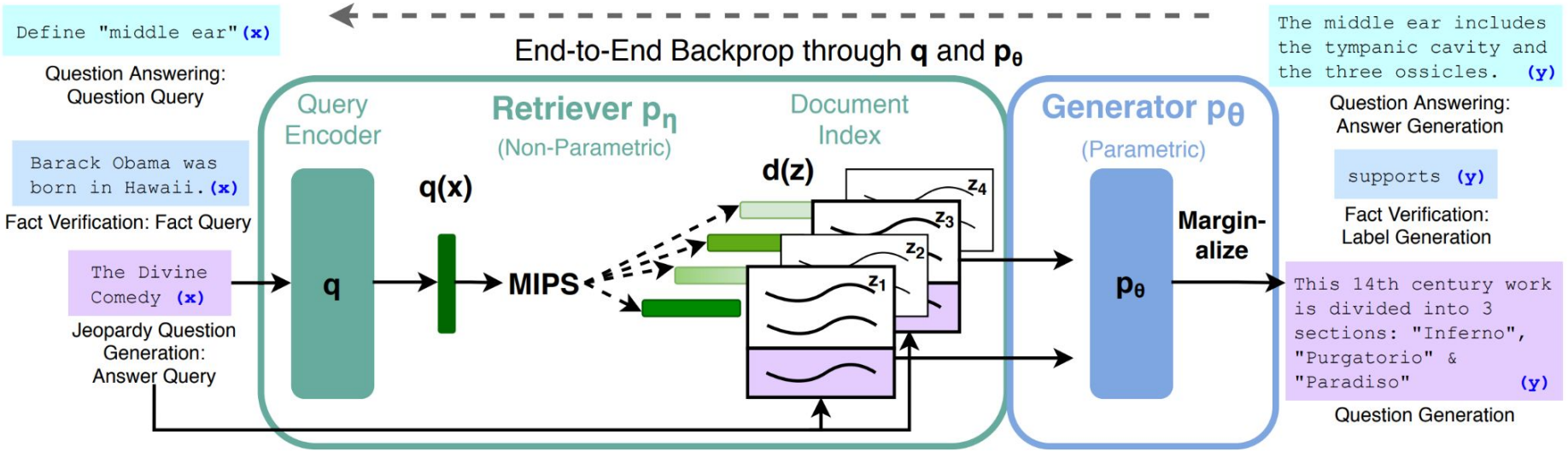
Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

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


Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and pθ
Query Encoder
q
q(x)
Retriever pη
(Non-Parametric)
MIPS
Document Index
d(z)
z1
z2
z3
z4
Generator pθ
(Parametric)
pθ
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Component	Type	Role	Trainable?
Index	Data structure	Stores document embeddings for nearest-neighbor search	No
Retriever	Model	Computes similarity between query and documents; retrieves top-K	Yes (query encoder usually)
Query encoder	Neural network	Encodes the input text into an embedding	Yes
Document encoder	Neural network	Encodes documents into embeddings used to build the index	Sometimes frozen
Generator	Seq2seq model	Generates text conditioned on the input and retrieved docs	Yes

Dense Passage Retrieval for Open-Domain Question Answering

Vladimir Karpukhin[*], **Barlas Oğuz**[*], **Sewon Min**[†], **Patrick Lewis**,
Ledell Wu, **Sergey Edunov**, **Danqi Chen**[‡], **Wen-tau Yih**
Facebook AI [†]University of Washington [‡]Princeton University
{vladk, barlaso, plewis, ledell, edunov, scottyih}@fb.com
sewon@cs.washington.edu
danqic@cs.princeton.edu

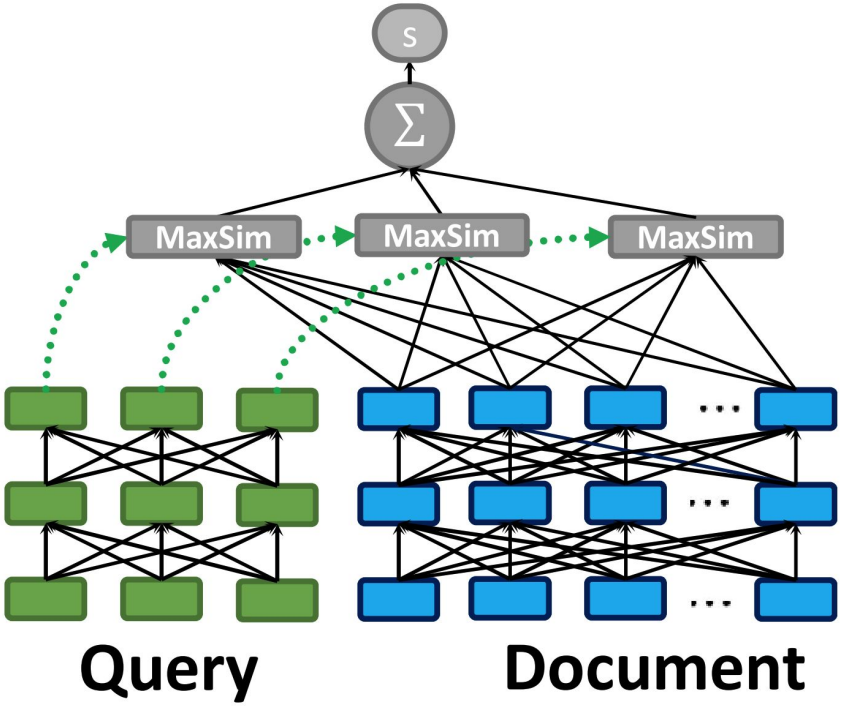
Dense Passage Retrieval (DPR)

Let $\mathcal{D} = \{(q_i, p_i^+, p_{i,1}^-, \ldots, p_{i,n_i}^-)\}_{i=1}^m$ denote the training set that contains m instances. Each instance includes a question q_i, one relevant (positive) passage p_i^+, and n_i irrelevant (negative) passages $p_{i,j}^-$.

The loss function is optimized as the negative log-likelihood of the positive passage:

$$\mathcal{L}(q_i, p_i^+, p_{i,1}^-, \ldots, p_{i,n_i}^-) = -\log \frac{\exp\big(\mathrm{sim}(q_i, p_i^+)\big)}{\exp\big(\mathrm{sim}(q_i, p_i^+)\big) + \sum_{j=1}^{n_i} \exp\big(\mathrm{sim}(q_i, p_{i,j}^-)\big)}.$$

ColBert's late interaction



(d) Late Interaction

(i.e., the proposed ColBERT)

https://arxiv.org/abs/2004.12832

Google DeepMind

On the Theoretical Limitations of Embedding-Based Retrieval

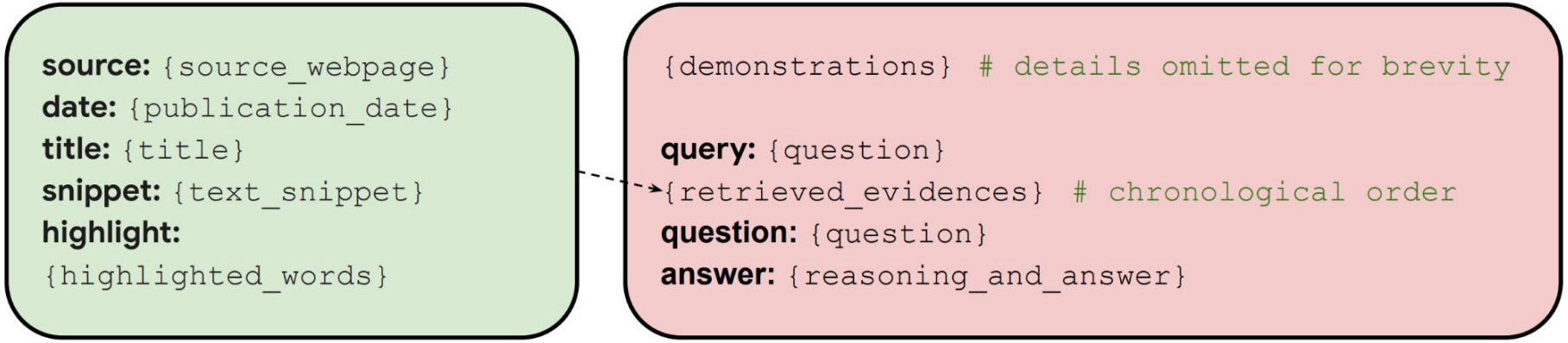
Orion Weller[*,1,2], **Michael Boratko**[1], **Iftekhar Naim**[1] **and Jinhyuk Lee**[1]
[1]Google DeepMind, [2]Johns Hopkins University

FRESHLLMS: REFRESHING LARGE LANGUAGE MODELS WITH SEARCH ENGINE AUGMENTATION

Tu Vu[1] **Mohit Iyyer**[2] **Xuezhi Wang**[1] **Noah Constant**[1] **Jerry Wei**[1]
Jason Wei[3*] **Chris Tar**[1] **Yun-Hsuan Sung**[1] **Denny Zhou**[1] **Quoc Le**[1] **Thang Luong**[1]

Google[1] University of Massachusetts Amherst[2] OpenAI[3]
freshllms@google.com

FreshPrompt



FreshPrompt uses few-shot in-context learning to teach a model to reason over retrieved evidences and figure out the right answer

Unleash the Power of Perplexity AI's Fresh Prompt Approach

Updated on Feb 28,2024

AI / LARGE LANGUAGE MODELS

How Perplexity's Online LLM Was Inspired by FreshLLMs Paper

We dig into the technology behind Perplexity's Copilot, which was inspired by the FreshLLMs paper that proposed search engine-augmented LLMs.

Jan 24th, 2024 4:00am by Janakiram MSV

Perplexity AI's Fresh Prompt Approach

e

Online LLMs

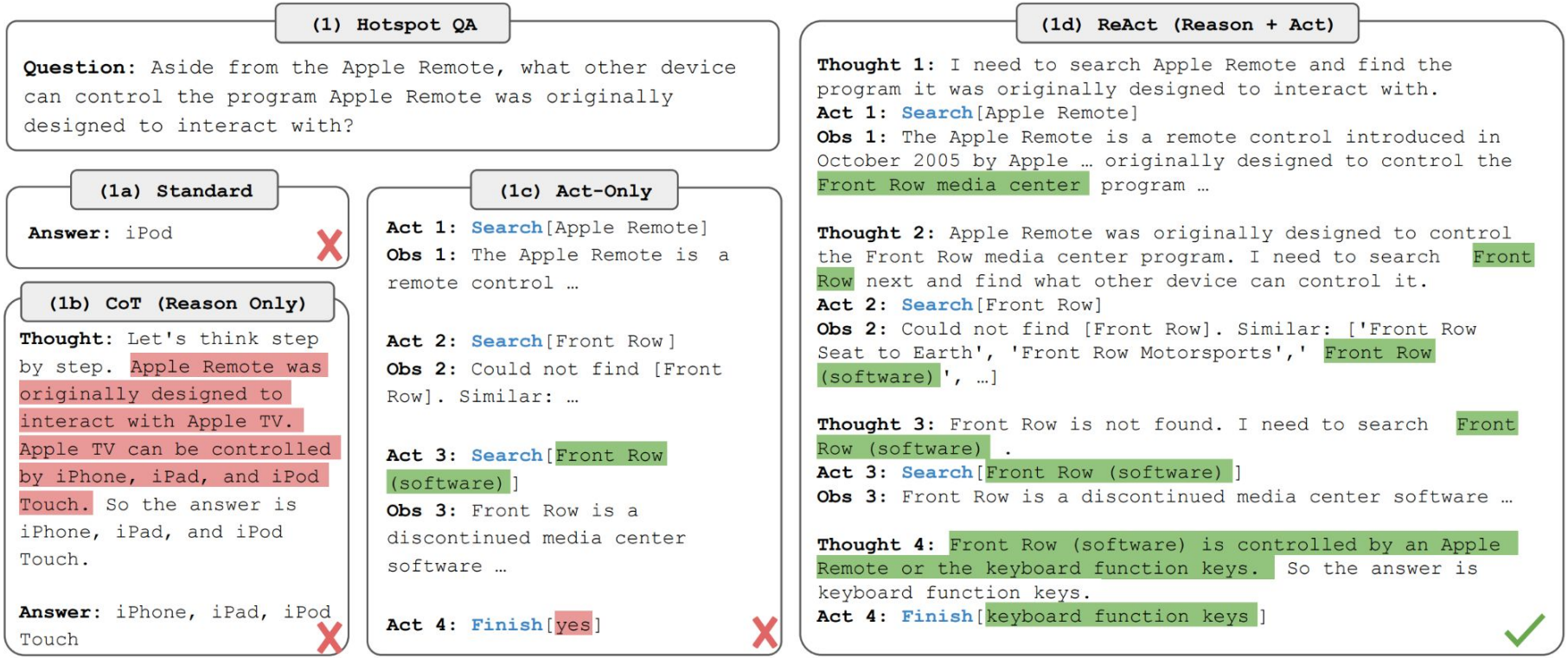
lels

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao[*,1], Jeffrey Zhao[2], Dian Yu[2], Nan Du[2], Izhak Shafran[2], Karthik Narasimhan[1], Yuan Cao[2]

[1]Department of Computer Science, Princeton University
[2]Google Research, Brain team
[1]{shunyuy,karthikn}@princeton.edu
[2]{jeffreyzhao,dianyu,dunan,izhak,yuancao}@google.com

ReAct


(1) Hotspot QA
Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?
(1a) Standard
Answer: iPod
(1b) CoT (Reason Only)
Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.
Answer: iPhone, iPad, iPod Touch
(1c) Act-Only
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control …
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: …
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …
Act 4: Finish[yes]
(1d) ReAct (Reason + Act)
Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …
Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software)', …]
Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …
Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]

Toolformer: Language Models Can Teach Themselves to Use Tools

Timo Schick **Jane Dwivedi-Yu** **Roberto Dessì**[†] **Roberta Raileanu**
Maria Lomeli **Luke Zettlemoyer** **Nicola Cancedda** **Thomas Scialom**

Meta AI Research [†]Universitat Pompeu Fabra

Exemplary predictions of Toolformer

The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

Using in-context learning to generate API calls

Use an LLM to annotate a huge language modeling dataset with potential API calls

Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing "[QA(question)]" where "question" is the question you want to ask. Here are some examples of API calls:

Input: Joe Biden was born in Scranton, Pennsylvania.

Output: Joe Biden was born in **[QA("Where was Joe Biden born?")]** Scranton, **[QA("In which state is Scranton?")]** Pennsylvania.

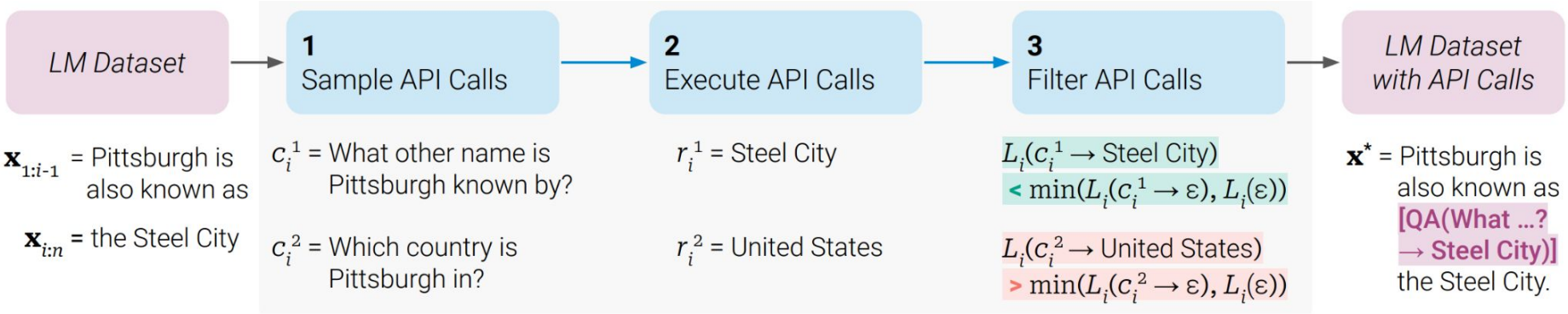
Input: Coca-Cola, or Coke, is a carbonated soft drink manufactured by the Coca-Cola Company.

Output: Coca-Cola, or **[QA("What other name is Coca-Cola known by?")]** Coke, is a carbonated soft drink manufactured by **[QA("Who manufactures Coca-Cola?")]** the Coca-Cola Company.

Input: x

Output:

Filtering out all API calls which do not reduce the loss over the next tokens



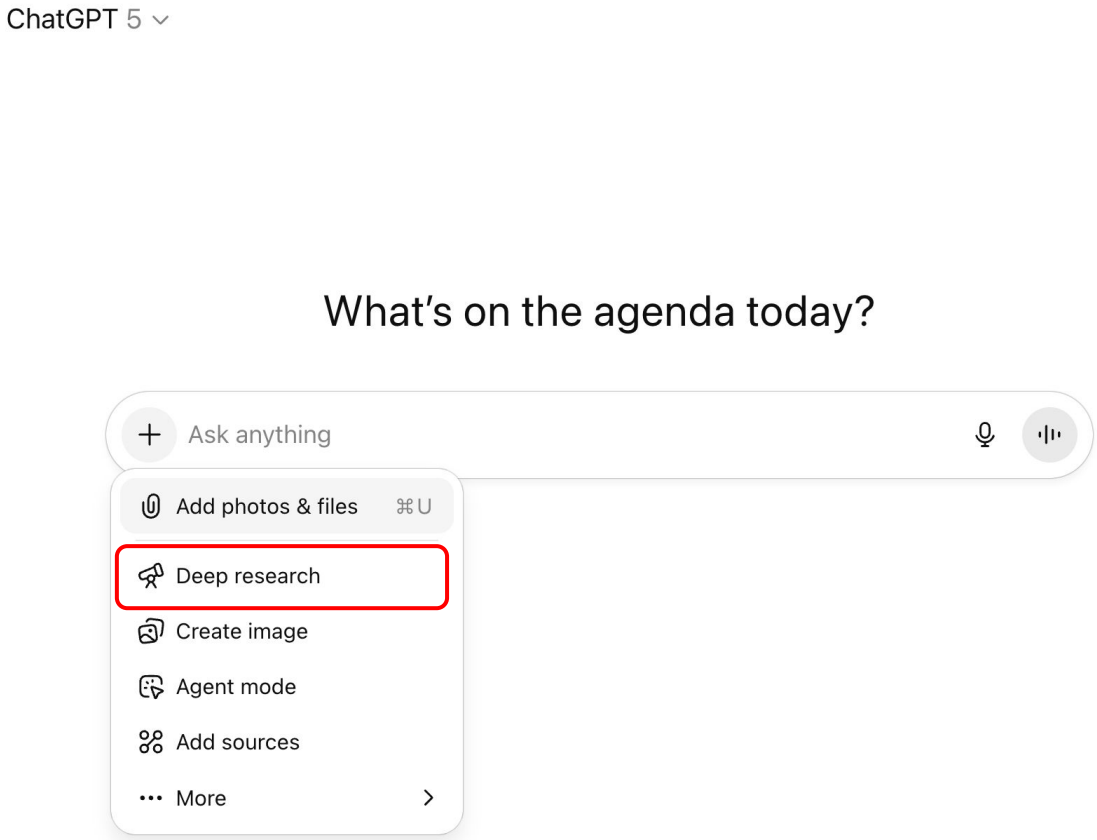
Intuitively, an API call is helpful if providing it with both the input and the output of this call makes it easier for the model to predict future tokens, compared to not receiving the API call at all, or receiving only its input

Toolformer (6.7B) achieves much stronger zero-shot results than OPT (66B) and GPT-3 (175B)

Model	ASDiv	SVAMP	MAWPS
GPT-J	7.5	5.2	9.9
GPT-J + CC	9.6	5.0	9.3
Toolformer (disabled)	14.8	6.3	15.0
Toolformer	**40.4**	**29.4**	**44.0**
OPT (66B)	6.0	4.9	7.9
GPT-3 (175B)	14.0	10.0	19.8

Model	WebQS	NQ	TriviaQA
GPT-J	18.5	12.8	43.9
GPT-J + CC	18.4	12.2	45.6
Toolformer (disabled)	18.9	12.6	46.7
Toolformer	**26.3**	**17.7**	**48.8**
OPT (66B)	18.6	11.4	45.7
GPT-3 (175B)	29.0	22.6	65.9

Deep Research



Deep Research (cont’d)

- https://openai.com/index/introducing-deep-research/
- An agent that uses reasoning to synthesize large amounts of online information and complete multi-step research tasks for you.

Deep Research (cont’d)

https://openai.com/index/introducing-deep-research/

Deep research is OpenAI's next agent that can do work for you independently—you give it a prompt, and ChatGPT will find, analyze, and synthesize hundreds of online sources to create a comprehensive report at the level of a research analyst. Powered by a version of the upcoming OpenAI o3 model that’s optimized for web browsing and data analysis, it leverages reasoning to search, interpret, and analyze massive amounts of text, images, and PDFs on the internet, pivoting as needed in reaction to information it encounters.

Deep Research (cont'd)

Deep research is built for people who do intensive knowledge work in areas like finance, science, policy, and engineering and need thorough, precise, and reliable research. It can be equally useful for discerning shoppers looking for hyper-personalized recommendations on purchases that typically require careful research, like cars, appliances, and furniture. Every output is fully documented, with clear citations and a summary of its thinking, making it easy to reference and verify the information. It is particularly effective at finding niche, non-intuitive information that would require browsing numerous websites. Deep research frees up valuable time by allowing you to offload and expedite complex, time-intensive web research with just one query.

Deep Research (cont'd)

Deep research independently discovers, reasons about, and consolidates insights from across the web. To accomplish this, it was trained on real-world tasks requiring browser and Python tool use, using the same reinforcement learning methods behind OpenAI o1, our first reasoning model. While o1 demonstrates impressive capabilities in coding, math, and other technical domains, many real-world challenges demand extensive context and information gathering from diverse online sources. Deep research builds on these reasoning capabilities to bridge that gap, allowing it to take on the types of problems people face in work and everyday life.

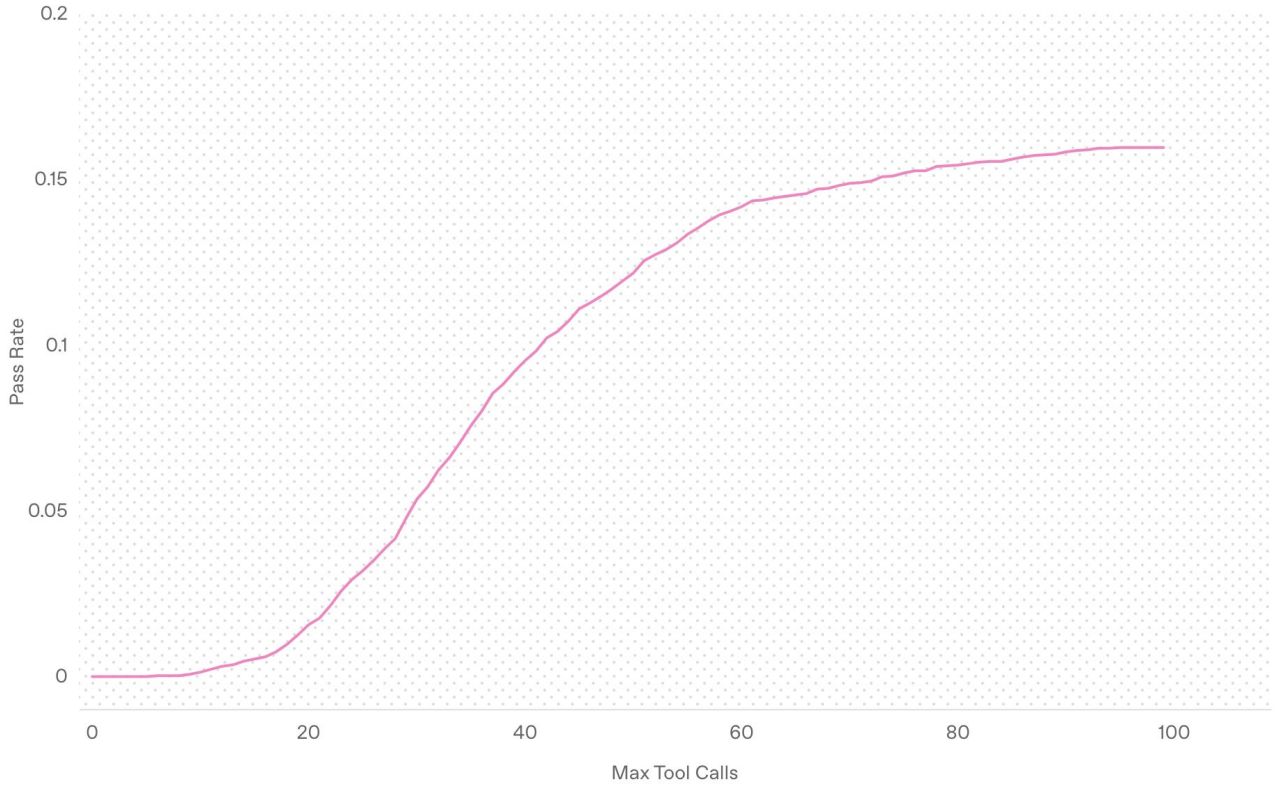
Deep Research on Humanity's Last Exam

Model	Accuracy (%)
GPT-4o	3.3
Grok-2	3.8
Claude 3.5 Sonnet	4.3
Gemini Thinking	6.2
OpenAI o1	**9.1**
DeepSeek-R1*	9.4
OpenAI o3-mini (medium)*	10.5
OpenAI o3-mini (high)*	13.0
OpenAI deep research**	**26.6**

* Model is not multi-modal, evaluated on text-only subset.

**with browsing + python tools

The more the model browses and thinks about what it is browsing, the better it does



Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning

Bowen Jin[1], **Hansi Zeng**[2], **Zhenrui Yue**[1], **Jinsung Yoon**[3], **Sercan Ö. Arık**[3], **Dong Wang**[1], **Hamed Zamani**[2], **Jiawei Han**[1]
[1] Department of Computer Science, University of Illinois at Urbana-Champaign
[2] Center for Intelligent Information Retrieval, University of Massachusetts Amherst
[3] Google Cloud AI Research
{bowenj4,zhenrui3,dwang24,hanj}@illinois.edu, {hzeng, zamani}@cs.umass.edu

Training template

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search>, and it will return the top searched results between <information> and </information>. You can search as many times as you want. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer> without detailed illustrations. For example, <answer> xxx </answer>. Question: question.

Search-R1's algorithm

Algorithm 1 LLM Response Rollout with Multi-Turn Search Engine Calls

Require: Input query x, policy model π_θ, search engine $\mathcal{R}$, maximum action budget B.
Ensure: Final response y.

Initialize rollout sequence $y \leftarrow \emptyset$
Initialize action count $b \leftarrow 0$
while $b < B$ **do**
    Initialize current action LLM rollout sequence $y_b \leftarrow \emptyset$
    while True **do**
        Generate response token $y_t \sim \pi_\theta(\cdot \mid x, y + y_b)$
        Append y_t to rollout sequence $y_b \leftarrow y_b + y_t$
        if y_t in [</search>, </answer>, <eos>] **then** break
        end if
    end while
    $y \leftarrow y + y_b$
    if <search> </search> detected in y_b **then**
        Extract search query $q \leftarrow \text{Parse}(y_b, \texttt{<search>}, \texttt{</search>})$
        Retrieve search results $d = \mathcal{R}(q)$
        Insert d into rollout $y \leftarrow y +$ <information>d</information>
    else if <answer> </answer> detected in y_b **then**
        return final generated response y
    else
        Ask for rethink $y \leftarrow y +$ "My action is not correct. Let me rethink."
    end if
    Increment action count $b \leftarrow b + 1$
end while
return final generated response y

Search-R1's performance

Methods	General QA			Multi-Hop QA				
	NQ†	TriviaQA⋆	PopQA⋆	HotpotQA†	2wiki⋆	Musique⋆	Bamboogle⋆	Avg.
Qwen2.5-7b-Base/Instruct								
Direct Inference	0.134	0.408	0.140	0.183	0.250	0.031	0.120	0.181
CoT	0.048	0.185	0.054	0.092	0.111	0.022	0.232	0.106
IRCoT	0.224	0.478	0.301	0.133	0.149	0.072	0.224	0.239
Search-o1	0.151	0.443	0.131	0.187	0.176	0.058	0.296	0.206
RAG	0.349	0.585	0.392	0.299	0.235	0.058	0.208	0.304
SFT	0.318	0.354	0.121	0.217	0.259	0.066	0.112	0.207
R1-base	0.297	0.539	0.202	0.242	0.273	0.083	0.296	0.276
R1-instruct	0.270	0.537	0.199	0.237	0.292	0.072	0.293	0.271
Search-R1-base	**0.480**	**0.638**	**0.457**	**0.433**	0.382	**0.196**	**0.432**	**0.431**
Search-R1-instruct	0.393	0.610	0.397	0.370	**0.414**	0.146	0.368	0.385
Qwen2.5-3b-Base/Instruct								
Direct Inference	0.106	0.288	0.108	0.149	0.244	0.020	0.024	0.134
CoT	0.023	0.032	0.005	0.021	0.021	0.002	0.000	0.015
IRCoT	0.111	0.312	0.200	0.164	0.171	0.067	0.240	0.181
Search-o1	0.238	0.472	0.262	0.221	0.218	0.054	**0.320**	0.255
RAG	0.348	0.544	0.387	0.255	0.226	0.047	0.080	0.270
SFT	0.249	0.292	0.104	0.186	0.248	0.044	0.112	0.176
R1-base	0.226	0.455	0.173	0.201	0.268	0.055	0.224	0.229
R1-instruct	0.210	0.449	0.171	0.208	0.275	0.060	0.192	0.224
Search-R1-base	**0.406**	**0.587**	**0.435**	0.284	0.273	0.049	0.088	0.303
Search-R1-instruct	0.341	0.545	0.378	**0.324**	**0.319**	**0.103**	0.264	**0.325**

RAG vs. long-context LLMs

RETRIEVAL MEETS LONG CONTEXT LARGE LANGUAGE MODELS

Peng Xu[†], Wei Ping[†], Xianchao Wu, Lawrence McAfee
Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina
Mohammad Shoeybi, Bryan Catanzaro
NVIDIA
[†]{pengx, wping}@nvidia.com

ABSTRACT

Extending the context window of large language models (LLMs) is getting popular recently, while the solution of augmenting LLMs with retrieval has existed for years. The natural questions are: *i) Retrieval-augmentation versus long context window, which one is better for downstream tasks? ii) Can both methods be combined to get the best of both worlds?* In this work, we answer these questions by studying both solutions using two state-of-the-art pretrained LLMs, i.e., a proprietary 43B GPT and Llama2-70B. Perhaps surprisingly, we find that LLM with 4K context window using simple retrieval-augmentation at generation can achieve comparable performance to finetuned LLM with 16K context window via *positional interpolation* on long context tasks, while taking much less computation. More importantly, we demonstrate that retrieval can significantly improve the performance of LLMs regardless of their extended context window sizes. Our best model, retrieval-augmented Llama2-70B with 32K context window, outperforms GPT-3.5-turbo-16k and Davinci003 in terms of average score on nine long context tasks including question answering, query-based summarization, and in-context few-shot learning tasks. It also outperforms its non-retrieval Llama2-70B-32k baseline by a margin, while being much faster at generation. Our study provides general insights on the choice of retrieval-augmentation versus long context extension of LLM for practitioners.

Can Long-Context Language Models Subsume Retrieval, RAG, SQL, and More?

Jinhyuk Lee* **Anthony Chen*** **Zhuyun Dai***
Dheeru Dua **Devendra Singh Sachan** **Michael Boratko** **Yi Luan**
Sébastien M. R. Arnold **Vincent Perot** **Siddharth Dalmia** **Hexiang Hu**
Xudong Lin **Panupong Pasupat** **Aida Amini** **Jeremy R. Cole**
Sebastian Riedel **Iftekhar Naim** **Ming-Wei Chang** **Kelvin Guu**

Google DeepMind

Abstract

Long-context language models (LCLMs) have the potential to revolutionize our approach to tasks traditionally reliant on external tools like retrieval systems or databases. Leveraging LCLMs' ability to natively ingest and process entire corpora of information offers numerous advantages. It enhances user-friendliness by eliminating the need for specialized knowledge of tools, provides robust end-to-end modeling that minimizes cascading errors in complex pipelines, and allows for the application of sophisticated prompting techniques across the entire system. To assess this paradigm shift, we introduce LOFT, a benchmark of real-world tasks requiring context up to millions of tokens designed to evaluate LCLMs' performance on in-context retrieval and reasoning. Our findings reveal LCLMs' surprising ability to rival state-of-the-art retrieval and RAG systems, despite never having been explicitly trained for these tasks. However, LCLMs still face challenges in areas like compositional reasoning that are required in SQL-like tasks. Notably, prompting strategies significantly influence performance, emphasizing the need for continued research as context lengths grow. Overall, LOFT provides a rigorous testing ground for LCLMs, showcasing their potential to supplant existing paradigms and tackle novel tasks as model capabilities scale.[1]

Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach

Zhuowan Li[1] **Cheng Li**[1] **Mingyang Zhang**[1]
Qiaozhu Mei[2*] **Michael Bendersky**[1]
[1] Google DeepMind [2] University of Michigan
[1] {zhuowan,chgli,mingyang,bemike}@google.com [2] qmei@umich.edu

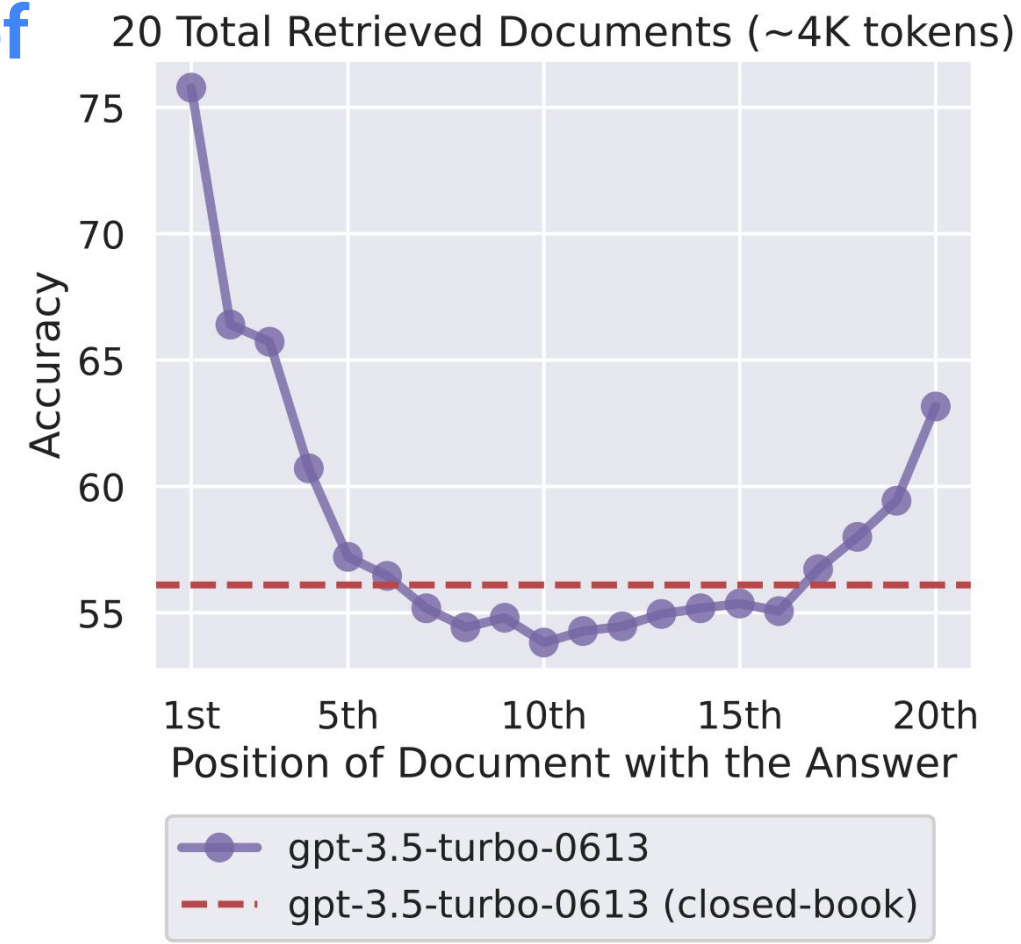
Abstract

Retrieval Augmented Generation (RAG) has been a powerful tool for *Large Language Models (LLMs)* to efficiently process overly lengthy contexts. However, recent LLMs like Gemini-1.5 and GPT-4 show exceptional capabilities to understand long contexts directly. We conduct a comprehensive comparison between RAG and long-context *(LC)* LLMs, aiming to leverage the strengths of both. We benchmark RAG and LC across various public datasets using three latest LLMs. Results reveal that when resourced sufficiently, LC consistently outperforms RAG in terms of average performance. However, RAG's significantly lower cost remains a distinct advantage. Based on this observation, we propose SELF-ROUTE, a simple yet effective method that routes queries to RAG or LC based on model self-reflection. SELF-ROUTE significantly reduces the computation cost while maintaining a comparable performance to LC. Our findings provide a guideline for long-context applications of LLMs using RAG and LC.

Lost in the Middle: How Language Models Use Long Contexts

Nelson F. Liu[1*] **Kevin Lin**[2] **John Hewitt**[1] **Ashwin Paranjape**[3]
Michele Bevilacqua[3] **Fabio Petroni**[3] **Percy Liang**[1]
[1]Stanford University [2]University of California, Berkeley [3]Samaya AI
nfliu@cs.stanford.edu

Changing the location of relevant information results in a U-shaped performance curve



Context rot: How increasing input tokens impacts model performance

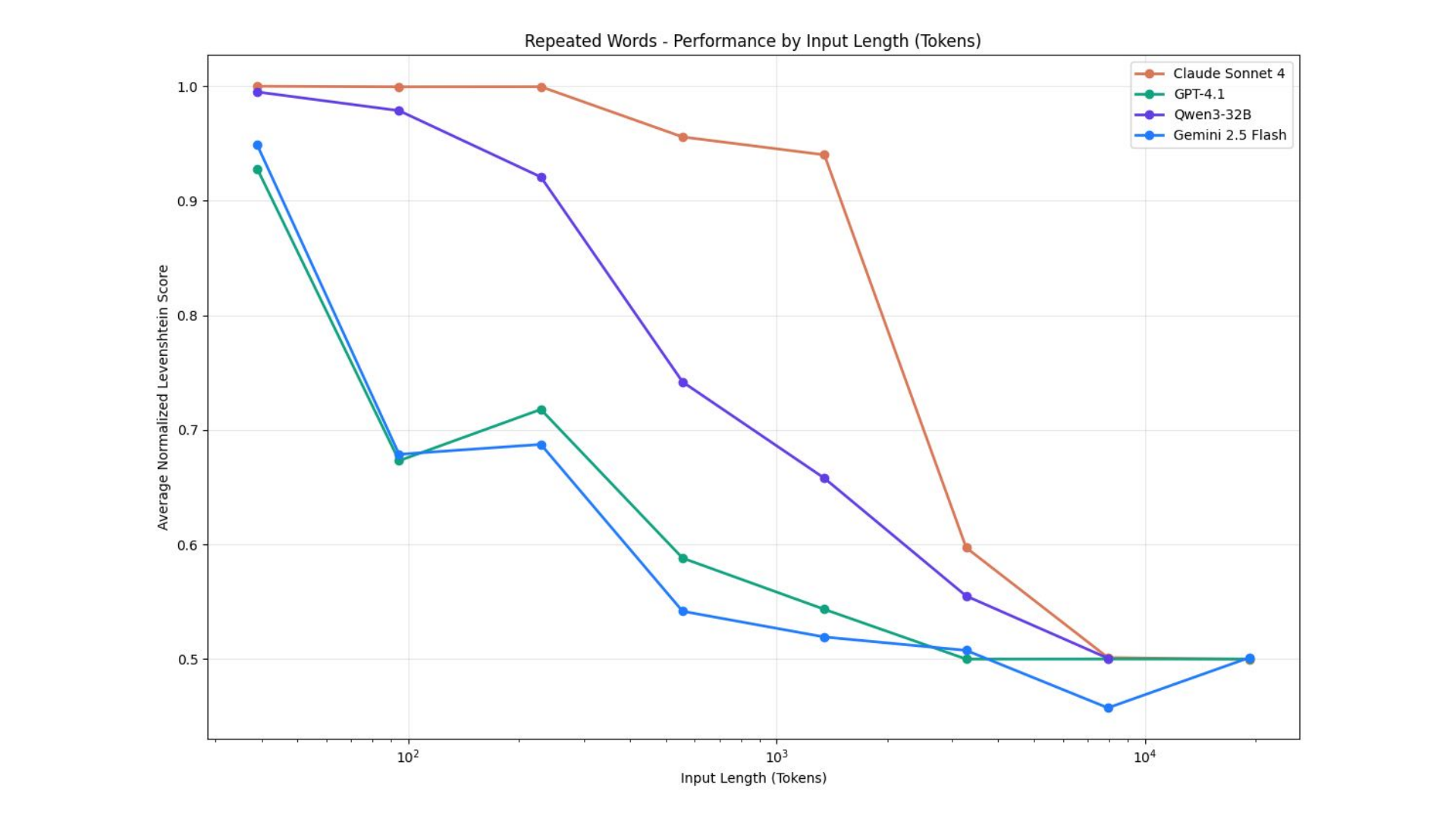
- https://research.trychroma.com/context-rot

We design a controlled task in which the model must replicate a sequence of repeated words, with a single unique word inserted at a specific position. The prompt explicitly instructs the model to reproduce the input text exactly.

One example prompt is:

```
Simply replicate the following text, output the exact same text: apple apple
apple apple apples apple apple apple apple apple apple apple apple apple
apple apple apple apple apple apple apple apple apple apple
```

Repeated Words - Sample Prompt Containing 'apple' as the repeated word, and 'apples' as the unique word


Repeated Words - Performance by Input Length (Tokens)
Claude Sonnet 4
GPT-4.1
Qwen3-32B
Gemini 2.5 Flash
Average Normalized Levenshtein Score
1.0
0.9
0.8
0.7
0.6
0.5
10^2
10^3
10^4
Input Length (Tokens)

Model context protocol (MCP)

- https://www.deeplearning.ai/short-courses/mcp-build-rich-context-ai-apps-with-anthropic/

Demo

- https://learn.deeplearning.ai/courses/mcp-build-rich-context-ai-apps-with-anthropic/lesson/ccsd0/why-mcp

What is the Model Context Protocol (MCP)

MCP is an open protocol that standardizes how your **LLM applications** connect to and work with your **tools & data sources**.

REST APIs

Standardize how **web applications** interact with the **backend**

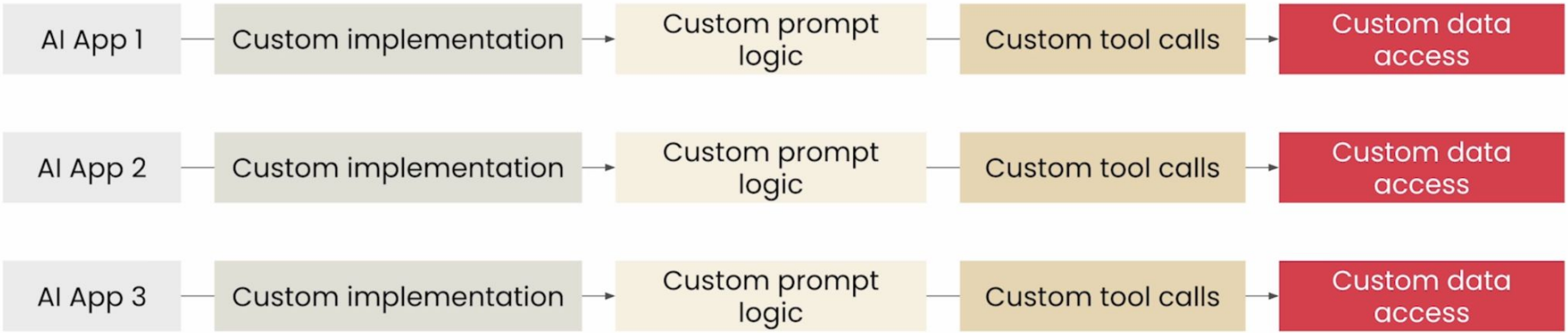
LSP

Standardizes how **IDEs** interact with **language-specific tools**

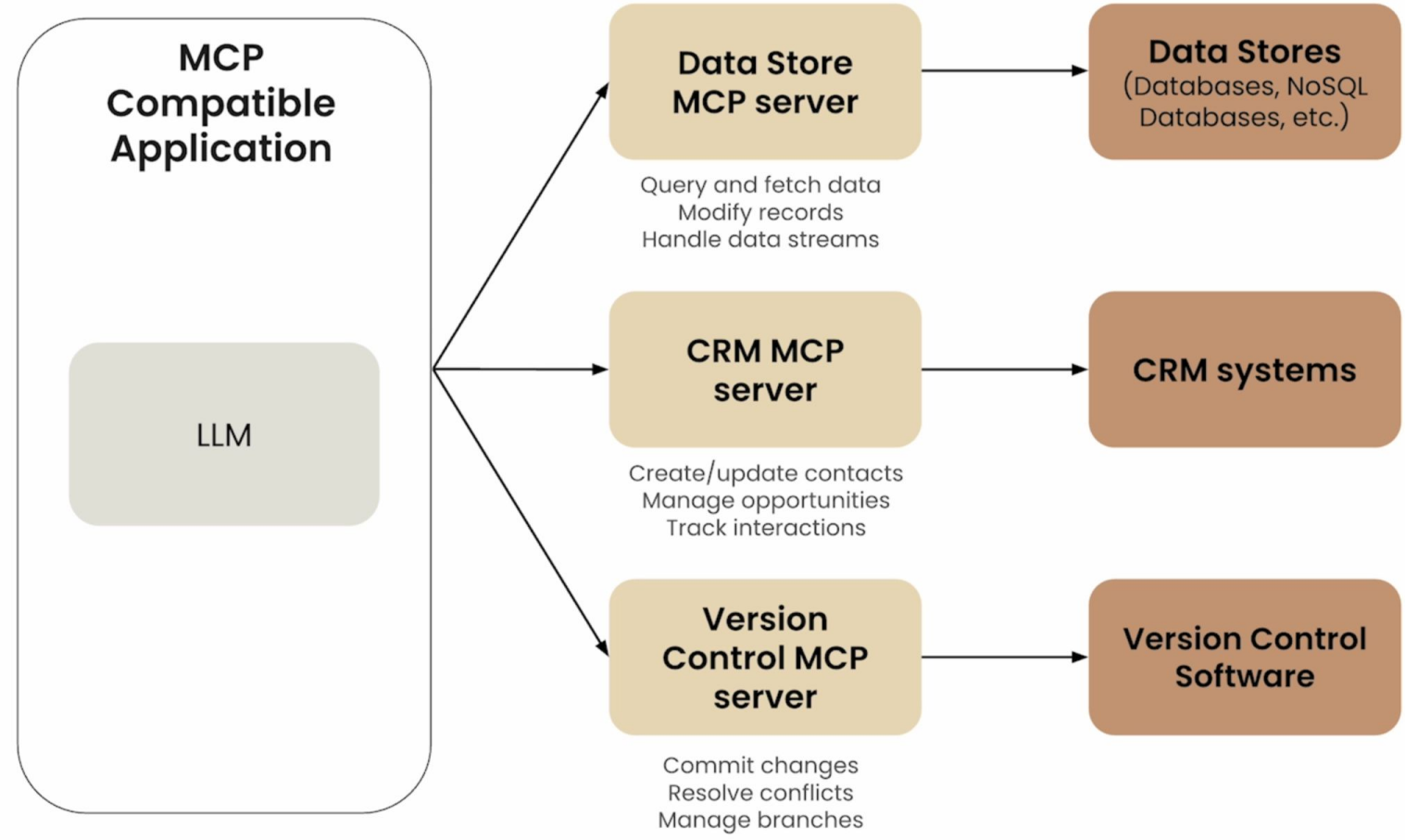
MCP

Standardizes how **AI applications** interact with **external systems**

Without MCP: Fragmented AI Development



With MCP: Standardized AI Development

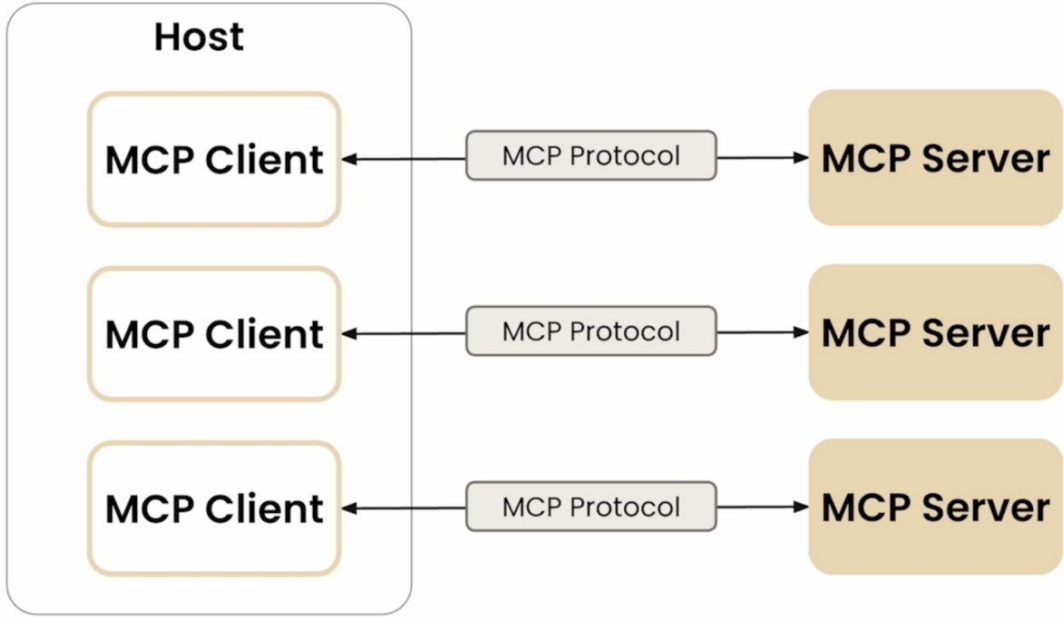


Client-Server Architecture

Host are LLM applications that want to access data through MCP (ex: Claude Desktop, IDEs, AI agents).

MCP Servers are lightweight programs that each expose specific capabilities through MCP.

MCP Clients maintain 1:1 connections with servers, inside the host application.



README | Code of conduct | MIT license | Security

Model Context Protocol servers

This repository is a collection of *reference implementations* for the Model Context Protocol (MCP), as well as references to community built servers and additional resources.

The servers in this repository showcase the versatility and extensibility of MCP, demonstrating how it can be used to give Large Language Models (LLMs) secure, controlled access to tools and data sources. Each MCP server is implemented with either the Typescript MCP SDK or Python MCP SDK.

> Note: Lists in this README are maintained in alphabetical order to minimize merge conflicts when adding new items.

🌟 Reference Servers

These servers aim to demonstrate MCP features and the TypeScript and Python SDKs.

- **AWS KB Retrieval** - Retrieval from AWS Knowledge Base using Bedrock Agent Runtime
- **Brave Search** - Web and local search using Brave's Search API
- **EverArt** - AI image generation using various models
- **Everything** - Reference / test server with prompts, resources, and tools
- **Fetch** - Web content fetching and conversion for efficient LLM usage
- **Filesystem** - Secure file operations with configurable access controls
- **Git** - Tools to read, search, and manipulate Git repositories
- **GitHub** - Repository management, file operations, and GitHub API integration
- **GitLab** - GitLab API, enabling project management
- **Google Drive** - File access and search capabilities for Google Drive
- **Google Maps** - Location services, directions, and place details
- **Memory** - Knowledge graph-based persistent memory system
- **PostgreSQL** - Read-only database access with schema inspection
- **Puppeteer** - Browser automation and web scraping
- **Redis** - Interact with Redis key-value stores
- **Sentry** - Retrieving and analyzing issues from Sentry.io
- **Sequential Thinking** - Dynamic and reflective problem-solving through thought sequences
- **Slack** - Channel management and messaging capabilities
- **Sqlite** - Database interaction and business intelligence capabilities
- **Time** - Time and timezone conversion capabilities

🤝 Third-Party Servers

🎖️ Official Integrations

Thank you!