Part 4: Challenges, Opportunities and Resources

LLM-Rankers and LLM-Judge (autoraters): two sides of the same coin

Rankers, Judges, and Assistants: Towards Understanding the Interplay of LLMs in Information Retrieval Evaluation

Krisztian Balog
Google DeepMind
Stavanger, Norway
krisztianb@google.com

Donald Metzler
Google DeepMind
Mountain View, USA
metzler@google.com

Zhen Qin
Google DeepMind
Mountain View, USA
zhenqin@google.com

Abstract

Large language models (LLMs) are increasingly integral to information retrieval (IR), powering ranking, evaluation, and AI-assisted content creation. This widespread adoption necessitates a critical examination of potential biases arising from the interplay between these LLM-based components. This paper synthesizes existing research and presents novel experiment designs that explore how LLM-based rankers and assistants influence LLM-based judges. We provide the first empirical evidence of LLM judges exhibiting significant bias towards LLM-based rankers. Furthermore, we observe limitations in LLM judges' ability to discern subtle system performance differences. Contrary to some previous findings, our preliminary
content
of the L
initial g
of LLMs

the potential risks alongside the undeniable benefits. Could this heavy reliance on LLMs across content creation, retrieval, ranking, evaluation, etc., inadvertently introduce or amplify biases within these systems?

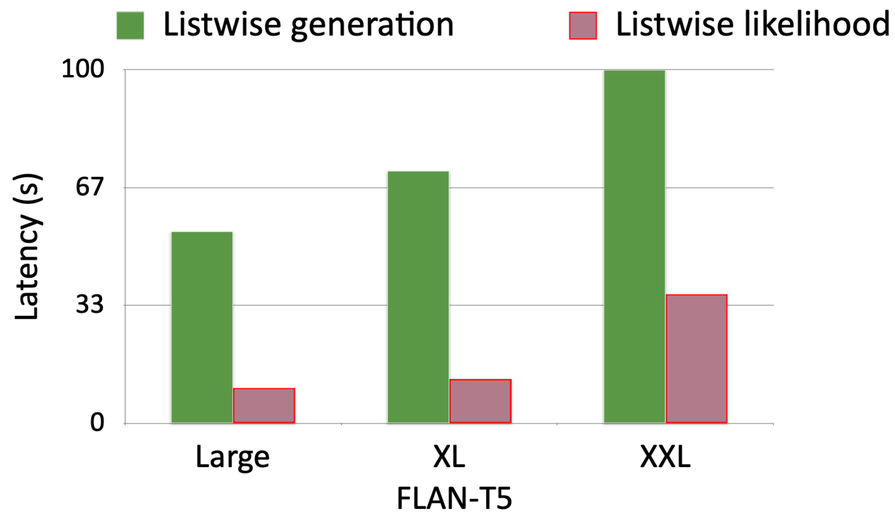
Recent research has begun to explore some of these emerging issues. For example, studies have shown that LLMs can exhibit biases in their output, favoring LLM-generated content over human-generated ones [10], and perpetuating biases present in their training data [15, 21, 33]. Furthermore, LLM-based rating systems have been found to be susceptible to manipulation [2], may not accurately reflect human preferences [28], and demonstrate self-inconsistency [50]. Additionally, the phenomenon of "model
on synthetic
n of quality
s for assess-
ons ranging
from complete rejection of LLMs for relevance assessment [48] to

LLM judges exhibiting significant bias towards LLM rankers

- LLM Rankers: assess the relevance of a doc to a q to score doc
- LLM Judge: assess the relevance of a doc to a q to label the doc for search evaluation
- Also in LLM Judge labelling can be pointwise, pairwise, listwise

Challenges & Opportunities: Latency, Costs, Scalability & Deployability

From Part 3: Rankers Efficiency at Inference



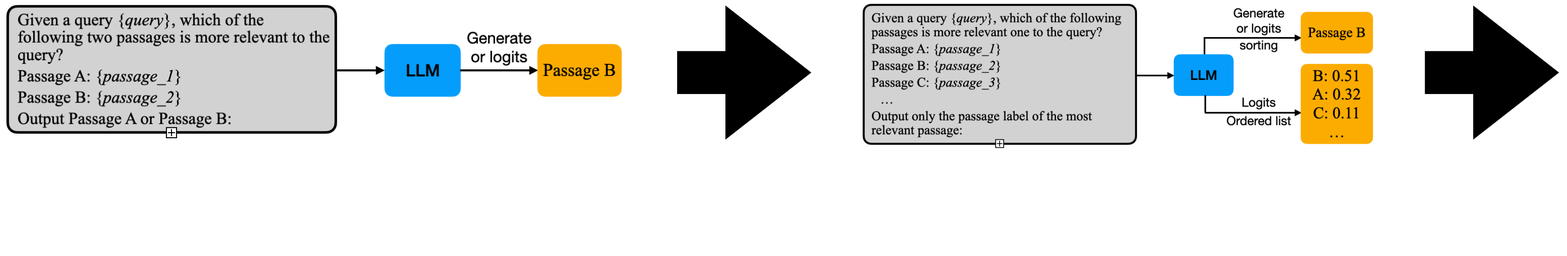
- Impractical latency
- High infrastructure and inference costs

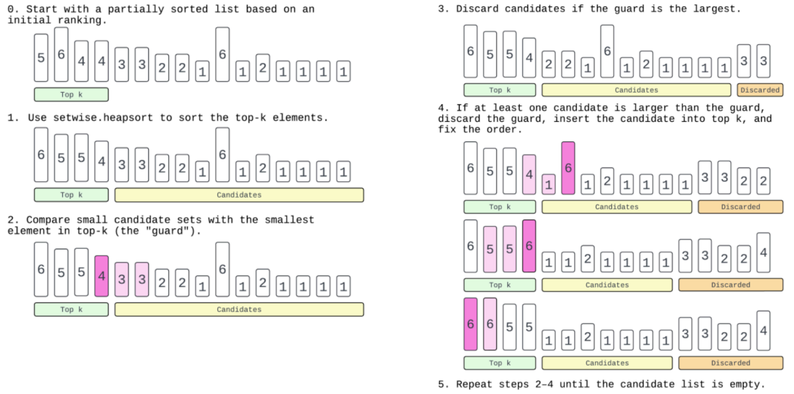
Deepseek R1 Gemini 2.5 Pro Claude 3.7 Sonnet OpenAI o1 Qwen3

Reasoning further exacerbates latency, costs

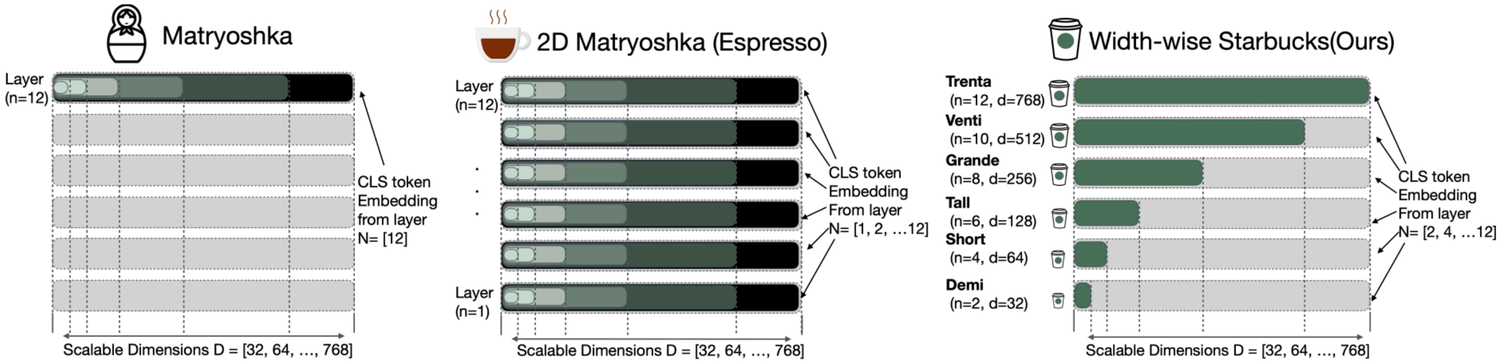
Challenges & Opportunities: Latency, Costs, Scalability & Deployability

Methods Efficiencies

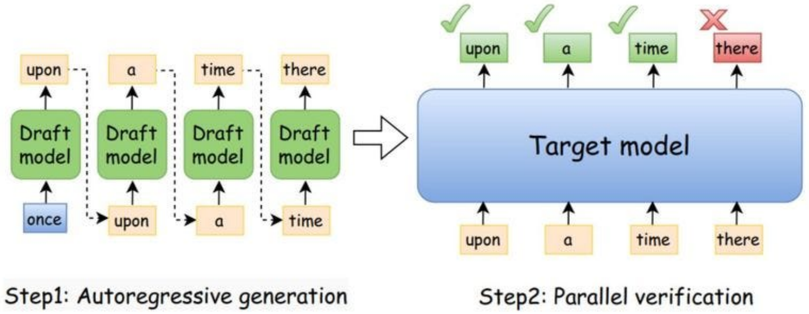




Backbone Efficiencies



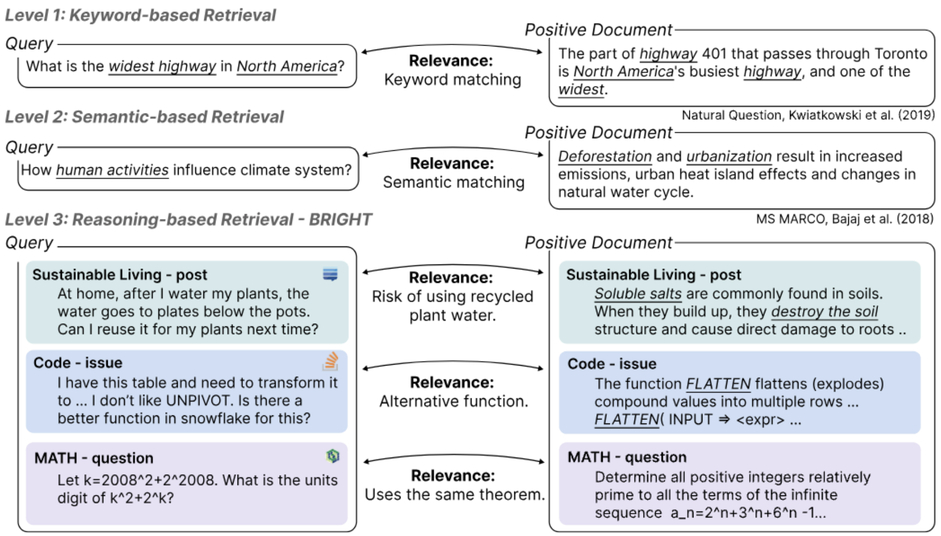
Matryoshka Representation Learning



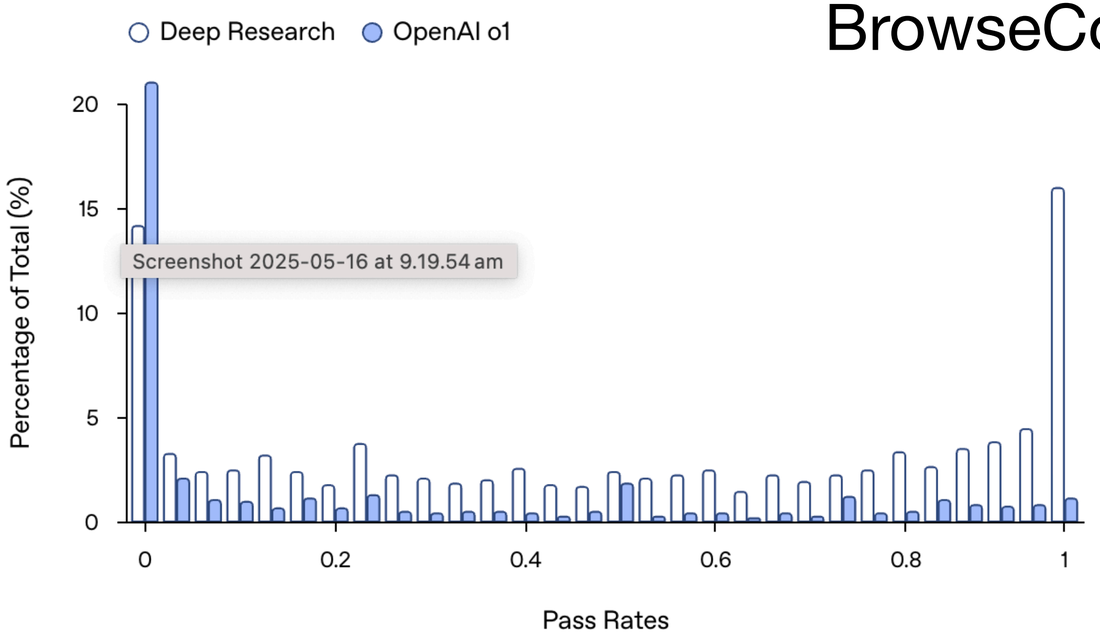
Speculative Decoding

Challenges & Opportunities: Evaluation of more complex retrieval & ranking tasks

BRIGHT



BrowseComp



- Complex, reasoning intensive tasks; but current evaluation resources have limits:
  - Very artificial user requests
  - Noisy labels
  - Lack of reference corpus for retrieval and ranking (BrowseComp)

Challenges & Opportunities: Personalise retrieval and ranking

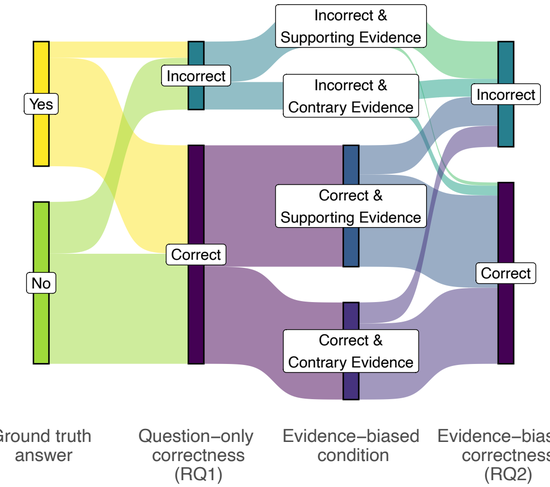
- LLMs prompt instructions offer the possibility to easily and explicitly integrate user preferences on how search works
  - Limited datasets, no retrieval/ranking focus

    Salemi & Zamani, LaMP-QA: A Benchmark for Personalized Long-form Question Answering. arXiv preprint arXiv:2506.00137. 2025

Challenges & Opportunities: robustness to attacks and biases

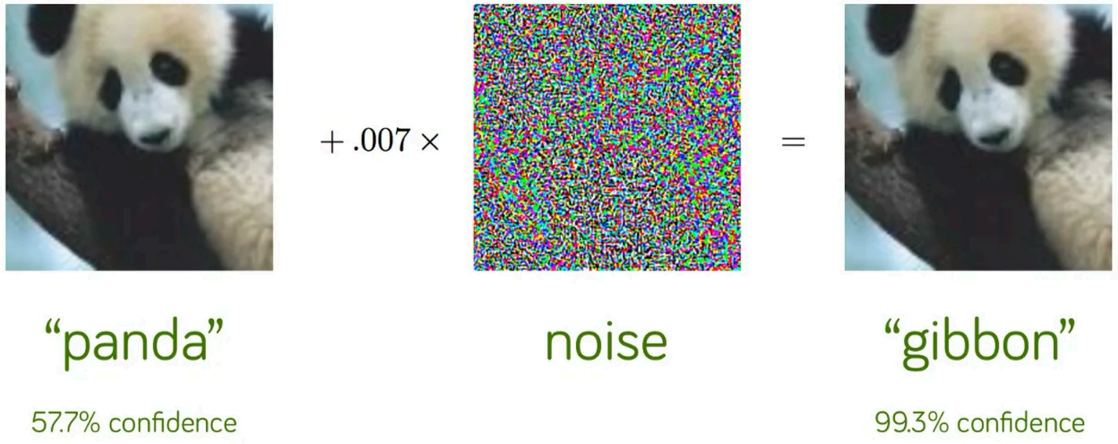
LLMs encode many types of biases, which are hardly identified, controlled

Koopman&Zuccon "Humans are more gullible than LLMs in believing common psychological myths." arXiv 2025



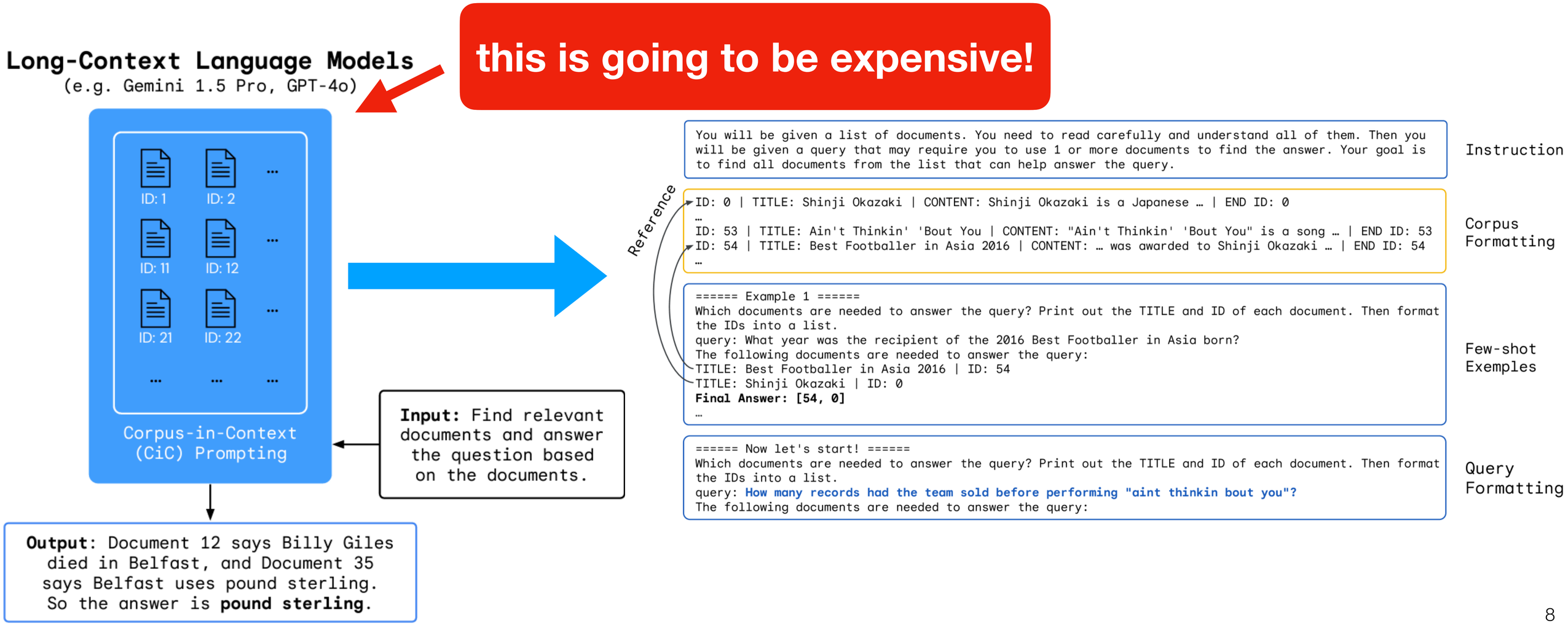
Koopman&Zuccon "Dr ChatGPT tell me what I want to hear: How different prompts impact health answer correctness." EMNLP 2023

Attacks are possible; effects not well understood yet, unclear how to identify

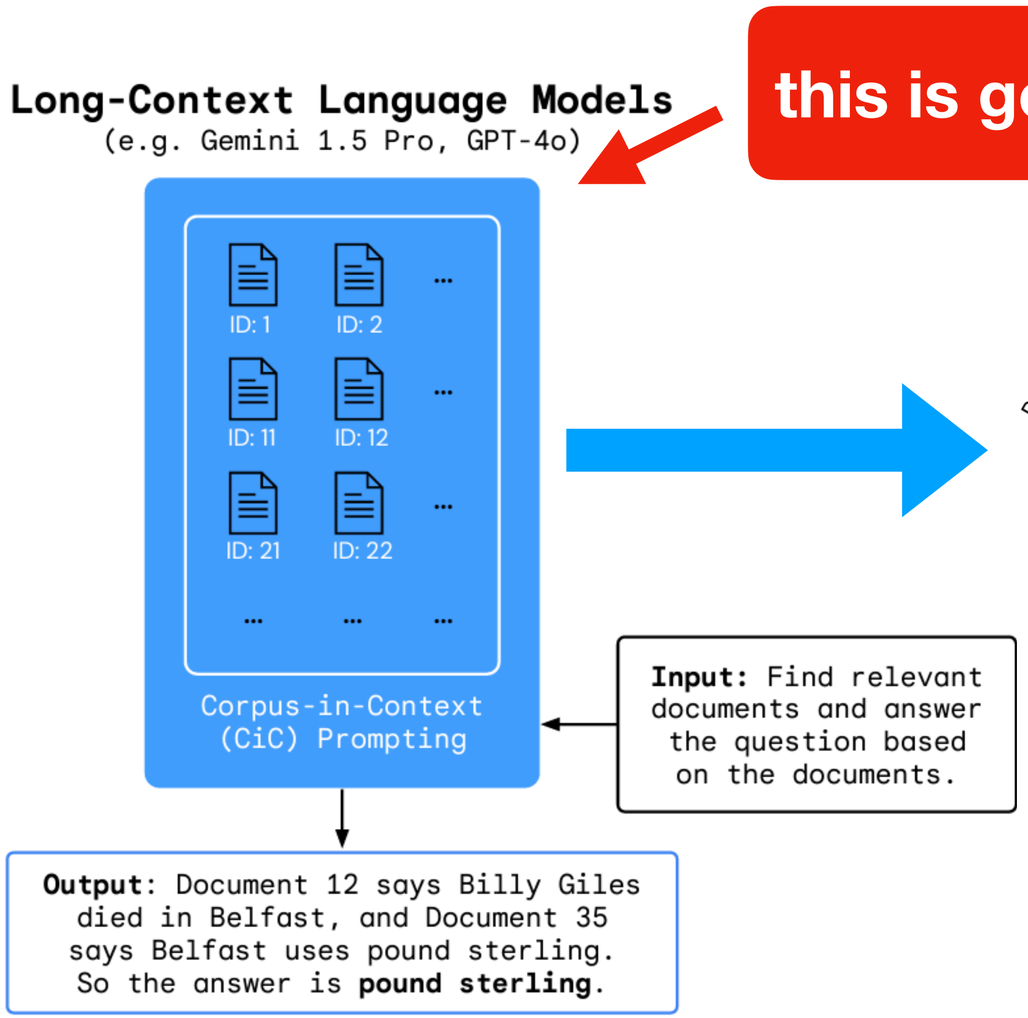


Zhuang,, et al. "Document screenshot retrievers are vulnerable to pixel poisoning attacks." SIGIR 2025

Challenges & Opportunities: LC-LLMs to rule it all



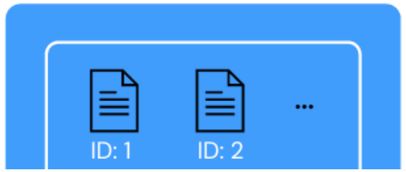
Challenges & Opportunities: LC-LLMs to rule it all



Limitations Our experiments were constrained by the computational resources and financial costs associated with utilizing LCLMs. The entire LOFT 128k test sets contain around 35 datasets $\times$ 100 prompts $\times$ 128k tokens $=$ 448M input tokens, which cost \$1,568 for Gemini 1.5 Pro, \$2,240 for GPT-4o, and \$6,720 for Claude 3 Opus at the time of writing. To reduce costs, we also release dev sets, which are 10x smaller and can be evaluated with around \$200 using Gemini 1.5 Pro or GPT-4o. We also expect LLM API prices to decrease over time. Another limitation of this work is that we focused on evaluating the quality of LCLMs, and leave efficiency considerations for future work. We could not measure the efficiency improvements from prefix caching [20] due to API constraints at the time of writing. Without caching, the Gemini 1.5 Pro API has a median latency of roughly four seconds for 32k input tokens, twelve seconds for 128k input tokens, and 100 seconds for 1 million input tokens. This speed is likely slower than specialized retrievers or SQL databases; the promising quality results on LOFT encourage further investigation into optimizing LCLMs efficiency. Additionally, our retrieval and RAG tasks was limited to 1 million tokens, which still leaves a large gap from real-world applications that may involve several million or even billions of documents.

Challenges & Opportunities: LC-LLMs to rule it all

Long-Context Language Models
(e.g. Gemini 1.5 Pro, GPT-4o)



this is going to be expensive!

But…

You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query that may require you to use 1 or more documents to find the answer. Your goal is to find all documents from the list that can help answer the query.

Instruction

Reference

ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 54
…

Corpus Formatting

====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 54
TITLE: Shinji Okazaki | ID: 0
Final Answer: [54, 0]
…

Few-shot Exemples

====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: **How many records had the team sold before performing "aint thinkin bout you"?**
The following documents are needed to answer the query:

Query Formatting

You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query that may require you to use 1 or more documents to find the answer. Your goal is

Instruction

Limitations Our experiments were constrained by the computational resources and financial costs associated with utilizing LCLMs. The entire LOFT 128k test sets contain around 35 datasets × ns = 448M input tokens, which cost $1,568 for Gemini 1.5 Pro, $2,240 for Claude 3 Opus at the time of writing. To reduce costs, we also release dev er and can be evaluated with around $200 using Gemini 1.5 Pro or GPT-4o. PI prices to decrease over time. Another limitation of this work is that we e quality of LCLMs, and leave efficiency considerations for future work. e efficiency improvements from prefix caching [20] due to API constraints Without caching, the Gemini 1.5 Pro API has a median latency of roughly ut tokens, twelve seconds for 128k input tokens, and 100 seconds for 1 is speed is likely slower than specialized retrievers or SQL databases; the on LOFT encourage further investigation into optimizing LCLMs efficiency. l and RAG tasks was limited to 1 million tokens, which still leaves a large ications that may involve several million or even billions of documents.

Likely to improve further in future

Challenges & Opportunities: The Role of Reasoning

- Initial steps to do reasoning for ranking — but the contribution to effectiveness is still unclear
  - Likely due to lack of adequate datasets to pick up these signals
- How to do reasoning for retrieval?
  - ReasonIR does not do architectural changes to integrate reasoning in DRs

Where from here? Pointers to resources

Large Language Models for Information Retrieval: A Survey

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng
Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen

Abstract—As a primary means of information acquisition, information retrieval (IR) systems, such as search engines, have integrated themselves into our daily lives. These systems also serve as components of dialogue, question-answering, and recommender systems. The trajectory of IR has evolved dynamically from its origins in term-based methods to its integration with advanced neural models. While the neural models excel at capturing complex contextual signals and semantic nuances, thereby reshaping the IR landscape, they still face challenges such as data scarcity, interpretability, and the generation of contextually plausible yet potentially inaccurate responses. This evolution requires a combination of both traditional methods (such as term-based sparse retrieval methods with rapid response) and modern neural architectures (such as language models with powerful language understanding capacity). Meanwhile, the emergence of large language models (LLMs), typified by ChatGPT and GPT-4, has revolutionized natural language processing due to their remarkable language understanding, generation, generalization, and reasoning abilities. Consequently, recent research has sought to leverage LLMs to improve IR systems. Given the rapid evolution of this research trajectory, it is necessary to consolidate existing methodologies and provide nuanced insights through a comprehensive overview. In this survey, we delve into the confluence of LLMs and IR systems, including crucial aspects such as query rewriters, retrievers, rerankers, and readers. Additionally, we explore promising directions, such as search agents, within this expanding field.

4 Sep 2024

Where from here? Pointers to resources

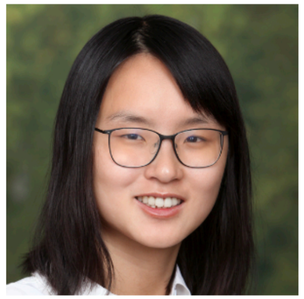
Large Language Models for Information

Yutao Zhu,

ACL 2023 Tutorial:

Retrieval-based Language Models and Applications



Akari Asai[1], Sewon Min[1], Zexuan Zhong[2], Danqi Chen[2]

[1]University of Washington, [2]Princeton University

Abstract—As a primar
themselves into our dai
The trajectory of IR ha
While the neural mode
they still face challenge
responses. This evoluti
response) and modern
the emergence of large
to their remarkable lan
sought to leverage LLM
existing methodologies
of LLMs and IR system
promising directions, su

4 Sep 2024

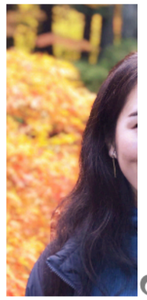
Where from here? Pointers to resources

Large Language Models for Information

ACL 2023 Tutorial:

Yutao Zhu, **Retrieval-based Language Models and Applications**

Abstract—As a primar
themselves into our dai
The trajectory of IR ha
While the neural mode
they still face challenge
responses. This evoluti
response) and modern
the emergence of large
to their remarkable lan
sought to leverage LLM
existing methodologies
of LLMs and IR system
promising directions, su



Akar

Information Retrieval Meets Large Language Models: A Strategic Report from Chinese IR Community

Qingyao AI[a], Ting BAI[b], Zhao CAO[c], Yi CHANG[d], Jiawei CHEN(✉)[e], Zhumin CHEN[f], Zhiyong CHENG[g], Shoubin DONG[h], Zhicheng DOU[i], Fuli FENG[j], Shen GAO[f], Jiafeng GUO[k], Xiangnan HE(✉)[j], Yanyan LAN[a], Chenliang LI[l], Yiqun LIU[a], Ziyu LYU[m], Weizhi MA[a], Jun MA[f], Zhaochun REN[f], Pengjie REN[f], Zhiqiang WANG[n], Mingwen WANG[o], Ji-Rong WEN[i], Le WU[p], Xin XIN[f], Jun XU[i], Dawei YIN[q], Peng ZHANG(✉)[r], Fan ZHANG[j], Weinan ZHANG[s], Min ZHANG[a], Xiaofei ZHU[t]

[a]*Tsinghua University*, [b]*Beijing University of Posts and Telecommunications*, [c]*Huawei Technologies Ltd. Co*, [d]*Jilin University*, [e]*Zhejiang University*, [f]*Shandong University*, [g]*Shandong Artificial Intelligence Institute*, [h]*South China University of Technology*, [i]*Renmin University of China*, [j]*University of Science and Technology of China*, [k]*Institute of Computing Technology, Chinese Academy of Sciences*, [l]*Wuhan University*, [m]*Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences*, [n]*Shanxi University*, [o]*Jiangxi Normal University*, [p]*Hefei University of Technology*, [q]*Baidu Inc.*, [r]*Tianjin University*, [s]*Shanghai Jiao Tong University*, [t]*Chongqing University of Technology*

Tools for research on LLM-retrievers

- **FlagEmbedding**: The BGE series, RAG-Retrieval: The Stella-embedding series
  https://github.com/FlagOpen/FlagEmbedding

- **Tevatron**: LLM/VLLM retriever and reranker finetuning
  https://github.com/texttron/tevatron

- **PyLate**: ColBERT style mutli-vector retriever training and inference
  https://github.com/lightonai/pylate

- **SentenceTransformer**: Support Most open source embedding and rerank model for inference and training
  https://github.com/UKPLab/sentence-transformers

- **Search-R1, Verl-Tool**: RL with Search
  https://github.com/PeterGriffinJin/Search-R1

Tools for research on LLM-rankers

- LLM-rankers: https://github.com/ielab/llm-rankers
  - Reference implementations of Zero-shot Pointwise, Listwise, Pairwise, Setwise, Rank-R1
- RankLLM: https://github.com/castorini/rank_llm
  - Python toolkit for rerankers, with a focus on listwise reranking; includes reference implementations of BERT backbones methods
- LLM4Ranking: https://github.com/liuqi6777/llm4ranking
  - Similar to LLM-rankers: Reference implementations of LLM-rankers, supports open-source or closed-source API-based LLMs, includes fine-tuning scripts

It’s time for Questions / Comments?