Retrieval-Augmented Generation (RAG)

Yu Meng
University of Virginia
yumeng5@virginia.edu

Oct 30, 2024

Join at
slido.com
#3719 083

Reminder

Assignment 4 is due next Monday **(11/04) 11:59pm**!

Join at
slido.com
#3719 083

Overview of Course Contents

- Week 1: Logistics & Overview
- Week 2: N-gram Language Models
- Week 3: Word Senses, Semantics & Classic Word Representations
- Week 4: Word Embeddings
- Week 5: Sequence Modeling and Neural Language Models
- Week 6-7: Language Modeling with Transformers (Pretraining + Fine-tuning)
- Week 8: Large Language Models (LLMs) & In-context Learning
- Week 9-10: Reasoning, Knowledge, and Retrieval-Augmented Generation (RAG)
- Week 11: LLM Alignment
- Week 12: Language Agents
- Week 13: Recap + Future of NLP
- Week 15 (after Thanksgiving): Project Presentations

Join at
slido.com
#3719 083

(Recap) Hallucination

- **Hallucination**: LM generates information that is factually incorrect, misleading, or fabricated, even though it may sound plausible or convincing
- Why does hallucination happen?
  - Limited knowledge: LLMs are trained on finite datasets, which don't have access to all possible information; when asked about topics outside their training data, they may generate plausible-sounding but incorrect responses
  - Overgeneralization: LLMs may apply patterns they've learned from one context to another where they don't apply, leading to incorrect conclusions
  - Lack of common sense: While LLMs can process and generate human-like text, they often lack the ability to apply commonsense reasoning to their outputs
  - …

Join at
slido.com
#3719 083

(Recap) Non-parametric Knowledge

- **Non-parametric knowledge**: (external) information not stored in the model's parameters but can be accessed or retrieved when needed
- Examples:
  - External knowledge bases/graphs
  - Pretraining corpora
  - User-provided documents/passages
- Non-parametric knowledge is typically used to **augment** parametric knowledge (typically via **retrieval**) for more accurate factoid question answering
- Benefits of **non-parametric knowledge**
  - Incorporate more information without increasing model size
  - Easier updates and modifications to the knowledge base
  - Improve model interpretability

Join at
slido.com
#3719 083

(Recap) Retrieval-Augmented Generation

- Use a **retriever** to obtain relevant documents to the query from an external text collection
- Use LLMs to generate answers given the documents and a prompt

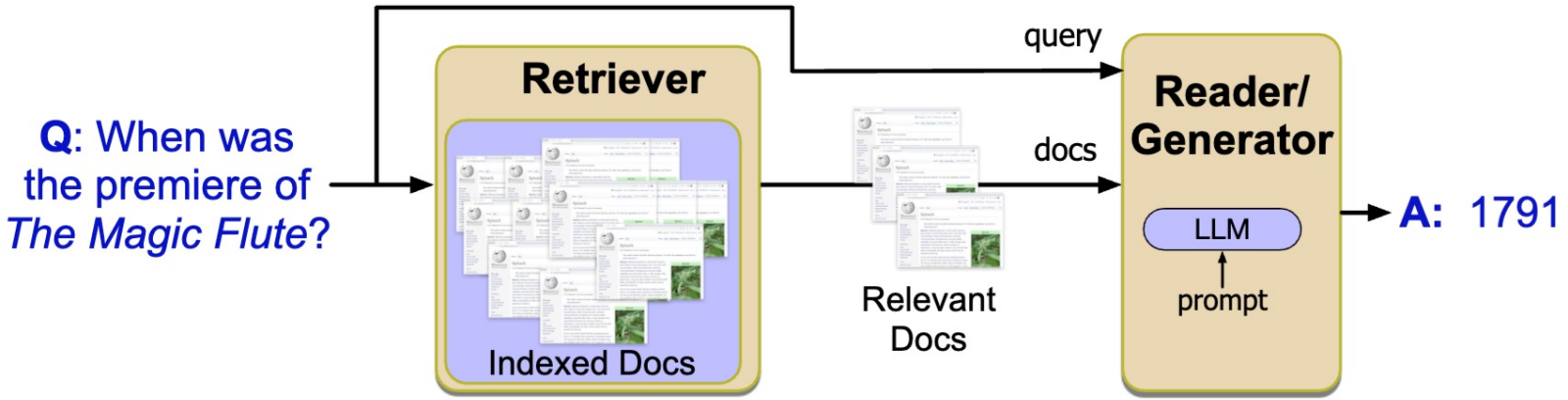


Figure source: https://web.stanford.edu/~jurafsky/slp3/14.pdf

Join at
slido.com
#3719 083

(Recap) Information Retrieval (IR)

- **Information retrieval** (**IR**): finding relevant information from a large collection of unstructured data (e.g., documents, web pages) in response to a user query
- **Query**: user-provided input (e.g., keywords or phrases), describing the information they are seeking
- **Documents/corpus**: the data collection that the system searches through
- **Ranking**: sort the search results by relevance based on specific metrics (e.g., keyword matching, semantic similarity)
- Web search engines (e.g., Google, Bing) are IR systems

Join at
slido.com
#3719 083

(Recap) Sparse vs. Dense Retrieval

- **Sparse** retrieval: based on traditional IR techniques where the representations of documents and queries are sparse (most vector values are zero)
  - Example: TF-IDF
  - Pros: simple and interpretable
  - Cons: lack semantic understanding
- **Dense** retrieval: encode documents and queries into dense vectors (embeddings) using deep neural networks
  - Example: BERT-based encoding methods
  - Pros: semantic & contextualized understanding
  - Cons: computationally more expensive and less interpretable

Join at
slido.com
#3719 083

(Recap) TF-IDF Weighting

- Introduced in week 3's lectures $\text{TF-IDF}(w, d) = \text{TF}(w, d) \times \text{IDF}(w)$
- Main idea: represent a document with frequent & distinctive words

TF-IDF weighted

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.246	0	0.454	0.520
good	0	0	0	0
fool	0.030	0.033	0.0012	0.0019
wit	0.085	0.081	0.048	0.054

$\cos(\boldsymbol{v}_{d_2}, \boldsymbol{v}_{d_3}) = 0.10 \quad \cos(\boldsymbol{v}_{d_3}, \boldsymbol{v}_{d_4}) = 0.99$

Raw counts

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

$\cos(\boldsymbol{v}_{d_2}, \boldsymbol{v}_{d_3}) = 0.81 \quad \cos(\boldsymbol{v}_{d_3}, \boldsymbol{v}_{d_4}) = 0.99$

Figure source: https://web.stanford.edu/~jurafsky/slp3/6.pdf

Join at
slido.com
#3719 083

(Recap) Example: TF-IDF for Sparse Retrieval

- Example query and mini-corpus:

Query:	sweet love
Doc 1:	Sweet sweet nurse! Love?
Doc 2:	Sweet sorrow
Doc 3:	How sweet is love?
Doc 4:	Nurse!

- Query & document vectors:

	Query					
word	**cnt**	**tf**	**df**	**idf**	**tf-idf**	**n'lized** = tf-idf/$\|q\|$
sweet	1	1	3	0.125	0.125	0.383
nurse	0	0	2	0.301	0	0
love	1	1	2	0.301	0.301	0.924
how	0	0	1	0.602	0	0
sorrow	0	0	1	0.602	0	0
is	0	0	1	0.602	0	0

	Document 1			
word	**cnt**	**tf**	**tf-idf**	**n'lized**
sweet	2	1.301	0.163	0.357
nurse	1	1.000	0.301	0.661
love	1	1.000	0.301	0.661
how	0	0	0	0
sorrow	0	0	0	0
is	0	0	0	0

$$\cos(\boldsymbol{q}, \boldsymbol{d}_1) = 0.747$$

	Document 2			
cnt	**tf**	**tf-idf**	**n'lized**
1	1.000	0.125	0.203
0	0	0	0
0	0	0	0
0	0	0	0
1	1.000	0.602	0.979
0	0	0	0

$$\cos(\boldsymbol{q}, \boldsymbol{d}_2) = 0.078$$

Figure source: https://web.stanford.edu/~jurafsky/slp3/14.pdf

Join at
slido.com
#3719 083

Agenda

- Dense Retrieval
- Evaluation of Retrieval
- Retrieval-Augmented Generation (RAG)
- Long-context Issues

Join at
slido.com
#3719 083

Dense Retrieval

- Motivation: sparse retrieval (e.g., TF-IDF) relies on the exact overlap of words between the query and document without considering semantic similarity
- Solution: use a language model to obtain (dense) distributed representations of query and document
- The retriever language model is typically a small text encoder model (e.g., BERT)
  - Retrieval is a natural language understanding task
  - Encoder-only models are more efficient than LLMs for this purpose
- Both query and document representations are computed by text encoders

Join at
slido.com
#3719 083

Dense Retrieval: Cross-encoder

- Process query-document pairs together
- Relevance score produced directly by the model output
- (+) Capture intricate interactions between the query and the document
- (-) Not scalable to large retrieval corpus
- Good for small document sets

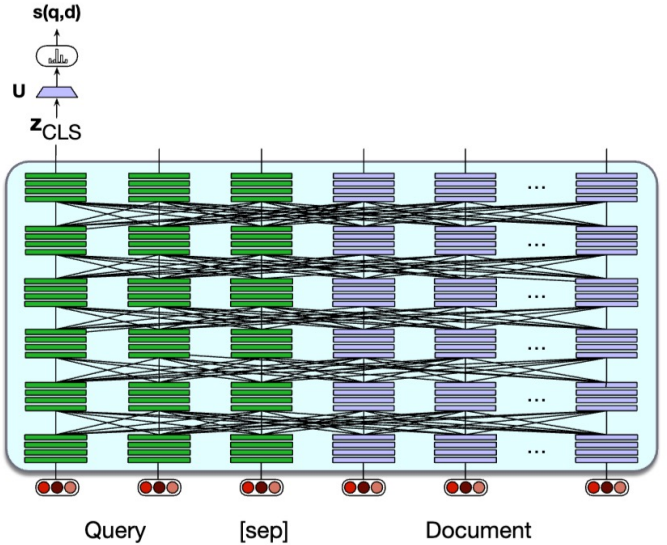


Figure source: https://web.stanford.edu/~jurafsky/slp3/14.pdf

Join at
slido.com
#3719 083

Dense Retrieval: Bi-encoder

- Independently encode the query and the document using two separate (but often identical) encoder models
- Use cosine similarity between the query and document vectors as relevance score
- (+) Document vectors can be precomputed
- (-) Cannot capture query-document interactions
- Common choice for large-scale retrieval

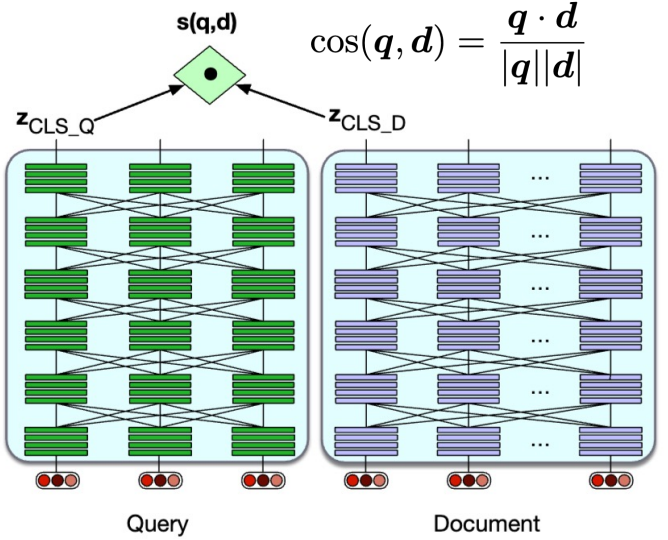


Figure source: https://web.stanford.edu/~jurafsky/slp3/14.pdf

Join at
slido.com
#3719 083

Further Reading on Dense Retrieval

- Latent Retrieval for Weakly Supervised Open Domain Question Answering [Lee et al., 2019]
- Dense Passage Retrieval for Open-Domain Question Answering [Karpukhin et al., 2020]
- Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval [Xiong et al., 2020]
- Improving language models by retrieving from trillions of tokens [Borgeaud et al., 2021]

Join at
slido.com
#3719 083

Agenda

- Dense Retrieval
- Evaluation of Retrieval
- Retrieval-Augmented Generation (RAG)
- Long-context Issues

Join at
slido.com
#3719 083

Evaluation of IR Systems

- Assume that each document returned by the IR system is either **relevant** to our purposes or **not relevant**
- Given a query, assume the system returns a set of ranked documents $\boldsymbol{T}$
  - A subset $\boldsymbol{R}$ of these are relevant (The remaining $\boldsymbol{N} = \boldsymbol{T} - \boldsymbol{R}$ is irrelevant)
  - There are $\boldsymbol{U}$ documents in the entire retrieval collection that are relevant to this query
- **Precision:** the fraction of the returned documents that are relevant

$$\text{Precision} = \frac{|R|}{|T|}$$

- **Recall:** the fraction of all relevant documents that are returned

$$\text{Recall} = \frac{|R|}{|U|}$$

Join at
slido.com
#3719 083

Precision & Recall @ k

- We hope to build a retrieval system that ranks the relevant documents higher
- Use precision & recall @ k (among the top-k items in the ranked list) to reflect this
- Recall @ k is non-decreasing wrt k

Rank	Judgment	Precision_{Rank}	Recall_{Rank}
1	R	1.0	.11
2	N	.50	.11
3	R	.66	.22
4	N	.50	.22
5	R	.60	.33
6	R	.66	.44
7	N	.57	.44
8	R	.63	.55
9	N	.55	.55
10	N	.50	.55

Assume there are 9 total relevant documents in the retrieval corpus

Figure source: https://web.stanford.edu/~jurafsky/slp3/14.pdf

Join at
slido.com
#3719 083

Average Precision

Average precision (AP): mean of the precision values at the points in the ranked list where a relevant document is retrieved

Indicator function of whether the document is relevant

$$\text{AP} = \frac{1}{|R|} \sum_{k=1}^{|T|} (\text{Precision@}k \times \mathbb{1}(d_k \text{ is relevant}))$$

Rank	Judgment	Precision_{Rank}	Recall_{Rank}
1	R	1.0	.11
2	N	.50	.11
3	R	.66	.22
4	N	.50	.22
5	R	.60	.33
6	R	.66	.44
7	N	.57	.44
8	R	.63	.55
9	N	.55	.55
10	N	.50	.55

Join at
slido.com
#3719 083

Agenda

- Dense Retrieval
- Evaluation of Retrieval
- Retrieval-Augmented Generation (RAG)
- Long-context Issues

Join at
slido.com
#3719 083

RAG for LLMs

RAG is a common technique to enhance factuality of LLM generation & mitigate hallucination

Time-sensitive query triggers RAG for ChatGPT

What is the latest iPhone model?

Searching "latest iPhone model 2024"

Retriever performs web search

Searched 5 sites

Retrieved webpages

"latest iPhone model 2024"
bing.com

Apple — Apple introduces iPhone 16 and iPhone 16 Plus - Apple
apple.com

Apple — Apple debuts iPhone 16 Pro and iPhone 16 Pro Max - Apple
apple.com

MacRumors — iPhone 16: Should You Buy? Advice, Features, Ordering
macrumors.com

TechRadar — iPhone 16: price, cameras, Apple Intelligence features, and everything you need to...
techradar.com

The latest iPhone model as of 2024 is the **iPhone 16**, which launched in September. It comes in two versions: the standard iPhone 16 and the iPhone 16 Plus. Both models feature Apple's new **A18 chip**, improving performance and efficiency, particularly in machine learning and gaming. They also introduce a **Camera Control button**, offering advanced camera functionalities, alongside the **Action button** for customizable shortcuts.

Screenshot taken: https://chatgpt.com/

Join at
slido.com
#3719 083

RAG vs. Direct Prompting

- Prompting relies on LM's parametric knowledge to directly answer the question:

$P(w|$ `Q: Who wrote the book ‘‘The Origin of Species"? A:` $)$ prompt

- RAG prepends the set of retrieved passages to the question

Schematic of a RAG Prompt

```
retrieved passage 1

retrieved passage 2

...

retrieved passage n
```

Returned by the retriever

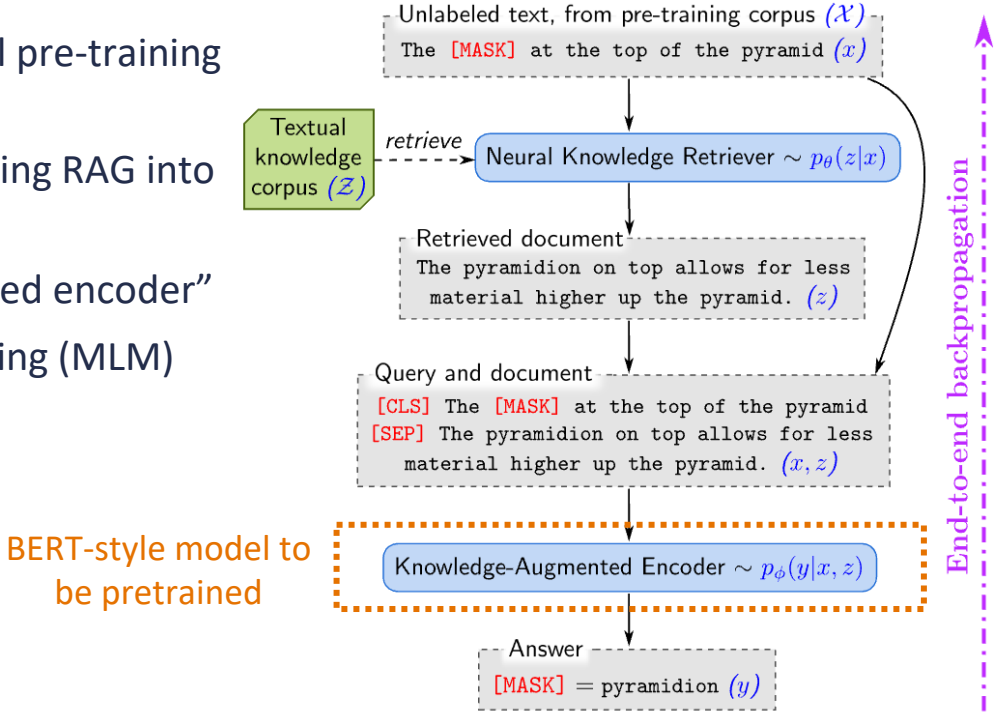
```
Based on these texts, answer this question: Q: Who wrote
the book ‘‘The Origin of Species"? A:
```

Figure source: https://web.stanford.edu/~jurafsky/slp3/14.pdf

Join at
slido.com
#3719 083

RAG in Pretraining

- Retrieval-Augmented Language Model pre-training (REALM)
- The first paper that studies incorporating RAG into encoder pretraining (BERT style)
- Main model is a "knowledge-augmented encoder"
- Pretrain with masked language modeling (MLM) loss conditioned on retrieved content



Paper: https://arxiv.org/pdf/2002.08909

Join at
slido.com
#3719 083

RAG for Text Generation

- The first paper that studies RAG for text generation
- Both the retriever (an encoder-only LM) and the generator (an LLM) are trained

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[†]

Paper: https://arxiv.org/pdf/2005.11401

Join at
slido.com
#3719 083

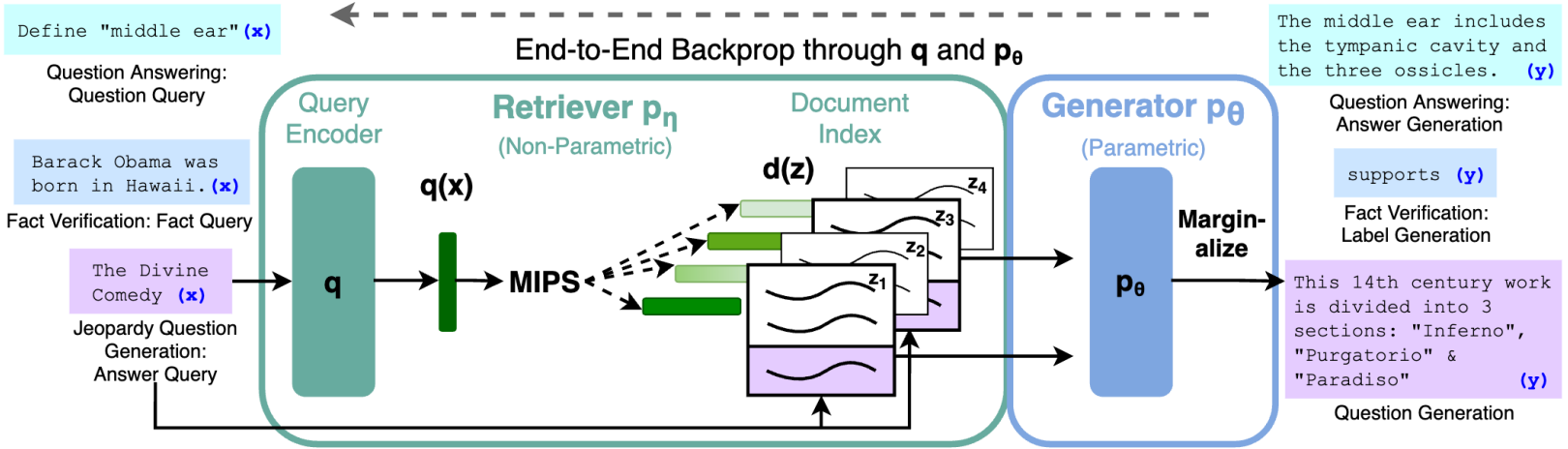
A Latent Variable Model

The retrieved documents are treated as latent variables (z) for generation

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{z}\in\mathcal{D}} p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})$$

Retrieve document (z) based on query (x)

Generate answer (y) based on retrieved docs (z) and query (x)



Join at
slido.com
#3719 083

RAG-Sequence Model

- Use the same retrieved document to generate the complete sequence
- Treat the retrieved document as a single latent variable
- Marginalize to get the generation probability $p(\boldsymbol{y}|\boldsymbol{x})$ via a top-K approximation

$$p_{\text{RAG-sequence}}(\boldsymbol{y}|\boldsymbol{x}) \approx \sum_{\boldsymbol{z} \in \text{top-K}(p(\cdot|\boldsymbol{x}))} p_\eta(\boldsymbol{z}|\boldsymbol{x}) p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{z} \in \text{top-K}(p(\cdot|\boldsymbol{x}))} p_\eta(\boldsymbol{z}|\boldsymbol{x}) \prod_{i=1}^{N} p_\theta(y_i|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_{<i})$$

Top-K approximation
(only consider the top-K retrieved docs)

The same retrieved doc (z) is used to generate all tokens in the sequence

Join at
slido.com
#3719 083

RAG-Token Model

- Can use different retrieved documents to generate different tokens in a sequence
- Marginalization is performed for each generated token (rather than at sequence level)

$$p_{\text{RAG-token}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N} p_\theta(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) \approx \prod_{i=1}^{N} \sum_{\boldsymbol{z} \in \text{top-K}(p(\cdot|\boldsymbol{x}, \boldsymbol{y}_{<i}))} p_\eta(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_{<i}) p_\theta(y_i|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_{<i})$$

Different retrieved doc (z) can be used to generate different tokens in the sequence

Join at
slido.com
#3719 083

RAG-Sequence & RAG-Token Results

Evaluation results on open-domain QA tasks:

- Natural Questions (NQ)
- TriviaQA (TQA)
- WebQuestions (WQ)
- CuratedTrec (CT)

	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**

Join at
slido.com
#3719 083

Further Reading on RAG

- Generalization through Memorization: Nearest Neighbor Language Models [Khandelwal et al., 2019]
- Active Retrieval Augmented Generation [Jiang et al., 2023]
- Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection [Asai et al., 2023]
- InstructRAG: Instructing Retrieval-Augmented Generation via Self-Synthesized Rationales [Wei et al., 2024]

Join at
slido.com
#3719 083

Agenda

- Dense Retrieval
- Evaluation of Retrieval
- Retrieval-Augmented Generation (RAG)
- Long-context Issues

Join at
slido.com
#3719 083

RAG & Long Context Issues in LLMs

- RAG significantly increases the input sequence length to LLMs ("**long context**") by prepending multiple retrieved passages
- **Inefficiency**: the complexity of self-attention is quadratic wrt number of tokens
- **Irrelevant information**: LLMs might get distracted by irrelevant retrieval content
- **Lost in the middle**: LLMs tend to focus more on the beginning and end of the input sequence, but missing important information located in the middle of a long context
- **Performance saturation**: LLMs do not always effectively using the extra context (more retrieved documents)

Join at
slido.com
#3719 083

Lost in the Middle

- Main finding: LLM performance (with RAG) can degrade significantly when changing the position of relevant information
- Performance is often highest when relevant information occurs at the beginning or end of the input context
- Significantly degrades when models must access relevant information in the middle of long contexts
- Analogous to the serial-position effect: a person tends to recall the first and last items in a series best, and the middle items worst

Lost in the Middle: How Language Models Use Long Contexts

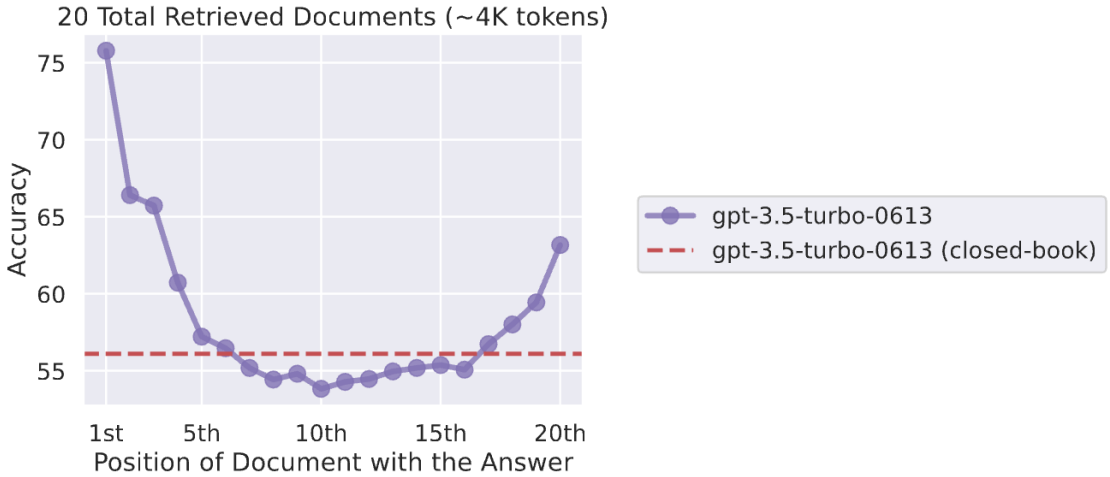
Nelson F. Liu[1*] **Kevin Lin**[2] **John Hewitt**[1] **Ashwin Paranjape**[3]
Michele Bevilacqua[3] **Fabio Petroni**[3] **Percy Liang**[1]
[1]Stanford University [2]University of California, Berkeley [3]Samaya AI

Paper: https://arxiv.org/pdf/2307.03172

Join at
slido.com
#3719 083

Primacy & Recency Bias

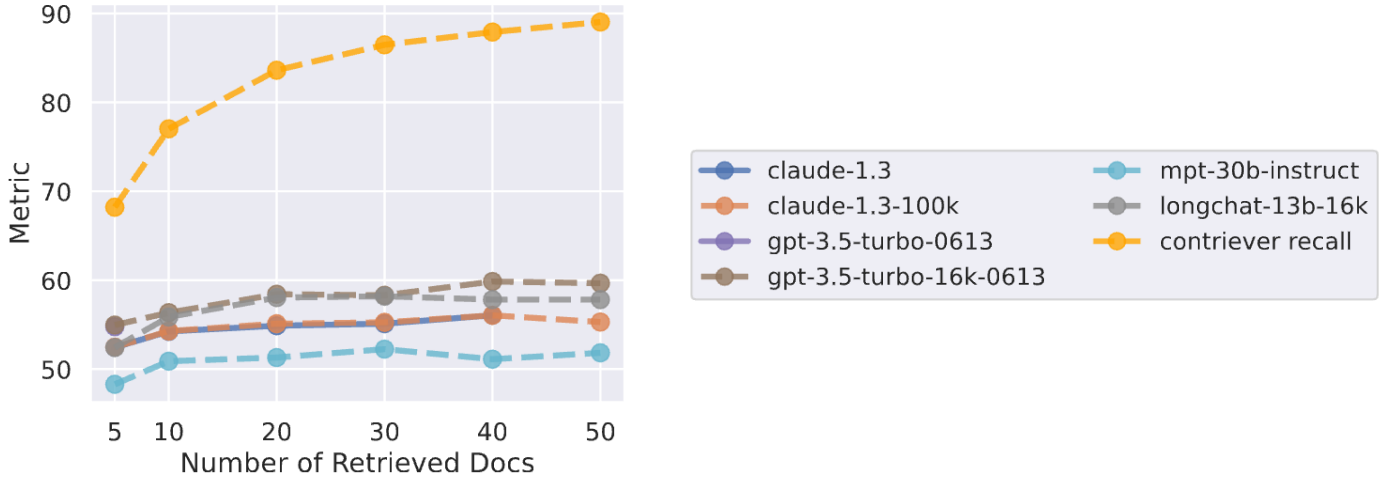
- Exactly one of the documents contains the answer, with other “distractor” documents
- Vary the position of the gold document
- U-shaped performance curve: LLMs are better at using relevant information that occurs at the very beginning (**primacy bias**) or end of its input context (**recency bias**)



Join at
slido.com
#3719 083

Performance Saturation Under More Context

- Retriever recall always improves with more retrieved docs
- LLM performance saturates long before retriever performance saturates (using more than 20 retrieved documents only marginally improves LLM performance)



Join at
slido.com
#3719 083

Practical Considerations of RAG

- Retrieve fewer documents when appropriate
- Avoid inputting irrelevant information to LLMs if possible
- Re-rank retrieved documents to push relevant information closer to the start of the input context
- Adjust & refine retrieval queries based on intermediate results or feedback
- Optimize document chunking: break entire documents into smaller, semantically coherent chunks to ensure only the most relevant parts are input to the LLM

Join at
slido.com
#3719 083

Further Reading on Long-context LLMs

- Efficient Streaming Language Models with Attention Sinks [Xiao et al., 2023]
- LongNet: Scaling Transformers to 1,000,000,000 Tokens [Ding et al., 2023]
- Adapting Language Models to Compress Contexts [Chevalier et al., 2023]
- LLM Maybe LongLM: Self-Extend LLM Context Window Without Tuning [Jin et al., 2024]

Thank You!

Yu Meng
University of Virginia
yumeng5@virginia.edu