



Non-Parametric Knowledge & Retrieval

Yu Meng

University of Virginia
yumeng5@virginia.edu

Oct 28, 2024

Overview of Course Contents

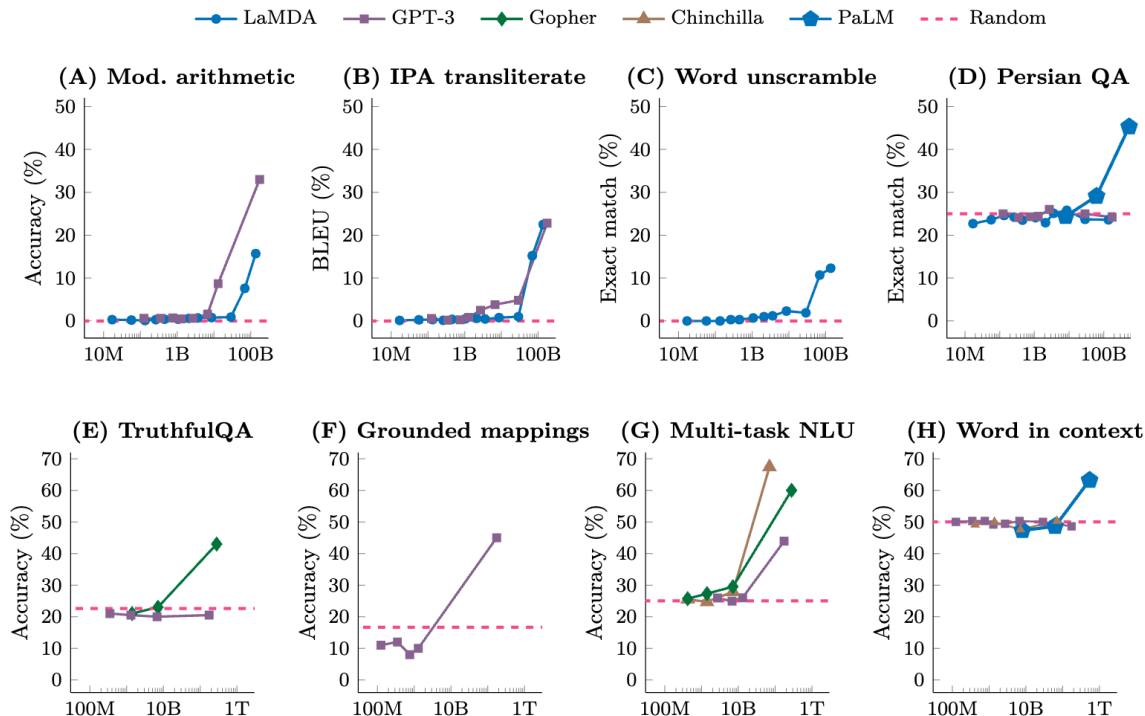
Join at
slido.com
#3045 797



- 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

(Recap) Emergent Ability

Join at
 slido.com
 #3045 797



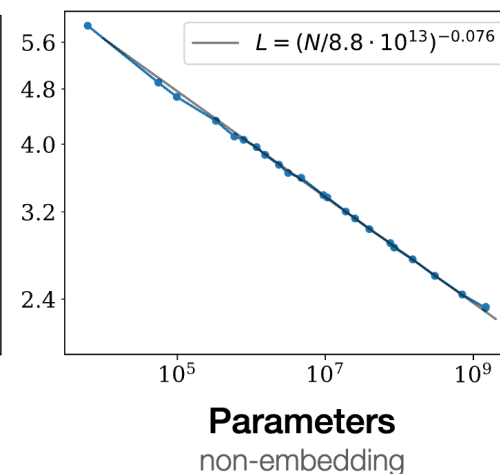
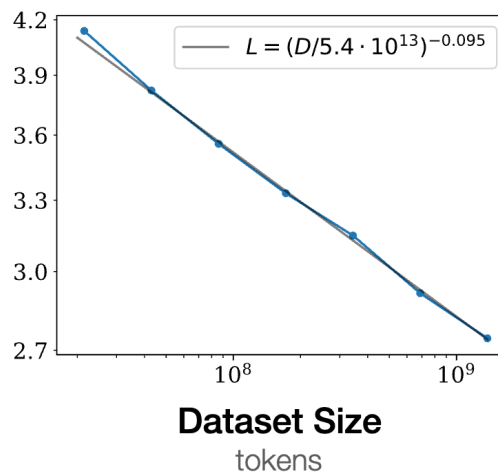
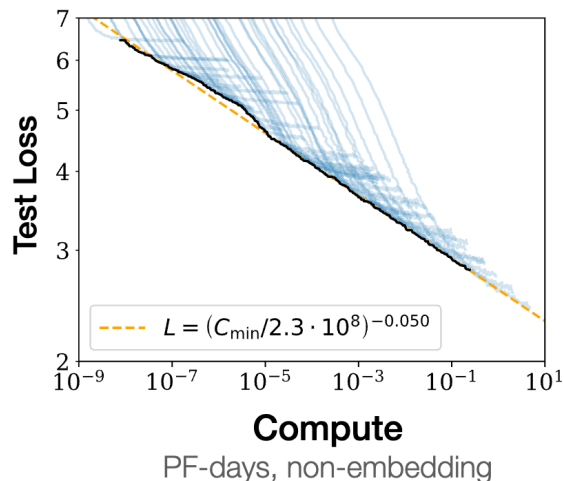
Model scale (number of parameters)

Models exhibit random performance until a certain scale, after which performance significantly increases



(Recap) Scaling Laws of LLMs

Performance has a power-law relationship with each of the three scale factors (model size, dataset size, compute) when not bottlenecked by the other two





(Recap) Reasoning: Overview

- **Reasoning** (rough definition): perform deductive, inductive, commonsense, or logical reasoning via generating or analyzing text with language models
- Deductive reasoning: draw specific conclusions from general principles or premises
 - E.g.: “All humans are mortal” + “Socrates is a human” => “Socrates is mortal”
- Inductive reasoning: make generalizations based on specific observations
 - E.g.: “The sun has risen in the east every day” => “The sun will rise in the east tomorrow”
- Commonsense reasoning: rely on world knowledge or commonsense understanding to make predictions or answer questions
 - E.g.: “If I drop a ball, what will happen?” => “It will fall”
- Mathematical/logical reasoning: follow specific rules or procedures to arrive at a correct answer
 - E.g.: “If 3 apples cost \$6, how much do 5 apples cost?” => “\$10”



(Recap) Standard vs. CoT Prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅



(Recap) CoT Can Be Triggered Zero-shot

Just add “Let’s think step by step” at the beginning of the answer

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) *The answer is 8.* ❌

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) *The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4.* ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) *8* ❌

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

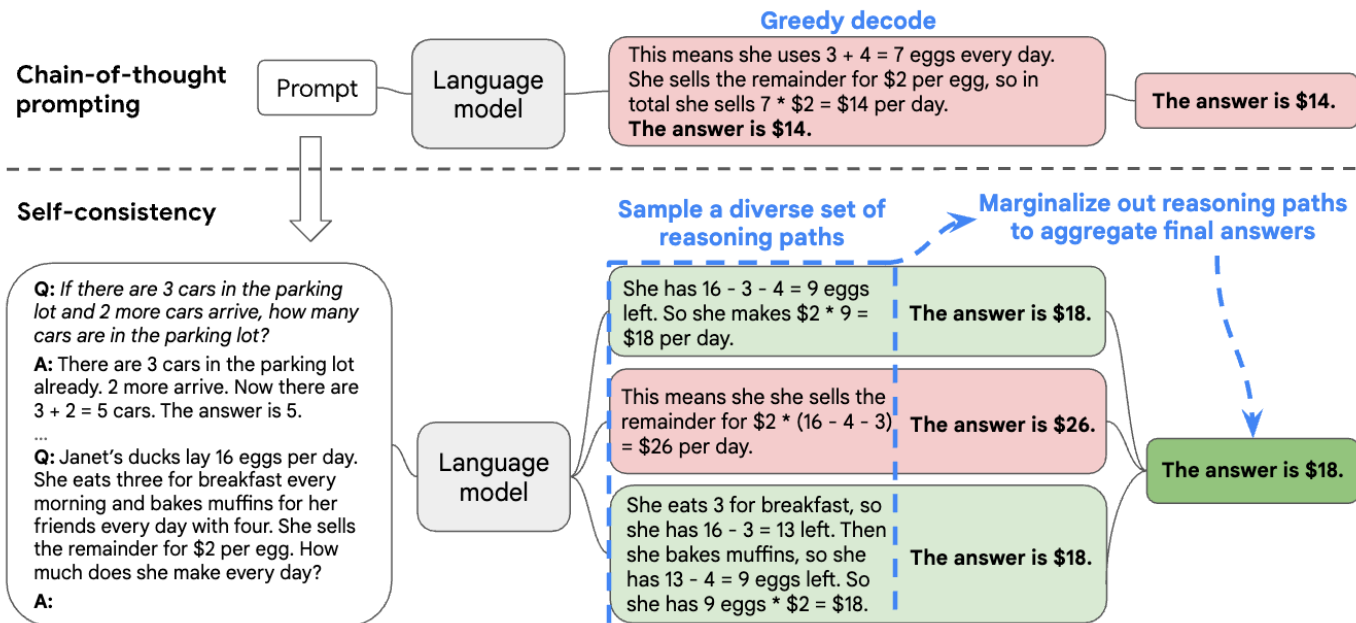
A: **Let’s think step by step.**

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✓



(Recap) Self-consistency CoT

Intuition: if multiple different ways of thinking lead to the same answer, one has greater confidence that the final answer is correct





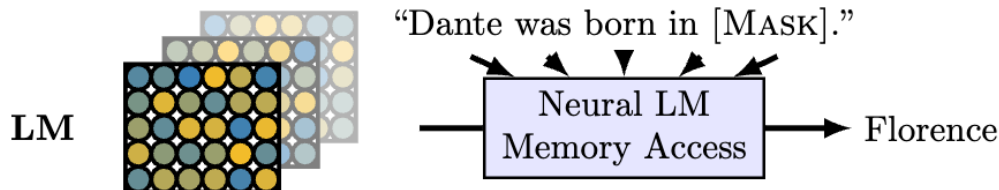
(Recap) Question Answering

- **Question Answering (QA):** build systems that can automatically answer questions posed by humans in natural language
- Categorization by application domain: closed-domain vs. open-domain QA
 - **Closed-domain** QA: answer questions within a specific domain
 - **Open-domain** QA: answer questions from any domain
- Categorization by modeling approach: extractive vs. abstractive QA
 - **Extractive** QA: output a span of text extracted directly from a given context
 - **Abstractive** QA: synthesize the answer in its own words (rephrasing/summarizing)
- Categorization by access to external source: closed-book vs. open-book QA
 - **Closed-book** QA: answer questions without access to any external information
 - **Open-book** QA: can access external knowledge source to answer the questions



(Recap) Language Model as Knowledge Bases

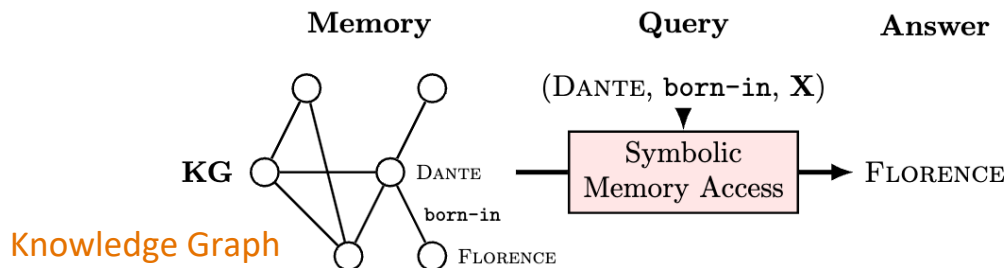
- **Acquisition:** LM's knowledge is derived from the vast amount of pretraining data
- **Access:** information is accessed through natural language prompts
- **Update/maintenance:** re-training/fine-tuning the model with new data
- **Pros:**
 - Handle a wide range of natural language queries with contextual understanding
 - Generalize to unseen queries not seen during training
- **Cons:**
 - May produce incorrect/outdated information
 - Lack interpretability/transparency





(Recap) Real Knowledge Bases

- **Acquisition:** manually constructed by human annotators
- **Access:** information is accessed through queries in specific formats
- **Update/maintenance:** adding/modifying/deleting entries (incrementally) by humans
- **Pros:**
 - Precise & verifiable
- **Cons:**
 - Not able to handle natural language
 - Require massive human efforts to construct & maintain





(Recap) FFNs Are Neural Memories

Viewing FFNs as key-value memories

$$\text{FFN}(\mathbf{x}_i) = \text{ReLU}(\mathbf{x}_i \mathbf{W}_1) \mathbf{W}_2$$



$$\mathbf{x}_i \in \mathbb{R}^{d_1}$$

$$\text{FFN}(\mathbf{x}_i) = \text{ReLU}(\mathbf{x}_i \mathbf{K}) \mathbf{V}$$

$$\mathbf{K} \in \mathbb{R}^{d_1 \times d_2}$$

$$\mathbf{V} \in \mathbb{R}^{d_2 \times d_1}$$

key vectors (column vectors in \mathbf{K}) act as **pattern detectors** over the input sequence

value vectors (row vectors in \mathbf{V}) represent **distributions over the output vocabulary**

$$\text{FFN}(\mathbf{x}_i) = \sum_{j=1}^{d_2} \text{ReLU}(\mathbf{x}_i \cdot \mathbf{k}_j) \mathbf{v}_j$$

weights of value vectors

Agenda

- Hallucination
- Non-parametric Knowledge & Retrieval
- Sparse Retrieval (TF-IDF)
- Dense Retrieval
- Evaluation of Retrieval

Join at
slido.com
#3045 797



Hallucination

Join at
slido.com
#3045 797



- **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
 - ...



Hallucination Examples

- **(Limited knowledge)** Q: “What were the main features of the iPhone 15 Pro Max?”
LLM (trained before 2023): “The iPhone 15 Pro Max features a revolutionary holographic display, quantum computing chip, and telepathic user interface.”
- **(Overgeneralization)** Q: “How do you form the past tense in Japanese?”
LLM: “In Japanese, you typically add '-ed' to the end of verbs to form the past tense, just like in English.” (incorrect)
- **(Lack of common sense)** Q: “How many tennis balls can fit in a typical smartphone?”
LLM: “Approximately 15-20 tennis balls can fit in a typical smartphone, depending on the model and screen size.”

what's your knowledge cutoff date?



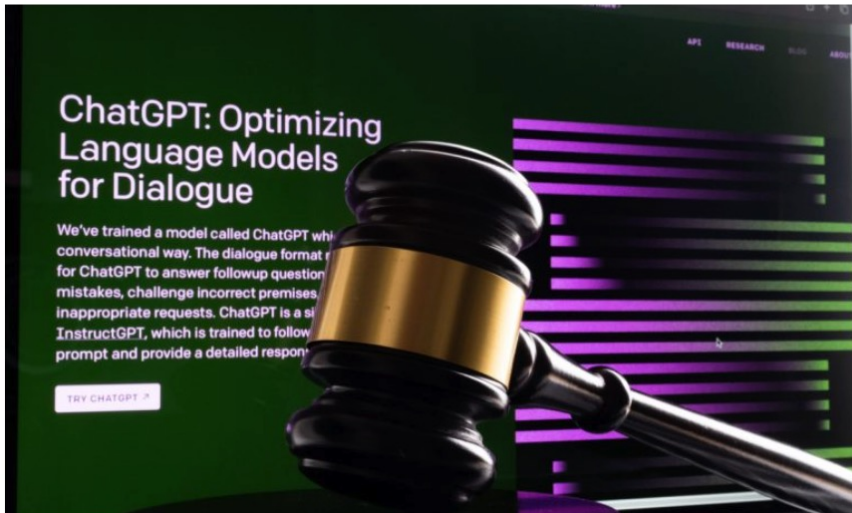
My knowledge cutoff date is October 2023. This means I don't have information on events or developments that have occurred after that time. How can I assist you with your question?

Concerns About Hallucination

Still a concerning issue in modern LLMs!

Attorneys Face Sanctions After Citing Case Law 'Hallucinated' by ChatGPT

BY PYMNTS | MAY 30, 2023



Join at
slido.com
#3045 797



Figure source: <https://www.pymnts.com/artificial-intelligence-2/2023/attorneys-face-sanctions-after-citing-information-hallucinated-by-chatgpt/>



Further Reading on Hallucination

- [LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples](#) [Yao et al., 2023]
- [Towards Mitigating Hallucination in Large Language Models via Self-Reflection](#) [Ji et al., 2023]
- [Hallucination is Inevitable: An Innate Limitation of Large Language Models](#) [Xu et al., 2024]

Agenda

- Hallucination
- Non-parametric Knowledge & Information Retrieval
- Sparse Retrieval (TF-IDF)
- Dense Retrieval
- Evaluation of Retrieval

Join at
slido.com
#3045 797





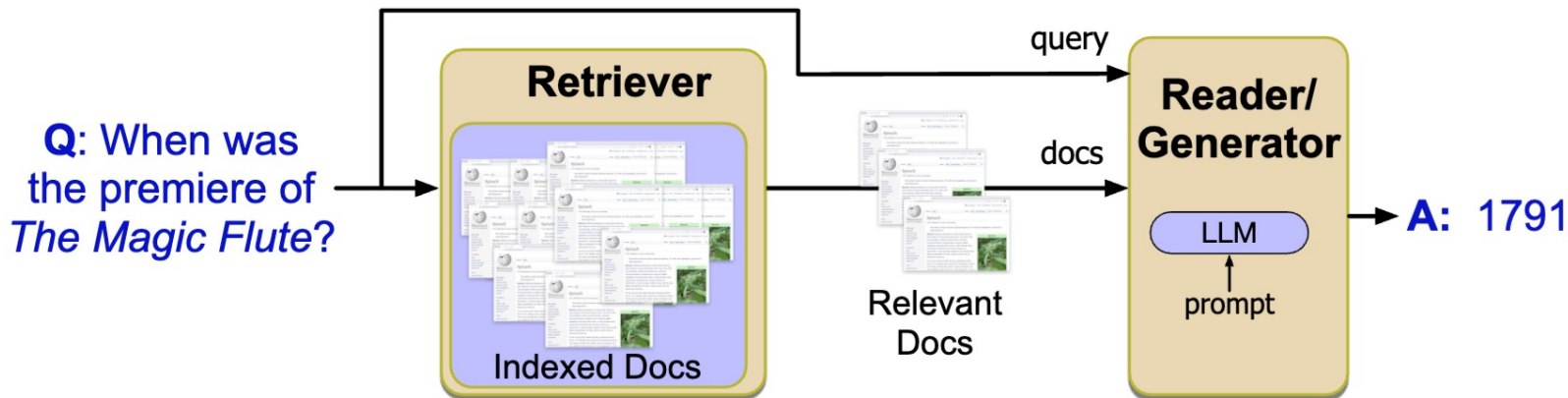
Non-parametric Knowledge

- **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



Overview: 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





Overview: 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



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

Agenda

- Hallucination
- Non-parametric Knowledge & Information Retrieval
- Sparse Retrieval (TF-IDF)
- Dense Retrieval
- Evaluation of Retrieval

Join at
slido.com
#3045 797





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

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

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

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

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

Term Frequency (TF)

Join at
slido.com
#3045 797



- A word appearing 100 times in a document doesn't make it 100 times more likely to be relevant to the meaning of the document
- Instead of using the raw counts, we squash the counts with log scale

$$\text{TF}(w, d) = \begin{cases} 1 + \log_{10} \text{count}(w, d) & \text{count}(w, d) > 0 \\ 0 & \text{otherwise} \end{cases}$$



Inverse Document Frequency (IDF)

- We want to emphasize discriminative words (with low DF)
- Inverse document frequency (IDF): total number of documents (N) divided by DF, in log scale

$$\text{IDF}(w) = \log_{10} \left(\frac{N}{\text{DF}(w)} \right)$$

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

DF & IDF statistics in the
Shakespeare corpus
(37 documents)

TF-IDF for Sparse Retrieval

Join at

slido.com

#3045 797



- Score document-query semantic similarity by cosine similarity

$$\cos(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{|\mathbf{q}| |\mathbf{d}|}$$

- Both document and query vectors use TF-IDF weighting
- Can also adopt other weighting schemes (e.g., BM25)



Example: TF-IDF for Sparse Retrieval

- Example query and mini-corpus:
- Query & document vectors:

Query: sweet love

Doc 1: Sweet sweet nurse! Love?

Doc 2: Sweet sorrow

Doc 3: How sweet is love?

Doc 4: Nurse!

word	Query					
	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

word	Document 1			
	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

	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	

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

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

Agenda

- Hallucination
- Non-parametric Knowledge & Information Retrieval
- Sparse Retrieval (TF-IDF)
- Dense Retrieval
- Evaluation of Retrieval

Join at
slido.com
#3045 797



Dense Retrieval

Join at
slido.com
#3045 797

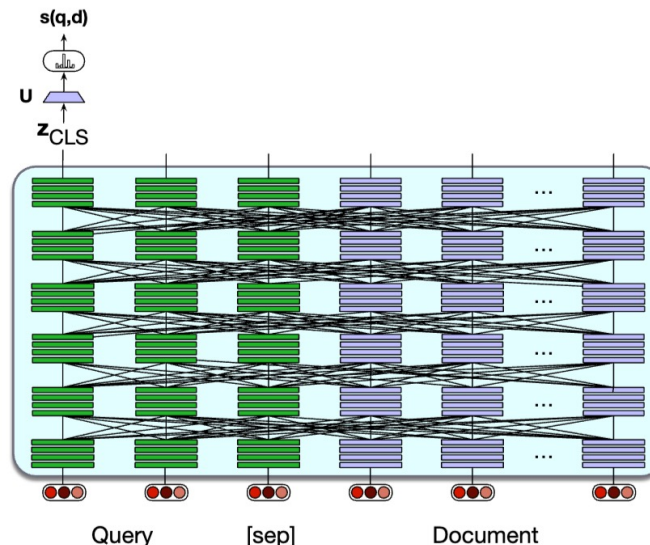


- Motivation: sparse retrieval (e.g., TF-IDF) relies on the exact overlap of words between the query and document without considering semantic similarity
- Solution: use a language model to obtain (dense) distributed representations of query and document
- The retriever language model is typically a 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



Dense Retrieval: Cross-encoder

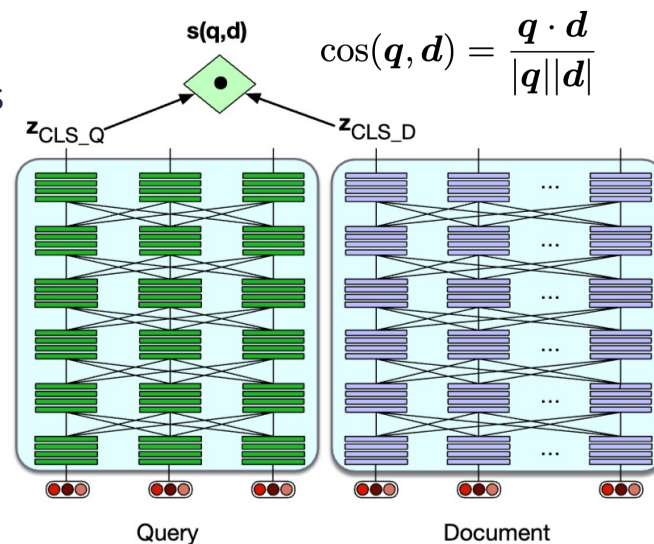
- Process query-document pairs together
- Relevance score produced directly by the model output
- (+) Capture intricate interactions between the query and the document
- (-) Not scalable to large retrieval corpus
- Good for small document sets





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



Agenda

- Hallucination
- Non-parametric Knowledge & Information Retrieval
- Sparse Retrieval (TF-IDF)
- Dense Retrieval
- Evaluation of Retrieval

Join at
slido.com
#3045 797





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 T
 - A subset R of these are relevant (The remaining $N = T - R$ is irrelevant)
 - There are 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

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



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

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



Average Precision

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

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

Indicator function of whether
the document 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



Thank You!

Yu Meng

University of Virginia

yumeng5@virginia.edu