Retrieval-based LMs

Large Language Models: Introduction and Recent Advances

ELL881 · AIL821

Yatin Nandwani
Research Scientist, IBM Research

Outline

- Motivation
  - Drawbacks of Parametric LLMs – *hallucination, verification …*
  - Motivating Retrieval-based LLMs *– close book vs open book*
- Major components of Retrieval-based LLMs *– index, retrieve, read …*
- Retrieval Methods – *sparse, dense, reranking, black-box*
- kNN, RETRO, REALM, RAG – *seminal works*
- Overview of Training Techniques – *independent, sequential, joint training …*
- Limitations – *lost in the middle, still hallucinating, retriever failures …*

Closed Book vs Open Book Exams

Parametric LLMs

Retrieval-based LLMs

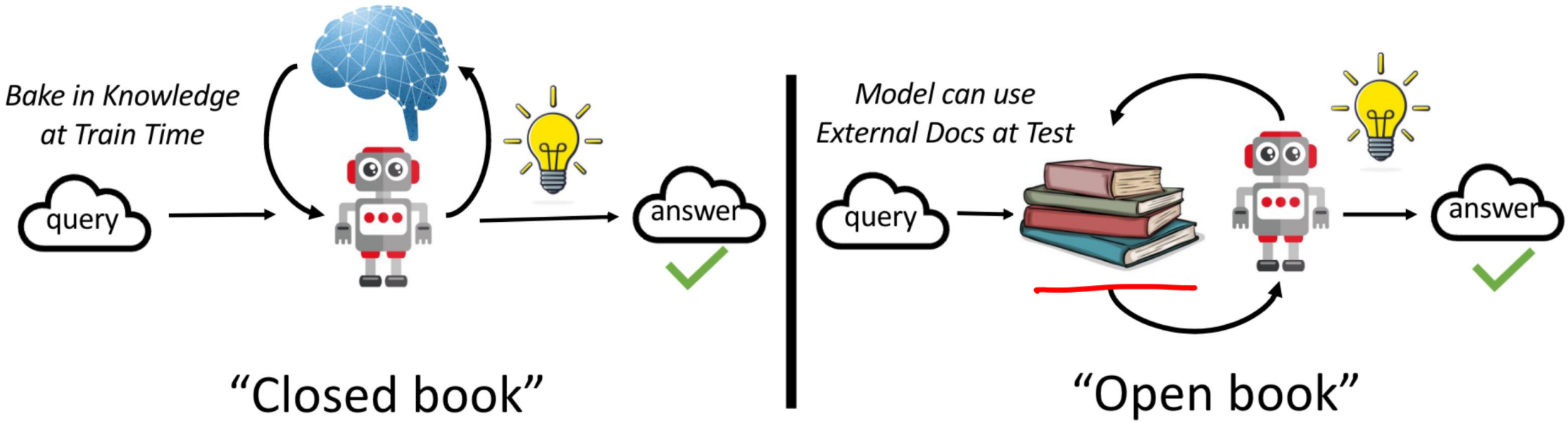
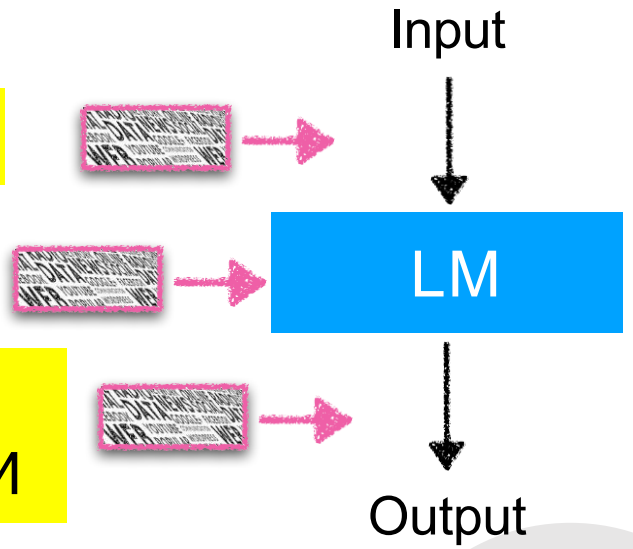


Image source: http://arxiv.org/abs/2403.10131

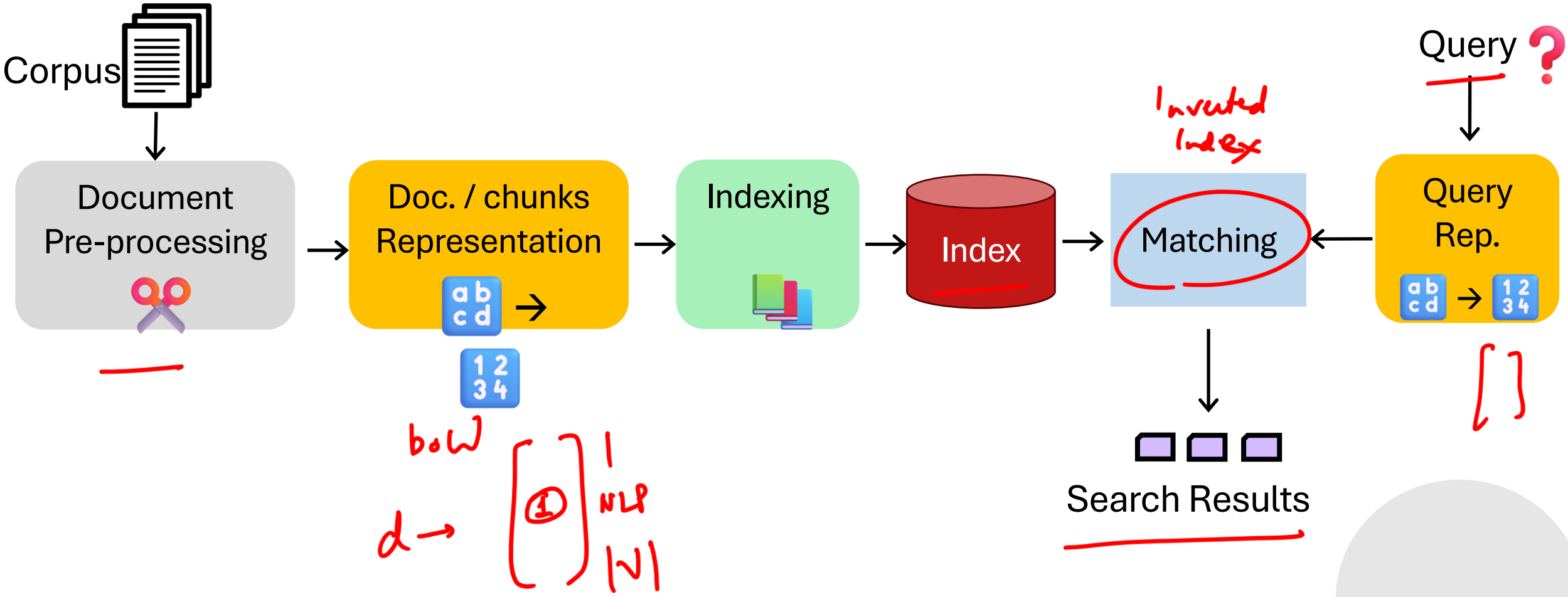
How to use the Book?

- Output interpolations - After solving the question yourself? kNN LMs
- Intermediate fusion – modify the LM architecture to be aware of the book? RETRO
- Input augmentation (RAG) - Before you start solving? RAG, REALM

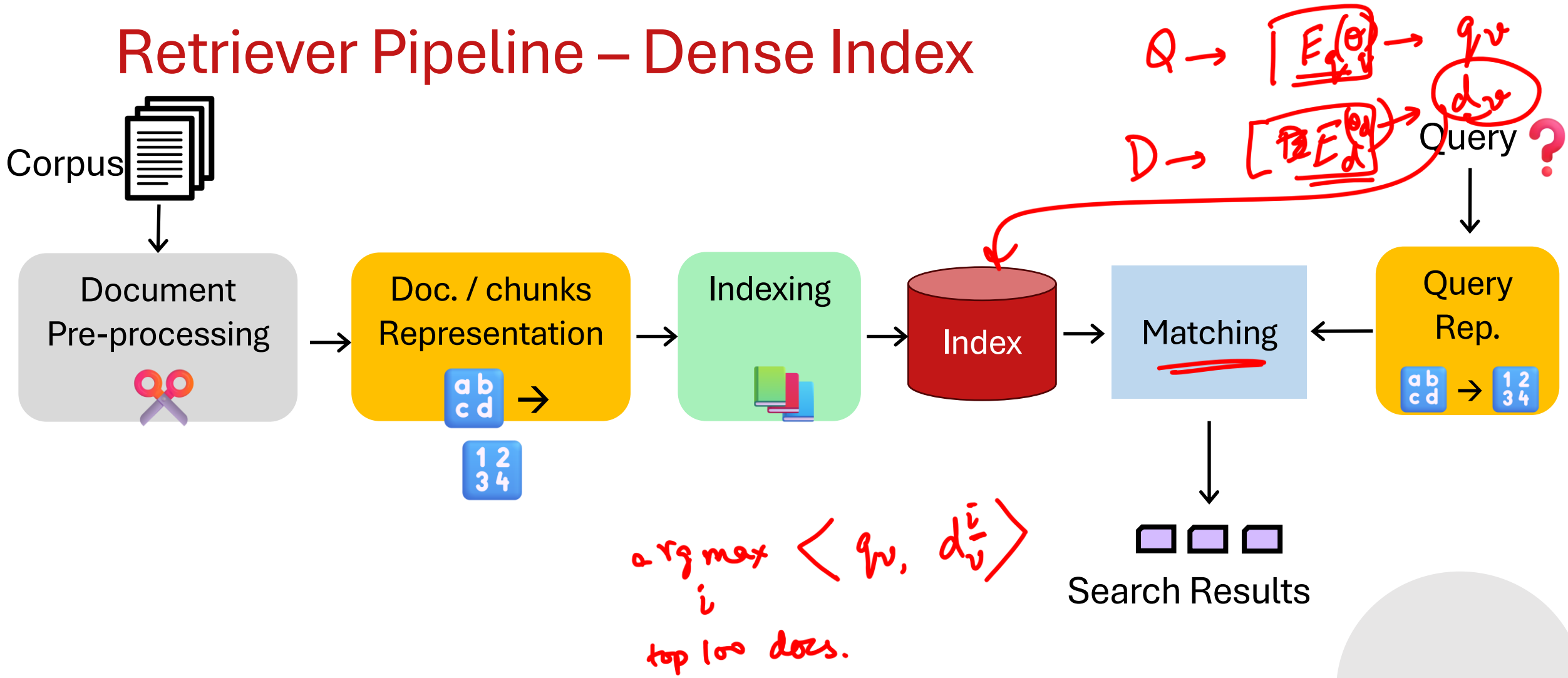


Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

Retriever Pipeline – Sparse Index



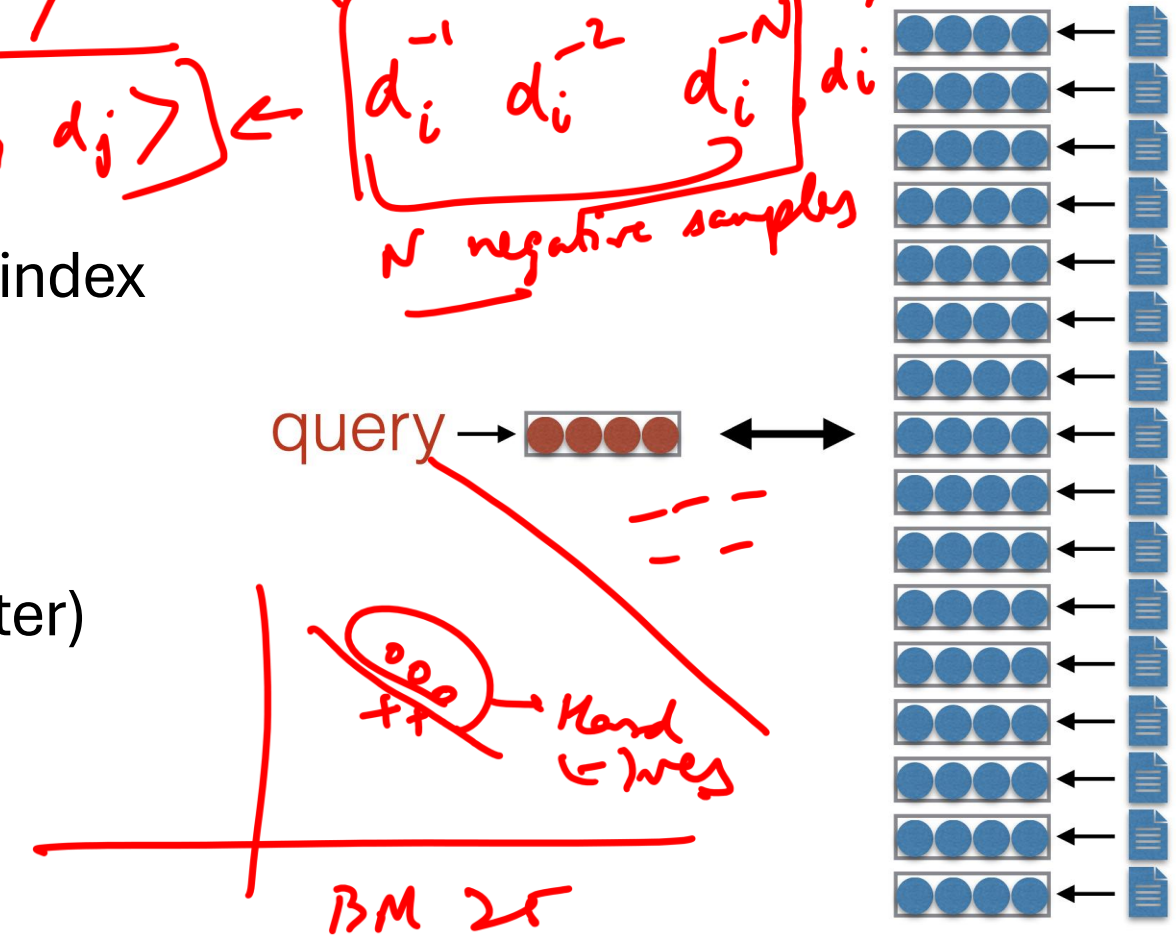
Retriever Pipeline – Dense Index



Dense Embeddings

- Encode all documents using a LM and index them (one time task). Can use:
  - ✓ Out-of-the-box embeddings. E.g. BERT
  - ✓ Learned embeddings (covered later)

- At test time:
  - Encode Query
  - Use Nearest Neighbor Search to find similar documents

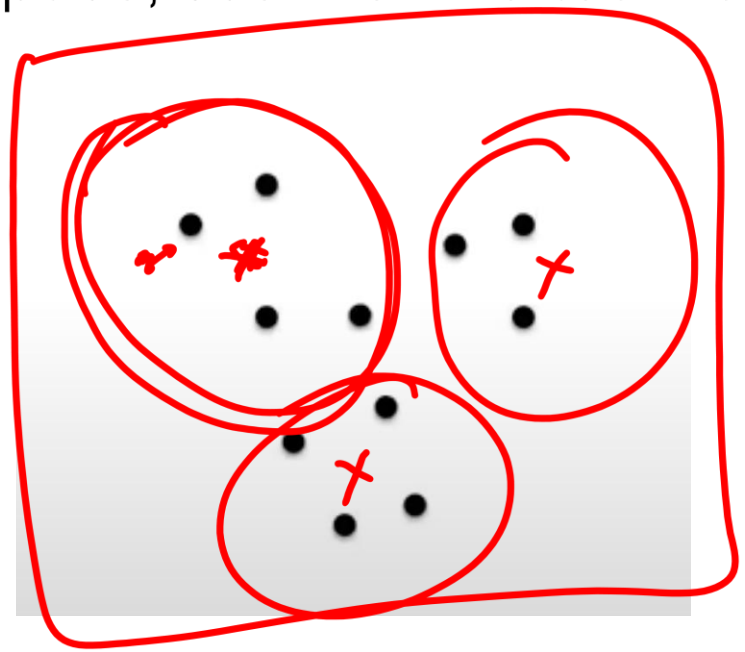


Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Approximate Nearest Neighbor Search
Maximum Inner Product Search (MIPS)

- Methods to retrieve embeddings in sub-linear time

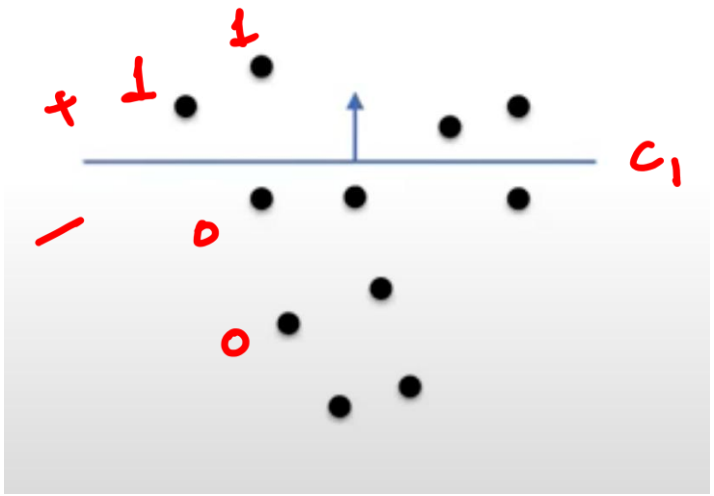
Locality sensitive hashing:
make partitions in continuous space, use like inverted index



Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Approximate Nearest Neighbor Search (MIPS)

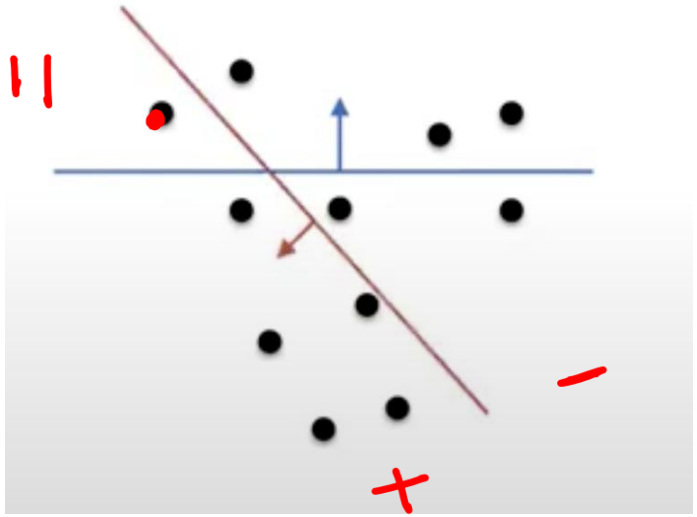
Locality sensitive hashing:
make partitions in continuous space, use like inverted index



Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Approximate Nearest Neighbor Search (MIPS)

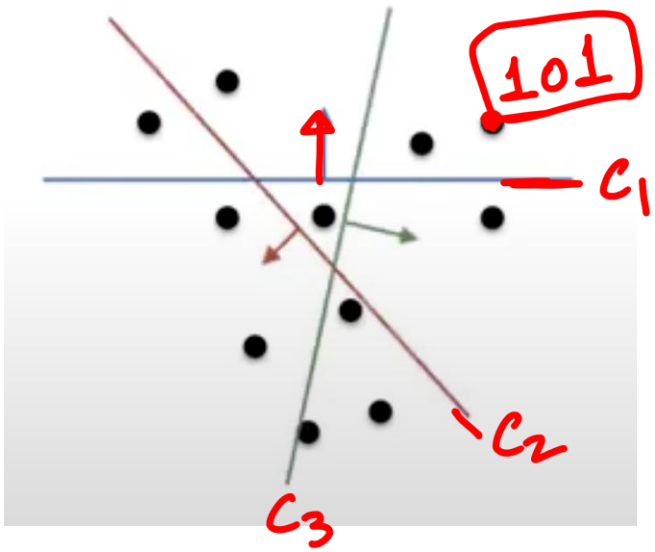
Locality sensitive hashing:
make partitions in continuous space, use like inverted index



Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Approximate Nearest Neighbor Search (MIPS)

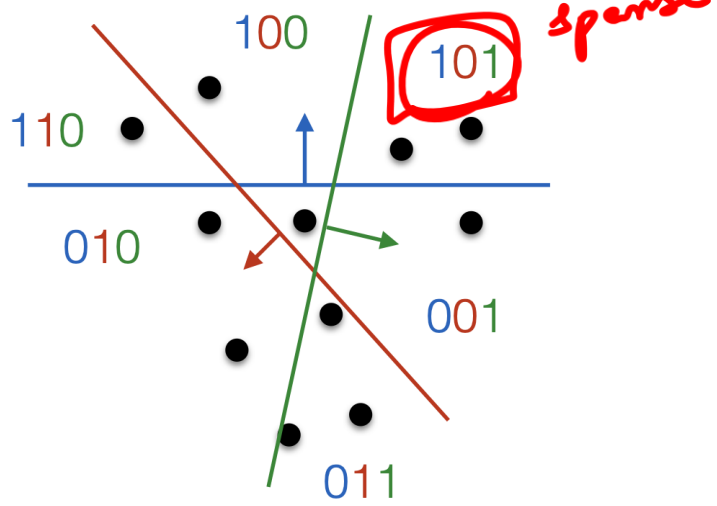
Locality sensitive hashing:
make partitions in continuous space, use like inverted index



Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Approximate Nearest Neighbor Search (MIPS)

Locality sensitive hashing: make partitions in continuous space, use like inverted index

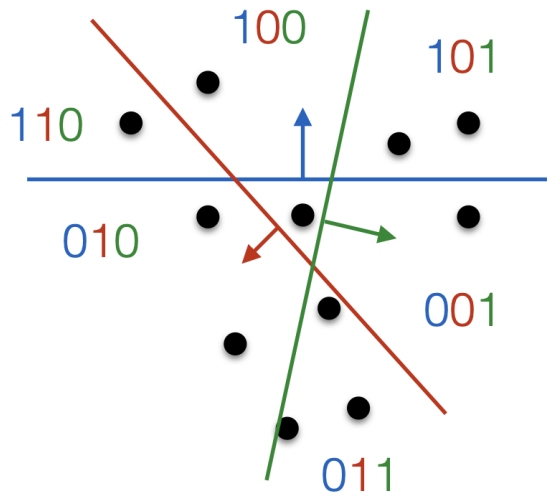


Graph-based search: create "hubs" and search from there

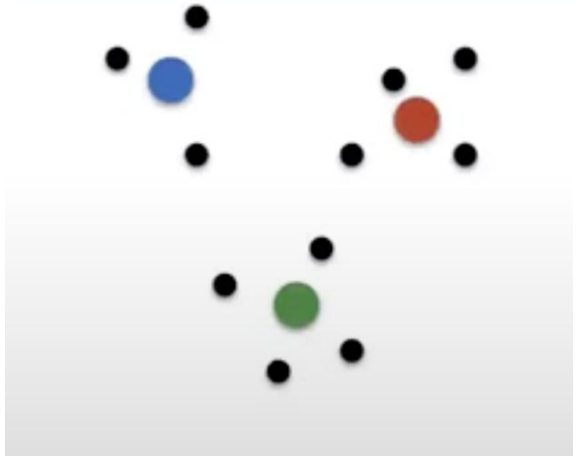
Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Approximate Nearest Neighbor Search (MIPS)

Locality sensitive hashing: make partitions in continuous space, use like inverted index



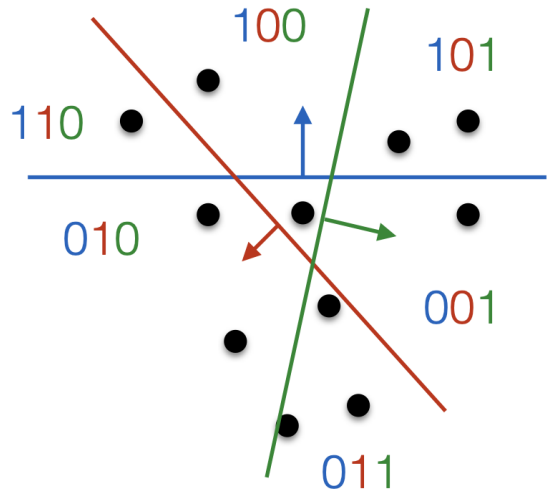
Graph-based search: create "hubs" and search from there



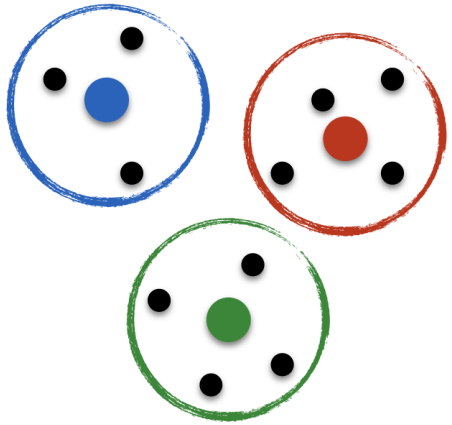
Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Approximate Nearest Neighbor Search (MIPS)

Locality sensitive hashing: make partitions in continuous space, use like inverted index

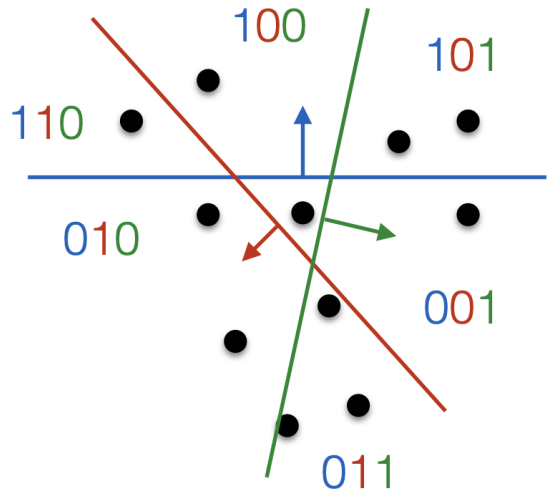


Graph-based search: create "hubs" and search from there

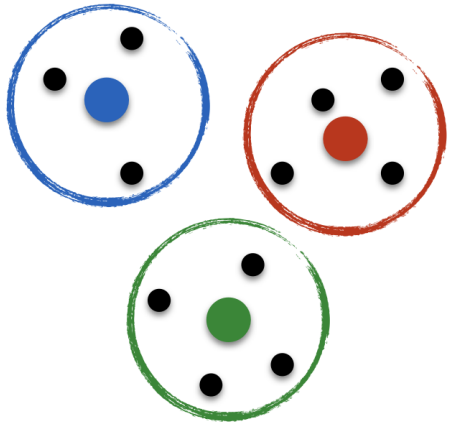


Approximate Nearest Neighbor Search (MIPS)

Locality sensitive hashing: make partitions in continuous space, use like inverted index

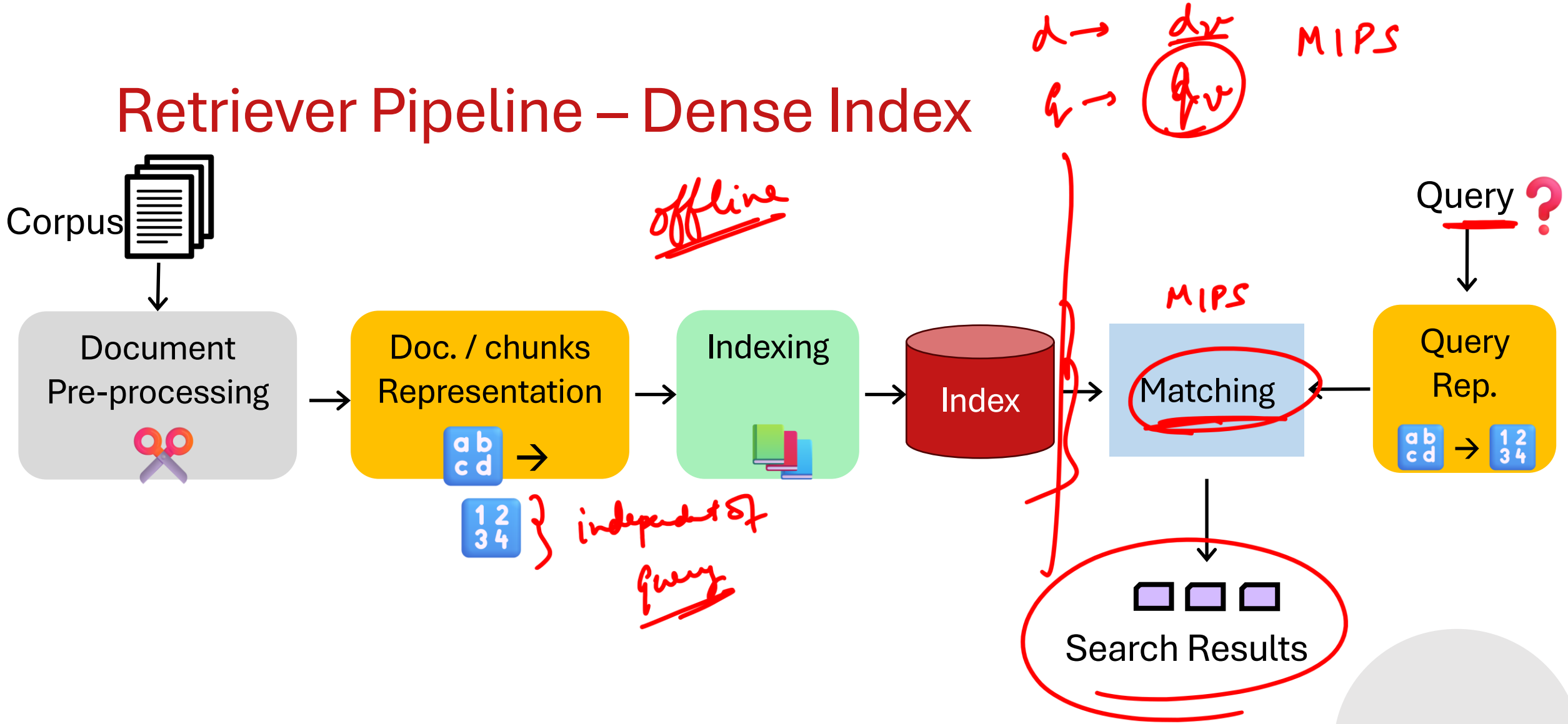


Graph-based search: create "hubs" and search from there

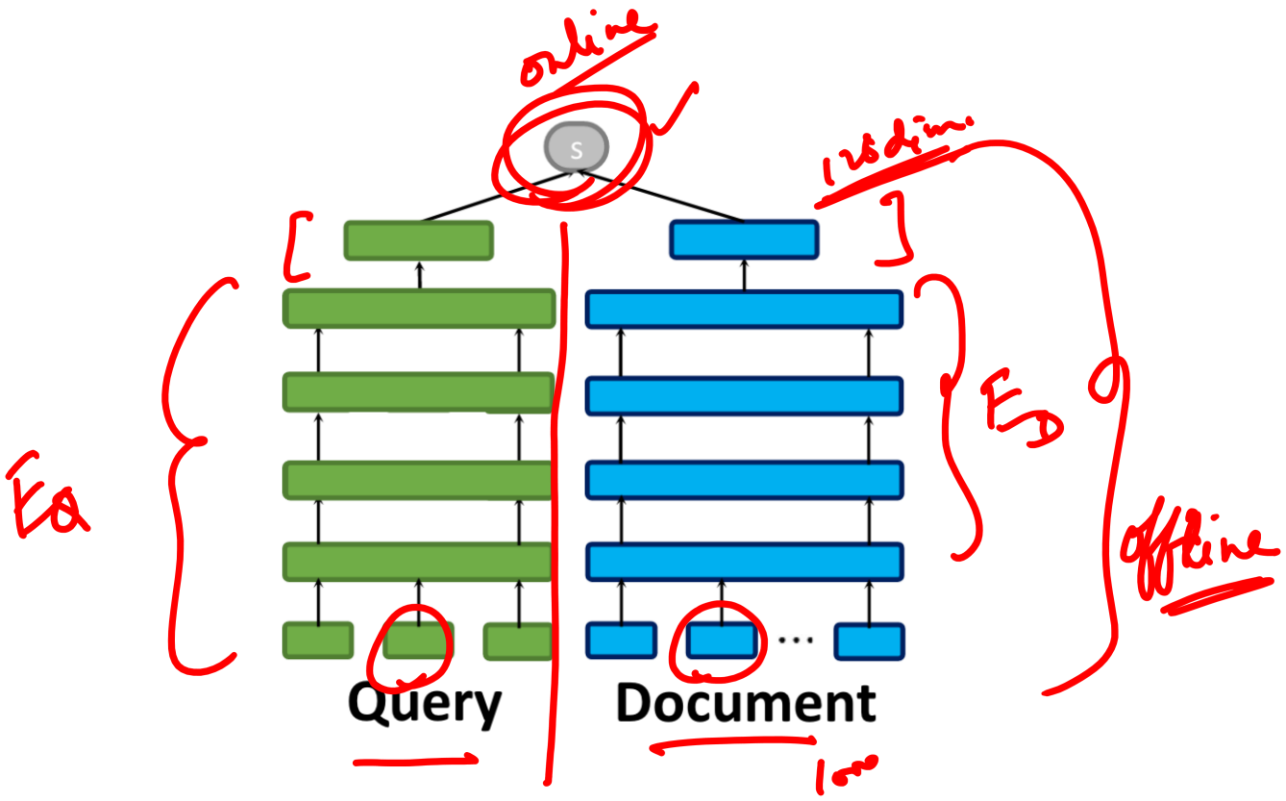


- Software: ANNOY (Spotify), FAISS

Retriever Pipeline – Dense Index



Bi-Encoder Scoring

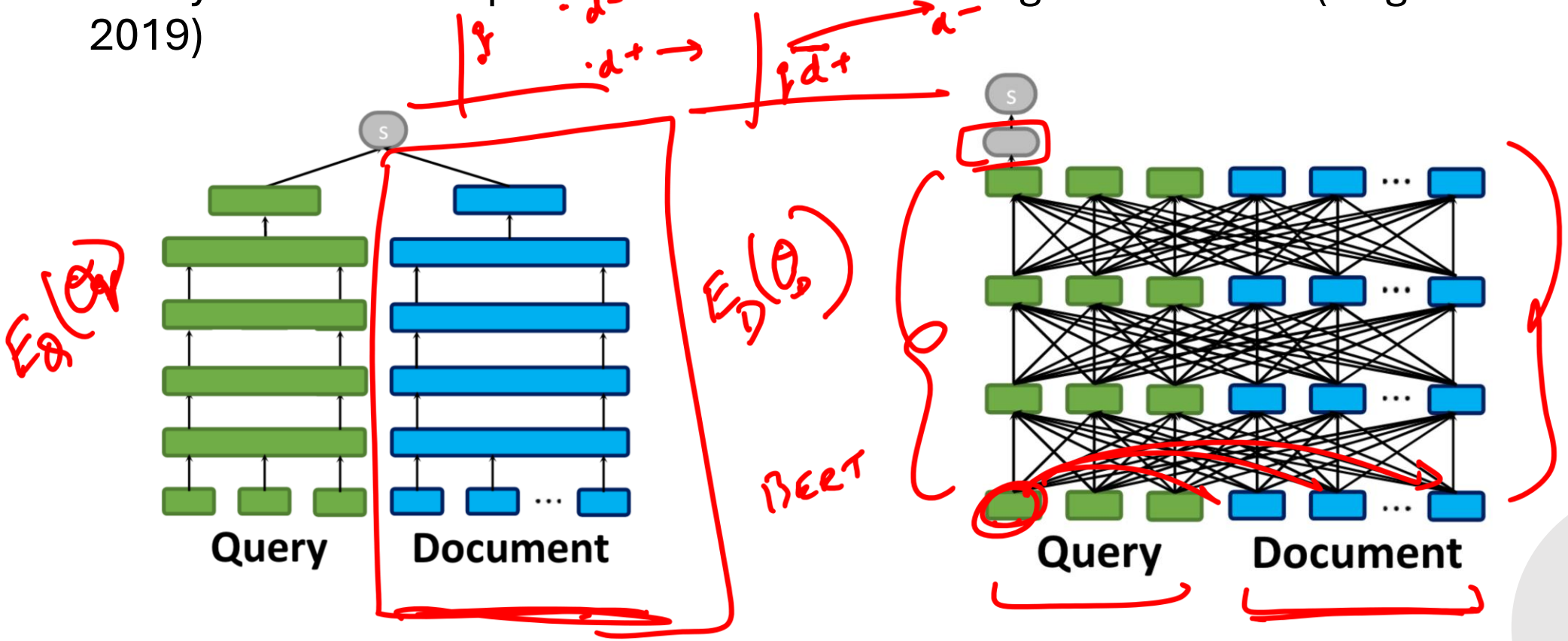


Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Figure from Khattab et al. (2020)

Cross-Encoder Reranking

- Jointly encode both queries and documents using neural model (Nogueira et al. 2019)

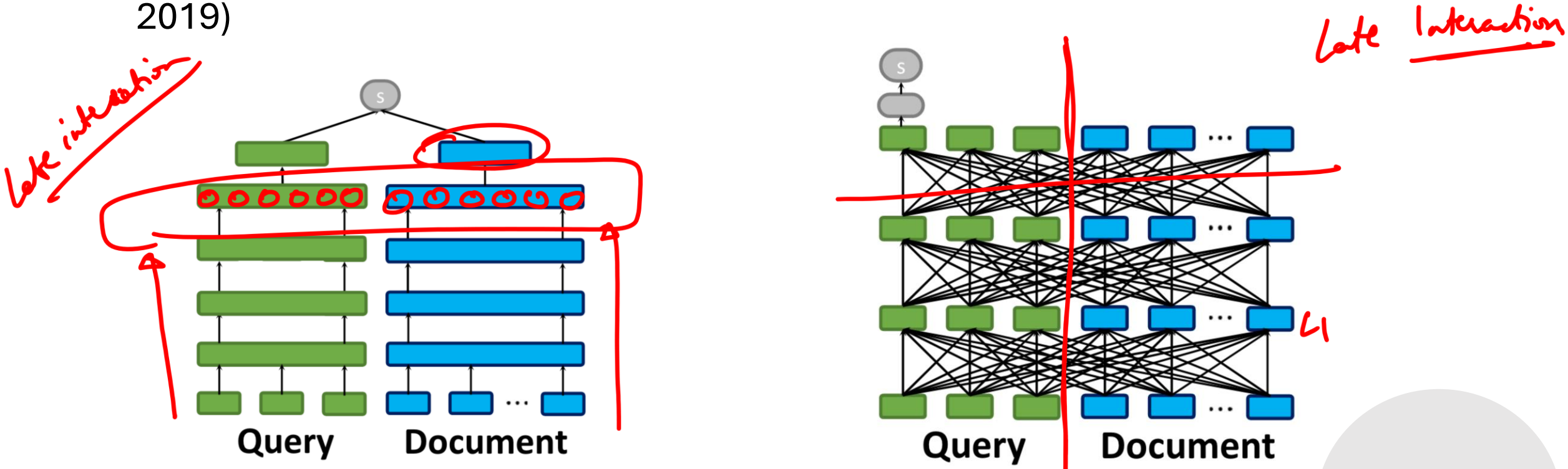


Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Figure from Khattab et al. (2020)

Cross-Encoder Reranking

- Jointly encode both queries and documents using neural model (Nogueira et al. 2019)



- Precludes approximate nearest neighbour lookup, so can only be used on small number of candidates

Figure from Khattab et al. (2020)

Token-level Dense Retrieval

ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

Significantly more effective (but more costly) than single-vector retrieval

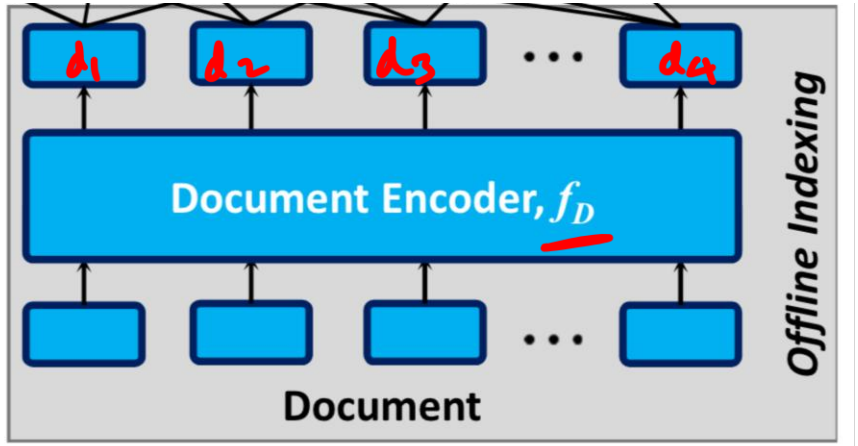
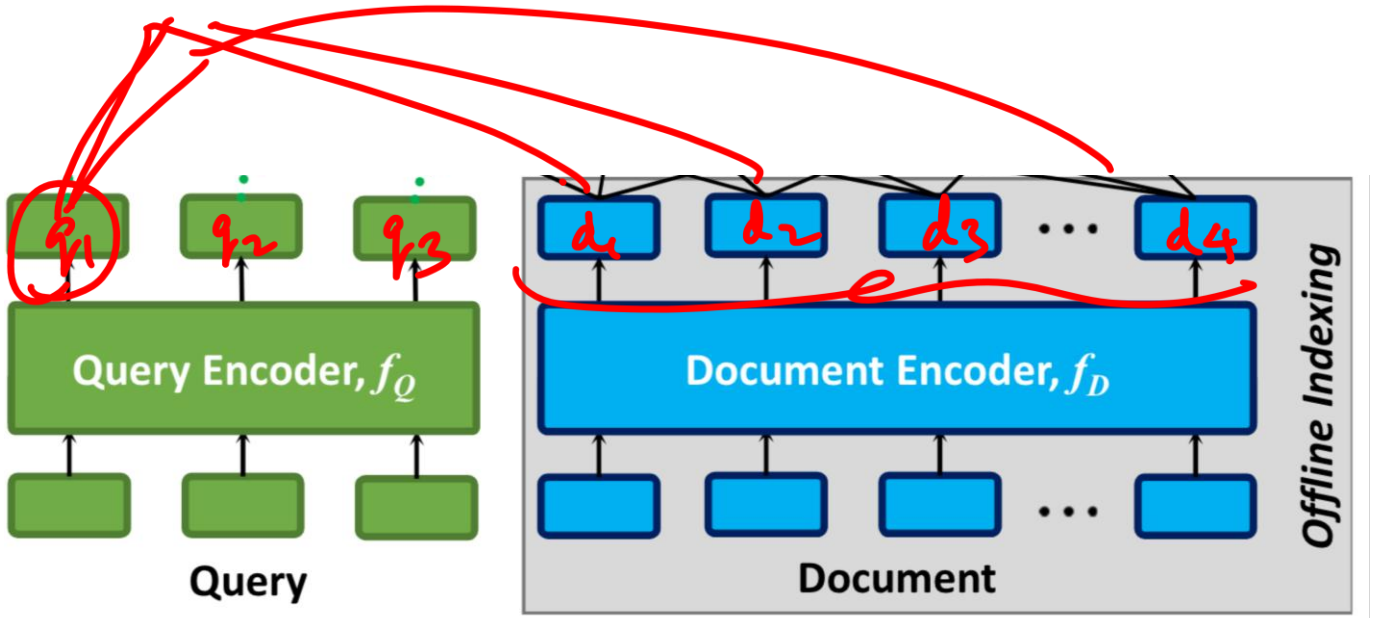


Figure from Khattab et al. (2020)

Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Token-level Dense Retrieval



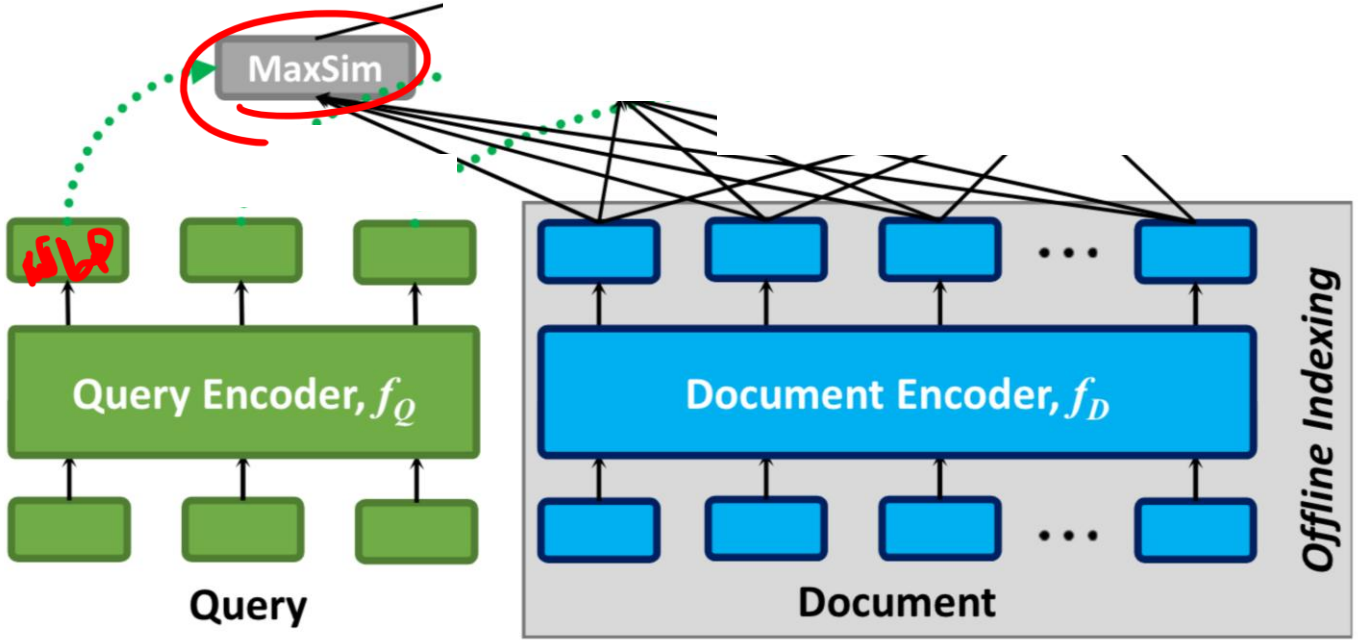
ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

Significantly more effective (but more costly) than single-vector retrieval

Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Figure from Khattab et al. (2020)

Token-level Dense Retrieval



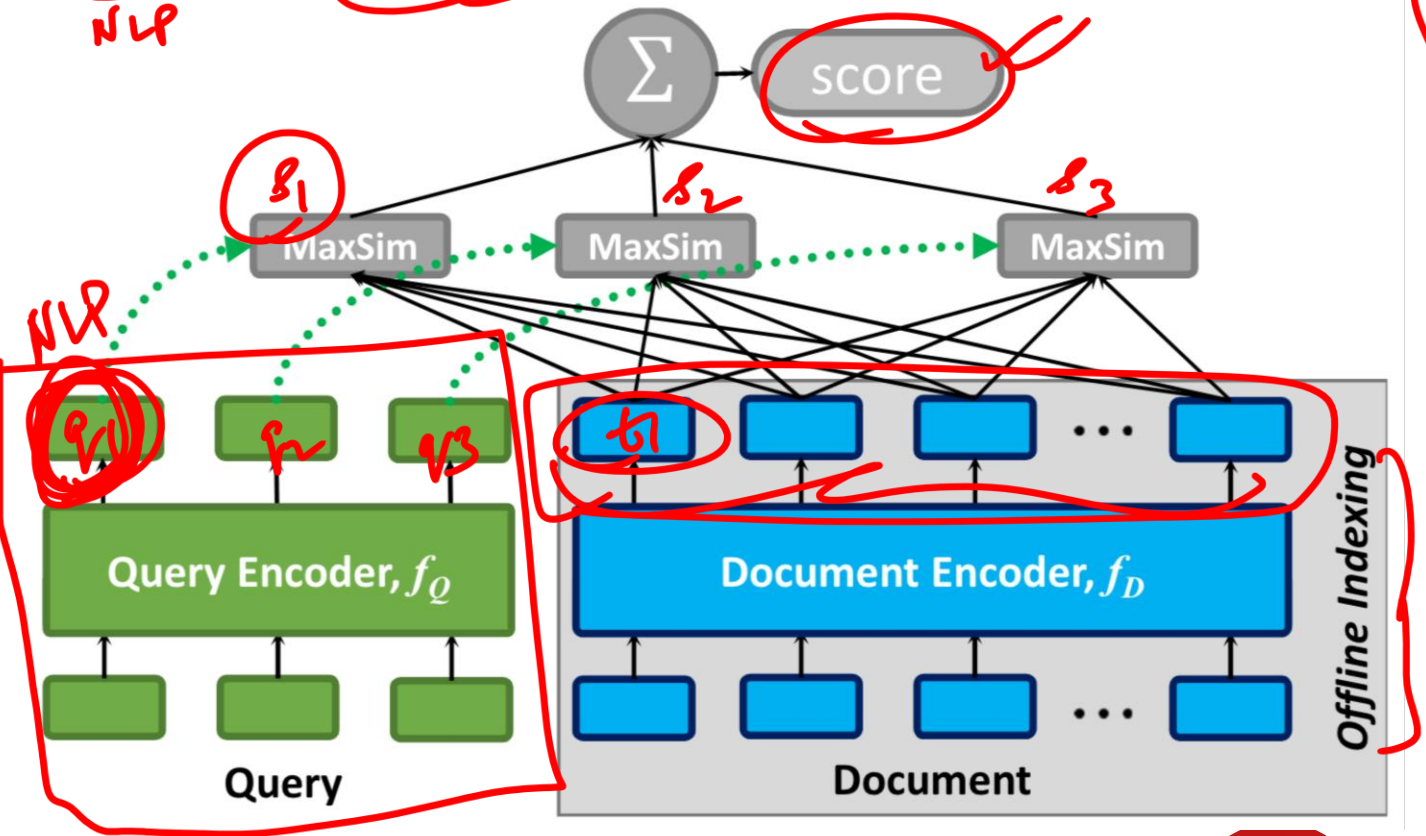
ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

Significantly more effective (but more costly) than single-vector retrieval

Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Figure from Khattab et al. (2020)

Token-level Dense Retrieval



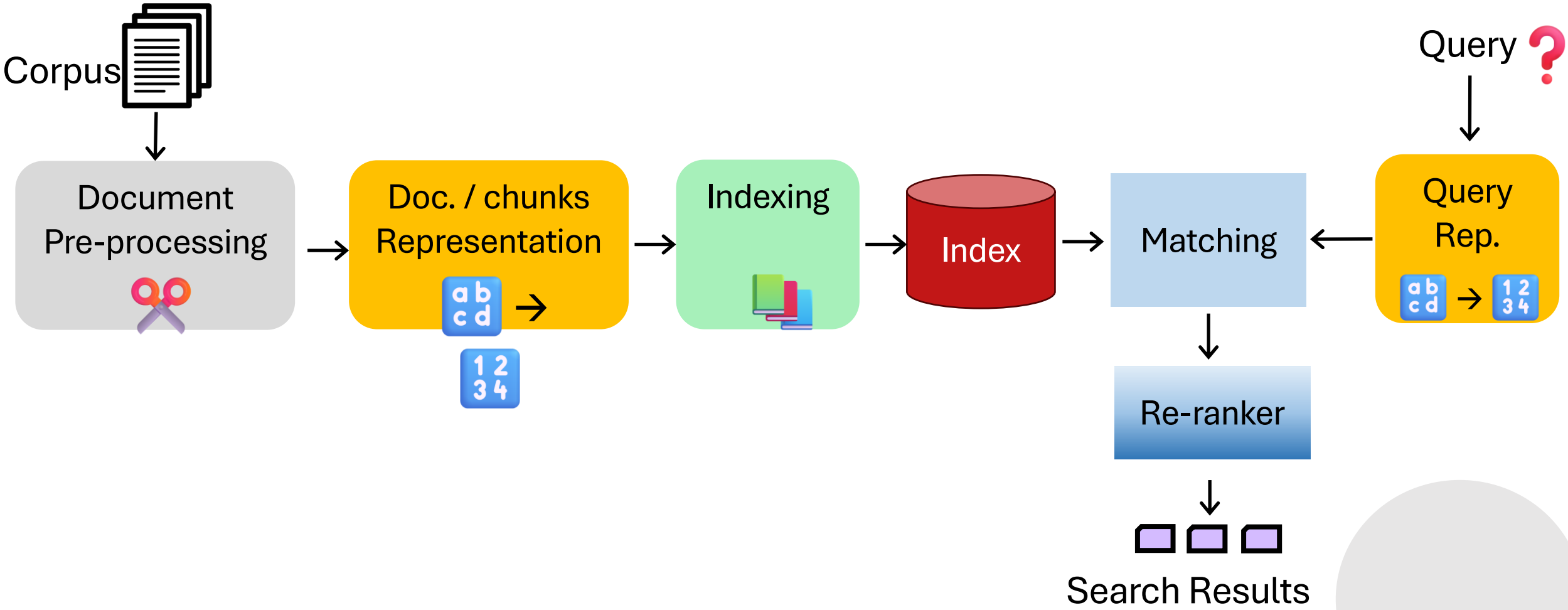
ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

Significantly more effective (but more costly) than single-vector retrieval

Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

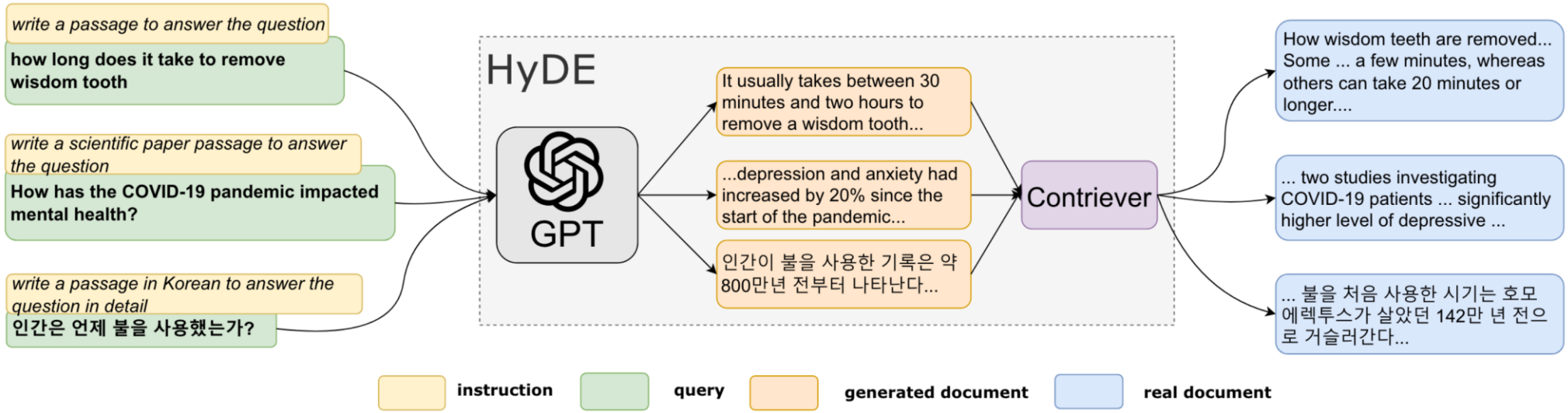
Figure from Khattab et al. (2020)

Retriever Pipeline – Dense Index w/ Reranker



Hypothetical Document Embeddings (Gao et al. 2023)

- Generate a “hypothetical document” for the query using an LLM, and try to look it up
- Can be easier than trying to match under-specified query



Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

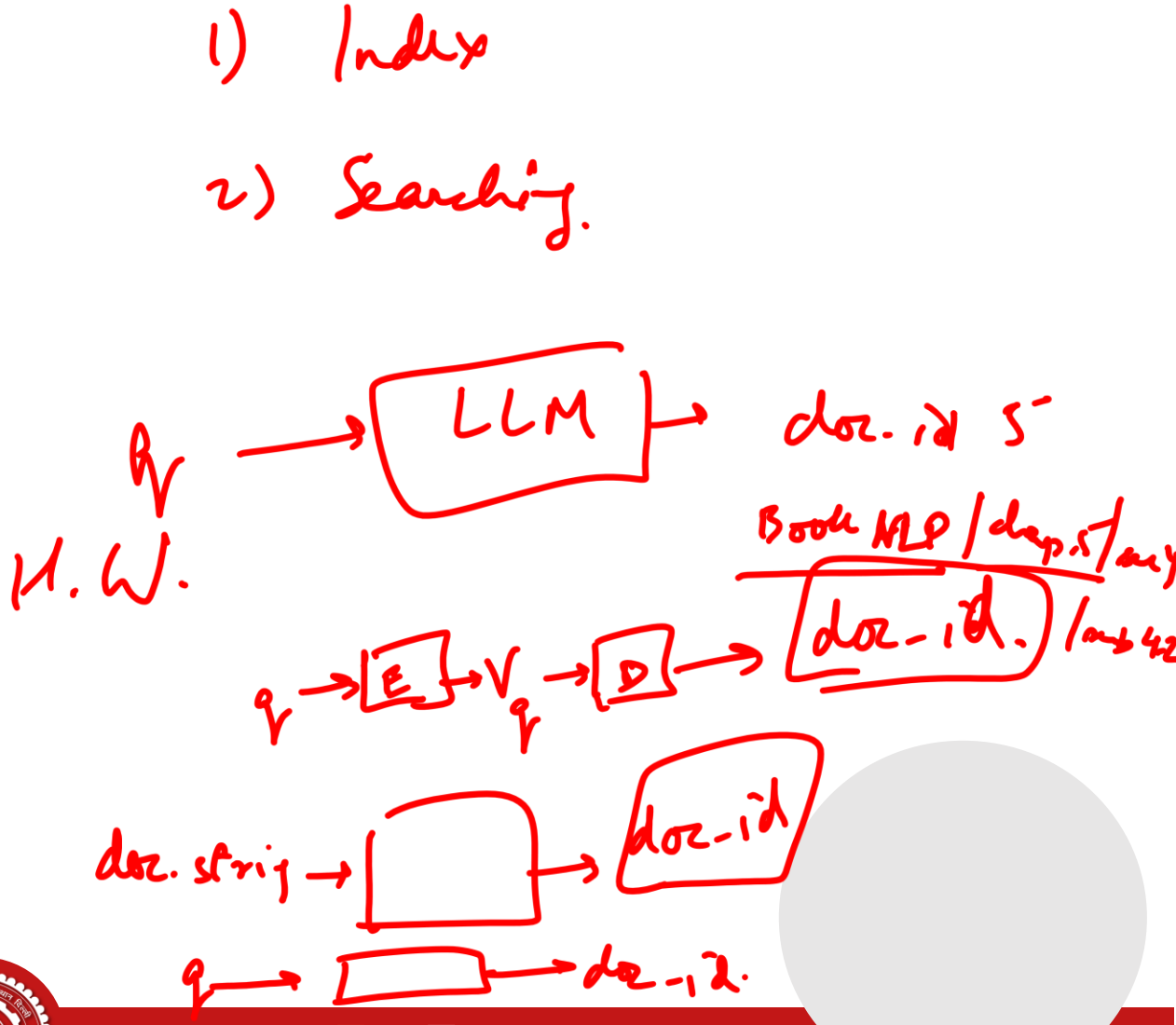
Retrieval Methods

- Sparse retrieval
- Document-level dense retrieval
- Token-level dense retrieval
- Cross-encoder reranking
- Differentiable search index (DSI)
- Table of Contents based search
- **Black-box retrieval** (just ask Google/Bing)

Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Retrieval Methods

- Sparse retrieval
- Document-level dense retrieval
- Token-level dense retrieval
- Cross-encoder reranking
- Differentiable search index (DSI)
- Table of Contents based search
- Black-box retrieval (just ask Google/Bing)

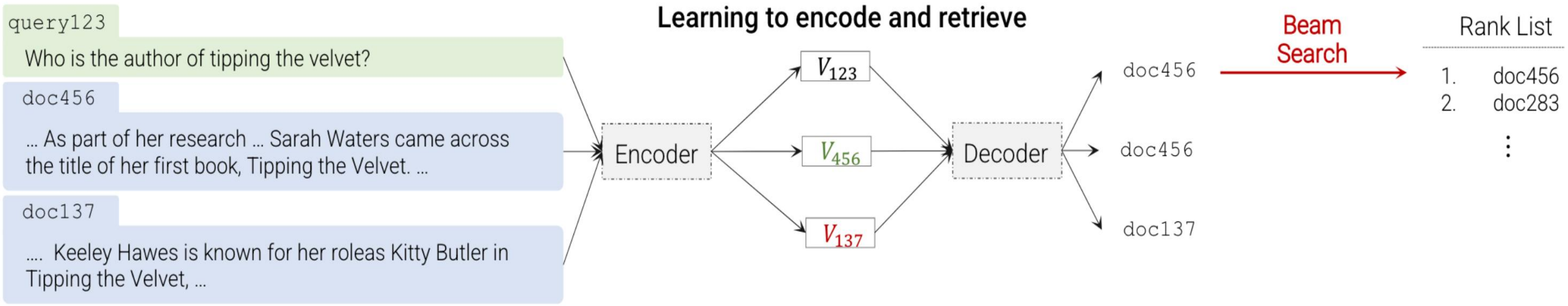


Slide source: https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

Differentiable Search Index

- LLMs are powerful enough to memorize the entire corpus.
- Can we use them directly as retriever?

Differentiable Search Index



Indexing task: Map each document to its doc id
Retrieval task: Map each query to the corresponding doc id

Differentiable Search Index

- Fully parameterize multi-stage retrieve then rank pipeline with a single neural model
- Train a seq2seq LLM for:
  - **Indexing task:** Map each document to its doc id
  - **Retrieval task:** Map each query to the corresponding doc id

- How to represent a document?
  - How to represent doc id?

Document Representation

- **Direct Indexing:** first L tokens of the document
- **Set indexing:** represent as set of words after removing stopwords.
- **Inverted Index:** Random contiguous chunks

Document Representation

- **Direct Indexing:** first L tokens of the document
- **Set indexing:** represent as set of words after removing stopwords.
- **Inverted Index:** Random contiguous chunks

Representation of doc ids

- **Unstructured Atomic Identifiers**
  - Use a new token to represent id of a document
  - Take softmax over the doc_id tokens
- **Naively Structured String Identifiers**
  - Decode the string representation of the doc_id
- **Semantically Structured Identifiers**
  - Create a hierarchical Tree structure over document embeddings.

Representation of doc ids

- **Unstructured Atomic Identifiers**
  - Use a new token to represent id of a document
  - Take softmax over the doc_id tokens
- **Naively Structured String Identifiers**
  - Decode the string representation of the doc_id
- **Semantically Structured Identifiers**
  - Create a hierarchical Tree structure over document embeddings.

Semantically Structured Ids

1. Capture some information about the semantics of the doc.
2. Structured in a way that the search space is effectively reduced after each decoding step

Representation of doc ids

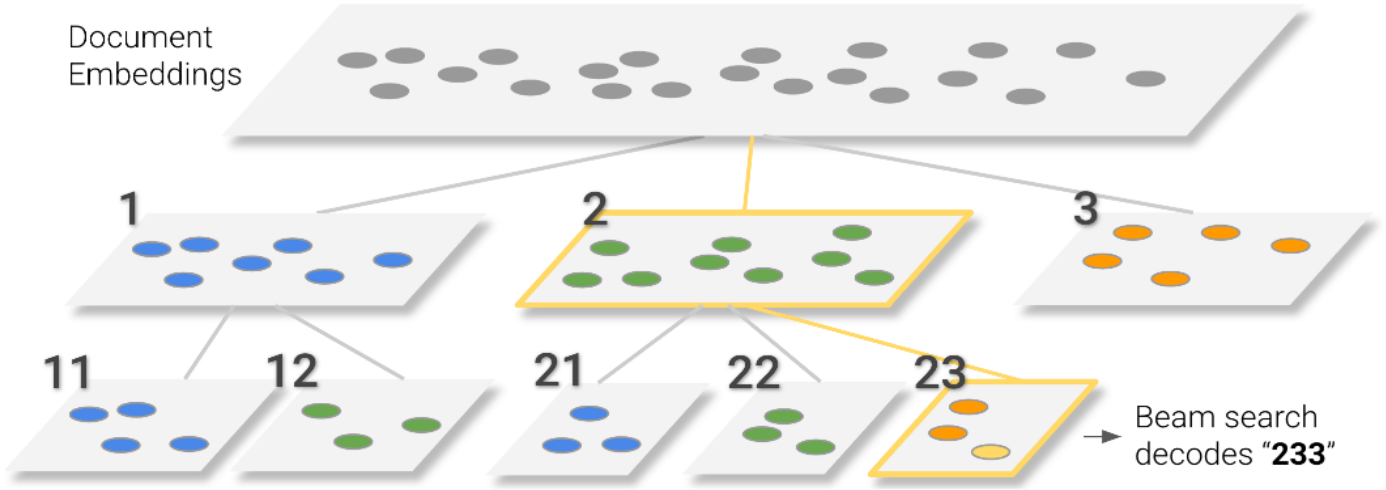


Figure 2: Visual example of a hierarchical clustering process used to assign semantically structured identifiers. During inference, beam search navigates this trie to decode the correct docid.

Results

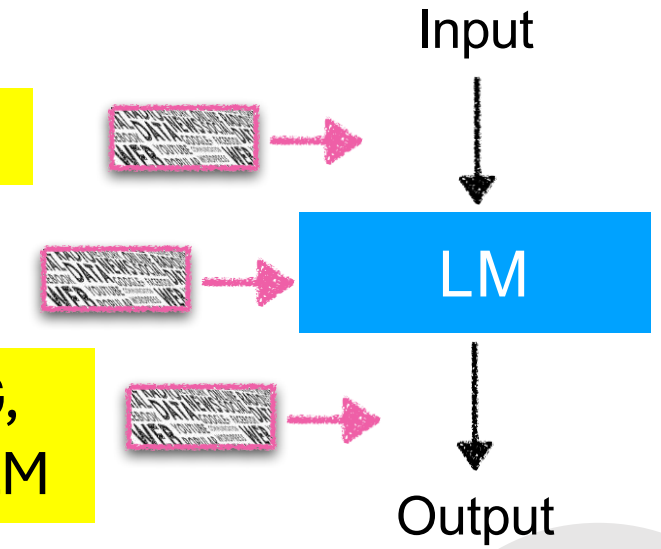
Model	Size	Params	Method	NQ10K		NQ100K		NQ320K	
				Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
BM25	-	-	-	12.4	33.5	20.9	46.4	11.6	34.4
T5	Base	220M	Dual Encoder	16.2	48.6	18.7	55.2	20.5	58.3
T5	Large	800M	Dual Encoder	18.8	55.7	22.3	60.5	22.4	63.3
T5	XL	3B	Dual Encoder	20.8	59.6	23.3	63.2	23.9	65.8
T5	XXL	11B	Dual Encoder	22.1	61.6	24.1	64.5	24.3	67.3
DSI	Base	250M	Atomic Docid	13.0	38.4	23.8	58.6	20.7	40.9
DSI	Large	800M	Atomic Docid	31.3	59.4	17.1	52.3	11.6	37.6
DSI	XL	3B	Atomic Docid	40.1	76.9	19.0	55.3	28.1	61.9
DSI	XXL	11B	Atomic Docid	39.4	77.0	25.3	**67.9**	24.0	55.1
DSI	Base	250M	Naive String Docid	28.1	48.0	18.7	44.6	6.7	21.0
DSI	Large	800M	Naive String Docid	34.7	60.5	21.2	50.7	13.3	33.6
DSI	XL	3B	Naive String Docid	44.7	66.4	24.0	55.1	16.7	58.1
DSI	XXL	11B	Naive String Docid	46.7	**77.9**	**27.5**	62.4	23.8	55.9
DSI	Base	250M	Semantic String Docid	33.9	57.3	19.0	44.9	27.4	56.6
DSI	Large	800M	Semantic String Docid	37.5	65.1	20.4	50.2	35.6	62.6
DSI	XL	3B	Semantic String Docid	41.9	67.1	22.4	52.2	39.1	66.8
DSI	XXL	11B	Semantic String Docid	**48.5**	72.1	26.9	59.5	**40.4**	**70.3**

Outline

- Motivation
  - Drawbacks of Parametric LLMs – *hallucination, verification …*
  - Motivating Retrieval-based LLMs *– close book vs open book*
- Major components of Retrieval-based LLMs *– index, retrieve, read …*
- Retrieval Methods – *sparse, dense, reranking, black-box*
- kNN, RETRO, REALM, RAG – *seminal works*
- Overview of Training Techniques – *independent, sequential, joint training …*
- Limitations – *lost in the middle, still hallucinating, retriever failures …*

How to use the Book?

- Output interpolations - After solving the question yourself? kNN LMs
- Intermediate fusion – modify the LM architecture to be aware of the book? RETRO
- Input augmentation (RAG) - Before you start solving? RAG, REALM



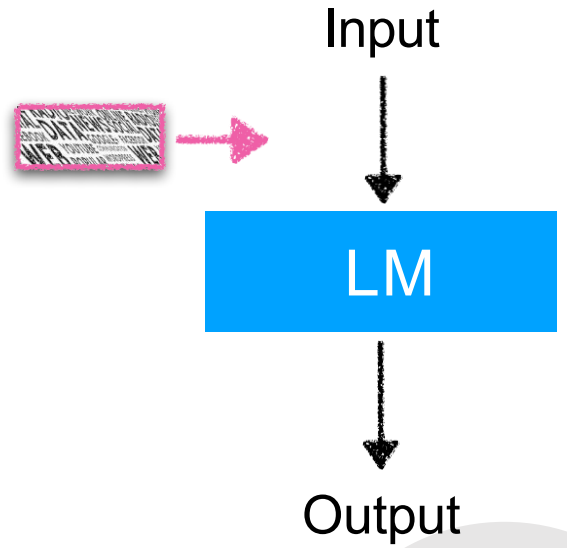
Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

How to use the Book?

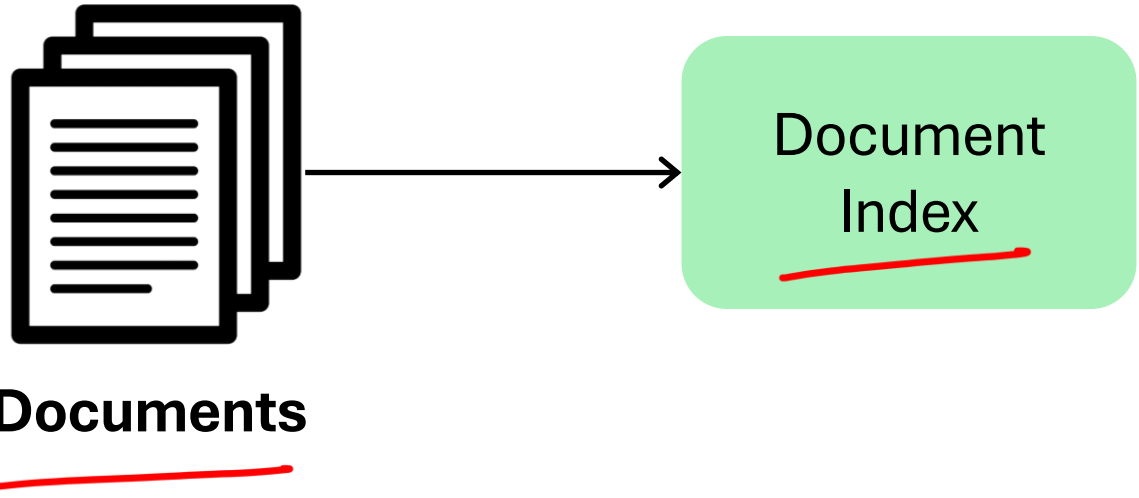
- Output interpolations - After solving the question yourself?

- Intermediate fusion – modify the LM architecture to be aware of the book?

- Input augmentation (RAG) - Before you start solving?

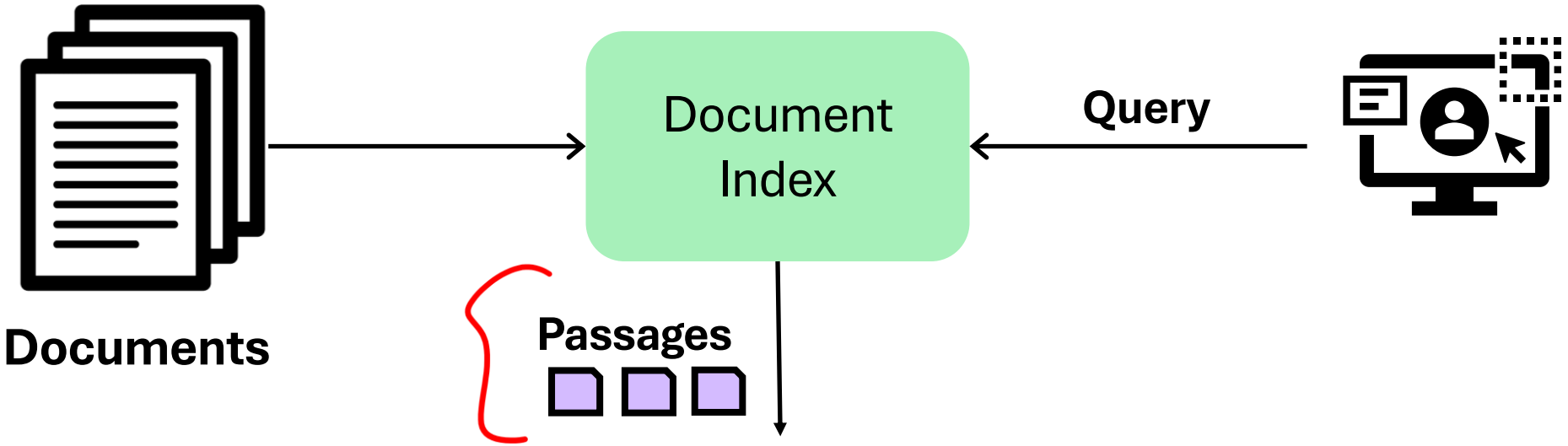


RAG - Architecture

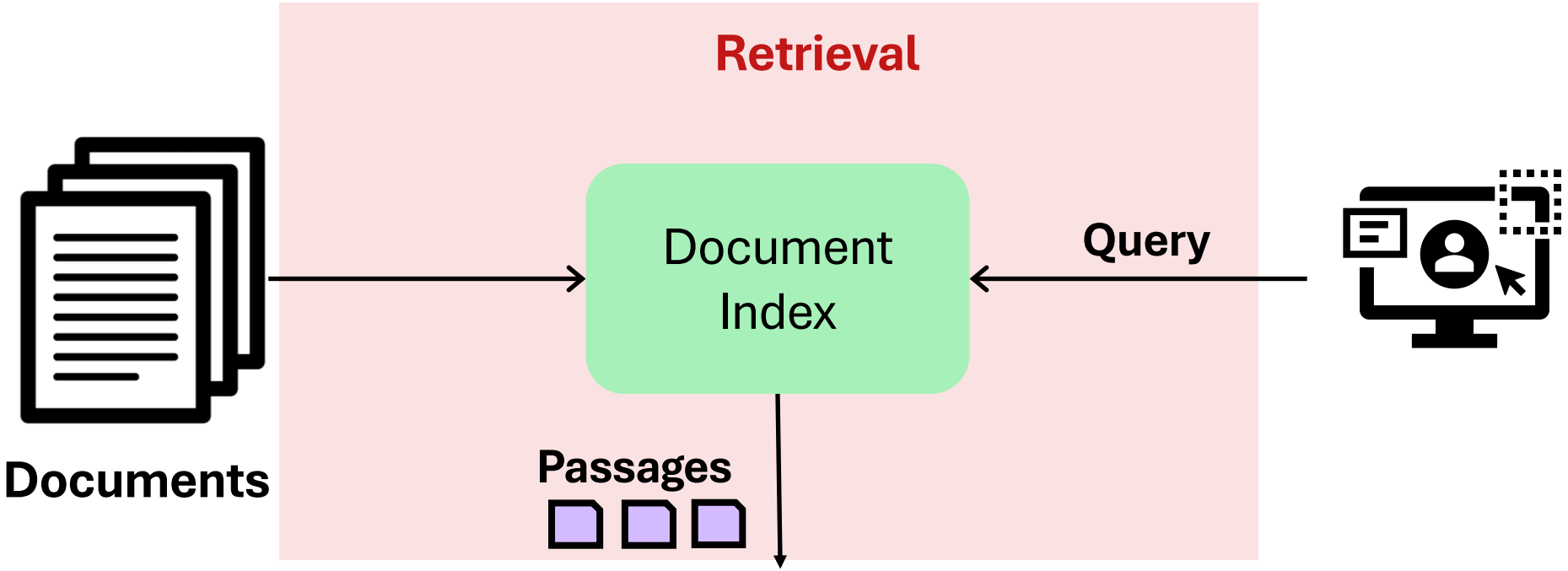


RAG - Architecture

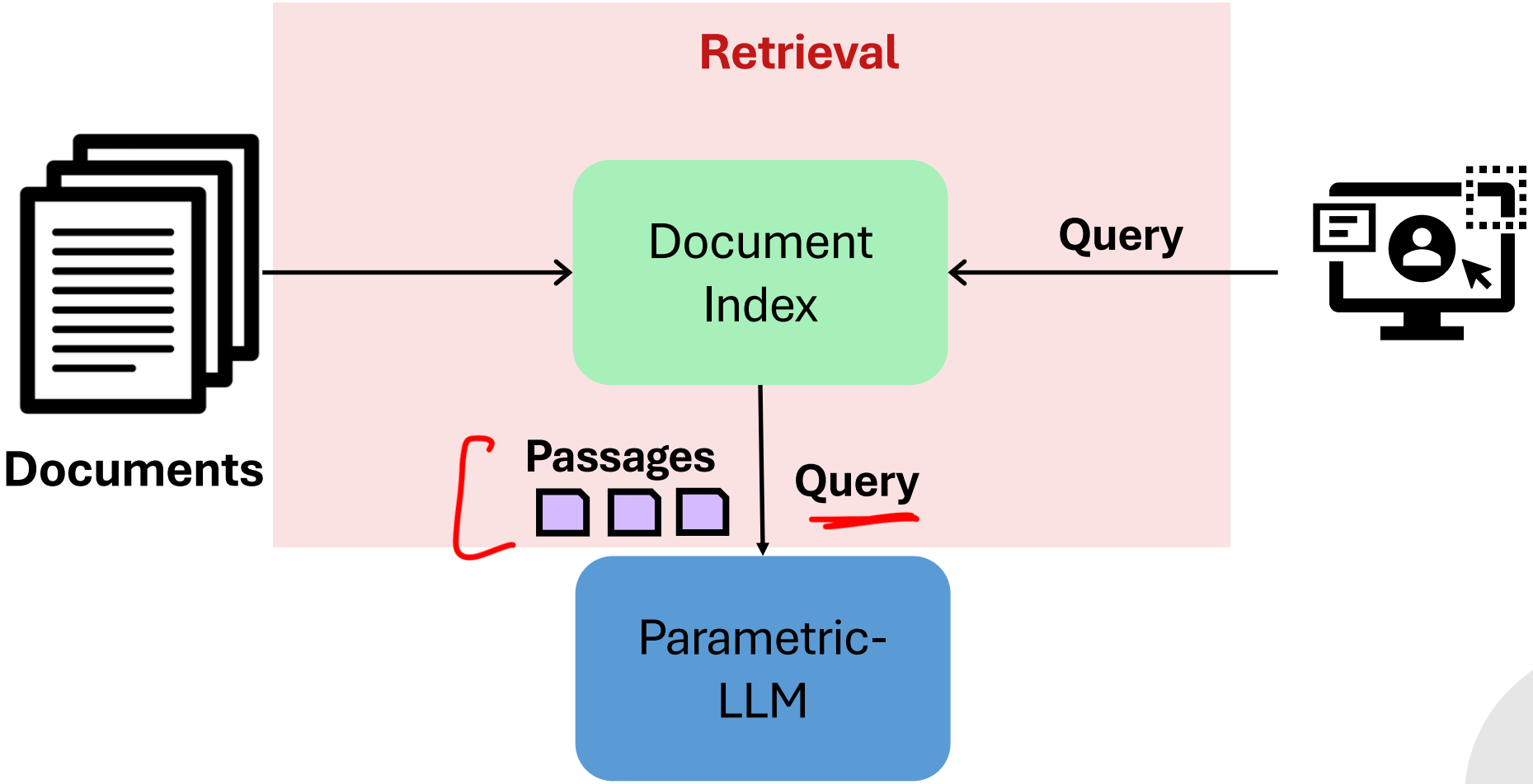
RAG - Architecture



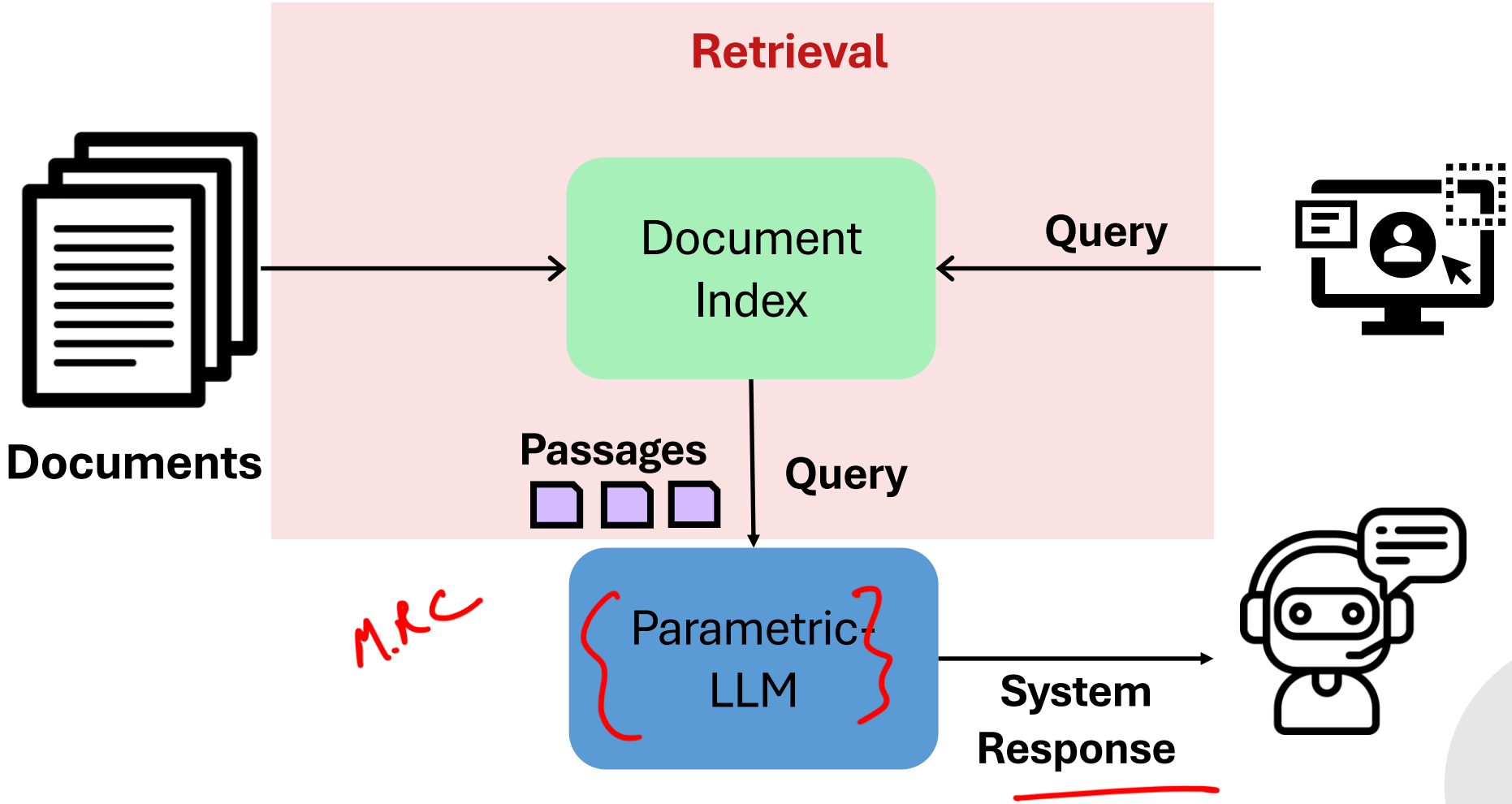
RAG - Architecture



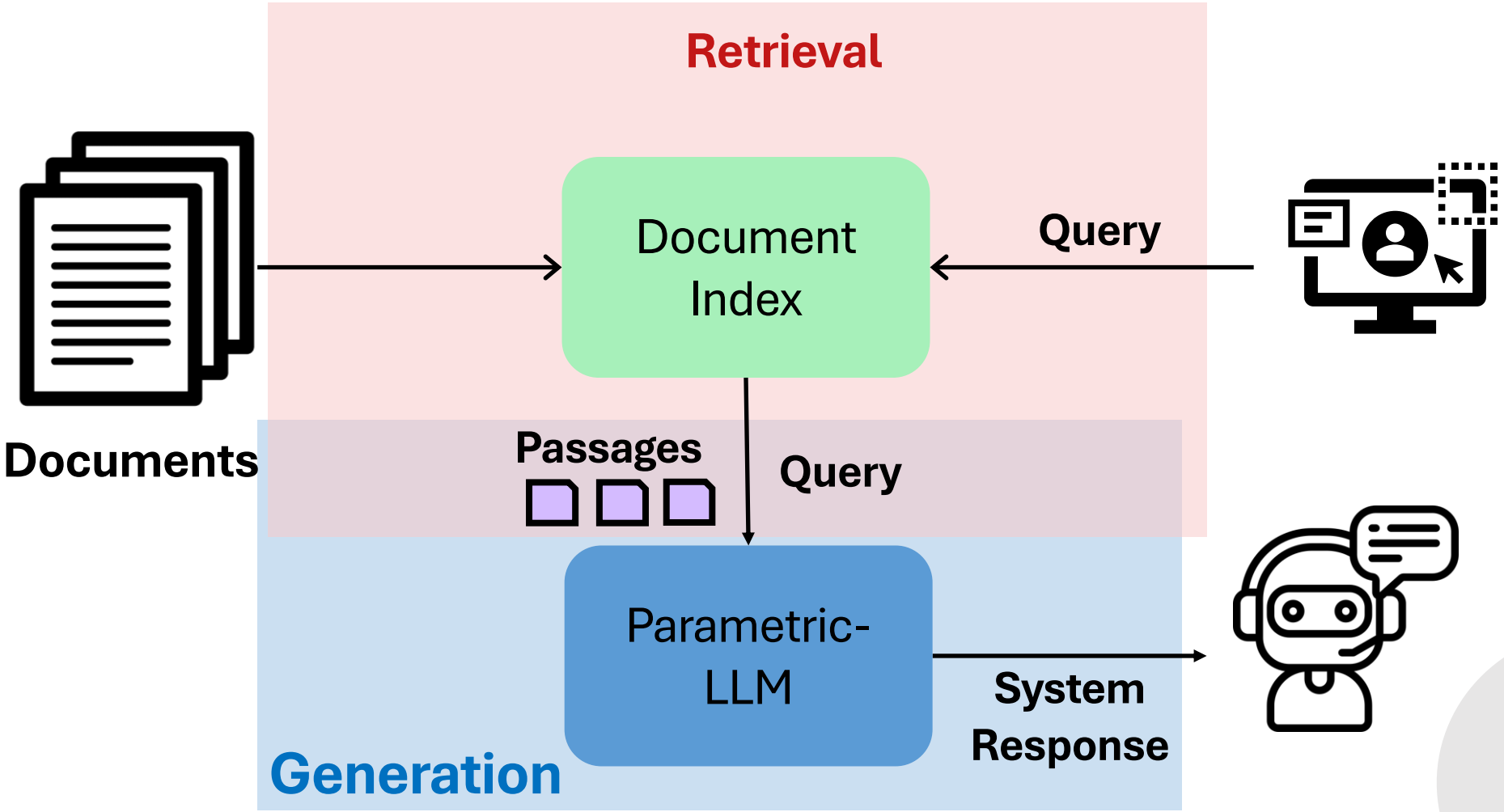
RAG - Architecture



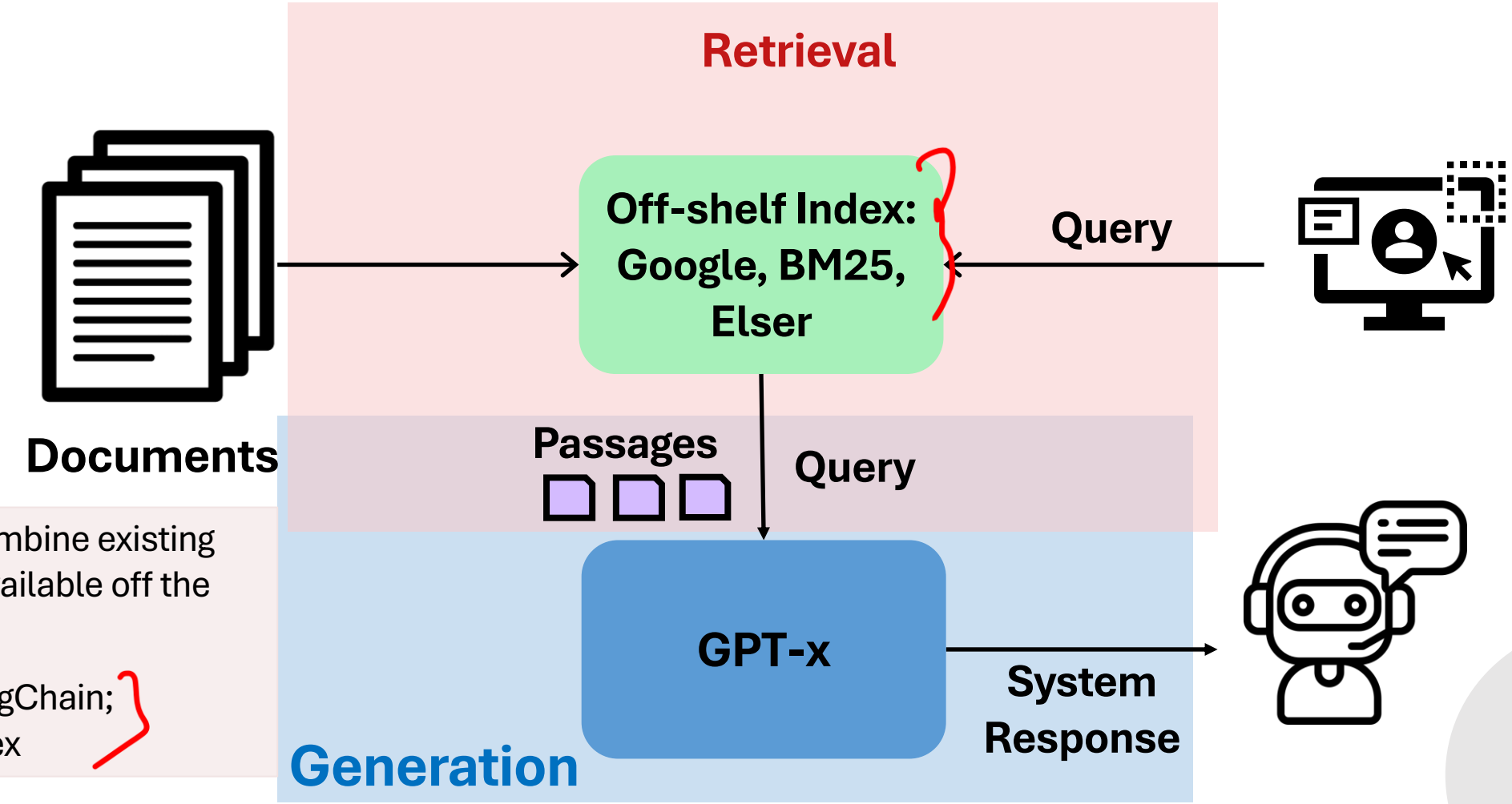
RAG - Architecture



Retrieval Based LLMs - Architecture



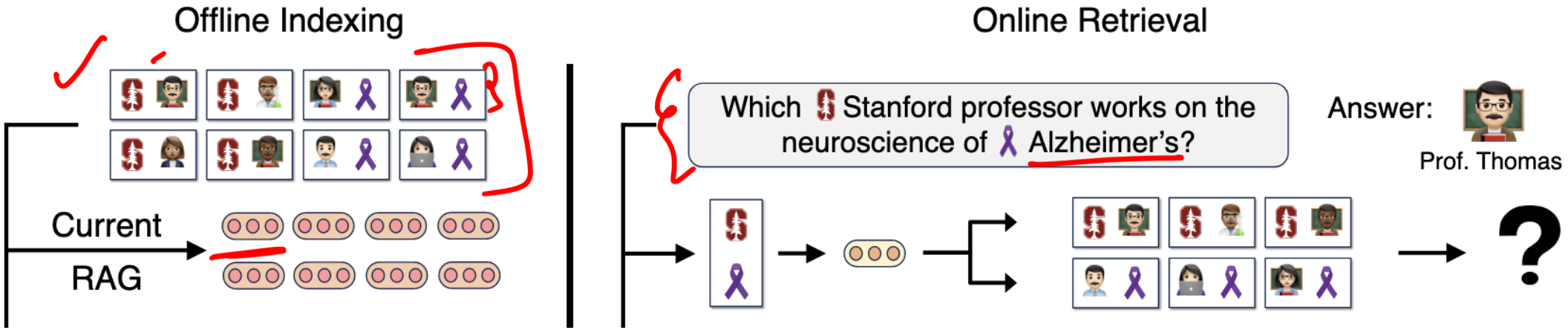
RAG - Architecture



- ✓ Simply combine existing models available off the shelf!
- ✓ Tools: LangChain; LlamaIndex

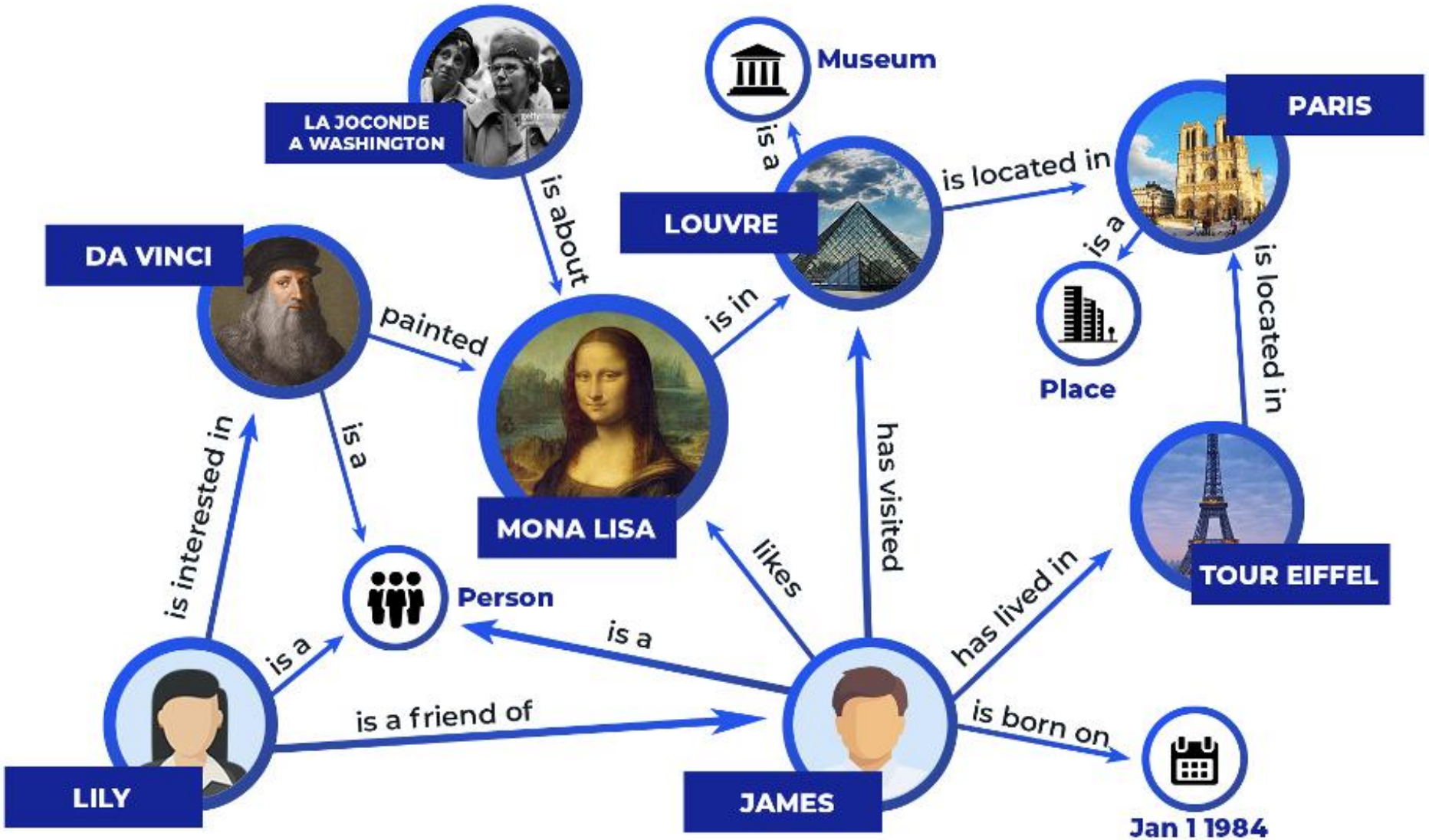
Issues with RAG 1.0

- May fail for complex multi-hop queries

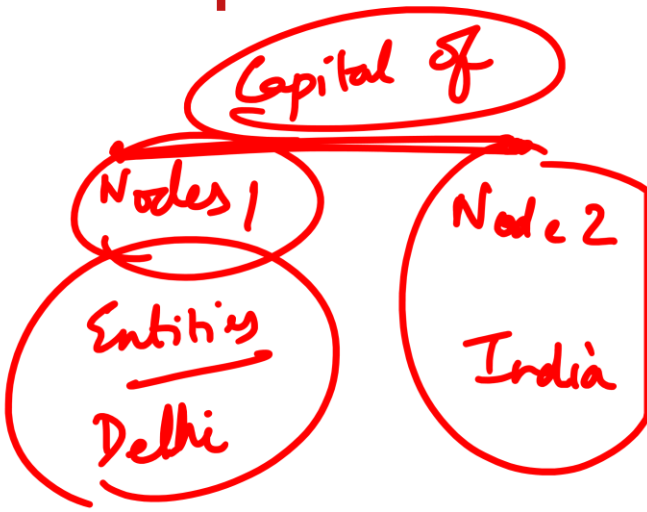


source: HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

Open IE

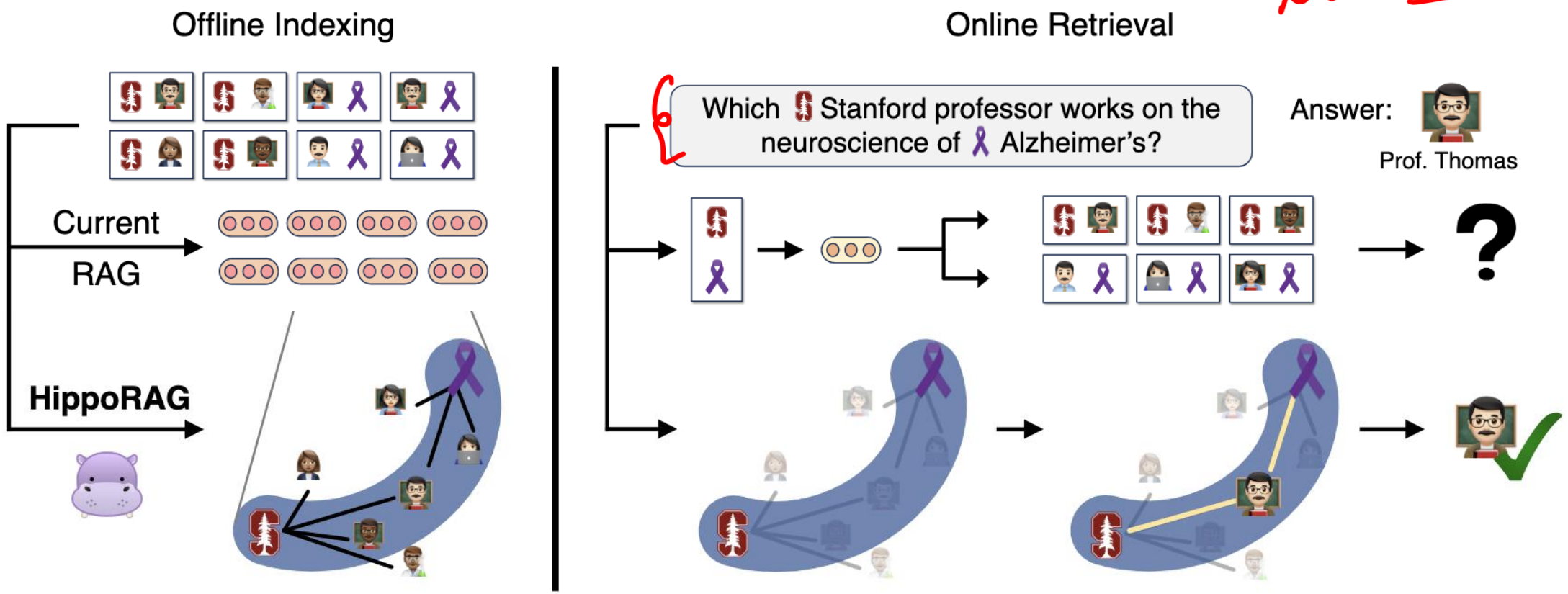


What is a Knowledge Graph?



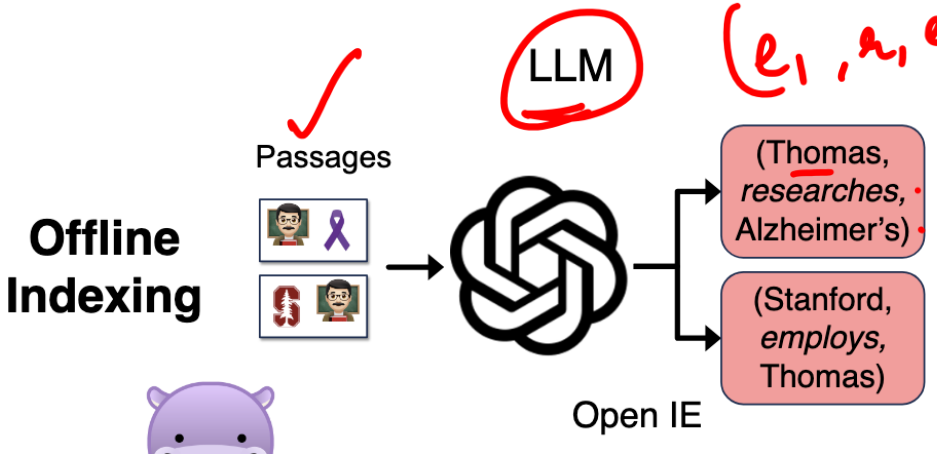
Graph RAG

multi-hop Q.



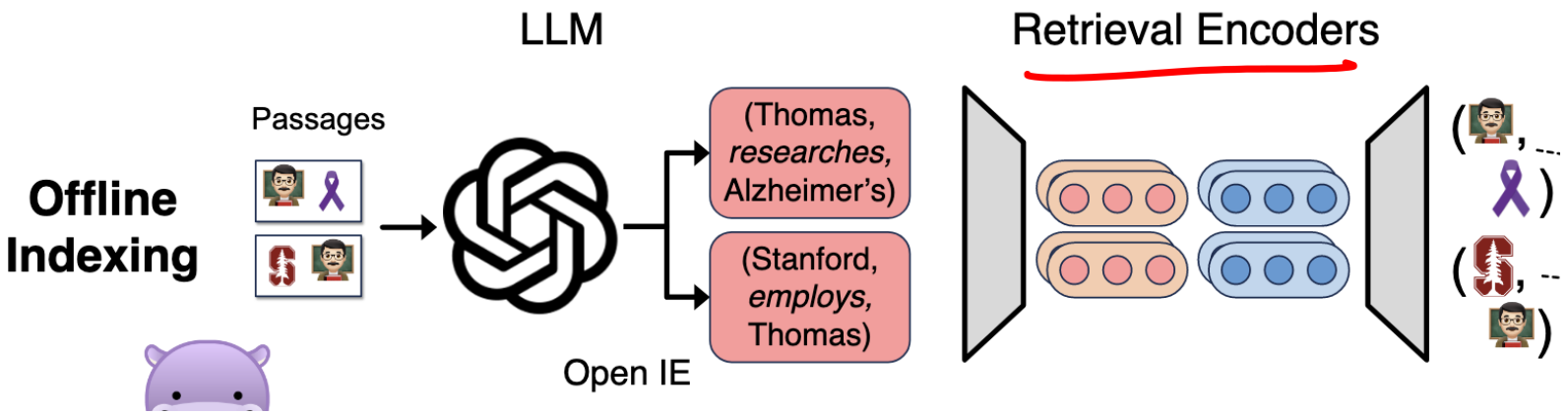
source: HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

Hippo RAG – Offline Indexing



1. OpenIE - Extract noun phrases & relations

Hippo RAG – Offline Indexing

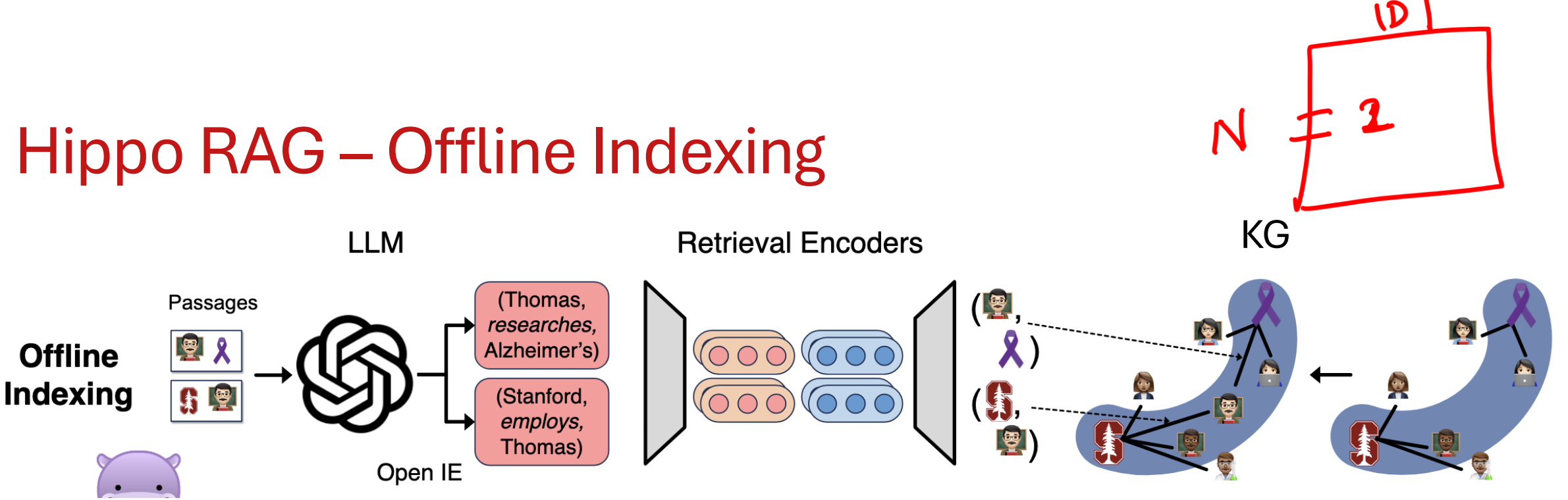


1. OpenIE - Extract noun phrases & relations

2. Add Synonymy relations: Encode noun phrases via encoder & join similar nodes

source: HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

Hippo RAG – Offline Indexing



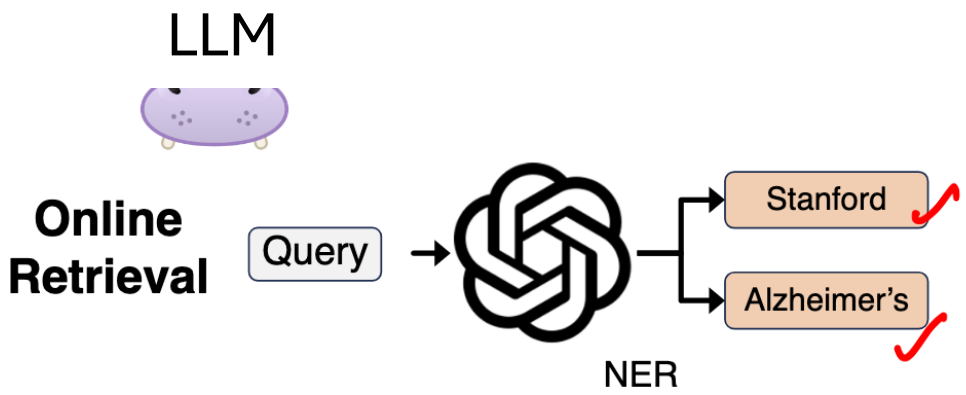
1. OpenIE - Extract noun phrases & relations

2. Add Synonymy relations: Encode noun phrases via encoder & join similar nodes

3. Store node embeddings
4 . Store KG
5. Store noun-phrase → Passage id mapping

source: HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

Hippo RAG – Query Time

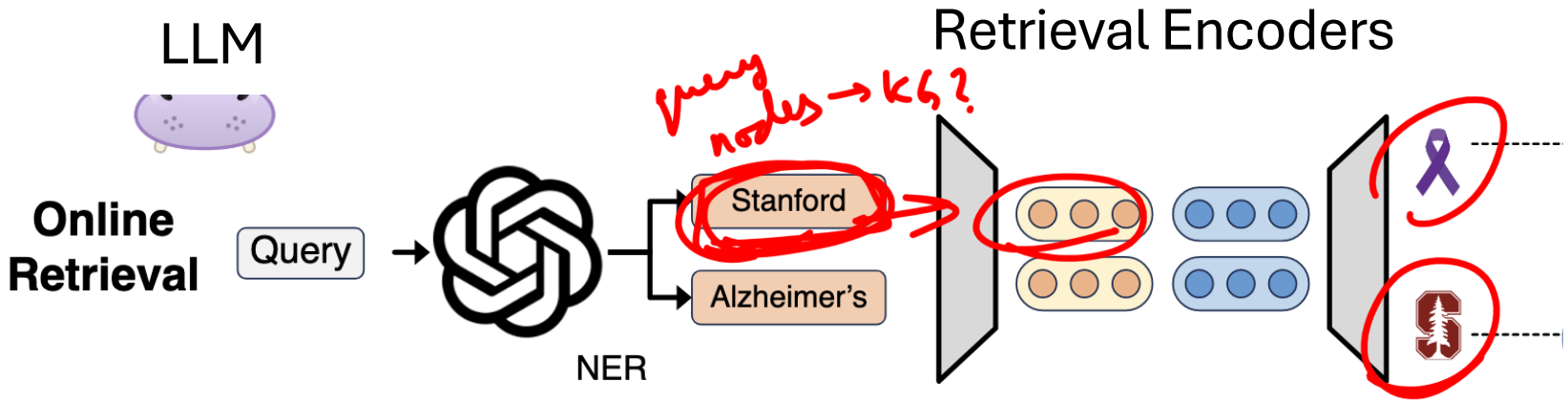


1. Extract named entities from query

source: HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

?? How to search for "Stanford" in the KG ?

Hippo RAG – Query Time

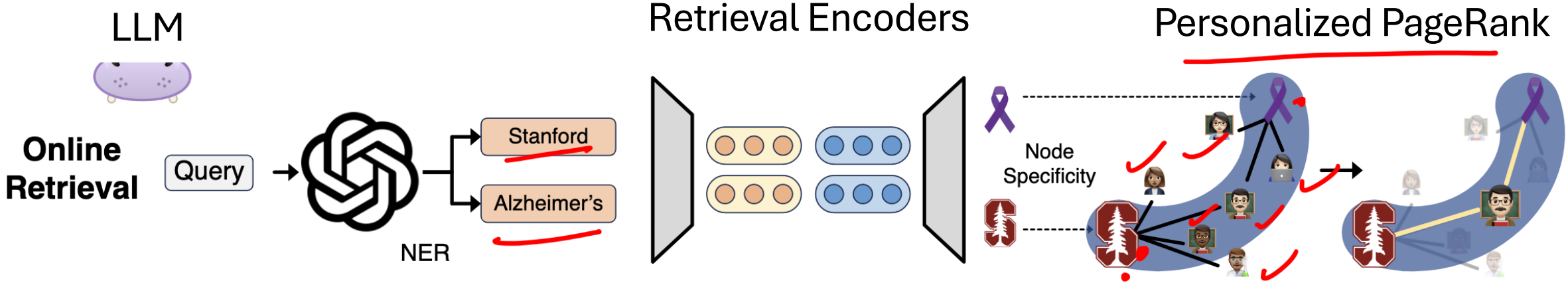


1. Extract named entities from query

2. Encode named entities
3. Search in the index (recall we indexed node embeddings)

source: HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

Hippo RAG – Query Time



1. Extract named entities from query

2. Encode named entities
3. Search in the index (recall we indexed node embeddings)

3. Subgraph sampling via PPR – extract nodes (noun phrases) in the nbhrhood

4. Extract the passages that mention the noun phrases

source: HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

Hippo RAG – Results

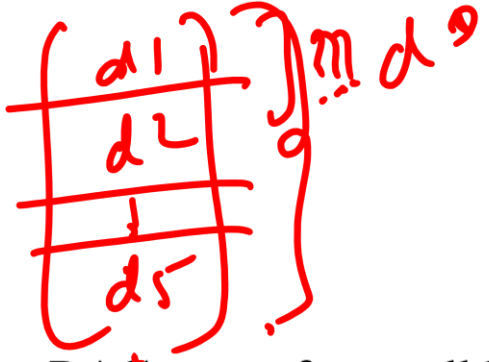


Table 2: **Single-step retrieval performance.** HippoRAG outperforms all baselines on MuSiQue and 2WikiMultiHopQA and achieves comparable performance on the less challenging HotpotQA dataset.

	MuSiQue	
	R@2	R@5
BM25 [69]	32.3	41.2
Contriever [35]	34.8	46.6
GTR [53]	37.4	49.1
ColBERTv2 [70]	37.9	49.2
RAPTOR [71]	35.7	45.3
RAPTOR (ColBERTv2)	36.9	46.5
Proposition [10]	37.6	49.3
Proposition (ColBERTv2)	37.8	50.1
HippoRAG (Contriever)	**41.0**	**52.1**
HippoRAG (ColBERTv2)	40.9	51.9

Hippo RAG – Results

Table 2: **Single-step retrieval performance.** HippoRAG outperforms all baselines on MuSiQue and 2WikiMultiHopQA and achieves comparable performance on the less challenging HotpotQA dataset.

	MuSiQue		2Wiki		HotpotQA		Average	
	R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
BM25 [69]	32.3	41.2	51.8	61.9	55.4	72.2	46.5	58.4
Contriever [35]	34.8	46.6	46.6	57.5	57.2	75.5	46.2	59.9
GTR [53]	37.4	49.1	60.2	67.9	59.4	73.3	52.3	63.4
ColBERTv2 [70]	37.9	49.2	59.2	68.2	**64.7**	**79.3**	53.9	65.6
RAPTOR [71]	35.7	45.3	46.3	53.8	58.1	71.2	46.7	56.8
RAPTOR (ColBERTv2)	36.9	46.5	57.3	64.7	63.1	75.6	52.4	62.3
Proposition [10]	37.6	49.3	56.4	63.1	58.7	71.1	50.9	61.2
Proposition (ColBERTv2)	37.8	50.1	55.9	64.9	63.9	78.1	52.5	64.4
HippoRAG (Contriever)	**41.0**	**52.1**	**71.5**	**89.5**	59.0	76.2	57.2	72.6
HippoRAG (ColBERTv2)	40.9	51.9	70.7	89.1	60.5	77.7	**57.4**	**72.9**

Further reading

- From Local to Global: A Graph RAG Approach to Query-Focused Summarization

From Local to Global: A Graph RAG Approach to Query-Focused Summarization

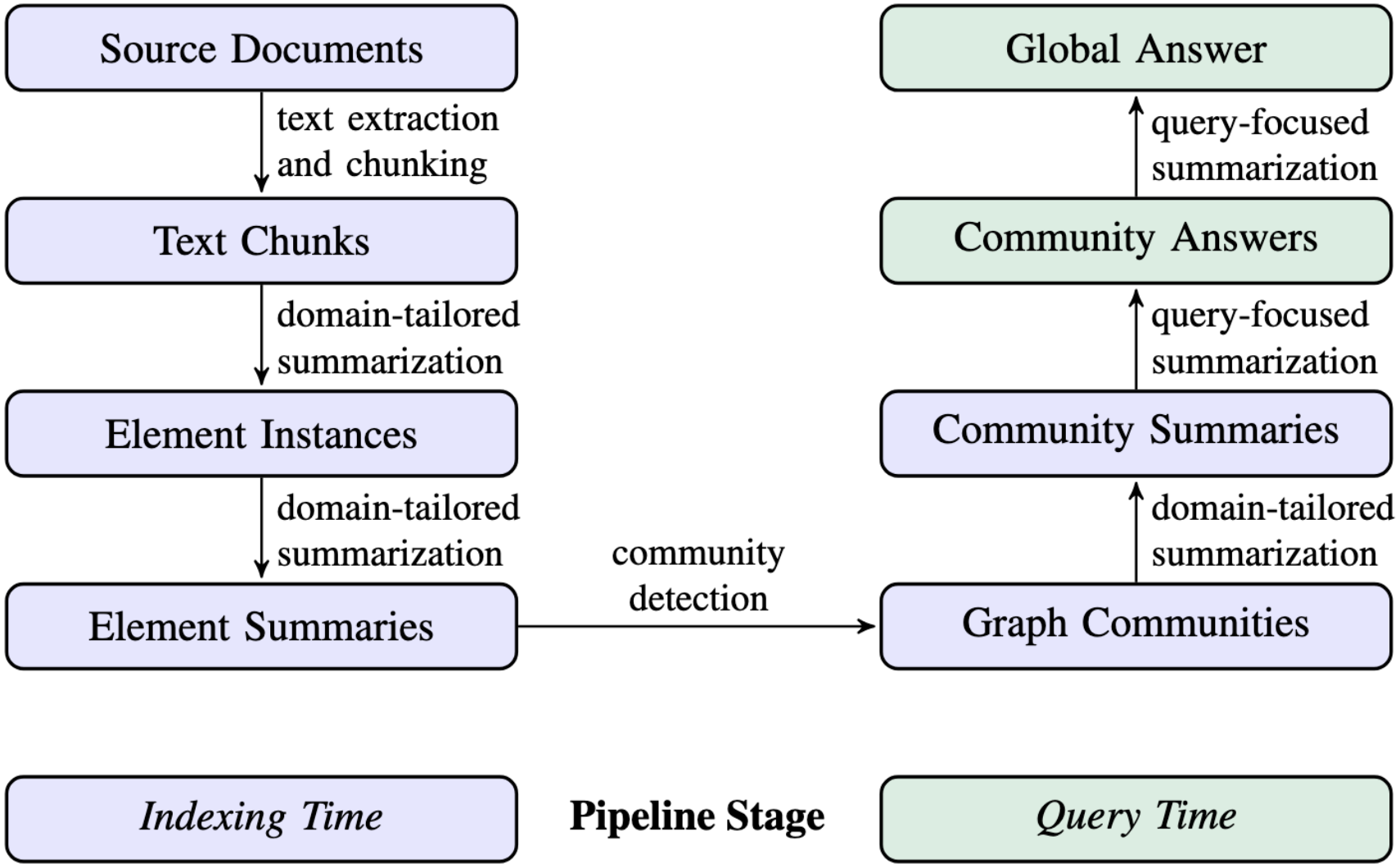
Darren Edge[1†] **Ha Trinh**[1†] **Newman Cheng**[2] **Joshua Bradley**[2] **Alex Chao**[3]

Apurva Mody[3] **Steven Truitt**[2]

Jonathan Larson[1]

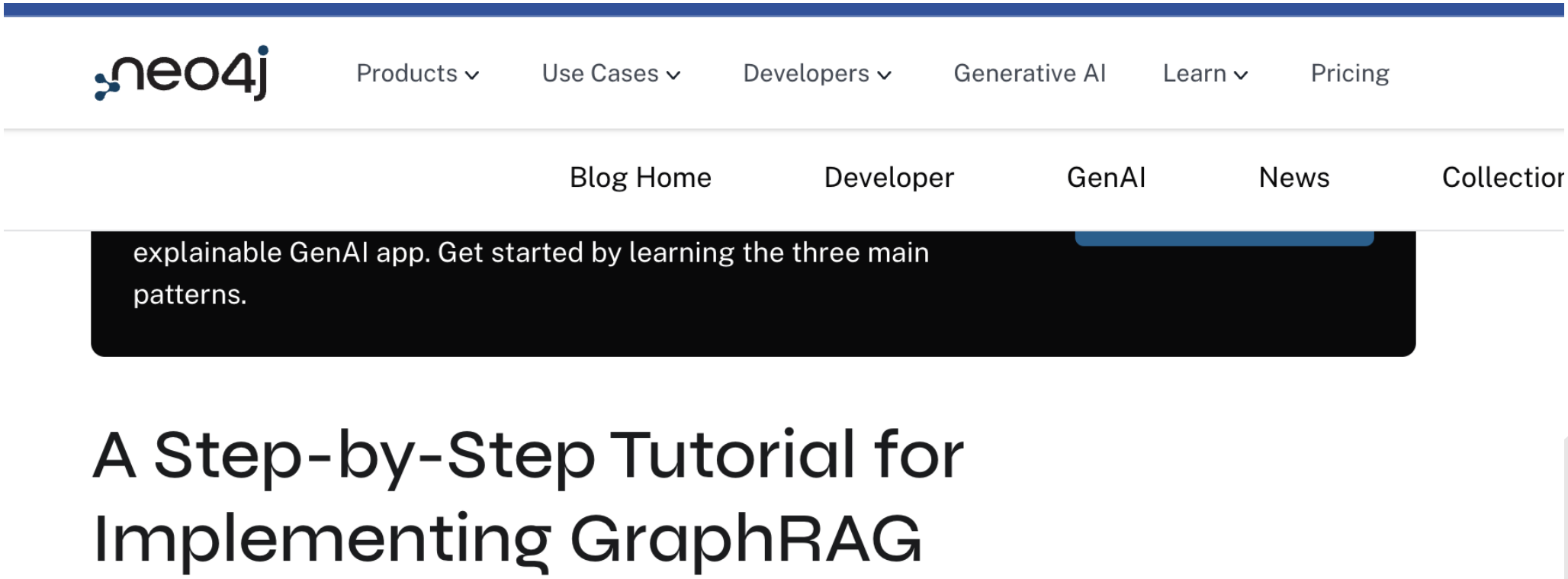
[1]Microsoft Research
[2]Microsoft Strategic Missions and Technologies
[3]Microsoft Office of the CTO

Graph RAG


Source Documents
text extraction and chunking
Text Chunks
domain-tailored summarization
Element Instances
domain-tailored summarization
Element Summaries
community detection
Graph Communities
domain-tailored summarization
Community Summaries
query-focused summarization
Community Answers
query-focused summarization
Global Answer
Indexing Time
Pipeline Stage
Query Time

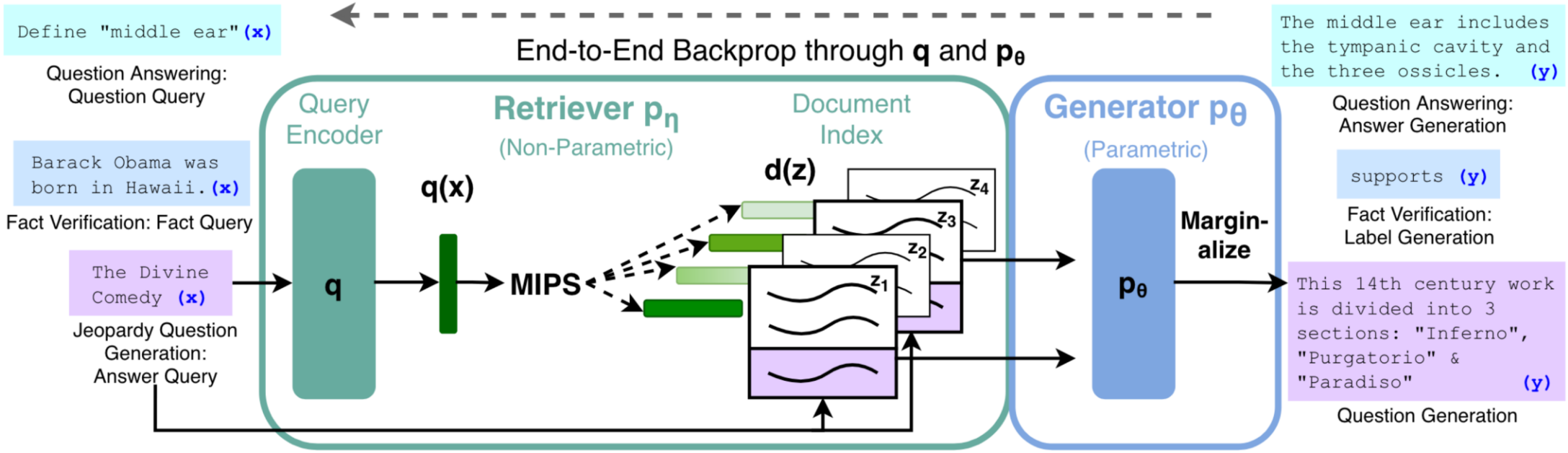
Resources for Graph RAG

- https://neo4j.com/blog/developer/rag-tutorial/



Joint Training of Retriever and Generator

RAG: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis et al. 2020)



RAG: Joint Training

Trainable components

- Retriever
  - Document Encoder
  - Query Encoder
- Reader: LM

RAG: Joint Training Equation (Lewis et al. 2020)

Retriever Reader

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z)$$

$$p_\eta(z|x) \propto \exp(\mathbf{d}(z)^\top \mathbf{q}(x)) \quad \mathbf{d}(z) = \text{enc}_d(z), \quad \mathbf{q}(x) = \text{enc}_q(x)$$

RAG: Joint Training Equation (Lewis et al. 2020)

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x)$$

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

- Given a retrieved document, generate the entire sequence y
- Marginalize over all the retrieved documents
- Can we generate one token given all documents, and then proceed to the next token?

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

Maximize $$\prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

Probability of decoding y_i given document z

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

Probability of decoding y_i given document z

Marginalize over all documents

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

Maximize $$\prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

Product over all N tokens

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

RAG-Token Model

Maximize $$\prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

RAG: Training Approximations

- Freeze top-k documents
- Freeze index (document embeddings), but search top-k documents
- Update index every T steps

Cold Start Problem

- **Reader:** Next token prediction pretraining
- **Inverse Cloze Task:** used to pretrain retriever embeddings

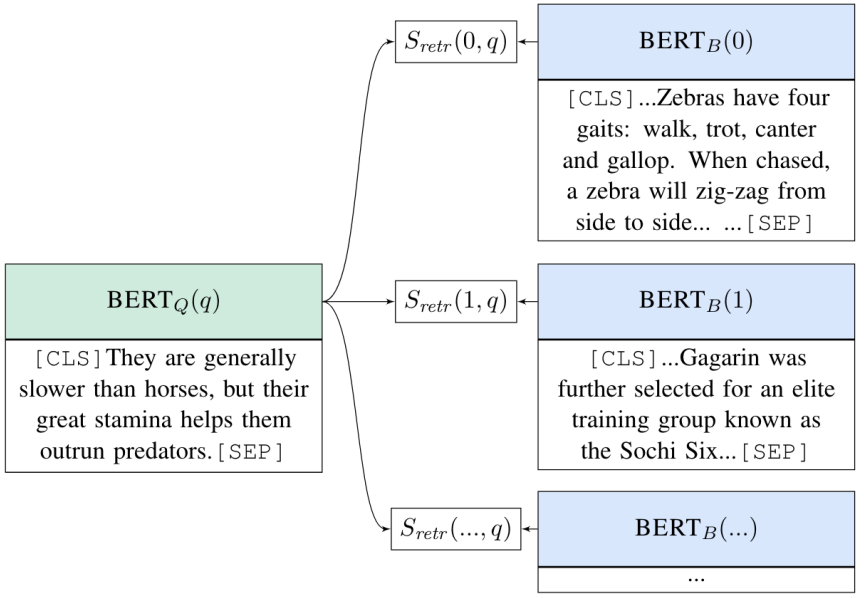


Figure 2: Example of the Inverse Cloze Task (ICT), used for retrieval pre-training. A random sentence (pseudo-query) and its context (pseudo evidence text) are derived from the text snippet: *"...Zebras have four gaits: walk, trot, canter and gallop.* ***They are generally slower than horses, but their great stamina helps them outrun predators****. When chased, a zebra will zig-zag from side to side..."* The objective is to select the true context among candidates in the batch.

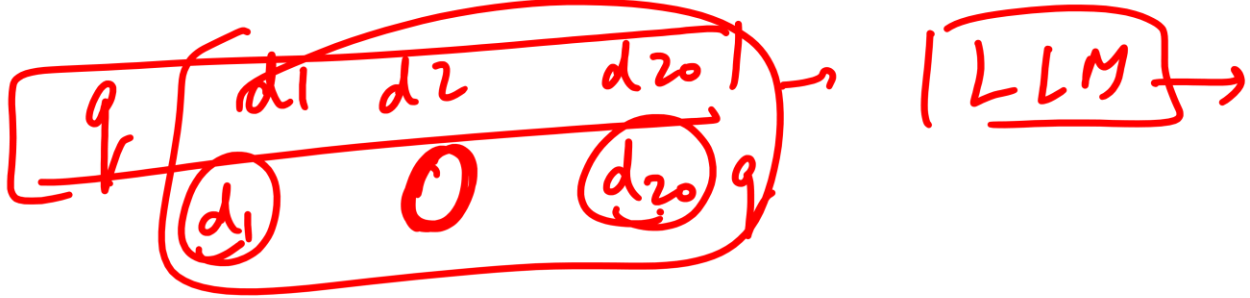
Image source: Lee et al 2019, Latent Retrieval for Weakly Supervised Open Domain Question Answering

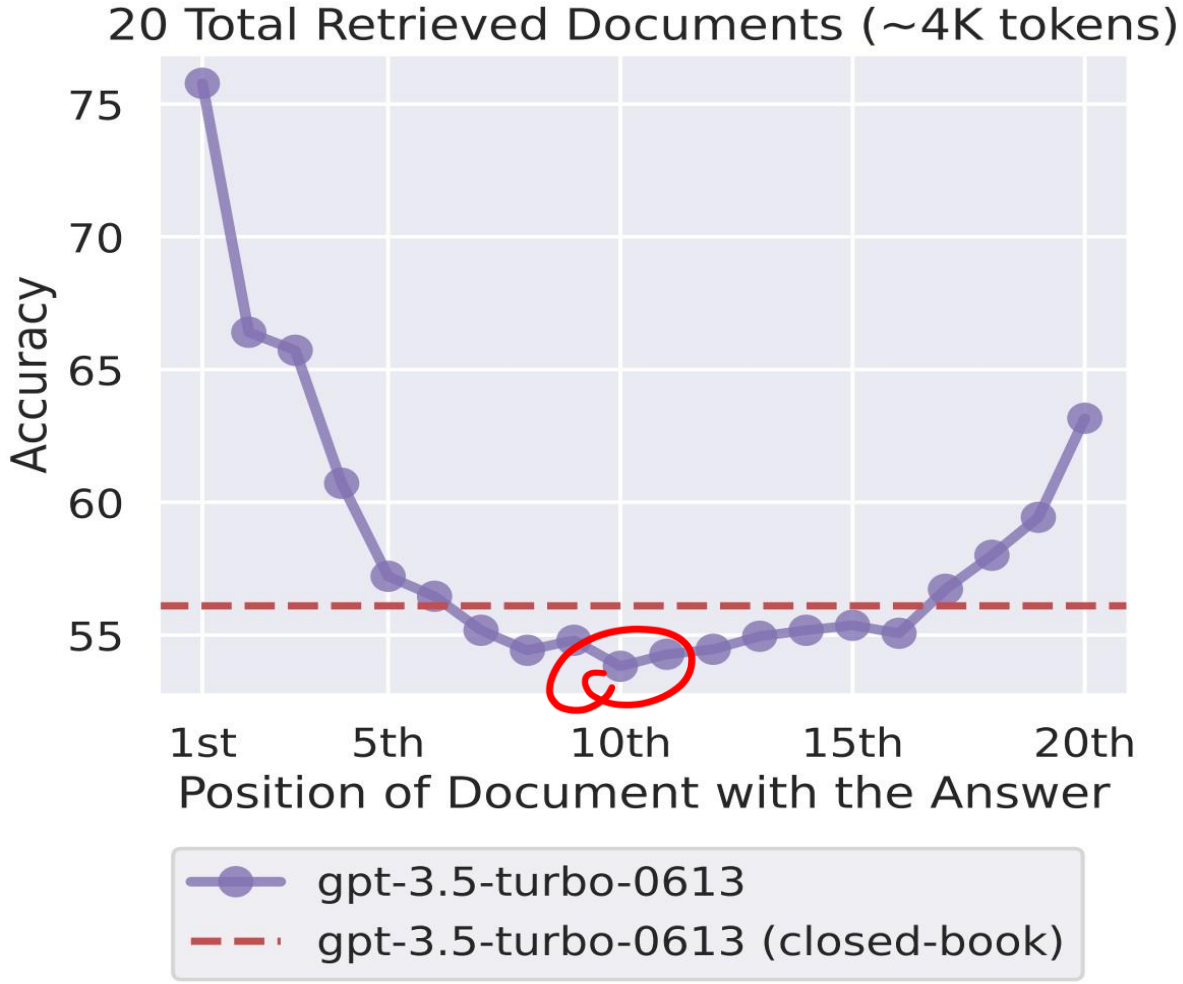
RAG: Results

Table 1: Open-Domain QA Test Scores. For TQA, left column uses the standard test set for Open-Domain QA, right column uses the TQA-Wiki test set. See Appendix D for further details.

	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**

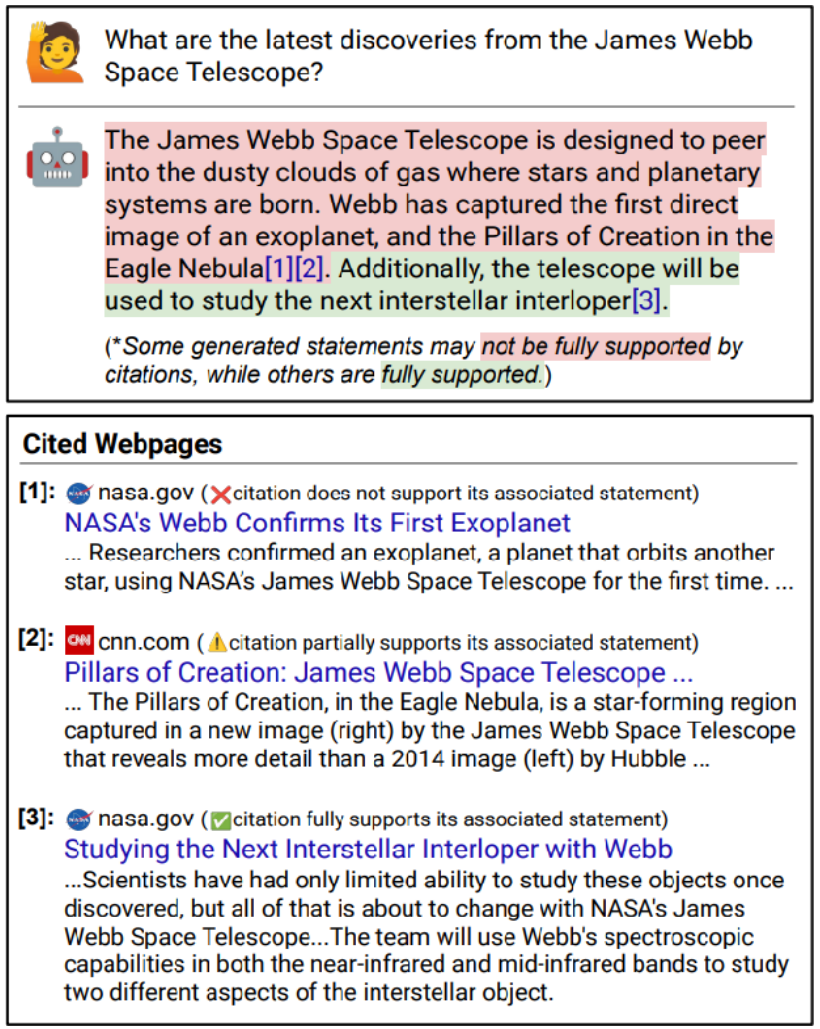
Lost in the Middle!





- As Context Increases, Models Miss Relevant Info
- "lost-in-the- middle" (Liu et al. 2023) demonstrates that models pay less attention to things in the middle of context windows

Retrieval-augmented LMs can still hallucinate



Liu et al. Evaluating Verifiability in Generative Search Engines. Findings of EMNLP 2023.

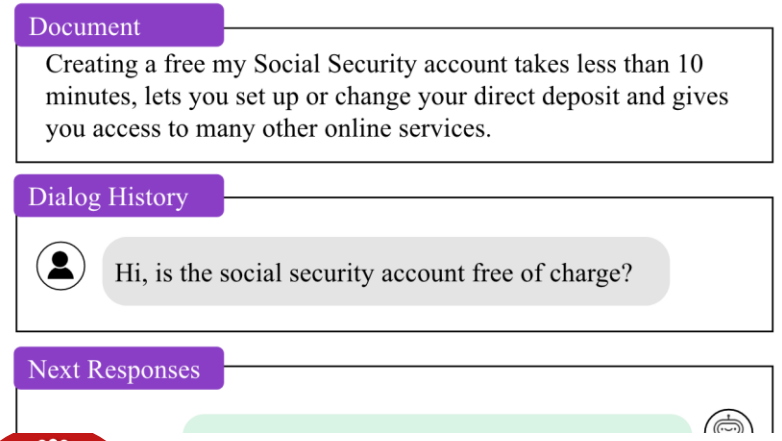
Quantifying Hallucination

Pointwise Mutual Information Based Metric and Decoding Strategy for Faithful Generation in Document Grounded Dialogs

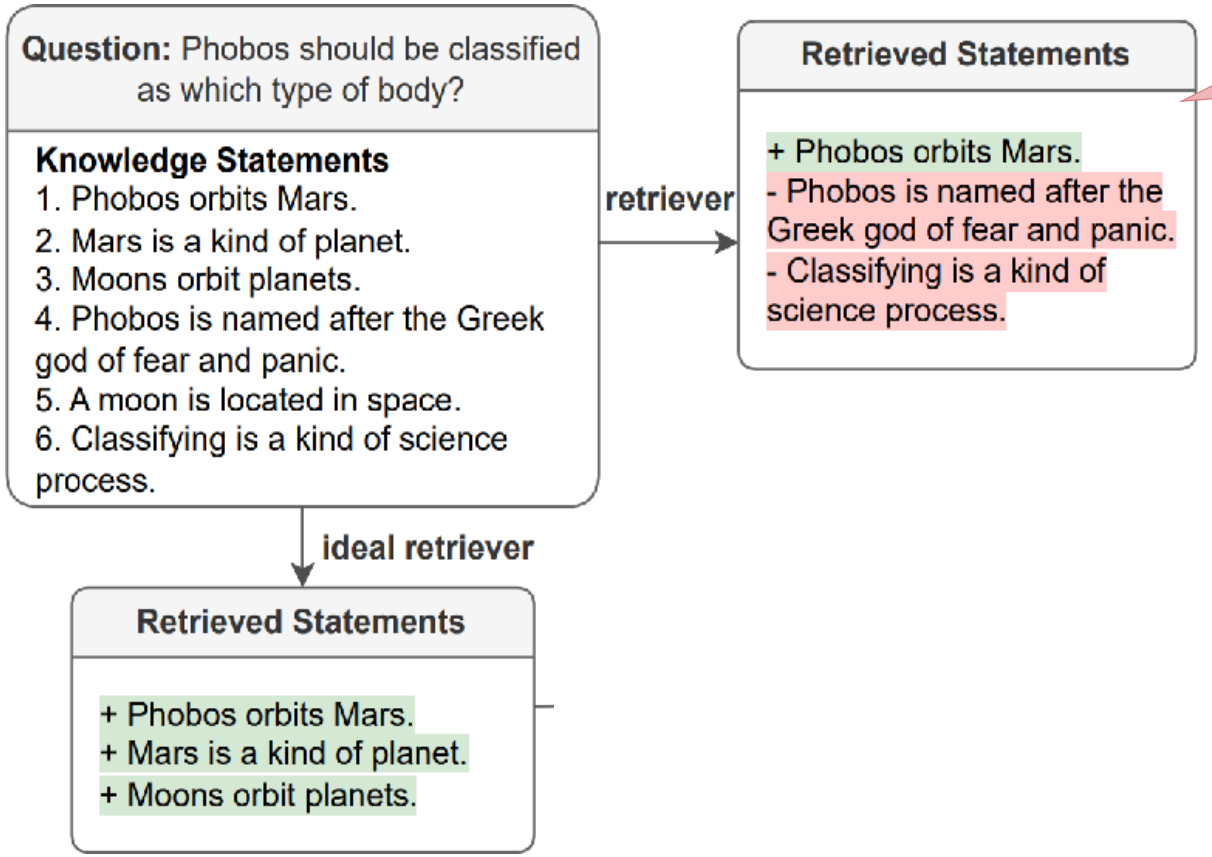
Yatin Nandwani, Vineet Kumar, Dinesh Raghu, Sachindra Joshi and **Luis A. Lastras**
IBM Research, AI
{yatin.nandwani@, vineeku6@in, diraghu1@in, jsachind@in, lastrasl@us}.ibm.com

Abstract

A major concern in using deep learning based generative models for document-grounded dialogs is the potential generation of responses that are not *faithful* to the underlying document. Existing automated metrics used for evaluating the faithfulness of response with respect to the grounding document measure the degree of similarity between the generated response and the document's content. However, these automated



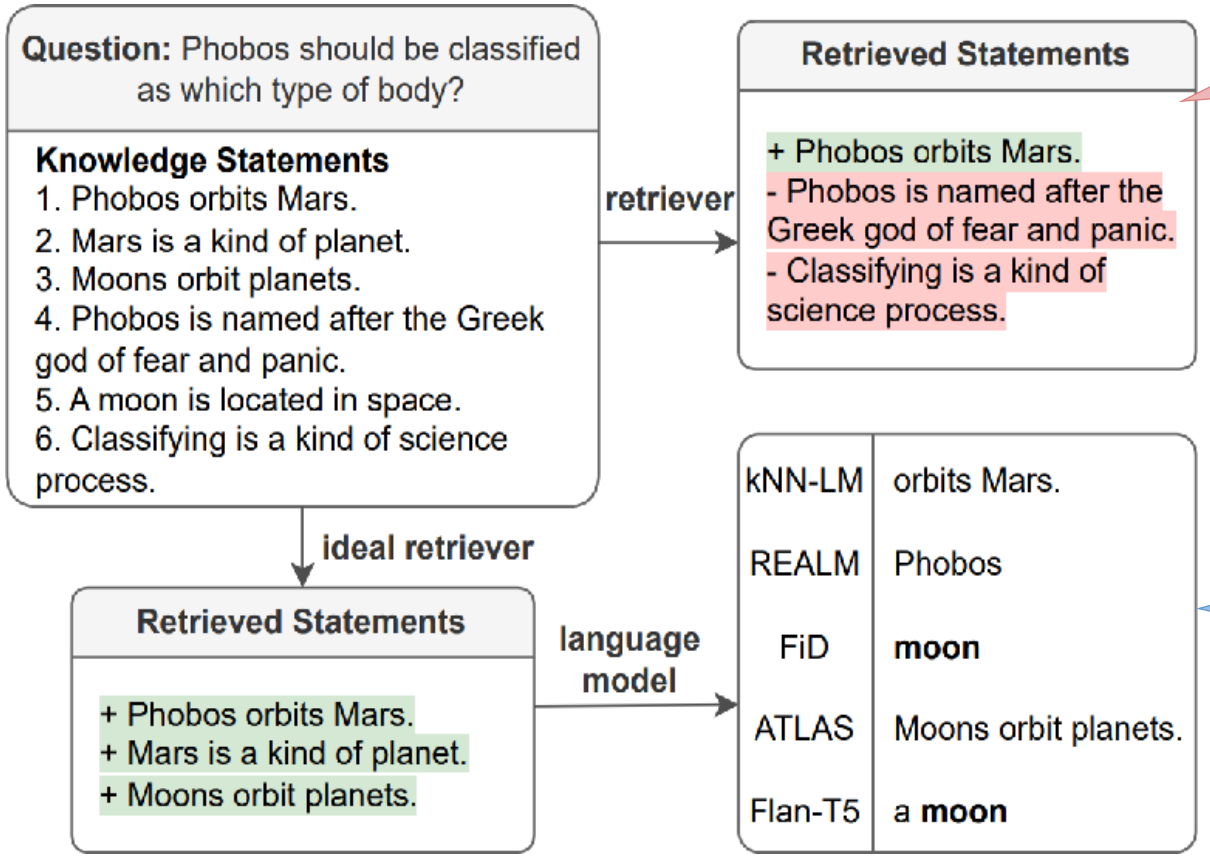
Retrieval Failures



- Retrieval fails to fetch correct information.

BehnamGhader et al. Can Retriever-Augmented Language Models Reason? The Blame Game Between the Retriever and the Language Model. EMNLP Findings 2023.

Reasoning Failures



- Retrieval fails to fetch correct information.
- Even with ideal retriever, LM fails to give right answer.

BehnamGhader et al. Can Retriever-Augmented Language Models Reason? The Blame Game Between the Retriever and the Language Model. EMNLP Findings 2023.

Systematic Knowledge Injection into Large Language Models via Diverse Augmentation for Domain-Specific RAG

Kushagra Bhushan*†[1], Yatin Nandwani*[2], Dinesh Khandelwal[2], Sonam Gupta[2], Gaurav Pandey[2], Dinesh Raghu[2], and Sachindra Joshi[2]

20je0506@mc.iitism.ac.in, {yatin.nandwani@, dikhand1@in., sonam.gupta7@, gpandey1@in.,diraghu1@in., jsachind@in.}ibm.com

[1]IIT(ISM) Dhanbad [2]IBM Research

Abstract

Retrieval-Augmented Generation (RAG) has emerged as a prominent method for incorporating domain knowledge into Large Language Models (LLMs). While RAG enhances response relevance by incorporating retrieved domain knowledge in the context, retrieval errors can still lead to hallucinations and incorrect answers. To recover from retriever failures, domain knowledge is injected by fine-tuning the model to generate the correct response, even in the case of retrieval errors. However, we observe that without systematic knowledge augmentation, fine-tuned LLMs may memorize new information but still fail to extract relevant domain knowledge, leading to poor performance. In this work, we present a novel framework that significantly enhances the fine-tuning process by augmenting the training data in two ways – context augmentation and knowledge paraphrasing. In context augmentation,

for domain-specific applications, such as question answering over specialized corpora, Retrieval-Augmented Generation (RAG) has emerged as a popular approach (Lewis et al., 2020; Karpukhin et al., 2020). Although RAG enhances the relevance of generated answers, it is prone to hallucinations (Ji et al., 2023; Nandwani et al., 2023), especially when the retriever fails to fetch relevant documents from the corpus.

To address this issue, *knowledge injection* techniques (Zhang et al., 2024a; Yoran et al., 2024) have been proposed. Knowledge injection has two main objectives: (1) when the retriever succeeds in fetching correct documents, then the LLM should be able to leverage that information to generate an appropriate response, and (2) when the retriever fails, the LLM should recall the domain specific information from the infused parametric knowledge to generate the response. To achieve them, existing

Adapt LM to Domain Corpus?

Important Resources

- LangChain ; LlamaIndex – *overall frameworks*
- Lucene – *BM25 sparse retriever*
- ANNOY, FAISS, CromaDB - *dense embeddings and retrievers*
- Comprehensive RAG (CRAG) Benchmark – *KDD Cup 2024*

Content credits

- Graham Neubig's lecture - https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

- ACL 2023 Tutorial - https://acl2023-retrieval-lm.github.io/

Slide source: