南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Advanced Natural Language Processing

Lecture 13: Retrieval Augmented Generation with LLMs

陈冠华 CHEN Guanhua

Department of Statistics and Data Science

Information Retrieval

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Information retrieval (IR)
  - Locate and retrieve information that is relevant to a user's query
  - Retrieval and ranking
- Neural-based information retrieval
  - BERT, sentence-bert
  - ColBERT
  - OpenAI Embeddings

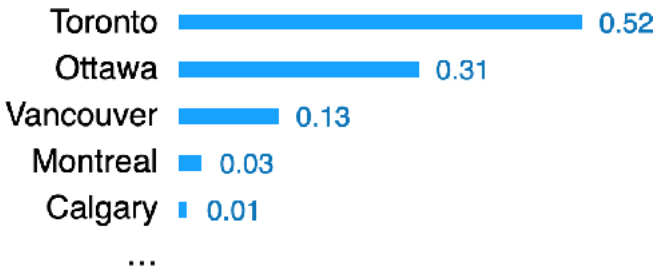
Retrieval-based Language Models

- It is a **language model**

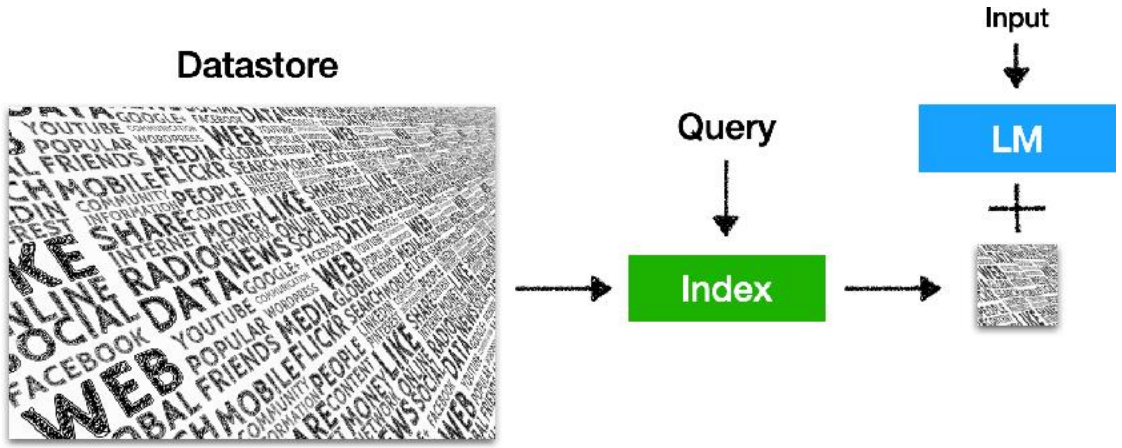
$$P(x_n | x_1, x_2, \cdots, x_{n-1})$$

The capital city of Ontario is _

(can be broadly extended to masked language models or encoder-decoder models)



- It retrieves from an **external datastore** (at least during inference time)



(Also referred to semi-parametric and non-parametric models)

Retrieval for knowledge-intensive NLP tasks

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Representative tasks: open-domain QA, fact checking, entity linking, ..

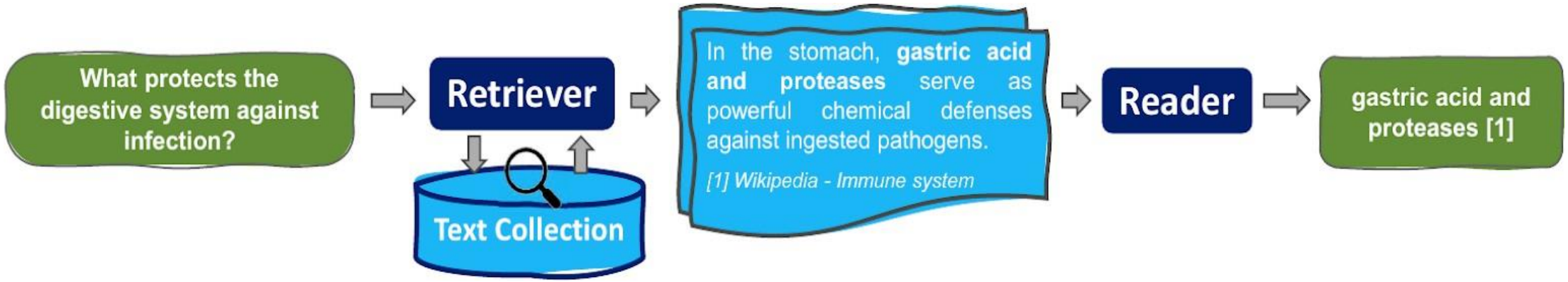


Image: http://ai.stanford.edu/blog/retrieval-based-NLP/

Drives a lot of research on better algorithms for **dense retrieval**, e.g., **DPR, ColBERT**, **ANCE**, **Contriever**…

Why retrieval LMs?

Why Retrieval-Based LMs?

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

LLMs can't memorize all (long-tail) knowledge in their parameters

List 5 important papers authored by Geoffrey Hinton

ChatGPT

Geoffrey Hinton is a renowned computer scientist … Here are five important papers authored by him:

1. "**Learning Internal Representations by Error Propagation**" (with **D. E. Rumelhart** and **R. J. Williams**) - This paper, published in **1986**, ..

2. "**Deep Boltzmann Machines**" (with **R. Salakhutdinov**) - Published in **2009**, ..

…

4. "**Deep Learning**" (with Y. Bengio and A. Courville) - Published as a book in **2016**,…

5. "**Attention Is All You Need**" (with V. Vaswani, N. Shazeer, et al.) - Published in **2017**, this paper introduced the Transformer model,…

Why Retrieval-Based LMs?

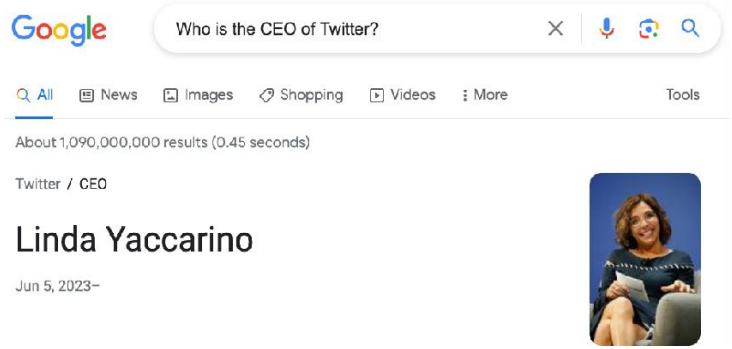
南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

LLMs' knowledge is easily outdated and hard to update

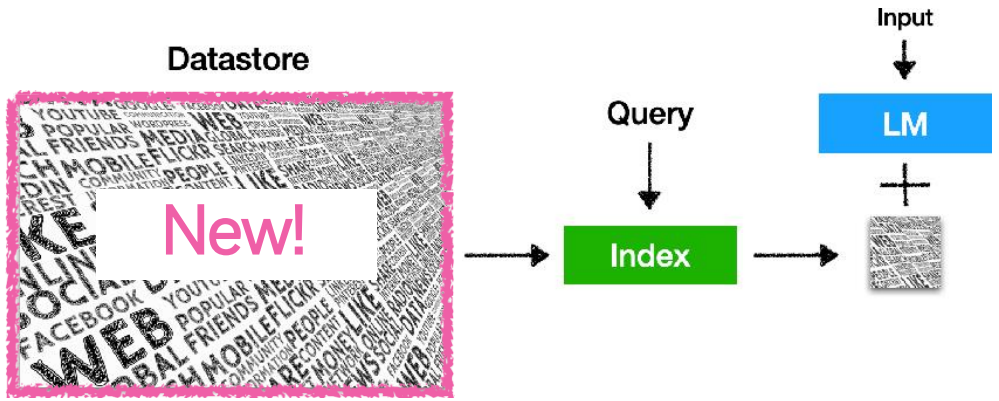
Who is the CEO of Twitter?

ChatGPT

As of my **knowledge cutoff in September 2021**, the CEO of Twitter is **Jack Dorsey**....

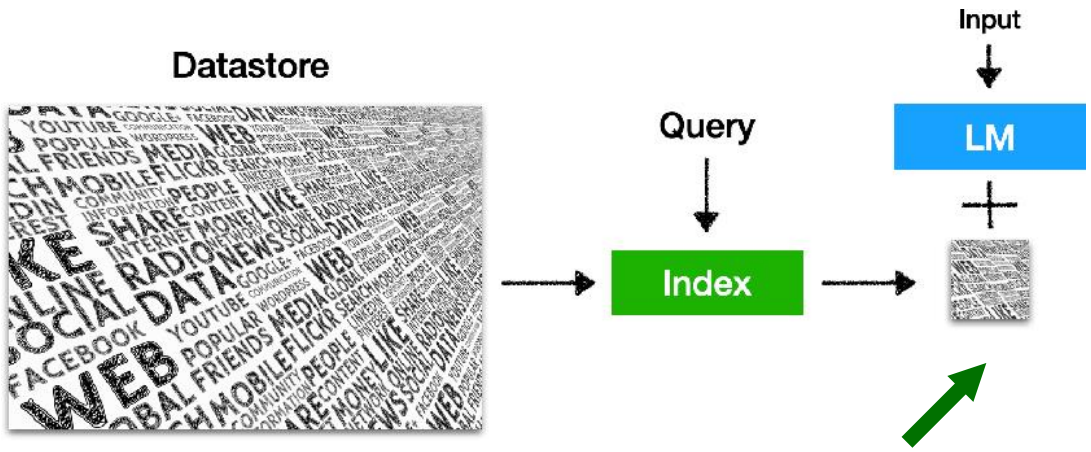


- Existing **knowledge editing** methods are still NOT scalable
- The datastore can be easily **updated** and **expanded** - even without retraining!



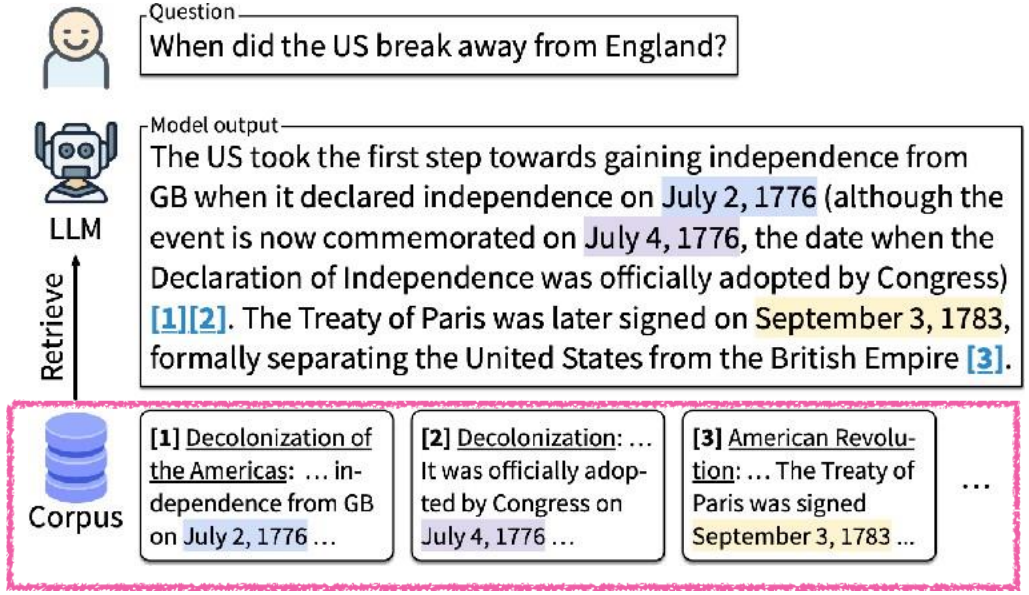
Why Retrieval-Based LMs?

LLMs' output is challenging to interpret and verify



Can trace knowledge source from retrieval results - better interpretability & control

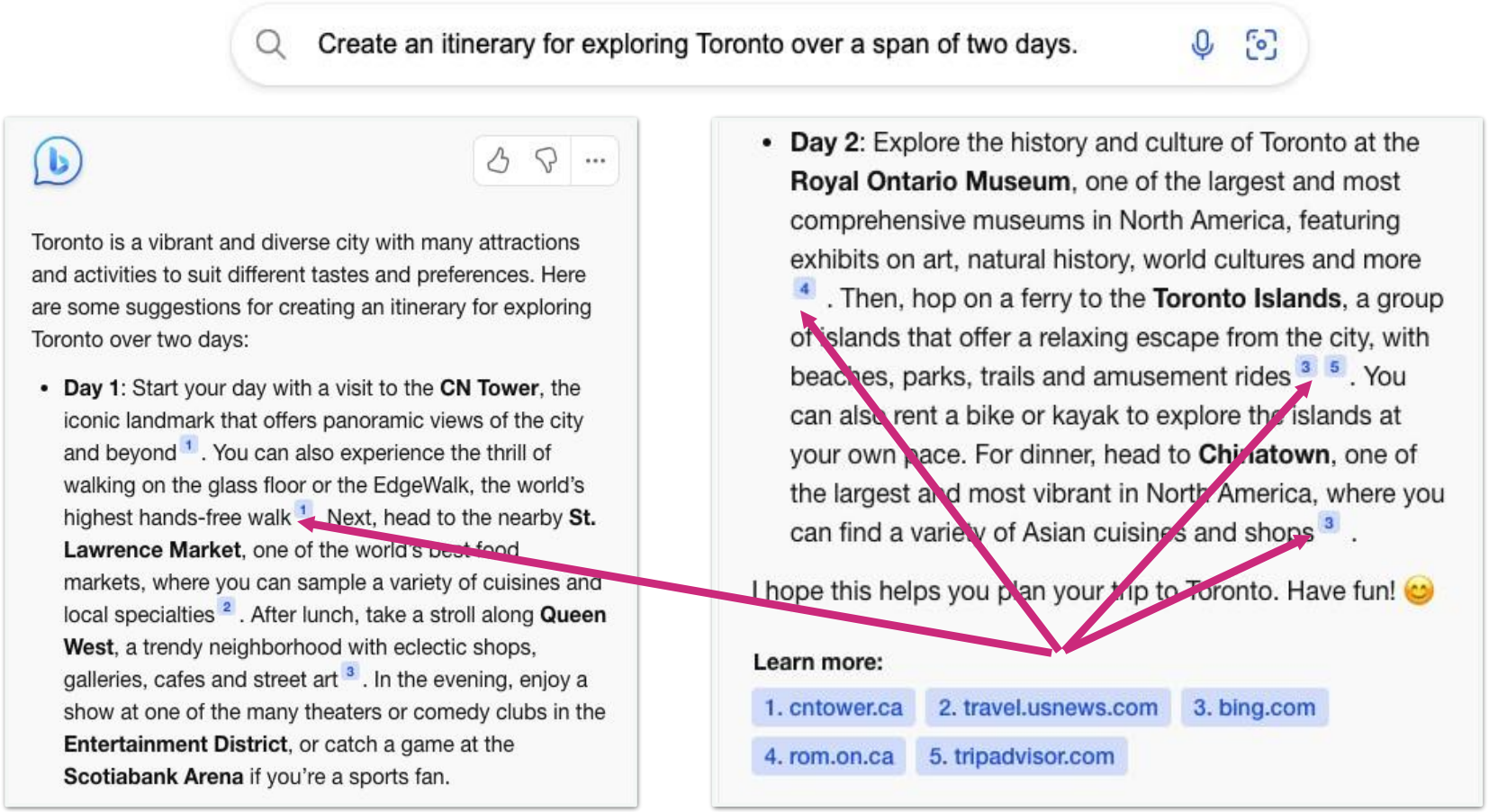
Generating text with citations



Why Retrieval-Based LMs?

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

LLMs' output is challenging to interpret and verify



Google Shares Drop $100 Billion

- An accompanying GIF prompts Bard with the question, "What new discoveries from the James Webb Space Telescope can I tell my 9-year-old about?“
- The chatbot responds with a few bullet points, including the claim that the telescope took the very first pictures of "exoplanets“, or planets outside of earth's solar system.
- But the James Webb Telescope didn't discover exoplanets.

TECHNOLOGY

Google shares drop $100 billion after its new AI chatbot makes a mistake

February 9, 2023 · 10:15 AM ET

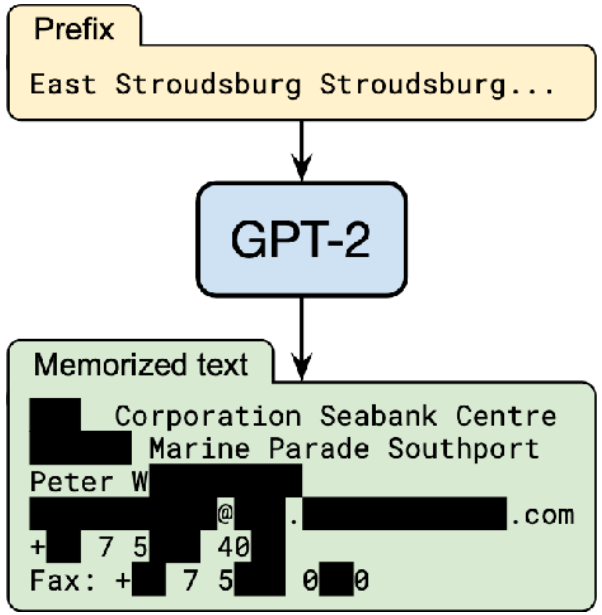
By Emily Olson

Shares for Google's parent company, Alphabet, dropped 9% Wednesday after its AI chatbot, Bard, gave an incorrect answer.

Dan Kitwood/Getty Images

Why Retrieval-Based LMs?

LLMs are shown to easily leak private training data



Individualization on private data by storing it in the datastore

Carlini et al. 2021. "Extracting Training Data from Large Language Models"

Why Retrieval-Based LMs?

LLMs are large and expensive to train and run

- Long-term goal
  - Can we possibly reduce the training and inference costs, and scale down the size of LLMs?
  - e.g., RETRO (Borgeaud et al., 2021): "obtains comparable performance to GPT-3 on the Pile, despite using 25x fewer parameters"

Why Retrieval-Based LMs?

LLMs are large and expensive to train and run

- It seems scaling larger and larger models is the main way of improving the performance
- But with a tremendous increase in training energy cost
  - Additional computations at training and inference time
  - Increased memorization of the training data

Can we separate language information from world knowledge information?

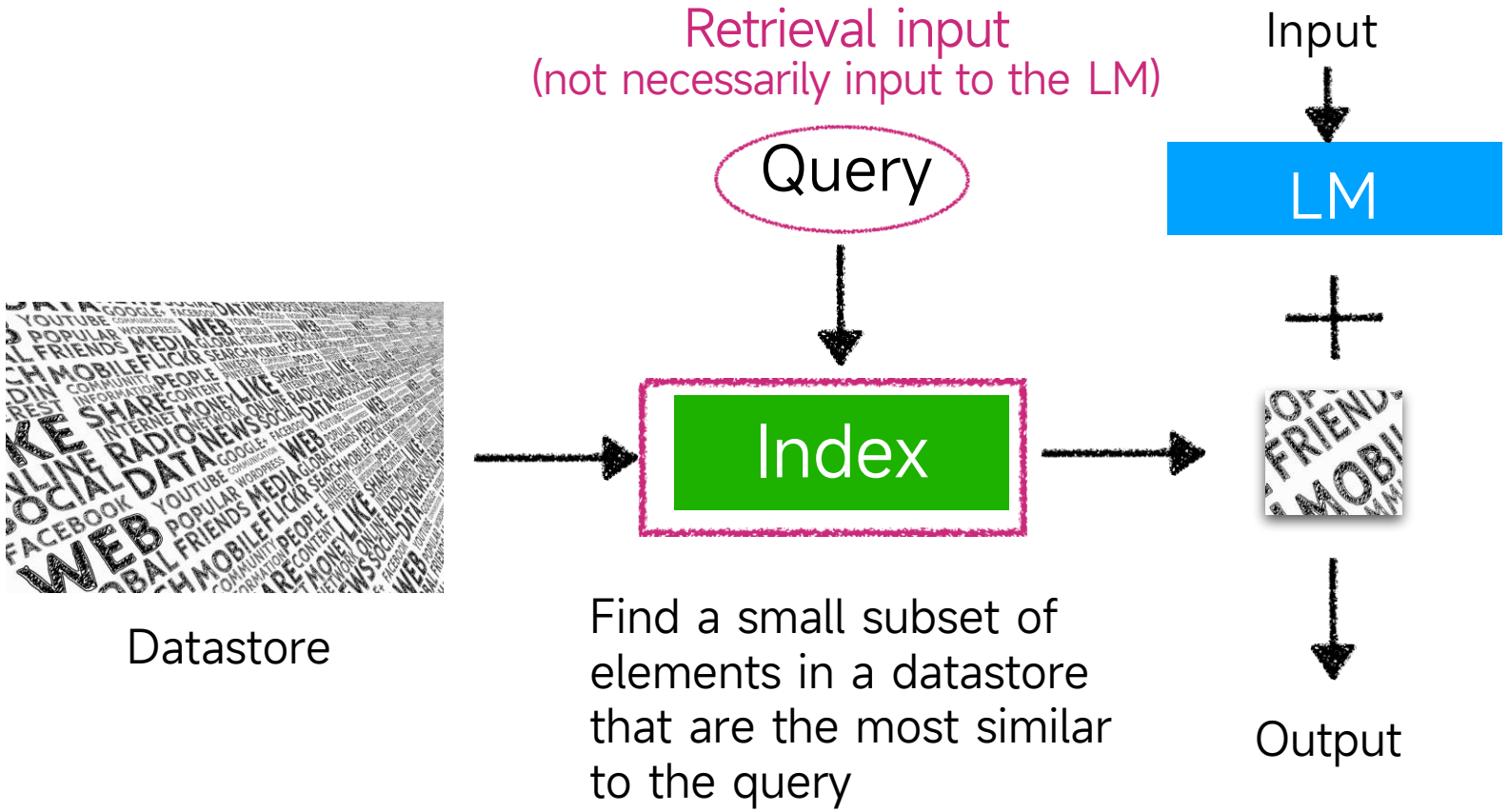
Why Retrieval-Based LMs?

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Tackling Inefficiency
  - Retrieval-based models can be much smaller and faster
- Tackling Opaqueness
  - When the model produces an answer, we can read the sources it retrieved and judge their relevance and credibility for ourselves.
- Tackling Static Knowledge
  - The retrieval knowledge store can be efficiently updated or expanded by modifying the text corpus
  - Real-time/Dynamic data

- When you fine-tune a model, it's like studying for an exam one week away.
- When you insert knowledge into the prompt (e.g., via retrieval), it's like taking an exam with open notes.

Retrieval-augmented Language Model



Similarity Score

Goal: find a small subset of elements in a datastore that are the most similar to the query

A similarity score between two pieces of text

Example $\text{sim}(i, j) = \text{tf} \times \log \frac{N}{\text{df}_j}$

N: # of total docs

df_j: # of docs containing

tf: # of occurrences of in j

Example $\text{sim}(i, j) = \text{Encoder}(i) \cdot \text{Encoder}(j)$

Maps the text into an d-dimensional vector

An entire field of study on how to get (or learn) the similarity function better

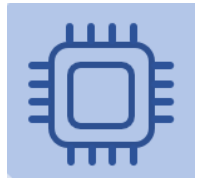
Faiss Wiki Documentation — Faiss documentation

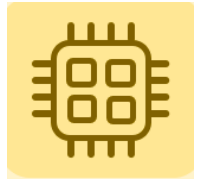
FAISS

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Method	Class name	index_factory	Main parameters	Bytes/vector	Exhaustive	Comments
Exact Search for L2	IndexFlatL2	"Flat"	d	4*d	yes	brute-force
Exact Search for Inner Product	IndexFlatIP	"Flat"	d	4*d	yes	also for cosine (normalize vectors beforehand)
Hierarchical Navigable Small World graph exploration	IndexHNSWFlat	"HNSW,Flat"	d, M	4*d + x * M * 2 * 4	no	
Inverted file with exact post-verification	IndexIVFFlat	"IVFx,Flat"	quantizer, d, nlists, metric	4*d + 8	no	Takes another index to assign vectors to inverted lists. The 8 additional bytes are the vector id that needs to be stored.
Locality-Sensitive Hashing (binary flat index)	IndexLSH	-	d, nbits	ceil(nbits/8)	yes	optimized by using random rotation instead of random projections
Scalar quantizer (SQ) in flat mode	IndexScalarQuantizer	"SQ8"	d	d	yes	4 and 6 bits per component are also implemented.
Product quantizer (PQ) in flat mode	IndexPQ	"PQx", "PQ"M"x"nbits	d, M, nbits	ceil(M * nbits / 8)	yes	
IVF and scalar quantizer	IndexIVFScalarQuantizer	"IVFx,SQ4" "IVFx,SQ8"	quantizer, d, nlists, qtype	SQfp16: 2 * d + 8, SQ8: d + 8 or SQ4: d/2 + 8	no	Same as the IndexScalarQuantizer
IVFADC (coarse quantizer+PQ on residuals)	IndexIVFPQ	"IVFx,PQ"y"x"nbits	quantizer, d, nlists, M, nbits	ceil(M * nbits/8)+8	no	
IVFADC+R (same as IVFADC with re-ranking based on codes)	IndexIVFPQR	"IVFx,PQy+z"	quantizer, d, nlists, M, nbits, M_refine, nbits_refine	M+M_refine+8	no	

Exact Search





CPU vs. GPU

Approximate Search
(Relatively easy to scale to ~1B elements)

https://github.com/facebookresearch/faiss/wiki

Vector Database

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- A type of database that indexes and stores vector embeddings for fast retrieval and similarity search
- Advantages over vector indices like Faiss
  - Data management
  - Metadata storage and filtering
  - Scalability
  - Real-time updates
  - Backups and collections
  - Data security and access control

Pinecone

Chroma

Milvus

[Vector stores | Langchain]

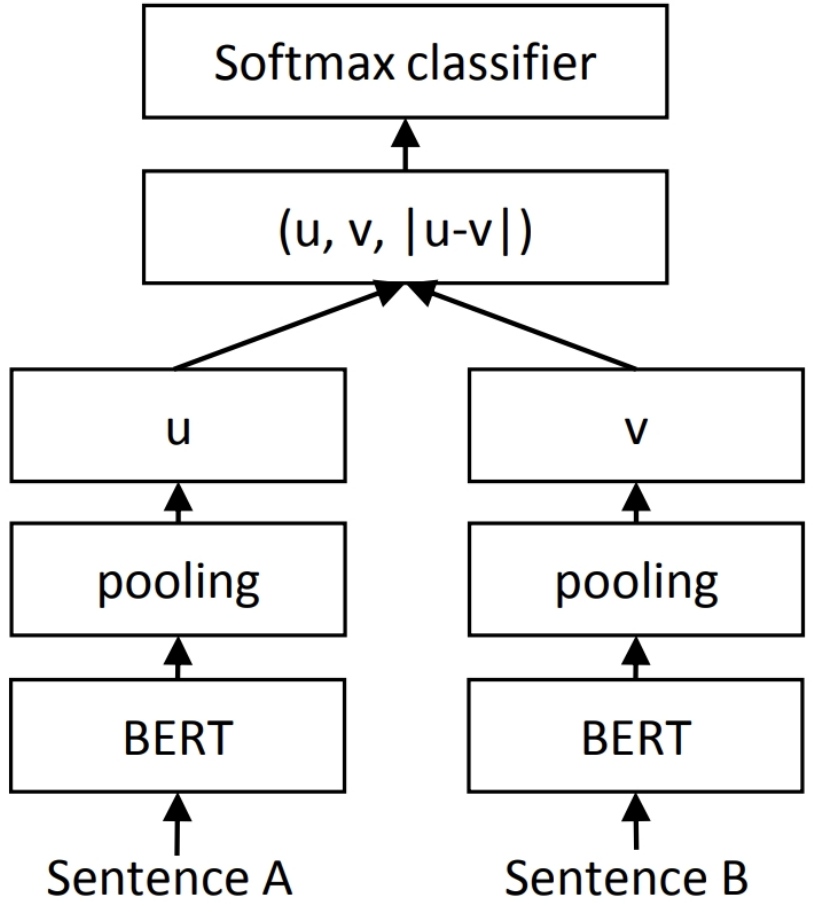
Sentence-BERT

- Training data:
  - Combination of the SNLI and the Multi-Genre NLI dataset
- Default pooling strategy is MEAN.

Classification Objective Function. We concatenate the sentence embeddings u and v with the element-wise difference $|u-v|$ and multiply it with the trainable weight $W_t \in \mathbb{R}^{3n \times k}$:

$$o = \text{softmax}(W_t(u, v, |u-v|))$$

where n is the dimension of the sentence embeddings and k the number of labels. We optimize cross-entropy loss. This structure is depicted in



[1908.10084] Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (arxiv.org)

Dense Passage Retriever (DPR)

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Retrieve k passages from M documents
  - k=20~100, M=millions~billions
- Passage encoder + query encoder (based on BERT)
  - Take the representation at the [CLS] token as the output
  - Given a question q at run-time, we derive its embedding $v_q = E_Q(q)$ and retrieve the top k passages with embeddings closest to v_q.

are the closest to the question vector. We define the similarity between the question and the passage using the dot product of their vectors:

$$\text{sim}(q, p) = E_Q(q)^\intercal E_P(p). \qquad (1)$$

[2004.04906] Dense Passage Retrieval for Open-Domain Question Answering (arxiv.org)

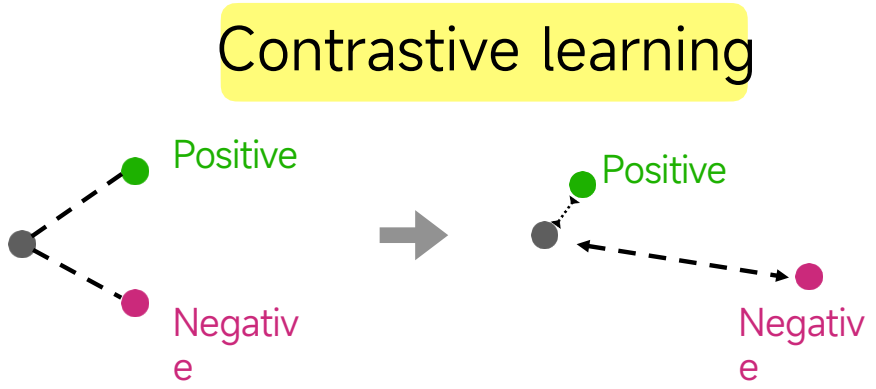
Dense Passage Retriever (DPR)

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

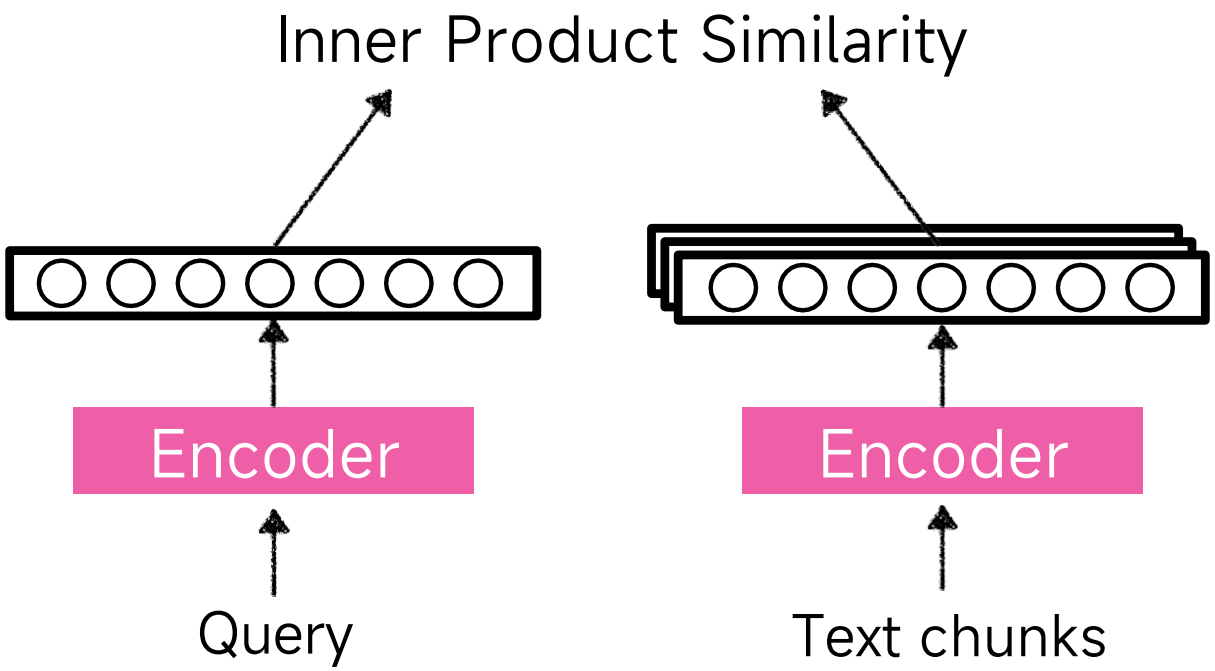
- Training data
  - Question Answering Datasets
  - Each instance contains one question q_i and one relevant (positive) passage p_i^+ , along with n irrelevant (negative) passages $p_{i,j}^-$.
- In-batch negative
  - Re-using gold passages from the same batch as negatives
  - Any (q_i, p_j) pair is a positive example when $i = j$, and negative otherwise

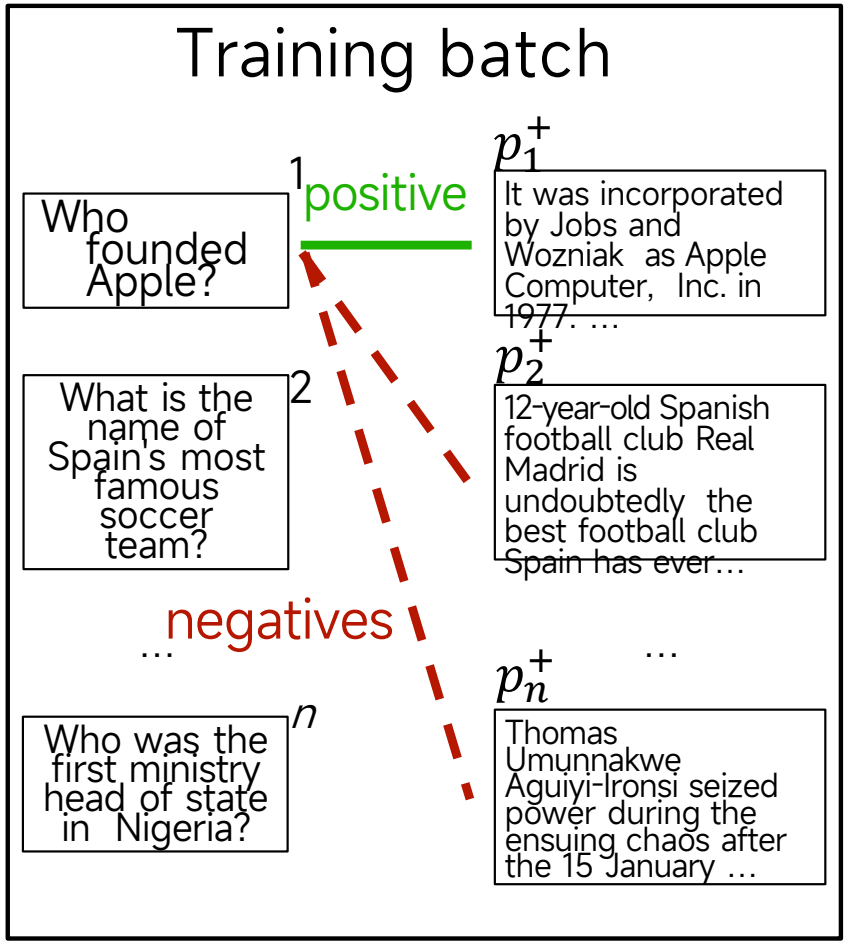
$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-)$$

$$= -\log \frac{e^{\text{sim}(q_i, p_i^+)}}{e^{\text{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(q_i, p_{i,j}^-)}}$$



Dense Passage Retriever (DPR)





[DPR/train_dense_encoder.py at main · facebookresearch/DPR]

REALM

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Augment pretraining with a latent knowledge retriever
  - Allows the model to retrieve and attend over documents from a large corpus during pre-training, fine-tuning and inference.
- Pretrain a knowledge retriever in an unsupervised manner
  - Using masked language modeling as the learning signal
  - Backpropagating through a retrieval step that considers millions of documents
- Key intuition
  - Train the retriever using a performance-based signal from unsupervised text
  - A retrieval that improves the language model's perplexity is helpful and should be rewarded, while an uninformative retrieval should be penalized.

[2002.08909] REALM: Retrieval-Augmented Language Model Pre-Training (arxiv.org)

REALM

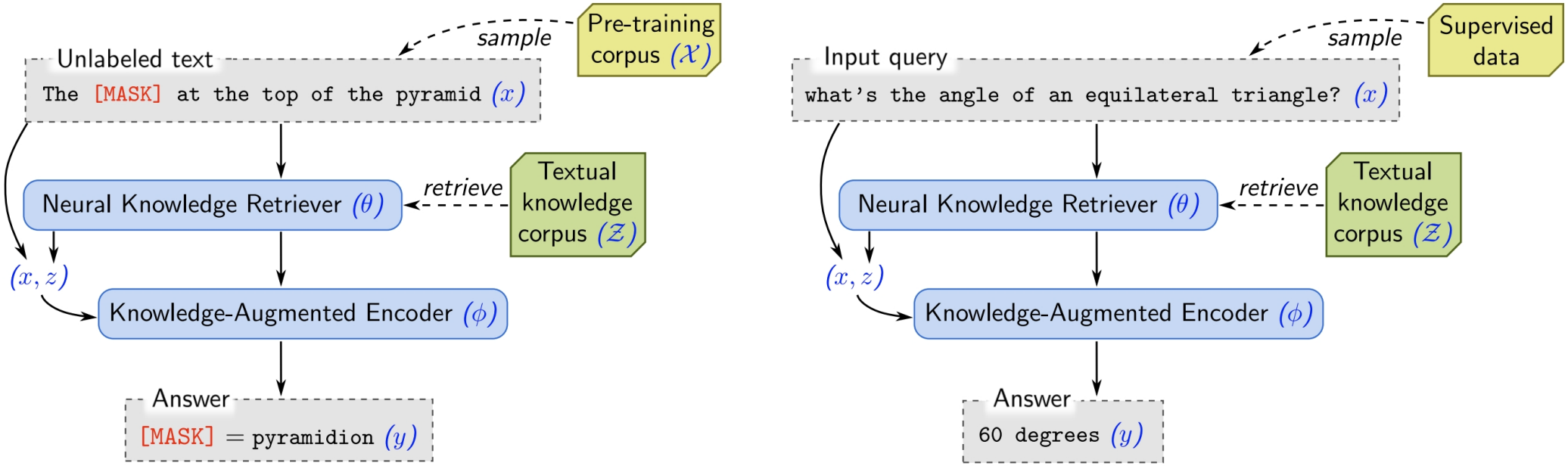


Figure 2. The overall framework of REALM. **Left:** *Unsupervised pre-training.* The knowledge retriever and knowledge-augmented encoder are jointly pre-trained on the unsupervised language modeling task. **Right:** *Supervised fine-tuning.* After the parameters of the retriever (θ) and encoder (ϕ) have been pre-trained, they are then fine-tuned on a task of primary interest, using supervised examples.

REALM

- MLM pretraining
  - Retrieve, then predict
  - Knowledge retriever $p(z|x)$
  - knowledge-augmented encoder $p(y|z,x)$
- Finetune on open-QA data
  - Assume that the answer y can be found as a contiguous sequence of tokens in some document z.
- Find the approximate top k documents
- "Refresh" the index by asynchronously re-embedding and re-indexing all documents every several hundred training steps.

$$p(y \mid x) = \sum_{z \in \mathcal{Z}} p(y \mid z, x)\, p(z \mid x).$$

$$p(z \mid x) = \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')},$$

$$f(x, z) = \texttt{Embed}_{\texttt{input}}(x)^\top \texttt{Embed}_{\texttt{doc}}(z),$$

$$p(y \mid z, x) \propto \sum_{s \in S(z,y)} \exp\left(\texttt{MLP}\left(\left[h_{\texttt{START(s)}}; h_{\texttt{END(s)}}\right]\right)\right)$$

$$h_{\texttt{START(s)}} = \texttt{BERT}_{\texttt{START(s)}}(\texttt{join}_{\texttt{BERT}}(x, z_{\text{body}})),$$

$$h_{\texttt{END(s)}} = \texttt{BERT}_{\texttt{END(s)}}(\texttt{join}_{\texttt{BERT}}(x, z_{\text{body}})),$$

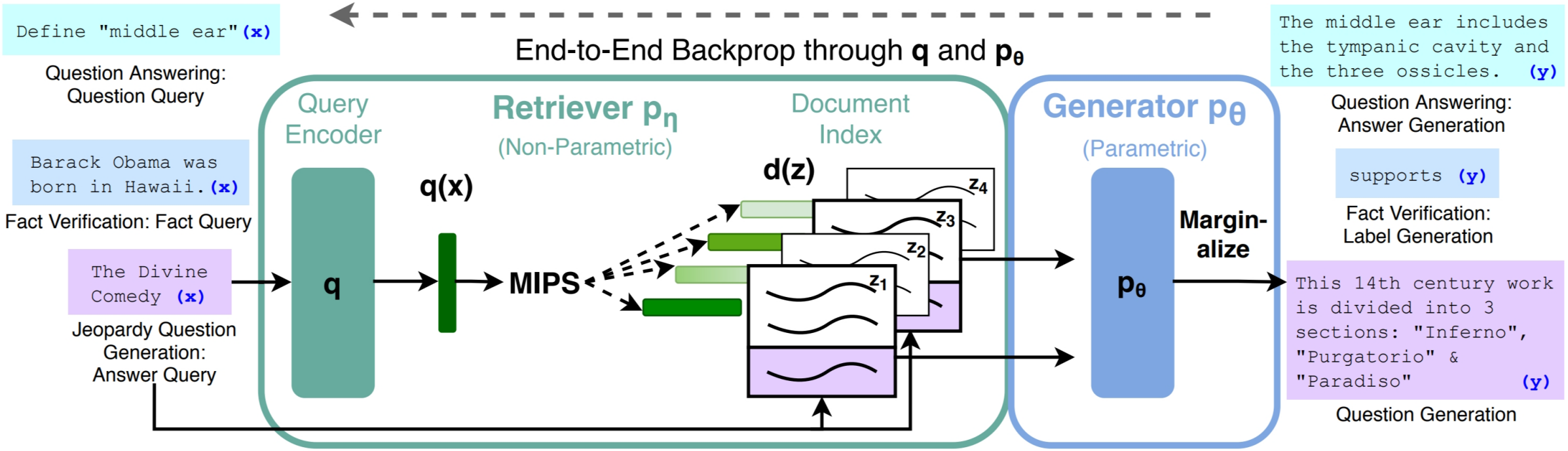
RAG

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Motivation
  - REALM only uses MLM which only works for NLU tasks
- Combine pre-trained parametric and non-parametric memory for language generation
  - Parametric memory is a pre-trained seq2seq model
  - The non-parametric memory is a dense vector index of Wikipedia, accessed with a pre-trained neural retriever DPR
- RAG models generate more specific, diverse and factual language than a state-of-the-art parametric-only seq2seq baseline

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

RAG


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

RAG

RAG-Token Model

- The top K documents are retrieved using the retriever
- The generator produces a distribution for the next output token for each document before marginalizing
- Repeating the process with the following output token

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

transformers/src/transformers/models/rag/modeling_rag.py at main · huggingface/transformers (github.com)

RAG

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

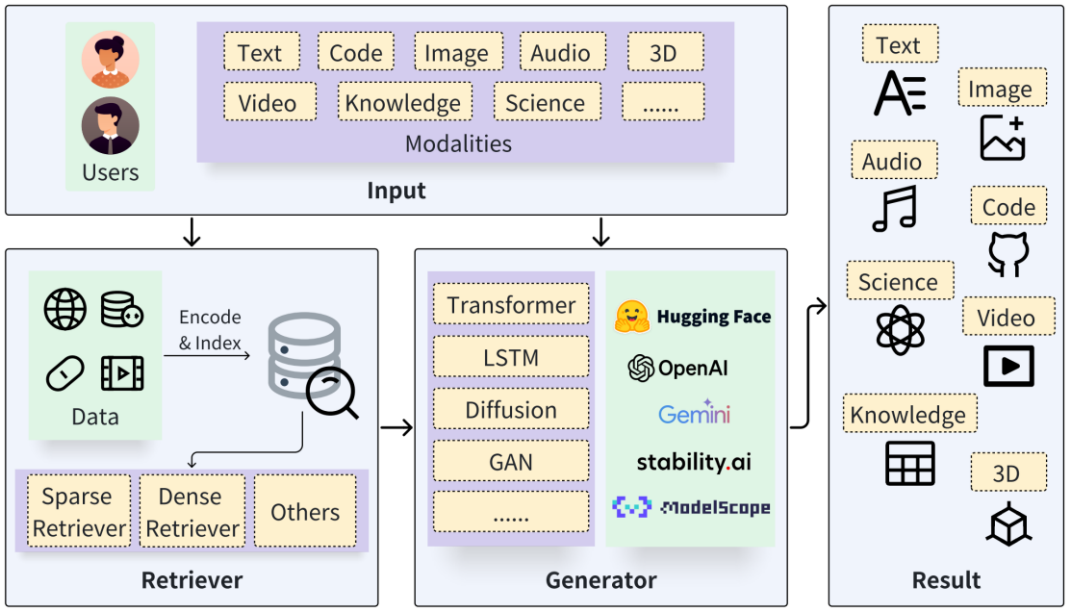
- RAG-Sequence Model
  - Treats the retrieved document as a single latent variable that is marginalized to get the seq2seq probability $p(y|x)$ via a top-K approximation.

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

token likelihood, hence we cannot solve it with a single beam search. Instead, we run beam search for each document z, scoring each hypothesis using $p_\theta(y_i|x,z,y_{1:i-1})$. This yields a set of hypotheses Y, some of which may not have appeared in the beams of all documents. To estimate the probability

RAG Background

- **Basic Definition:**
  - RAG = Retrieval + Augmentation + Generation
  - Traditional LLM generation:
    - Relies solely on the knowledge in the model parameters to generate answers.
  - RAG systems:
    - First, relevant information is retrieved from an external knowledge base, and then the search results are combined to generate answers.



Advantages of RAG

- Knowledge base can be updated anytime (no need to retrain models)
- Can store massive long-tail knowledge
- Facts based on retrieval reduce hallucination
- Sensitive data can be stored in private knowledge bases

References

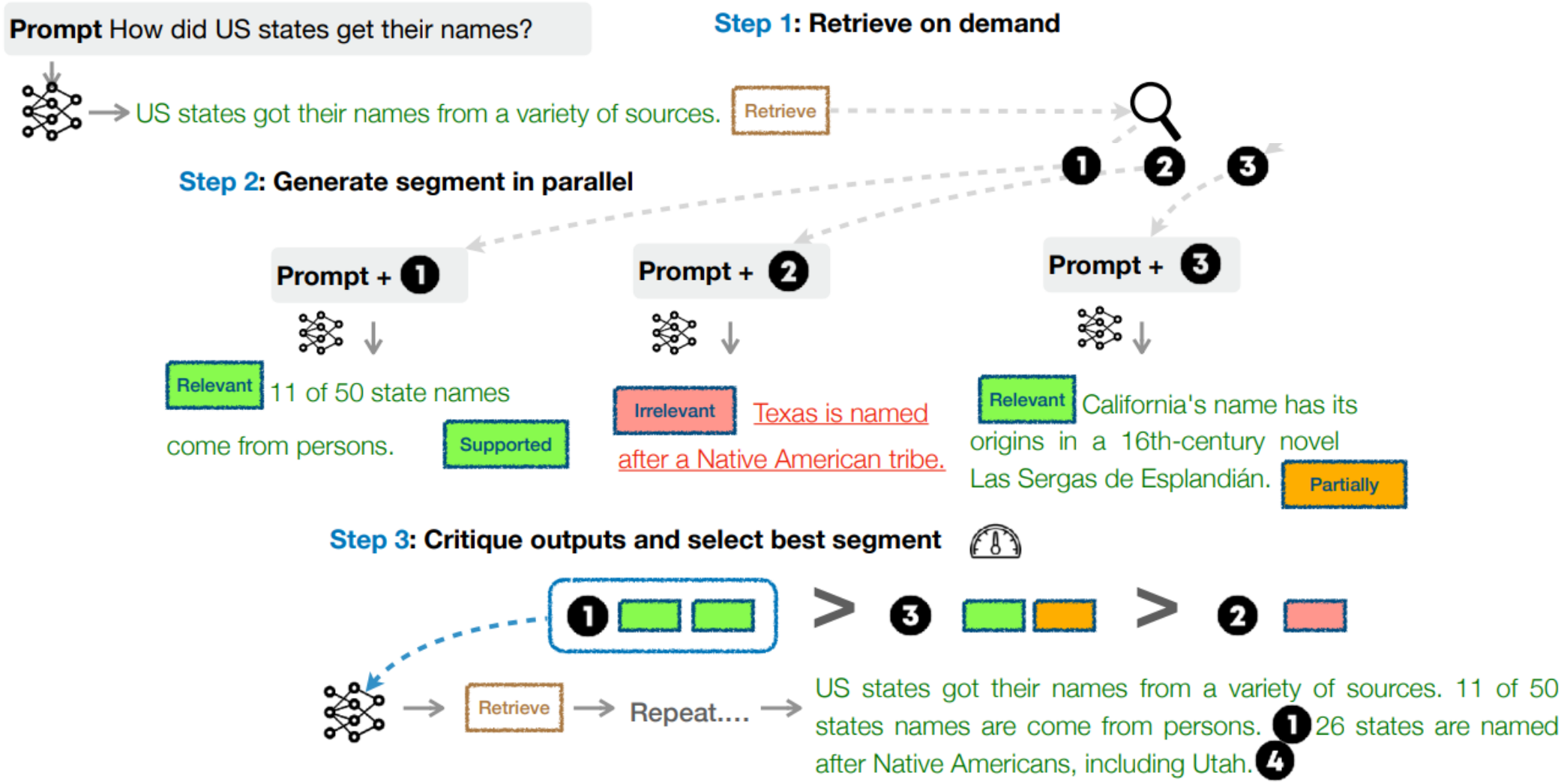
- Embeddings
  - jinaai/jina-embeddings-v4 · Hugging Face
  - Qwen/Qwen3-Embedding-4B · Hugging Face
- RAG framework
  - LangGraph, LangGraph, langgraph/examples/rag/langgraph_agentic_rag.ipynb
  - Dify, langgenius/dify: Production-ready platform for agentic workflow development.
  - UltraRAG, OpenBMB/UltraRAG: UltraRAG 2.0
- RAG tutorials
  - Danielskry/Awesome-RAG
  - lizhe2004/Awesome-LLM-RAG-Application

RAG Enhancement Methods

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Retrieval-enhanced LLMs
  - **Self-RAG**
  - GraphRAG
- RAG Instruction Tuning
  - REPLUG
  - RA-DIT
- RAG Optimization
  - FILCO
  - Adaptive Retrieval
- Embedding Model
  - Qwen3-embedding
- RL
  - Search-R1

Self-RAG



Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Problems of RAG Models

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Robustness to Irrelevant Contexts

"The 2022 FIFA World Cup, also branded as Qatar 2022 was the 22nd FIFA World Cup, the world championship for national football teams organized by FIFA."

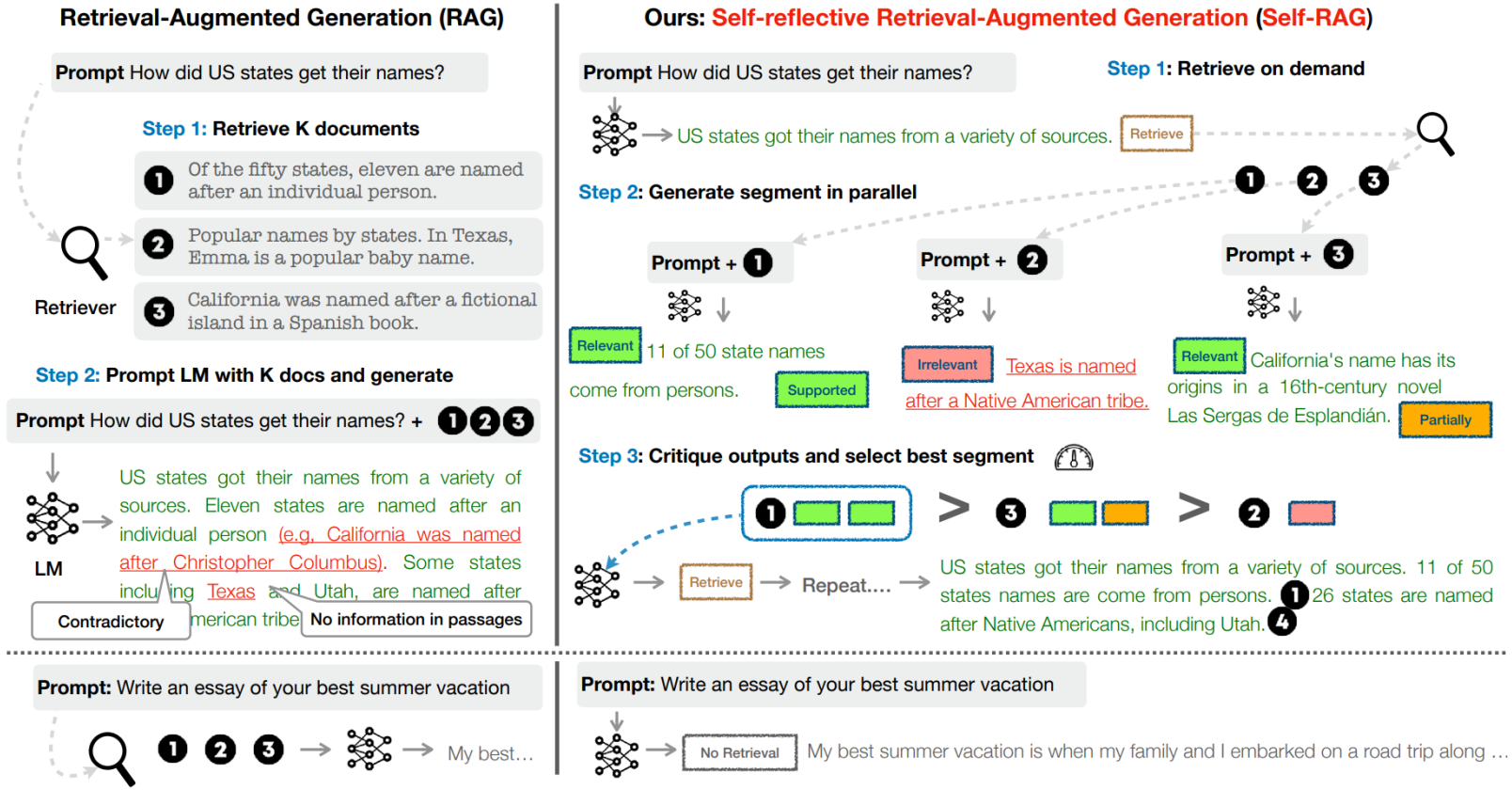
"Write an essay of your best summer vacation"

Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection
https://cs.uwaterloo.ca/~wenhuche/teaching/cs886/

Self-RAG

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Learning to Retrieve, Generate, and Critique through Self-Reflection
  - Make retrieval decisions
  - Evaluate retrieved passages
  - Critique its own generation



Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Self-RAG

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

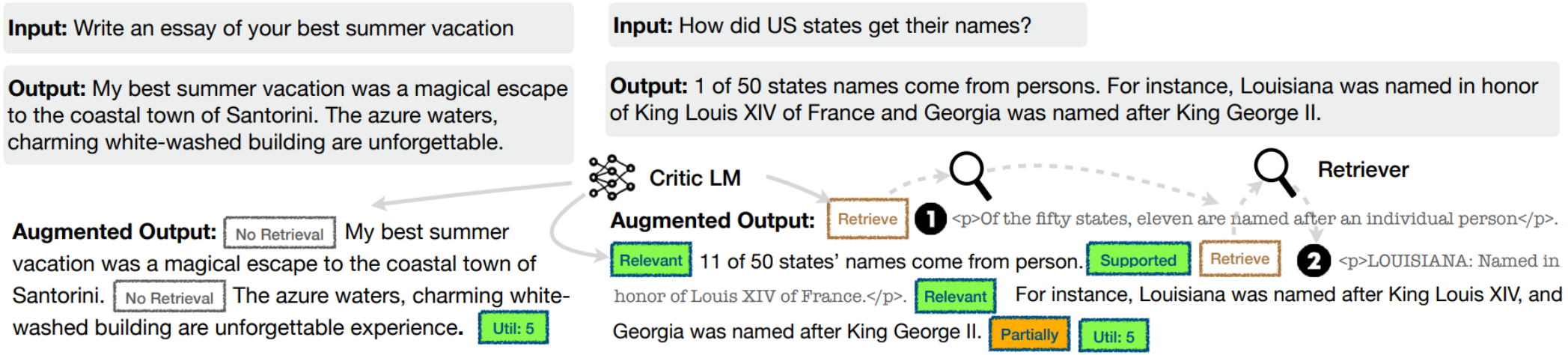
- Self-RAG introduces several special tokens that guide the generation process
- These tokens are embedded directly into the generation process, allowing the model to make dynamic decisions about retrieval and content quality.

Type	Input	Output	Definitions
Retrieve	x / x, y	{yes, no, continue}	Decides when to retrieve with $\mathcal{R}$
ISREL	x, d	{**relevant**, irrelevant}	d provides useful information to solve x.
ISSUP	x, d, y	{**fully supported**, partially supported, no support}	All of the verification-worthy statement in y is supported by d.
ISUSE	x, y	{**5**, 4, 3, 2, 1}	y is a useful response to x.

Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Self-RAG

- *Examples of augmented outputs containing reflection tokens for both retrieval-unnecessary (left) and retrieval-necessary (right) scenarios. The model can adaptively determine when retrieval is needed and assess the quality of retrieved information.*



Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Self-RAG

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Training Process
  - **Data Preparation**: A diverse training corpus is created by:
    - Collecting inputs from various tasks requiring different levels of external knowledge
    - Using a retriever to fetch relevant passages for each input
    - Employing the critic model to generate reflection tokens
  - **Augmented Dataset Creation**: The training data is augmented by interleaving:
    - Original text
    - Retrieved passages
    - Reflection tokens (retrieval decisions, relevance assessments, support evaluations)
  - **Model Training**: A generator model is trained using standard language modeling objectives to predict:
    - Regular text tokens, Reflection tokens, The relationships between them

Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Self-RAG

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Inference
  - Retrieval frequency adjustment
  - Support weight customization

Algorithm 1 SELF-RAG Inference

Require: Generator LM $\mathcal{M}$, Retriever $\mathcal{R}$, Large-scale passage collections $\{d_1, \ldots, d_N\}$

Input: input prompt x and preceding generation $y_{<t}$, **Output:** next output segment y_t
$\mathcal{M}$ predicts [Retrieve] given $(x, y_{<t})$
if [Retrieve] == Yes **then**
    Retrieve relevant text passages $\mathbf{D}$ using $\mathcal{R}$ given (x, y_{t-1}) ▷ Retrieve
    $\mathcal{M}$ predicts [IsRel] given x, d and y_t given $x, d, y_{<t}$ for each $d \in \mathbf{D}$ ▷ Generate
    $\mathcal{M}$ predicts [IsSup] and [IsUse] given x, y_t, d for each $d \in \mathbf{D}$ ▷ Critique
    Rank y_t based on [IsRel], [IsSup], [IsUse] ▷ Detailed in Section 3.3
else if [Retrieve] == No **then**
    $\mathcal{M}_{gen}$ predicts y_t given x ▷ Generate
    $\mathcal{M}_{gen}$ predicts [IsUse] given x, y_t ▷ Critique

Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Self-RAG

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Input Identify the two types of computer memory. (from `gpt4_alpaca`)

Output [Retrieve=Yes]<p>Computer memory Computer memory – In computing, memory refers to the computer hardware integrated circuits that store information for immediate use in a computer; it is synonymous with the term "primary storage". Computer memory operates at a high speed, for example random-access memory (RAM), as a distinction from storage that provides slow-to-access information but offers higher capacities. If needed, contents of the computer memory can be transferred to secondary storage; a very common way of doing this is through a memory management technique called "virtual memory". An archaic synonym for memory is store. The term "memory", meaning "primary storage" or "main memory", is<p>[IsRel =Relevant]Primary memory, also known as main memory, is the memory that the computer can directly access and manipulate. [IsSup =Fully Supported] [Retrieve=Yes]<p>Computer memory–often associated with addressable semiconductor memory, i.e. integrated circuits consisting of silicon-based transistors, used for example as primary storage but also other purposes in computers and other digital electronic devices. There are two main kinds of semiconductor memory, volatile and non-volatile. Examples of non-volatile memory are flash memory (used as secondary memory) and ROM, PROM, EPROM and EEPROM memory (used for storing firmware such as BIOS). Examples of volatile memory are primary storage, which is typically dynamic random-access memory (DRAM), and fast CPU cache memory, which is typically static random-access memory (SRAM) that is fast but energy-consuming, offering lower memory<p>[IsRel =Relevant]Primary memory can be accessed quickly, but it is volatile, meaning that the data is lost if the power is turned off.[IsSup =Partially Supported]...[IsUse :5]

Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

GraphRAG

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- From Local to Global: A GraphRAG Approach to Query-Focused Summarization

  - Self-RAG achieves the interaction between retrieval and generation, but still relies on local similarity retrieval.
  - GraphRAG achieves global knowledge integration by introducing a graph structure.

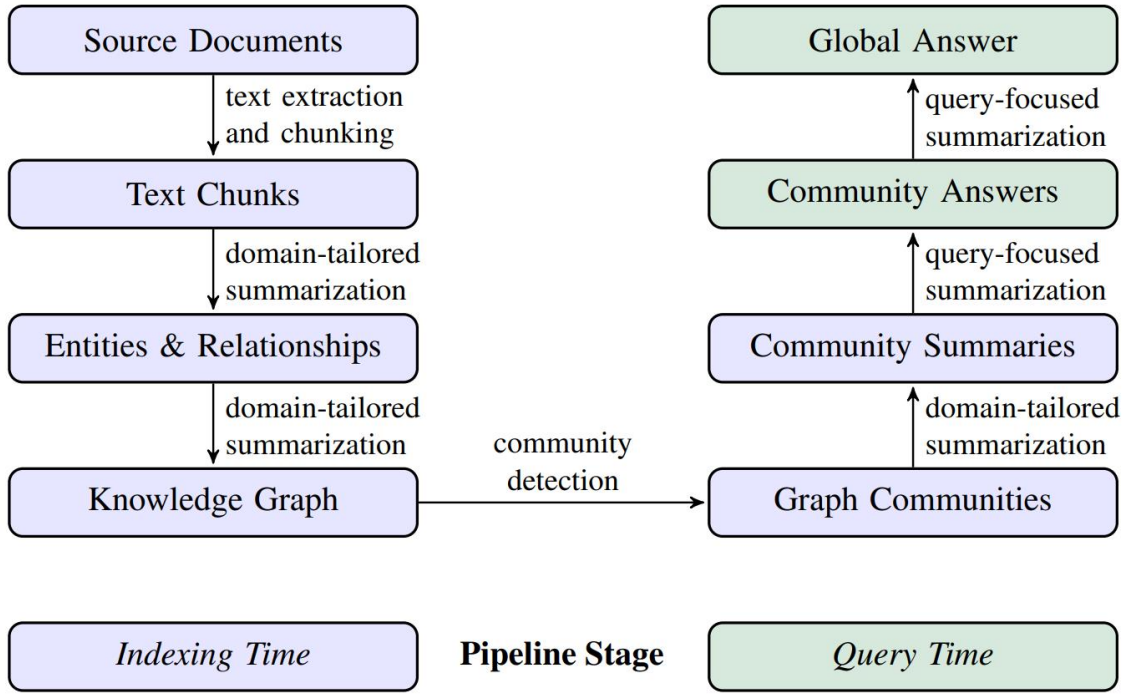
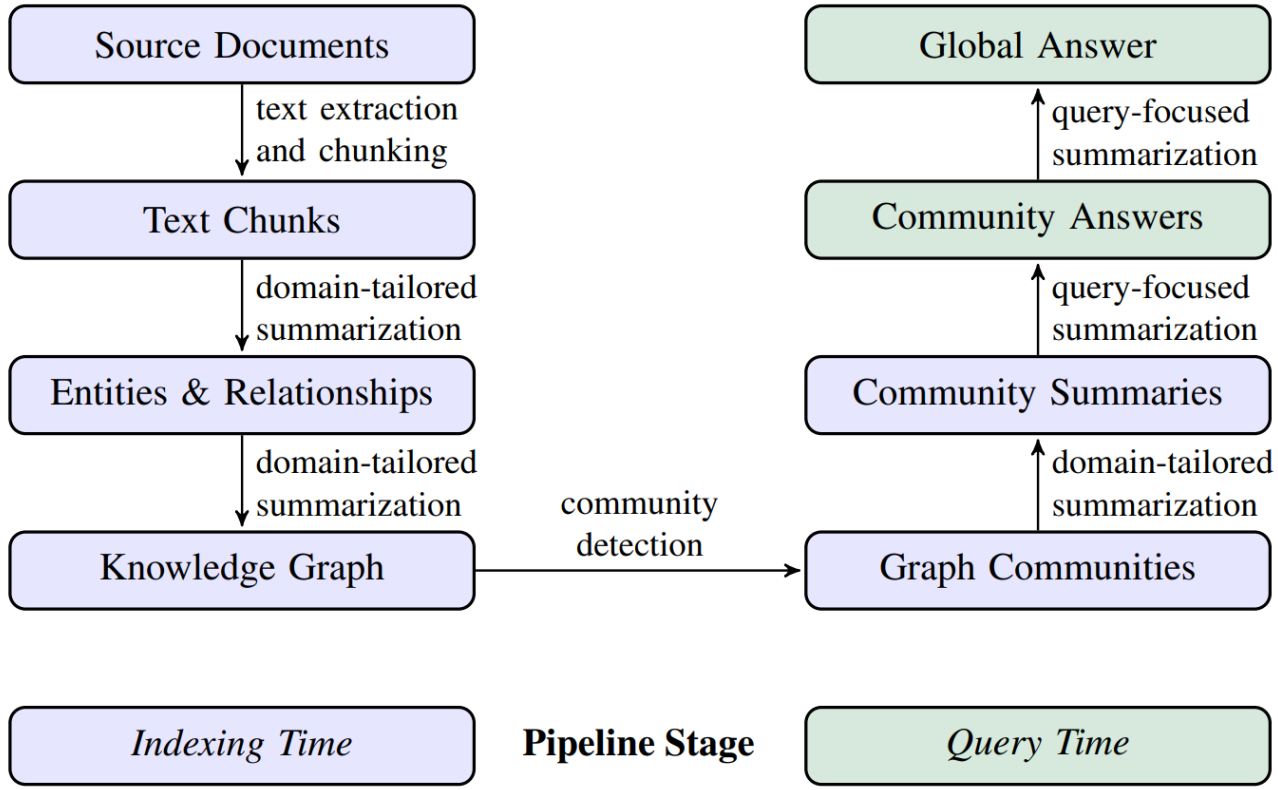


Figure 1: Graph RAG pipeline using an LLM-derived graph index of source document text. This graph index spans nodes (e.g., entities), edges (e.g., relationships), and covariates (e.g., claims) that have been detected, extracted, and summarized by LLM prompts tailored to the domain of the dataset. Community detection (e.g., Leiden, Traag et al., 2019) is used to partition the graph index into groups of elements (nodes, edges, covariates) that the LLM can summarize in parallel at both indexing time and query time. The "global answer" to a given query is produced using a final round of query-focused summarization over all community summaries reporting relevance to that query.

GraphRAG

- Unlike traditional RAG systems that rely on semantic similarity matching, GraphRAG constructs knowledge graphs from source documents and uses community detection algorithms to create thematic partitions, enabling comprehensive answers to complex, dataset-wide questions.

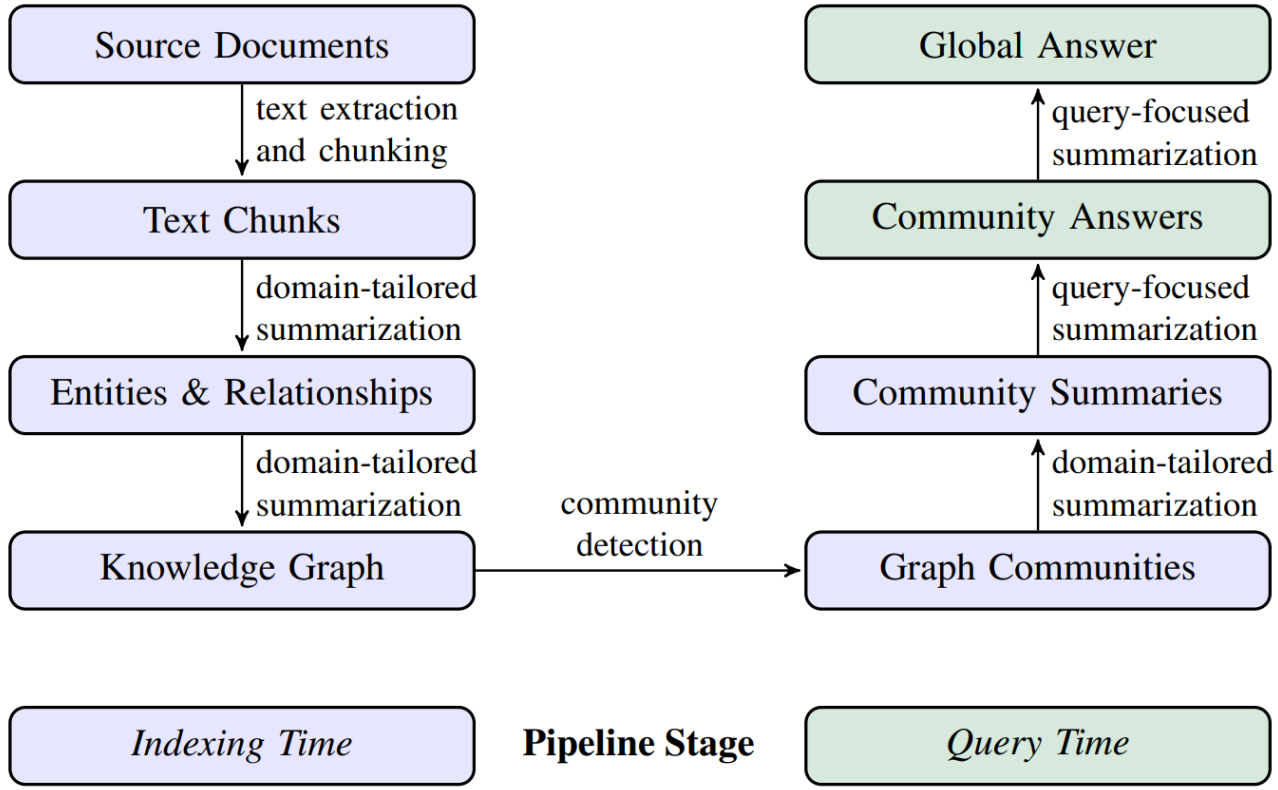


From Local to Global: A GraphRAG Approach to Query-Focused Summarization

GraphRAG

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- **Indexing Phase: Building the Knowledge Infrastructure**
  - The indexing process begins by chunking source documents into manageable segments (typically 600 tokens with 100-token overlaps).
  - Each chunk is then processed by an LLM to extract three types of information:

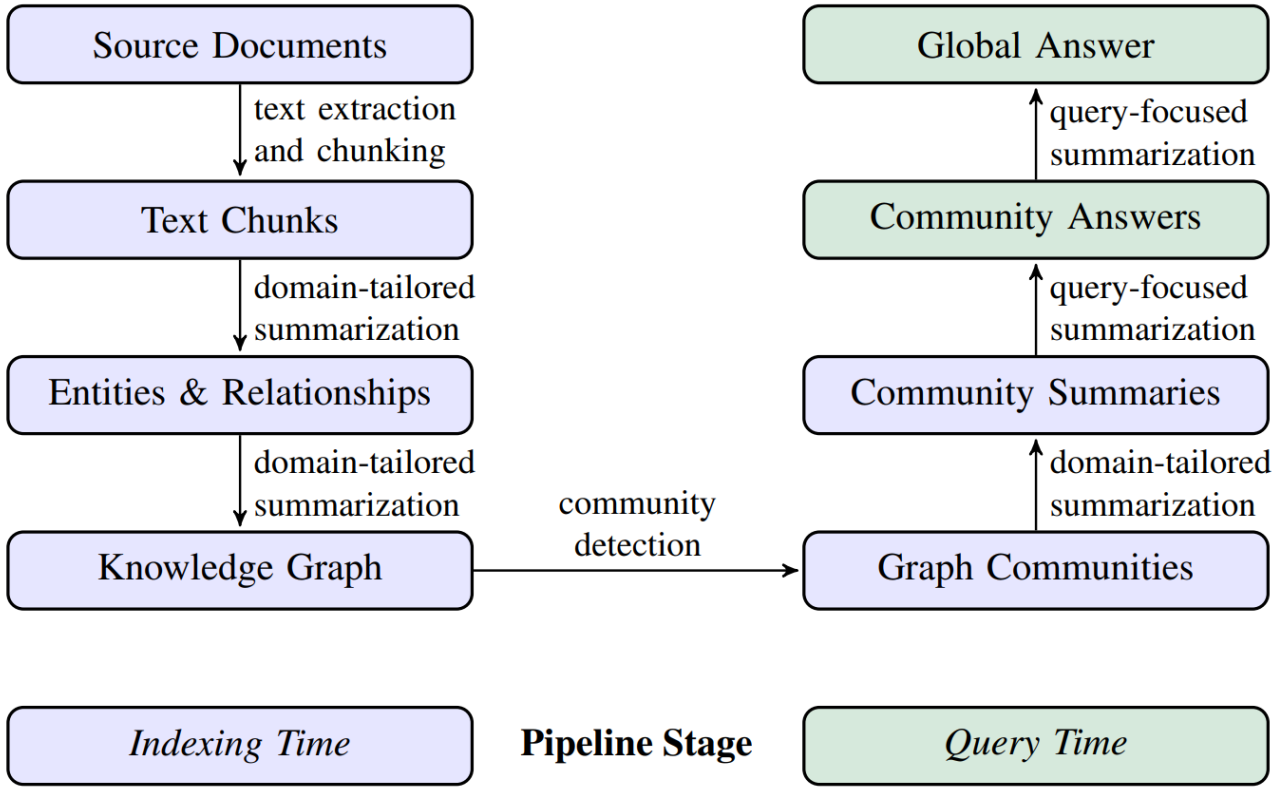


From Local to Global: A GraphRAG Approach to Query-Focused Summarization

GraphRAG

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

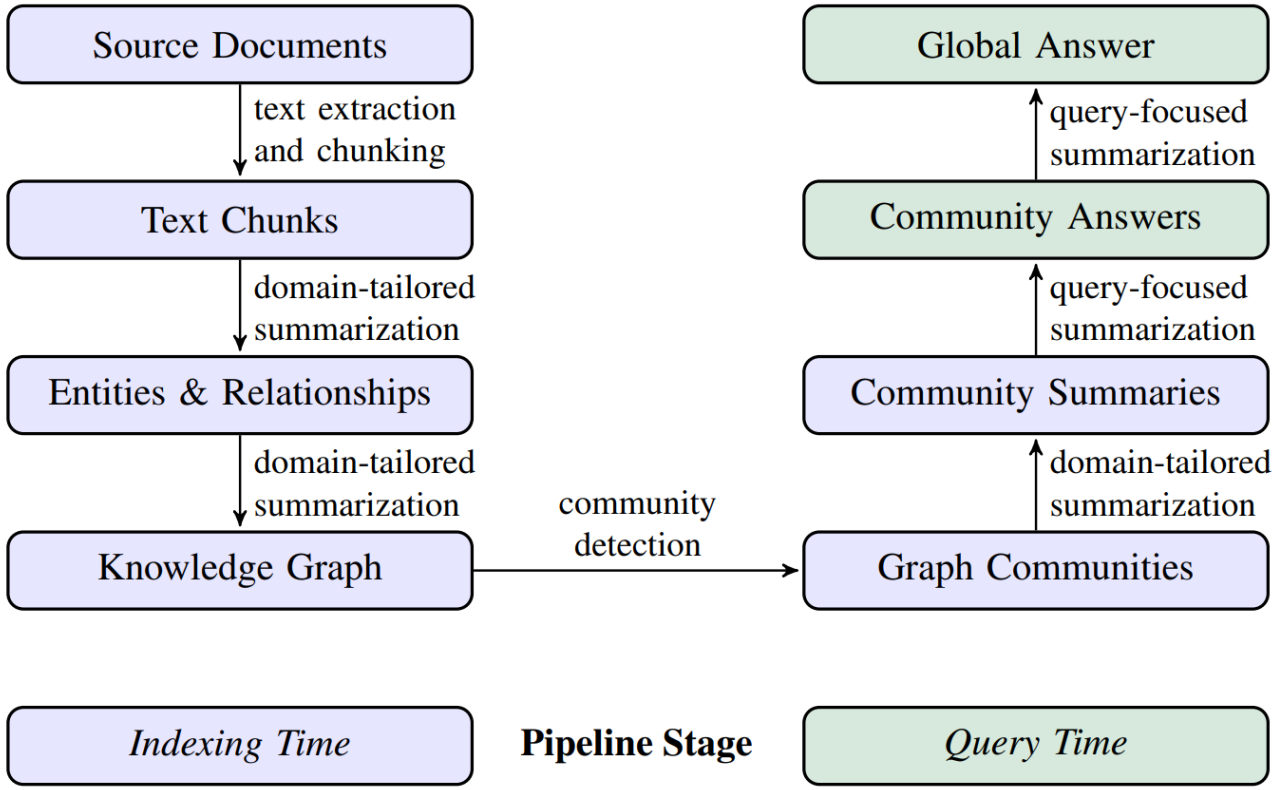
- **Indexing Phase: Building the Knowledge Infrastructure**
  - Step 1. Text Chunks
    - The indexing process begins by chunking source documents into manageable segments (typically 600 tokens with 100-token overlaps).
    - Each chunk is then processed by an LLM to extract three types of information



From Local to Global: A GraphRAG Approach to Query-Focused Summarization

GraphRAG

- **Indexing Phase: Building the Knowledge Infrastructure**
  - Step 2. Entities & Relationships
    - Entities: Named entities with descriptive summaries
    - Relationships: Connections between entities with strength scores
    - Claims: Important factual statements about entities



From Local to Global: A GraphRAG Approach to Query-Focused Summarization

GraphRAG

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

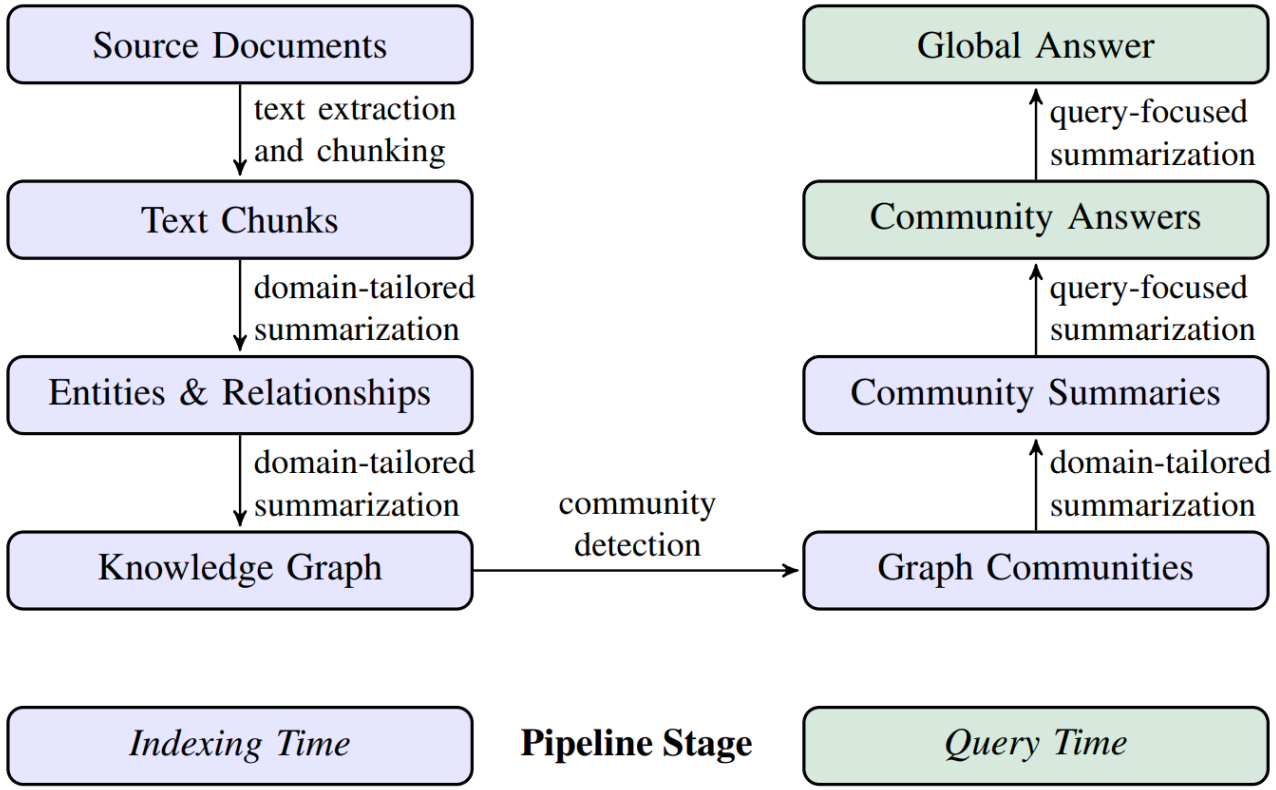
- **Indexing Phase: Building the Knowledge Infrastructure**
  - Step 3. Knowledge Graph
    - The extracted information is aggregated into a comprehensive knowledge graph, where entity instances are reconciled into unique nodes and relationship instances become weighted edges.
  - Step 4. Graph Communities
    - The system then applies hierarchical community detection using the Leiden algorithm to partition the graph into thematically coherent clusters.
  - Step 5. Community Summaries
    - For each community at each hierarchical level, LLMs generate concise summaries following a bottom-up approach. Leaf-level communities are summarized from their constituent elements, while higher-level summaries are generated from sub-community summaries, ensuring scalability.

From Local to Global: A GraphRAG Approach to Query-Focused Summarization

GraphRAG

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- **Query Phase: Map-Reduce Answer Generation**
  - Map-reduce strategy using the pre-generated community summaries.
    - Selects relevant community summaries from a chosen hierarchical level
    - Generates partial answers for each summary chunk in parallel
    - Scores each partial answer for helpfulness (0-100 scale)
    - Combines the most helpful partial answers to produce a comprehensive global response



From Local to Global: A GraphRAG Approach to Query-Focused Summarization

RAG Enhancement Methods

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Retrieval-enhanced LLMs
  - Self-RAG
  - GraphRAG
- RAG Instruction Tuning
  - REPLUG
  - RA-DIT
- RAG Optimization
  - FILCO
  - Adaptive Retrieval
- Embedding Model
  - Qwen3-embedding
- RL
  - **Search-R1**

Search-R1

- From Local to Global: A GraphRAG Approach to Query-Focused Summarization

  - Search-R1 introduces a reinforcement learning framework that trains LLMs to autonomously perform multi-turn reasoning while dynamically interacting with search engines.
  - The work enables LLMs to learn when and how to search for external information during complex reasoning tasks, rather than relying on predetermined retrieval strategies or fixed prompting patterns.

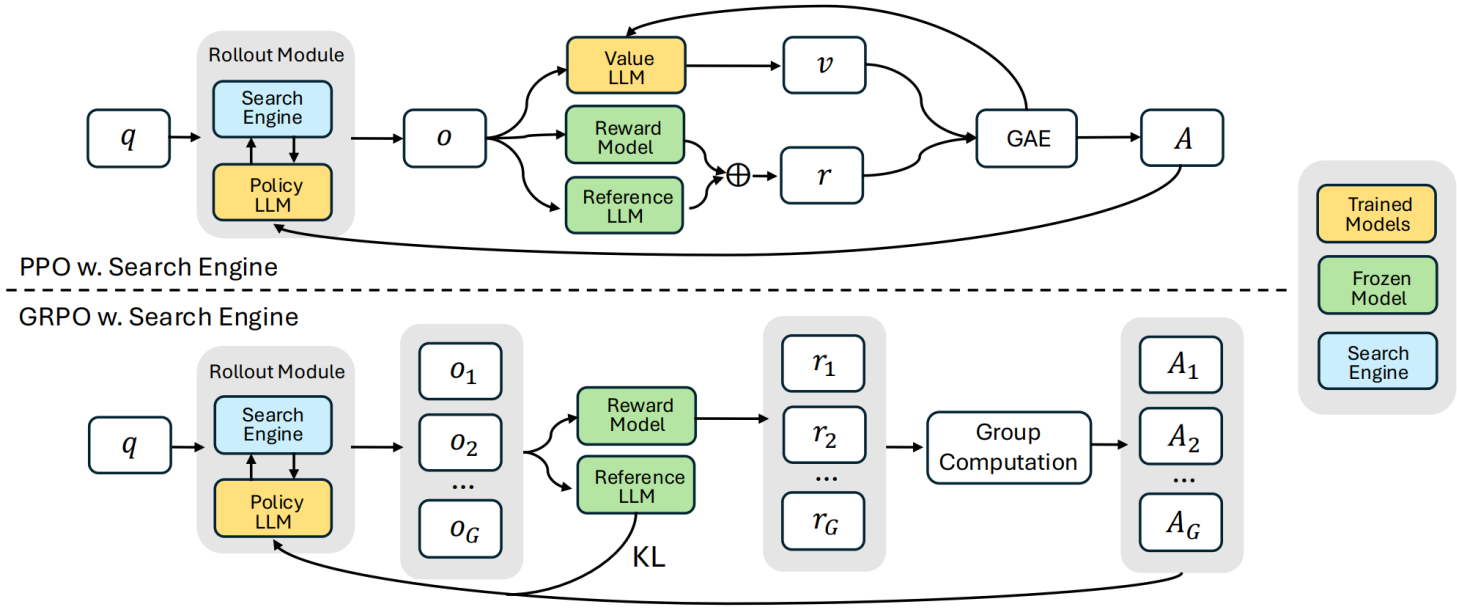


Figure 1: Demonstration of PPO and GRPO training with the search engine (SEARCH-R1). During the rollout, LLMs can conduct multi-turn interactions with the search engine.

Search-R1

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Reinforcement Learning Framework with Search Integration
  - $$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x;\mathcal{R})} \left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x;\mathcal{R}) \,||\, \pi_{\mathrm{ref}}(y \mid x;\mathcal{R})\right],$$

    where π_θ is the policy LLM, π_{ref} is the reference LLM, r_ϕ is the reward function and $\mathbb{D}_{\mathrm{KL}}$ is KL-divergence measure. x denote input samples drawn from the dataset $\mathcal{D}$, and y represent the generated outputs interleaved with search engine calling results, sampled from the reference policy $\pi_{\mathrm{ref}}(y \mid x)$ and retrieved from the search engine $\mathcal{R}$. Unlike prior RL

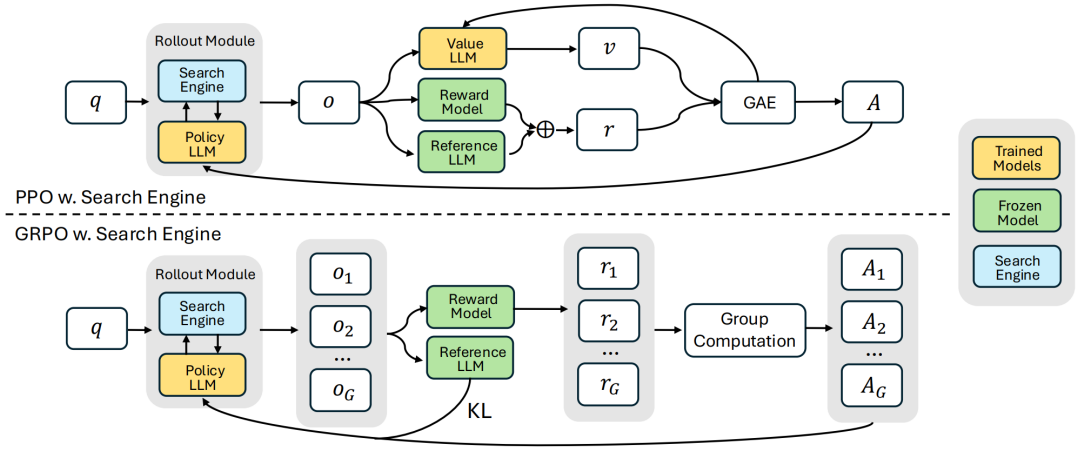


Figure 1: Demonstration of PPO and GRPO training with the search engine (SEARCH-R1). During the rollout, LLMs can conduct multi-turn interactions with the search engine.

From Local to Global: A GraphRAG Approach to Query-Focused Summarization

Search-R1

- Loss Masking for Retrieved Tokens
  - During interaction, the LLM's output includes both self-generated tokens and externally retrieved passages inserted by the search engine.
  - Loss masking ensures that policy gradients are computed only over LLM-generated tokens, excluding retrieved passages.
  - This focuses optimization on the LLM's decision-making capabilities while preventing interference from external content, resulting in more stable training and improved performance.

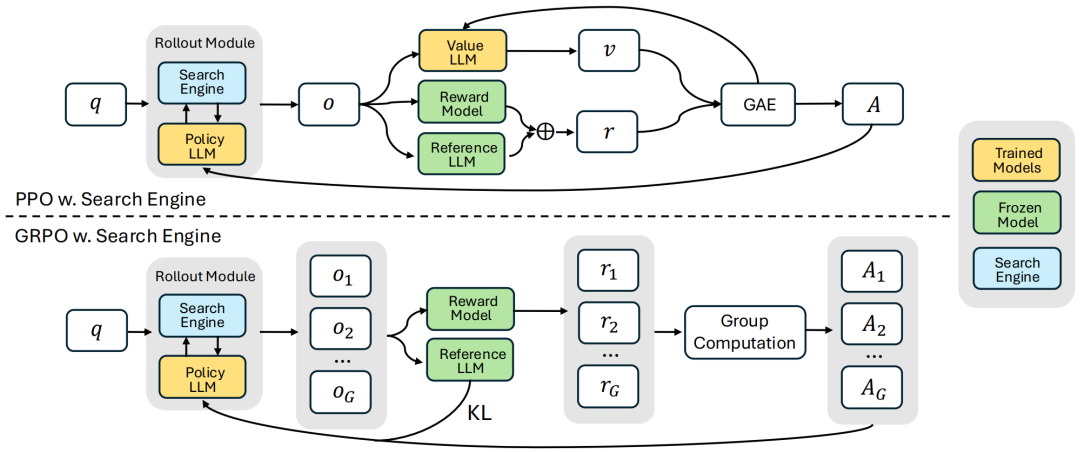


Figure 1: Demonstration of PPO and GRPO training with the search engine (SEARCH-R1). During the rollout, LLMs can conduct multi-turn interactions with the search engine.

From Local to Global: A GraphRAG Approach to Query-Focused Summarization

Search-R1

- Multi-Turn Interaction Protocol
  - The LLM interacts with the search engine through structured special tokens:
    - <search> query </search> for explicit search calls
    - <information> ... </information> for retrieved content insertion
    - <think> ... </think> for internal reasoning
    - <answer> ... </answer> for final responses
  - This protocol enables iterative reasoning where the LLM autonomously decides when to search, what to query, and how to integrate retrieved information into its reasoning process.

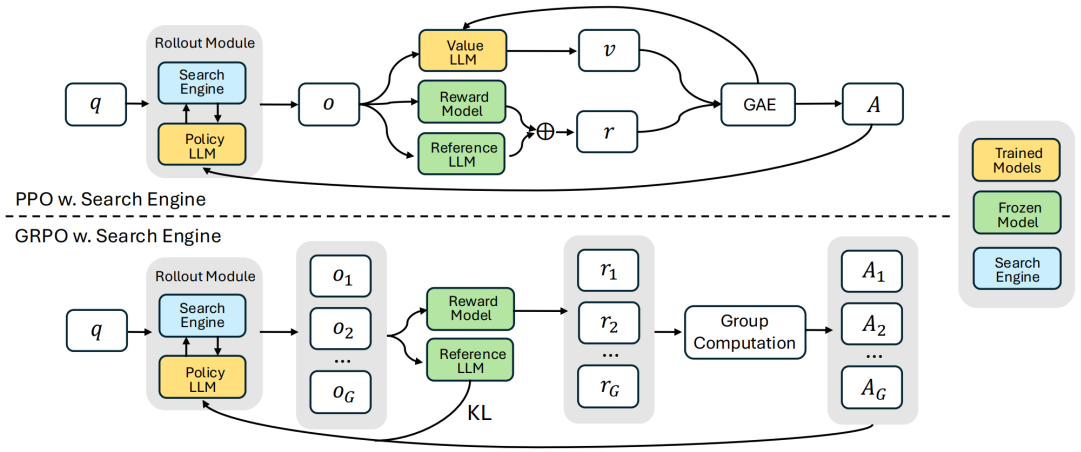


Figure 1: Demonstration of PPO and GRPO training with the search engine (SEARCH-R1). During the rollout, LLMs can conduct multi-turn interactions with the search engine.

From Local to Global: A GraphRAG Approach to Query-Focused Summarization

Search-R1

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Reward Design
  - Search-R1 employs a straightforward outcome-based reward function using Exact Match (EM) for factual question-answering tasks:

$$r_\phi(x, y) = \mathrm{EM}(a_{\text{pred}}, a_{\text{gold}})$$

  - This binary reward (1 for correct, 0 for incorrect) tests whether simple outcomes can drive complex learned behaviors without requiring intricate process-based rewards or neural reward models.

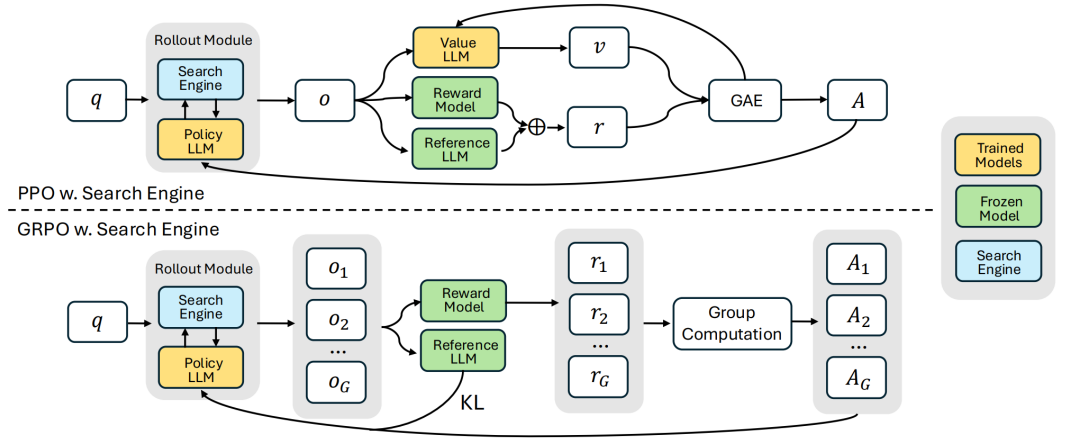


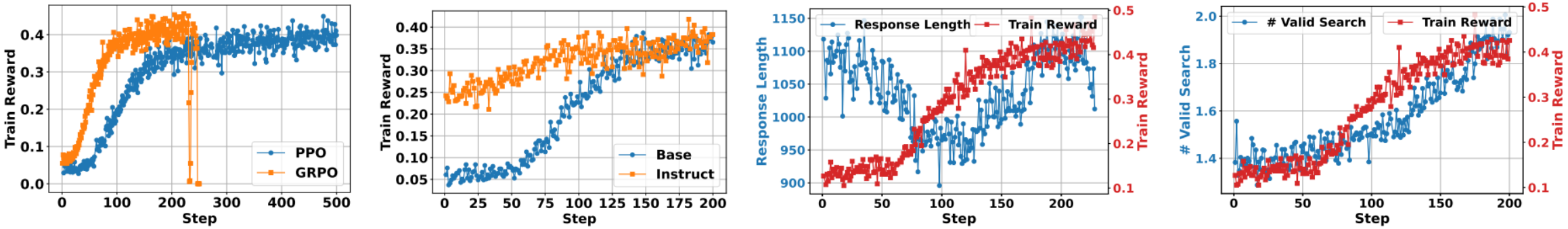
Figure 1: Demonstration of PPO and GRPO training with the search engine (SEARCH-R1). During the rollout, LLMs can conduct multi-turn interactions with the search engine.

From Local to Global: A GraphRAG Approach to Query-Focused Summarization

Search-R1

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Training Dynamics and Optimization
  - The learning process exhibits distinct phases. Initially, response length decreases as the LLM eliminates unnecessary content, with modest reward improvements. Subsequently, both response length and training reward increase substantially as the LLM learns to frequently call the search engine and effectively leverage retrieved information.



(a) PPO vs. GRPO (b) Base vs. Instruct (c) Response length (d) # Valid search

From Local to Global: A GraphRAG Approach to Query-Focused Summarization

Thank you