中国科学院大学
University of Chinese Academy of Sciences

自然语言处理

第十一讲 检索增强生成-RAG

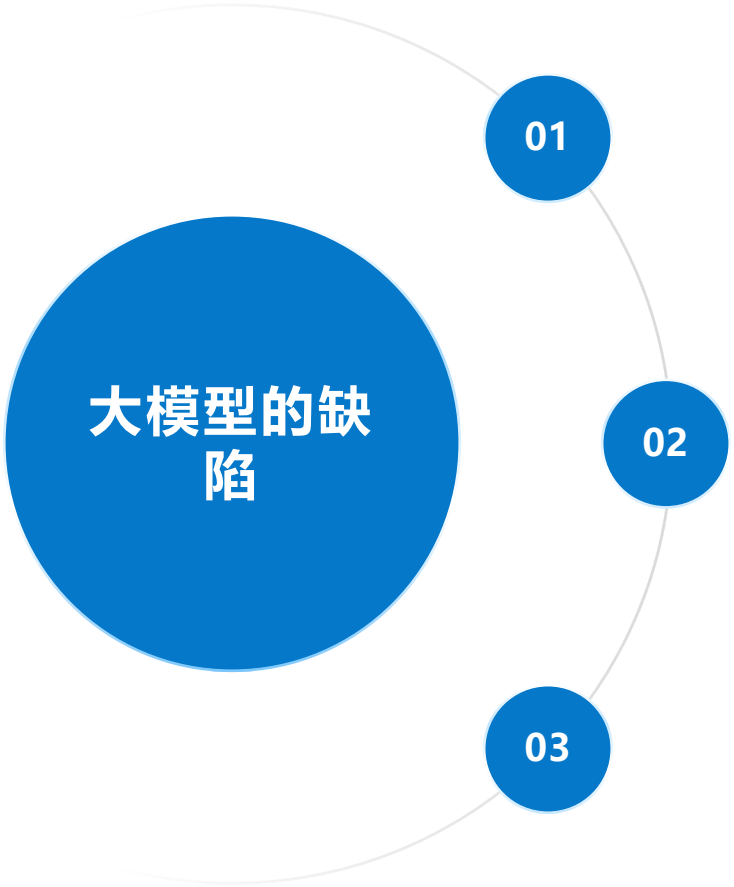
王 石 资康莉 刘 瑜

2026年春季课程

https://ictkc.github.io/teaching/

第十一讲
检索增强生成

背景



幻觉

由于大模型主要依靠参数中的统计模式生成答案，在缺少外部事实依据时，可能会编造不存在的概念、数据、出处或结论。幻觉问题使模型难以直接承担严肃业务中的事实问答和决策支持任务。

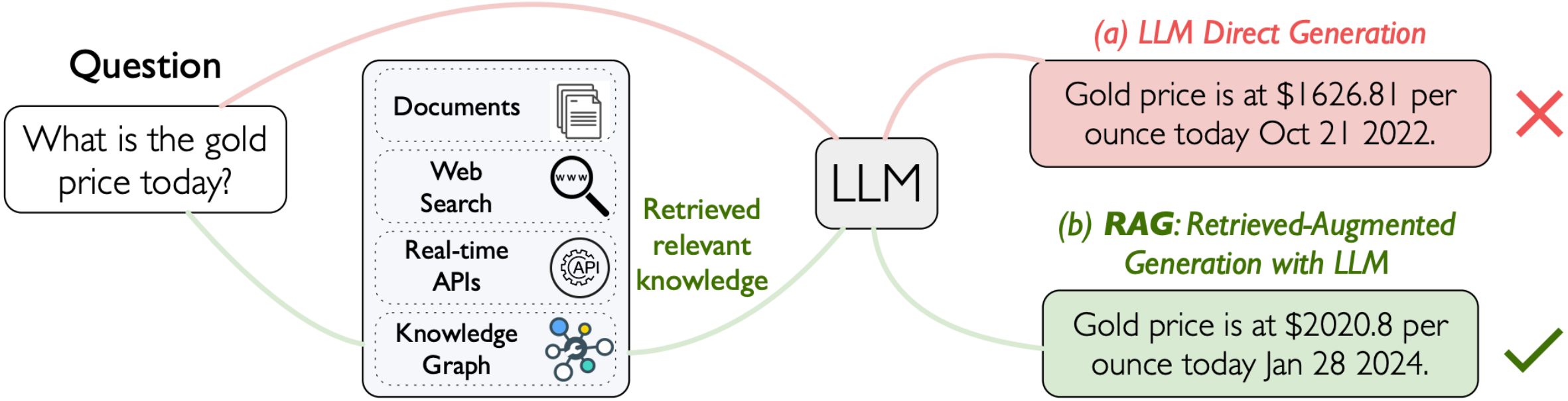
过旧的知识

大模型的知识存在训练截止时间，无法天然感知最新发生的事实变化，例如政策调整、产品更新、市场价格、企业制度变更和实时业务状态等。

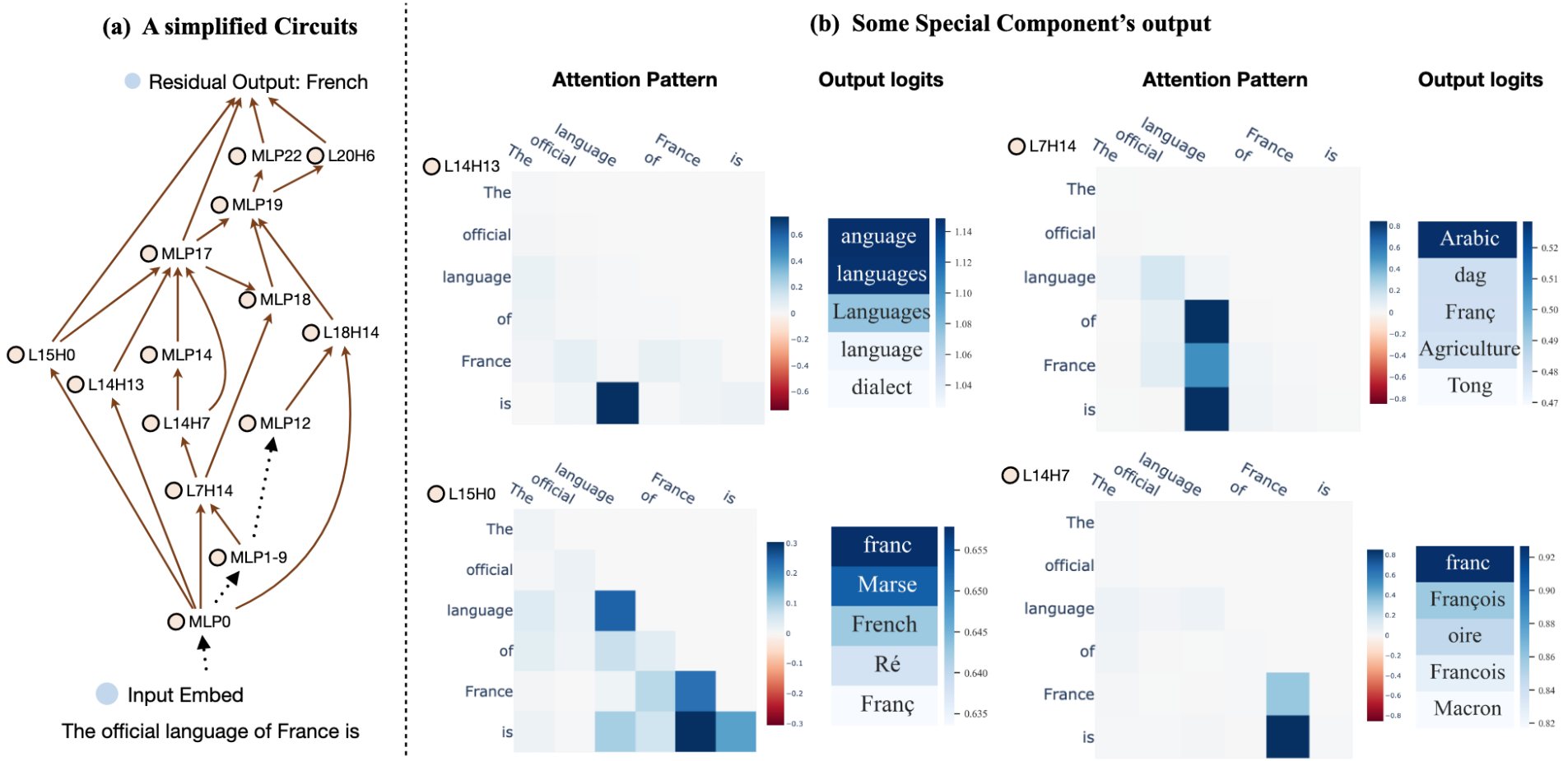
缺乏特定领域的专业知识

通用大模型通常缺少企业私有数据和垂直行业知识，对内部流程、专业术语、业务规则、项目经验和历史案例了解不足。

幻觉问题


Question
What is the gold price today?
Documents
Web Search
Real-time APIs
Knowledge Graph
Retrieved relevant knowledge
LLM
(a) LLM Direct Generation
Gold price is at $1626.81 per ounce today Oct 21 2022.
(b) RAG: Retrieved-Augmented Generation with LLM
Gold price is at $2020.8 per ounce today Jan 28 2024.

大模型的知识来自哪里



Knowledge Circuits in Pretrained Transformers

参数记忆的局限性

知识过时

训练完成后内部知识固化，无法回答动态变化的问题。

私有知识缺失

私域数据不会公开发布到互联网，通用大模型在专业领域存在知识盲点

难以溯源

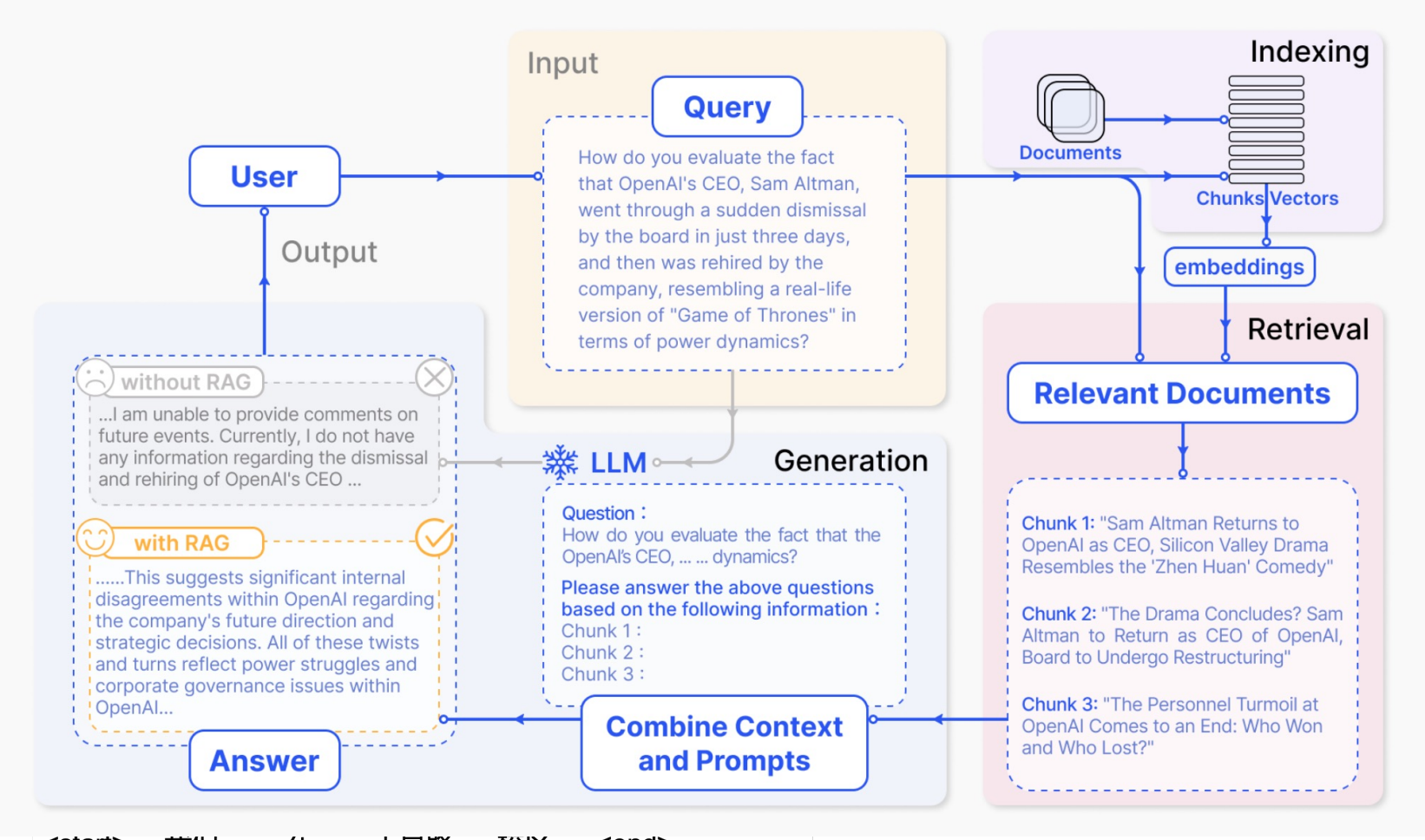
模型答案依赖海量训练中的知识共现，无法追溯回答的依据

更新成本高

通过SFT把频繁变化的知识写进参数，通常不如更新知识库方便。

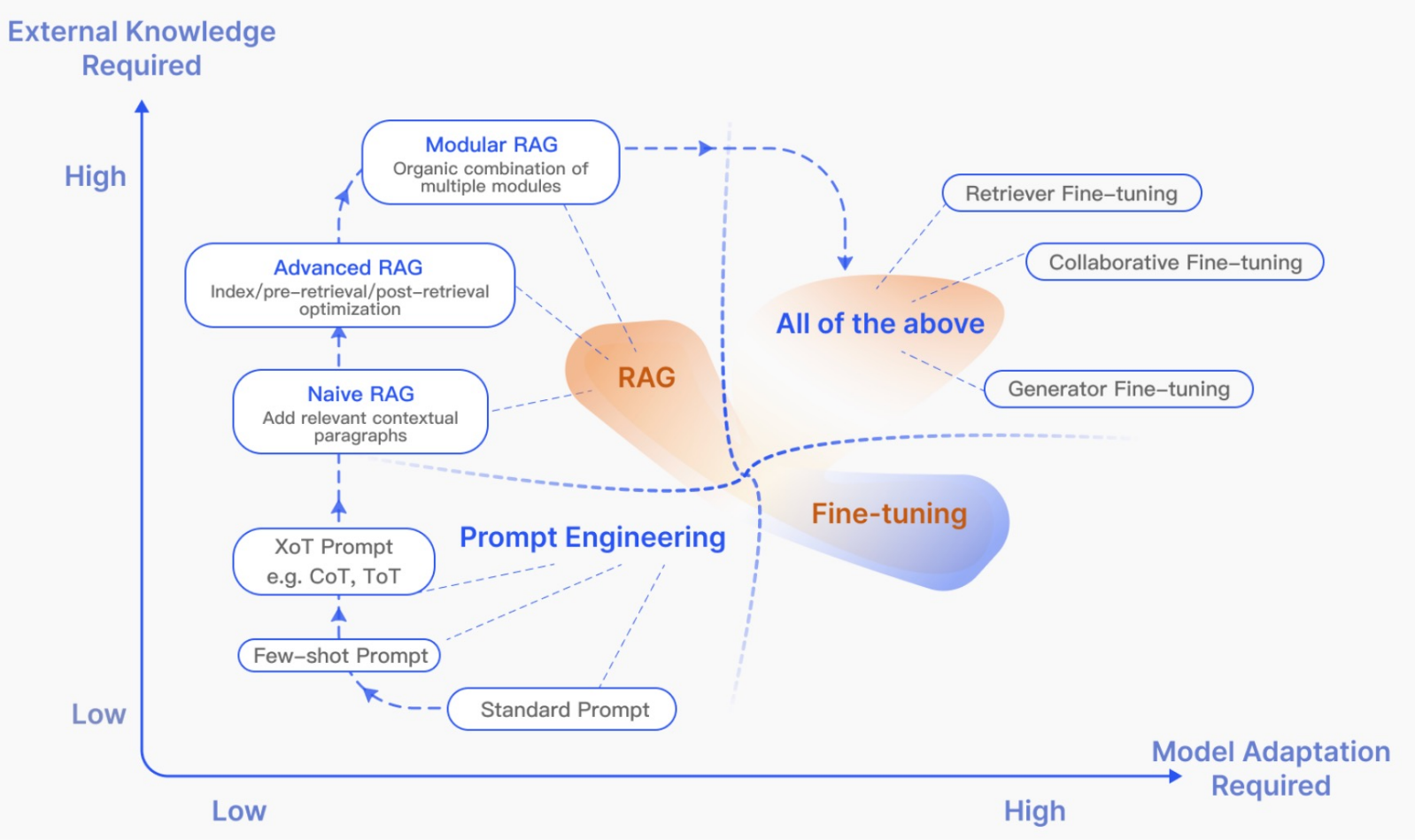
关键缺口 **新知识、私有知识、来源引用、实时更新。**

检索增强生成



在回答问题或生成文本时，RAG 会先从大量文档中检索与问题相关的信息，然后由大语言模型基于这些信息生成答案。通过接入外部知识库，模型无需针对每一个具体任务重新训练整个大模型。RAG 尤其适用于知识密集型任务。

RAG-微调-提示词



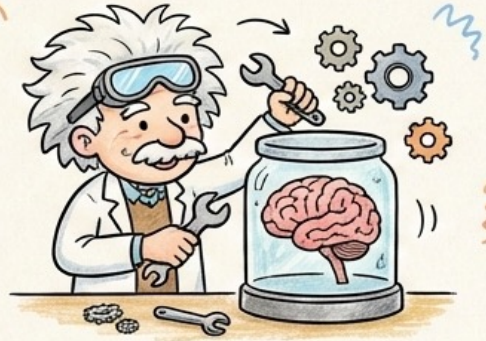
<start> 我们 在 玉泉路 校区 <end>

大模型的增强方式

- 提示词工程
- 微调
- 检索增强生成

闭卷考试VS开卷考试

Fine-tuning（微调）
—— 定制大脑



✓ 效果稳定、风格可控

✗ 成本高、更新慢

适用场景：固定领域任务

Prompt（提示词）
—— 考场小抄

✓ 快速、零成本

✗ 上下文窗口限制

适用场景：简单问答

RAG（检索增强生成）
—— 开卷考试

✓ 实时、安全、可扩展

✳ 检索质量依赖数据

VS

适用场景：知识密集型场景

RAG的价值

知识可更新
资料变了，更新知识库比重训模型更直接。

接入私有知识
课程讲义、企业制度、个人笔记都可以成为知识源。

降低幻觉风险
模型基于检索证据回答，而不是完全凭记忆。

支持引用
答案可以指向文档、页码或 URL，便于检查。

降低更新成本
频繁变化的信息放在知识库里更灵活。

增强大模型应用的可解释性
大模型高风险高价值领域应用的可解释性

中国科学院大学
University of Chinese Academy of Sciences

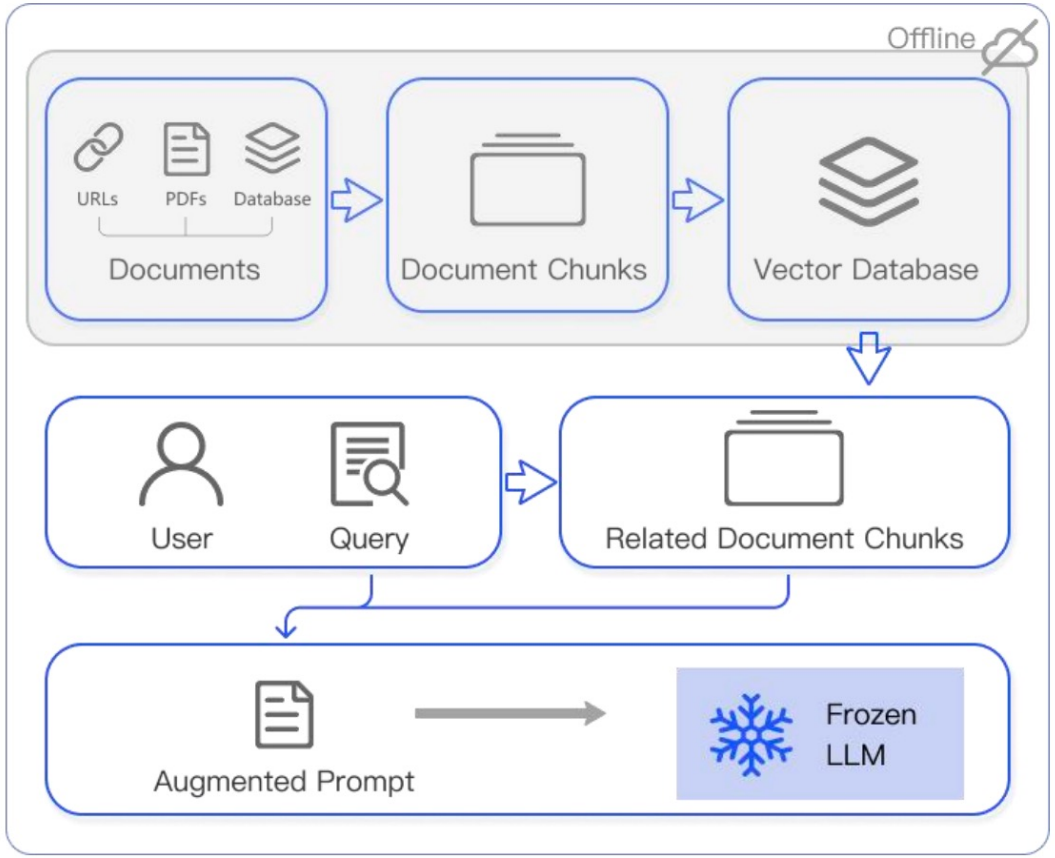
目 录

Naive RAG

索引 文档解析、切分、向量化，构建知识索引

召回 根据问题检索相关文档片段，补充上下文

生成 结合问题与检索结果，生成可靠答案

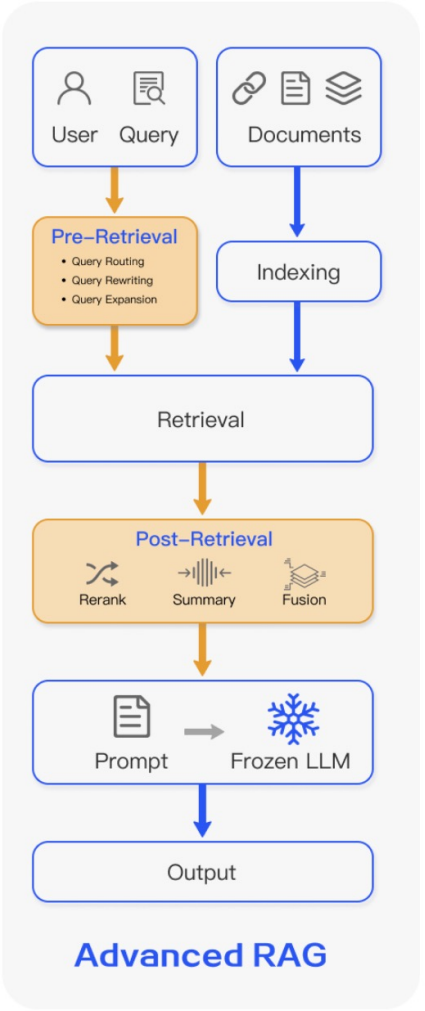


Advance RAG

数据索引优化 滑动窗口、细粒度分割、添加元数据

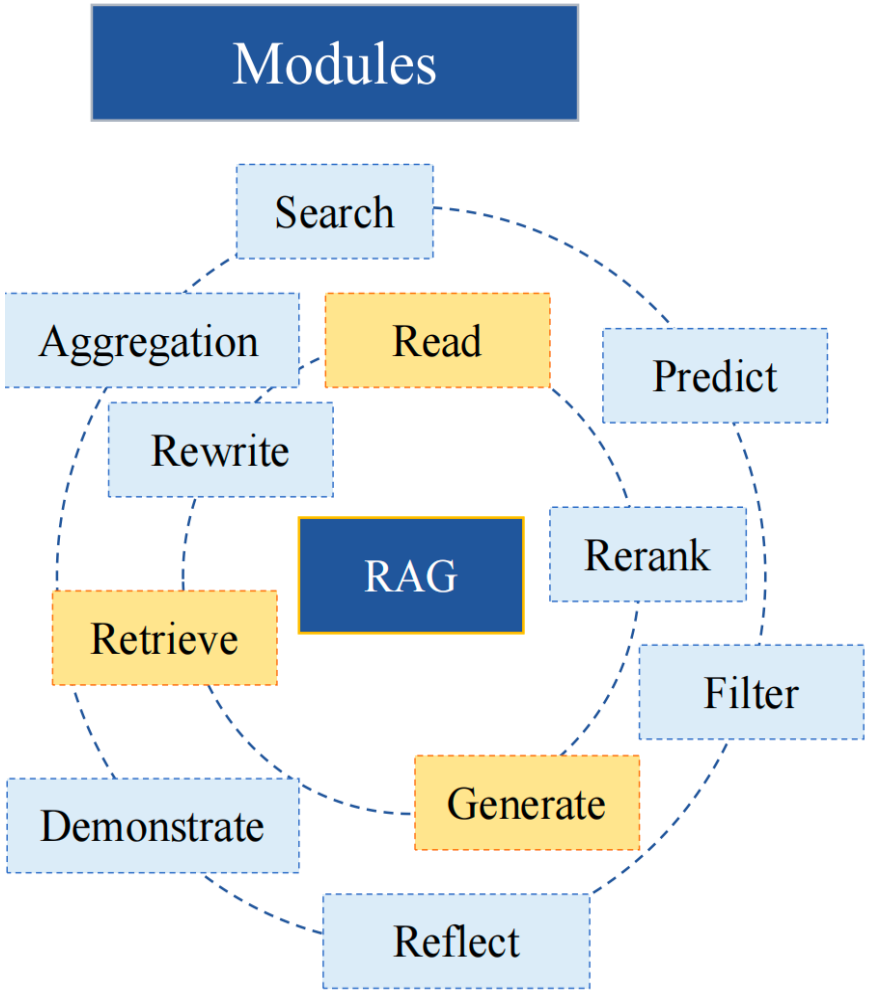
预召回处理 检索路径、摘要、重写及置信度判断

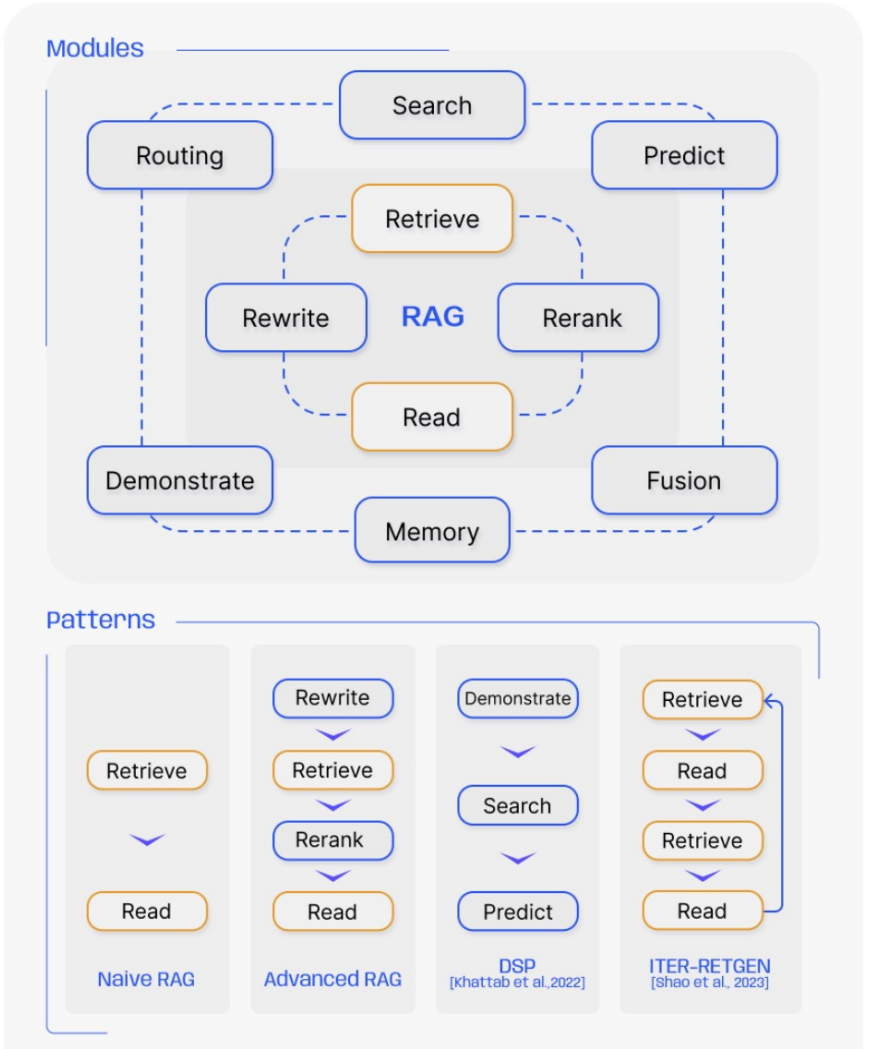
后召回处理 重排、过滤



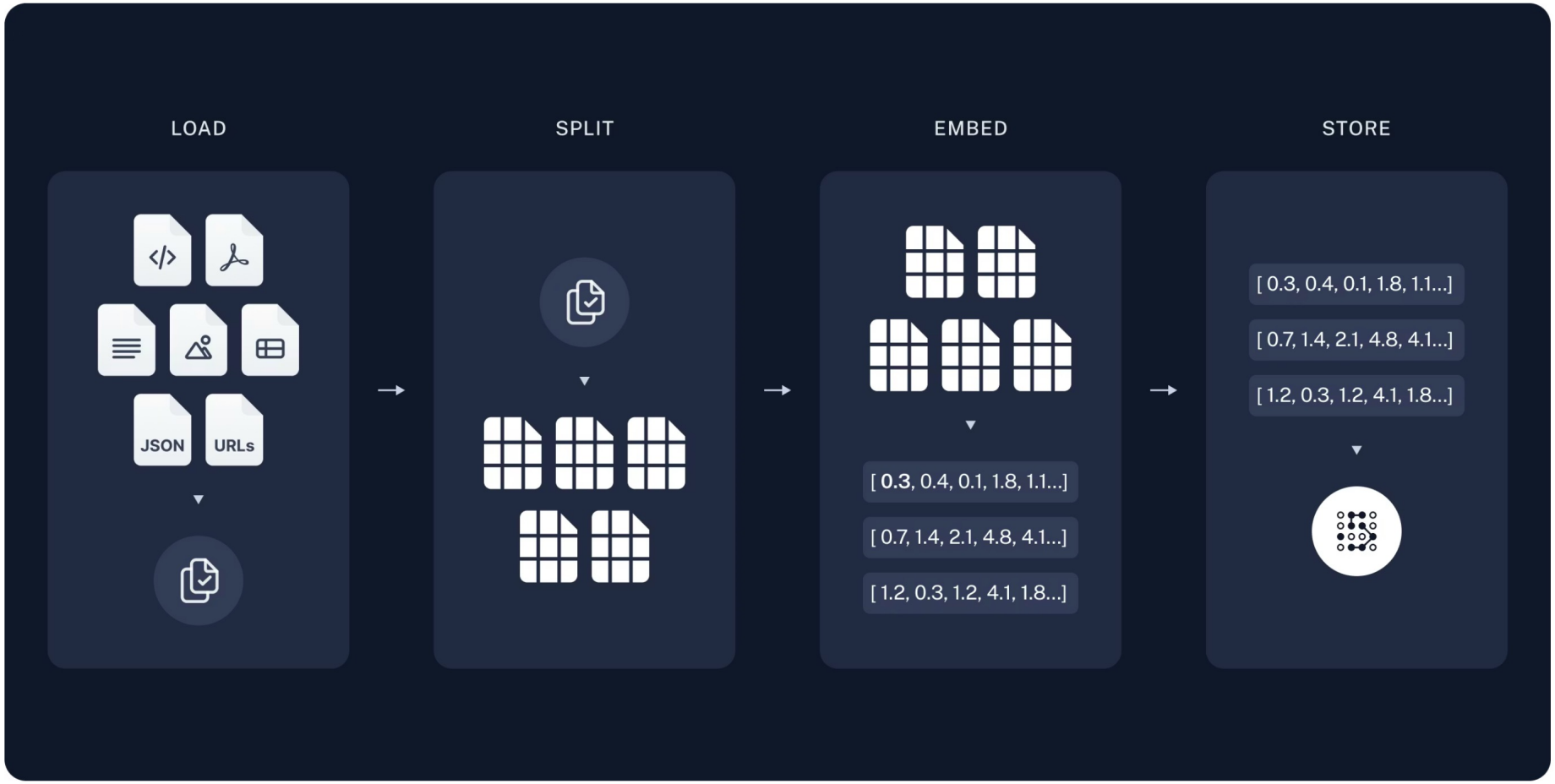
Modular RAG

模块化的 RAG 架构，将索引、检索、重排、路由、压缩、生成、验证等能力拆成独立模块，并通过灵活编排来提升 RAG 系统的准确性、可控性和适应性。

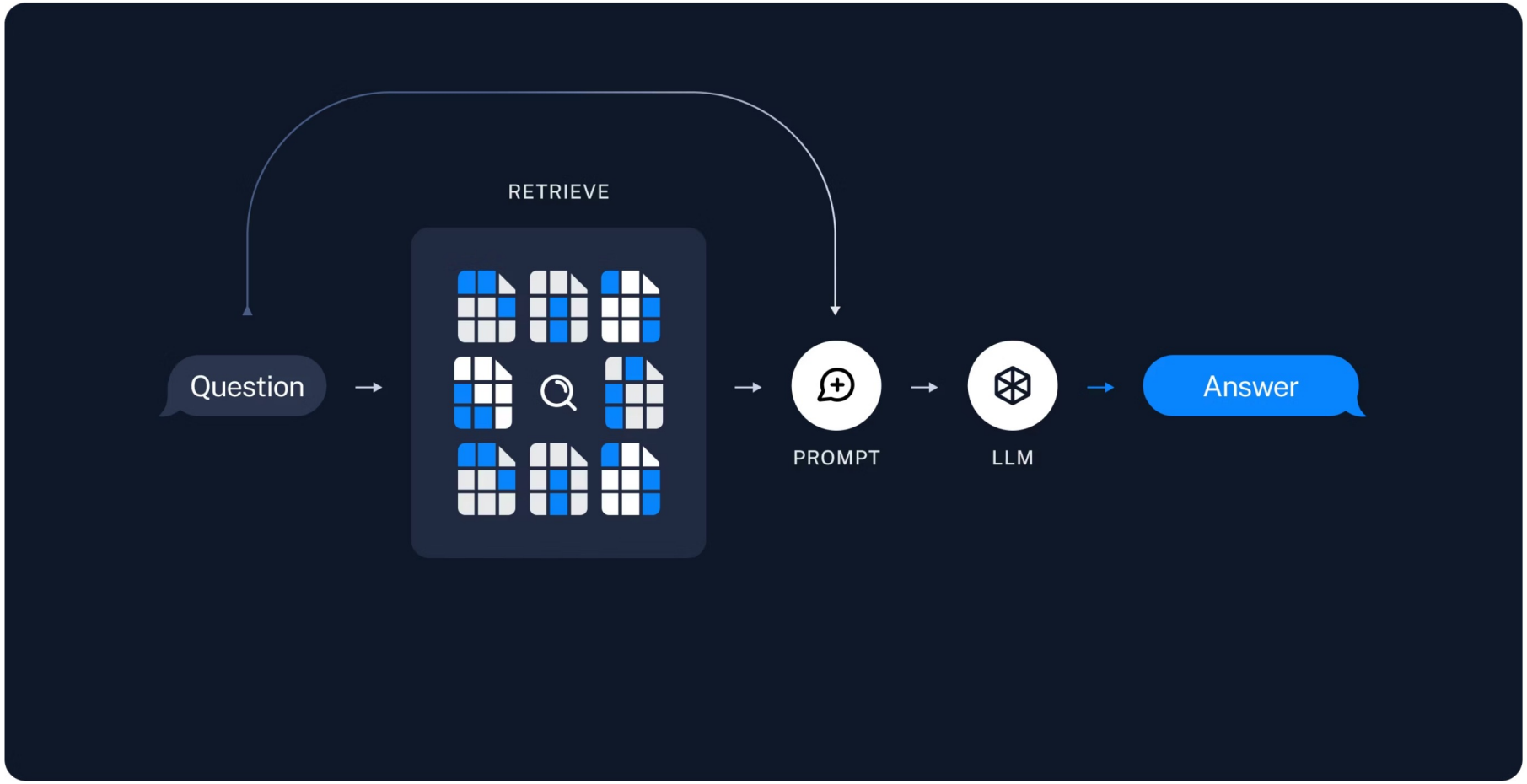




Part1:离线索引


LOAD
SPLIT
EMBED
STORE
JSON
URLs
[0.3, 0.4, 0.1, 1.8, 1.1...]
[0.7, 1.4, 2.1, 4.8, 4.1...]
[1.2, 0.3, 1.2, 4.1, 1.8...]
[0.3, 0.4, 0.1, 1.8, 1.1...]
[0.7, 1.4, 2.1, 4.8, 4.1...]
[1.2, 0.3, 1.2, 4.1, 1.8...]

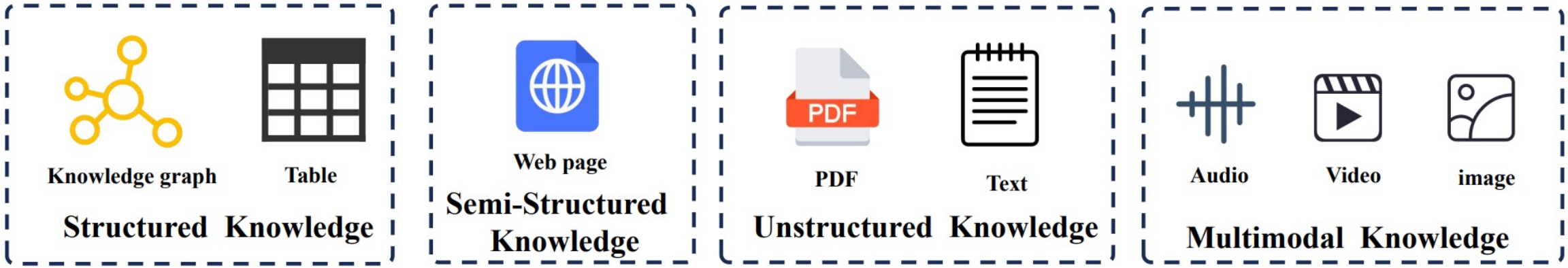
Part2：在线增强



中国科学院大学
University of Chinese Academy of Sciences

目 录

RAG数据源


Knowledge graph
Table
Structured Knowledge
Web page
Semi-Structured Knowledge
PDF
PDF
Text
Unstructured Knowledge
Audio
Video
image
Multimodal Knowledge

结构化数据

结构明确	数据遵循固定schema，例如表头、字段、主键、实体类型和关系类型
语义关系清晰	表格通过行列关系表达含义，知识图谱通过实体和边表达关系
适合精确查询	支持条件筛选、聚合统计、数值计算、实体查询和路径推理
可解释性较强	结果可以追溯到具体字段、单元格、实体、关系或子图

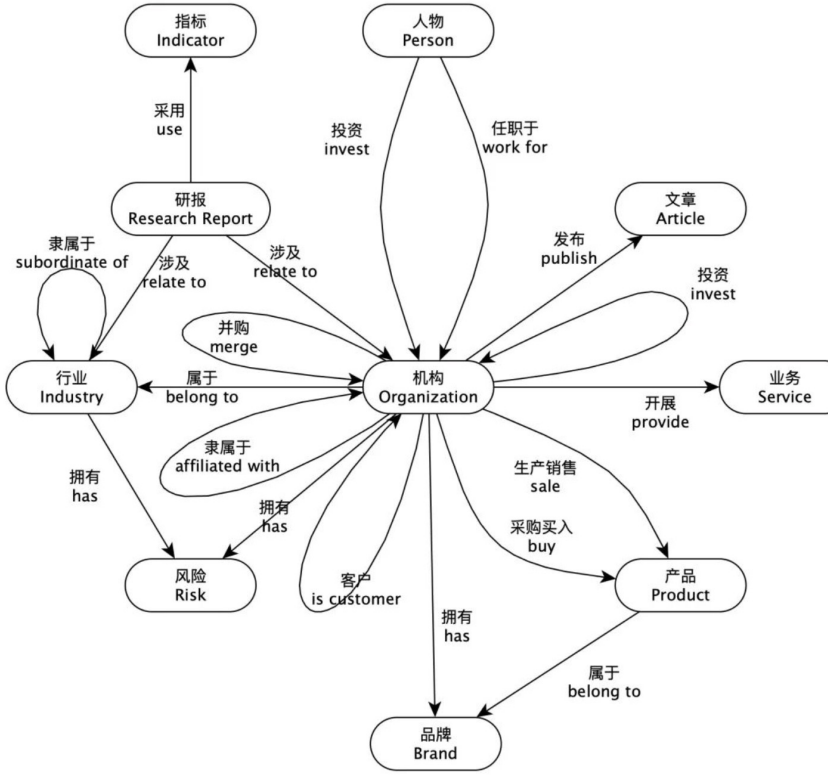


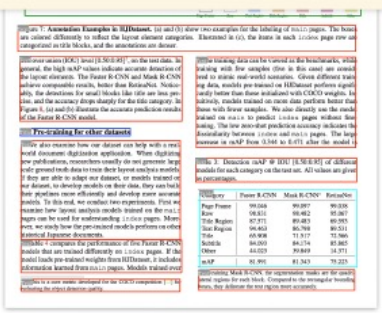
Table 6: Experimental results on WN18RR with different embedding dimensions and training batch sizes.

batch size	MR↓	MRR↑	Hit@1↑	Hit@3↑	Hit@10↑
d=100					
64	138	0.319	0.224	0.350	0.500
128	158	0.378	0.281	0.411	0.570
256	117	0.423	0.308	0.481	0.637
d=500					
64	151	0.313	0.207	0.353	0.518
128	145	0.403	0.304	0.452	0.588
256	123	0.452	0.351	0.501	0.650

半/非结构化数据

结构不固定	没有统一字段、行列或 schema，文档格式随来源和场景变化
内容形态复杂	同时包含正文、标题、表格、图片、公式、脚注、页眉页脚等
上下文依赖强	信息含义依赖章节层级、阅读顺序、页面位置和前后文关系

```
import layoutparser as lp
image = cv2.imread(...)
model = lp.Detectron2LayoutModel('lp://PrimaLayout/mask_rcnn_R_50_FPN_3x/config')
layout = model.detect(image)
```



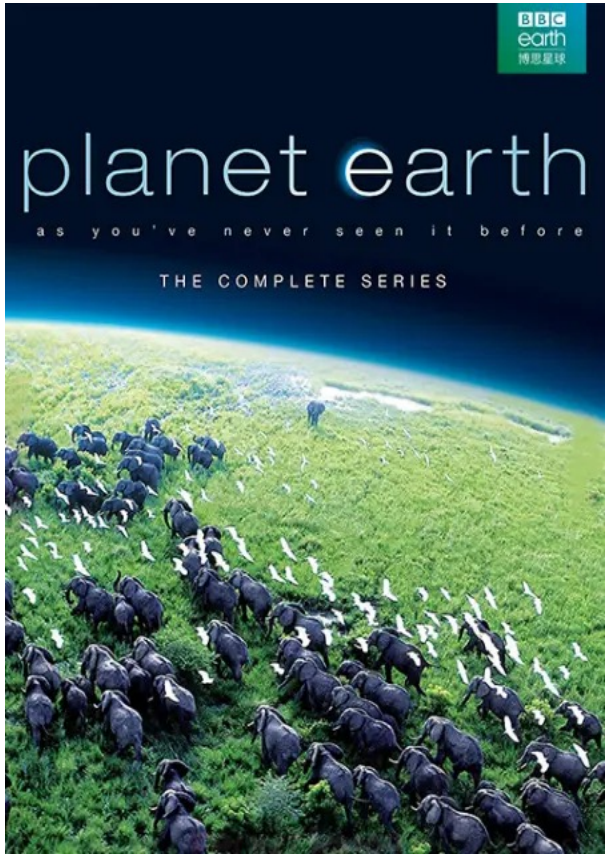
Paper with Complex Layouts

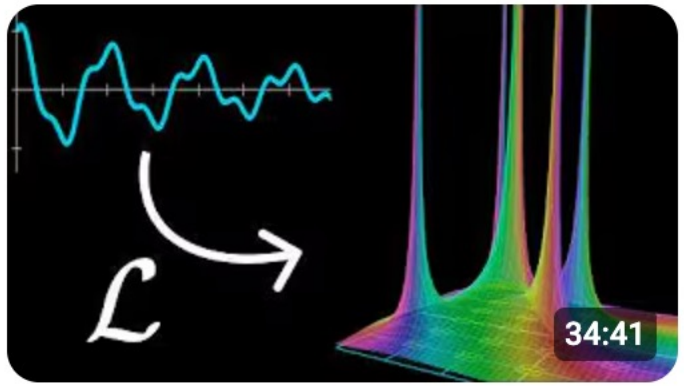
Magazine Scans & Websites

Historical Documents

多模态数据

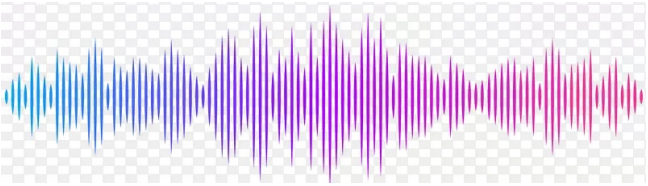
信息形态多样	包括图像、音频、视频等，不再局限于纯文本
语义更丰富	能表达场景、物体、声音、动作、情绪和时间变化
解析门槛更高	需要 OCR、ASR、视觉理解、视频分段等预处理
检索方式更复杂	可使用文本检索、图像检索、音频检索、视频检索或跨模态检索



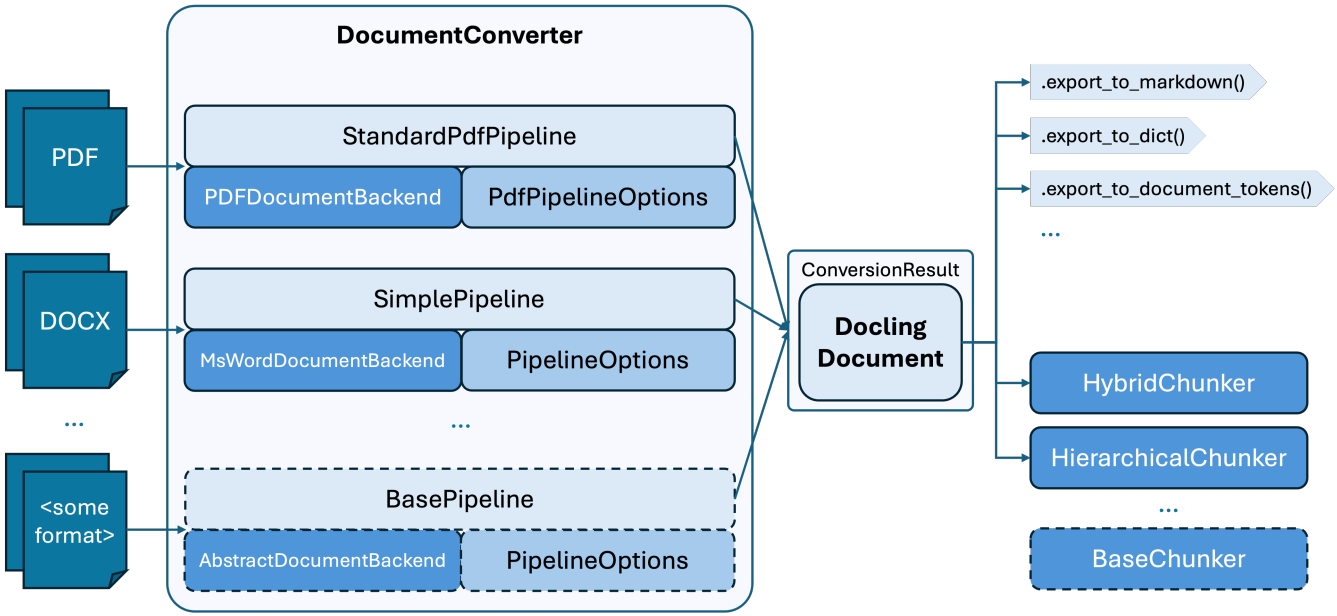


But what is a Laplace Transform?

156万次观看 · 6个月前



文档解析：从文件到文本

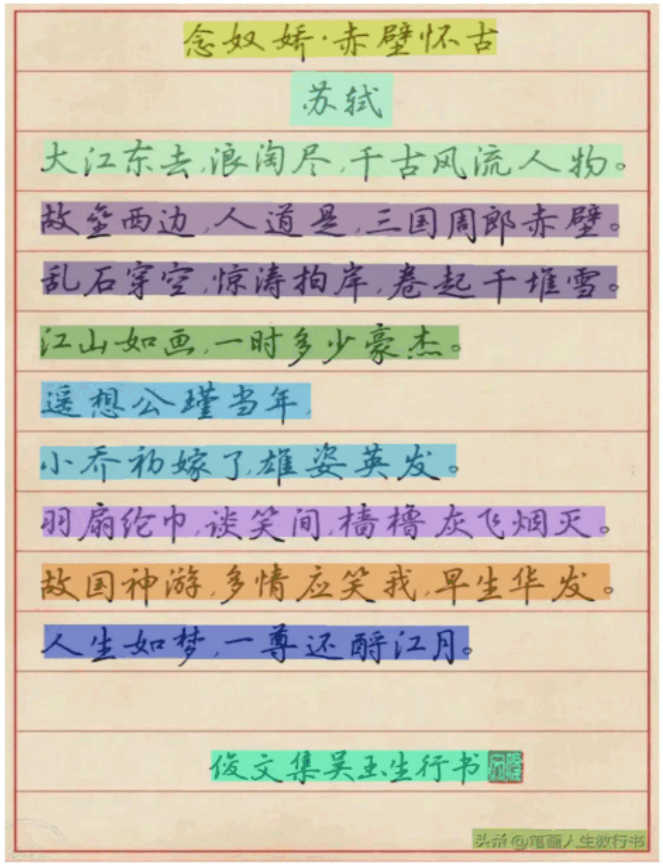

DocumentConverter
PDF
DOCX
...
<some format>
StandardPdfPipeline
PDFDocumentBackend
PdfPipelineOptions
SimplePipeline
MsWordDocumentBackend
PipelineOptions
...
BasePipeline
AbstractDocumentBackend
PipelineOptions
ConversionResult
Docling Document
.export_to_markdown()
.export_to_dict()
.export_to_document_tokens()
...
HybridChunker
HierarchicalChunker
...
BaseChunker

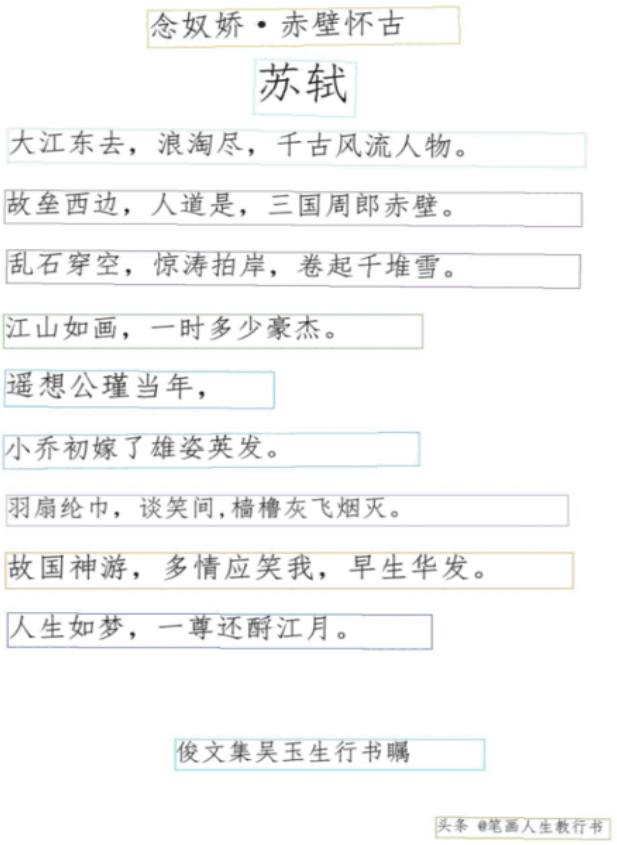
文档解析：从文件到文本

工具名称	特点	适用场景	性能表现
PyMuPDF4LLM	PDF→Markdown转换，OCR+表格识别	科研文献、技术手册	开源免费，GPU加速
TextLoader	基础文本文件加载	纯文本处理	轻量高效
DirectoryLoader	批量目录文件处理	混合格式文档库	支持多格式扩展
Unstructured	多格式文档解析	PDF、Word、HTML等	统一接口，智能解析
FireCrawlLoader	网页内容抓取	在线文档、新闻	实时内容获取
LlamaParse	深度PDF结构解析	法律合同、学术论文	解析精度高，商业API
Docling	模块化企业级解析	企业合同、报告	IBM生态兼容
Marker	PDF→Markdown，GPU加速	科研文献、书籍	专注PDF转换
MinerU	多模态集成解析	学术文献、财务报表	集成LayoutLMv3+YOLOv8

OCR

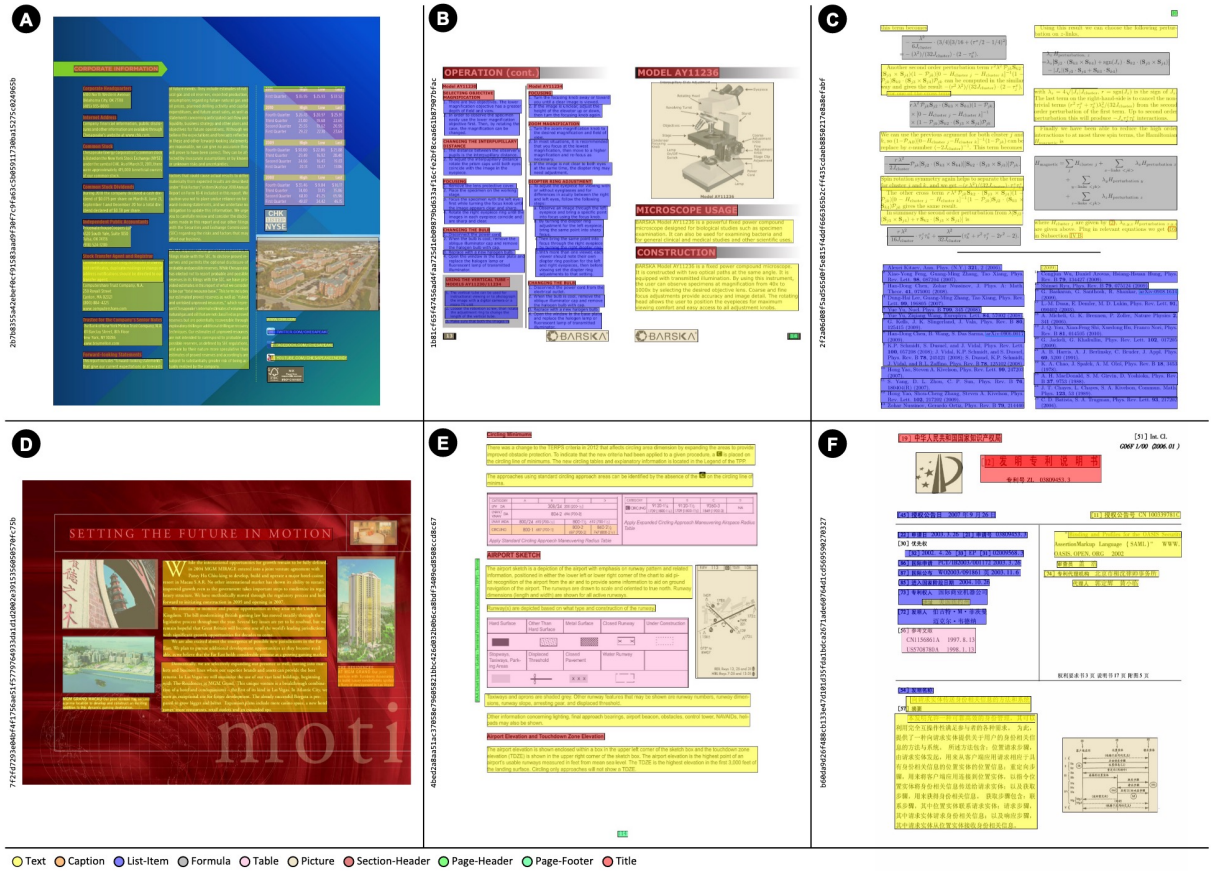
PP-OCRv5: 手写中文/Handwritten Chinese





OCR，即光学字符识别，用于从扫描件、图片、截图或手写文档中识别文字内容，并将其转换为机器可处理的文本。

版面结构识别



识别这些结构信息可以帮助 RAG 在文档解析阶段保留标题层级、段落边界、表格区域、图注和页码信息，从而提升 chunking、metadata 构建、引用定位和答案可信度。

表格解析

Table Detection

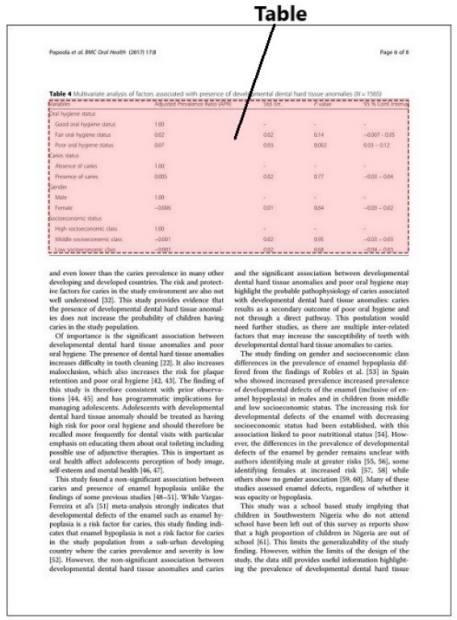


Table Structure Recognition

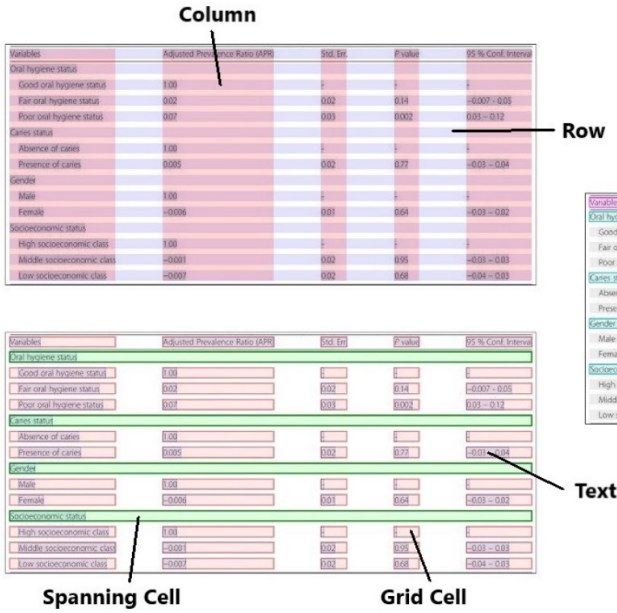
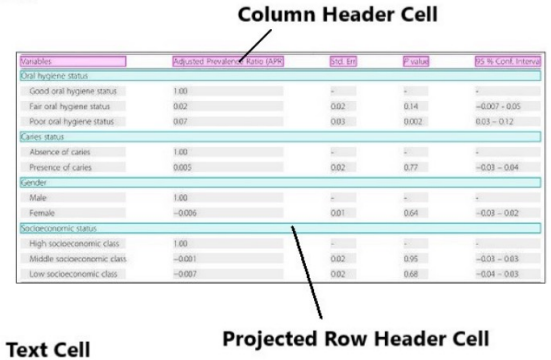


Table Functional Analysis



表头、行列关系、合并单元格和数据区域共同决定了单元格的真实含义。表格解析的目标是将 PDF 或图片中的视觉表格还原为结构化表示，使模型能够基于正确的行列语义进行检索、计算和回答。

Meta Data标记

TABLE I: Example of HDX Metadata.

Item	Content
Data name	Daily Summaries of Precipitation Indicators for Canada
Data summary	This dataset contains the daily summaries on base stations across Canada. The four indicators included are: TPCP: Total precipitation MXSD: Maximum snow depth TSNW: Total snow fall EMXP: Extreme maximum daily precipitation Indicators are compiled by the National Centers for Environmental Information (NCEI), which is administrated by National oceanic and Atmospheric Administration (NoAA) an organization part of the United States government. NoAA has access to data collected from thousands of base stations around the world, which collect data periodically on weather and climate conditions. This dataset contains the latest 5 years of available data.
Variables	'indicator', 'value', 'station', 'fl_cmiss', 'date', 'fl_miss', 'datatype', 'country'
Tags	'el nino', 'rainfall - precipitation', 'weather and climate'

Metadata 是文档内容之外的描述信息，用于标记来源、时间、作者、章节、页码、权限和业务标签

在 RAG 中，metadata 可以帮助系统进行检索过滤、结果排序、来源引用、权限控制和版本追踪，从而提升召回准确性和答案可信度。

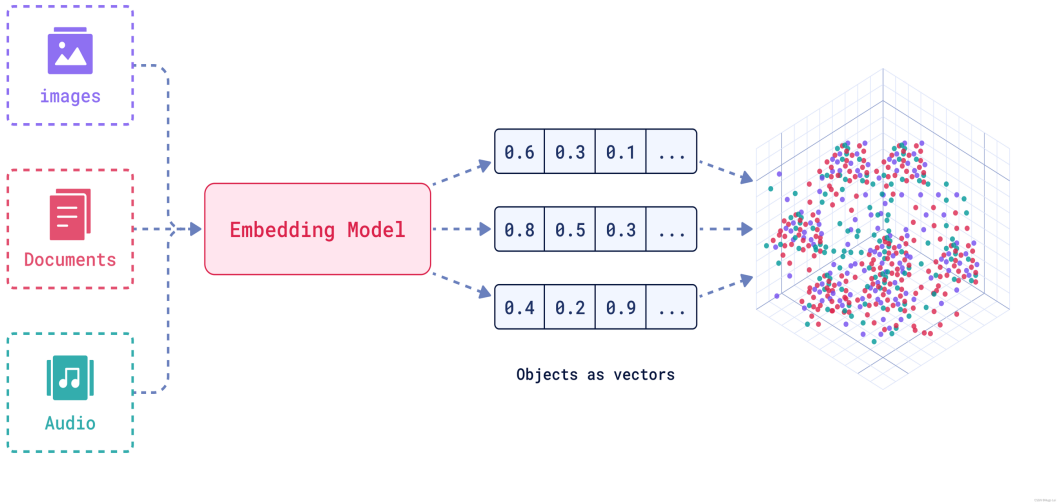
文件清洗：去重、去噪、去模版

去重	去噪	去模版
1.对完全相同的文本段落进行哈希化处理。 2.针对相似内容，使用 SimHash 或 MinHash 算法合并高相似度文档。 3.构建图谱时，对来自不同来源的实体进行实体对齐	1.内容净化： 去除HTML标签、Markdown格式字符、乱码、不必要的特殊符号。 2.删除文档页眉、页脚、广告文字、广告链接等。 3.合并被强行断开的行（PDF转义问题），统一标点和编码（如UTF-8）	删除反复出现、信息增量较低的文本。

目 录

为什么需要Chunking

嵌入模型限制



LLM上下文限制

目前闭源 API 的长上下文第一梯队	Grok 4.1 Fast：2M；GPT-5.5 / GPT-5.4、Claude Opus 4.7 / Sonnet 4.6、Gemini 3.1 Pro / 2.5 Pro / 2.5 Flash、Amazon Nova Premier：约 1M。
中文/国内模型长上下文主流档位	Qwen3-Max：262K，Qwen3 开放权重可原生 262K、扩展到 1M；Kimi K2：128K/部分版本 256K；GLM-4.5：128K；DeepSeek-V3.2：128K-164K。

什么是chunk

One of the most important things I didn't understand about the world when I was a child is the degree to which the returns for performance are superlinear.

Teachers and coaches implicitly told us the returns were linear. "You get out," I heard a thousand times, "what you put in." They meant well, but this is rarely true. If your product is only half as good as your competitor's, you don't get half as many customers. You get no customers, and you go out of business.

It's obviously true that the returns for performance are superlinear in business. Some think this is a flaw of capitalism, and that if we changed the rules it would stop being true. But superlinear returns for performance are a feature of the world, not an artifact of rules we've invented. We see the same pattern in fame, power, military victories, knowledge, and even benefit to humanity. In all of these, the rich get richer. [1]

You can't understand the world without understanding the concept of superlinear returns. And if you're ambitious you definitely should, because this will be the wave you surf on.

It may seem as if there are a lot of different situations with superlinear returns, but as far as I can tell they reduce to two fundamental causes: exponential growth and thresholds.

The most obvious case of superlinear returns is when you're working on something that grows exponentially. For example, growing bacterial cultures. When they grow at all, they grow exponentially. But they're tricky to grow. Which means the difference in outcome between someone who's adept at it and someone who's not is very great.

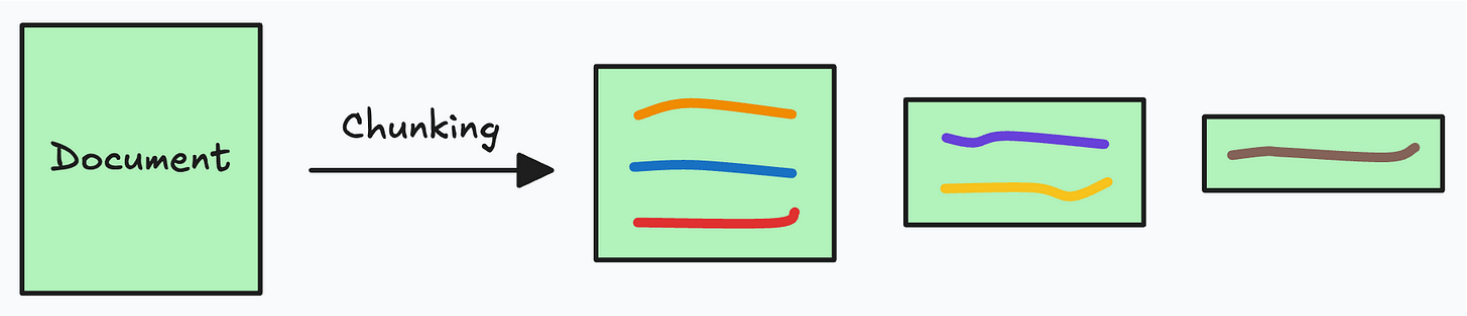
Startups can also grow exponentially, and we see the same pattern there. Some manage to achieve high growth rates. Most don't. And as a result you get qualitatively different outcomes: the companies with high growth rates tend to become immensely valuable, while the ones with lower growth rates may not even survive.

Y Combinator encourages founders to focus on growth rate rather than absolute numbers. It prevents them from being discouraged early on, when the absolute numbers are still low. It also helps them decide what to focus on: you can use growth rate as a compass to tell you how to evolve the company. But the main advantage is that by focusing on growth rate you tend to get something that grows exponentially.

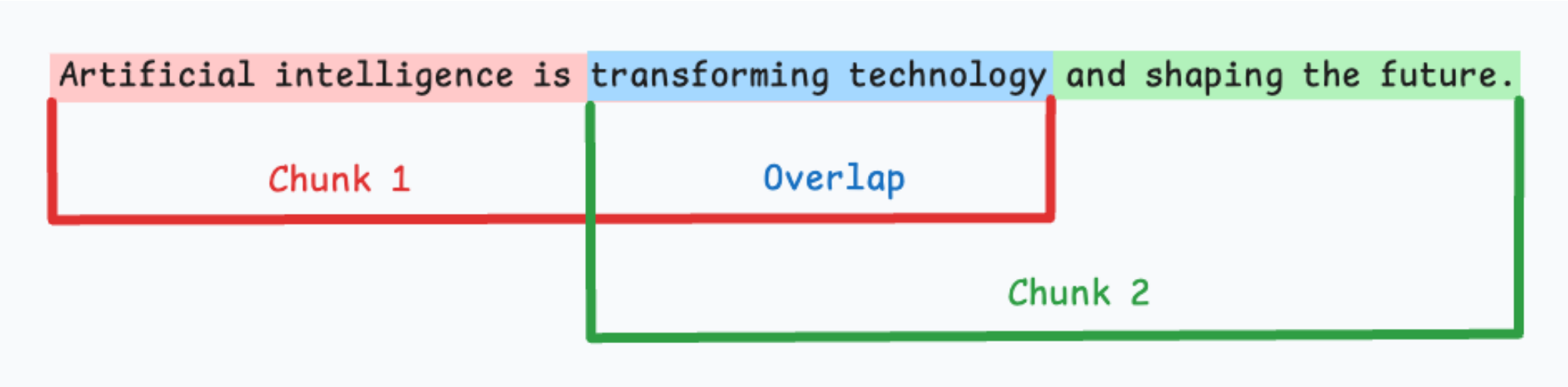
YC doesn't explicitly tell founders that with growth rate "you get out what you put in," but it's not far from the truth. And if growth rate were proportional to performance, then the reward for performance p over time t would be proportional to pt.

Even after decades of thinking about this, I find that sentence startling.

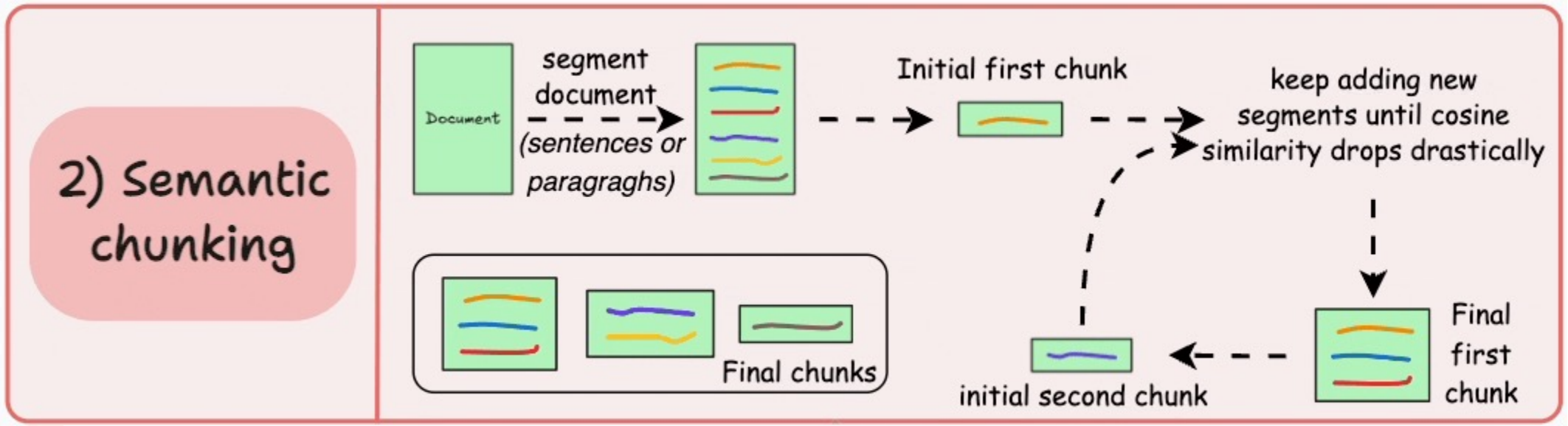
chunk size



分块策略-固定大小切分

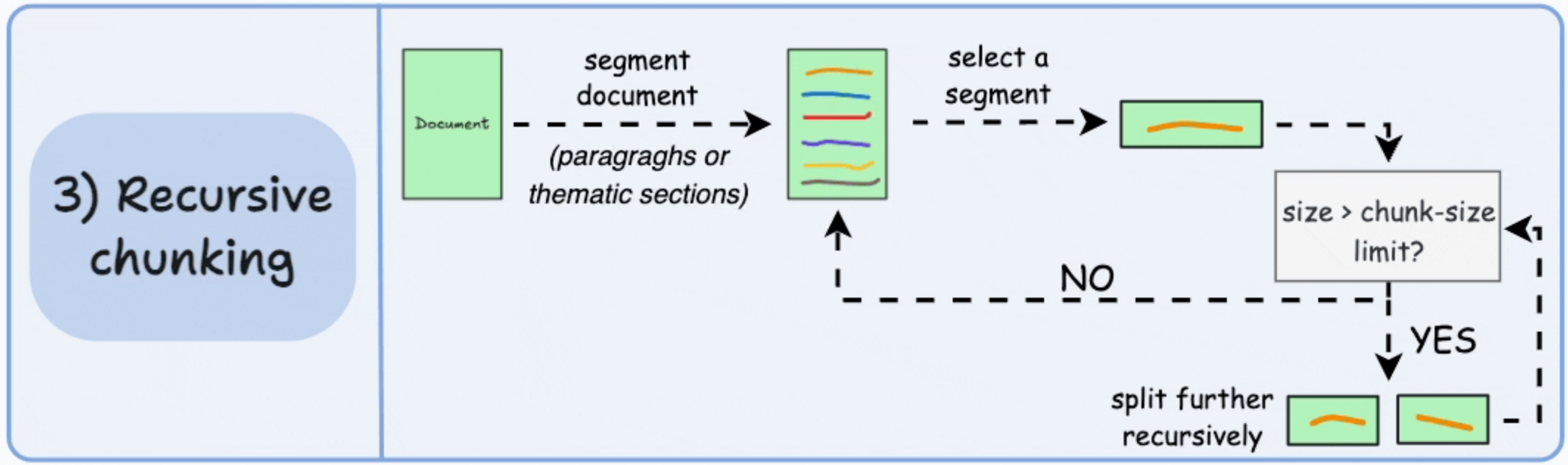

Artificial intelligence is
transforming technology
and shaping the future.
Chunk 1
Overlap
Chunk 2

语义切分



Artificial intelligence is transforming industries by automating processes, enhancing decision-making, and providing insights through data analysis. Machine learning, a subset of AI, enables systems to learn and improve from experience without explicit programming. Deep learning, a branch of machine learning, uses neural networks with multiple layers to model complex patterns in data.

递归切分


3) Recursive chunking
Document
segment document
(paragraghs or thematic sections)
select a segment
size > chunk-size limit?
NO
YES
split further recursively

递归切分

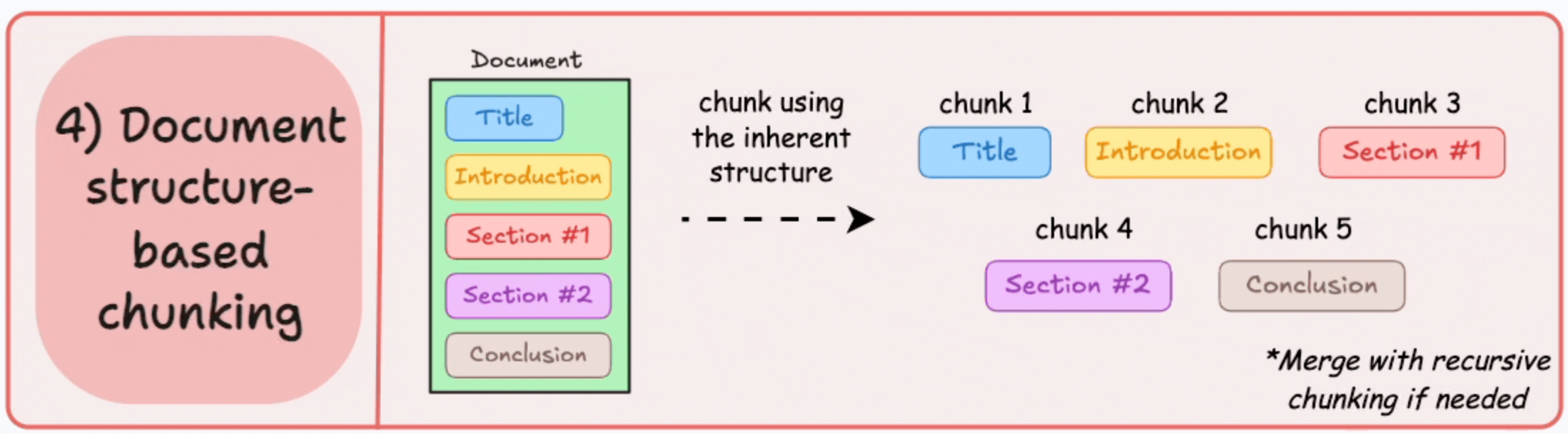
Paragraph 1

Artificial intelligence is transforming industries by automating processes, enhancing decision-making, and providing insights through data analysis. Machine learning, a subset of AI, enables systems to learn and improve from experience without explicit programming. Deep learning, a branch of machine learning, uses neural networks with multiple layers to model complex patterns in data.

Paragraph 2

AI is also improving natural language processing, enabling applications like chatbots and virtual assistants.

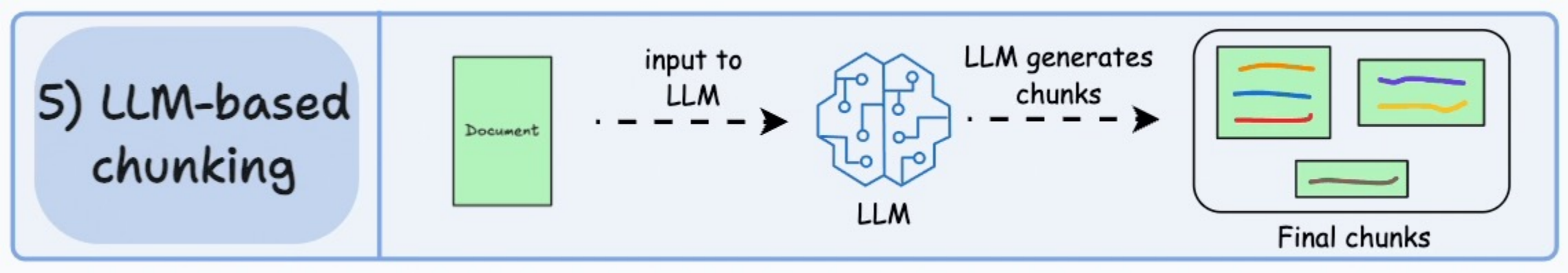
文档结构切分


4) Document structure-based chunking
Document
Title
Introduction
Section #1
Section #2
Conclusion
chunk using the inherent structure
chunk 1
Title
chunk 2
Introduction
chunk 3
Section #1
chunk 4
Section #2
chunk 5
Conclusion
*Merge with recursive chunking if needed

文档结构切分

Title: The Role of Artificial Intelligence in Modern Education	Chunk 1
Introduction Artificial intelligence (AI) is reshaping education by providing personalized learning experiences and automating administrative tasks.	Chunk 2
Section 1: Personalized Learning AI enables the customization of educational content to meet individual student needs, enhancing engagement and comprehension.	Chunk 3
Section 2: Administrative Automation From grading to scheduling, AI tools are streamlining administrative processes, allowing educators to focus more on teaching.	Chunk 4
Conclusion The integration of AI in education holds the promise of more efficient learning environments and improved student outcomes.	Chunk 5

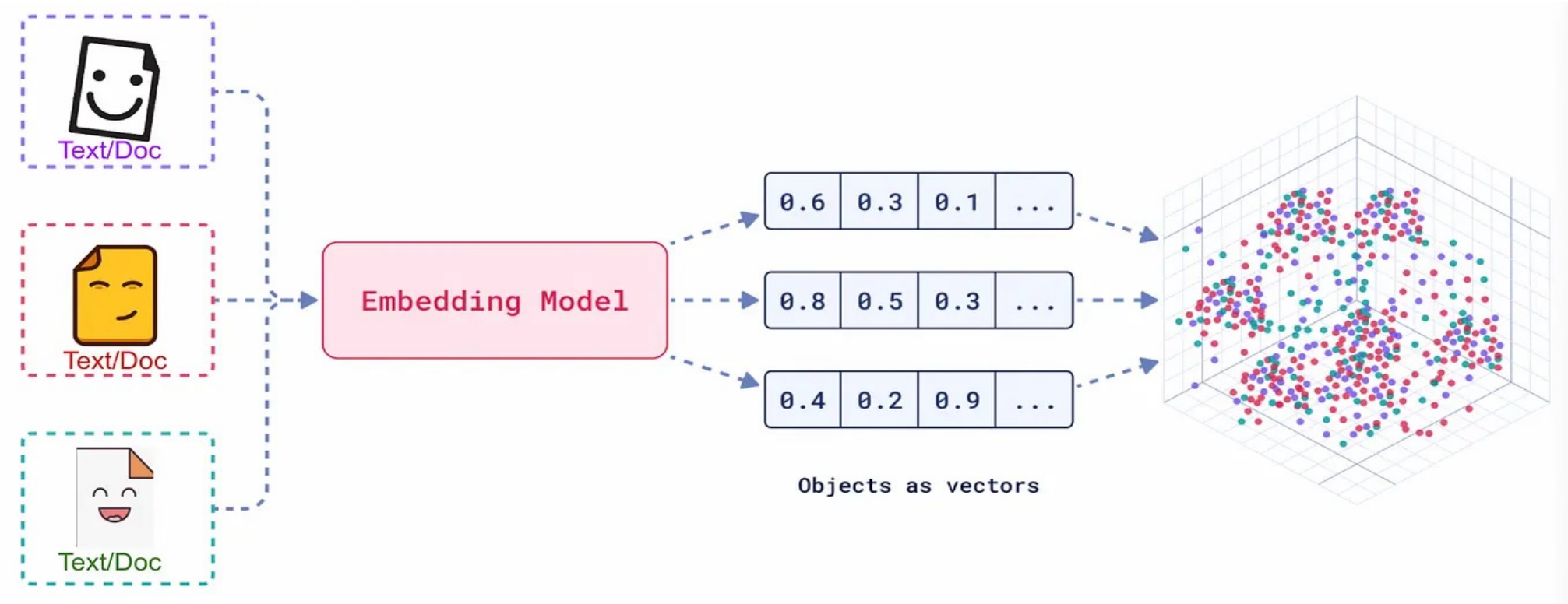
模型生成切分


5) LLM-based chunking
Document
input to LLM
LLM
LLM generates chunks
Final chunks

中国科学院大学
University of Chinese Academy of Sciences

目 录

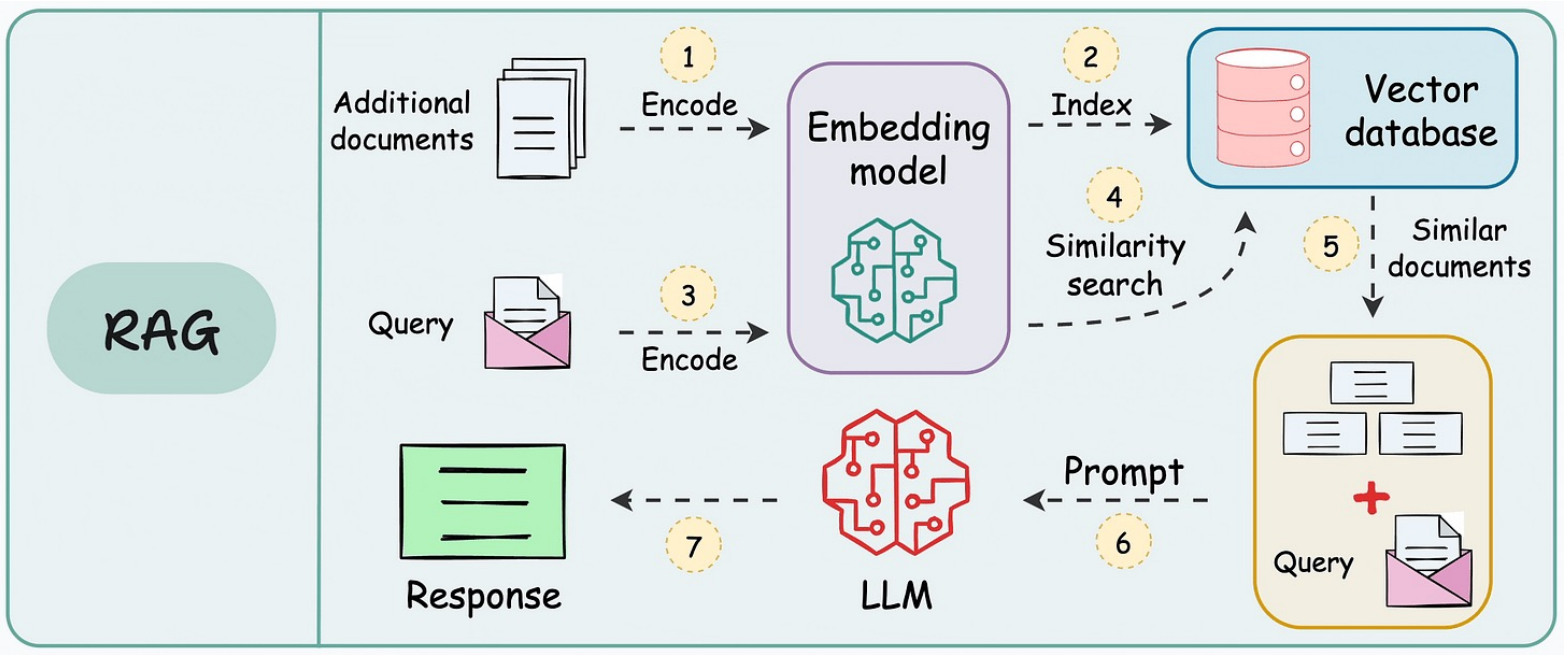
什么是Embeding



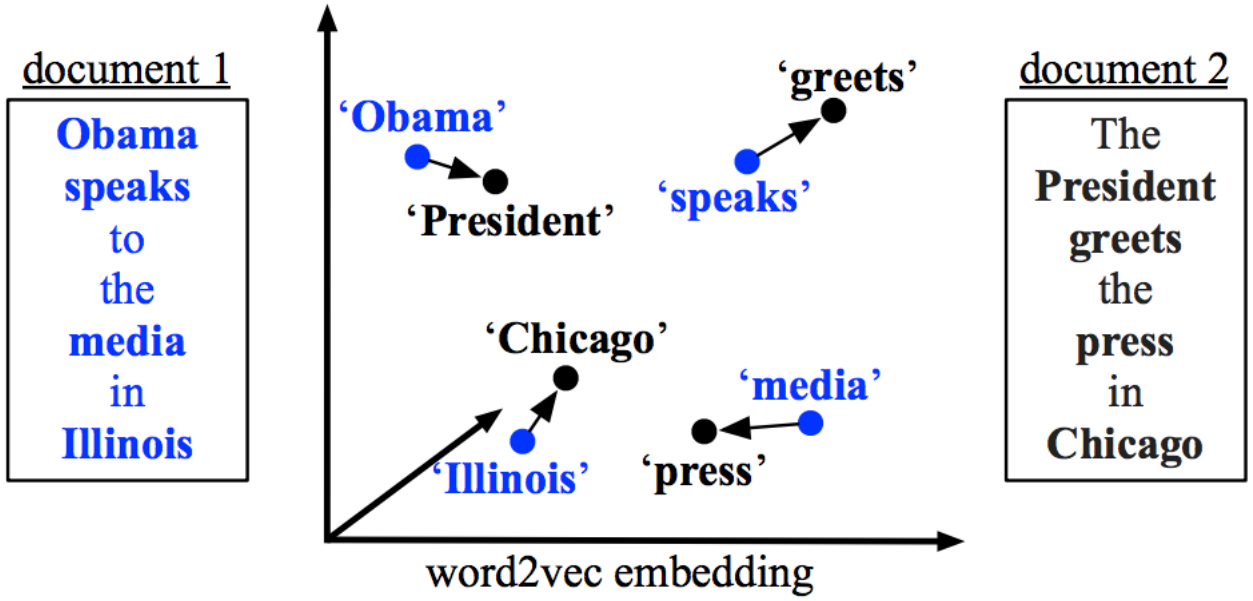
Embedding在检索中的作用

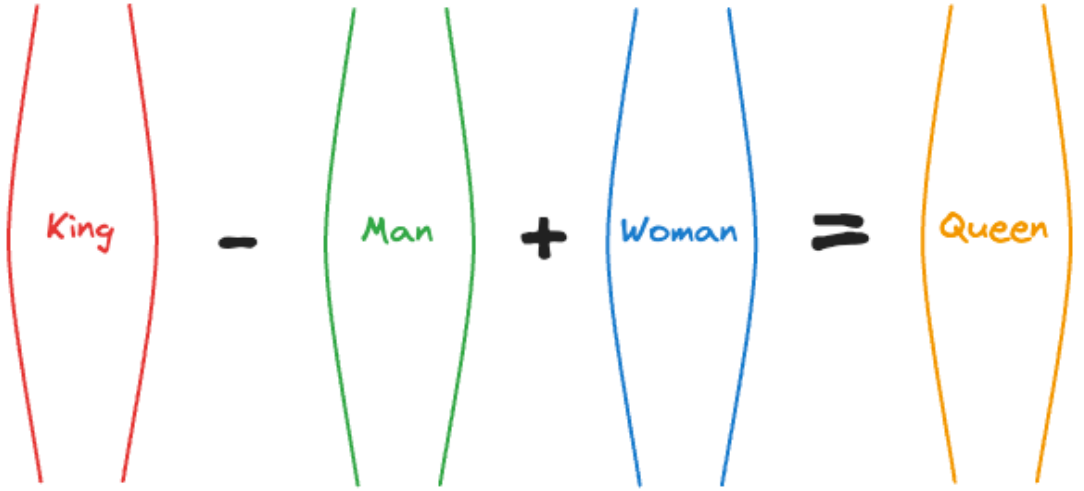
语义检索的基础

检索质量的关键

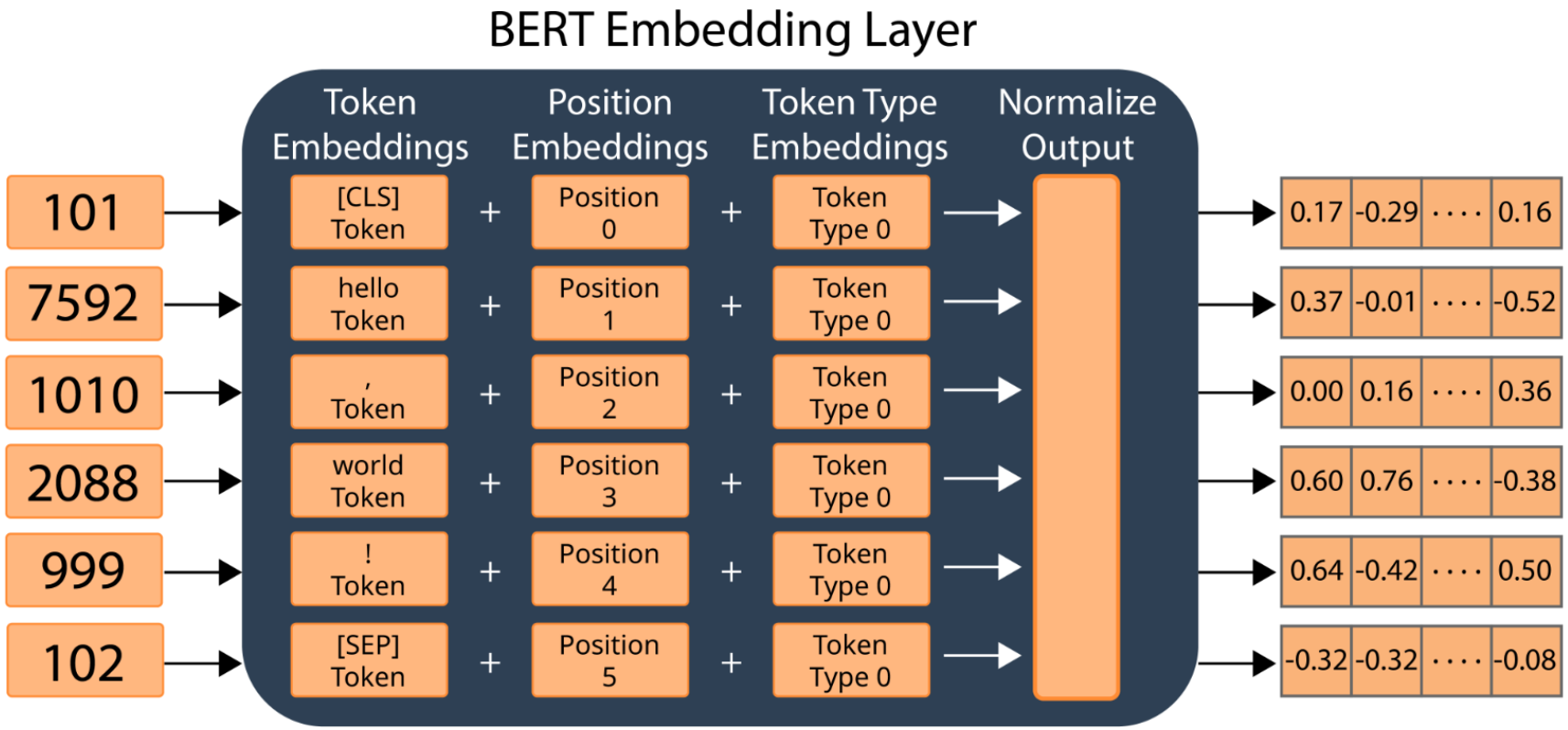


Embedding-静态词嵌入





Embedding-动态上下文嵌入

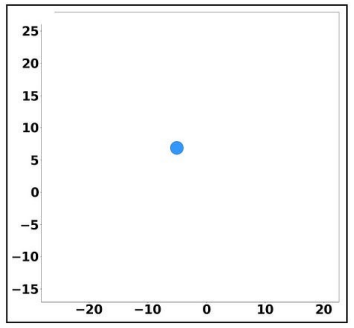


Embedding-动态上下文嵌入

Visualization of the embeddings obtained for

the word "Bank" in different contexts

Glove

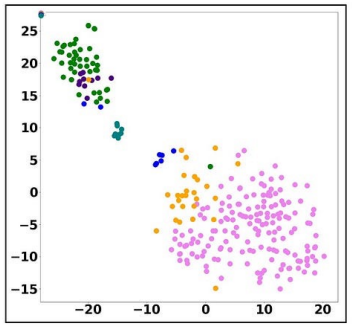


All different senses

get the same representation

❌

BERT



Model understands different

senses of a word

✅

bank

美式 英式

n. 銀行

the Bank of England/China 英格蘭 / 中國銀行

vt. 把...存入銀行

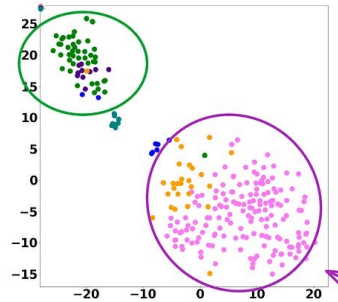
牛津中文字典

bank

岸,河邊

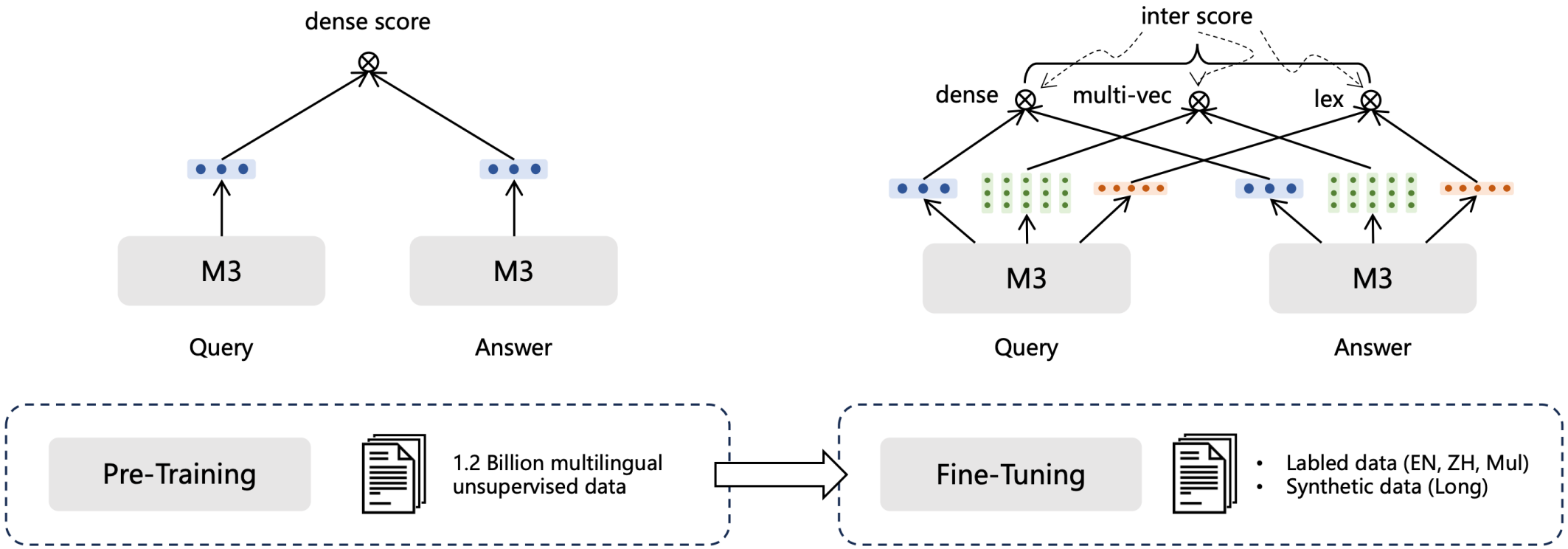
PyDict

The word 'bank' when used in water sense

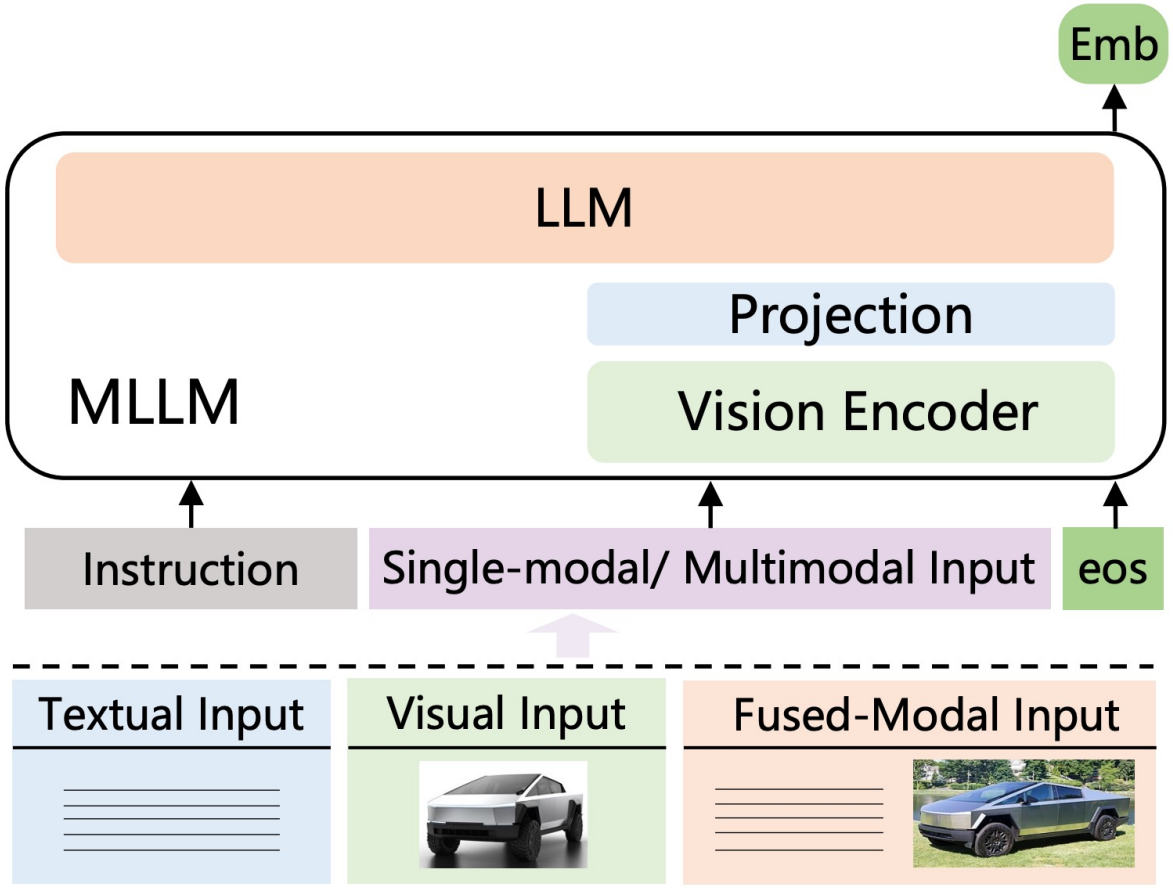


The word 'bank' when used in financial sense

Embedding-定向训练



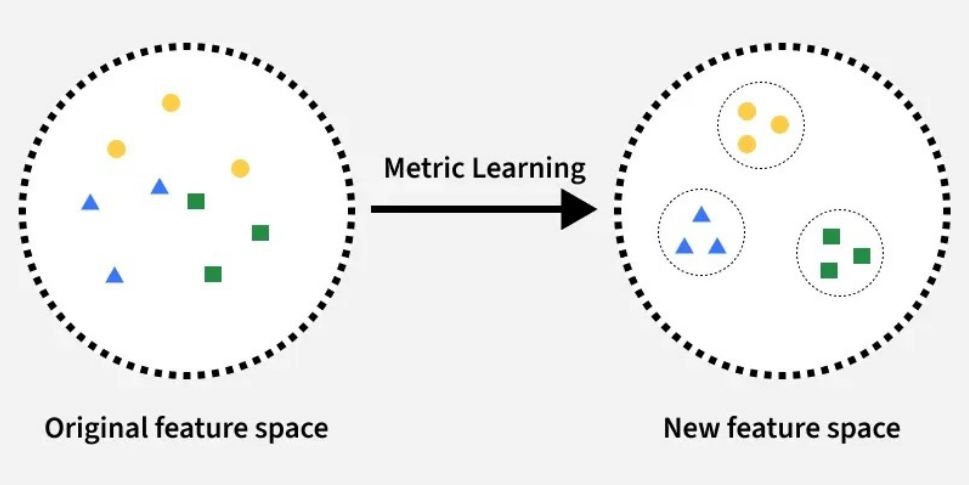
Embedding-多模态

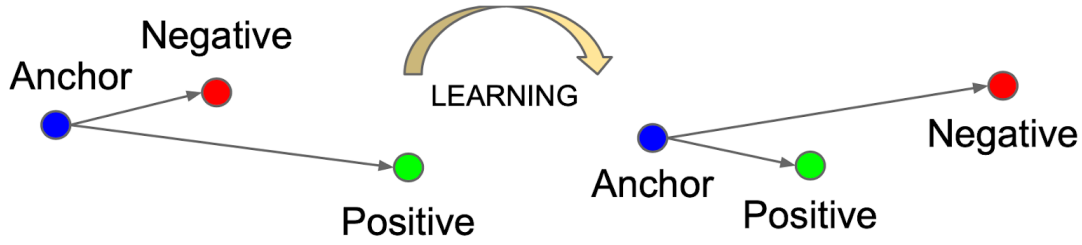


Embedding model 训练策略

度量学习

对比学习





嵌入模型选型

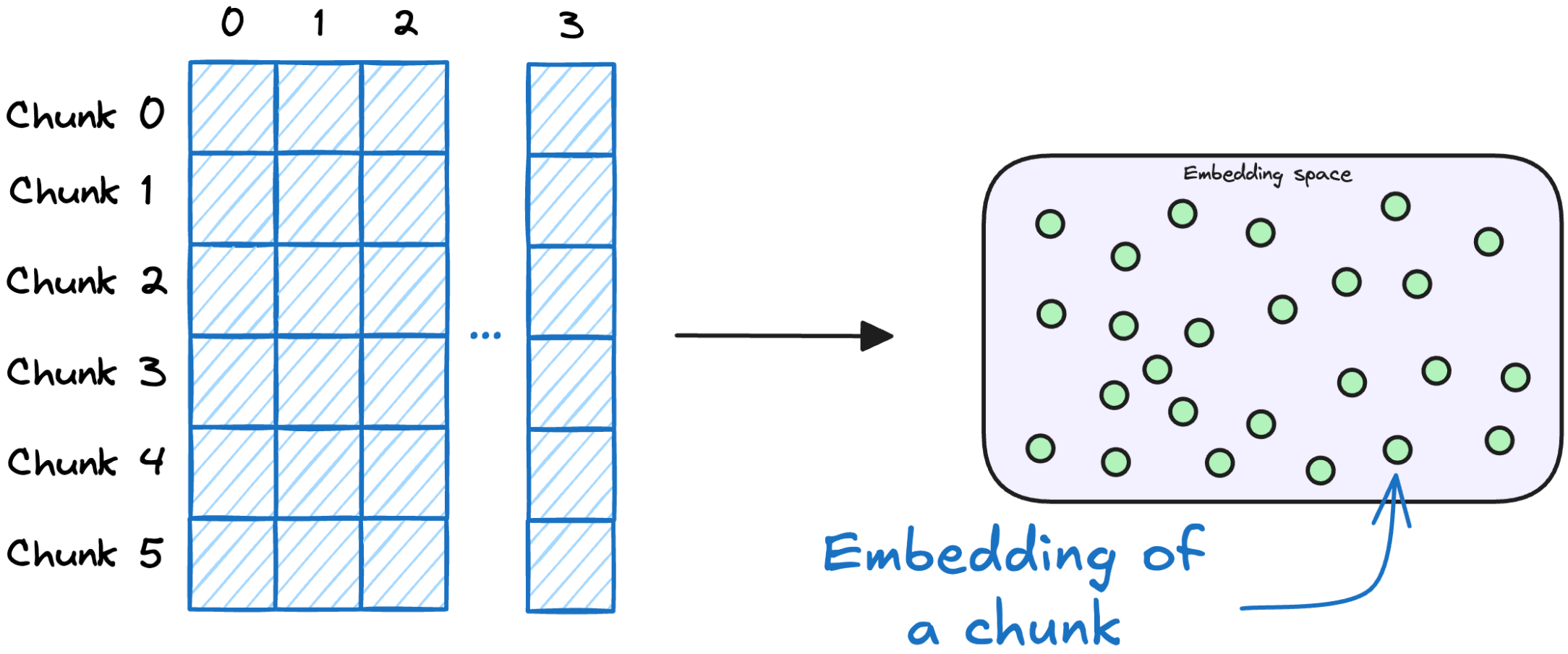
MTEB (Massive Text Embedding Benchmark

Rank (Bor…	Model	Zero-shot	Memory U…	Number of P…	Embedding D…	Max Tokens	Mean (T…	Mean (TaskT…	Bitext …	Classification
1	gemini-embedding-001	99%	Unknown	Unknown	3072	2048	68.37	59.59	79.28	71.82
2	Qwen3-Embedding-8B	99%	28866	7B	4096	32768	**70.58**	**61.69**	**80.89**	**74.00**
3	Qwen3-Embedding-4B	99%	15341	4B	2560	32768	69.45	60.86	79.36	72.33
4	Qwen3-Embedding-0.6B	99%	2272	595M	1024	32768	64.34	56.01	72.23	66.83
5	Linq-Embed-Mistral	99%	13563	7B	4096	32768	61.47	54.14	70.34	62.24
6	gte-Qwen2-7B-instruct	⚠ NA	29040	7B	3584	32768	62.51	55.93	73.92	61.55
7	multilingual-e5-large-instruct	99%	1068	560M	1024	514	63.22	55.08	80.13	64.94
8	SFR-Embedding-Mistral	96%	13563	7B	4096	32768	60.90	53.92	70.00	60.02
9	text-multilingual-embedding-002	99%	Unknown	Unknown	768	2048	62.16	54.25	70.73	64.64
10	GritLM-7B	99%	13813	7B	4096	4096	60.92	53.74	70.53	61.83

嵌入模型选型-指标

任务	对于 RAG 应用，需要重点关注模型在 Retrieval (检索) 任务下的排名。
语言	对于 RAG 应用，模型是否支持当前业务数据所使用的语言
模型规模	模型越大，通常性能越好，但对硬件（显存）的要求也越高，推理速度也越慢
模型上文长度	模型能处理的文本长度上限，影响分块策略的设计

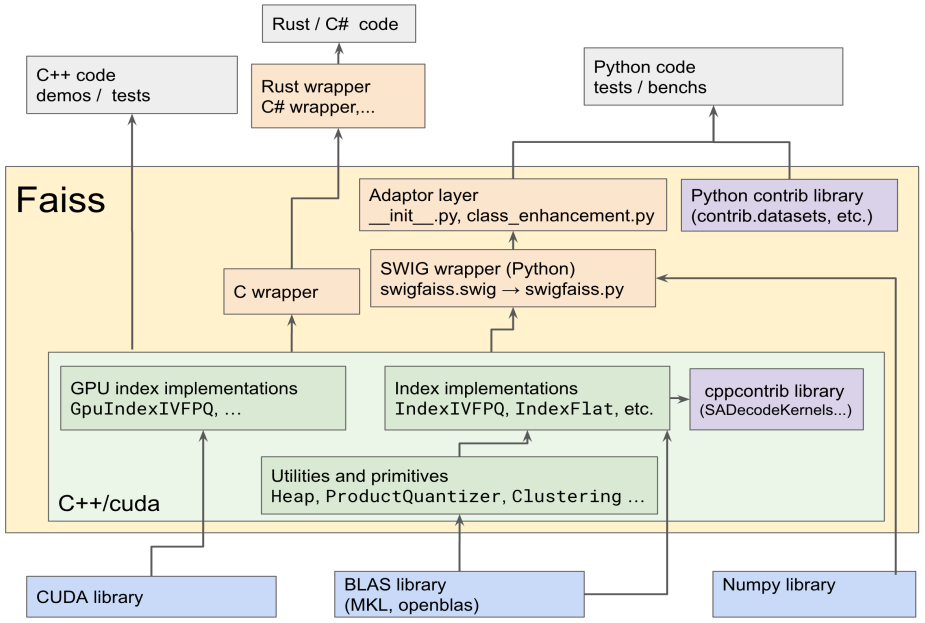
向量数据库-嵌入向量存储

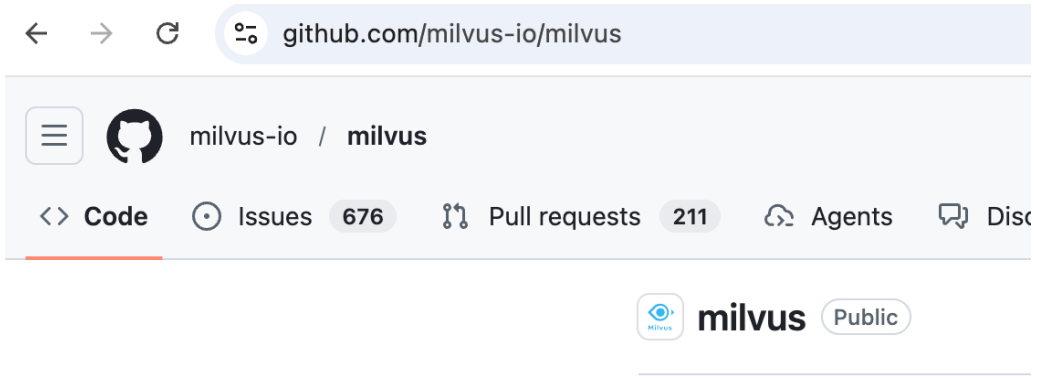


向量数据库-特点

维度	向量数据库	传统数据库 (RDBMS)
核心数据类型	高维向量 (Embeddings)	结构化数据 (文本、数字、日期)
查询方式	**相似性搜索** (ANN)	**精确匹配**
索引机制	HNSW, IVF, LSH 等 ANN 索引	B-Tree, Hash Index
主要应用场景	AI 应用、RAG、推荐系统、图像/语音识别	业务系统 (ERP, CRM)、金融交易、数据报表
数据规模	轻松应对千亿级向量	通常在千万到亿级行数据，更大规模需复杂分库分表
性能特点	高维数据检索性能极高，计算密集型	结构化数据查询快，高维数据查询性能呈指数级下降
一致性	通常为最终一致性	强一致性 (ACID 事务)

向量数据库选型

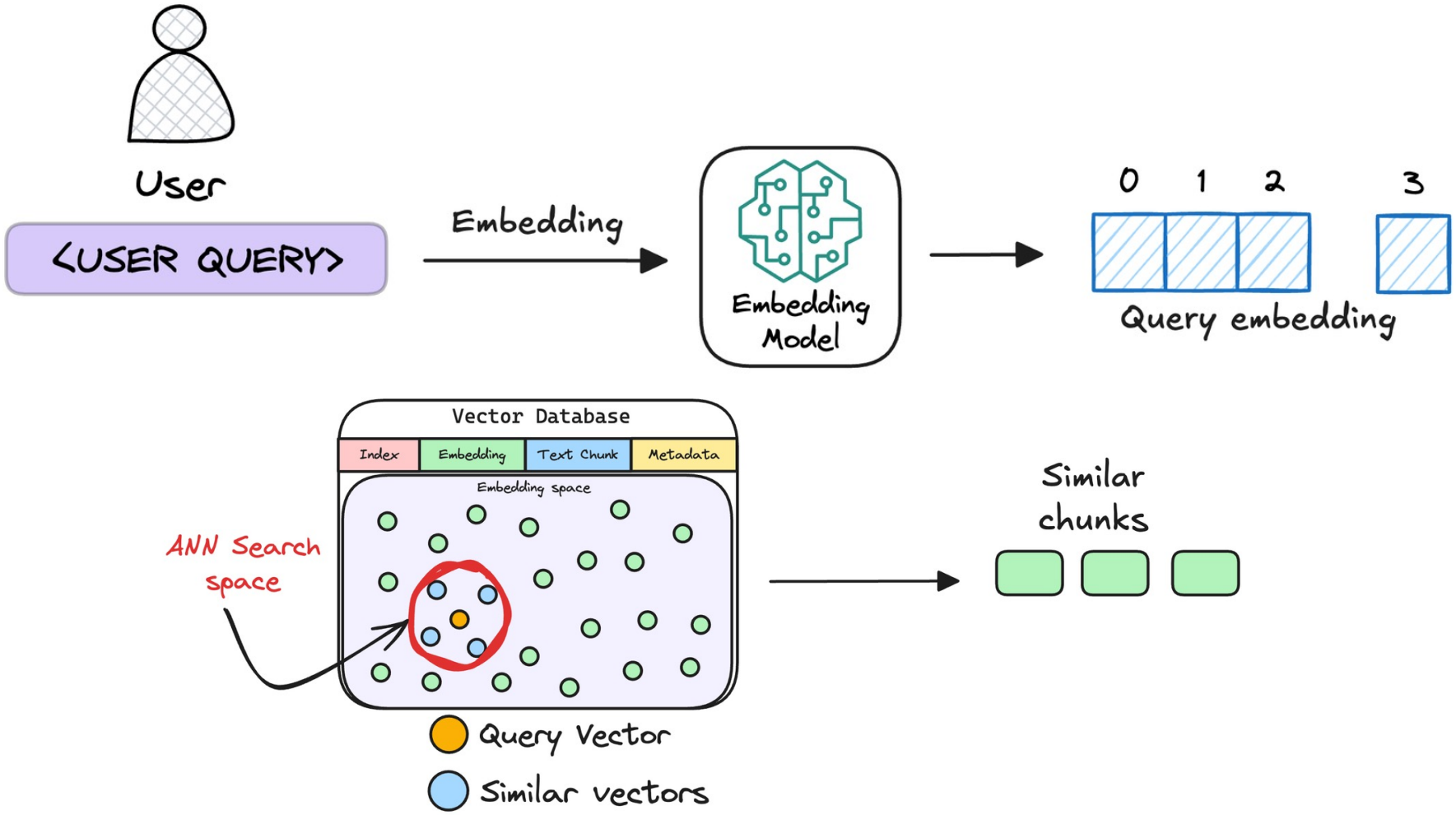

Rust / C# code
C++ code
demos / tests
Rust wrapper
C# wrapper,...
Python code
tests / benchs
Faiss
Adaptor layer
__init__.py, class_enhancement.py
Python contrib library
(contrib.datasets, etc.)
C wrapper
SWIG wrapper (Python)
swigfaiss.swig → swigfaiss.py
GPU index implementations
GpuIndexIVFPQ, ...
Index implementations
IndexIVFPQ, IndexFlat, etc.
cppcontrib library
(SADecodeKernels...)
Utilities and primitives
Heap, ProductQuantizer, Clustering ...
C++/cuda
CUDA library
BLAS library
(MKL, openblas)
Numpy library


github.com/milvus-io/milvus
milvus-io / milvus
Code
Issues 676
Pull requests 211
Agents
milvus
Public

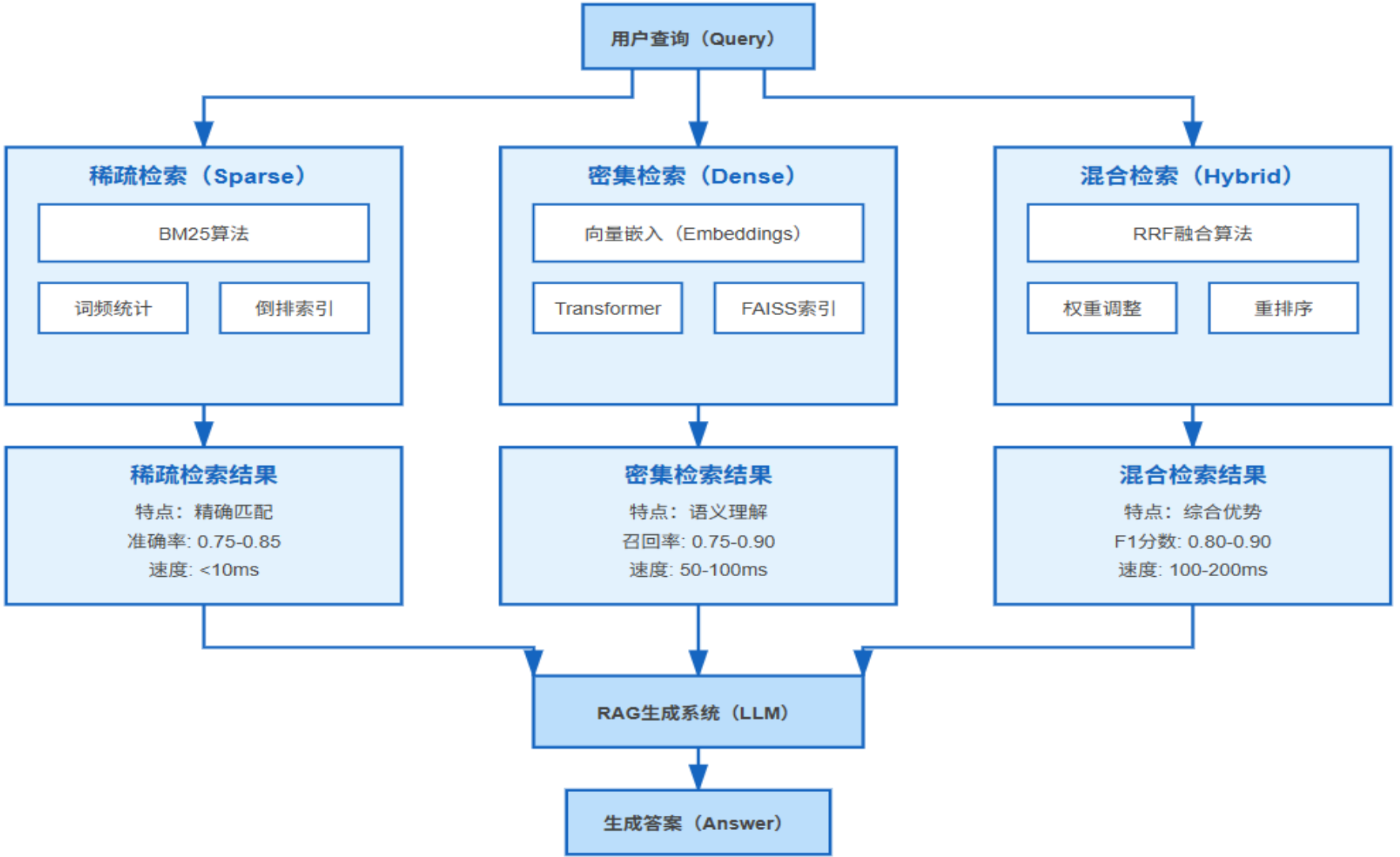
中国科学院大学
University of Chinese Academy of Sciences

目录

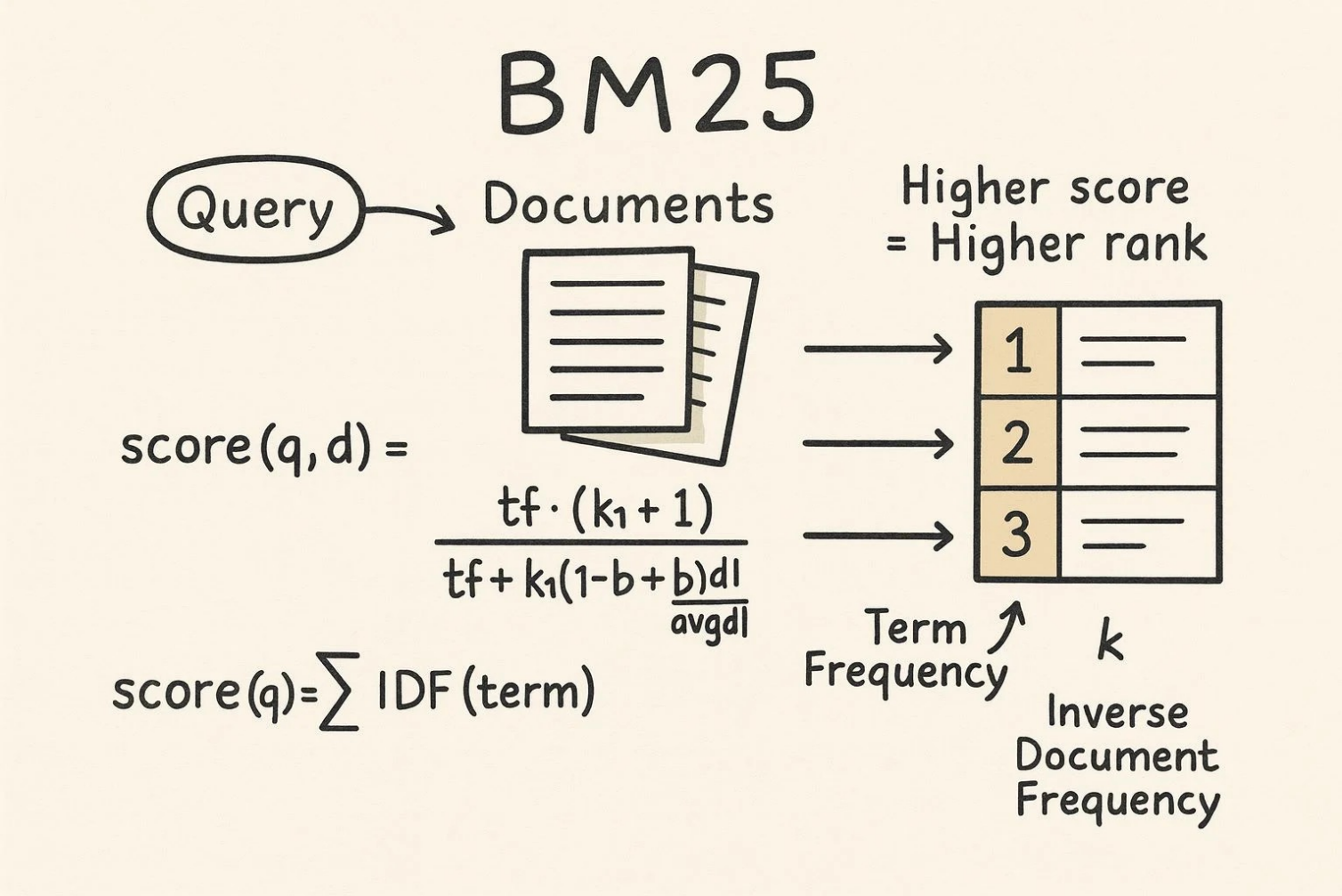
什么是检索


User
<USER QUERY>
Embedding
Embedding Model
0 1 2 3
Query embedding
Vector Database
Index
Embedding
Text Chunk
Metadata
Embedding space
ANN Search space
Similar chunks
Query Vector
Similar vectors

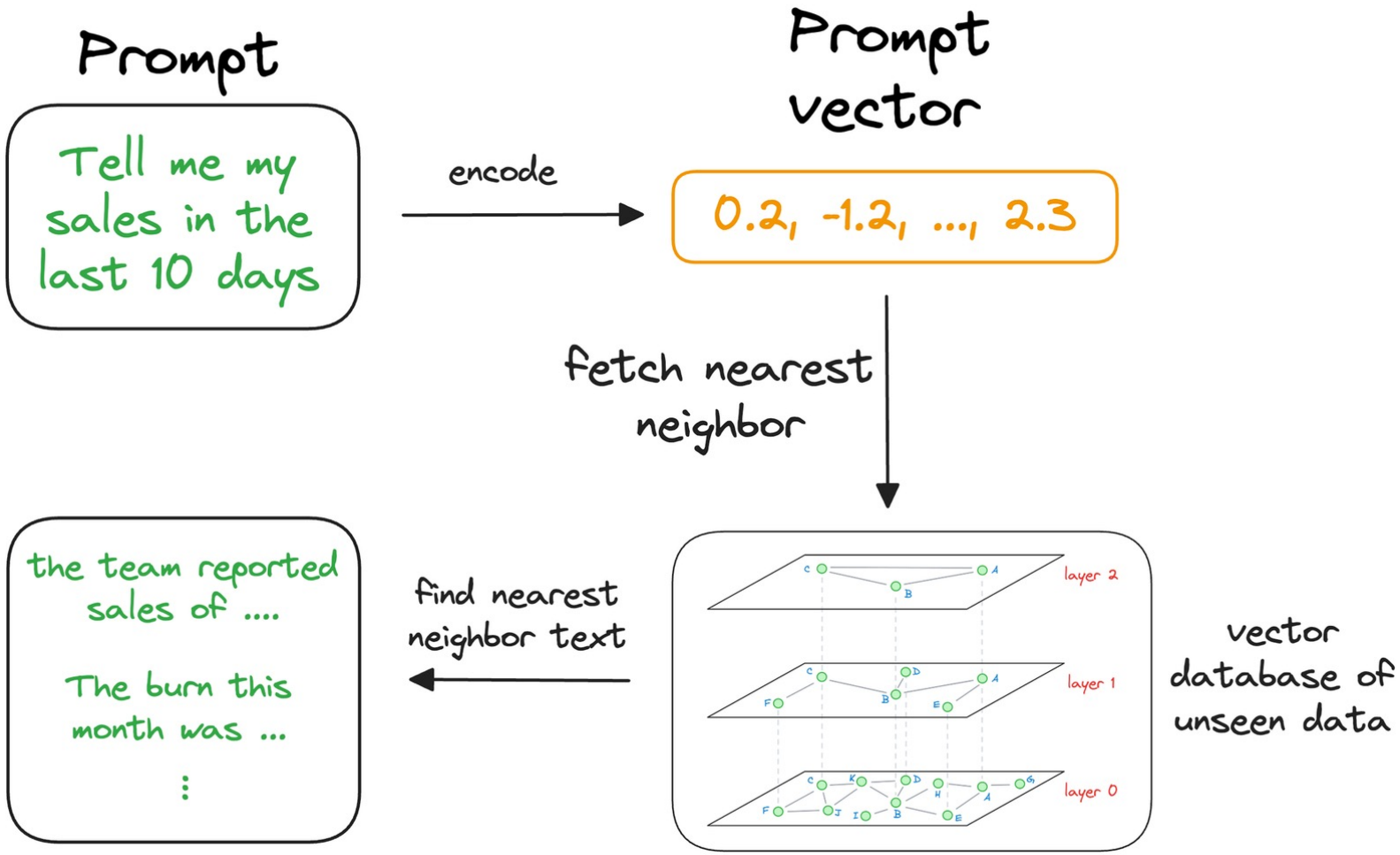
RAG检索


用户查询（Query）
稀疏检索（Sparse）
BM25算法
词频统计
倒排索引
密集检索（Dense）
向量嵌入（Embeddings）
Transformer
FAISS索引
混合检索（Hybrid）
RRF融合算法
权重调整
重排序
稀疏检索结果
特点：精确匹配
准确率: 0.75-0.85
速度: <10ms
密集检索结果
特点：语义理解
召回率: 0.75-0.90
速度: 50-100ms
混合检索结果
特点：综合优势
F1分数: 0.80-0.90
速度: 100-200ms
RAG生成系统（LLM）
生成答案（Answer）

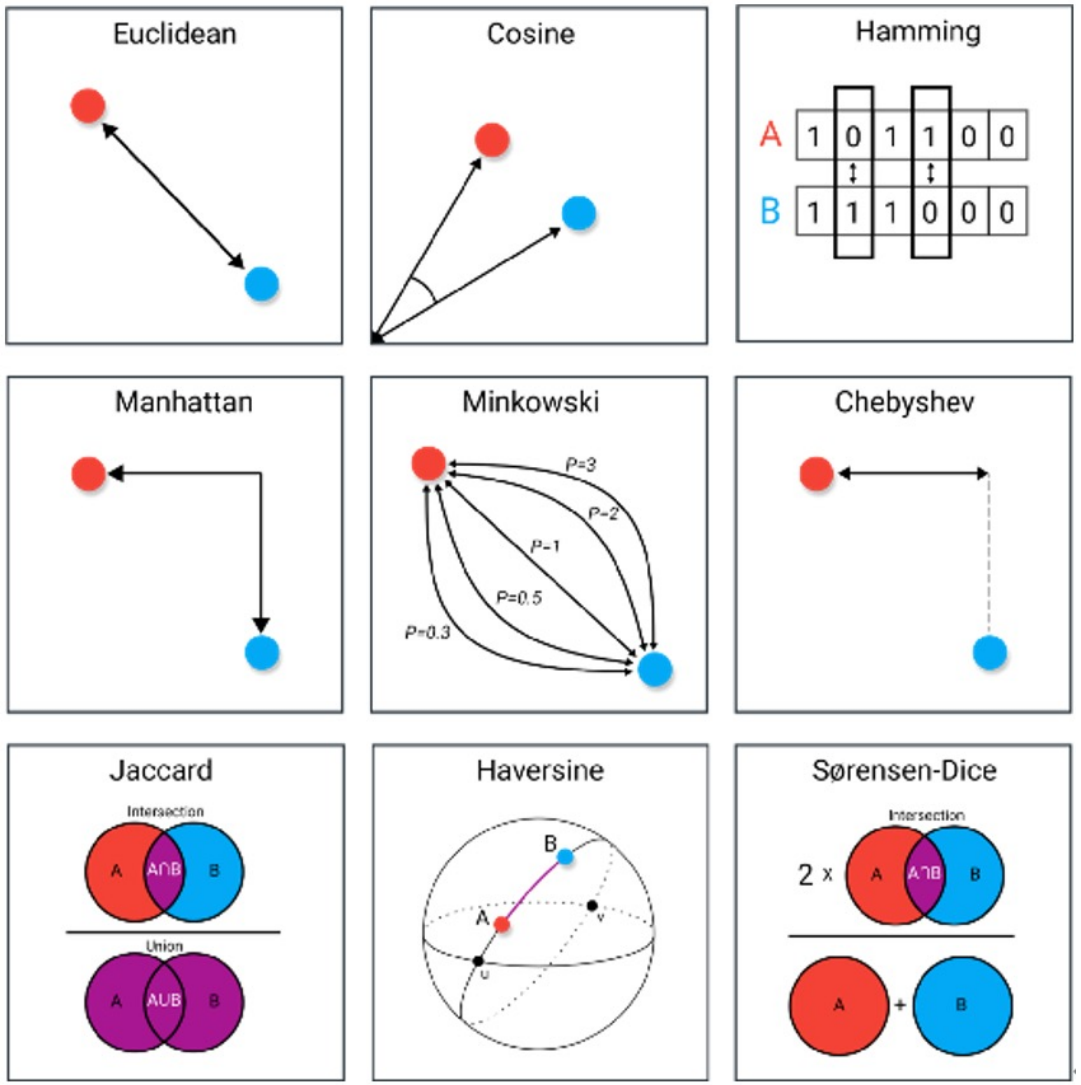
稀疏检索-BM25


BM25
Query
Documents
Higher score
= Higher rank
score(q,d) =
$\frac{tf \cdot (k_1+1)}{tf + k_1(1-b+b\frac{dl}{avgdl})}$
1
2
3
Term Frequency
k
Inverse Document Frequency
score(q)=∑ IDF(term)

密集检索-向量相似度检索

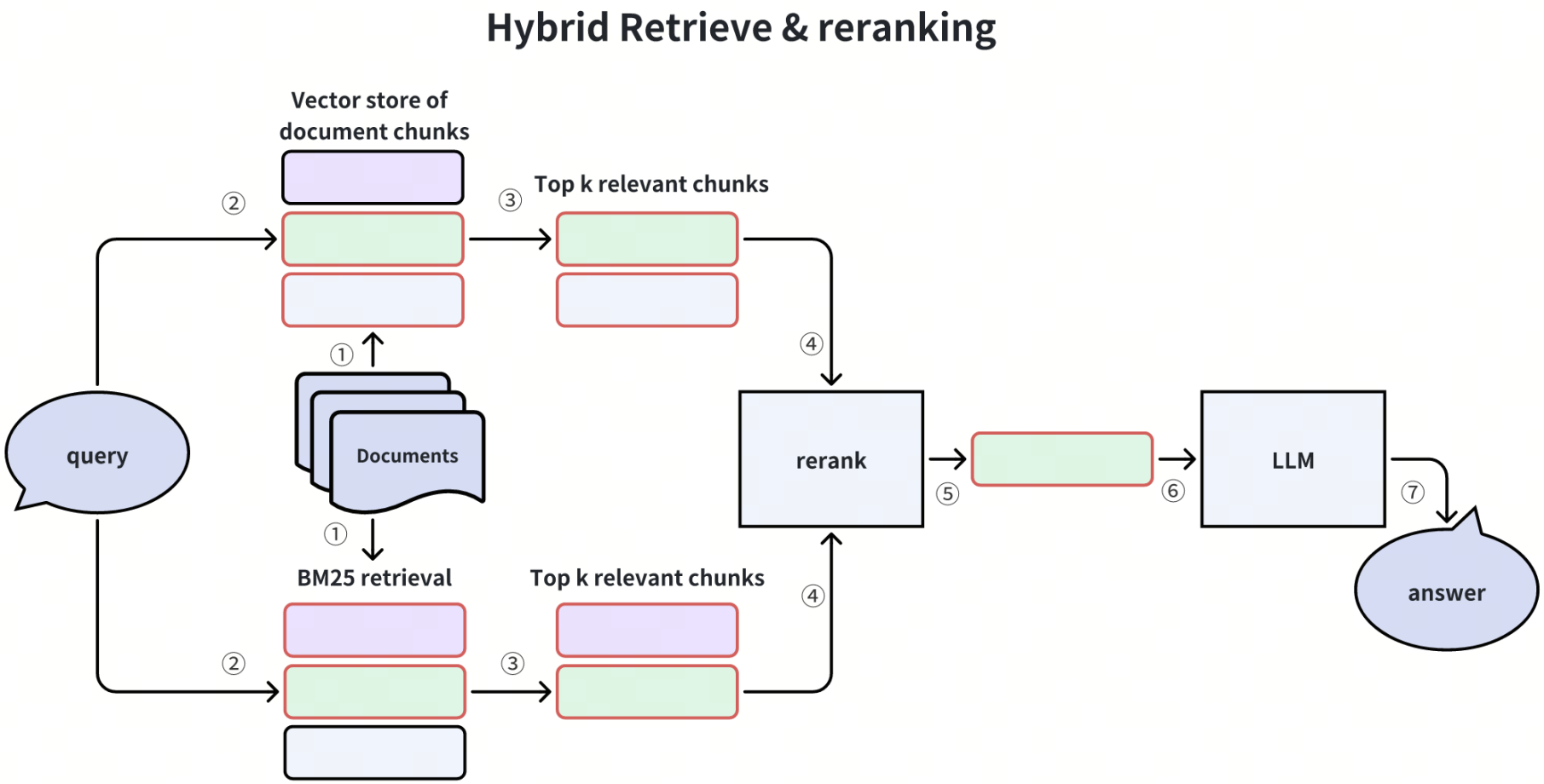


相似度度量

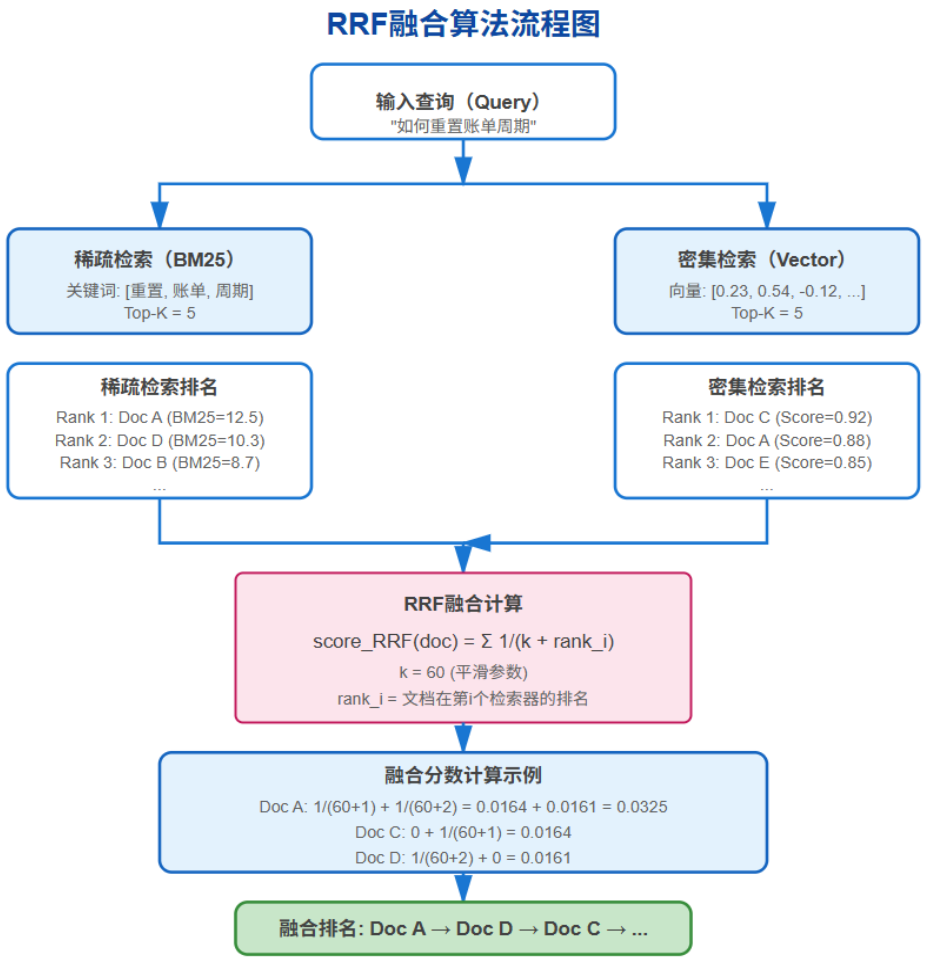


- 欧式距离
- 余弦相似度
- 点积相似度
- 曼哈顿距离
- 切比雪夫距离
- 。 。 。

混合检索


Hybrid Retrieve & reranking
Vector store of document chunks
Top k relevant chunks
query
Documents
rerank
LLM
answer
BM25 retrieval
Top k relevant chunks
①
②
③
④
⑤
⑥
⑦

混合检索-倒数排名融合RRF



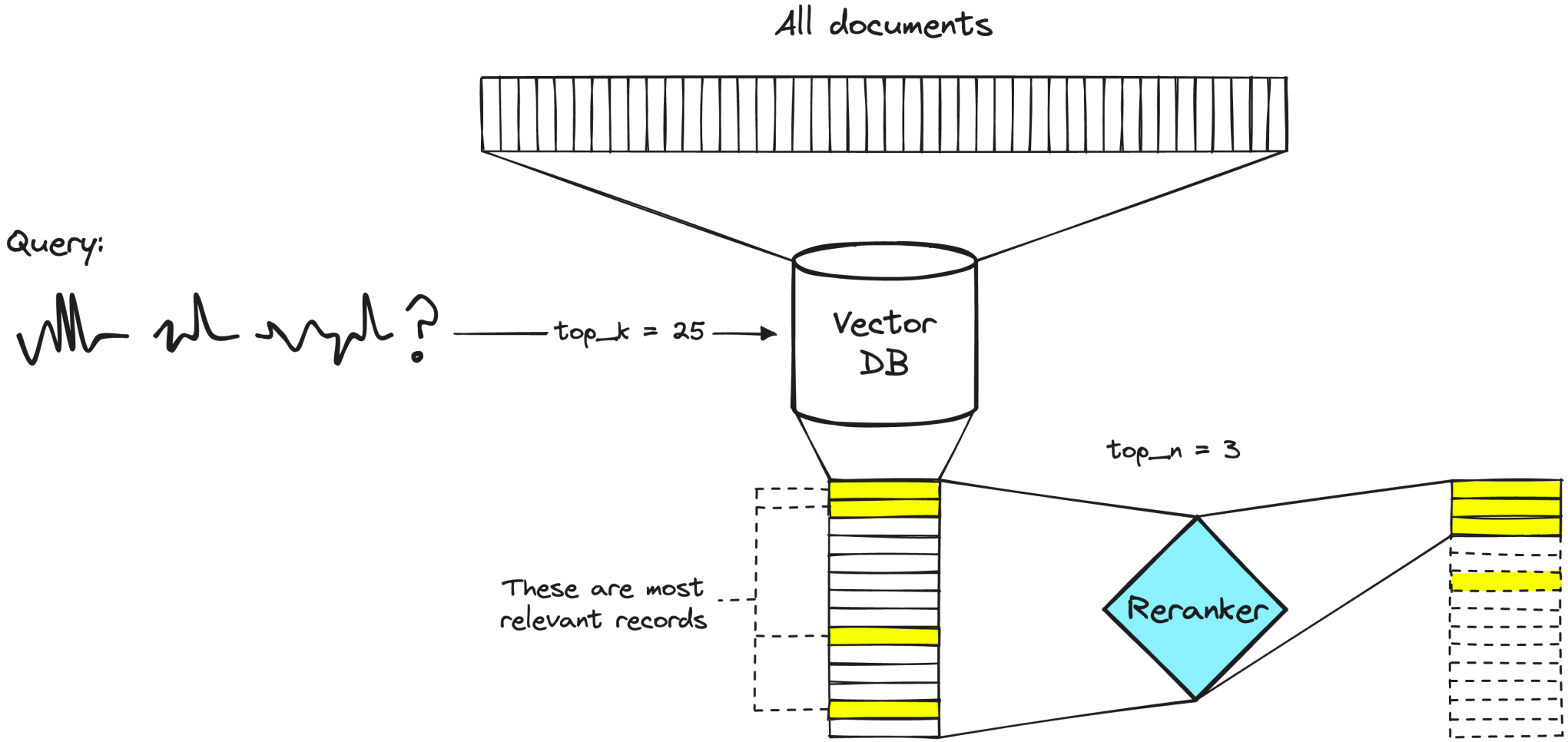
各检索器检索并计分

不同检索器得分排名

根据文档排名赋分

得到文本相似度最终排序

检索增强-Rerank


All documents
Query:
top_k = 25
Vector DB
top_n = 3
These are most relevant records
Reranker

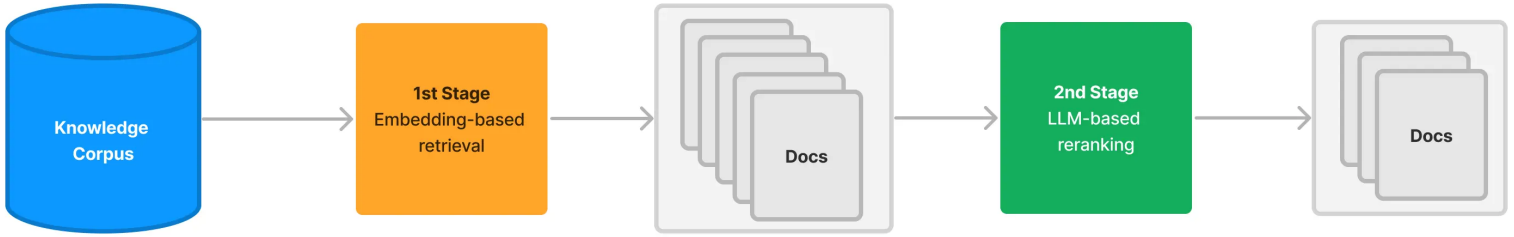
检索增强-Rerank

以下是一个文档列表，每个文档都有一个编号和摘要。同时提供一个问题。请根据问题，按相关性顺序列出您认为需要查阅的文档编号，并给出相关性分数（1-10分）。请不要包含与问题无关的文档。

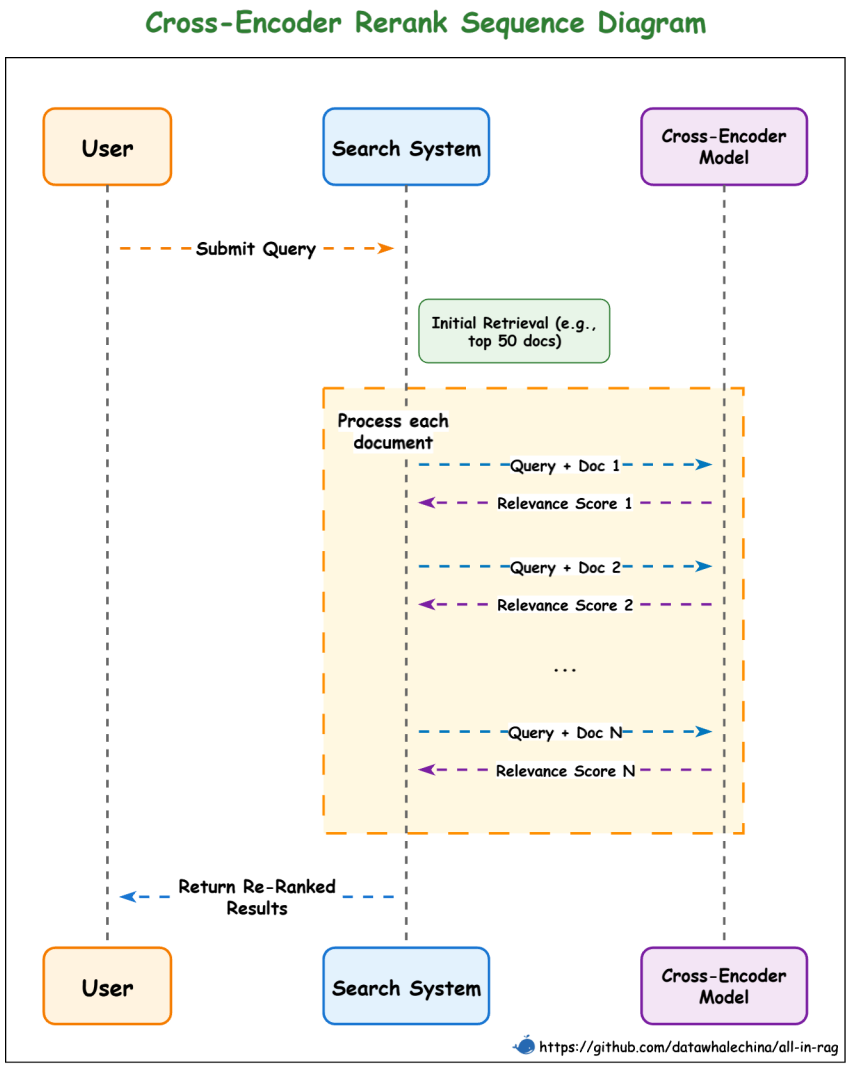
示例格式:
文档 1: <文档1的摘要>
文档 2: <文档2的摘要>
...
文档 10: <文档10的摘要>

问题: <用户的问题>

回答:
Doc: 9, Relevance: 7
Doc: 3, Relevance: 4
Doc: 7, Relevance: 3



检索增强-Cross Encoder

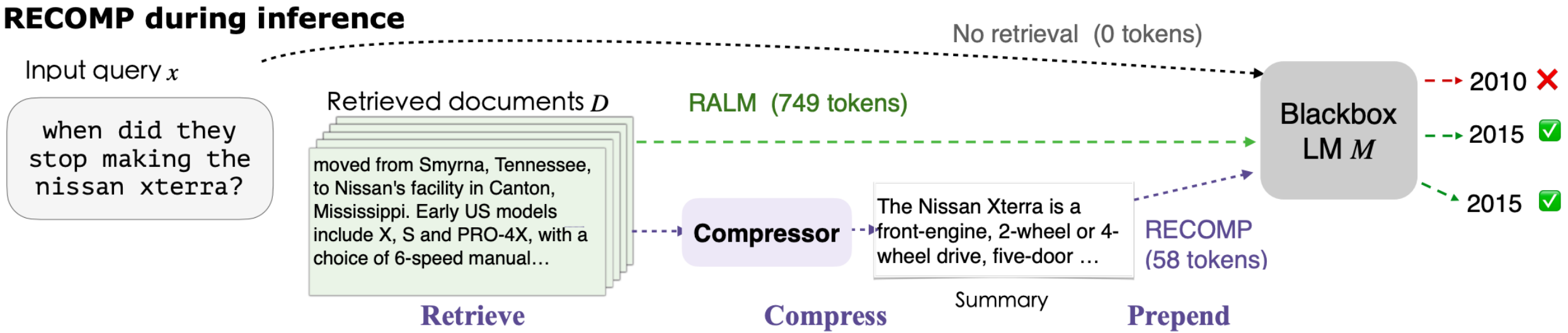


每个文档单独计分

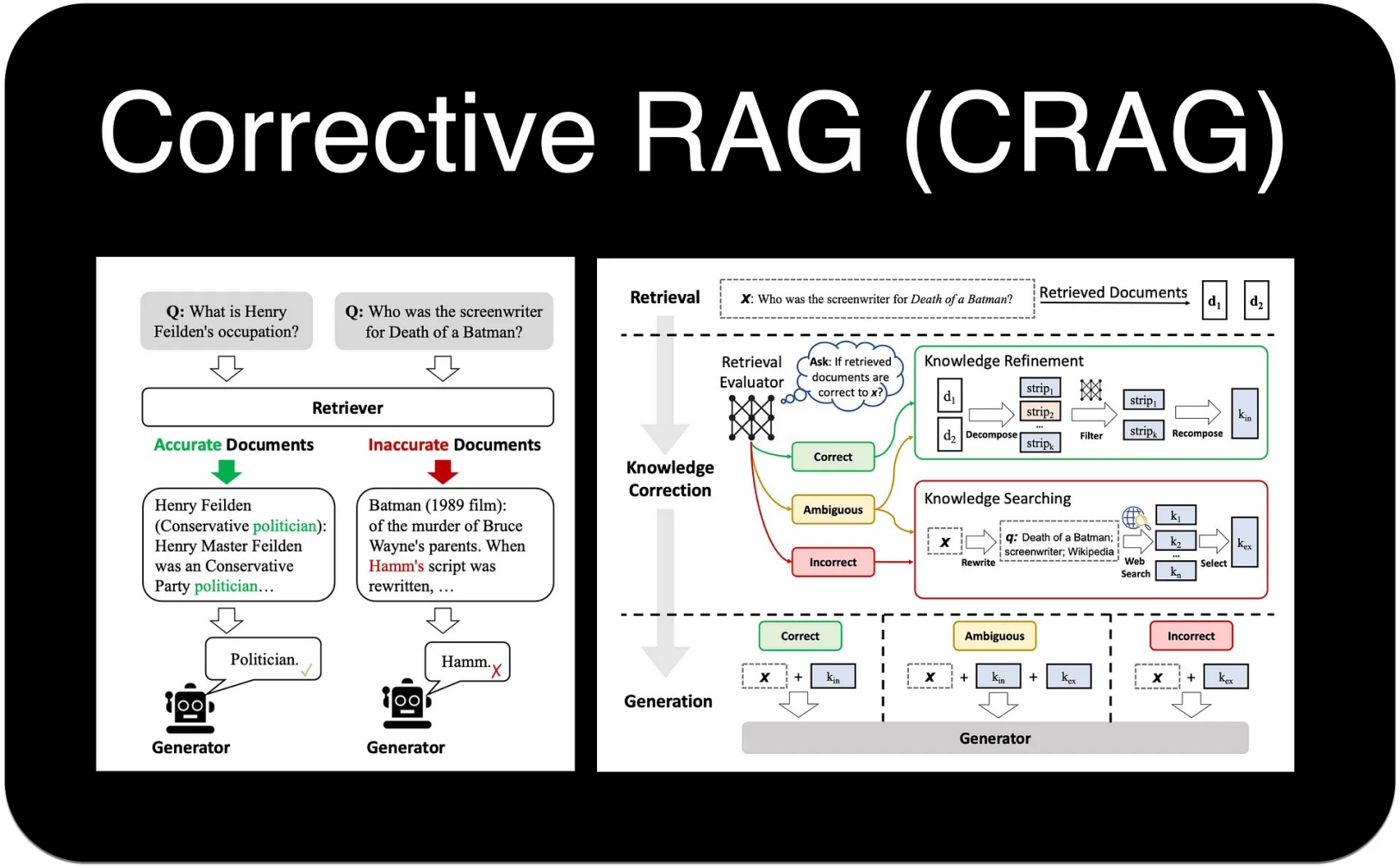
检索增强-比较

特性	RRF	RankLLM	Cross-Encoder
核心机制	融合多个排名	LLM 推理，生成排序列表	联合编码查询与文档，计算单一相关分
计算成本	低（简单数学计算）	中（API 费用与延迟）	高（N次模型推理）
交互粒度	无（仅排名）	概念/语义级	句子级（Query-Doc Pair）
适用场景	多路召回结果融合	高价值语义理解场景	Top-K 精排

检索后处理-压缩


RECOMP during inference
Input query x
when did they stop making the nissan xterra?
No retrieval (0 tokens)
Retrieved documents D
moved from Smyrna, Tennessee, to Nissan's facility in Canton, Mississippi. Early US models include X, S and PRO-4X, with a choice of 6-speed manual…
RALM (749 tokens)
Compressor
The Nissan Xterra is a front-engine, 2-wheel or 4-wheel drive, five-door …
Summary
RECOMP (58 tokens)
Blackbox LM M
2010
2015
2015
Retrieve
Compress
Prepend

检索后处理-矫正

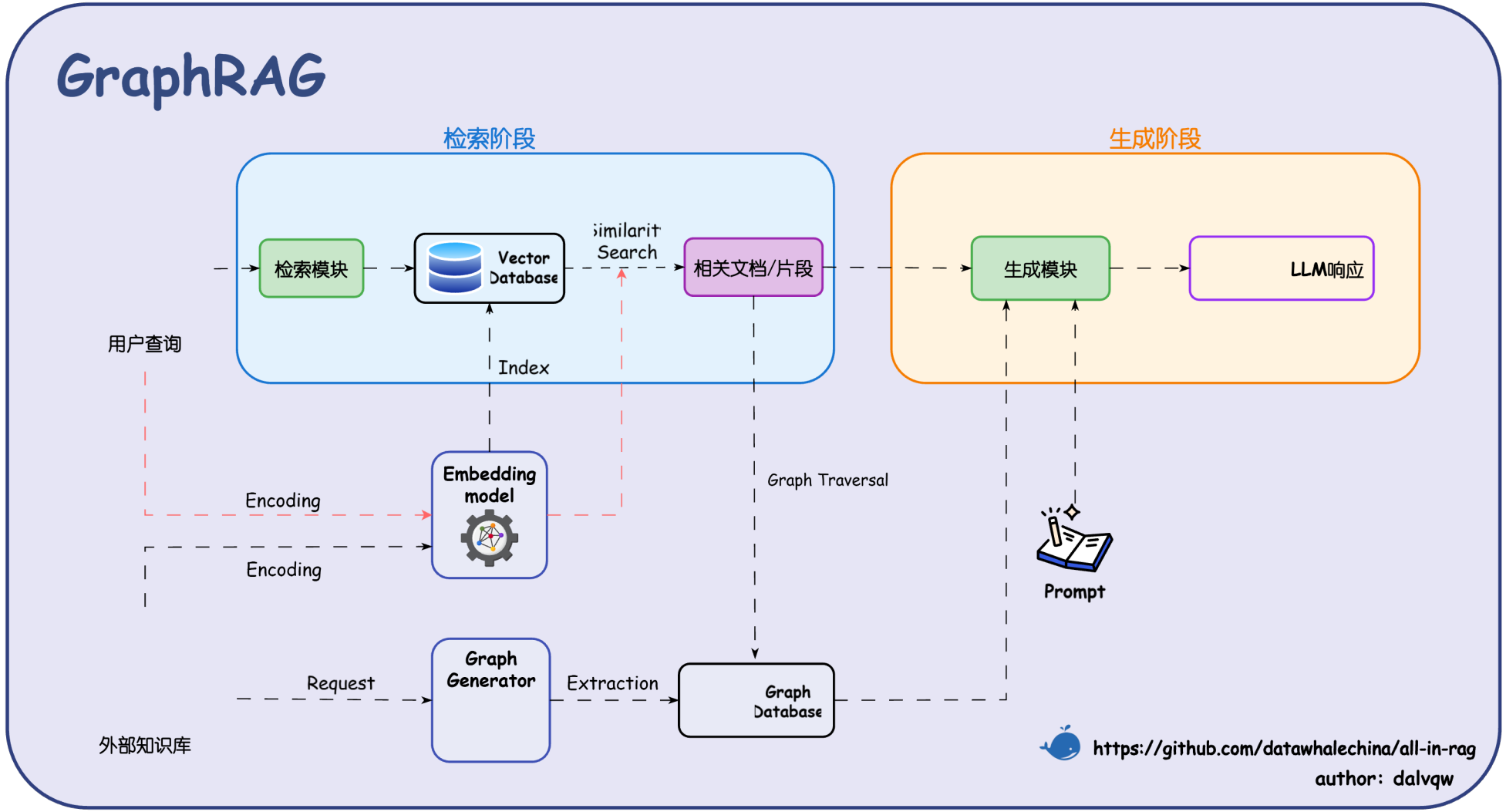

Corrective RAG (CRAG)
Q: What is Henry Feilden's occupation?
Q: Who was the screenwriter for Death of a Batman?
Retriever
Accurate Documents
Inaccurate Documents
Henry Feilden (Conservative politician): Henry Master Feilden was an Conservative Party politician…
Batman (1989 film): of the murder of Bruce Wayne's parents. When Hamm's script was rewritten, …
Politician.
Hamm.
Generator
Generator
Retrieval
x: Who was the screenwriter for Death of a Batman?
Retrieved Documents
Retrieval Evaluator
Ask: If retrieved documents are correct to x?
Knowledge Refinement
Decompose
Filter
Recompose
Knowledge Correction
Correct
Ambiguous
Incorrect
Knowledge Searching
Rewrite
q: Death of a Batman; screenwriter; Wikipedia
Web Search
Select
Generation
Correct
Ambiguous
Incorrect
Generator

中国科学院大学
University of Chinese Academy of Sciences

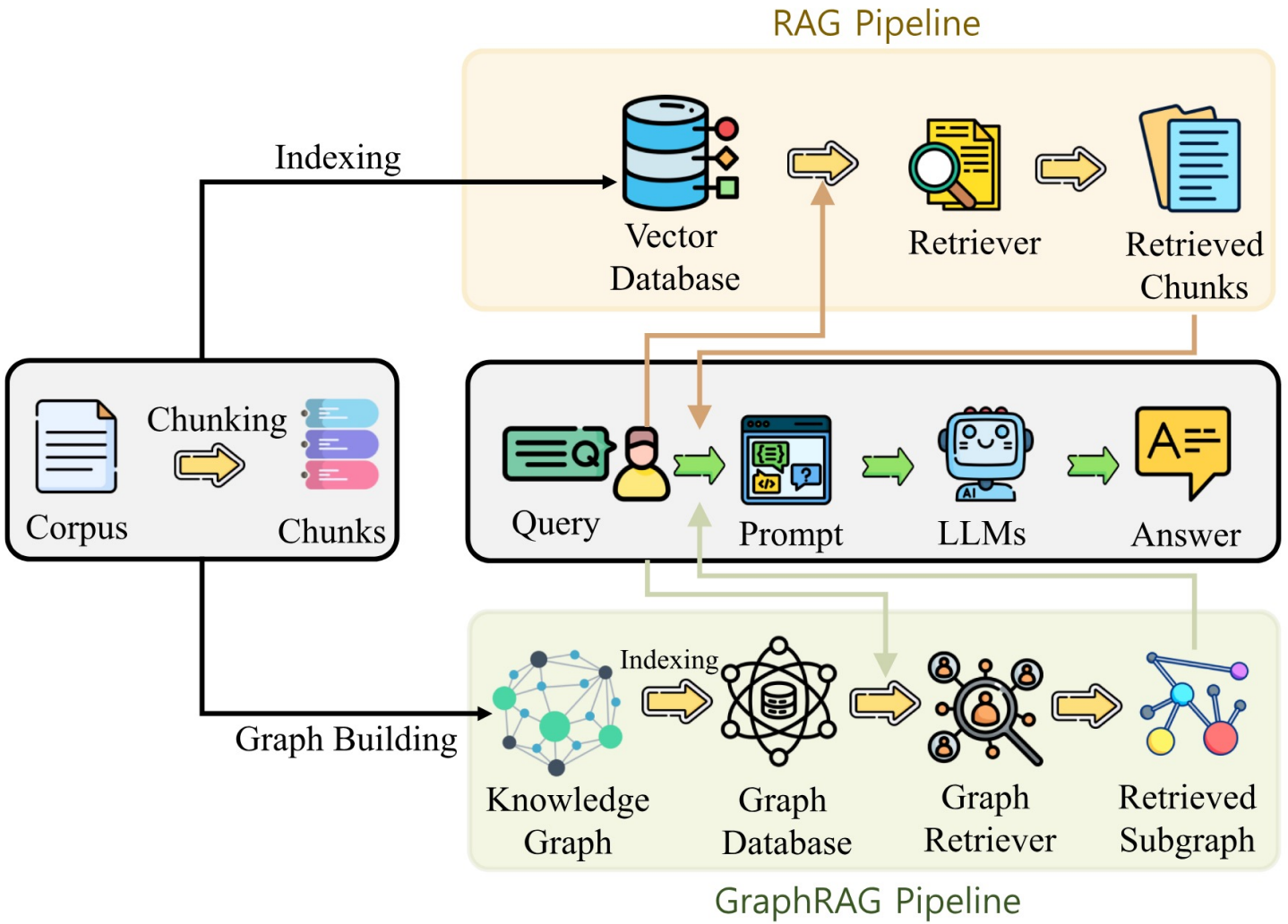
目录

原始RAG的缺陷

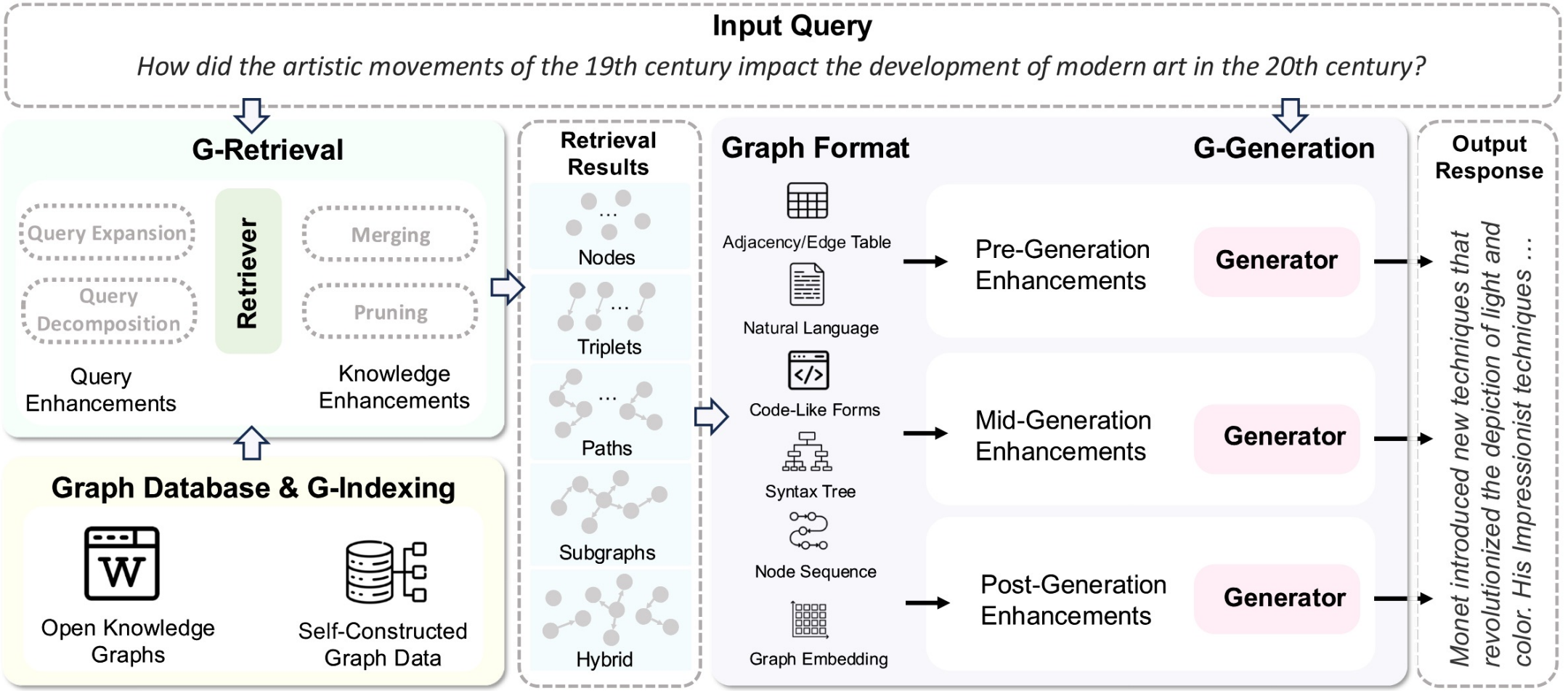
Graph RAG


GraphRAG
检索阶段
生成阶段
检索模块
Vector Database
similarit
Search
相关文档/片段
生成模块
LLM响应
用户查询
Index
Embedding model
Encoding
Encoding
Graph Traversal
Prompt
Request
Graph Generator
Extraction
Graph Database
外部知识库
https://github.com/datawhalechina/all-in-rag
author: dalvqw

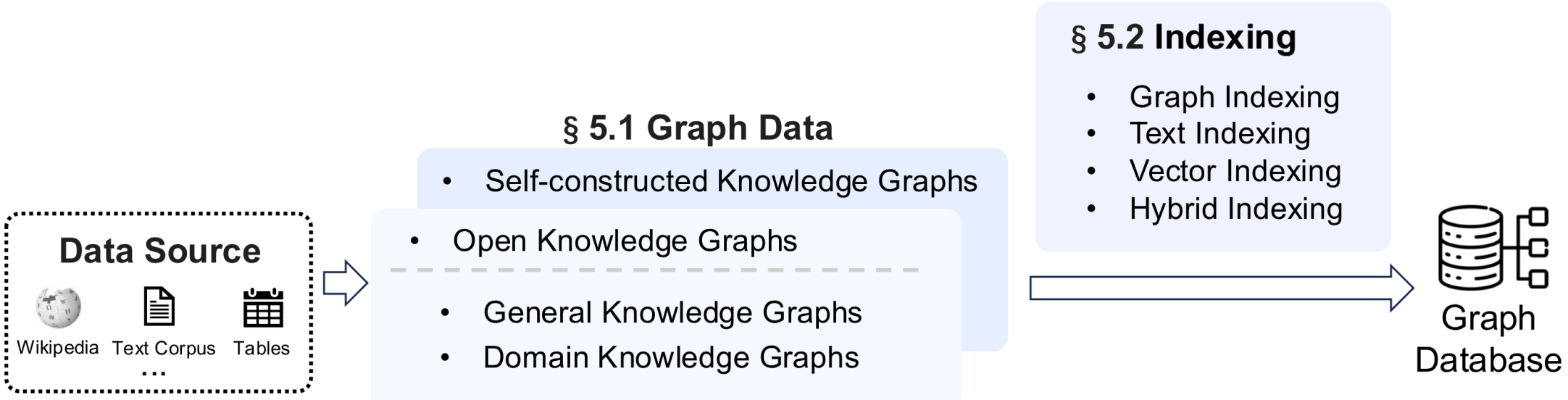
Graph RAG vs RAG



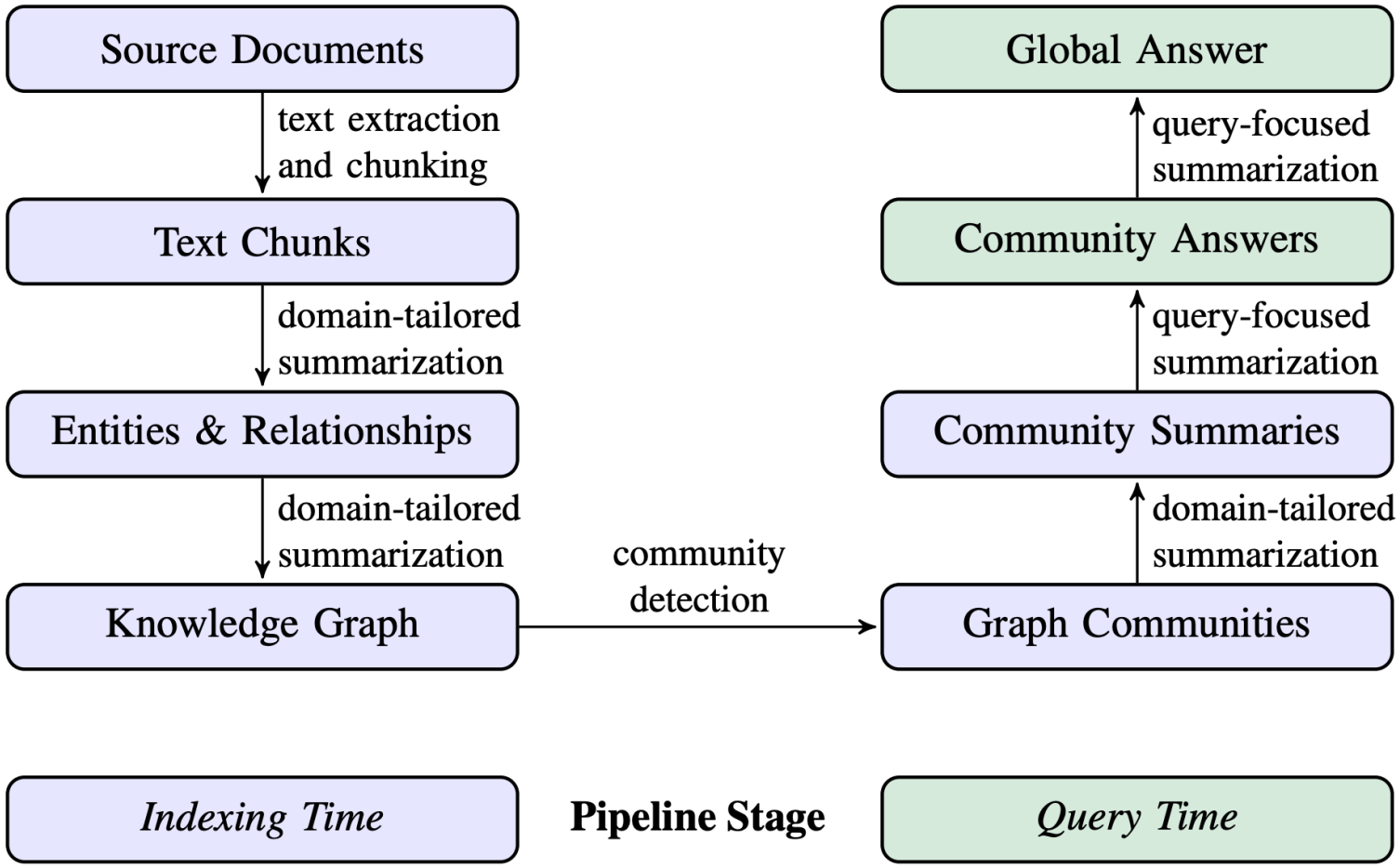
Graph RAG pipeline



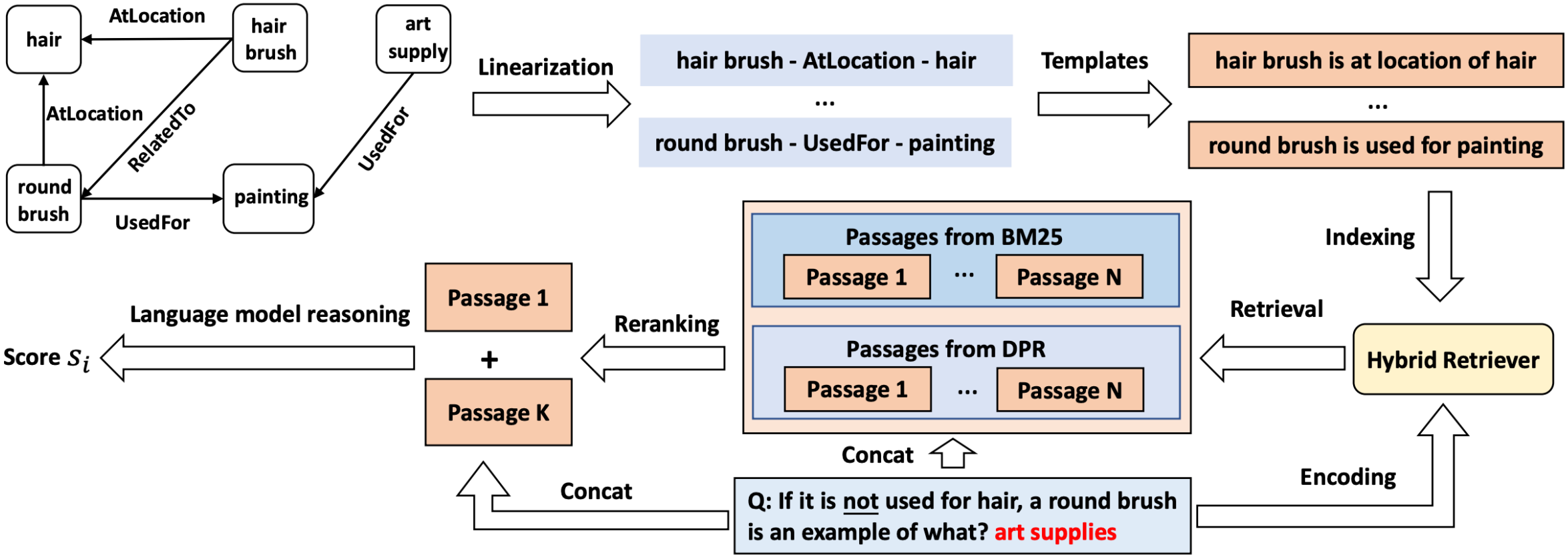
G-Indexing（索引构建）



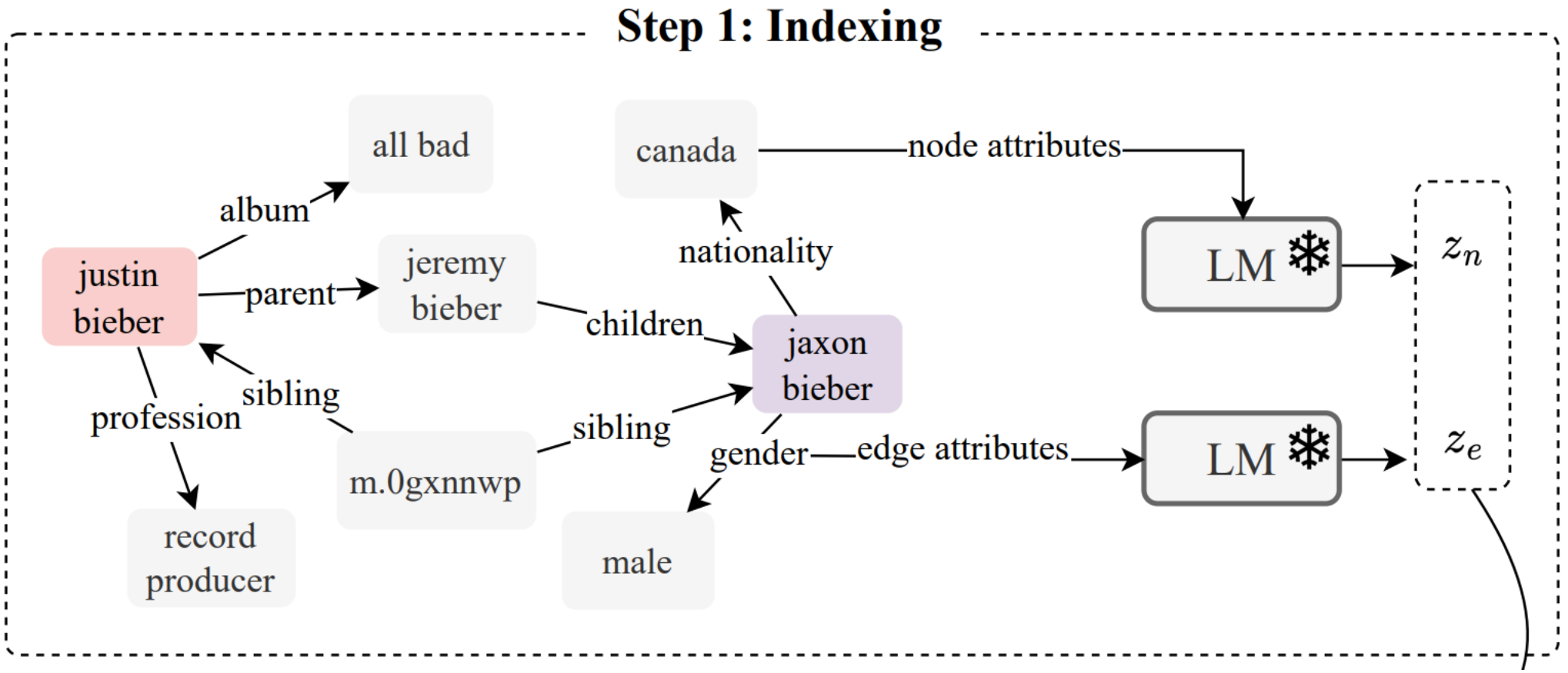
G-Indexing (Graph Indexing)



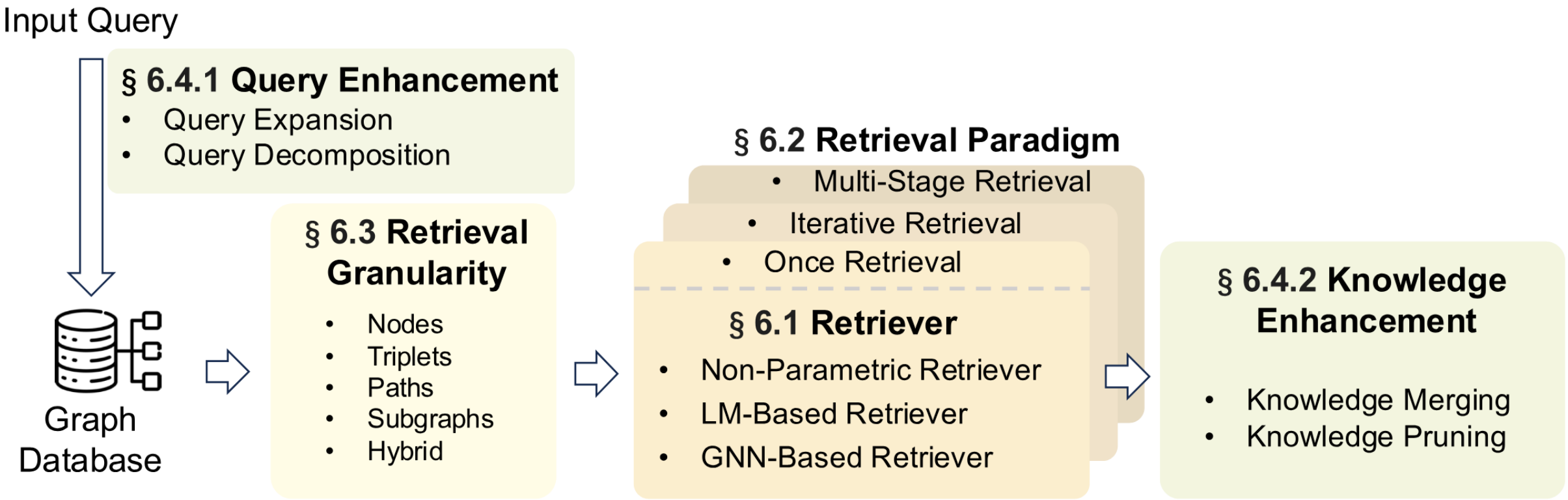
G-Indexing (text Indexing)



G-Indexing (vector Indexing)

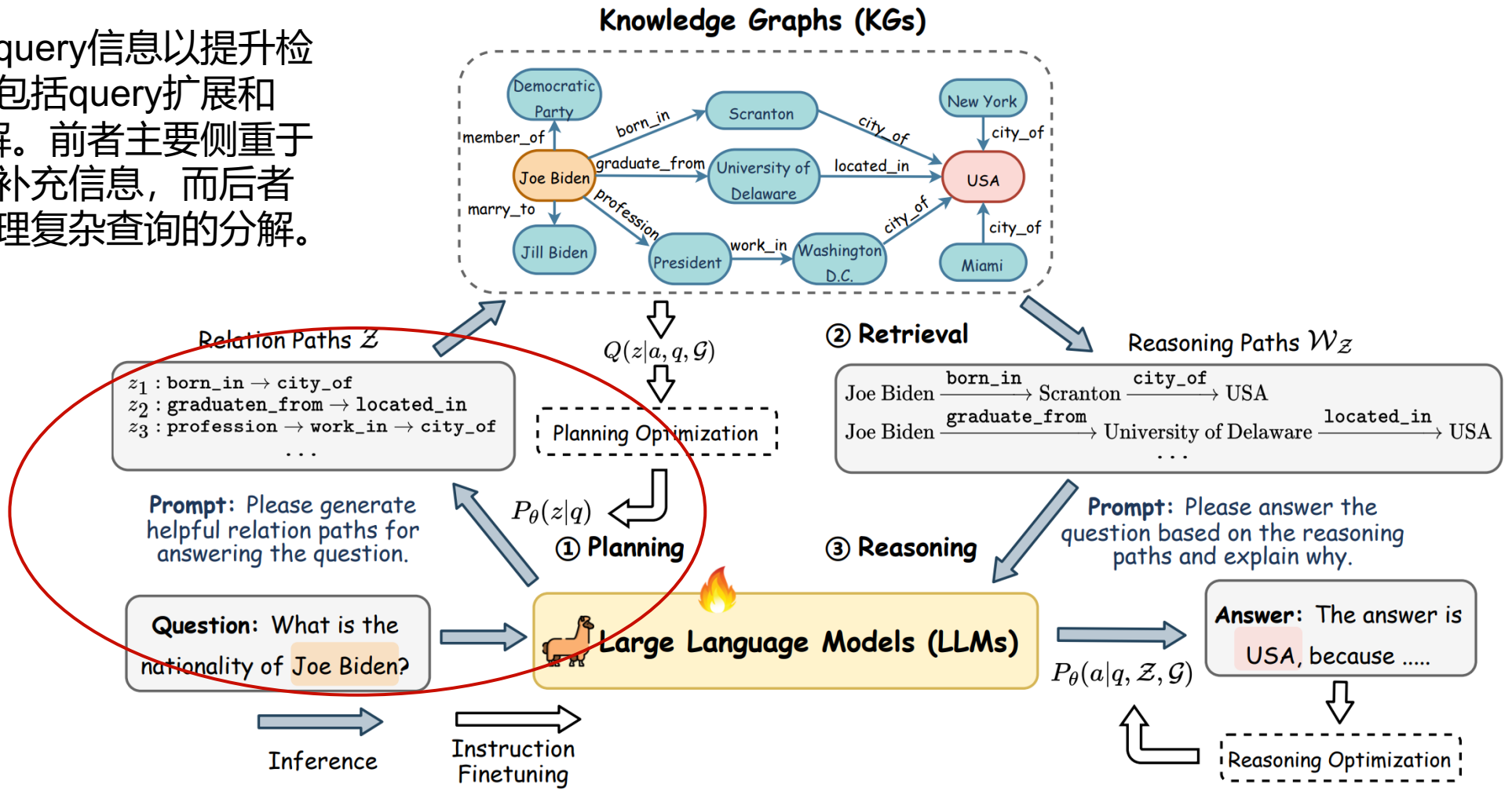


G-Retrieval（图检索）

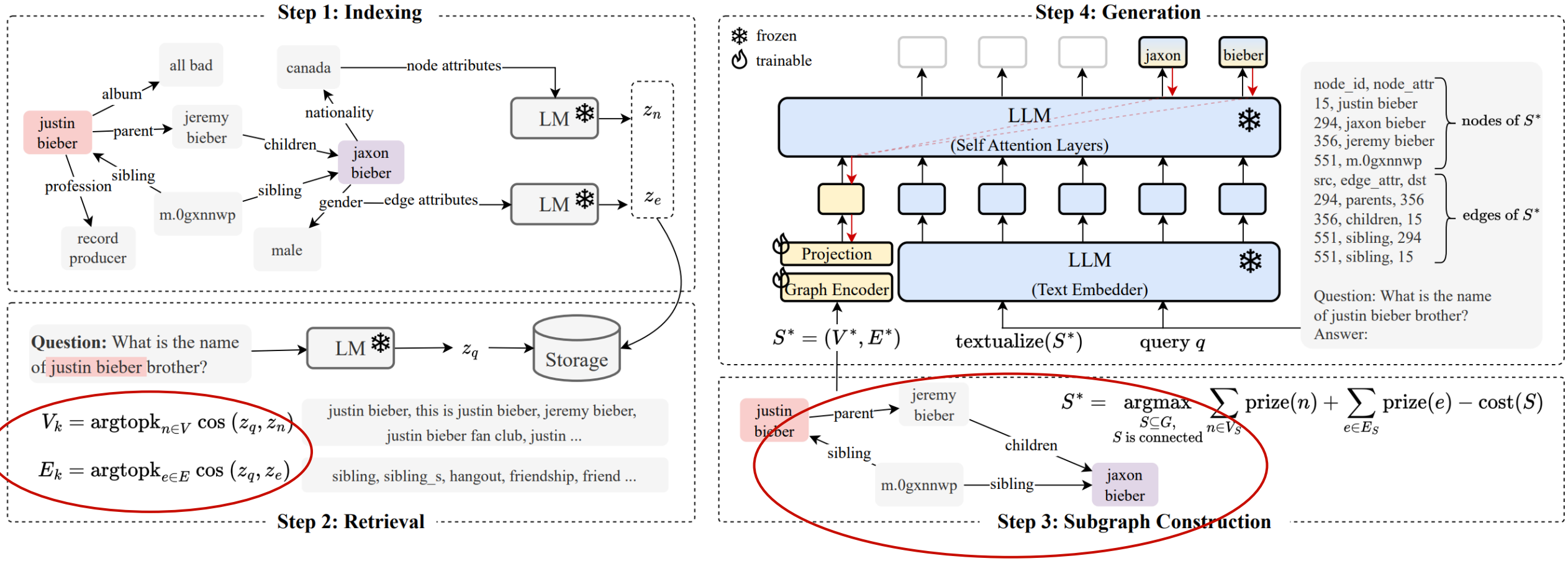


G-Retrieval（输入增强）

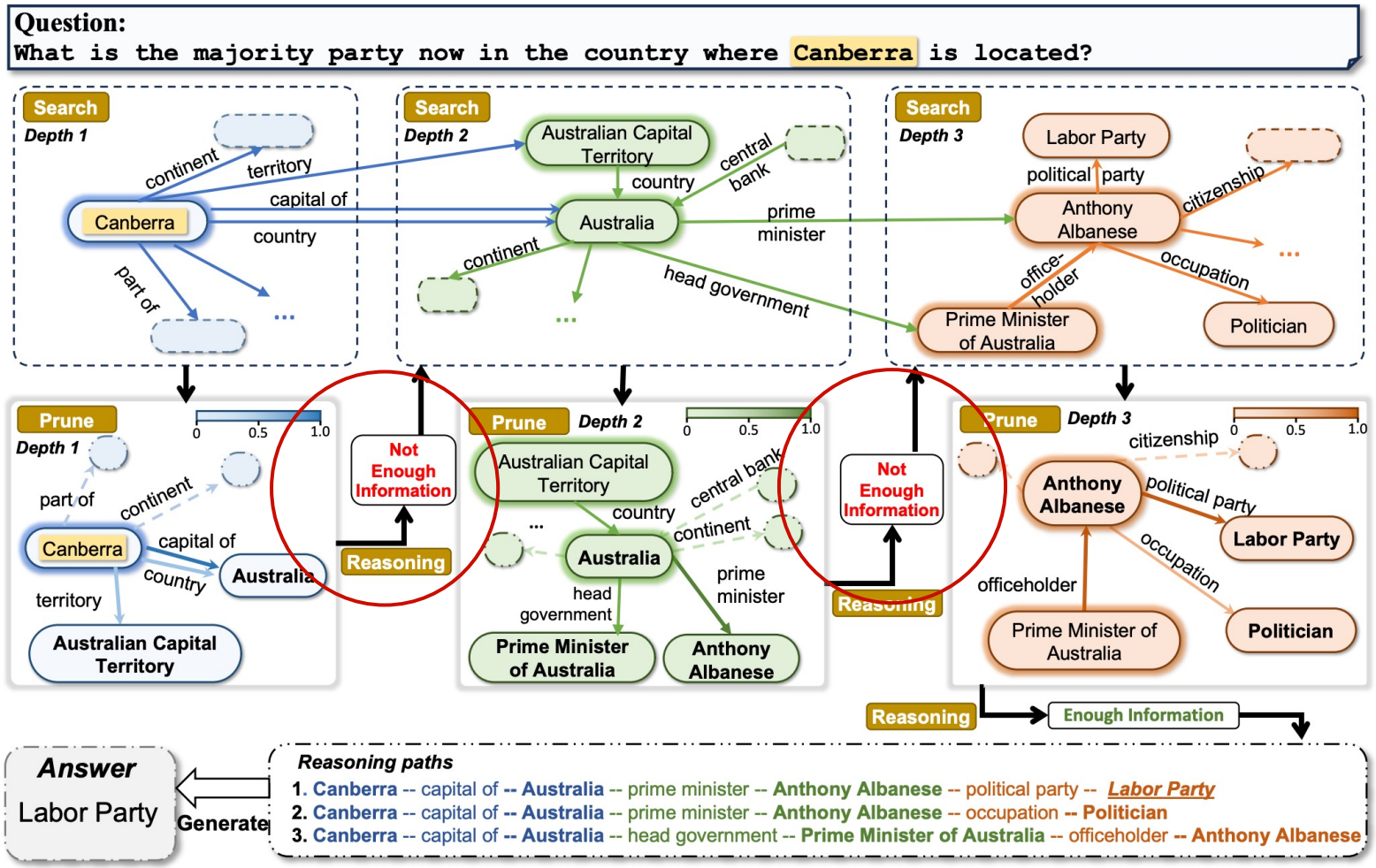
通过丰富query信息以提升检索效果。包括query扩展和query分解。前者主要侧重于为短查询补充信息，而后者则专门处理复杂查询的分解。



G-Retrieval（检索粒度）



G-Retrieval（检索范式）

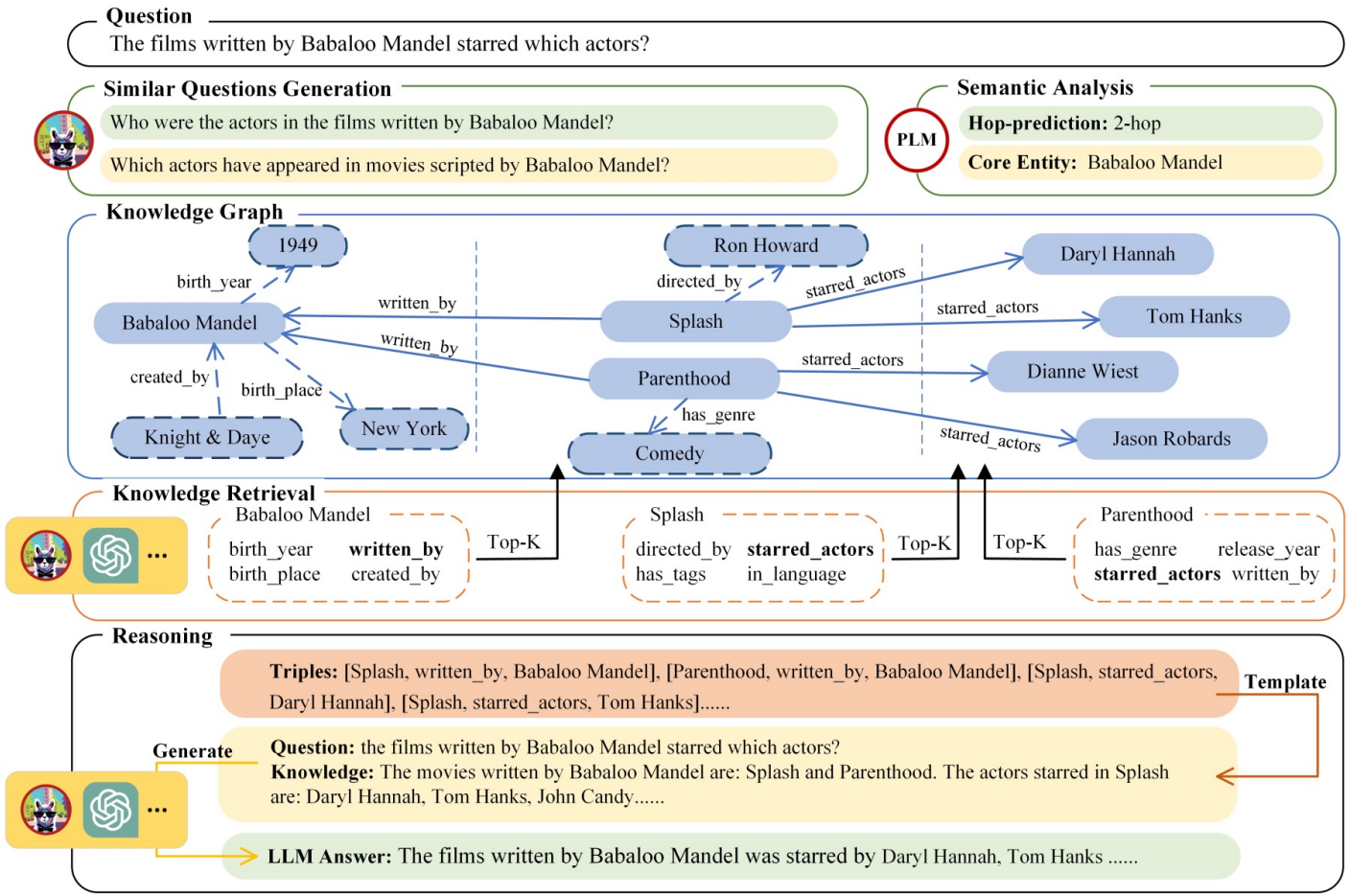


单轮检索

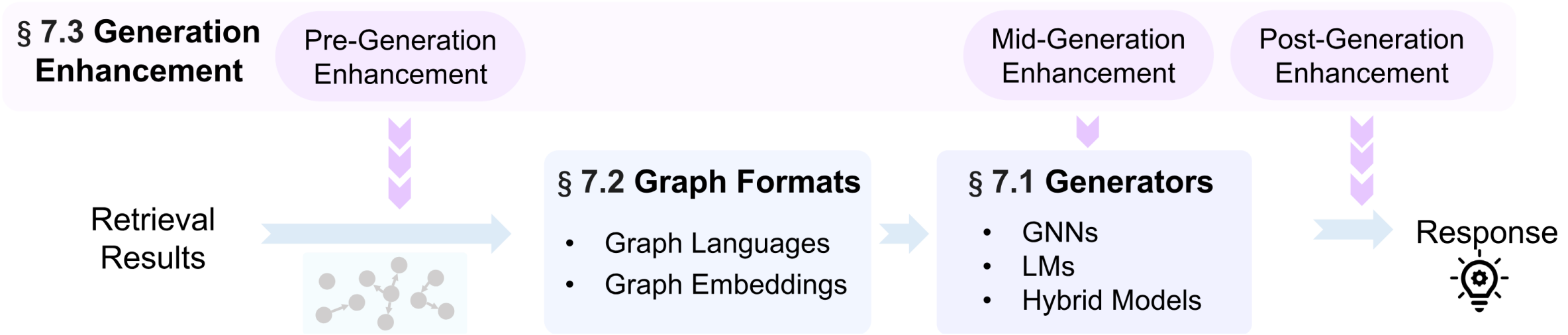
迭代检索

多阶段检索

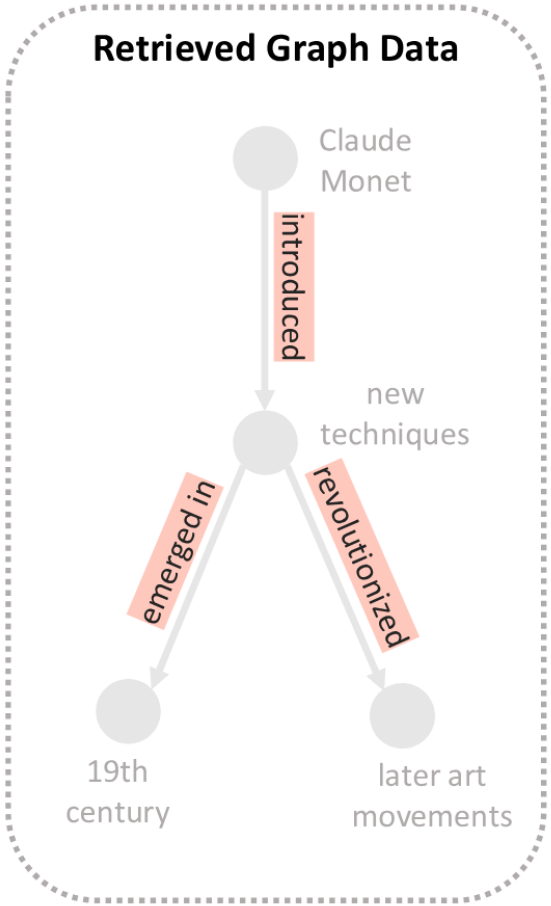
G-Retrieval（检索结果增强）



G-Generation （基于图检索的LLM生成）



怎么把图转换为LLM输入



transform

Adjacency/Edge Table

(Claude Monet, introduced, new techniques)
(new techniques, emerged in, 19th century)
(new techniques, revolutionized, later art movements)

Natural Language

Claude Monet introduced new techniques. These new techniques emerged in 19th century. These new techniques revolutionized later art movements.

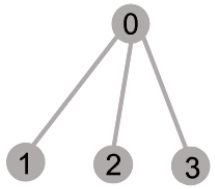
Node Sequence

Claude Monet→ new techniques → later art movements
Claude Monet→ new techniques → 19th century

Code-like Forms

```
< graphml >
< key id ="d0" for ="node" attr.name ="name" attr.type ="string" > </ key >
< key id ="d1" for ="edge" attr.name ="name" attr.type ="string" > </ key >
< graph id ="G" edgedefault ="directed" >
< node id ="n0" > < data key ="d0" > Claude Monet </ data > </ node >
< node id ="n1" > < data key ="d0" > new techniques </ data > </ node >
< node id ="n2" > < data key ="d0" > 19th century </ data > </ node >
< node id ="n3" > < data key ="d0" > later art movements </ data > </ node >
< edge id ="e0" source ="n0" target ="n1" > < data key ="d1" > introduced </ data > </ edge >
< edge id ="e1" source ="n1" target ="n2" > < data key ="d1" > emerged in </ data > </ edge >
< edge id ="e2" source ="n1" target ="n3" > < data key ="d1" > revolutionized </ data > </ edge >
</ graph >
</ graphml >
```

Syntax Tree

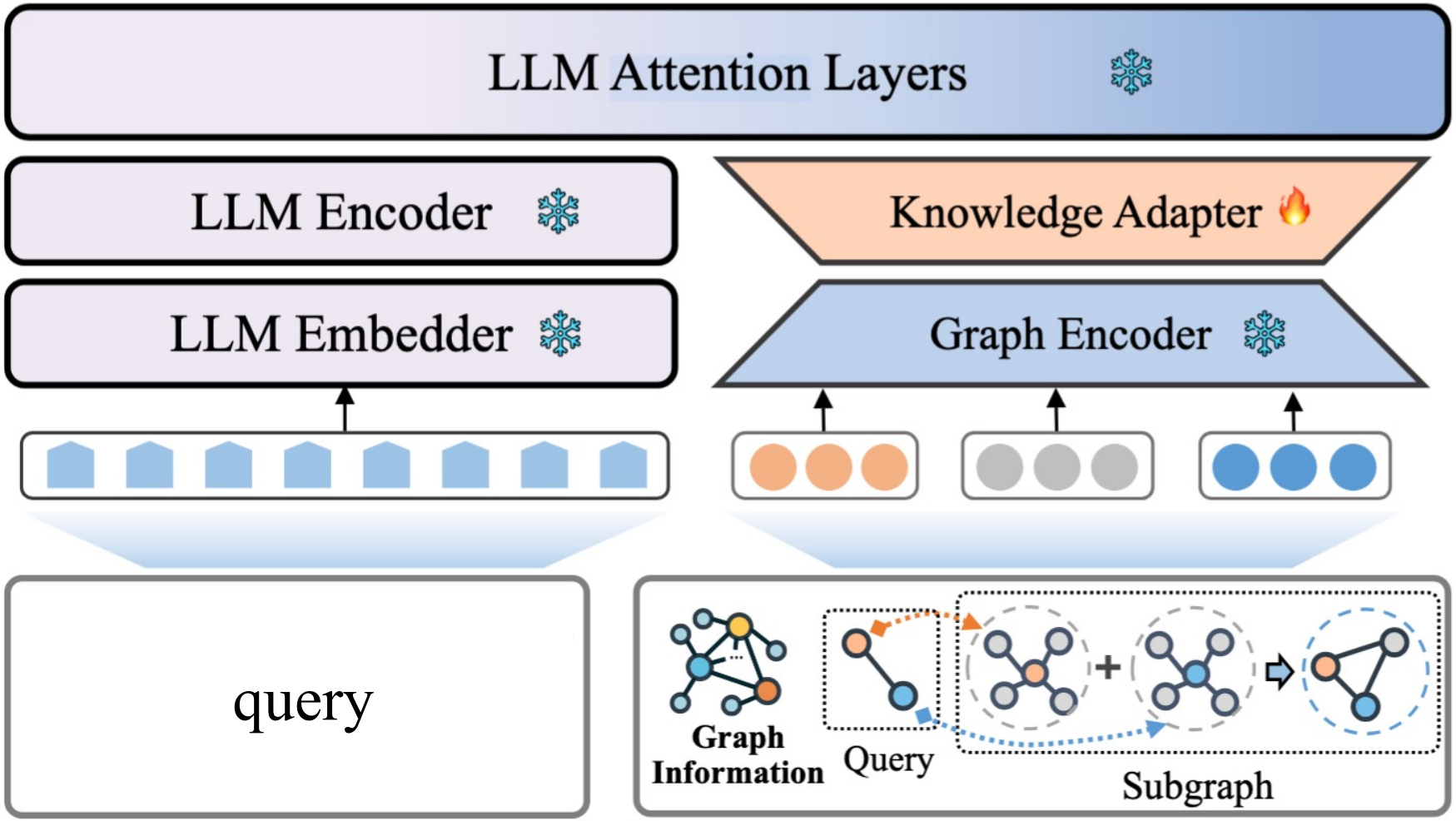


Tree Construction

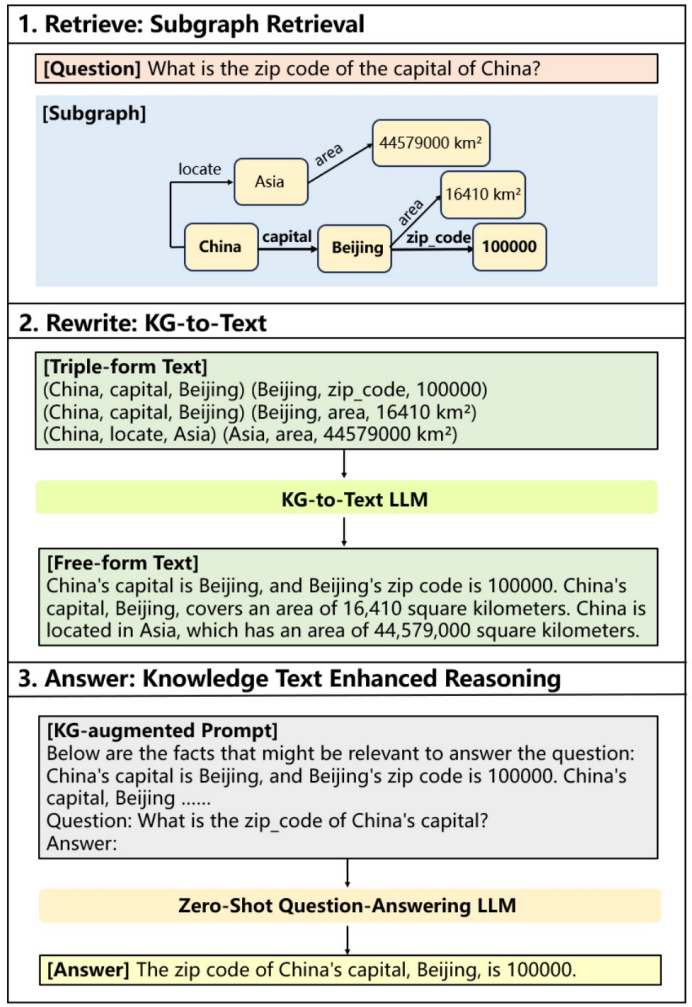
traverse

Node feature:
0: Claude Monet
1: new techniques
2: 19th century
3: later art movements
Edge feature:
(0,1): introduced
(0,2): emerged in
(0,3): revolutionized
Structure:
center node: 0
1st-hop: 1
2nd-hop: 2, 3

怎么把图转换为LLM输入



G-Generation （生成前增强）



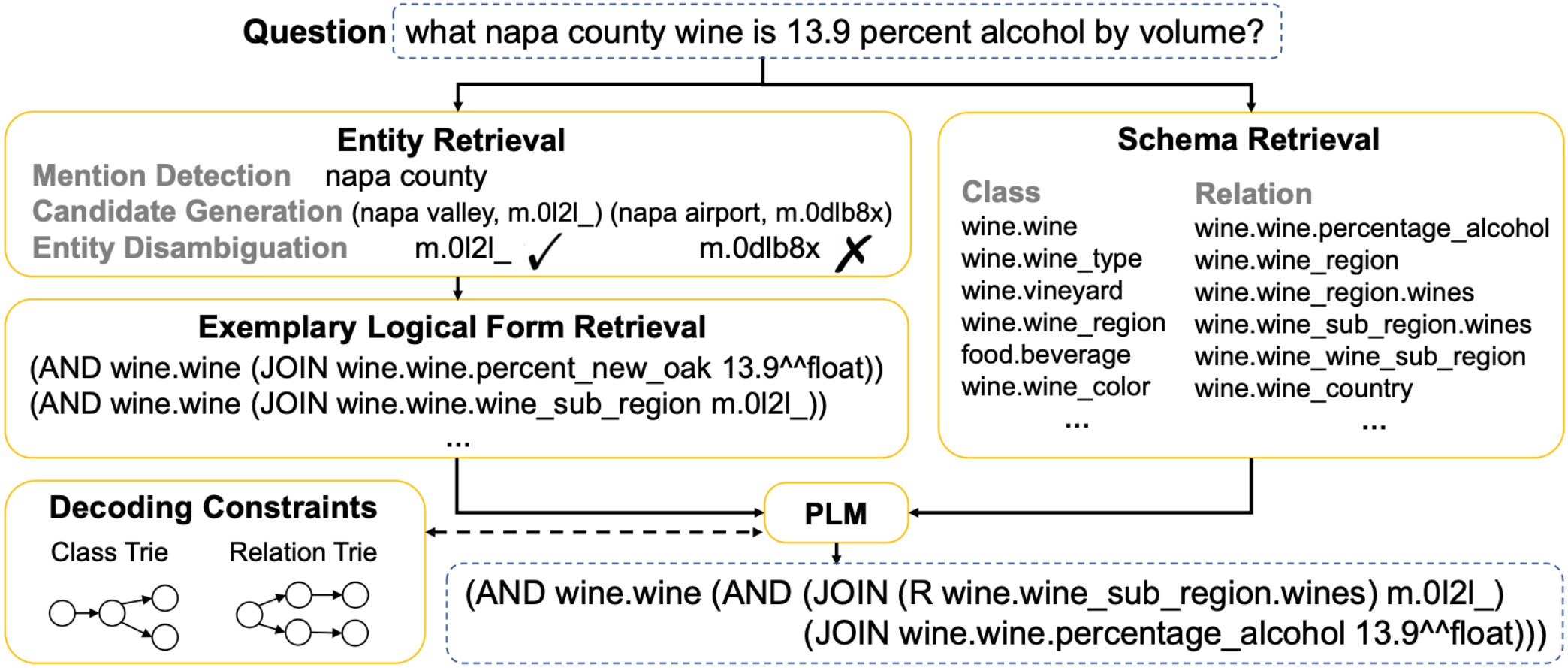
图的自然语言改写

query重写

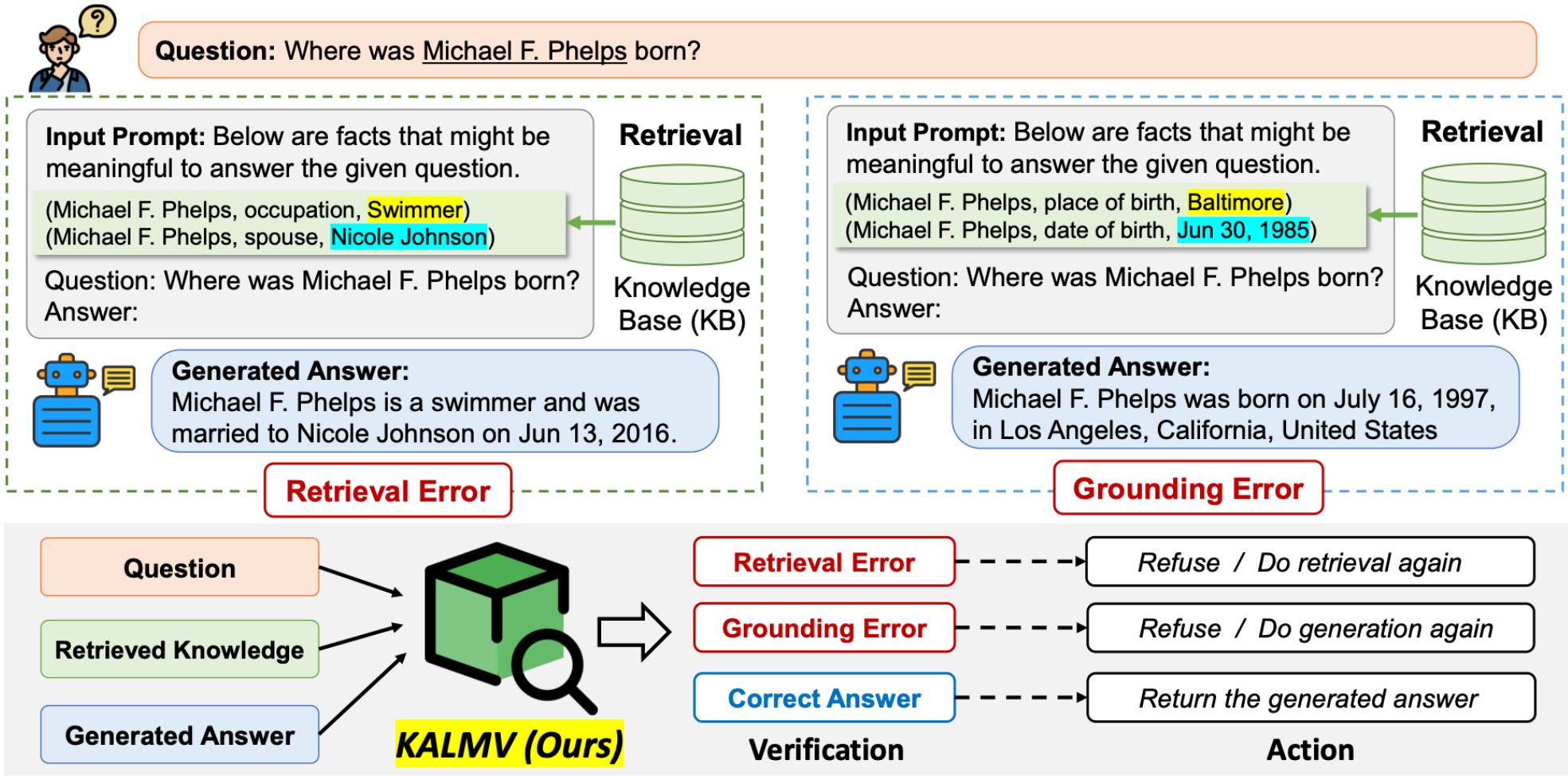
子图信息补充

推理规划

G-Generation（生成中增强）



G-Generation（生成后增强）



致 谢

- 胡玥、曹亚男、方芳：国科大《自然语言处理基础》
- 曹亚男、任昱冰：国科大《深度学习与自然语言处理概述》


THANKS
https://ictkc.github.io/teaching/2026spring-nlp