51CTO WOT

World Of Tech 2024

WOT全球技术创新大会

2024

智启新纪
慧创万物

51CTO WOT

生成式信息检索

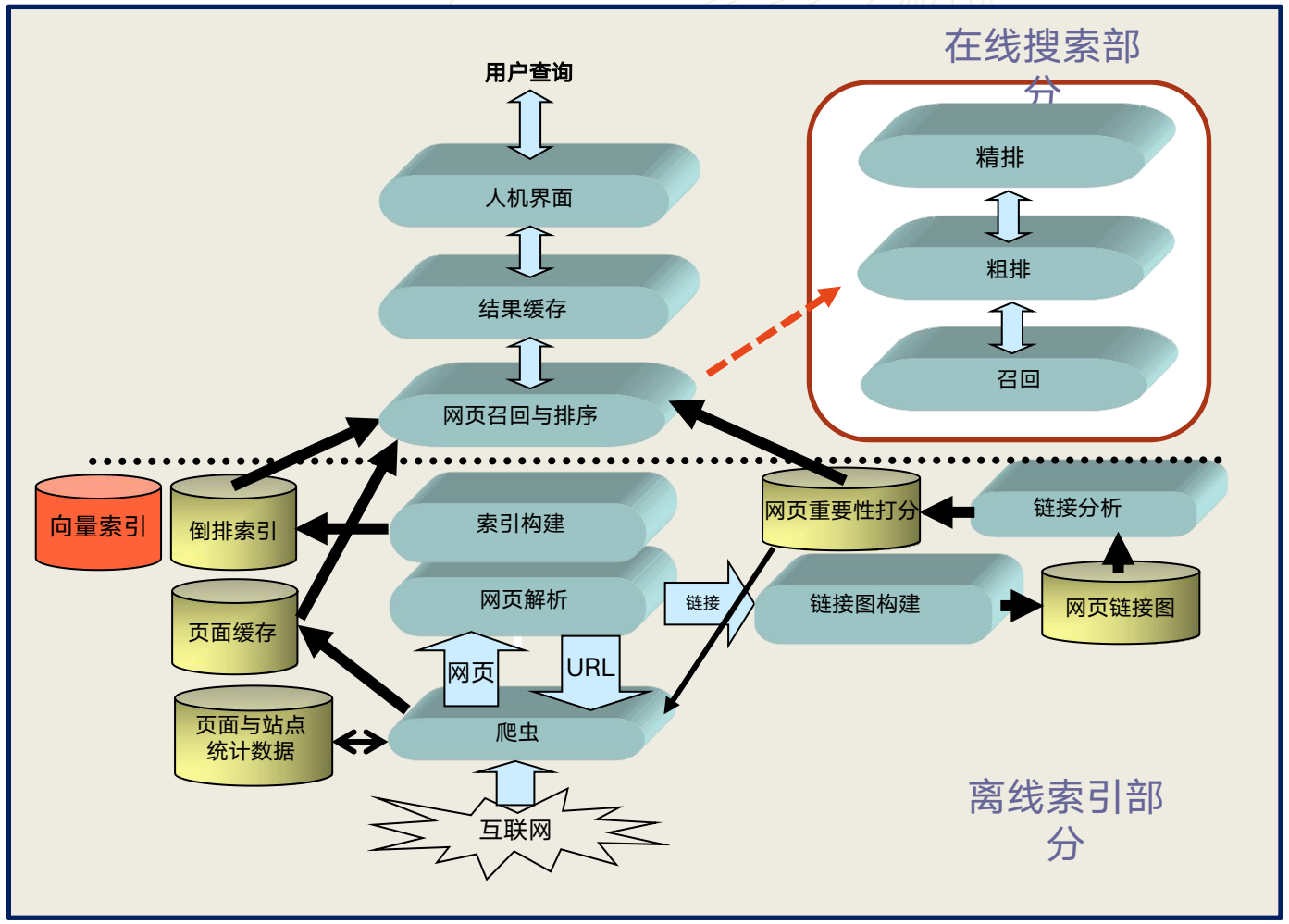
迈入智能信息获取新时代

窦志成
中国人民大学高瓴人工智能学院
教授、副院长

信息检索系统——搜索引擎

- 搜索引擎：最成功的大规模人工智能系统之一

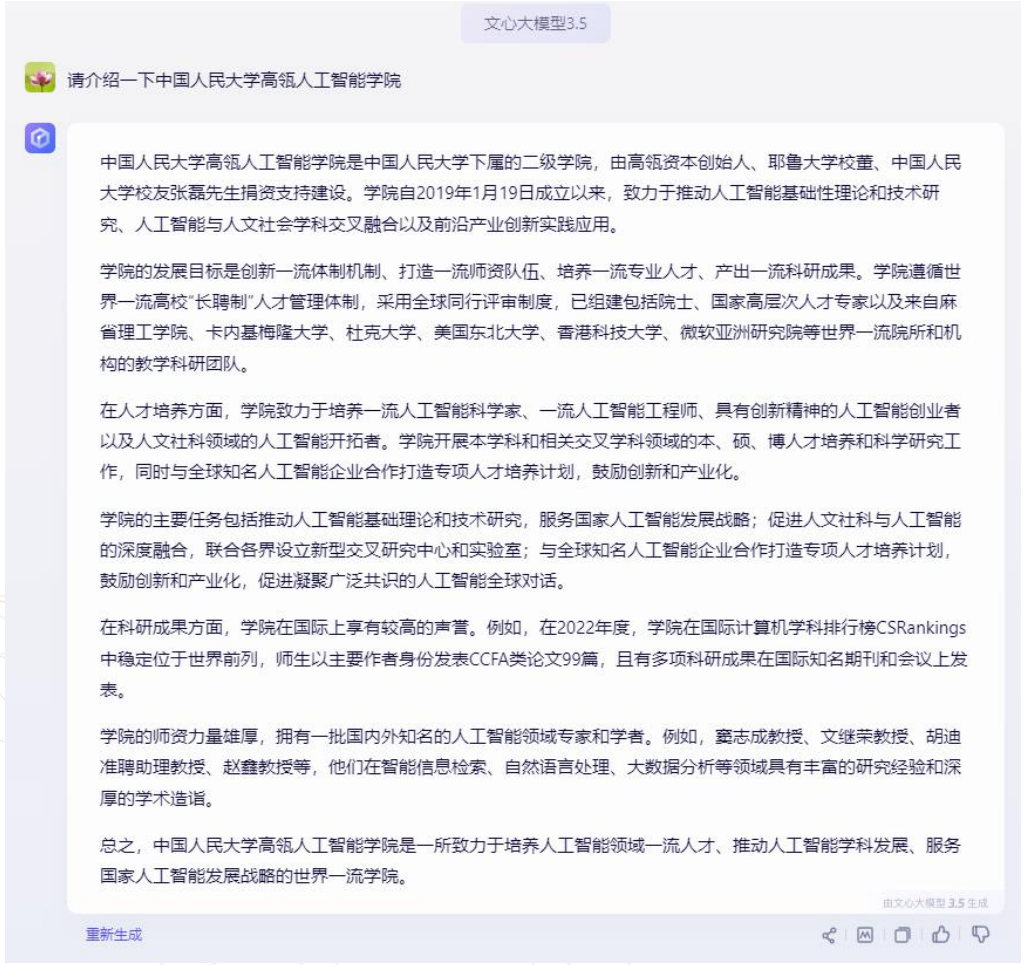
搜索引擎用户界面

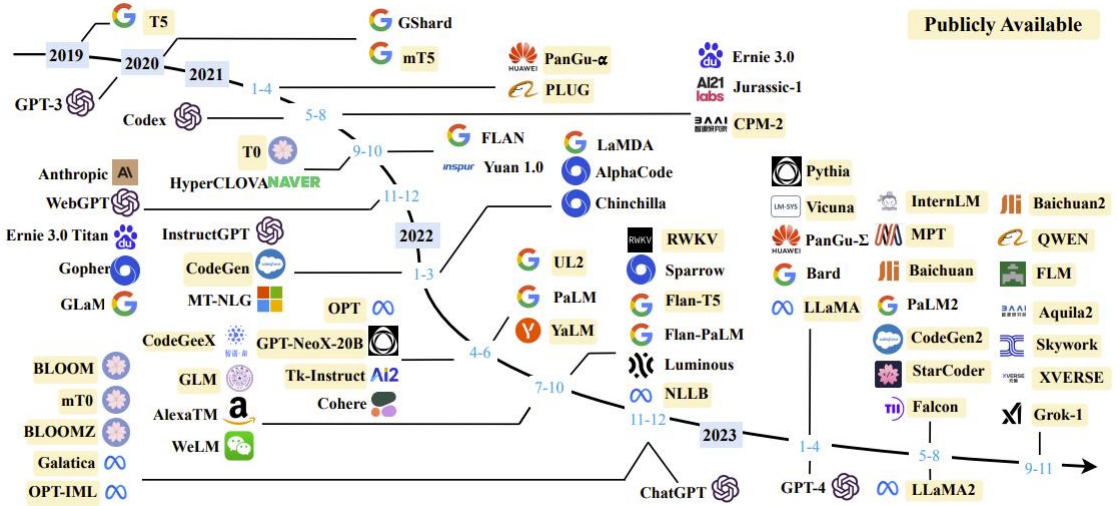


搜索引擎架构

大语言模型

- 2022年11月30日起，生成式大模型成为人们获取信息的一种新手段





来源：https://arxiv.org/pdf/2303.18223

最新！大模型备案117家，正式进入百模大战！

来源：https://zhuanlan.zhihu.com/p/690506768

生成式人工智能服务已备案信息（2024年3月）

序号	属地	模型名称	备案单位	备案号	备案时间
1	北京市	文心一言	北京百度网讯科技有限公司	Beijing-WenXinYiYan-20230821	2023/8/31
2	北京市	智谱清言（ChatGLM）	北京智谱华章科技有限公司	Beijing-ChatGLM-20230821	2023/8/31
3	北京市	云雀大模型	北京抖音信息服务有限公司	Beijing-YunQue-20230821	2023/8/31
4	北京市	百应	北京百川智能科技有限公司	Beijing-BaiYing-20230821	2023/8/31
5	北京市	紫东太初大模型开放平台	中国科学院自动化研究所	Beijing-ZiDongTaiChu-20230821	2023/8/31
6	上海市	abab	上海稀宇科技有限公司	Shanghai-Abab-20230821	2023/8/31
7	上海市	日日新	上海商汤智能科技有限公司	Shanghai-RiRiXin-20230821	2023/8/31
8	上海市	书生·通语	上海人工智能创新中心（上海人工智能实验室）	Shanghai-ShuShengPuYu-20230821	2023/8/31
9	安徽省	星火认知大模型	科大讯飞股份有限公司	Anhui-XingHuoRenZhiDaMoXing-20230823	2023/9/4
10	天津市	360智脑大模型	三六零科技集团有限公司	Tianjin-360ZhiNaoDaMoXing-20230831	2023/9/11
11	浙江省	通义千问大模型	阿里巴巴达摩院（杭州）科技有限公司	ZheJiang-TongYiQianWen-20230901	2023/9/12
12	广东省	腾讯混元助手大模型	深圳市腾讯计算机系统有限公司	Guangdong-TencentHunyuan-20230901	2023/9/14

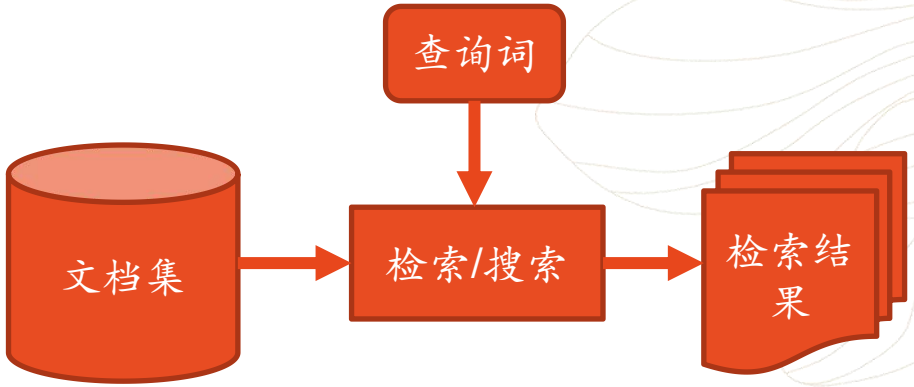
来源：https://www.gov.cn/lianbo/bumen/202404/content_6943924.htm

大语言模型

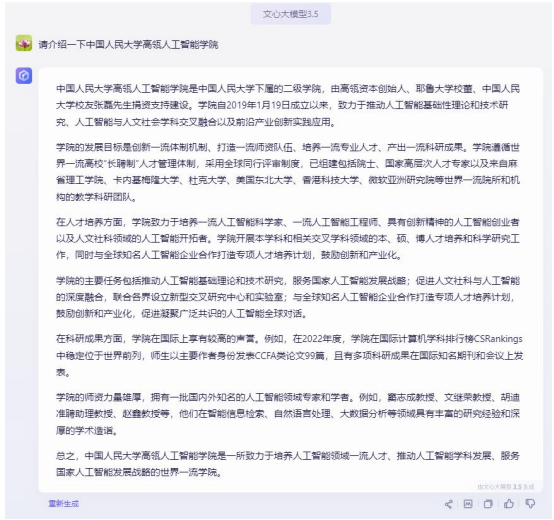
- 但是，大语言模型也有自己的问题
- 幻觉问题、知识缺陷、时效性低、套话连篇

检索与大模型

- **检索/搜索**
  - 以索引为核心
  - 高效快速地从海量文档中检索出相关的文档或者信息片断
  - 缺点：无法有效“聚合”信息，无法创造



- **生成式大模型**
  - 一步到位：直接生成满足用户信息需求的内容，有创造力
  - 缺点：可能会存在虚假、过时和错误内容
  - 缺点：无法回答私有问题



- **检索+大模型**

大模型+检索→生成式信息检索（GenIR）

- 生成式大模型与检索系统紧密结合（例如，New Bing）
  - 信息检索范式迁移：匹配式信息检索 到 生成式信息检索
  - 检索作为大模型工具，为大模型提供外部知识，提升生成质量

生成式信息检索
Generative Information Retrieval

大模型+检索→生成式信息检索（GenIR）

- 几种典型的生成式检索范式
  - Discriminative Retriever + Discriminative Reranker + **Generated/Synthetic Training Data** → **LLM Enhanced IR**
  - Discriminative Retriever + **Generative Reranker** → **LLM Enhanced IR**
  - Discriminative Retriever + Discriminative/**Generative Reranker**+ **Generative Reader** → **Retrieval Augmented Generation (RAG)**
  - Generative Retriever **(+ Generative Reranker+ Generative Reader)** → Model-based IR (Generative Retrieval)

生成式信息检索 示例

检索增强大模型

请介绍一下高瓴人工智能学院

找到了 9 篇资料作为参考：

高瓴人工智能学院是中国人民大学下属的二级学院，由高瓴资本创始人兼CEO、耶鲁大学校董、中国人民大学校友张磊先生捐资支持建设。学院成立于2019年1月19日，旨在承担人工智能学科的规划与建设，开展本学科和相关交叉学科领域的本、硕、博人才培养和科学研究工作。

学院的愿景是打造一所能够影响和塑造未来人工智能时代的世界一流学院，为全球思考并创造“智能而有温度”的未来。其人才培养目标是培养一流的人工智能科学家、工程师、具有创新精神的人工智能创业者以及人文社科领域的人工智能开拓者。

高瓴人工智能学院坚持“高起点、高水平、国际性、创新性”的发展方针，致力于建设世界一流的人工智能学科，提升学校的国际影响力和竞争力。学院在人工智能基础理论和技术研究方面展现出强劲的科研实力，2022年学院师生以主要作者身份发表（含录用待发表）CCF A类论文99篇，并且在大语言模型等前沿方向进行了全面、深入的探索性研究。

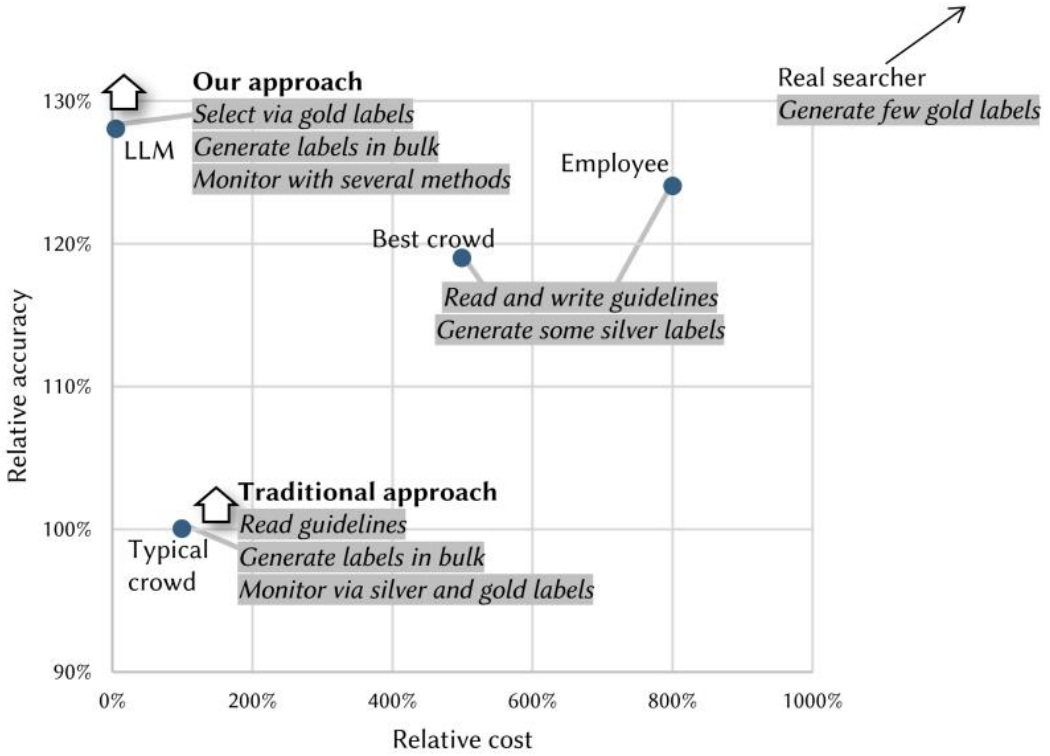
学院还注重校企合作，与华为、中国联通、快手等企业签署了研究项目合作协议，开展联合创新与长期合作。此外，学院开展贯通培养，与加拿大蒙特利尔大学计算机系开展双博士学位联培项目，并支持学生申请校级交换、派出项目，伙伴学校包括英国牛津大学、美国哥伦比亚大学等海外一流高校。

在人才培养方面，学院采用全员拔尖培养的模式，设计了本硕博一体化培养方案，允许高年级本科生提前选修研究生的课程。学院还注重科技人才的伦理和职业道德教育，鼓励学生选修人文和社会学科基础课程，并开展前沿交叉研究。

高瓴人工智能学院的毕业生就业质量一流，2023届毕业生就业率达100%，其中50%以上年薪超35万元。毕业生中约32%升学国内外一流高校，37%去往金融行业和大型央企，21%去往头部科技公司，10%的同学去往公务员岗位。

学院的成立和发展得到了社会的广泛关注和积极评价，被认为是中国人民大学在人工智能领域的一次重要布局，对于推动我国人工智能教育和研究具有重要意义。

大模型赋能检索



使用LLM标签重新训练Bing排名器的一部分，取得了显著的检索相关性提升。

来源：Large language models can accurately predict searcher preferences, Microsoft

报告内容

1 大模型赋能的信息检索

2 检索增强的大模型

3 生成式文档检索

报告内容

1 大模型赋能的信息检索

2 检索增强的大模型

3 生成式文档检索

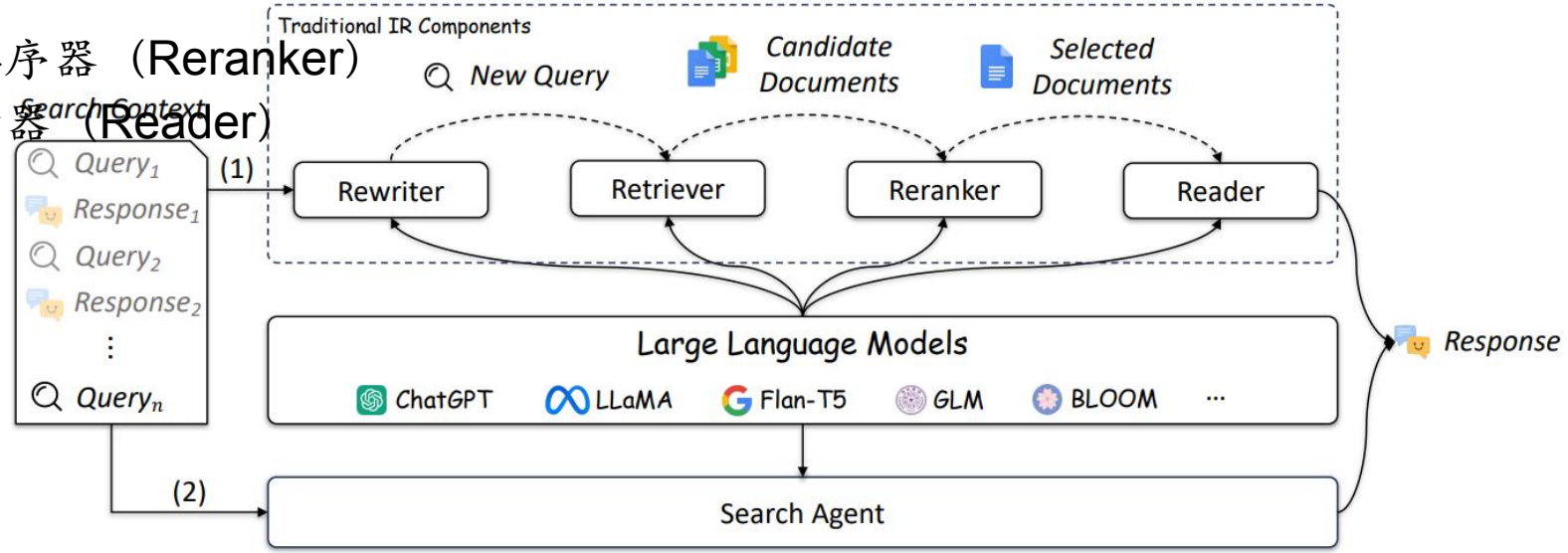
大模型赋能信息检索

[Submitted on 14 Aug 2023 (v1), last revised 19 Jan 2024 (this version, v3)]

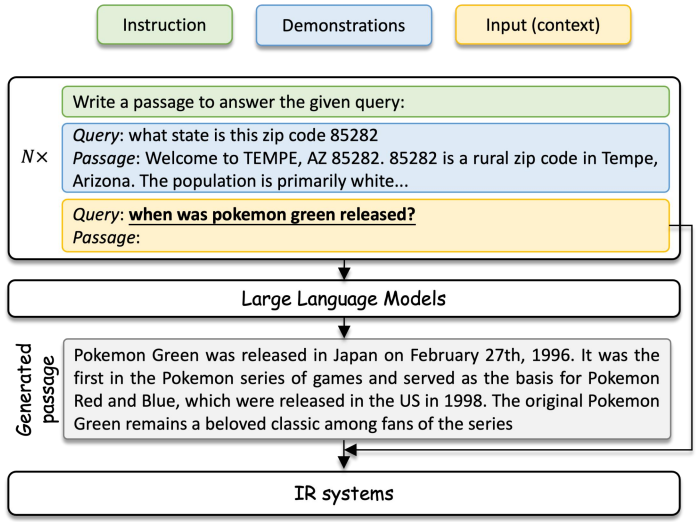
Large Language Models for Information Retrieval: A Survey

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, Ji-Rong Wen

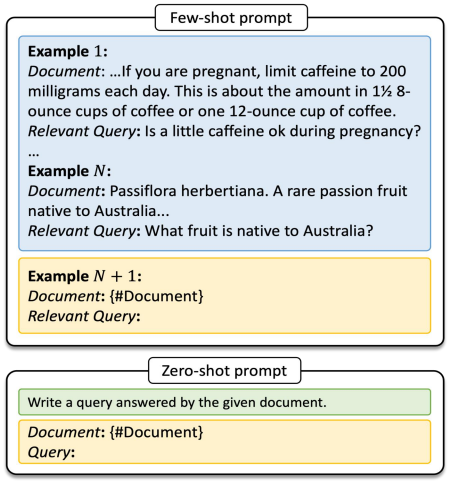
- LLM深刻影响IR系统的各个组件
  - ① 查询改写器（Query Rewriter）
  - ② 检索器（Retriever）
- 搜索智能体推动信息获取方式变革
  - ③ 重排序器（Reranker）
  - ④ 阅读器（Reader）

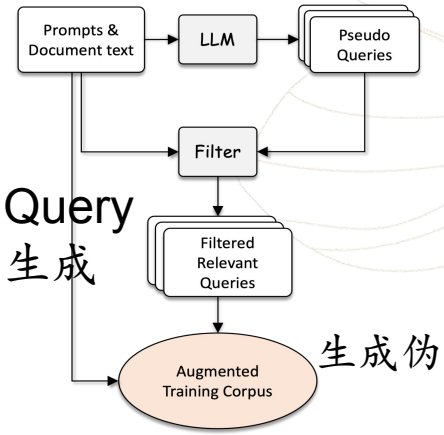


大模型赋能信息检索

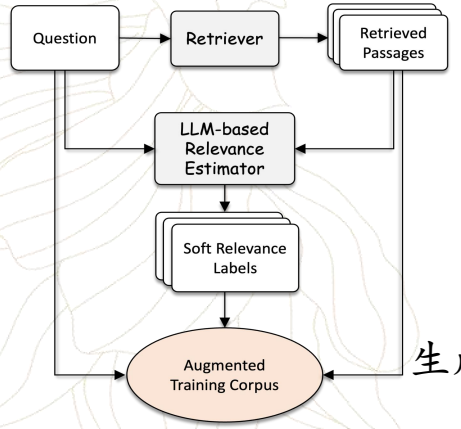


查询改写



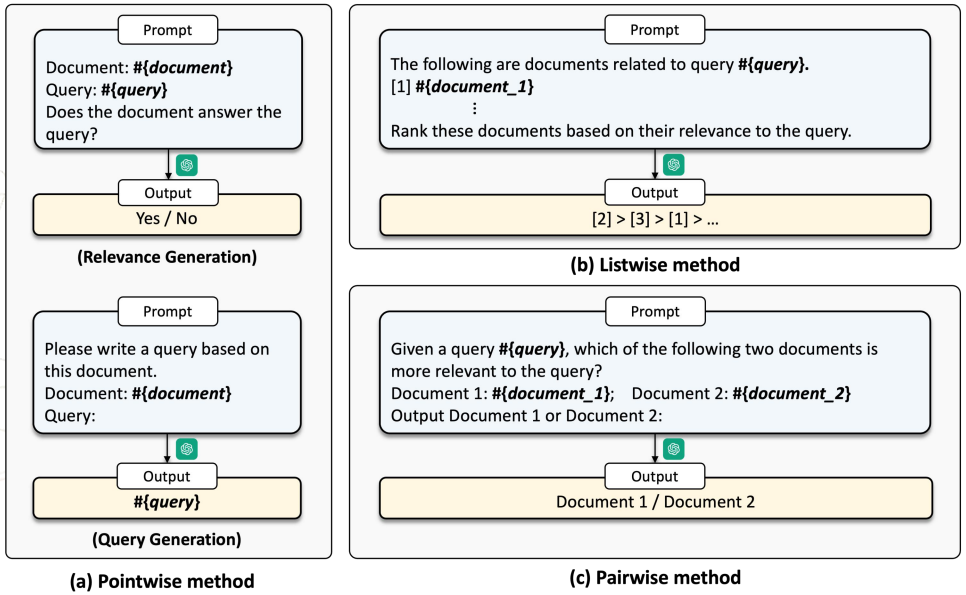


Framework of pseudo query generation



Framework of relevance label generation

检索器训练数据生成



提示大模型做重排

面向IR任务微调大模型：动机

- 问题：IR的效果高度依赖于大模型的指令理解能力
  - 在IR任务上，大模型并没有显著优于小模型
  - 每个环节分别设计和使用LLM
- 大模型的预训练中缺乏对IR概念的理解，已有指令微调数据集缺乏IR相关任务
  - 例如：查询、文档、相关性、用户意图等
- 解决方法 ⇒ **面向IR任务的LLM**

INTERS: Unlocking the Power of Large Language Models in Search with Instruction Tuning

Yutao Zhu[1], Peitian Zhang[1], Chenghao Zhang[1,2*], Yifei Chen[1,3*], Binyu Xie[1]
Zheng Liu[4], Ji-Rong Wen[1], and Zhicheng Dou[1†]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Computer Science, Beijing University of Posts and Telecommunications
[3]School of Artificial Intelligence, Nankai University, [4]Beijing Academy of Artificial Intelligence
yutaozhu94@gmail.com, dou@ruc.edu.cn

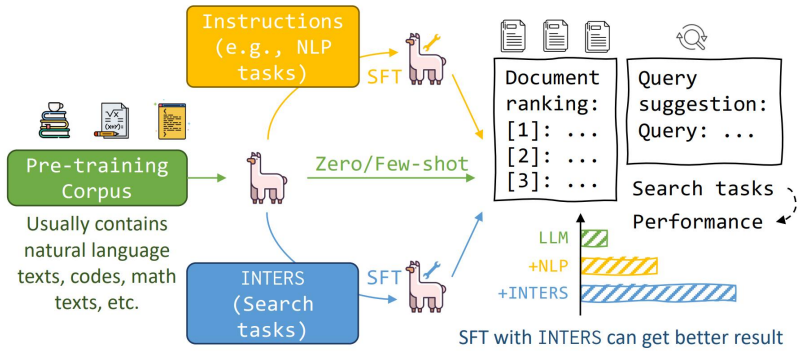
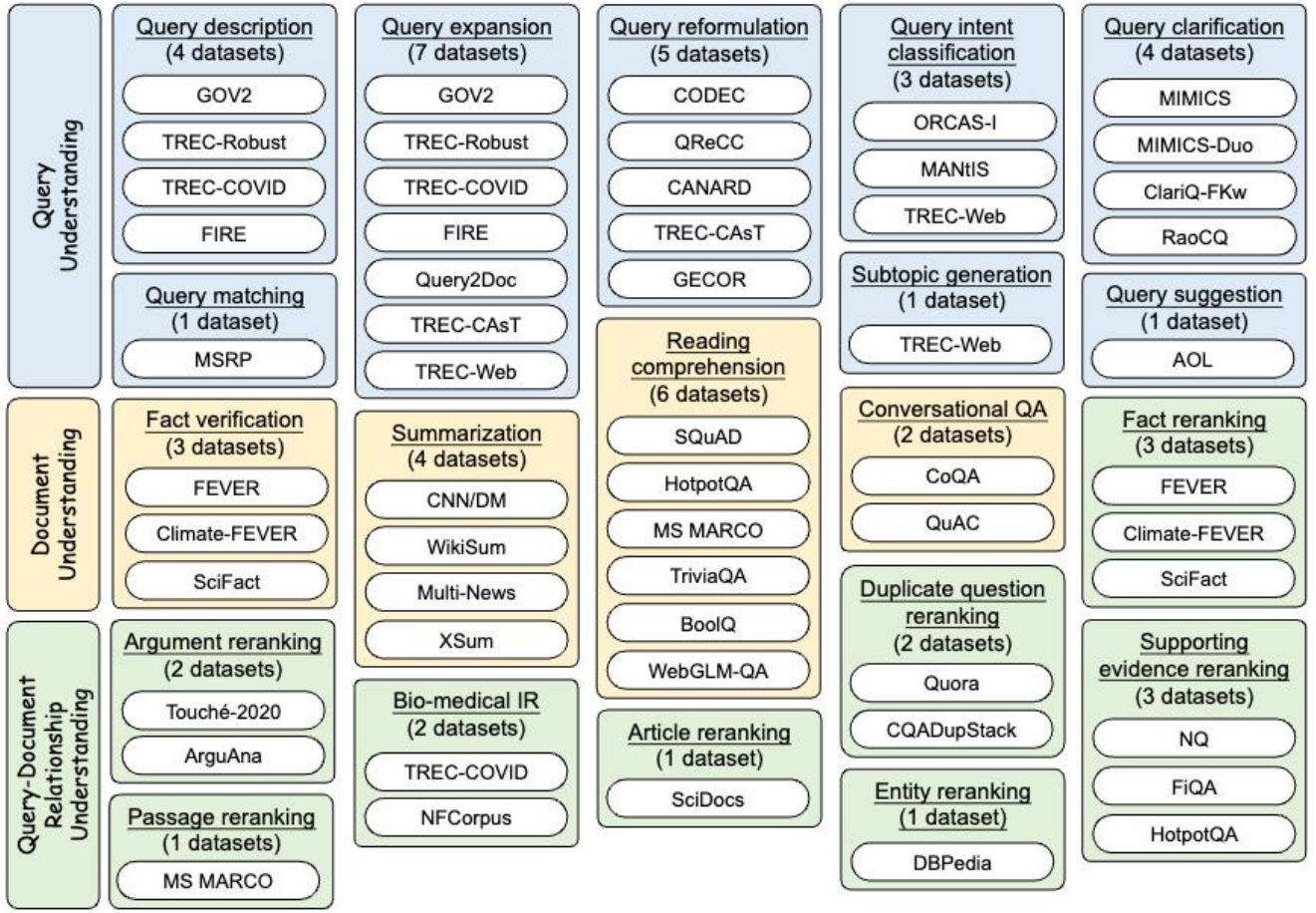


Figure 1: Compared with existing datasets, INTERS is designed specifically for search tasks.

Yutao Zhu, Peitian Zhang, Chenghao Zhang, Yifei Chen, Binyu Xie, Zhicheng Dou, Zheng Liu, Ji-Rong Wen: INTERS: Unlocking the Power of Large Language Models in Search with Instruction Tuning. **ACL 2024**

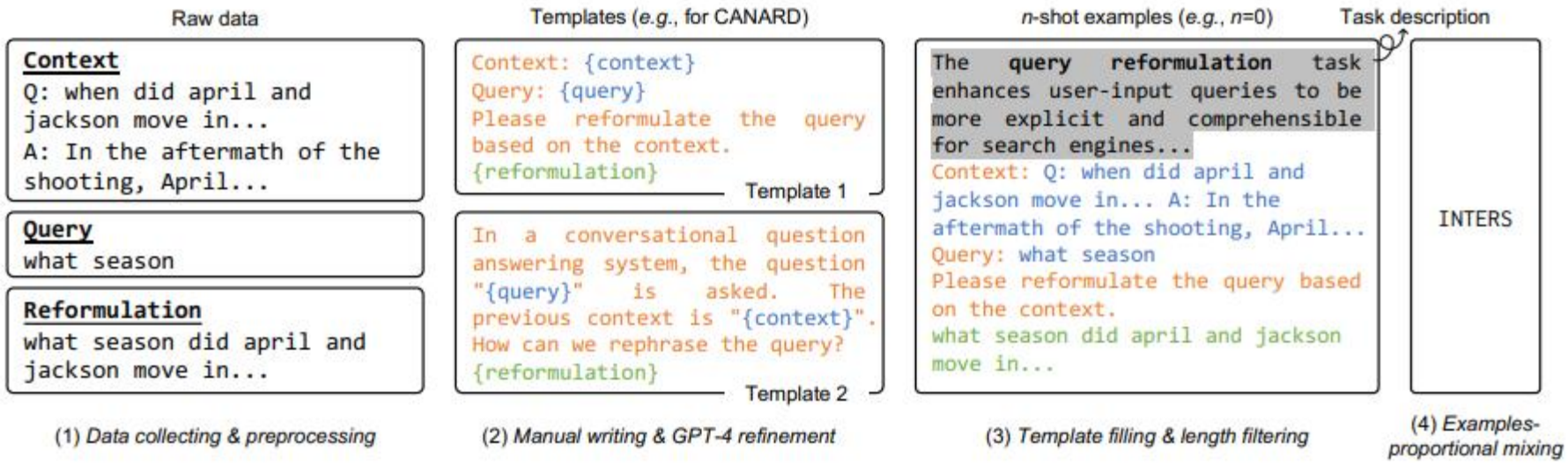
面向IR任务微调大模型：基本思路

- IR任务分类
  - 3个任务类
    - 查询理解
    - 文档理解
    - 查询-文档关系理解
  - 20个任务
  - 43个数据集
- 收集并构建指令微调数据集
- 进行大量实验与分析



Yutao Zhu, Peitian Zhang, Chenghao Zhang, Yifei Chen, Binyu Xie, Zhicheng Dou, Zheng Liu, Ji-Rong Wen:
INTERS: Unlocking the Power of Large Language Models in Search with Instruction Tuning. **ACL 2024**

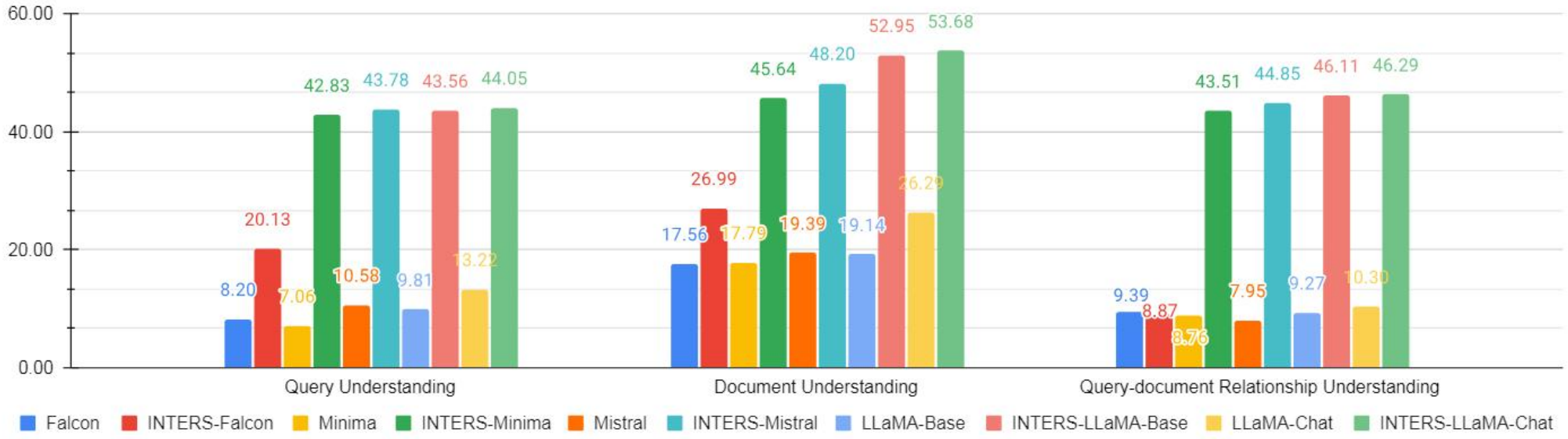
面向IR任务微调大模型：指令集合构建



- 模板收集
  - 为每个任务写一个任务说明 ⇒ 辅助理解任务并增强模型泛化性
  - 为每个数据集手写12个模板并考虑反转模板 ⇒ 增加指令多样性
- 样例生成
  - 用任务描述+模板构建数据样例
  - 构造{0, 1, 2, 3, 4, 5}-shot的数据 →Demonstrations
  - 其中few-shot样本的模板有50%概率相同，50%概率随机采样 ⇒ 增加指令多样性
- 样例混合
  - 采用数据比例混合策略 ⇒ 防止数据量大的数据集主导训练

面向IR任务微调大模型：实验结果

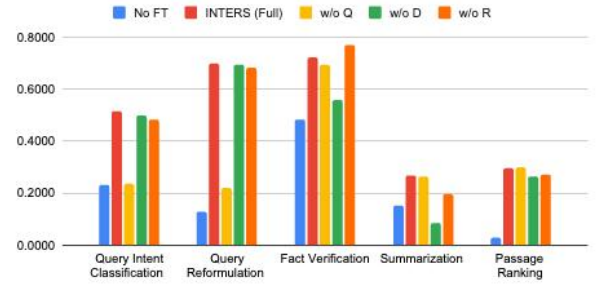
1) INTERS可以提高所有模型在各类检索任务上的性能 ⇒ INTERS带来的能力提升具备普遍性
2) 尺寸更大的模型能够获得更加显著的提升⇒ 大规模参数带来的收益是显著的
3) 微调后，部分任务上3B模型可能超越7B模型的性能 ⇒ 如果只做特定任务，使用小模型微调可能是一种经济的方案



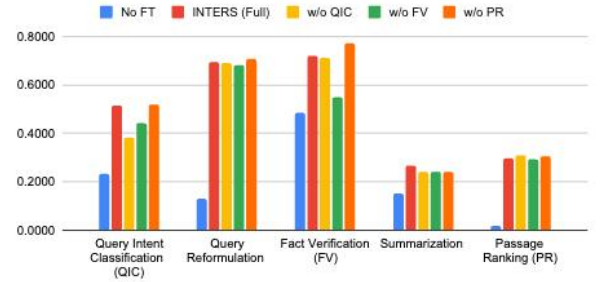
使用四种不同尺寸的开源模型在INTERS上进行微调：Falcon-RW-1B，Minima-2-3B， Mistral-7B，LLaMA-2-7B （Base & Chat）

面向IR任务微调大模型：泛化能力

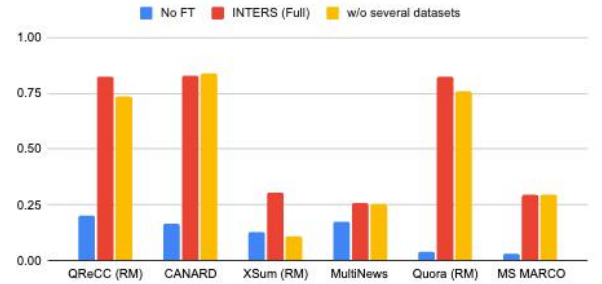
- **类别级别的泛化性**
  1) 移除一个类别的任务后，模型在改类别上性能下降（相比全数据集微调），但比不训练仍然有提升 ⇒ 不同任务类别起到了辅助学习的作用
- **任务级别的泛化性**
  - 移除一个任务后，该任务性能下降（相比全数据集微调），但也好于不训练；
  - 与被移除任务相关的任务也略有下降 ⇒ 模型具备跨任务泛化性，且不同任务之间会产生互相影响
- 数据集级别的泛化性
  - 移除若干数据集后，模型在该数据集上性能下降（相比全数据集微调），但也好于不训练；
  - 但也有数据集相比不训练下降了；
  - 与被移除数据集同任务的数据集变化无规律 ⇒ 模型具备跨数据集泛化性，但不同数据集难度不同，泛化性有区别，且不同数据集之间的影响需要进一步探究



(a) Performance of removing different task groups. We use "Q", "D", and "R" to denote query understanding, document understanding, and query-document relationship understanding, respectively.



(b) Performance of removing different tasks.



(c) Performance of removing different tasks. "RM" indicates the dataset is removed from the training set and becomes unseen during test.

报告内容

1 大模型赋能的信息检索

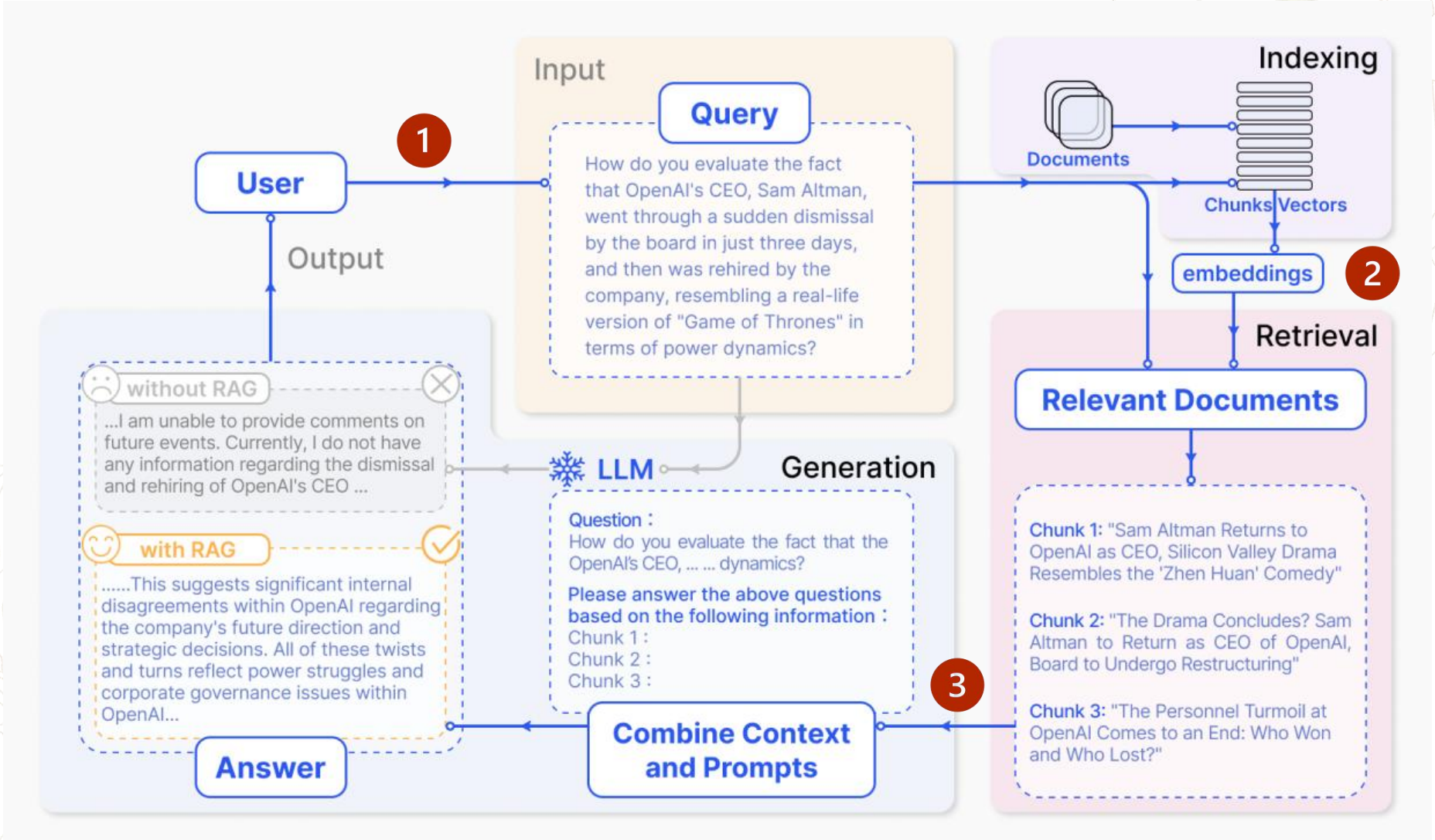
2 检索增强的大模型

3 生成式文档检索

大模型的检索增强（RAG）

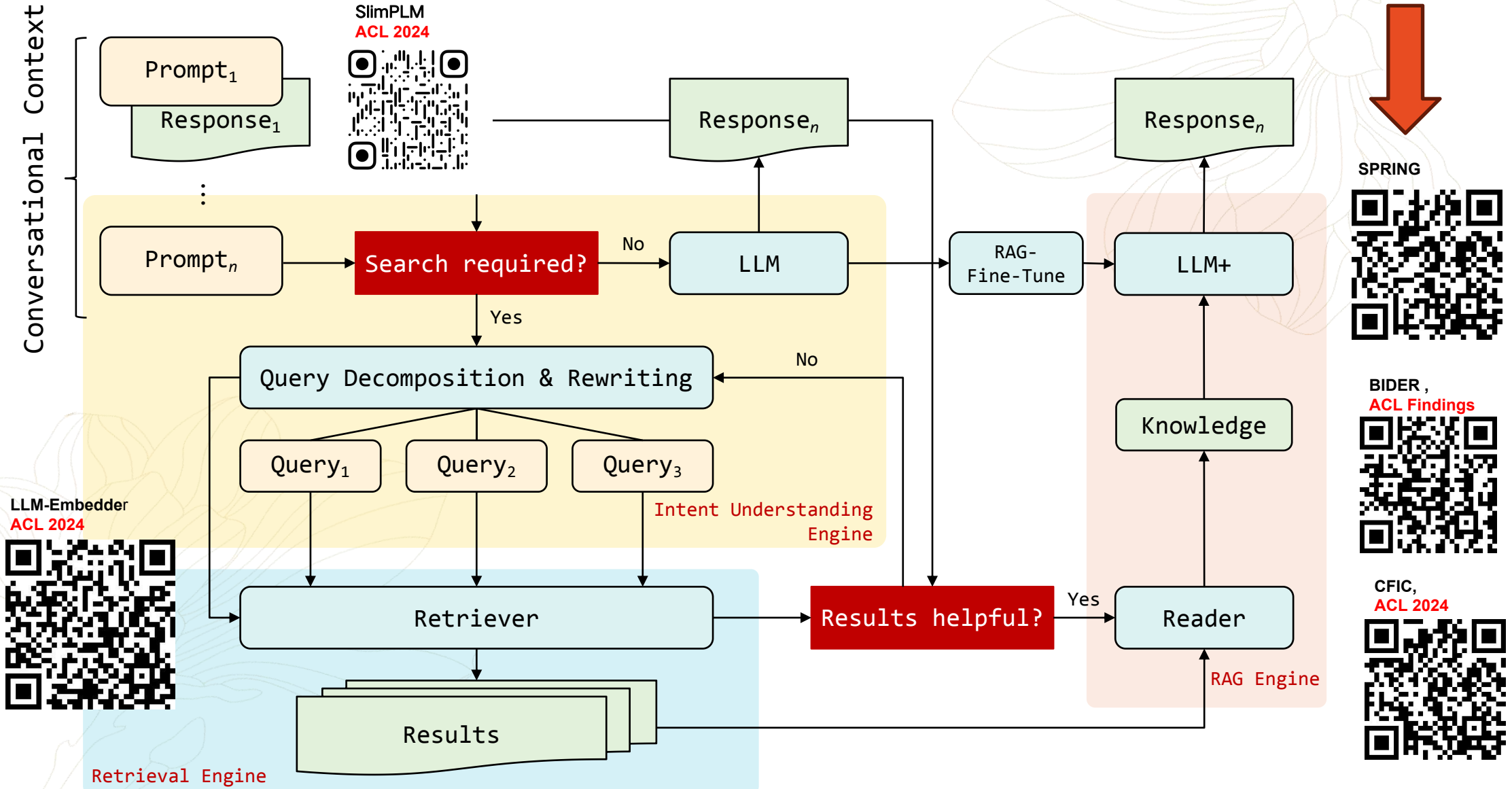
- 大模型并不完美，RAG是提升大模型质量的有效途径

RAG典型流程



Retrieval-Augmented Generation for Large Language Models: A Survey, Gao et al., 2024

RAG – 关键研究问题


Conversational Context
Prompt$_1$
Response$_1$
Prompt$_n$
SlimPLM
ACL 2024
Response$_n$
Search required?
No
LLM
Yes
Query Decomposition & Rewriting
No
Query$_1$
Query$_2$
Query$_3$
Intent Understanding Engine
LLM-Embedder
ACL 2024
Retriever
Results
Retrieval Engine
Results helpful?
Yes
RAG-Fine-Tune
Response$_n$
LLM+
Knowledge
Reader
RAG Engine
SPRING
BIDER ,
ACL Findings
CFIC,
ACL 2024

高瓴人工智能学院
Gaoling School of Artificial Intelligence

One Token Can Help! Learning Scalable and Pluggable Virtual Tokens for Retrieval-Augmented Large Language Models

用一个token激活LLM的RAG能力

Gaoling School of Artificial Intelligence, Renmin University of China
Yutao Zhu, Zhaoheng Huang, Zhicheng Dou, and Ji-Rong Wen*

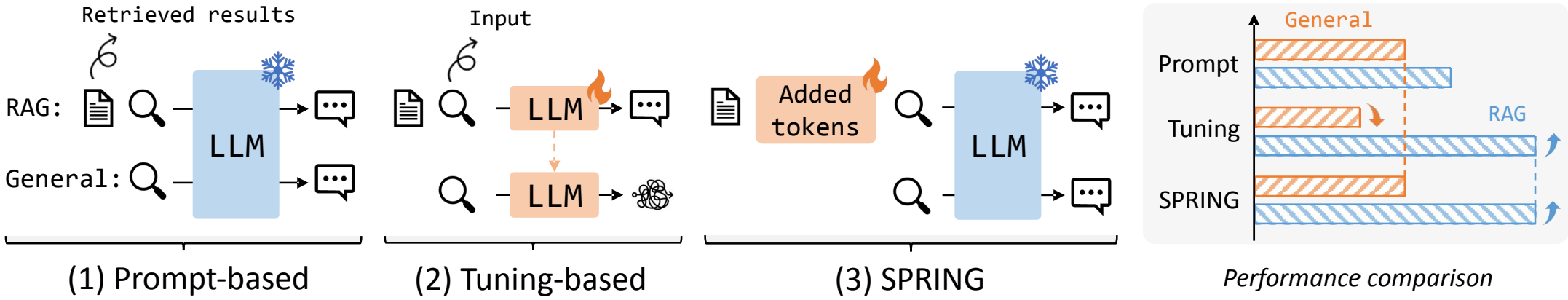
Paper: https://arxiv.org/abs/2405.19670
Github: https://github.com/DaoD/SPRING

背景 & 动机

- RAG是解决大模型事实性、准确性、时效性问题的有效手段之一
- 已有的RAG for LLM的方法有两种
  - 基于提示工程（prompt-engineering）的方法
    - ↪ 可以适用于任何LLM
    - ↪ 效果有限且依赖于提示的质量以及LLM理解提示的能力
  - 基于训练的方法（预训练 or 微调）
    - ↪ 通常性能更好，但需要大量计算资源（可使用PEFT方法解决）
    - ↪ 适应了RAG任务后可能影响原始LLM的能力
- 能否实现增强LLM的RAG能力但不损伤其原始能力？

  对于已经部署的LLM尤为重要!

基本思路



- 在检索结果与用户输入之间插入可学习的虚拟token
  - 高效的轻量化方法
  - 可延展性（灵活性）
  - 即插即用
  - 良好的泛化性

- SPRING: Scalable and Pluggable ViRtual Tokens for RetrIeval-augmeNted Generation （“弹簧”）

问题定义

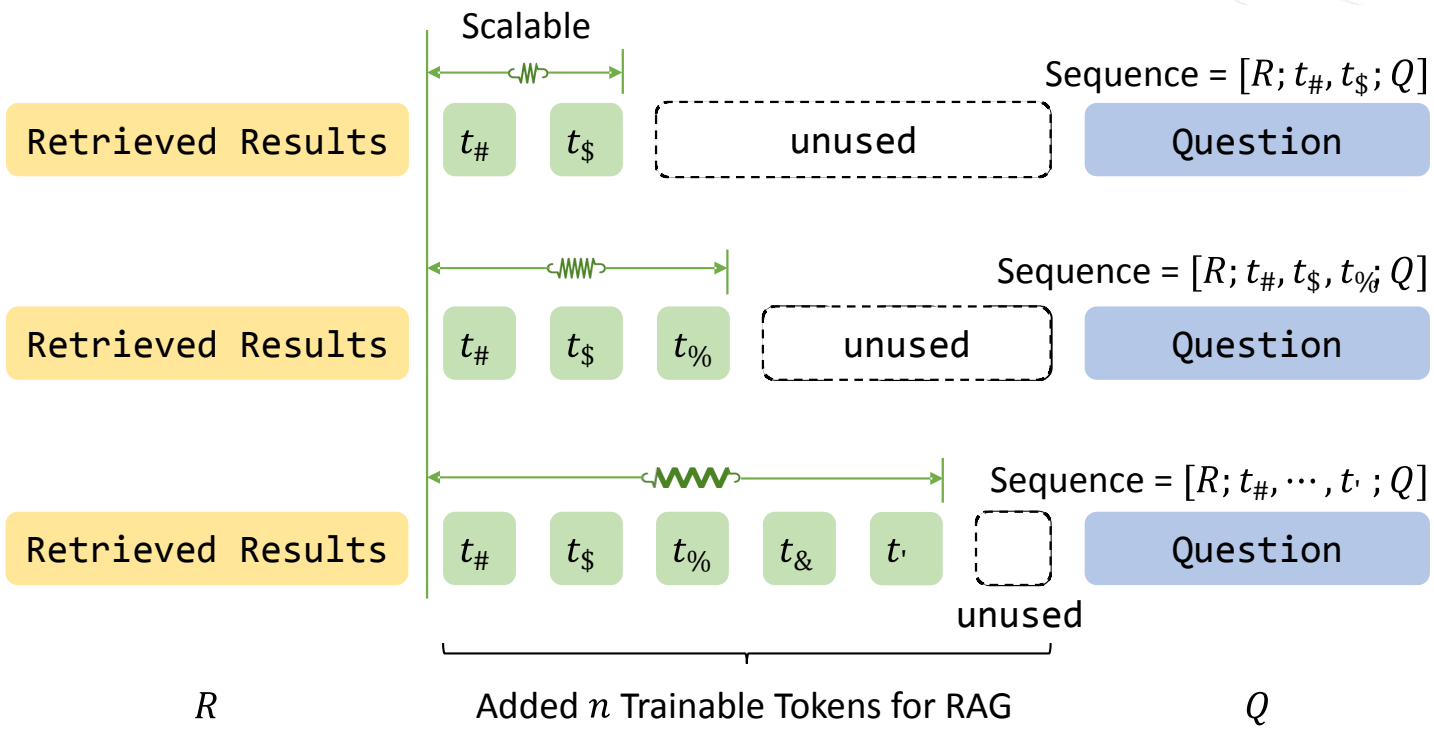
- 以QA任务为例
- 检索结果R，问题Q
- 目标为生成答案A

$$p_{\text{RAG-QA}} = \prod_{i=1}^{n} p_\theta(a_i|R;Q)$$

- 检索结果可能有多个，采用拼接方法

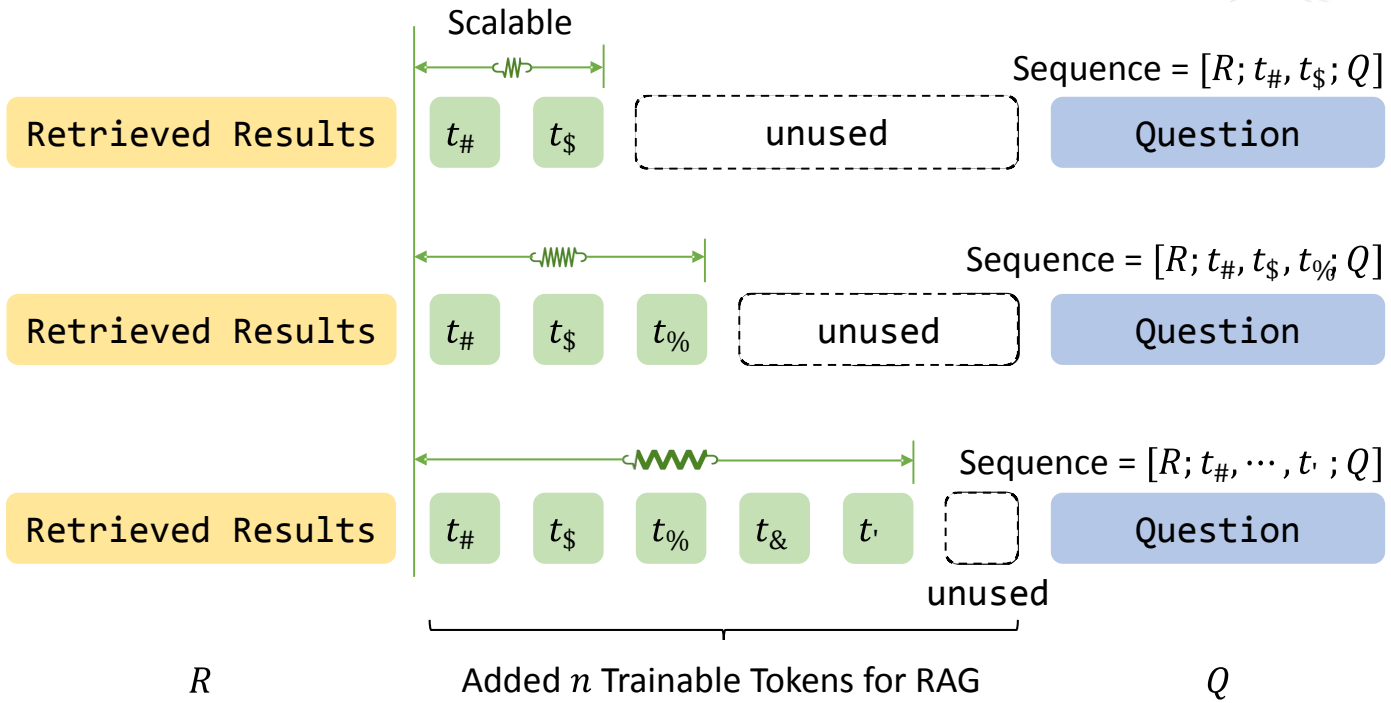
注：此处公式不体现更复杂的提示设计

方法设计



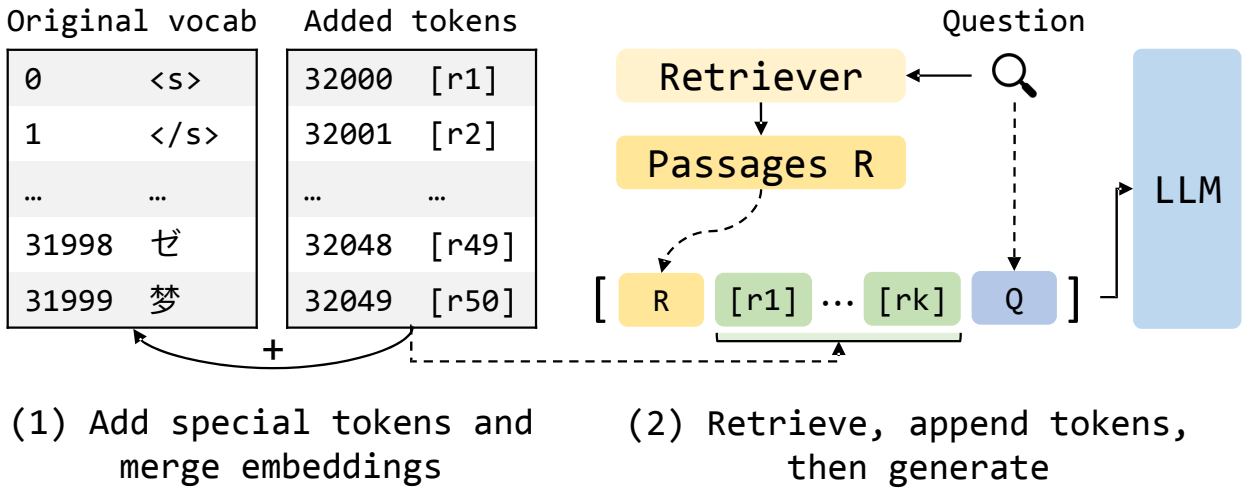
- $p_{\text{SPRING}} = \prod_{i=1}^{n} p_{\theta,\delta}(a_i | R; [t_1, \cdots, t_n]; Q)$
- 插入n个可学习虚拟tokens，引入额外参数$\delta \in \mathbb{R}^{n \times d}$，$d$为LLM的embedding大小（7b为4,096）
- 以50个tokens为例，额外参数为$50 \times 4{,}096 = 0.2\text{M}$
- 插入位置的选择：(1) 可访问检索结果；(2) 让Q接近生成目标

可延展性 & 即插即用



- 随机使用前$k(k \leq n)$个tokens进行训练，其他tokens不用
- 让模型学会在任意k个tokens的帮助下完成RAG任务
- 非RAG场景可以把虚拟tokens移除，还原为原模型

推理



(1)

```python
def add_special_tokens(model, tokenizer, new_embeddings):
    # get original LLMs' embeddings
    embedding_layer = model.embed_tokens
    embedding_weights = embedding_layer.weight
    original_vocab_size, embedding_dim = embedding_weights.shape

    # initialize special tokens and add them into the vocabulary
    added_tokens = [f"[ref{i}]" for i in range(1, 51)]
    tokenizer.add_tokens(added_tokens)

    # add trained embeddings into the original embeddings
    new_vocab_size = len(tokenizer)
    new_embedding_weights = torch.zeros(new_vocab_size, embedding_dim)
    new_embedding_weights[:original_vocab_size, :] = embedding_weights
    new_embedding_weights[original_vocab_size:, :] = new_embeddings
    embedding_layer.weight.data = new_embedding_weights
    return model, tokenizer
```

(2)

```python
new_embeddings = ...
model, tokenizer = add_special_tokens(model, tokenizer, new_embeddings)
added_tokens = [f"[ref{i}]" for i in range(1, 51)]
added_tokens = "".join(added_tokens)
retrieved_results = ...
question = ...
input_text = retrieved_results + added_tokens + question
# using the input text for generation...
```

实验结果

- SPRING显著提升RAG效果
- LoRA在RAG效果上更好，但训练后无法适应non-RAG的场景（过拟合）
- 手写Prompt确实是有效的方法
- SPRING可以提升所有模型的性能（类型、尺寸）
- Prefix-tuning效果很差，SPRING的token插入位置更加合理

		with Retrieval					without Retrieval				
Dataset	Metric	Concat	Prompt	Prefix	LoRA	SPRING	Concat	Prompt	Prefix	LoRA	SPRING
Tuning Parameters		0	0	0.2M	4M	0.2M	0	0	0.2M	4M	0.2M
TQA	EM	0.00	57.79	11.74	62.76	**65.71**	0.01	39.90	0.00	0.03	**46.56**
	F1	65.60	80.33	59.97	**85.44**	85.26	63.96	69.72	28.30	34.91	**74.48**
NQ	EM	0.00	28.99	13.04	**47.95**	42.35	0.00	13.36	0.00	0.00	**18.80**
	F1	41.77	58.72	38.22	**74.15**	70.73	43.74	48.63	17.74	29.10	**55.75**
HQA	EM	0.00	26.36	5.79	**39.95**	35.26	0.00	17.07	0.00	0.03	**20.15**
	F1	44.91	56.15	42.59	**68.93**	65.44	47.54	49.50	17.45	27.57	**54.79**
SQuAD	EM	0.00	23.92	7.19	**35.71**	33.67	0.00	8.61	0.00	0.00	**12.71**
	F1	43.05	57.66	39.75	**68.05**	66.99	43.32	46.81	21.88	27.51	**53.58**
WebQ	EM	0.00	17.53	4.44	**43.65**	31.84	0.00	14.79	0.00	0.00	**24.95**
	F1	37.46	52.10	31.55	**71.99**	64.78	44.34	50.60	20.14	32.36	**59.83**
2Wiki	EM	0.00	22.64	4.38	**35.93**	31.80	0.00	23.45	0.00	0.01	**24.62**
	F1	47.77	55.58	41.82	**63.85**	62.03	52.83	53.55	21.14	37.09	**56.83**
CoQA	EM	0.00	8.20	1.56	12.89	**13.28**	0.00	8.59	0.00	0.00	**9.96**
	F1	27.98	36.72	20.02	41.19	**42.41**	32.97	36.58	13.15	18.99	**39.96**
MS MARCO	EM	0.00	5.73	0.60	**8.13**	6.57	0.00	**2.56**	0.00	0.01	2.09
	F1	56.44	53.81	50.56	**54.81**	53.48	49.50	47.75	47.44	52.44	**51.41**
PopQA	EM	0.00	39.79	10.02	47.15	**48.71**	0.00	16.05	0.00	0.00	**20.25**
	F1	56.49	68.26	44.54	73.12	**73.90**	53.61	54.85	20.39	25.09	**58.32**
Average	EM	0.00	25.66	6.53	**37.13**	34.35	0.00	16.04	14.31	0.01	**20.01**
	F1	46.83	57.70	41.00	**66.84**	65.00	47.98	50.89	23.07	31.67	**56.11**

LoRA的过拟合问题

- 测试LLM推理、数学、世界知识
- SPRING可以无损转化为原始模型，实现模型能力的有效保留
- LoRA出现过拟合问题，无法适应其他任务

Dataset	n-shot	LoRA	SPRING	Diff
BoolQ	0	79.30	**82.97**	3.67
CommonsenseQA	0	55.45	**63.80**	8.35
CommonsenseQA	4	59.87	**67.07**	7.20
GSM8K	8	17.33	**31.89**	14.56
MMLU	0	51.30	**53.62**	2.32
MMLU	5	48.76	**54.96**	6.20

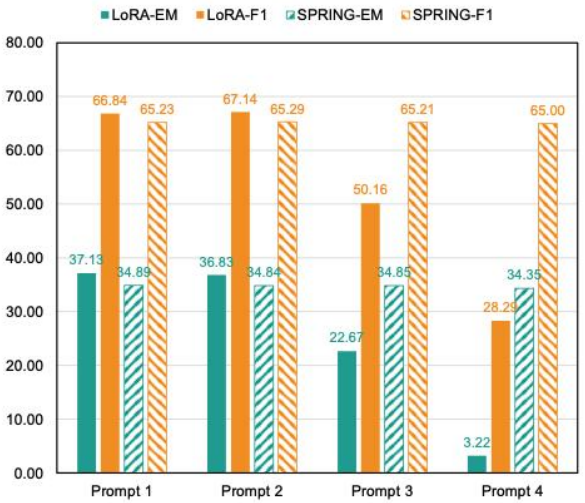
- 测试方法对提示的鲁棒性
- LoRA只能在与训练提示高度相近的提示上保持良好能力，无法泛化到其他提示

Prompt 1:
According to the previous relevant passages, please answer the following question. Only return the answer without any other words.\n

Prompt 2:
According to the previous relevant passages, please answer the following question.\n

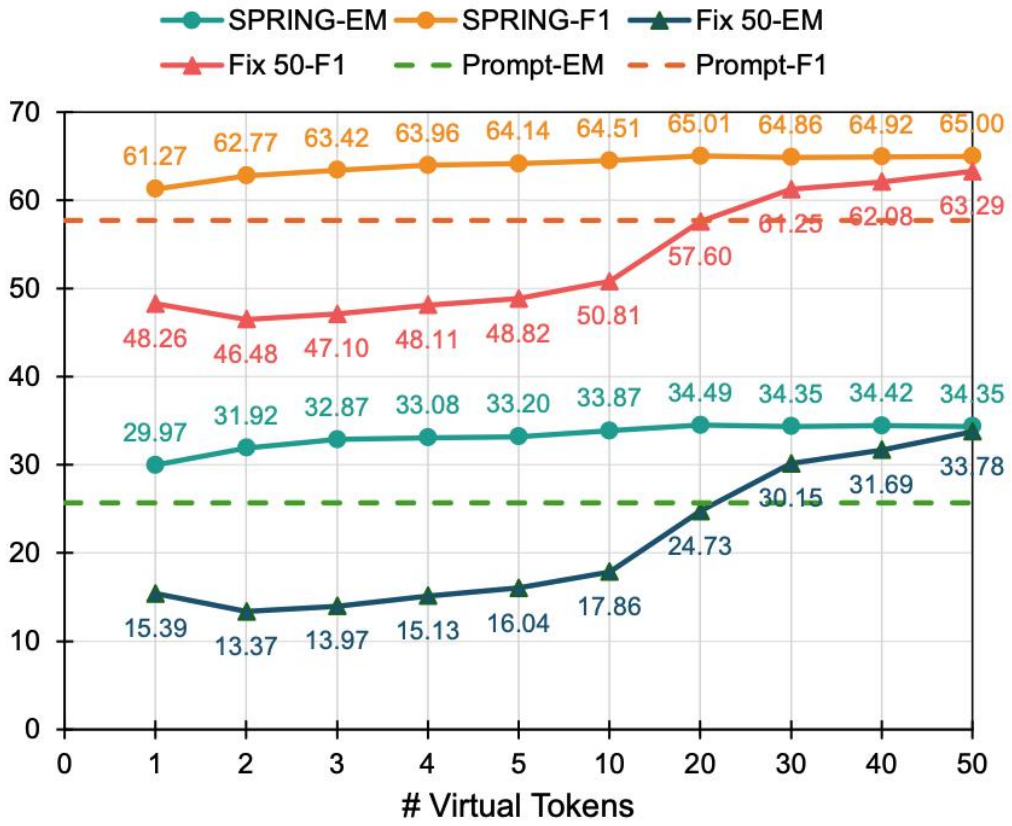
Prompt 3:
Answer the following question based on the provided passages.\n

Prompt 4:
(None)



虚拟token数量的影响

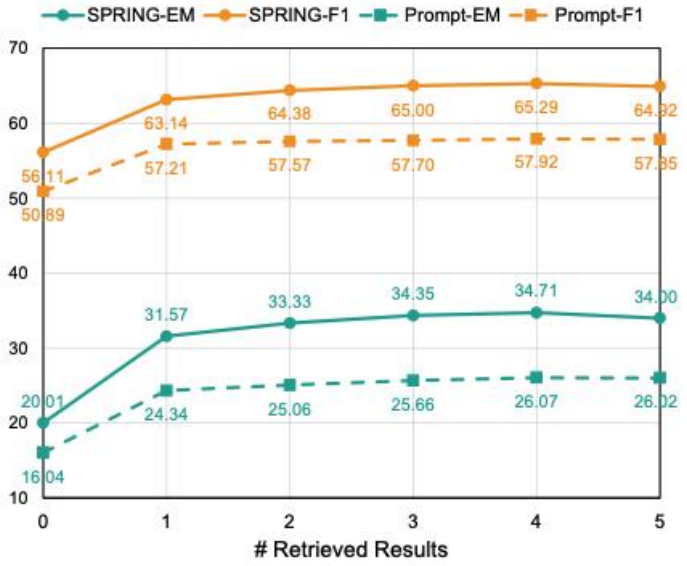
- 最多训练50个，使用可延展训练方法
- 性能可以随着虚拟token数量增加获得提升
- 最少1个token即可提升LLM的RAG能力！
- 使用固定50个token的训练方法仅能在50个token时取得较好效果



检索质量与数量的影响

- SPRING获得一致的提升，具备面向检索器的泛化性
- 相比于Prompt方法， SPRING对检索器的质量更鲁棒
- 尽管使用e5-large训练，SPRING仍然可以在BM25检索结果上有较好的效果，体现其较好的适应性
- 面对检索器的迭代升级，无需重新训练SPRING！
- 检索结果带来提升，但更多的（多于4个）检索结果可能引入噪声

	Prompt		SPRING	
Retriever	EM	F1	EM	F1
BM25	21.23	54.94	30.94	62.73
BGE-base	23.07	56.12	31.81	63.46
E5-base	24.38	56.84	33.34	64.49
E5-large	25.66	57.70	**34.35**	**65.00**
Average	23.58	56.40	32.61	63.92
Variance	2.69	1.02	1.75	0.78



虚拟token插入位置的影响

- 检索结果R，问题Q，和虚拟词T的三种位置关系
  - $[T, R, Q]$ – Prefix-tuning
  - $[R, T, Q]$ – SPRING
  - $[R, Q, T]$ – 末尾插入
- Prefix-tuning效果极差，与原论文不符，因为RAG任务与多任务学习目的不同
- 在末尾插入也有良好效果，但略逊于SPRING，可能是SPRING保持了问题与生成答案之间的连贯性

	$[T, R, Q]$		SPRING		$[R, Q, T]$	
	EM	F1	EM	F1	EM	F1
TQA	11.74	59.97	**65.71**	**85.26**	64.90	84.65
NQ	13.04	38.22	42.35	70.73	**43.16**	**71.30**
HQA	5.79	42.59	**35.26**	**65.44**	34.00	64.27
SQuAD	7.19	39.75	33.67	**66.99**	**34.09**	66.75
WebQ	4.44	31.55	**31.84**	64.78	31.35	**64.84**
2Wiki	4.38	41.82	**31.80**	**62.03**	30.64	61.01
CoQA	1.56	20.02	**13.28**	**42.41**	12.30	41.18
MS MARCO	0.60	50.56	6.57	**53.48**	**6.94**	49.40
PopQA	10.02	44.54	**48.71**	**73.90**	46.55	72.33
Average	6.53	41.00	**34.35**	**65.00**	33.77	63.97

FlashRAG: A Python Toolkit for Efficient RAG Research

arxiv.org/abs/2405.13576

benchmark datasets large-language-models retrieval-augmented-generation

Readme
MIT license
Cite this repository
Activity
Custom properties
798 stars
7 watching
59 forks

FlashRAG: A Modular Toolkit for Efficient Retrieval-Augmented Generation Research

Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, Zhicheng Dou*
Gaoling School of Artificial Intelligence
Renmin University of China
{jinjiajie, dou}@ruc.edu.cn, yutaozhu94@gmail.com

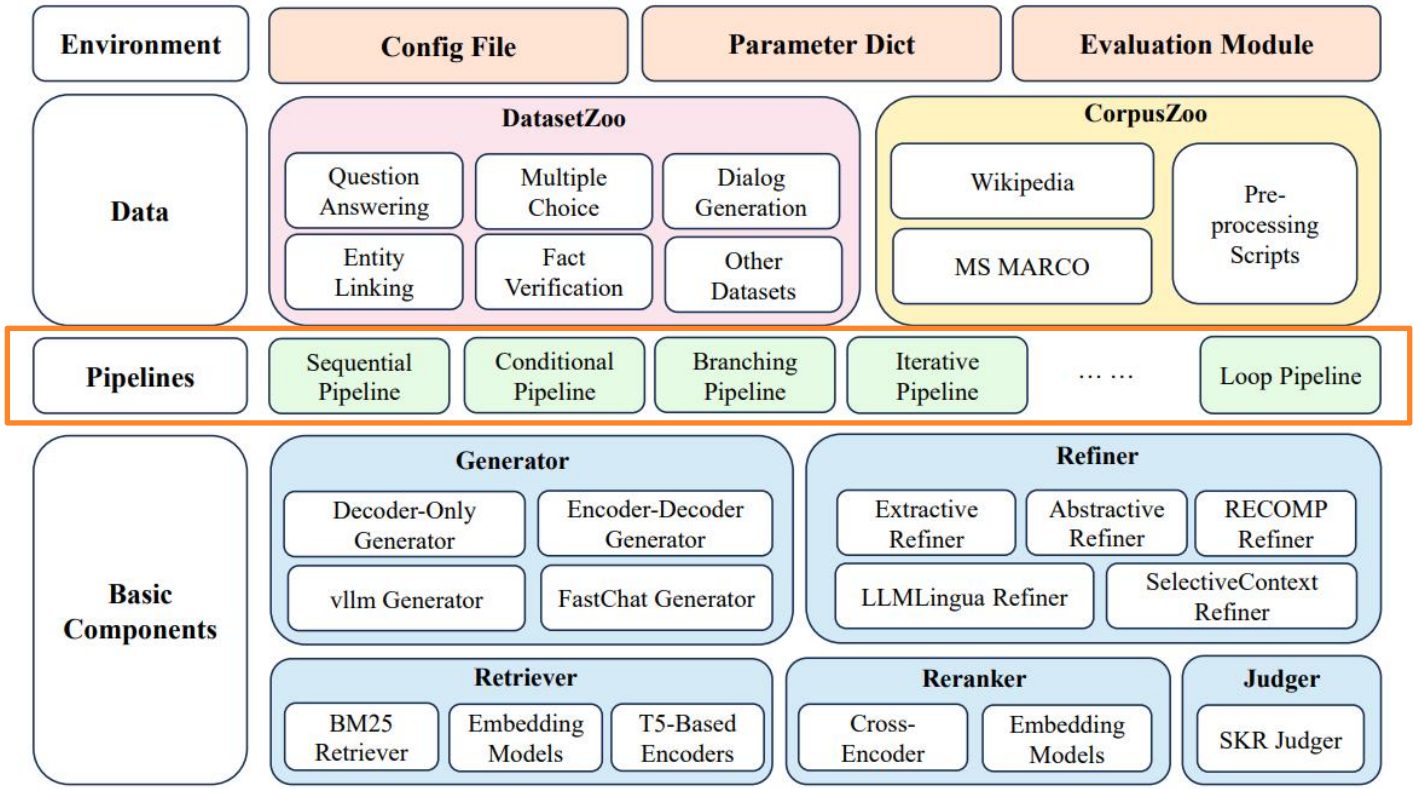
FlashRAG：快速实现RAG方法的工具包

FlashRAG：快速实现RAG方法的工具包

- **动机**
  - RAG系统组件众多，研究人员往往需要花费很多时间在各类工程实现上
  - 现有RAG工作缺少统一的实现框架，导致复现非常耗时并且难以公平比较
  - 已有的LangChain, LlamaIndex等工具包封装复杂，难以满足定制化研究需求
- **特点**
  - 模块化RAG框架，包含检索器、生成器、精炼器等多种组件，支持自定义RAG流程
  - 目前实现12种RAG工作，能够轻松在不同设置下评估结果
  - 包含32个RAG工作中常用数据集，并预处理为统一格式
  - 包含多种辅助脚本，包含Wikipedia预处理与分块、索引构建、检索结果预准备等

框架设计与数据集

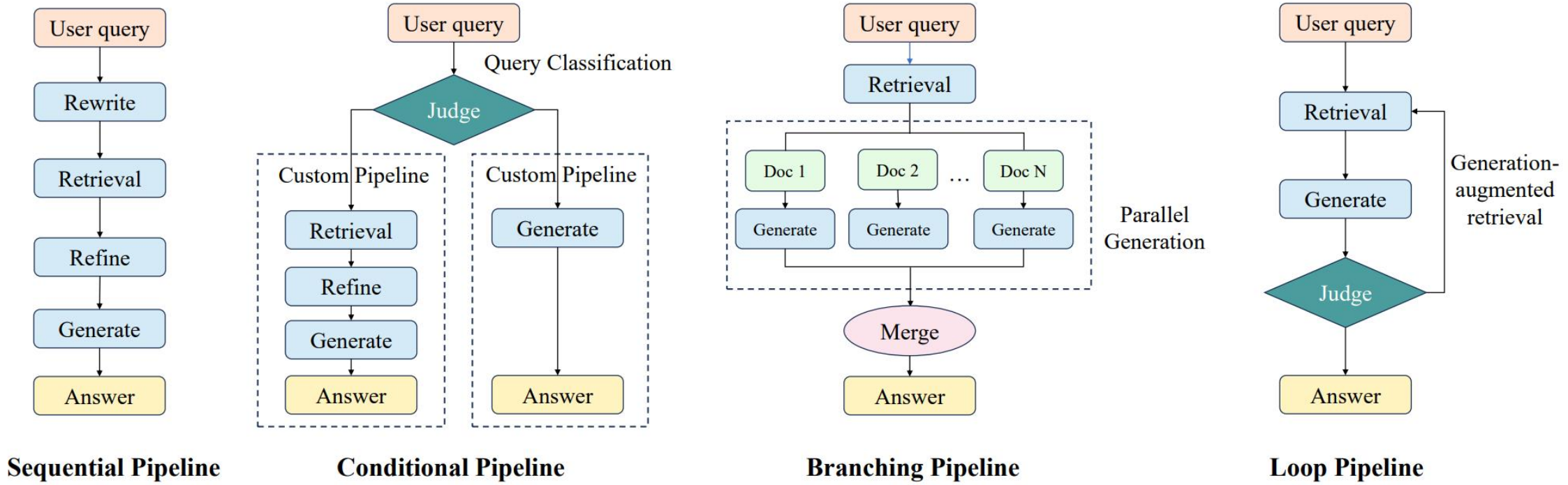
- 基于基础组件构建Pipeline实现常用RAG流程
- 涵盖32种常用数据集
- 大多数基于维基百科作为知识源



Task	Dataset Name	Knowledge Source	# Train	# Dev	# Test
QA	NQ [38]	Wiki	79,168	8,757	3,610
	TriviaQA [39]	Wiki & Web	78,785	8,837	11,313
	PopQA [40]	Wiki	/	/	14,267
	SQuAD [41]	Wiki	87,599	10,570	/
	MSMARCO-QA [42]	Web	808,731	101,093	/
	NarrativeQA [43]	Books, movie scripts	32,747	3,461	10,557
	WikiQA [44]	Wiki	20,360	2,733	6,165
	WebQuestions [45]	Google Freebase	3,778	/	2,032
	AmbigQA [46, 38]	Wiki	10,036	2,002	/
	SIQA [47]	-	33,410	1,954	/
	CommenseQA [48]	-	9,741	1,221	/
	BoolQ [49]	Wiki	9,427	3,270	/
	PIQA [50]	-	16,113	1,838	/
	Fermi [51]	Wiki	8,000	1,000	1,000
Multi-Hop QA	HotpotQA [52]	Wiki	90,447	7,405	/
	2WikiMultiHopQA [53]	Wiki	15,000	12,576	/
	Musique [54]	Wiki	19,938	2,417	/
	Bamboogle [32]	Wiki	/	/	125
Long-Form QA	ASQA [55]	Wiki	4,353	948	/
	ELI5 [56]	Reddit	272,634	1,507	/
Multiple-Choice	MMLU [35, 36]	-	99,842	1,531	14,042
	TruthfulQA [57]	Wiki	/	817	/
	HellaSwag [58]	ActivityNet	39,905	10,042	/
	ARC [59]	-	3,370	869	3,548
	OpenBookQA [37]	-	4,957	500	500
Entity-linking	AIDA CoNLL-YAGO [60, 61]	Wiki & Freebase	18,395	4,784	/
	WNED [62, 61]	Wiki	/	8,995	/
Slot filling	T-REx [63, 61]	DBPedia	2,284,168	5,000	/
	Zero-shot RE [64, 61]	Wiki	147,909	3,724	/
Fact Verification	FEVER [65, 61]	Wiki	104,966	10,444	/
Dialog Generation	WOW [66, 61]	Wiki	63,734	3,054	/
Open-domain Summarization*	WikiAsp [67]	Wiki	300,636	37,046	37,368

不同的RAG流水线

- 目前已支持常见的四种不同流水线的RAG工作



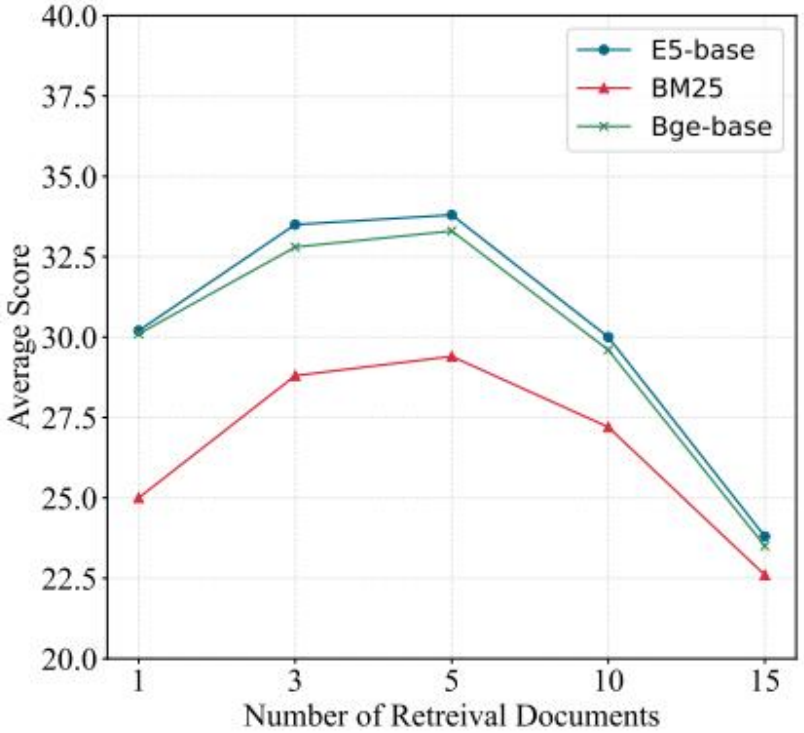
实验结果

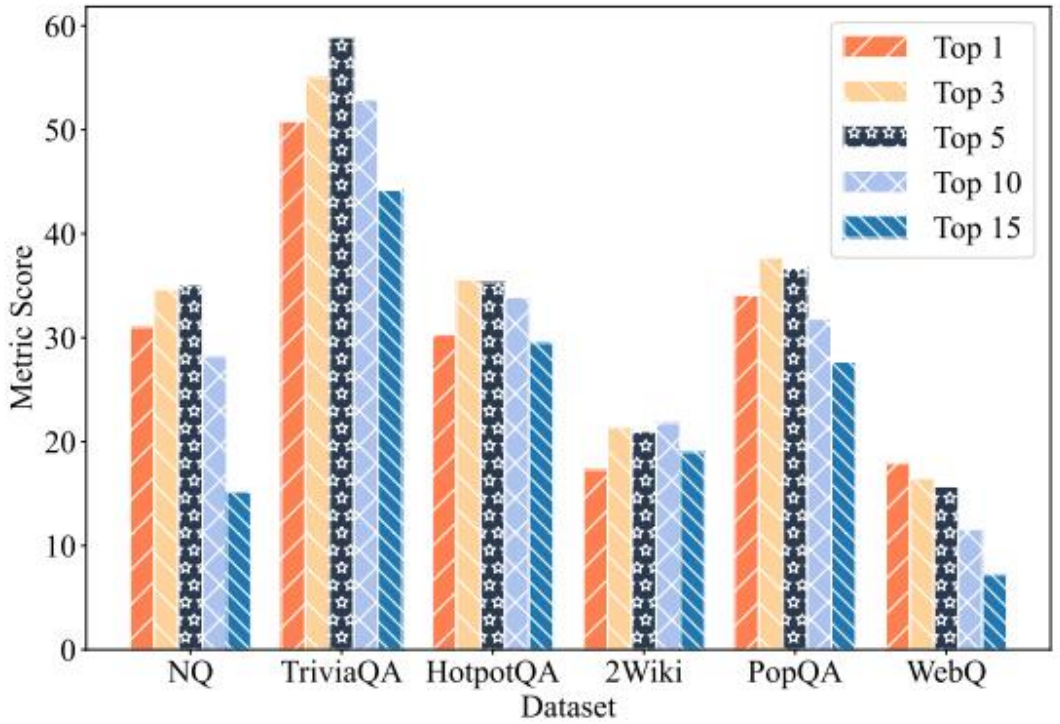
- 在统一的设置下实现评估各类方法 (E5 + Llama3-8B-instruct)

Method	Optimize component	Pipeline type	NQ (EM)	TriviaQA (EM)	HotpotQA (F1)	2Wiki (F1)	PopQA (F1)	WebQA (EM)
Naive Generation	-	Sequential	22.6	55.7	28.4	33.9	21.7	18.8
Standard RAG	-	Sequential	35.1	58.8	35.3	21.0	36.7	15.7
AAR [72]	Retriever	Sequential	30.1	56.8	33.4	19.8	36.1	16.1
LongLLMLingua [20]	Refiner	Sequential	32.2	59.2	37.5	25.0	38.7	17.5
RECOMP-abstractive [18]	Refiner	Sequential	33.1	56.4	37.5	32.4	39.9	20.2
Selective-Context [21]	Refiner	Sequential	30.5	55.6	34.4	18.5	33.5	17.3
Ret-Robust* [73]	Generator	Sequential	42.9	68.2	35.8	43.4	57.2	33.7
SuRe [29]	Flow	Branching	37.1	53.2	33.4	20.6	48.1	24.2
REPLUG [28]	Generator	Branching	28.9	57.7	31.2	21.1	27.8	20.2
SKR [10]	Judger	Conditional	25.5	55.9	29.8	28.5	24.5	18.6
Self-RAG* [33]	Flow	Loop	36.4	38.2	29.6	25.1	32.7	21.9
FLARE [34]	Flow	Loop	22.5	55.8	28.0	33.9	20.7	20.2
Iter-RetGen [30], ITRG [31]	Flow	Loop	36.8	60.1	38.3	21.6	37.9	18.2

检索结果对RAG流程的影响

- 过多或过少的检索结果会造成性能下降
- 检索文档数量越大，BM25和稠密检索方法的差距越小





报告内容

1 大模型赋能的信息检索

2 检索增强的大模型

3 生成式文档检索

生成式检索

传统搜索引擎

缺点：
indexing, retrieval, ranking是分阶段进行的，无法联合优化
检索模型无法与下游的生成模型联合优化

基于（生成式）模型的检索

能够更好的与下游任务（尤其是生成式任务）联合优化

生成式检索(Model-based IR) 有望成为新的检索架构

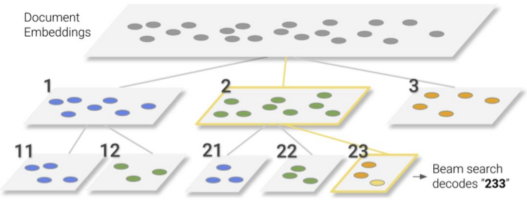
Rethinking search: making domain experts out of dilettantes. SIGIR Forum 55(1): 13:1–13:27 (2021)

大模型与检索

- 一般意义上讲，大模型（LLMs）本身并没有检索能力
- 大模型瞎编烂造的能力在检索相关的问题上体现的淋漓尽致
- 大模型需要定向微调，才能实现“检索能力”

生成式检索的基本框架

- 模型架构：Seq2Seq架构/自回归生成架构
  - Indexing: 生成模型的训练过程，将文档内容有效地嵌入到生成模型中。
  - Retrieval: 调用生成模型自回归的生成docid

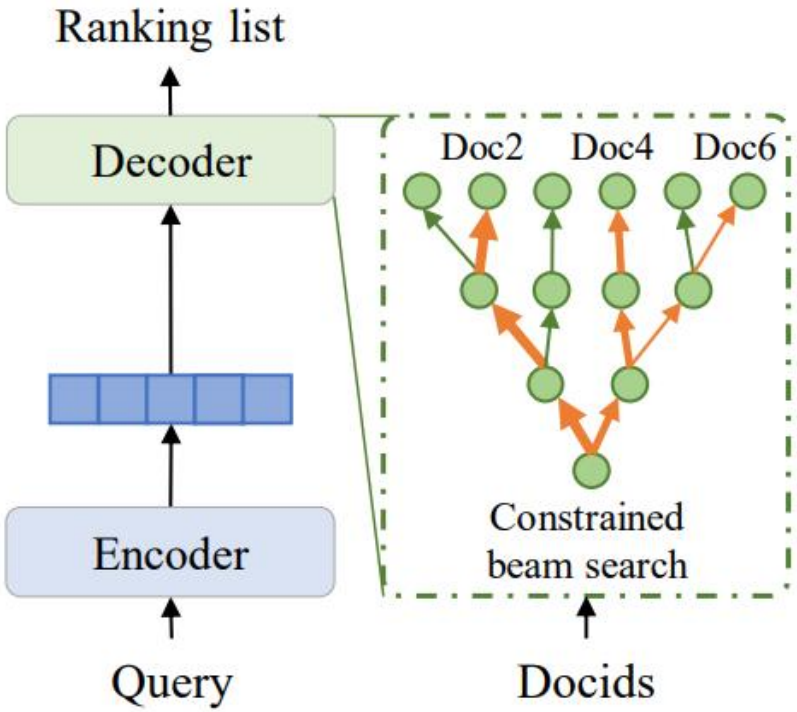


输出： DocId Token序列

doc12334

输入：查询词Token序列

Renmin Univeristy

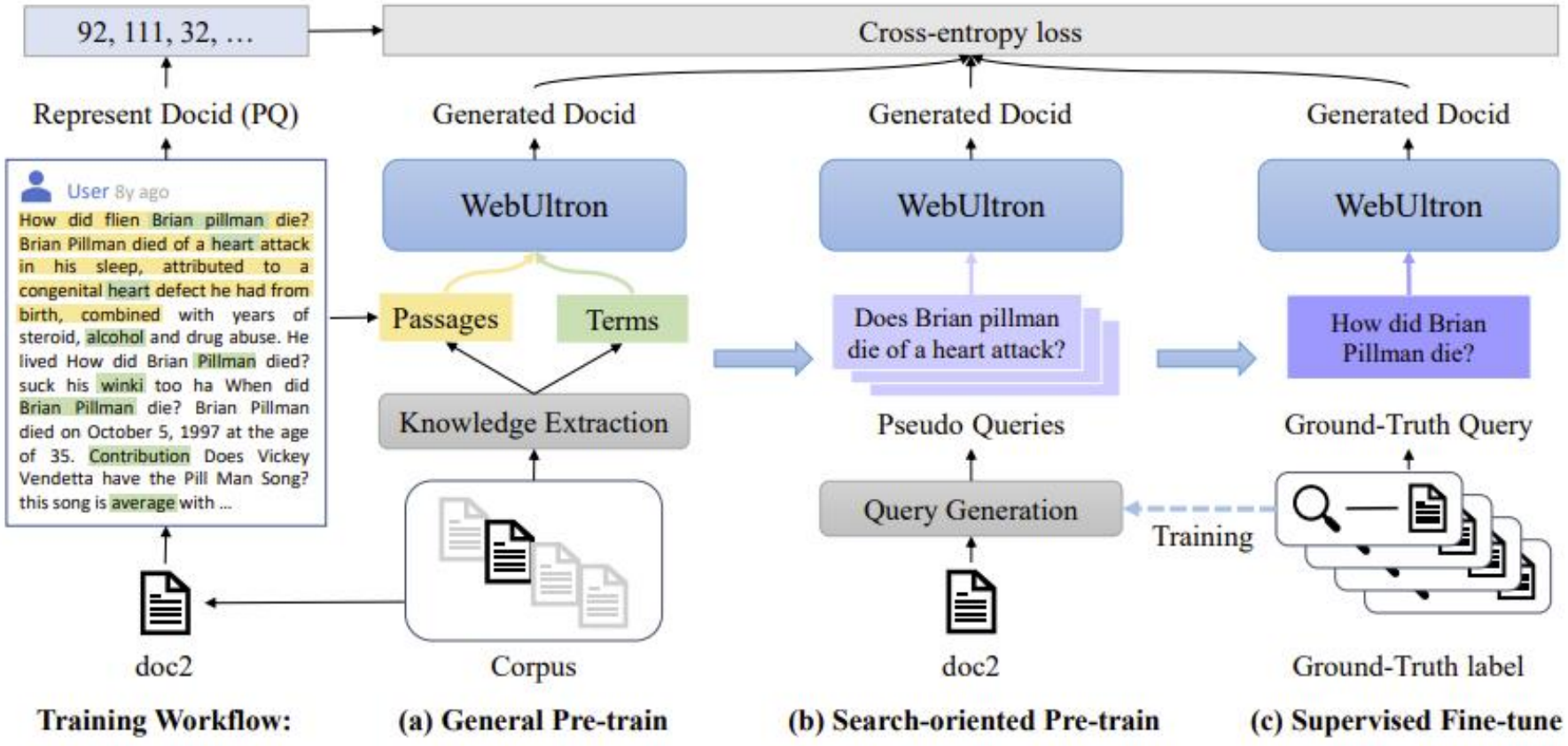


常用的文档标识符种类：

- 数字序号
- 数字串
- 层次化聚类ID
- 重要的关键词/ngram
- URL变种
- 文档标题
- …

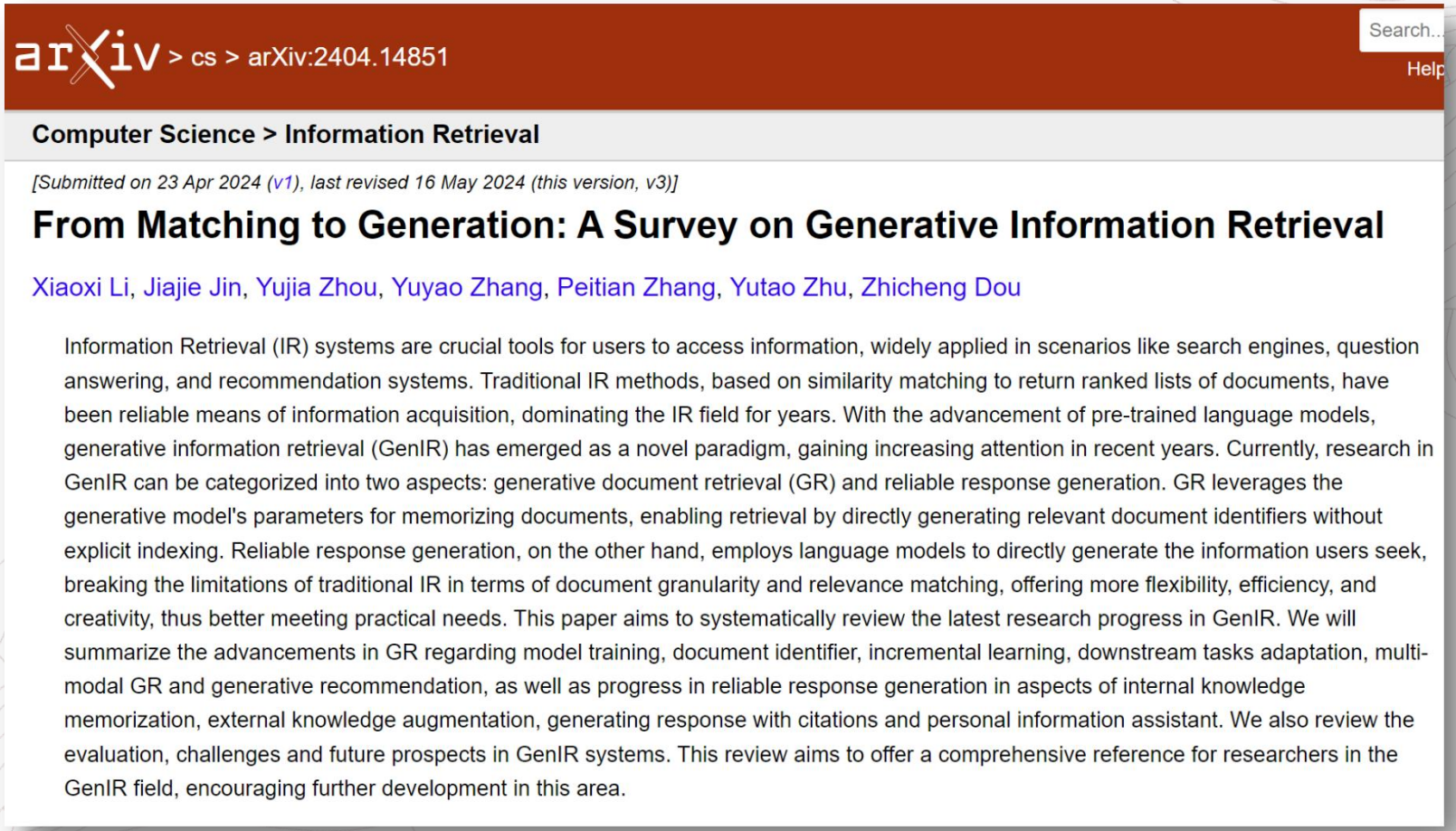
生成式检索模型：WebUltron

- 三阶段训练框架：将文档信息有效地“索引”到模型中



Yujia Zhou, Jing Yao, Ledell Wu, Zhicheng Dou* and Ji-Rong Wen. 2023. WebUltron: An Ultimate Retriever on Webpages Under the Model-Centric Paradigm. IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2023.3332858.

参考资料



arXiv > cs > arXiv:2404.14851

Search...

Help

Computer Science > Information Retrieval

[Submitted on 23 Apr 2024 (v1), last revised 16 May 2024 (this version, v3)]

From Matching to Generation: A Survey on Generative Information Retrieval

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, Zhicheng Dou

Information Retrieval (IR) systems are crucial tools for users to access information, widely applied in scenarios like search engines, question answering, and recommendation systems. Traditional IR methods, based on similarity matching to return ranked lists of documents, have been reliable means of information acquisition, dominating the IR field for years. With the advancement of pre-trained language models, generative information retrieval (GenIR) has emerged as a novel paradigm, gaining increasing attention in recent years. Currently, research in GenIR can be categorized into two aspects: generative document retrieval (GR) and reliable response generation. GR leverages the generative model's parameters for memorizing documents, enabling retrieval by directly generating relevant document identifiers without explicit indexing. Reliable response generation, on the other hand, employs language models to directly generate the information users seek, breaking the limitations of traditional IR in terms of document granularity and relevance matching, offering more flexibility, efficiency, and creativity, thus better meeting practical needs. This paper aims to systematically review the latest research progress in GenIR. We will summarize the advancements in GR regarding model training, document identifier, incremental learning, downstream tasks adaptation, multi-modal GR and generative recommendation, as well as progress in reliable response generation in aspects of internal knowledge memorization, external knowledge augmentation, generating response with citations and personal information assistant. We also review the evaluation, challenges and future prospects in GenIR systems. This review aims to offer a comprehensive reference for researchers in the GenIR field, encouraging further development in this area.

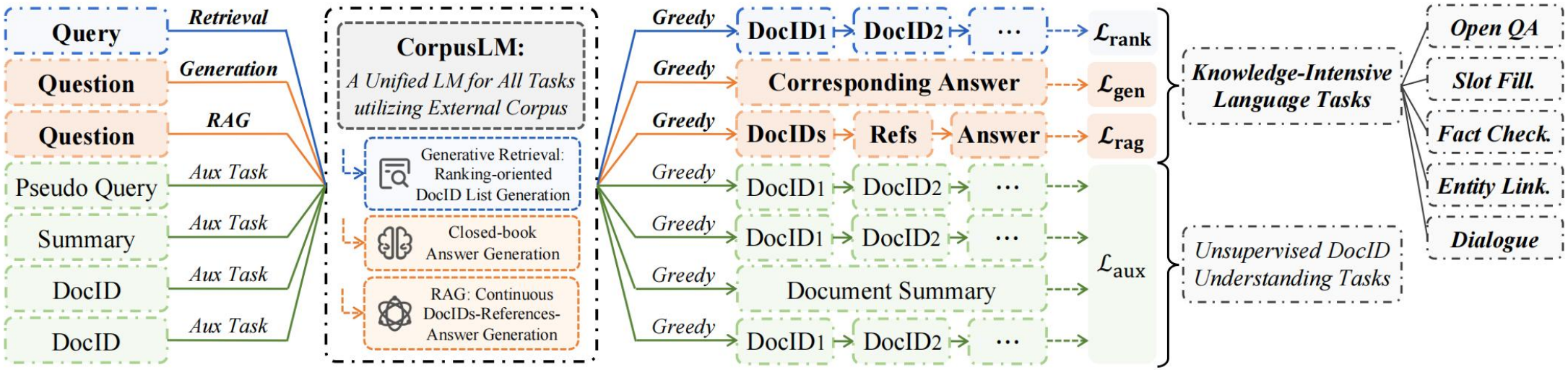
论文链接：
https://arxiv.org/abs/2404.14851

论文链接：https://arxiv.org/abs/2404.14851

GitHub项目链接：https://github.com/RUC-NLPIR/GenIR-Survey

统一检索+生成模型CorpusLM

挑战：生成式文档检索和大模型如何融合？



- **统一的语言模型：**集成生成式检索、闭卷生成和检索增强生成（RAG），辅助知识密集型任务。
- **面向排名的DocID列表生成策略：**贪婪解码生成DocID排序。
- **RAG导向的连续生成策略：**连续解码方法。
- **无监督DocID理解任务：**无监督的DocID理解任务，加深模型对DocIDs背后含义的理解，以进一步提高了CorpusLM的检索和生成性能。

Xiaoxi Li, Zhicheng Dou, Yujia Zhou, and Fangchao Liu. 2024. CorpusLM: Towards a Unified Language Model on Corpus for Knowledge-Intensive Tasks. In SIGIR 2024

大模型+检索→生成式信息检索（GenIR）

- 生成式大模型与检索系统紧密结合（例如，New Bing）
  - 信息检索范式迁移：匹配式信息检索 到 生成式信息检索
  - 检索作为大模型工具，为大模型提供外部知识，提升生成质量

生成式信息检索
Generative Information Retrieval

挑战与开放问题

- 大模型会成为大规模检索召回的新方法么？
  - Web search: billion level documents, quick update
  - SOTA: still underperforms traditional matching-based retrieval models
- RAG 2.0 -> 大模型是否会和检索模型联合优化?
  - Can the web serve as the memory of LLM?
- 未来的RAG是否是一个轻量级检索器 (such as embedding models like BGE, E5) + 一个强大的LLM(with a long input window)的组合?
- RAG是否可以在token层面融合Can be integrate search & LLM in token level (simultaneously retrieval and generation)?
- ...

什么时候会出现新搜索公司?

51CTO WOT

谢 谢 观 看

THANKS

智启新纪
慧创万物
2024