

# 基于RAG构建生成式 AI应用

最佳实践与“避坑指南”

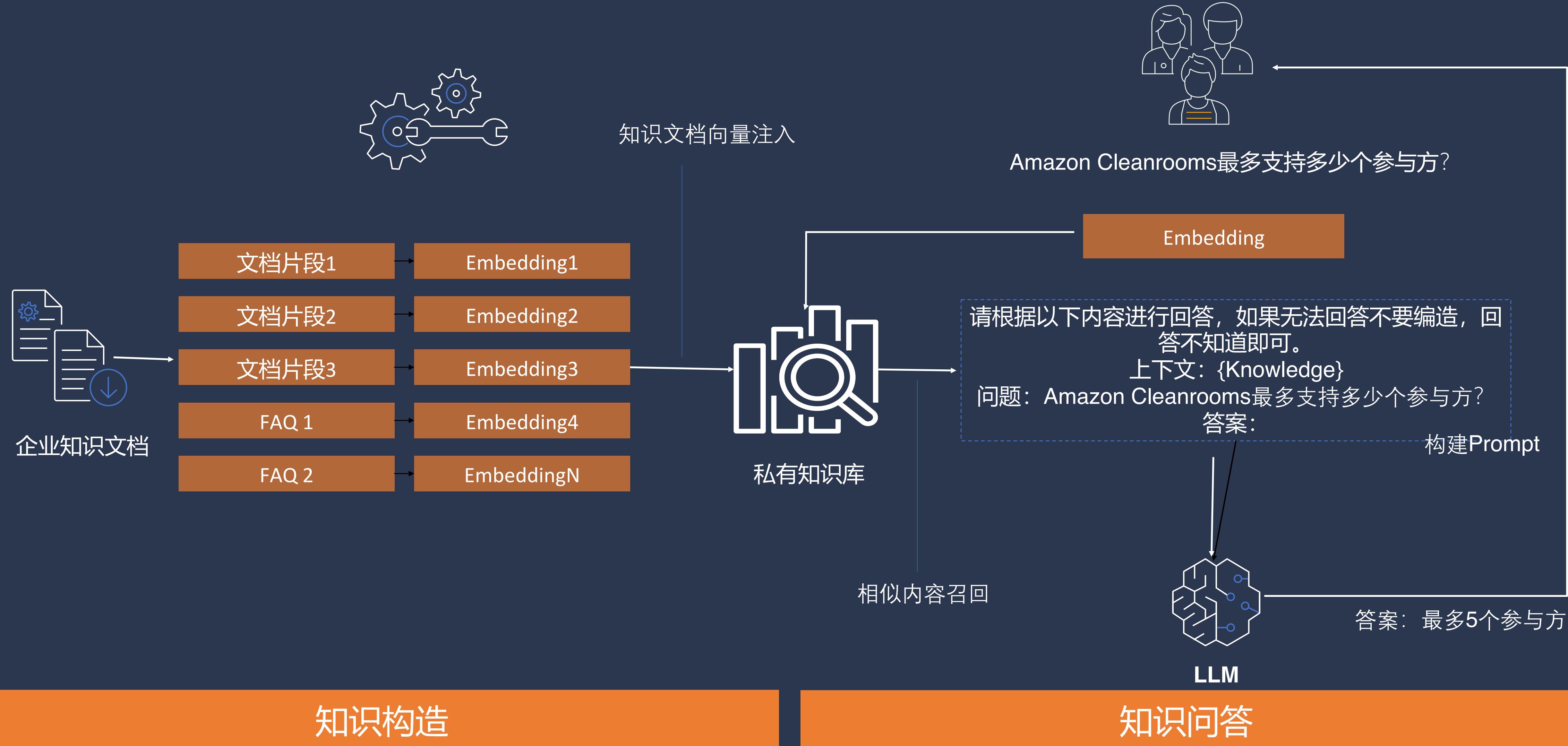
亚马逊云科技人工智能技术专家 / 李元博

# 目录

- RAG场景及技术特点
- RAG实践经验总结
- RAG场景中的亚马逊云产品亮点

# RAG场景及技术特点

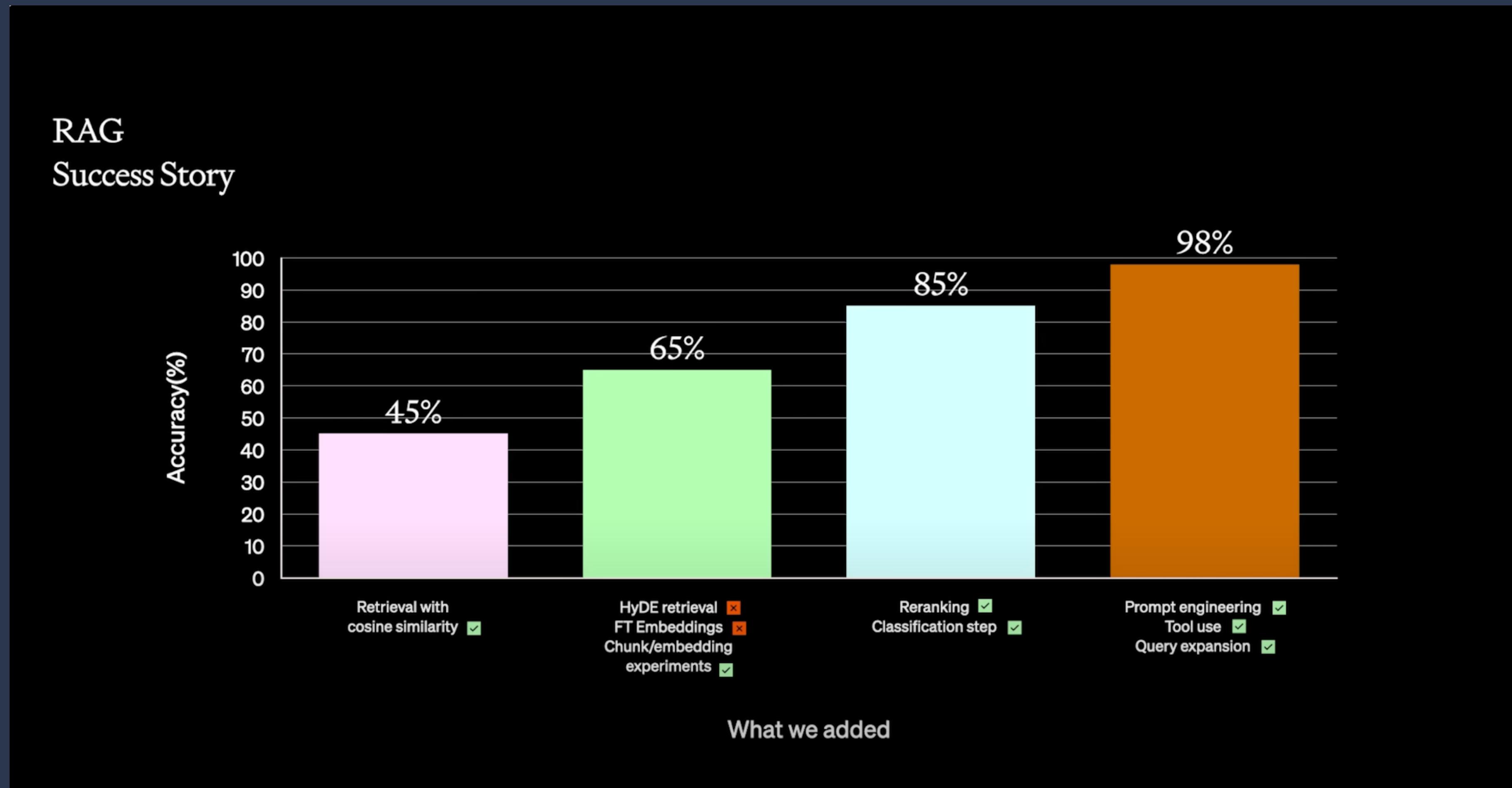
# RAG概念流程图



# RAG适用典型的木桶理论场景



# OpenAI 关于RAG的分享



# RAG场景分类

维度	分类	描述	Tips
按使用场景	Chatbot	<ul style="list-style-type: none"><li>➤一问一答聊天交互</li><li>➤召回topK记录给到LLM归纳总结，返回答案</li></ul>	<ul style="list-style-type: none"><li>➤对于置信度高的召回，可以直接返回top one而不走LLM归纳总结（避免LLM幻觉或自由发挥）。比如某车企的Chatbot通过此方式获得&gt;95%的准确率</li><li>➤误答容忍度低</li></ul>
	智能检索	<ul style="list-style-type: none"><li>➤知识检索形式交互</li><li>➤除了返回LLM总结的答案，可选择展示top K的召回记录以及知识源语料</li></ul>	<ul style="list-style-type: none"><li>➤因为返回多条记录，对知识相关性容忍度相对较高</li><li>➤可使用引导式检索，逐步获取精准答案</li></ul>
按知识类型	FAQ对	<ul style="list-style-type: none"><li>➤按问答对的方式构建知识</li><li>➤知识信息量完整，知识质量高</li></ul>	<ul style="list-style-type: none"><li>➤按FAQ对进行切片，保证语义完整性</li><li>➤现有客服场景通常都有语料积累，知识构建相对高效</li><li>➤上线相对容易</li></ul>
	PDF/word等各类文档	<ul style="list-style-type: none"><li>➤原始文档格式多样，比如各类wiki，产品说明书等，含有表格、图文等信息</li><li>➤知识质量可能参差不齐，密度低</li></ul>	<ul style="list-style-type: none"><li>➤通常按句子，段落等进行切片，较FAQ对切片方式复杂，需要结合实际文档来保证语义完整性</li><li>➤表格需要额外处理，暂时不建议对图片处理</li></ul>
按使用对象	服务内部用户	<ul style="list-style-type: none"><li>➤比如企业内部知识库，IT/HR 知识库</li></ul>	<ul style="list-style-type: none"><li>➤内部人员和并发等因素可控，上线相对容易</li></ul>
	服务外部用户	<ul style="list-style-type: none"><li>➤比如各行业对外的智能客服，游戏的NPC</li></ul>	<ul style="list-style-type: none"><li>➤对并发/吞吐等性能指标有较高要求，另外需要考虑内容风控比如屏蔽涉黄/涉恐/涉暴话题</li></ul>

# RAG实践经验总结

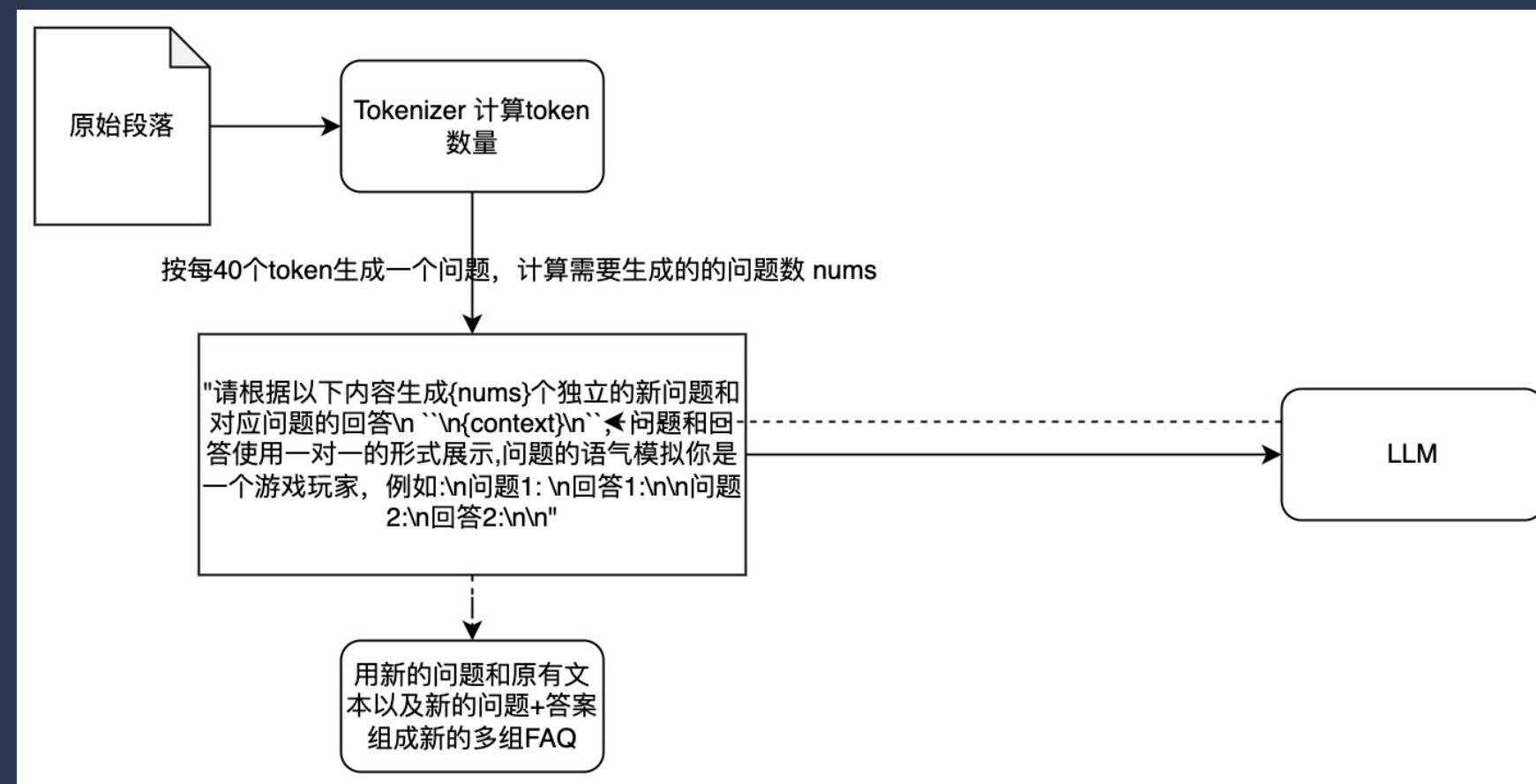
# 技术层面经验总结

- 知识构建 – 知识质量不好咋办？格式复杂还有表格咋办？量大注入慢咋办？
- 核心工作 – 知识召回效果差咋办？LLM幻觉咋办？怎么判断超出知识库范围？
- 上线必备 – 效果如何评估和持续监控？Badcase 怎么排查？Streaming 咋支持？
- 经验洞察 – Agent API 设计思考？反问机制设计？Web Search集成方法？飞书的集成与实现？知识共创UGC机制？

# 知识构建 – 知识质量不好咋办？

✓ 利用商业LLM做知识增强(原始FAQ/文档都适用), 建议结合人审

□ 解决手段:



□ 参考效果:

问：什么是 Amazon EMR？  
答：Amazon EMR 是行业领先的云大数据平台，适用于使用多种开源框架进行数据处理、交互分析和机器学习，例如 Apache Spark、Apache Hive、Presto。借助 EMR，您可以用不到传统本地解决方案一半的成本运行 PB 级分析，并且其速度比标准 Apache Spark 快 1.7 倍以上。

□ 实践检验

某制造业的相关项目投标中作为一个独立价值展示，助力项目投标成功。缓解了招标方对于知识质量方面的担忧

□ 核心目的：知识点更加具体。生成多角度提问，利于召回。

□ 参考实现：[Enhance FAQ.py](#) [Enhance Doc.py](#)

问：Amazon EMR适用于哪些开源框架进行数据处理、交互分析和机器学习？  
答：Amazon EMR适用于使用多种开源框架进行数据处理、交互分析和机器学习，例如 Apache Spark、Apache Hive、Presto。

问：通过使用Amazon EMR，可以以什么样的成本运行PB级分析？  
答：通过使用Amazon EMR，可以用不到传统本地解决方案一半的成本运行PB级分析。

问：Amazon EMR相比标准Apache Spark有什么优势？  
答：Amazon EMR相比标准Apache Spark的速度更快，速度快比标准Apache Spark快1.7倍以上。

# 知识构建 – 格式复杂还有表格咋办?

✓ 结合开源代码+Amazon AIML SAAS服务

## □ 实践检验:

解决海外一个金融POC中，针对基金相关PDF的表格信息提问的问题

## □ 解决方案

- 利用Langchain开源代码实现PDF转HTML，可保留字号和像素位置信息，按字号进行合并，提高分段质量
- 利用Textract提取表格，实现Textract输出到Json信息的转换
- 利用前两步结果进行位置映射，克服Textract不支持中文的问题
- 其他手段(NSP模型&Layout分析)

## □ 使用说明:

- [PDF SPLITER README](#)
- [Workshop \(第三步实验\)](#)

## □ 参考效果:

The screenshot shows a PDF page from Allspring's website. At the top, it says "AS OF DECEMBER 31, 2022 | FACTSHEET | ALLSPRINGGLOBAL.COM". Below that is the "Allspring" logo and the fund name "Common Stock Fund". A purple bar at the top indicates the asset class: "Asset Class: U.S. Equity".  
  
Section 1 (highlighted in red) is titled "FUND STRATEGY". It contains two bullet points:

- Public equity markets are often driven by emotion, requiring successful investors to have conviction in individual securities and diversification across sectors.
- Our team's conviction comes from an in-depth private market valuation (PMV, the price an acquirer would pay to purchase the entire company) process of analyzing the business model, competitive positioning, key trends, management, and other proprietary metrics.

Section 2 (highlighted in red) is titled "Competitive advantages". It contains two bullet points:

- Private market valuation (PMV) approach: By constantly measuring a company's "private market value," the team is better able to assess a company's worth and act decisively when "market emotion" drives the price of a solid business down to discount levels. Additionally, the PMV investment process helps to discern differences between mispriced stocks and those with cheap valuations, improving the team's likelihood to generate alpha.
- Opportunistic core approach: The PMV investment approach is designed to be growth- and value-neutral, with the flexibility to opportunistically invest in the best ideas at either end of the growth and value spectrum.

  
Section 3 (highlighted in red) is titled "Sector allocation (%)" and includes a table comparing the fund's allocation against the Russell 2500® Index.

	Fund	Russell 2500® Index <sup>2</sup>
Industrials	24	18
Information technology	18	14
Consumer discretionary	13	11
Health care	13	13
Financials	11	16
Real estate	9	8
Materials	8	6
Consumer staples	3	3
Communication services	1	3

Sector allocation is subject to change and may have changed since the date specified. Percent total may not add to 100% due to rounding.

  
The page also features sections for "Annual Returns" and "Fund Managers".

```
1 "content":"FUND STRATEGY • Public equity markets are often driven by emotion, requiring successful investors to have conviction in individual securities and diversification across sectors. • Our team's conviction comes from an in-depth private market valuation (PMV, the price an acquirer would pay to purchase the entire company) process of analyzing the business model, competitive positioning, key trends, management, and other proprietary metrics. • We believe that the PMV of a company is much more stable than its associated public market stock price.", "font_size":9, "doc_title":"Common Stock Fund"}, 2 "content":"Competitive advantages • Private market valuation (PMV) approach: By constantly measuring a company's "private market value," the team is better able to assess a company's worth and act decisively when "market emotion" drives the price of a solid business down to discount levels. Additionally, the PMV investment process helps to discern differences between mispriced stocks and those with cheap valuations, improving the team's likelihood to generate alpha. • Opportunistic core approach: The PMV investment approach is designed to be growth- and value- neutral, with the flexibility to opportunistically invest in the best ideas at either end of the growth and value spectrum.", "font_size":12, "doc_title":"Common Stock Fund"}, 3 "content":{ "table":"Sector allocation (%)", "footer":"Sector allocation is subject to change and may have changed since the date specified. Percent total may not add to 100% due to rounding.", "data":{ "row_key":"Industrials ", "Fund ":"24 ", "Russell 2500" Index2 18 ":"Russell 2500" Index2 18 "}, { "row_key":"Information technology ", "Fund ":"18 ", "Russell 2500" Index2 18 ":"14 "}, { "row_key":"Consumer discretionary ", "Fund ":"13 ", "Russell 2500" Index2 18 ":"11 "}, { "row_key":"Health care ", "Fund ":"13 ", "Russell 2500" Index2 18 ":"13 "}, { "row_key":"Financials ", "Fund ":"11 ", "Russell 2500" Index2 18 ":"16 "}, { "row_key":"Real estate ", "Fund ":"9 ", "Russell 2500" Index2 18 ":"8 "}, { "row_key":"Materials ", "Fund ":"8 ", "Russell 2500" Index2 18 ":"6 "}}}
```

# 核心工作 – 知识召回效果差咋办？

□ 有效手段

- ✓ 多路召回： 向量召回 + 倒排召回
- ✓ BM25打分调优
- ✓ 更优的向量模型选型
- ✓ 多种召回范式： 对称召回(Query-Question) + 非对称召回(Query-Document)
- ✓ 微调向量模型
- ✓ 引入Rerank模型
- ✓ 添加IUR步骤(Incomplete utterance rewrite)

□ 面临问题：

- 正确的知识没召回，导致回答没有引用到知识
- 召回了不正确的知识导致LLM产生了误解

□ 技术总结

1. [基于大语言模型知识问答应用落地实践 - 知识召回调优（上）](#)
2. [基于大语言模型知识问答应用落地实践 - 知识召回调优（下）](#)

# 核心工作 – 知识召回效果差咋办?

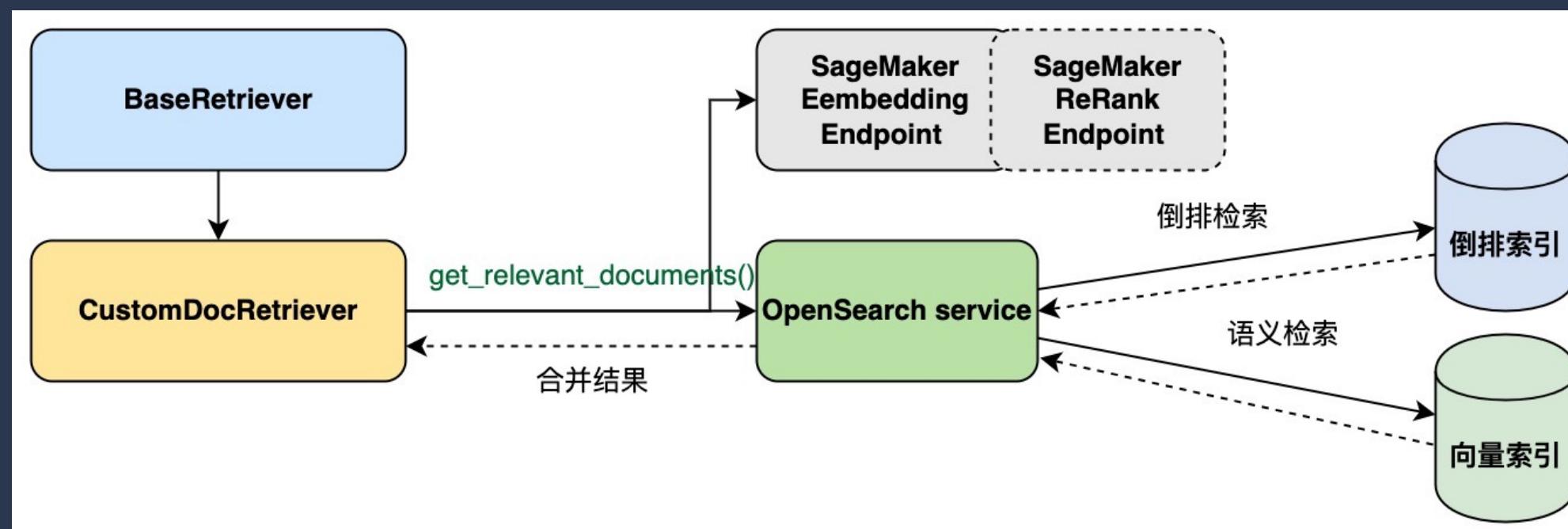
✓ 多路召回：向量召回 + 倒排召回

□ 面临问题：

- 向量模型不理解垂直领域专词
- 向量模型有时出现语义相似但主题不相似情况
- 可解释性弱，不易通过补丁解决bad case
- 倒排召回，缺乏语义信息，仅靠关键词匹配

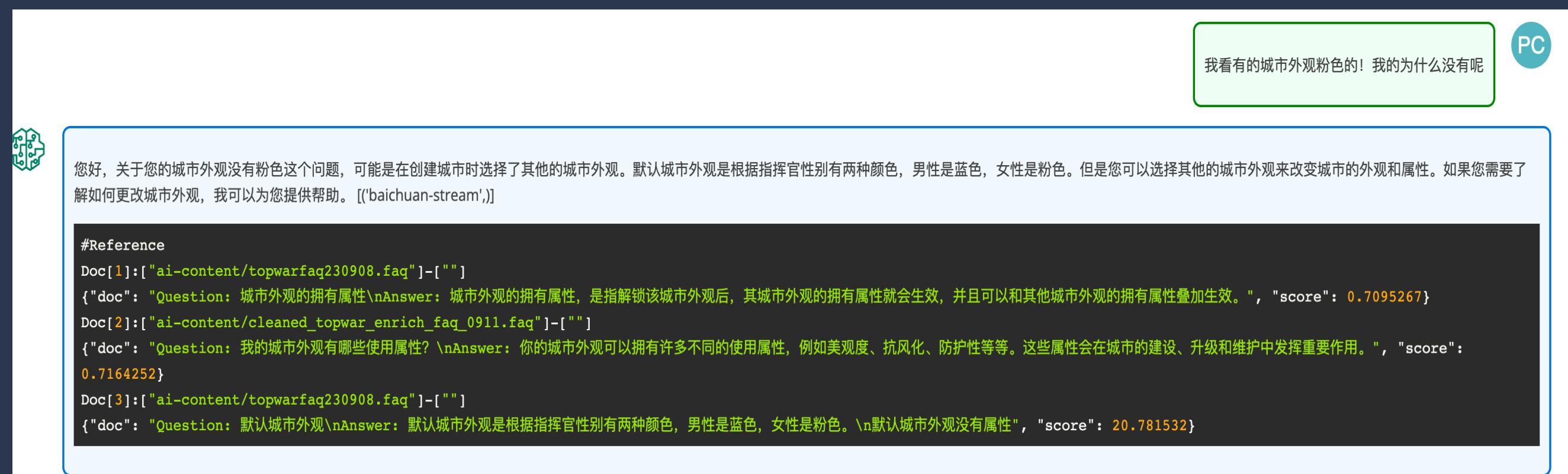
□ 工程实现：

- 重载Langchain Retriaver接口，融合多路召回结果



□ 优势样例：

用户的原始问题中“**城市外观粉色**”的信息，在向量召回的结果中并没有（前面2条 **score < 1** 的结果），向量调优难以解决该 Case，但是在第3条倒排召回（**score > 1**）的知识中，含有相关信息。



# 核心工作 – 知识召回效果差咋办？

✓ BM25打分调优

□ 面临问题：

- 用户的垂直数据中可能某些专词与停用词词频差不多导致IDF失真，引起得分计算有误
- 有些特定‘黑话’，数据不足时语义向量也无法解决

□ 有效手段：

- 构建停用词表，使得停用词均不参与BM25得分
- 构建同义词表，定向解决‘黑话’问题

□ 技术输出

1. blog 基于大语言模型知识问答应用落地实践 - 知识召回调优（上）

□ 视频Demo

如何调整BM25倒排优化知识召回效果

# 核心工作 – 知识召回效果差咋办?

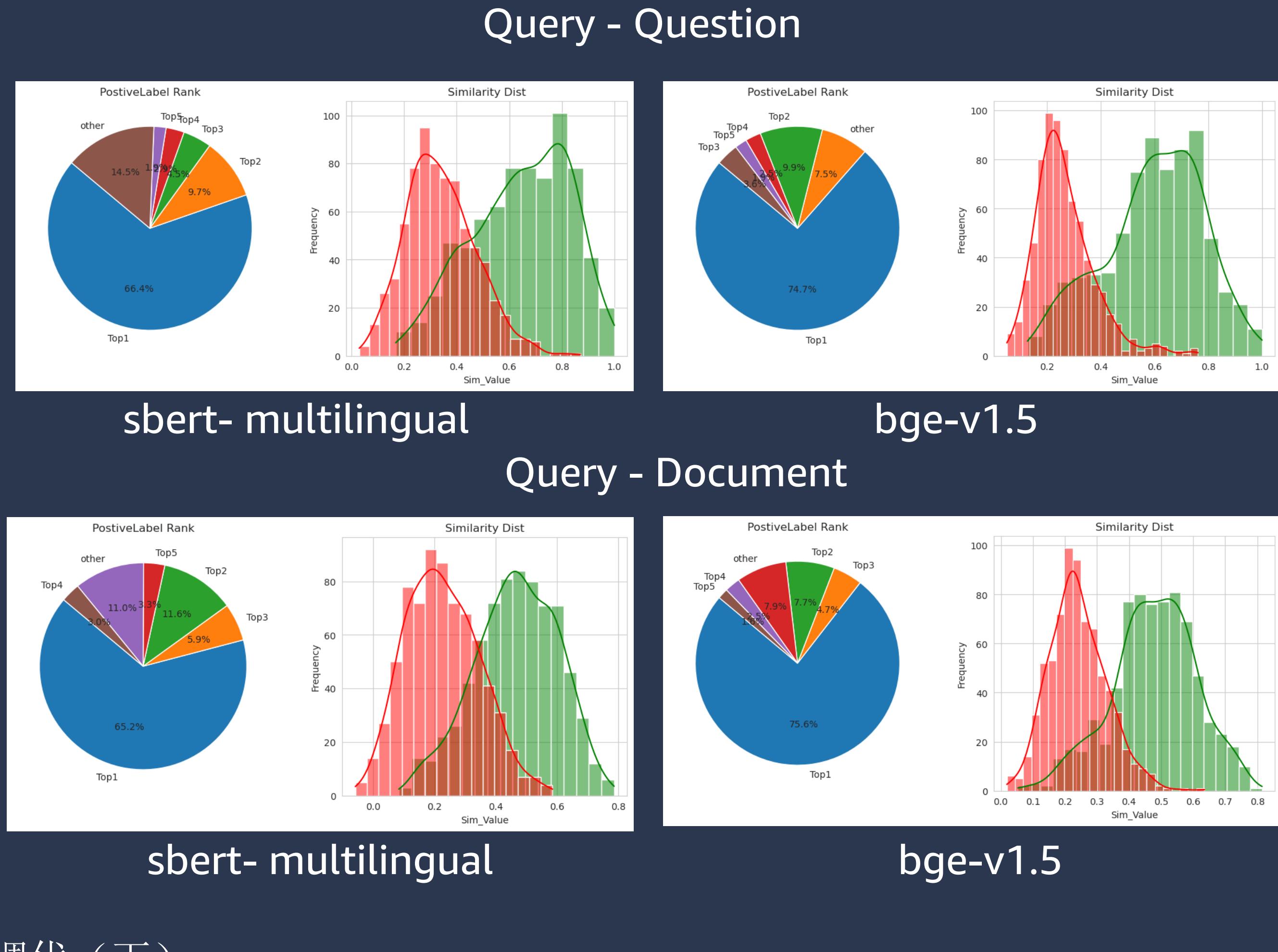
## ✓ 更好的向量模型选型

### □ 面临问题:

- 公开数据集上的表现，在垂直领域没有特别大的参考性
- 在自己数据场景中，通过少量case手工测试无法获取全面信息，难以客观比较

### □ 价值提供:

- 给出标准化的评估方式和可视化方法



### □ 基本结论:

- 优选bge-large-zh-v1.5
- 优选bge-large-en-v1.5

### □ 技术输出

1. 代码实现 [bge\\_zh\\_research.ipynb](#)
2. Blog 基于大语言模型知识问答应用落地实践 – 知识召回调优（下）

# 核心工作 – 知识召回效果差咋办?

✓ 多种召回范式： 对称召回 (Query-Question) + 非对称召回(Query-Document)

□ 面临问题：

○ 两种形式各有弊端

- 对于垂直领域做QD召回需要向量模型具备很强的理解能力，需要用这个领域数据的训练过
- 用户query中的一些信息只出现在知识的Document/Answer中，通过Query-Question匹配难度大

□ 例子：

用户的发问角度，或者query-Question的语义相似性不一定高

□ 视频Demo

知识问答中的对称召回+非对称召回策略

```
1 Question: AWS Clean Rooms的数据源必须在AWS上么?  
2 Answer: 对，目前必须在AWS上，而且必须是同一个region。  
3 ======  
4  
5 user : Clean Rooms的数据源可以不在同一个region么?
```

# 核心工作 – 知识召回效果差咋办?

## ✓ 向量模型微调

### □ 面临问题:

- 场景数据过于垂直，通用的模型表现不佳

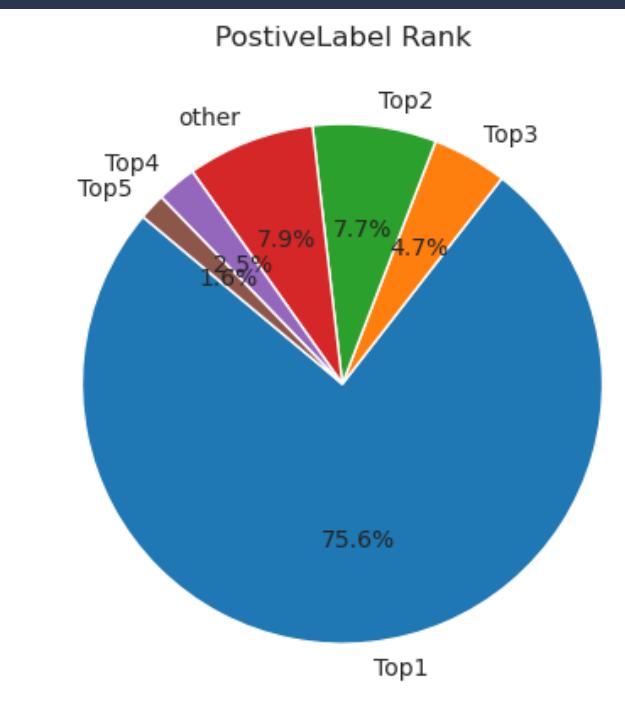
### □ 价值解读:

- 在训练集上效果非常好，意味着后续可以通过持续收集用户反馈，并纳入到训练集以更新模型，使得这个效果不断扩大覆盖范围。
- 测试集上效果没有下降，反而有小幅提升，意味着训练没有破坏模型原有语义能力，对于未被训练集覆盖到的场景，模型仍能以优于原模型的性能进行服务

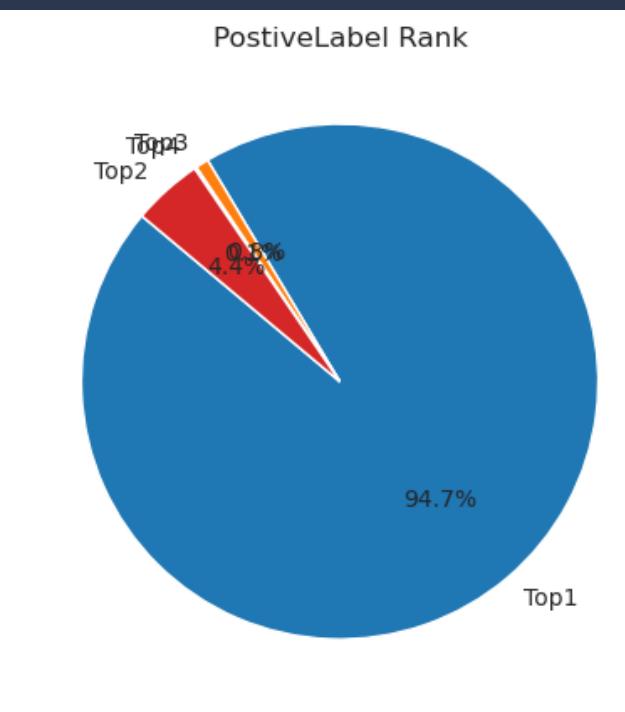
### □ 技术输出

1. 代码实现 [bge\\_zh\\_research.ipynb](#)，包含训练数据构造，训练部署
2. [blog 基于大语言模型知识问答应用落地实践 – 知识召回调优（下）](#)

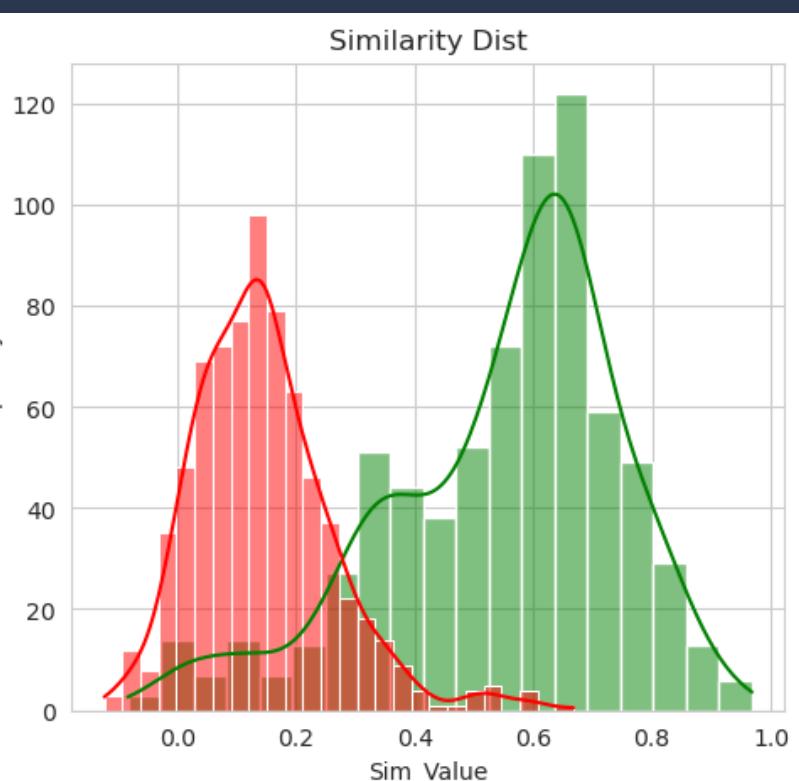
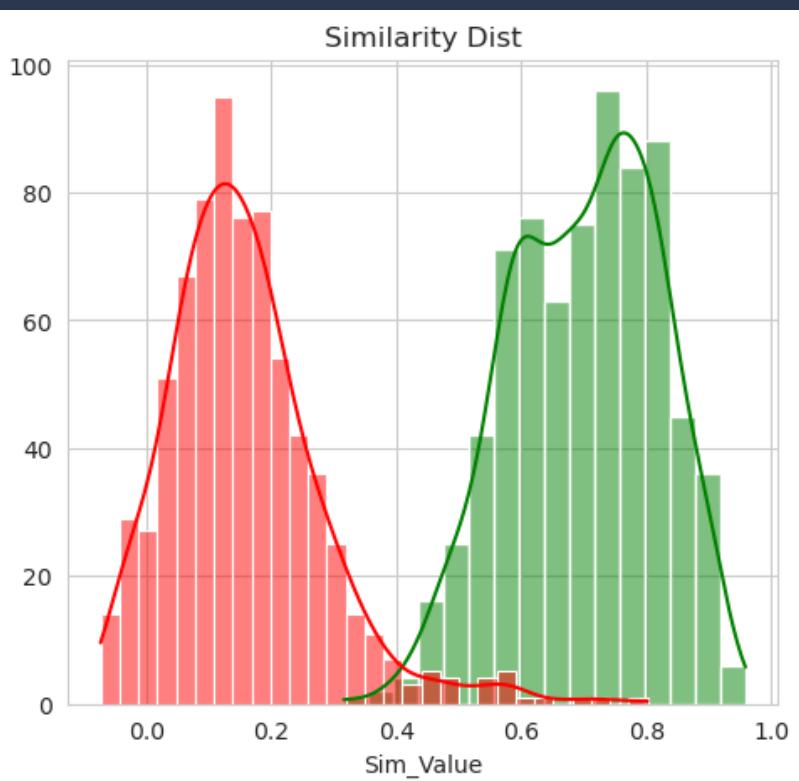
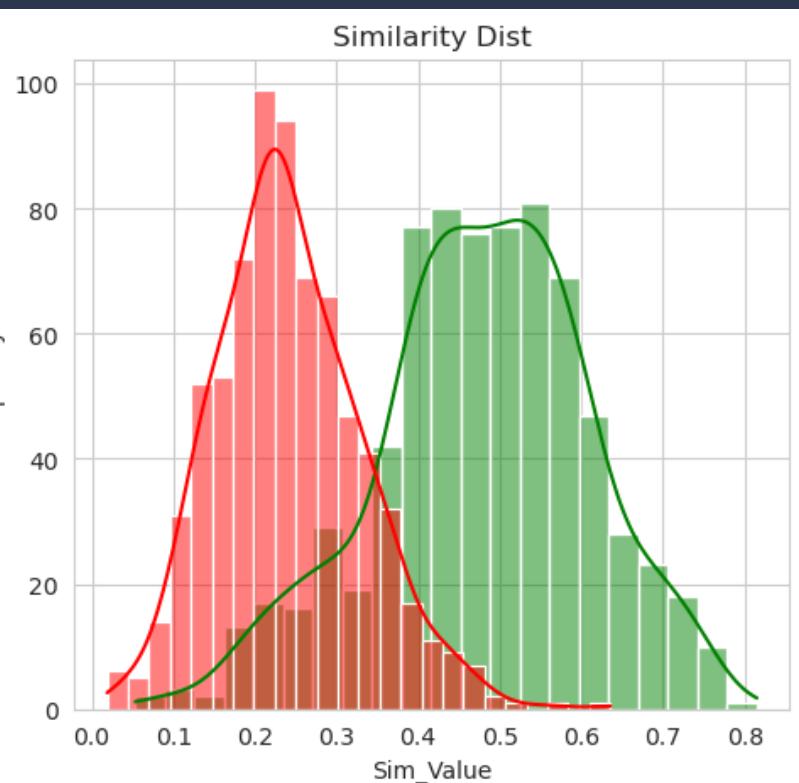
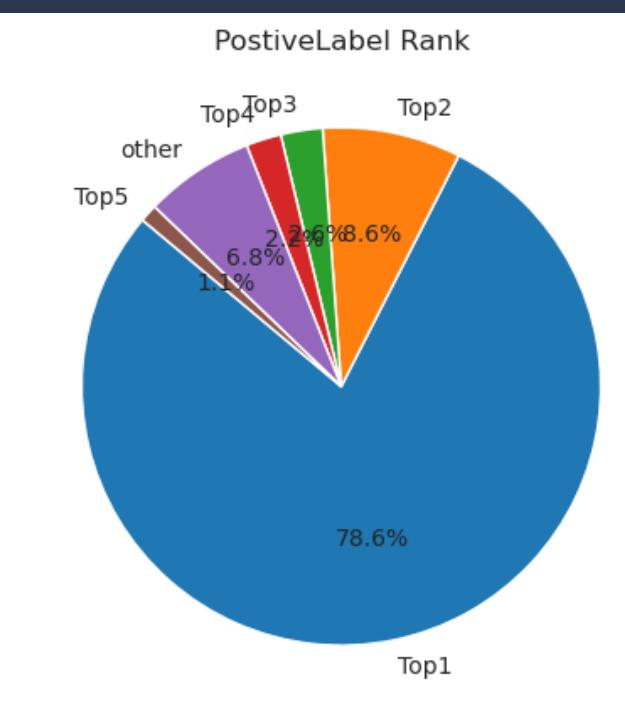
原始



微调后训练集



微调后测试集



# 核心工作 – 知识召回效果差咋办？

✓ 引入Rerank模型

□ 面临问题：

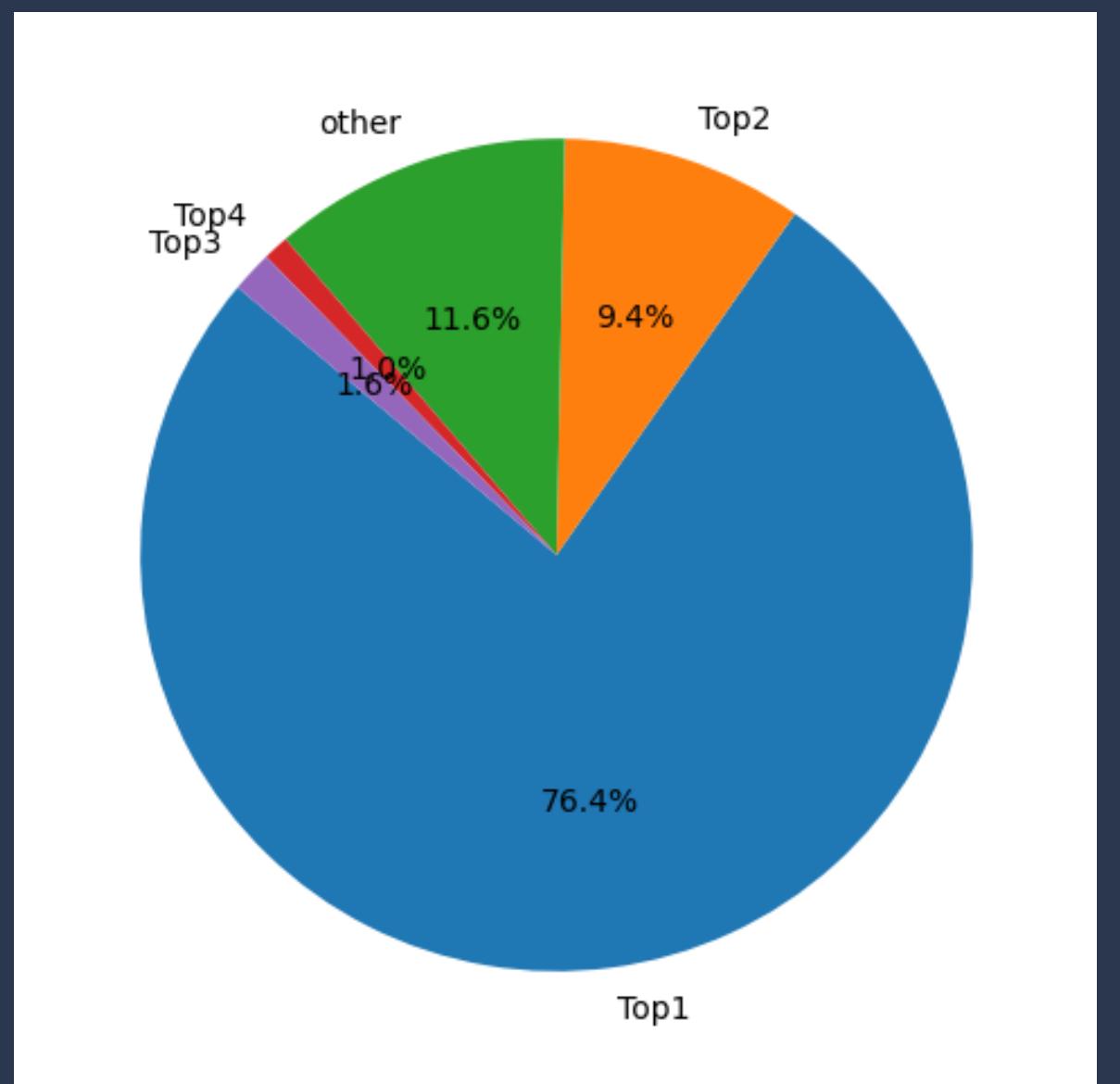
- 向量召回与倒排召回的评分体系不一致，只能随便各取TopK，缺乏依据

□ 价值解读：

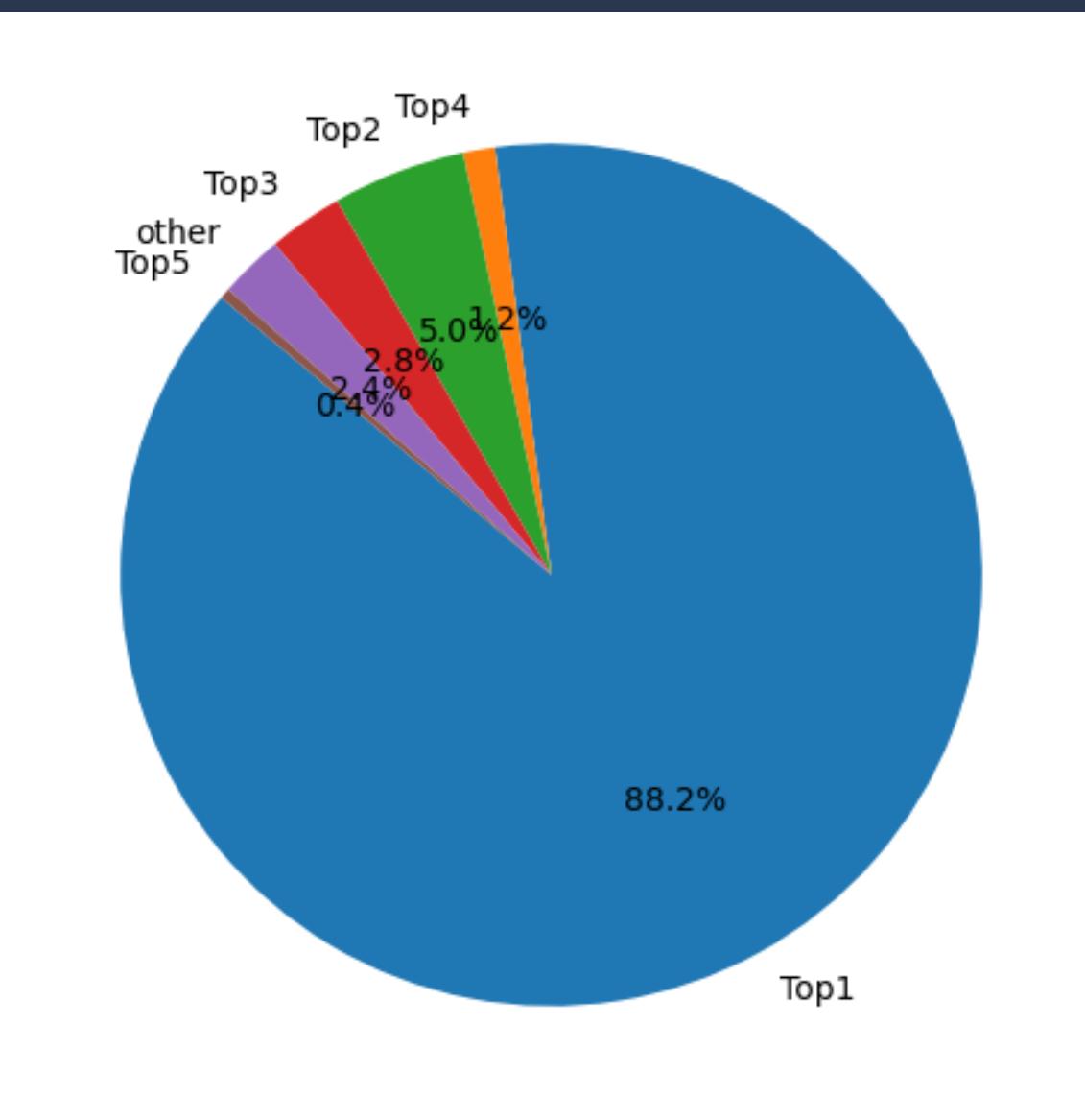
- 解决两路合并各取多少的问题。
- 进一步提高知识召回质量，作为一个独立插拔模块解决向量模型微调解决不了的问题

□ 技术输出

1. 代码实现 [bge\\_zh\\_research.ipynb](#)，包含难样本挖掘，训练部署等
2. [blog 基于大语言模型知识问答应用落地实践 – 知识召回优（下）](#)



无Rerank的正例排名



有Rerank的正例排名

# 核心工作 – 知识召回效果差咋办？

## ✓ 添加IUR步骤(Incomplete utterance rewrite)

### □ 面临问题：

- 在多轮对话情形下，用户的当前输入会存在一些隐含的指代关系和信息省略。缺乏上下文信息的语义缺失严重无法有效召回

### □ 解决思路：

- 利用LLM进行当前query的重写，对上下文隐含信息重新纳入到新生成的query中。重写效果好，但多调用一次LLM，会加重全流程latency问题
- 部署一个独立的IUR模型。重写效果没有前者好，但收集到数据后，可以基于采集数据进行微调，更加适应特定场景。

### □ 技术输出

- 参考代码 [query\\_rewrite.py](#)
- IUR 可用模型及部署方法 <https://huggingface.co/csdc-atl/dialogue-rewriter>

### □ 例子：

<history>

User: "有戴森的吹风机吗？"

Bot: "没有哦亲亲"

User: "戴森都没有"

Bot: "不好意思，看看其他品牌呢"

</history>

<query>"那有松下的吗"</query>

<rewrite>有松下品牌的吹风机吗?</rewrite>

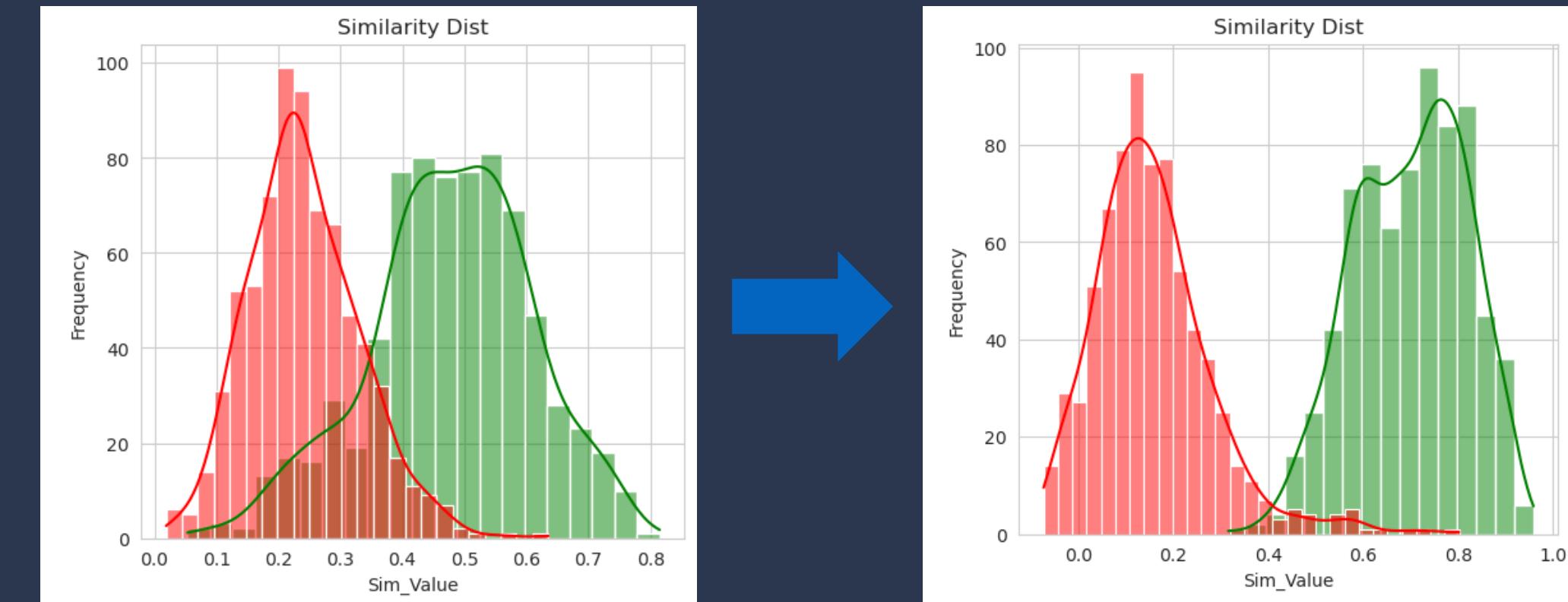
# 核心工作 – LLM幻觉咋办？怎么判断超出知识库范围？

✓ 根据多级召回阈值采取灵活降级策略并结合意图识别

□ 面临问题：

- 由于LLM幻觉编造一些内容可能误导用户，在某些情况下造成的影响非常大（跟钱，账单相关的）
- 通过Prompt提示LLM，要它依据知识不要胡说的方法是不靠谱
- 用户可能还会问一些不在服务范围的话题，滥用服务引起GPU浪费。

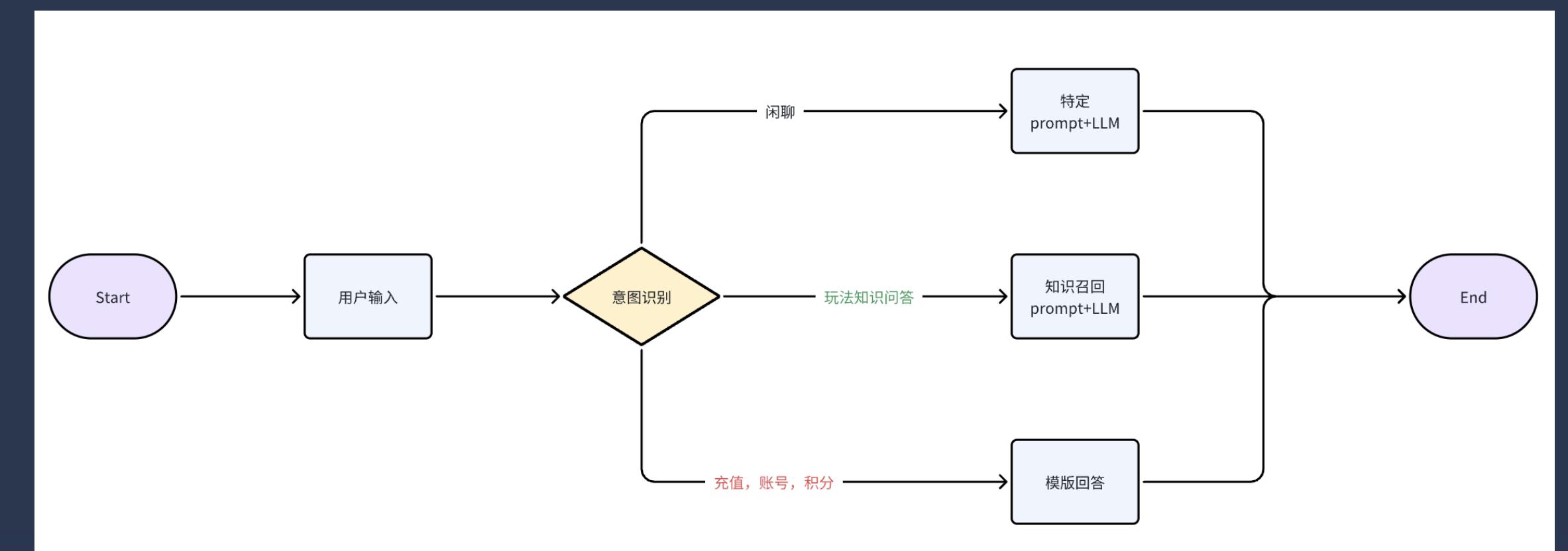
拉开值域分布



□ 有效手段：

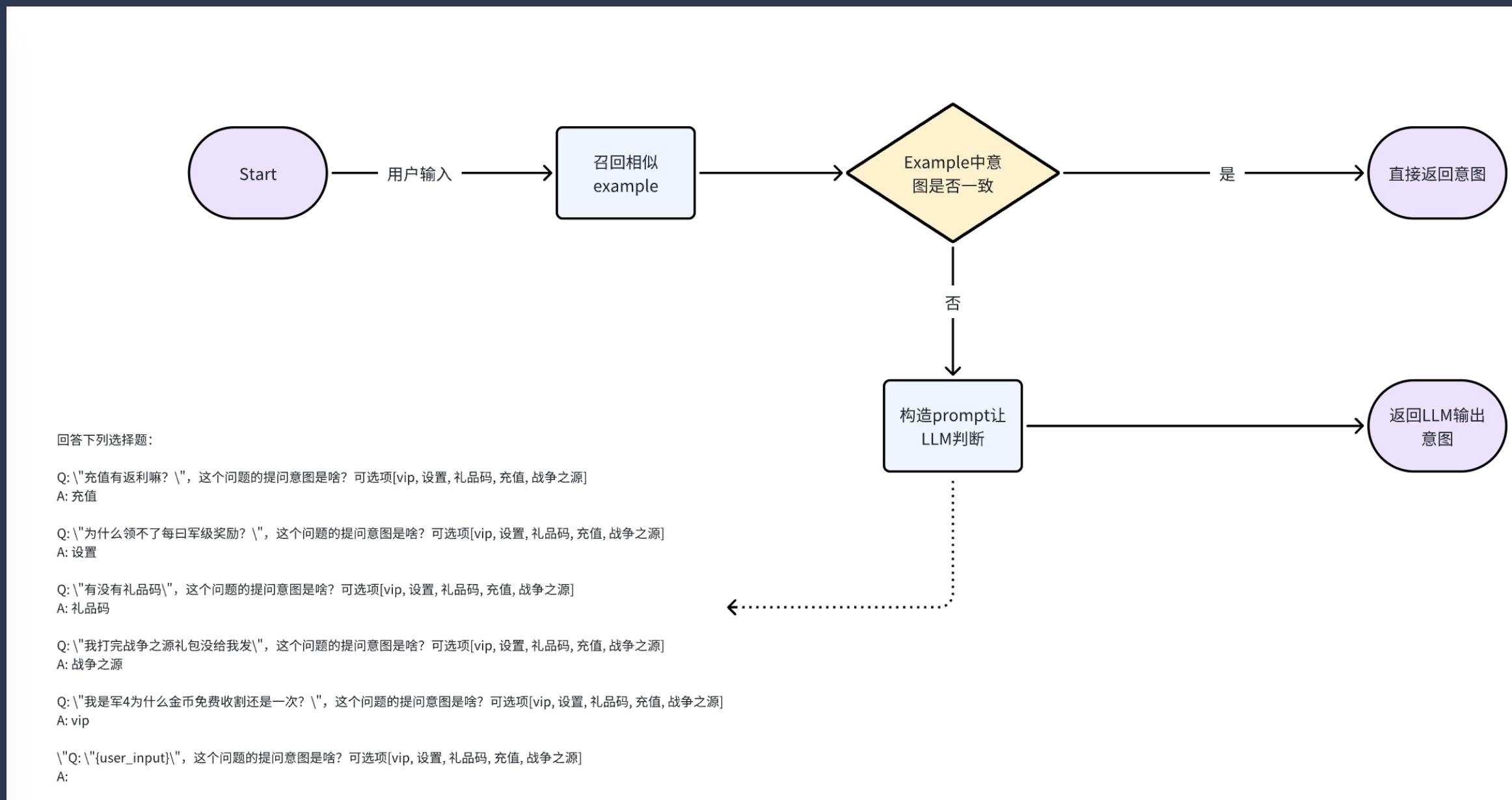
- 通过收集反馈数据，微调向量模型叠加微调Rerank模型，拉开相关召回和非相关召回的得分的值域分布。得分分为多级采用不同策略
  - 最置信的走LLM
  - 次置信的提示LLM如果不相关进行拒答
  - 不置信的仅返回召回TopK或直接拒答
- 意图识别进行场景分流，敏感场景避免LLM介入直接走预制答案。

意图识别避免敏感场景幻觉的示意图



# 核心工作 – 意图识别模块

## □ 实现方式



## □ 优势特点

- 简单易用，不太需要太多算法能力
- 随着积累example越多越准，越不需要LLM参与，性能越好
- 生活场景的用户输入比较易用，非生活场景可能需要向量微调

## □ 实践检验

采用bge\_large\_en向量模型 + Bedrock Claude大模型结合这个技术方案。

在一个IOT领域的的意图识别场景的POC中，把之前基于OpenAI的准确率从80%+提升到90%+

## □ 技术输出

1. 使用说明 [README.md](#)
2. 参考代码 [intention.py](#)

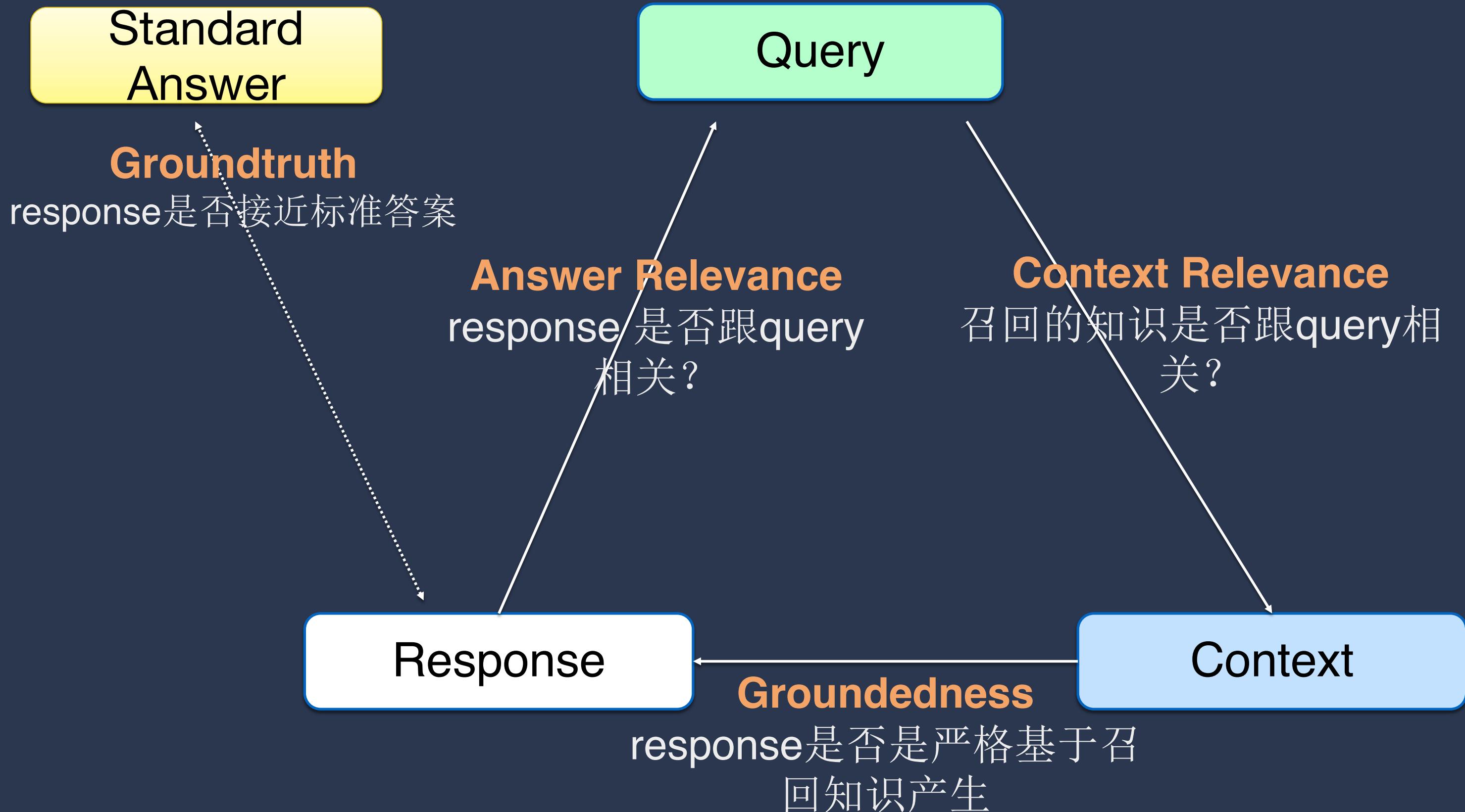
## □ 视频Demo

[知识问答中的意图识别方案](#)

# 上线必备 – 效果如何评估和持续监控

✓ 维护评估集 + 量化评估框架

测试另一个版本的提示词模板是否有改进?



Template v1 vs v2

Ground Truth	Groundedness	Context Relevance	Answer Relevance
0.9	0.5	0.2	1
0.9	1	1	1
1	1	1	1
0.7	0.525	1	1
0.9	1	1	1
0.9	0.9	1	1
0.9	1	1	1
0.9	1	1	1
0.9	1	1	1
0.9	0.5	1	1
0.9	0.75	1	1
0.9	0.8333333333	0.8	1
0.9	1	1	1
0.7	0.75	1	1
0.2	0.75	1	1
1	1	1	1
1	1	1	1
1	1	1	1
0.9	1	0.4	1
0.9	1	1	1
1	1	1	1
0.9	1	1	1
0.3	1	1	1
0.9	0.75	1	1
1	1	1	1
0.9	0.2	1	1
0.9	1	1	1
1	1	1	1
1	1	1	1
0.9	1	1	1
0.9	0.8571428571	1	1
0.9	1	1	1
0.9	0.8	1	1

结论: v2的提示词模板反而使回答质量下降了

# 上线必备 – BadCase 怎么排查?

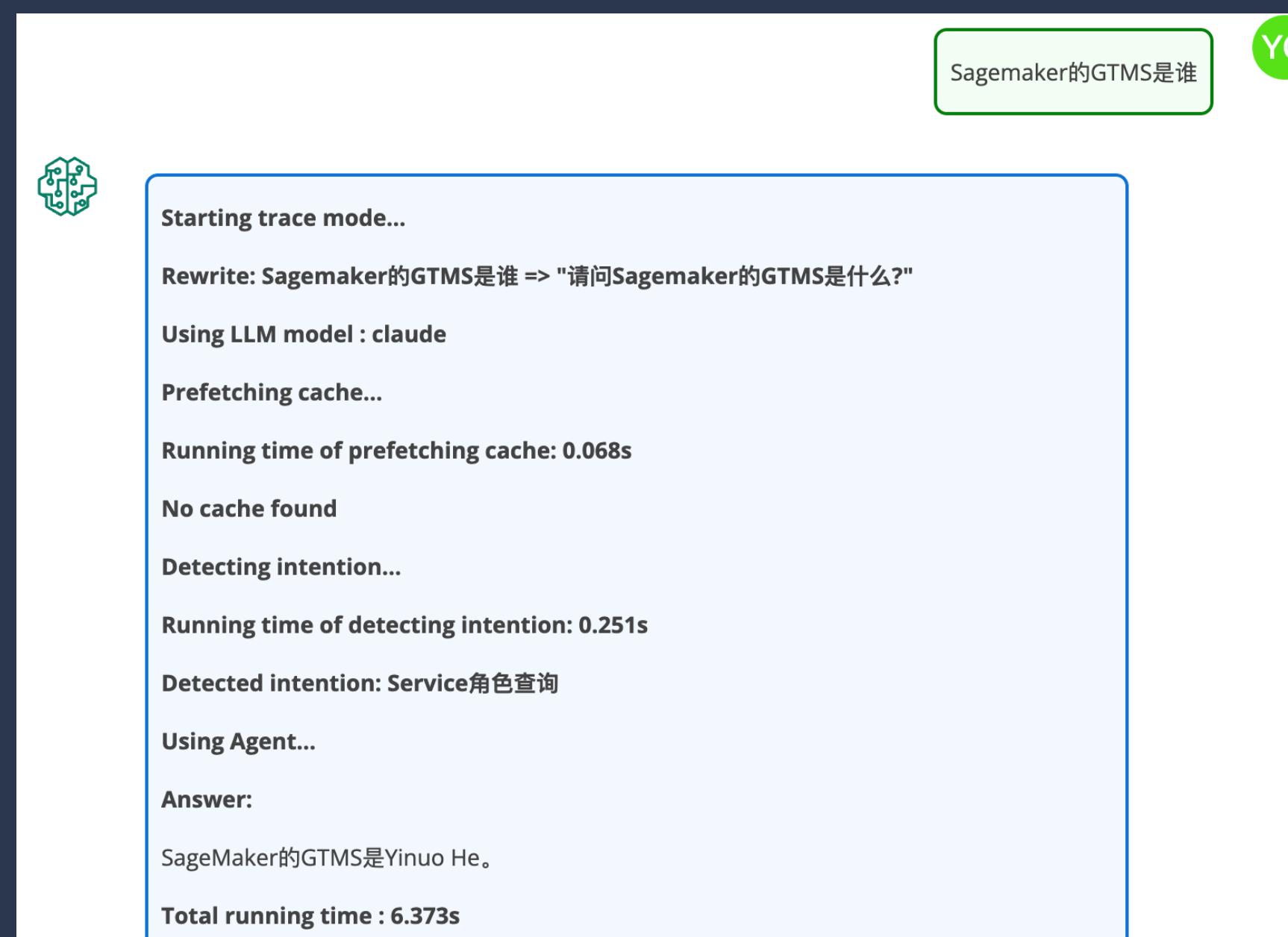
✓ 全流程日志输出并自动回流 + AOS 做日志检索 + 前端信息显示

## □ 面临问题:

- RAG项目不是一蹴而就的，需要长期的优化调优
- 链路中多阶段的中间结果需要独立调优

## □ 前端信息显示:

- 主要用于快速了解召回质量，提升排查速度



The screenshot shows the Amazon OpenSearch Dashboard Discover page with the following annotations:

- 识别的意图 (Detected Intention): Points to the "detect\_query\_type" column in the table.
- 构造的Prompt (Constructed Prompt): Points to the "LLM\_input" column in the table.
- 各路召回的知识&打分 (Recalled knowledge & Scoring): Points to the "opensearch\_knn\_doc" column in the table.
- LLM 版本 (LLM Version): Points to the "query" column in the table.
- 用户的query (User's query): Points to the "query" column in the table.

The table displays log entries with columns: Time, detect\_query\_type, knowledges, LLM\_input, query, opensearch\_knn\_doc. One entry is expanded to show the JSON structure of the query and its results.

# 上线必备 – BadCase 怎么排查?

✓ 通过用户在线反馈，对知识库进行修正，补充

□ 提供用户反馈搜集接口，对反馈问题进行排查。用户纠正过的答案可以作为新的FAQ知识，补充进知识库

The figure consists of three screenshots illustrating the process of collecting user feedback and updating knowledge bases.

**Screenshot 1: User Feedback Collection (Left)**

A screenshot of a user interface showing a conversation. A message from "AN" says: "GPU 内存限制会导致什么瓶颈". Below it, a box contains text about memory limits and training models. At the bottom, there are three buttons: "帮我纠正" (highlighted with a red border), "撤回" (收回), and "发送" (Send).

**Screenshot 2: Feedback Management (Middle)**

A screenshot of the AWS Chat Portal's Feedback Management page. It shows a list of feedback items. One item is highlighted: "怎么看现有的 Capacity?". The original answer is: "根据context内容,可以使用https://ec2-baywatch-prod-iad.iad.proxy.amazon.com/pages/poolViewer 查看现有的Capacity。 Tips:通常情况下使用默认值搜索即可。". On the right, there are buttons for "New Answer" and "Feedback".

**Screenshot 3: Knowledge Base Update (Bottom)**

A screenshot of a modal window titled "帮我纠正". It contains a list of corrections: "1. 模型的规模", "2. 由于内存限制", and "3. 标准数据并行技术". Below this is a text area labeled "Your answer" with a placeholder "Your answer". At the bottom are "取消" and "提交" buttons.

# 经验洞察 – 反问机制设计?

✓ 通过知识的元数据来提示LLM

□ 面临问题:

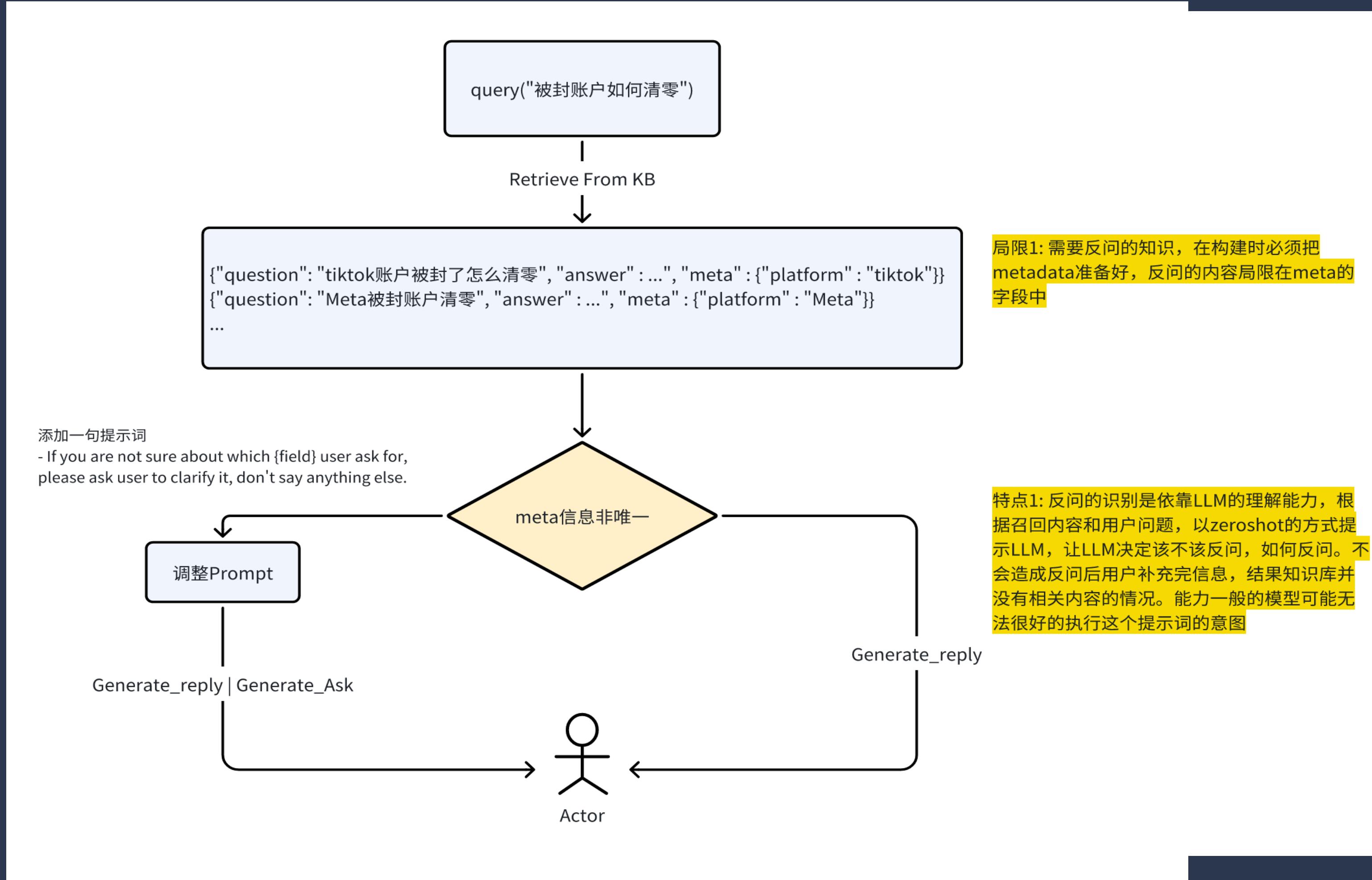
○ 如何判断是否需要反问?

LLM缺乏特定业务的理解，LLM在训练时的默认状态是给出一个泛泛的回答或拒答，反问较少。需要明确的提示词引导他进行反问。

○ 如何知道该反问什么?

只有明确的反问方向，用户才能知道下一步如何做

## 目 愚路1



# 经验洞察 – Agent API设计思考

✓ API 提供LLM易解读的报错信息

□ 面临问题：

- 用户的自然语言不可能很精确
- 结构化查询的字段值可能会存在错误

□ 例子：

- 用户的问题一般是：“美西2的g5.2x什么价格？”

但接口的schema如下表，存在多个字段才能定位price

region	price	instance_type	term	purchase_option
us-west-2	1.052\$	g5.2xlarge	On_Demand	All Upfront
ap-southeast-1	1.052\$	g4dn.2xlarge	Reserved	All Upfront

- 用户的问题可能有输入错误，比如“lex的产品经理是谁？”，注意“lex != Lex”

employee	role	domain	scope
Sofia	Product Manager	AIML	Lex
Jason	Tech	AIML	Lex

□ 手段：

- 查询接口的字段约束尽可能宽松, 单条件>复合条件
- 返回条数过多再叠加条件，或者则提示反问，并给出反问方向
- 返回条数过少，则进行近似查询(比如按照编辑距离/同义词映射)，把可能的取值通过接口返回。

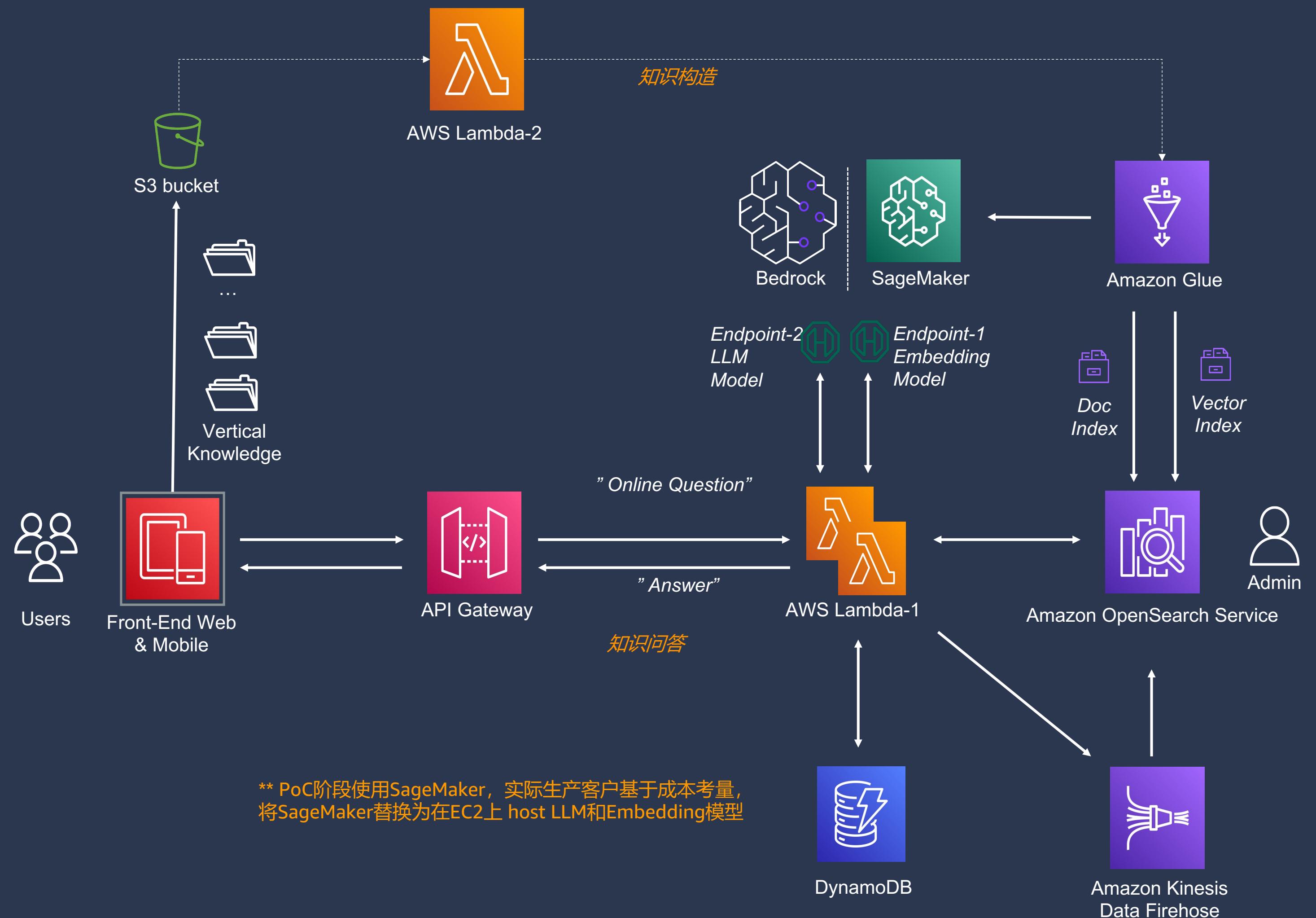
The screenshot shows a user input "lex的问题该找谁?" and an AI's processing log:

- Starting trace mode...
- Using LLM model : claude-v2
- Prefetching cache...
- Running time of prefetching cache: 0.130s
- No cache found
- Detecting intention...
- Running time of detecting: 1.588s
- Detected intention: get\_contact
- Using Agent...
- Function call result:
- 本次回答基于使用工具[get\_contact]为您查询到结果:
- {"statusCode": 404, "message": "Can't find relevant information by - lex.", "suggested\_question": "Lex的问题该找谁? "}
- Answer:
- 很抱歉,我无法根据您的问题"lex的问题该找谁?"找到任何相关信息。API建议您可以尝试询问"Lex的问题该找谁?"。
- Total running time : 3.674s

# RAG场景中的亚马逊云产品亮点

# 建议的整体技术方案

- LLM结合私有知识库，保证知识内容的准确性和时效性
- 方案中的组件可灵活插拔，可根据业务需求可选择不同的大语言和embedding模型，例如Bedrock
- 采用全托管服务或serverless，有效降低用户管理和运维成本
- 基于CDK快速部署



# Bedrock 在RAG上的优势

- ✓ Claude-V2 更强的语义理解和指令执行能力（长窗口，拒答，Agent，富文本）

Claude-V2.1 支持200k token ≈ 470页

- 能够覆盖大部分的企业内完整文档，绝大部分(99%+)的文章章节都能被覆盖
  - 召回阶段，可以支持文档粒度的输入，比chunk的召回更加简单
  - 数据处理阶段，可以把难以切分的上下文依赖严重的文章整体作为Claude-V2.1输入，让大模型进行知识整理和增强
- 经过Prompt优化后，在200k的长窗口中可以高准确率的内容定位，具体参考报告[claude-2-1-prompting](#)

# Bedrock 在RAG上的优势

✓ Claude-V2 更强的语义理解和指令执行能力（长窗口，拒答，Agent，富文本）

Claude-V2

Sagemaker 有什么主要的优势？

YO



对不起,我在提供的文档中没有找到与Sagemaker相关的信息。我无法确定Sagemaker的主要优势。

**Refer to 3 knowledge:**

[1] [ai-content/aws\_msk.faq] [FAQ] [7.578] auther:[s3\_upload]

Question: 什么是造成数据丢失的主要原因? Answer: 造成数据丢失的主要原因是: leader与replica之间存在同步落后, 且允许选举out-of-sync replica为leader。

[2] [ai-content/aws\_msk.faq] [FAQ] [0.624] auther:[s3\_upload]

Question: AWS MSK提供了哪些优势? Answer: AWS MSK提供了基础设施稳定运行的保障, 例如底层的EC2, EBS卷等, 分别有其对应的SLA保障。

[3] [ai-content/aws\_msk.faq] [FAQ] [0.616] auther:[s3\_upload]

Question: AWS MSK相较于传统的自建式Kafka集群有哪些优势? Answer: AWS MSK相较于传统的自建式Kafka集群有以下几点优势:

- 快速部署
- 降低运维复杂度
- 弹性扩展
- 与其他AWS服务紧密集成
- 推出Serverless版



# Bedrock 在RAG上的优势

- ✓ Claude-V2更强的语义理解和指令执行能力（长窗口，拒答，Agent，富文本）

```
▶ [25] 1 response = zero_shot_agent("\n\nHuman: Is Tesla a good investment choice right now? \n\nAssistant:")
[25] ✓ 1m 15.5s
...
> Entering new AgentExecutor chain...
Question: Is Tesla a good investment choice right now?

Thought: I need to first get the company name and ticker for Tesla.

Action: get company ticker

Action Input: Tesla

> Entering new SQLDatabaseChain chain...

Human: What is the ticker symbol for Tesla in stock ticker table?

Assistant:
SQLQuery:SELECT symbol FROM stock_ticker WHERE name LIKE '%Tesla%'
SQLResult: [('TSLA',)]
Answer:TSLA
> Finished chain.

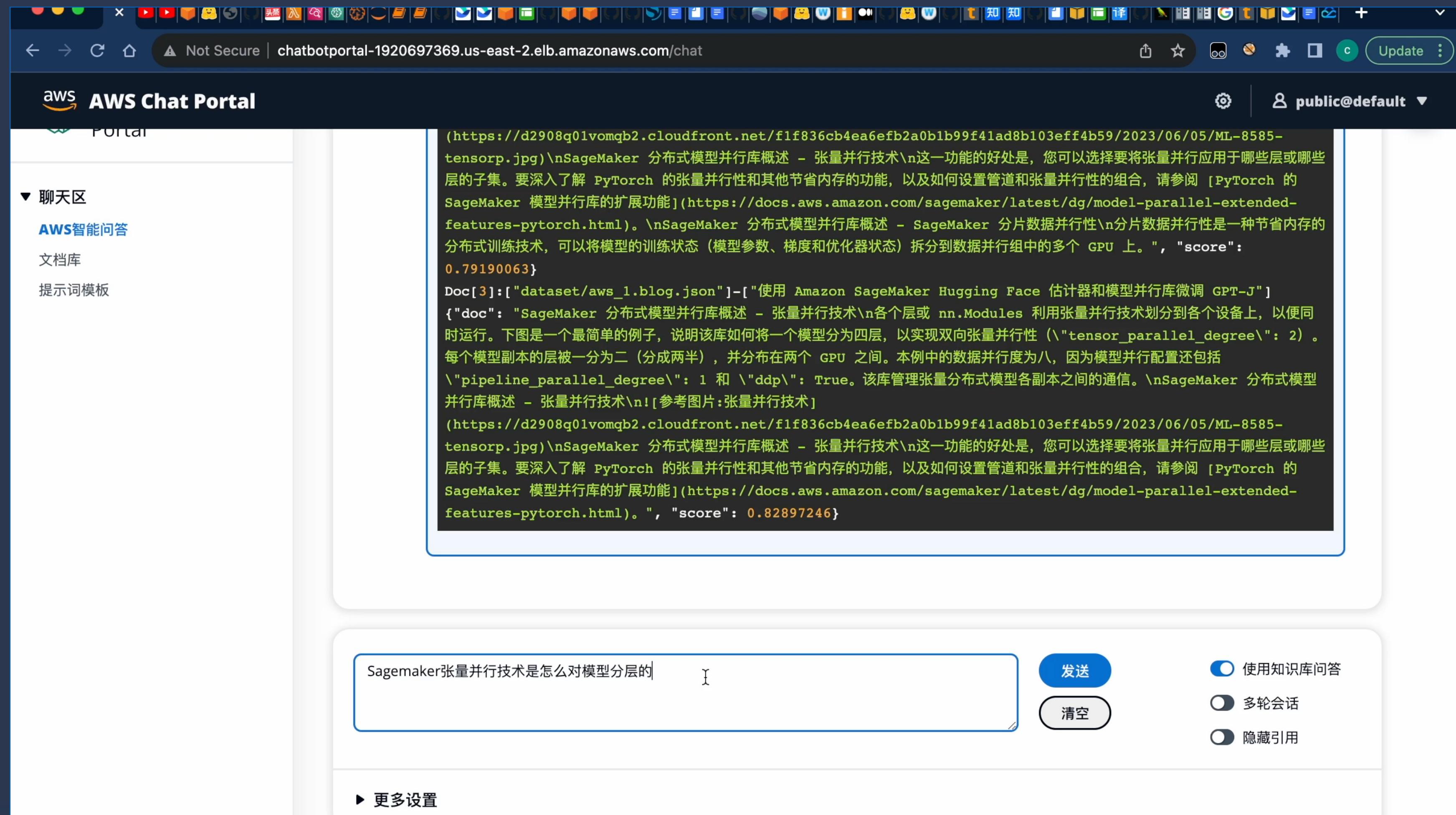
Observation: ('Tesla', 'TSLA')
Thought: Now that I have the ticker symbol TSLA for Tesla, I will get the stock data.

Action: get stock data
Action Input: TSLA
TSLA
Final Answer: Tesla is a speculative buy right now for investors with a high risk tolerance given the potential but uncertainty around the company's ambitious plans. More conservative investors may be better off waiting for more consistent financial results.

> Finished chain.
```

# Bedrock 在RAG上的优势

- ✓ Claude更强的语义理解和指令执行能力（长窗口，拒答，Agent，富文本）



# Bedrock 在RAG上的优势

✓ Claude Instant的响应速度(大小模型结合， 实现能力， 价格， 延迟之间的平衡)

Latency	P50	P75	P90	P99	Input/1k	output/1k
Claude-V2	2.65s	4.68s	7.99s	14.40s	0.01102\$	0.03268\$
Claude-Instant	1.34s	1.54s	1.95s	2.02s	0.00163\$	0.00551\$

## With Claude-V2

应对复杂的上下文输入和指令

举例：RAG问答

## With Claude-Instant-V1

应对简单的/容易定义的任务

举例：

1. Few-shot 意图分类

2. IUR 任务 - Query重写

□ 意图识别

Prompt:

参考下列Example，回答下列选择题：

<example>

Query: \"what is Free slots in baywatch?\", 这个问题的提问意图是啥？可选项[QA, Price]

Answer: 知识问答

...

</example>

Query: \"Baywatch里Free slots是什么意思？\",

这个问题的提问意图是啥？可选项[QA, Price]

Answer:

□ IUR 任务

<history>

User: "MSK能和哪些服务集成？"

Bot: "与IAM和..."

</history>

<query>“那EMR呢”</query>

<rewrite>EMR能和哪些服务集成?</rewrite>

# Bedrock 在RAG上的优势

## ✓ Cohere Embedding (100+ Multilingual能力)

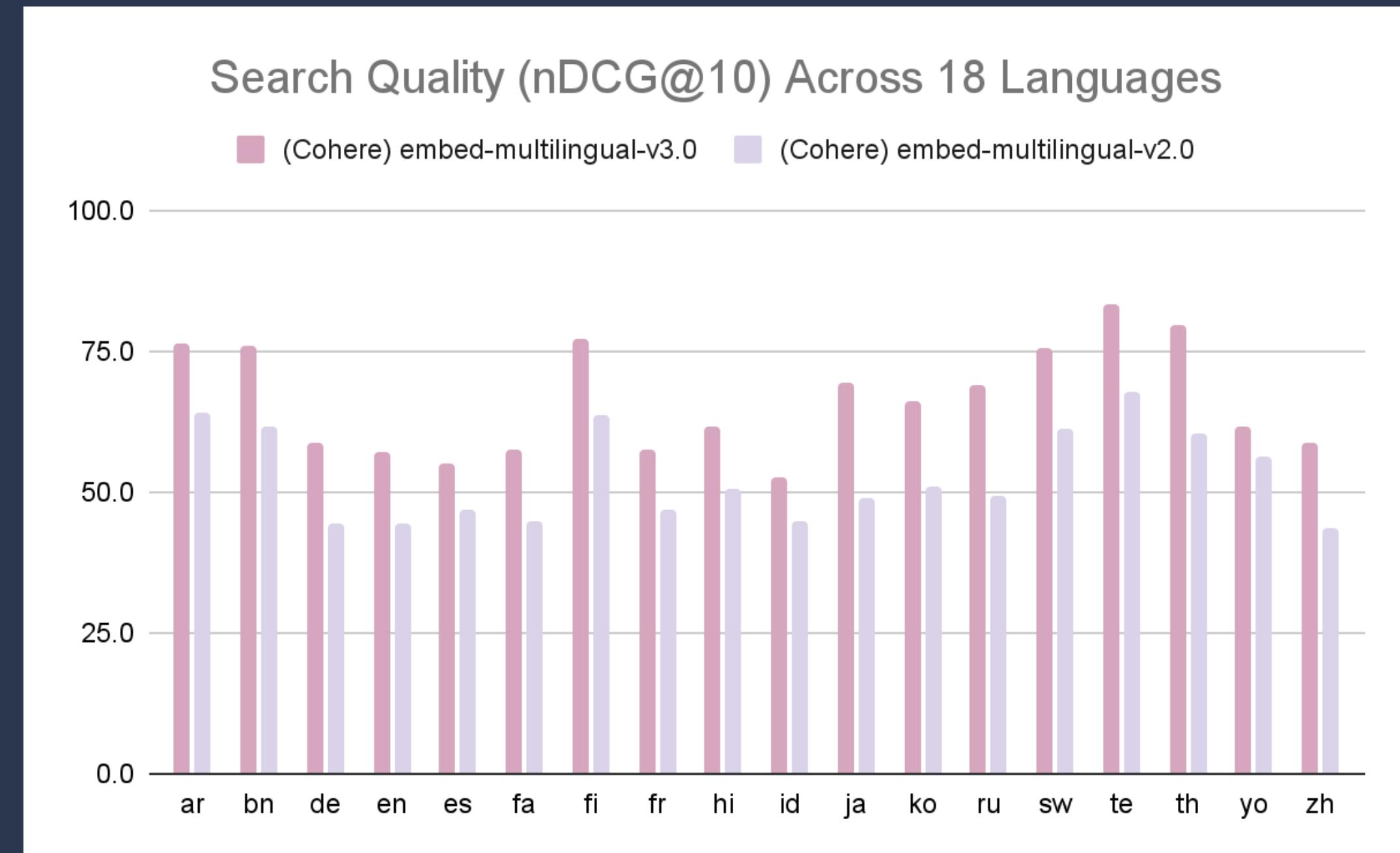
效果优秀

- 英文上超过bge-en-large-v.15
- METB Leaderboard 排名第二

Model	Dimensions	BEIR (nDCG@10, 14 datasets, higher=better)
embed-english-v3.0	1024	55.9
embed-english-light-v3.0	384	52.0
embed-multilingual-v3.0	1024	54.6
embed-multilingual-light-v3.0	384	50.9
<b>Other Models</b>		
BM25	-	43.0
OpenAI ada-002	1536	49.8
<a href="#">GTR-Base</a>	768	44.1
GTR-XXL	768	48.6
<a href="#">DupMAE</a>	768	49.7

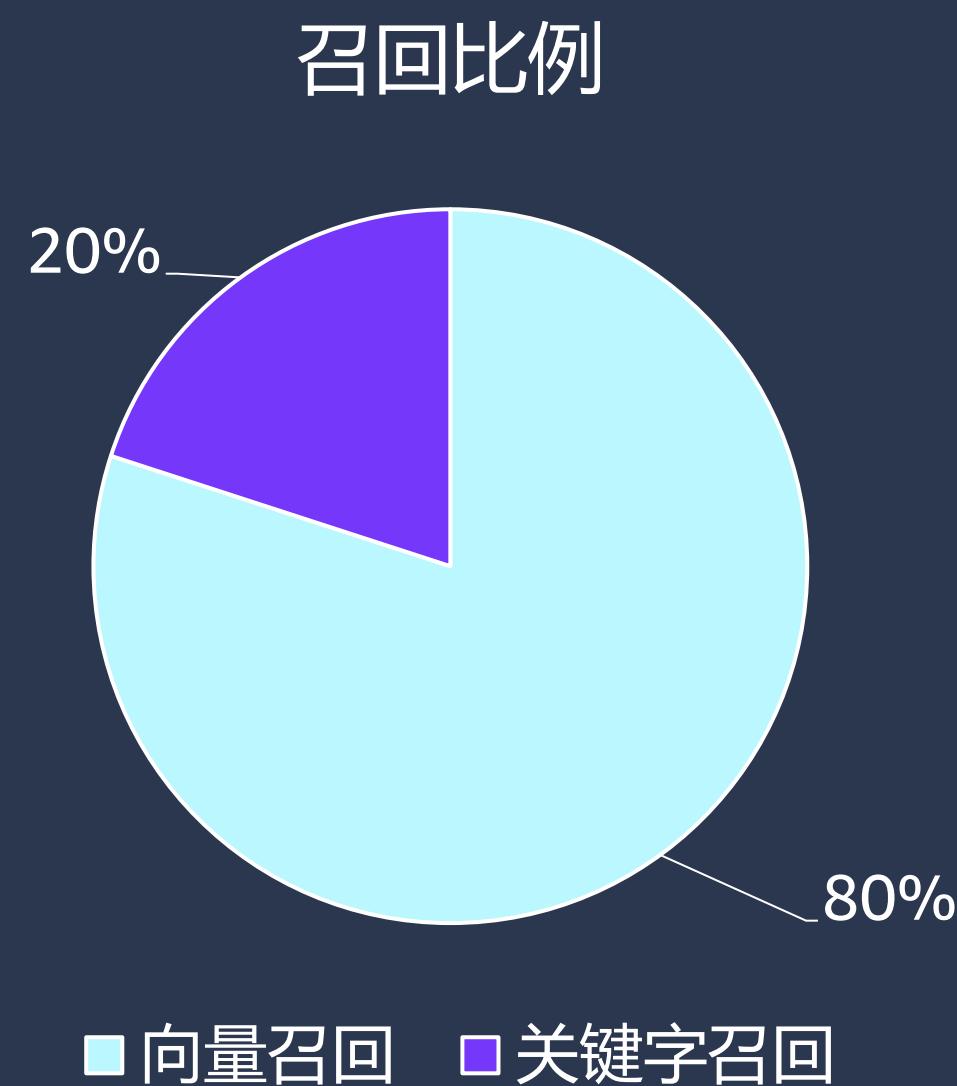
多语言能力均衡

- 具备跨语言对齐效果，可以通过中文召回类似语义的英文



# OpenSearch 在RAG上的优势

- ✓ 多路召回 (向量 + 关键字)
- ✓ Hybird召回 (Reference Doc)
- ✓ Sparse Vector Retrieval



## 口 优缺点互补:

- 向量模型不理解垂直领域专词
- 向量模型有时出现语义相似但主题不相似情况
- 可解释性弱，不易通过补丁解决bad case
- 倒排召回，缺乏语义信息，仅靠关键词匹配

# OpenSearch 在RAG上的优势

- ✓ 多路召回 (向量 + 关键字)
- ✓ Hybird召回 (Reference Doc)
- ✓ Sparse Vector Retrieval

好处：简单的内置多路召回，无需在客户端实现融合逻辑

- Step1: 创建Search pipeline

```
PUT /_search/pipeline/my-pipeline
{
  "description": "Post-processor for hybrid search",
  "phase_results_processors": [
    {
      "normalization-processor": {
        "normalization": {
          "technique": "l2" # min-max
        },
        "combination": {
          "technique": "arithmetic_mean" # geometric mean or
harmonic mean
        }
      }
    }
  ]
}
```

- Step2: 搜索时指定search pipeline

```
POST my_index/_search?search_pipeline=<pipeline>
{
  "query": {
    "hybrid": [
      {}, // First Query
      {}, // Second Query
      ... // Other Queries
    ]
  }
}
```

# OpenSearch 在RAG上的优势

- ✓ 1. 多路召回(向量 + 关键字)
- ✓ 2. Hybrid召回
- ✓ 3. Sparse Vector Retrieval(term expansion)

- 利用深度模型扩词(原理)



```
• 测试部署的SPLADE模型
POST /_plugins/_ml/models/2EsW_4sB0S9ucTLoyVvY/_predict
{
  "parameters": {
    "inputs": "Hi Altman"
  }
}

输出
{
  "inference_results": [
    {
      "output": [
        {
          "name": "response",
          "dataAsMap": {
            "response": [
              {
                "e": 0.1419215202331543,
                "he": 0.33063653111457825,
                "his": 0.424188494682312,
                "she": 0.10910777002573013,
                "him": 0.05982781946659088,
                "who": 0.47575441002845764,
                "american": 0.011252160184085369,
                ...
              ]
            }
          ]
        }
      ]
    }
  ],
  "status_code": 200
}
```

- 性能评估

Data Set	BM25		Dense (with TAS-B model)		Dense + BM25		Neural Sparse Bi-encoder		Neural Sparse Doc-only	
	NDCG@10	Rank	NDCG@10	Rank	NDCG@10	Rank	NDCG@10	Rank	NDCG@10	Rank
Trec Covid	0.688	4	0.481	5	0.698	3	0.771	1	0.707	2
NFCorpus	0.327	4	0.319	5	0.335	3	0.36	1	0.352	2
NQ	0.326	5	0.463	3	0.418	4	0.553	1	0.521	2
HotpotQA	0.602	4	0.579	5	0.636	3	0.697	1	0.677	2
FiQA	0.254	5	0.3	4	0.322	3	0.376	1	0.344	2
ArguAna	0.472	2	0.427	4	0.378	5	0.508	1	0.461	3
Touche	0.347	1	0.162	5	0.313	2	0.278	4	0.294	3
DBPedia	0.287	5	0.383	4	0.387	3	0.447	1	0.412	2
SCIDOCS	0.165	2	0.149	5	0.174	1	0.164	3	0.154	4
FEVER	0.649	5	0.697	4	0.77	2	0.821	1	0.743	3
Climate										
FEVER	0.186	5	0.228	3	0.251	2	0.263	1	0.202	4
SciFact	0.69	3	0.643	5	0.672	4	0.723	1	0.716	2
Quora	0.789	4	0.835	3	0.864	1	0.856	2	0.788	5
Amazon										
ESCI	0.081	3	0.071	5	0.086	2	0.077	4	0.095	1
Average	0.419	3.71	0.41	4.29	0.45	2.71	0.492	1.64	0.462	2.64

1. 是Dense Vector召回的重要补充，目前不支持多语言
2. 从NDCG@10和Latency上都超过 ELSER of Elasticsearch

# THANKS

---

软件正在重新定义世界  
Software Is Redefining The World