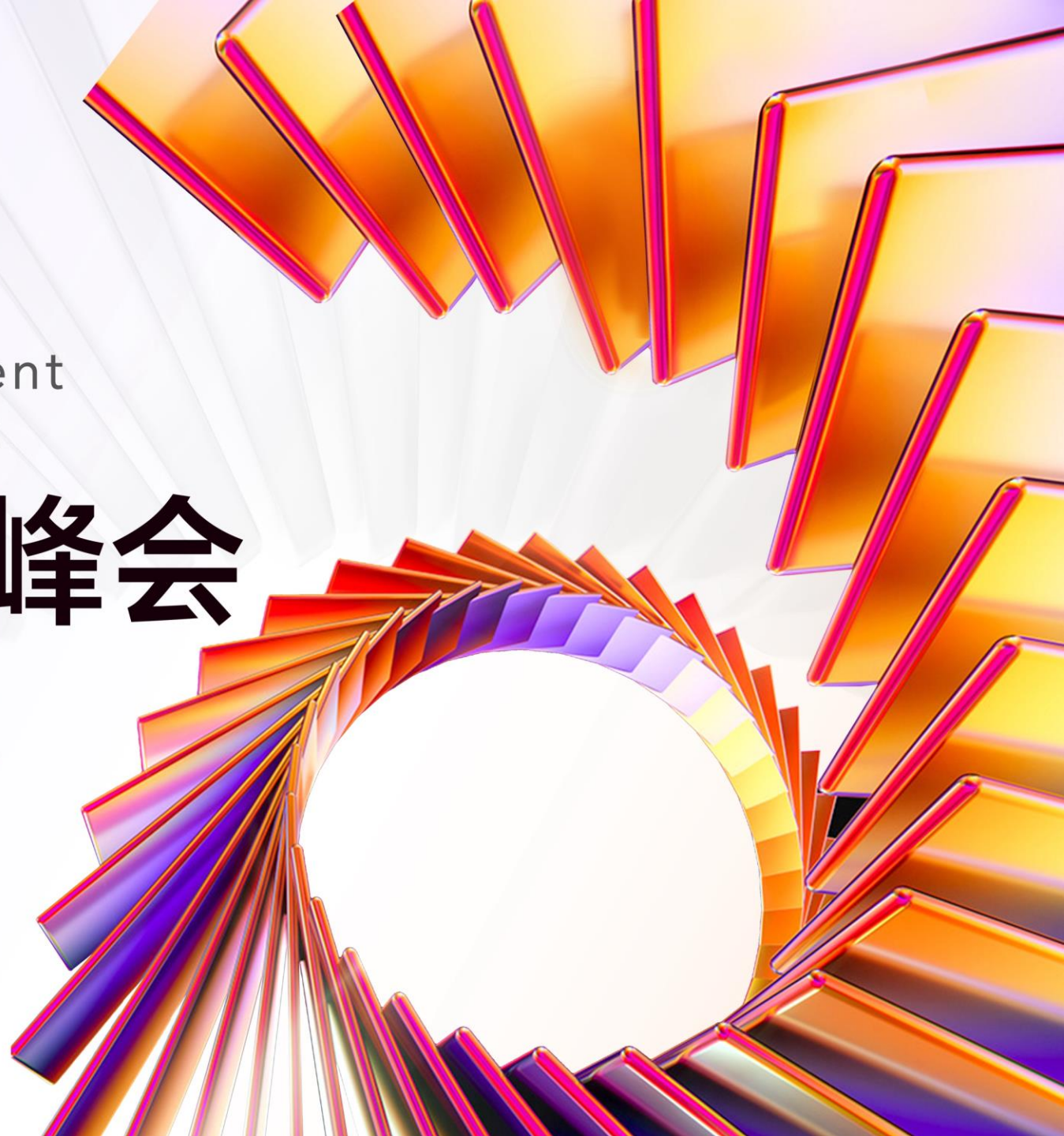
AiDD 7th 2025

第7届 AI+ Development Digital Summit AI+研发数字峰会

拥抱AI 重塑研发

8月8-9日 | 北京站



AiDD 8th 2025

第8届AI+研发数字峰会

拥抱AI 重塑研发 AI+ Development Digital Summit

下一站预告 11/14-15 | 深圳站 12/19-20 | 上海站

查看会议详情

深圳站论坛设置

智能装备与机器人

超越“编程 Copilot”

下一代知识工程

智能网联与汽车智能化

AI 测试工具开发与应用

AI 基础设施和运维

数据智能及其行业应用

可信 AI 安全工程

大模型和 AI 应用评测

多 Agent 协同框架

从智能测试到自主测试

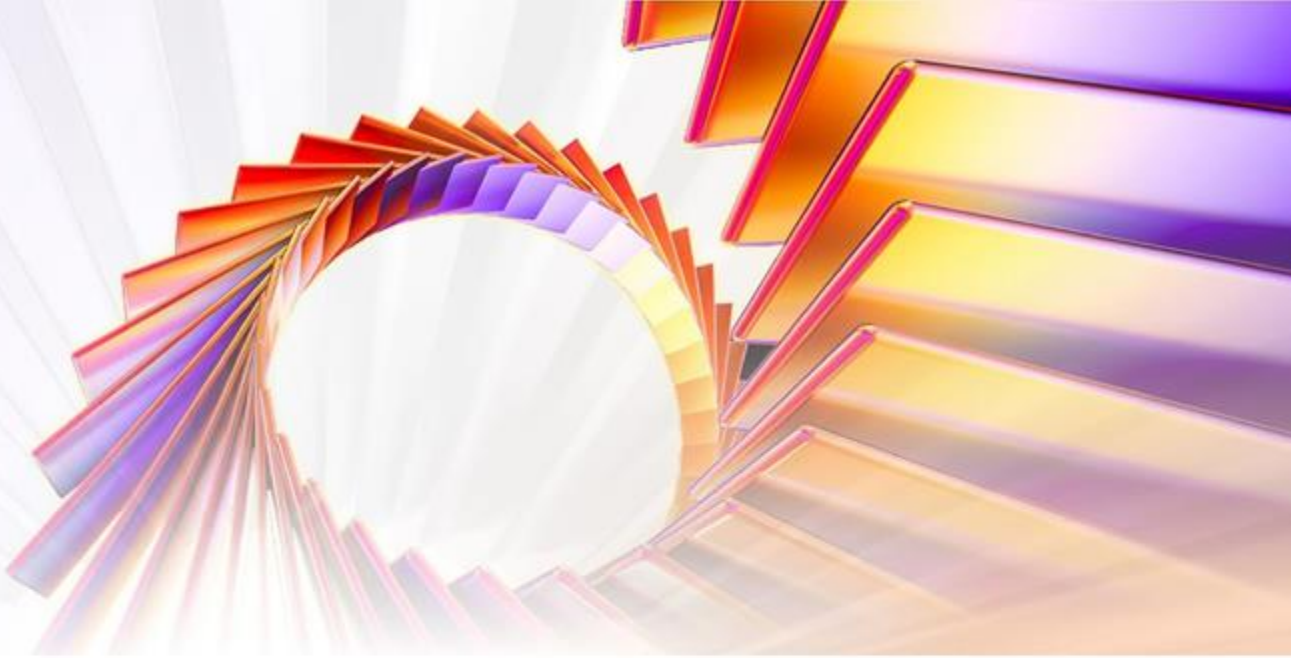
大模型推理优化

多模态 LLM 训练与应用

智能化 DevOps 流水线

上下文工程

AiDD 7th 2025 | 8月8-9日 | 北京站
第7届 AI+ Development Digital Summit
AI+研发数字峰会
拥抱AI 重塑研发



Elasticsearch-AI 驱动的搜索引擎

刘晓国 | Elastic

刘晓国

Elastic 社区首席布道师

新加坡国立大学硕士，西北工业大学本硕。曾就职于新加坡科技，康柏电脑，通用汽车，爱立信，诺基亚，Linaro 非营利组织 (Linux for ARM)，Ubuntu，Vantiq 等企业。从事过电脑设计，汽车电子，计算机操作系统，通信，云实时事件处理等行业。从爱立信开始，诺基亚，Ubuntu 到现在的 Elastic 从事社区工作有将近 20 年的经历。喜欢分享自己所学到的知识。帮助别人就是帮助自己。希望和大家一起分享及学习。

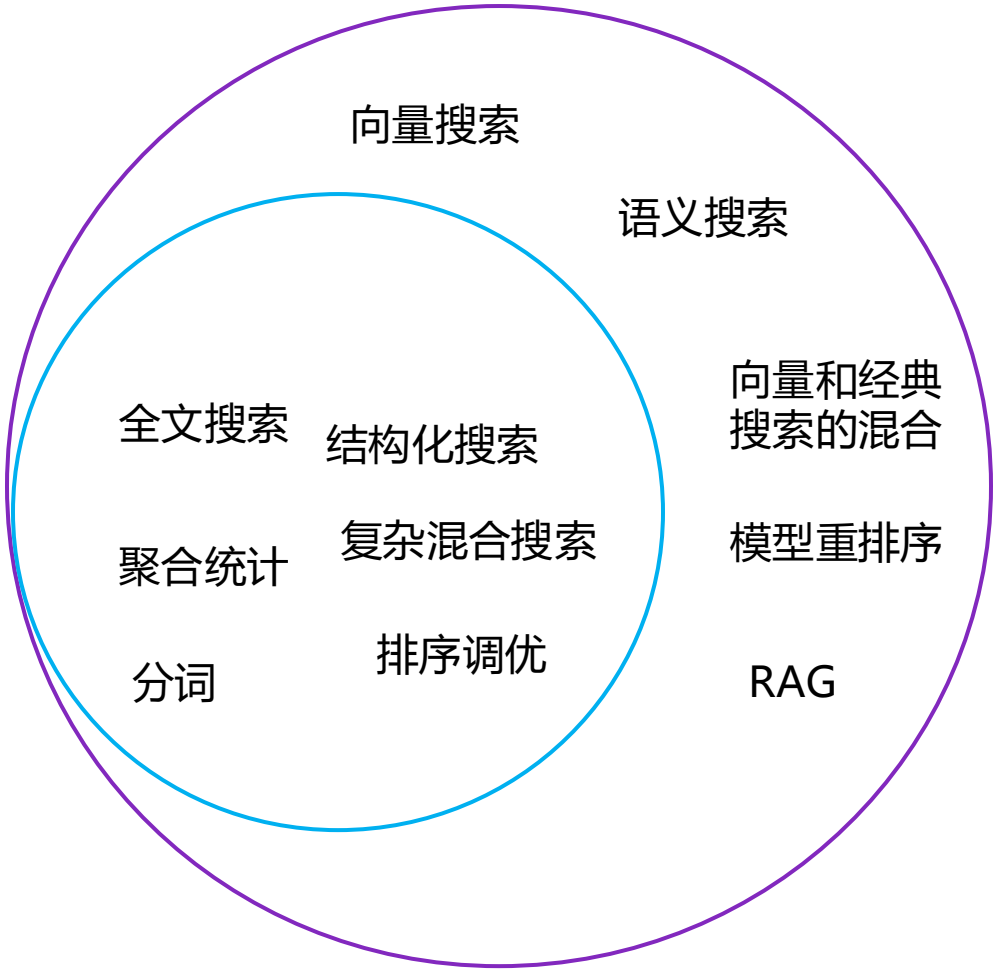
目录
CONTENTS

PART 01

智能时代的搜索需求

AI 时代对搜索提出了新要求

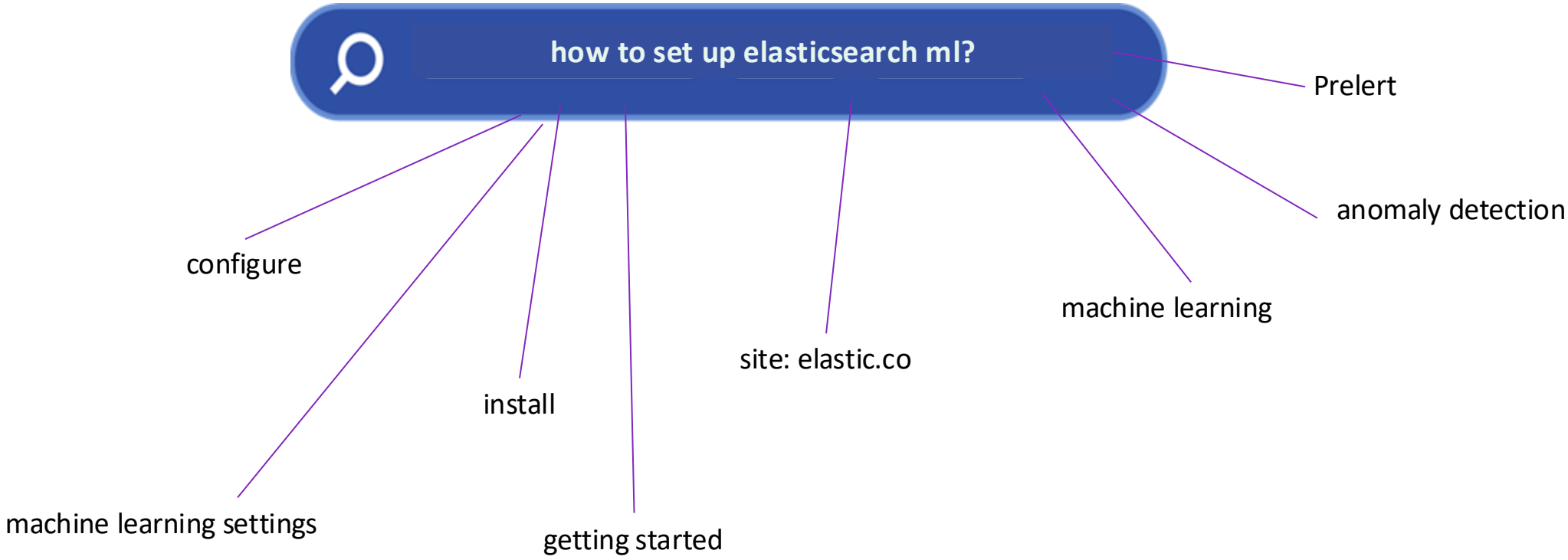
过去用户需求

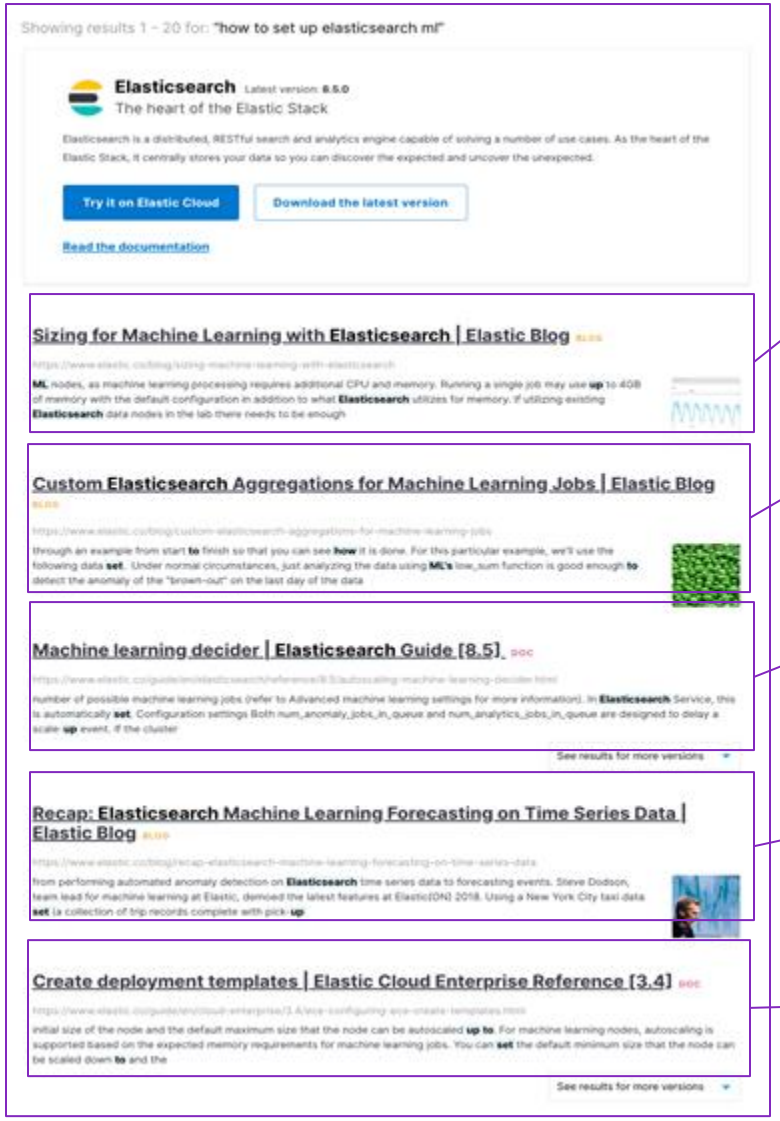


现在用户需求

语义搜索

根据搜索查询的意图和上下文含义检索结果，而不仅仅是关键字





Sizing for Machine Learning with Elasticsearch | Elastic Blog BLOG

https://www.elastic.co/blog/sizing-machine-learning-with-elasticsearch

ML nodes, as machine learning processing requires additional CPU and memory. Running a single job may use **up** to 4GB of memory with the default configuration in addition to what **Elasticsearch** utilizes for memory. If utilizing existing **Elasticsearch** data nodes in the lab there needs to be enough

Custom Elasticsearch Aggregations for Machine Learning Jobs | Elastic Blog

BLOG

https://www.elastic.co/blog/custom-elasticsearch-aggregations-for-machine-learning-jobs

through an example from start **to** finish so that you can see **how** it is done. For this particular example, we'll use the following data **set**. Under normal circumstances, just analyzing the data using **ML's** low_sum function is good enough **to** detect the anomaly of the "brown-out" on the last day of the data

Machine learning decider | Elasticsearch Guide [8.5] DOC

https://www.elastic.co/guide/en/elasticsearch/reference/8.5/autoscaling-machine-learning-decider.html

number of possible machine learning jobs (refer to Advanced machine learning settings for more information). In **Elasticsearch** Service, this is automatically **set**. Configuration settings Both num_anomaly_jobs_in_queue and num_analytics_jobs_in_queue are designed to delay a scale-**up** event. If the cluster

Recap: Elasticsearch Machine Learning Forecasting on Time Series Data | Elastic Blog BLOG

https://www.elastic.co/blog/recap-elasticsearch-machine-learning-forecasting-on-time-series-data

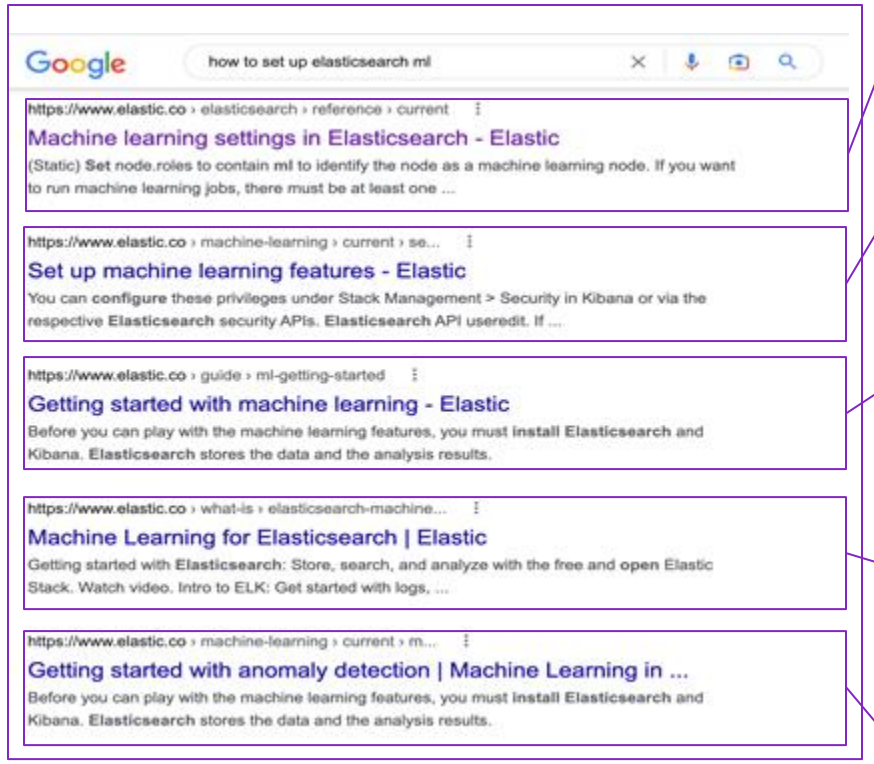
from performing automated anomaly detection on **Elasticsearch** time series data to forecasting events. Steve Dodson, team lead for machine learning at Elastic, demoed the latest features at Elastic{ON} 2018. Using a New York City taxi data **set** (a collection of trip records complete with pick-**up**

Create deployment templates | Elastic Cloud Enterprise Reference [3.4] DOC

https://www.elastic.co/guide/en/cloud-enterprise/3.4/ece-configuring-ece-create-templates.html

initial size of the node and the default maximum size that the node can be autoscaled **up to**. For machine learning nodes, autoscaling is supported based on the expected memory requirements for machine learning jobs. You can **set** the default minimum size that the node can be scaled down **to** and the

词汇搜索结果　　how to set up elasticsearch ml?



语义搜索结果

https://www.elastic.co › elasticsearch › reference › current

Machine learning settings in Elasticsearch - Elastic

(Static) **Set** node.roles to contain **ml** to identify the node as a machine learning node. If you want to run machine learning jobs, there must be at least one ...

https://www.elastic.co › machine-learning › current › se...

Set up machine learning features - Elastic

You can **configure** these privileges under Stack Management > Security in Kibana or via the respective **Elasticsearch** security APIs. **Elasticsearch** API useredit. If ...

https://www.elastic.co › guide › ml-getting-started

Getting started with machine learning - Elastic

Before you can play with the machine learning features, you must **install Elasticsearch** and Kibana. **Elasticsearch** stores the data and the analysis results.

https://www.elastic.co › what-is › elasticsearch-machine...

Machine Learning for Elasticsearch | Elastic

Getting started with **Elasticsearch**: Store, search, and analyze with the free and **open** Elastic Stack. Watch video. Intro to ELK: Get started with logs, ...

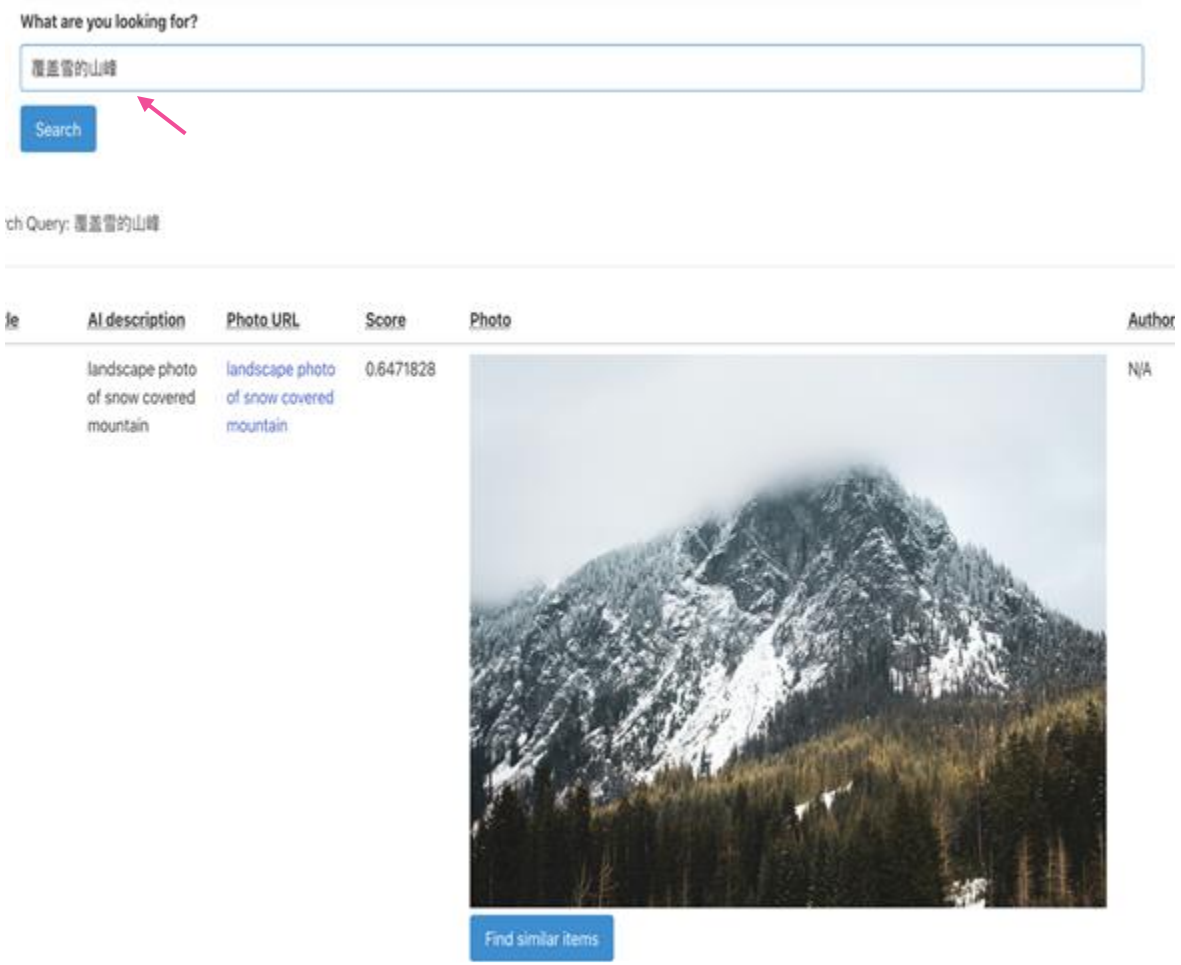
https://www.elastic.co › machine-learning › current › m...

Getting started with anomaly detection | Machine Learning in ...

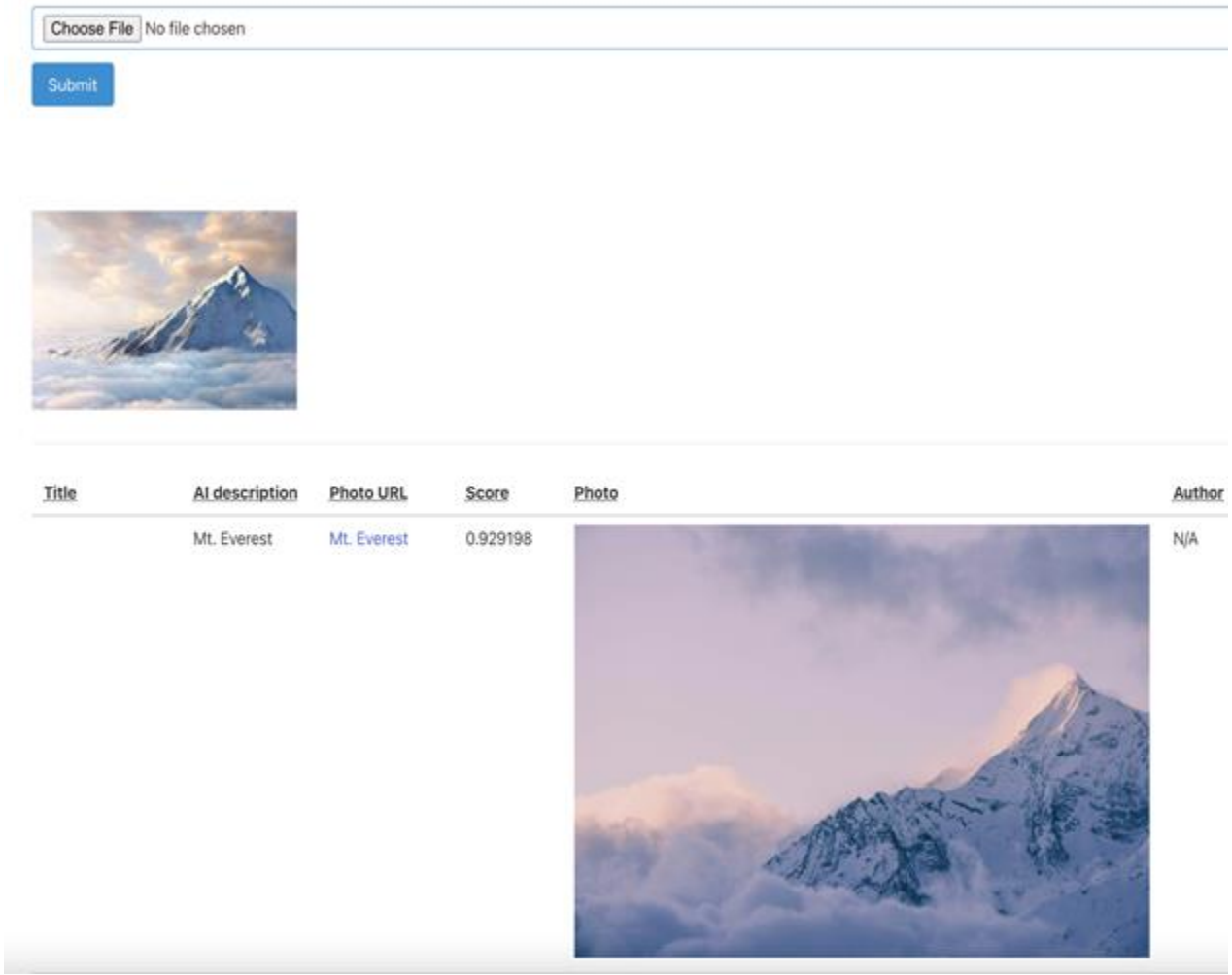
Before you can play with the machine learning features, you must **install Elasticsearch** and Kibana. **Elasticsearch** stores the data and the analysis results.

how to set up elasticsearch ml?

通过文字搜索找到图片：**覆盖雪的山峰**



通过图像比较找到相似的图片



[如何在 Elastic 中实现图片相似度搜](#)

AiDD 7th 2025

PART 02

Elasticsearch 向量搜索及最新进展

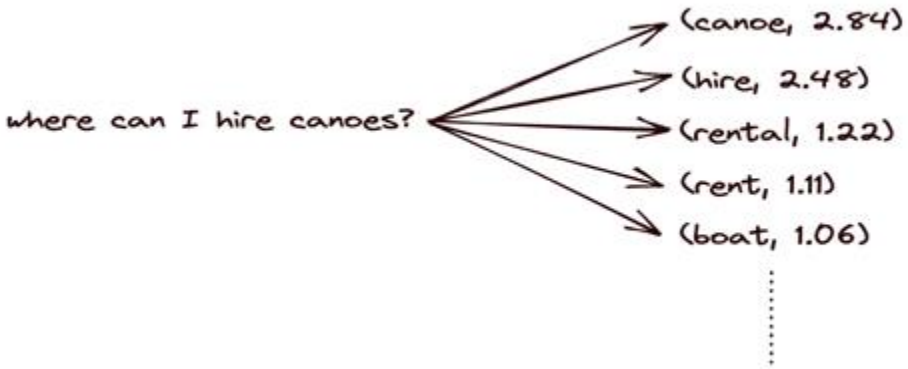
有两种向量模型

DENSE Vector
一长串数字，每个维度一个

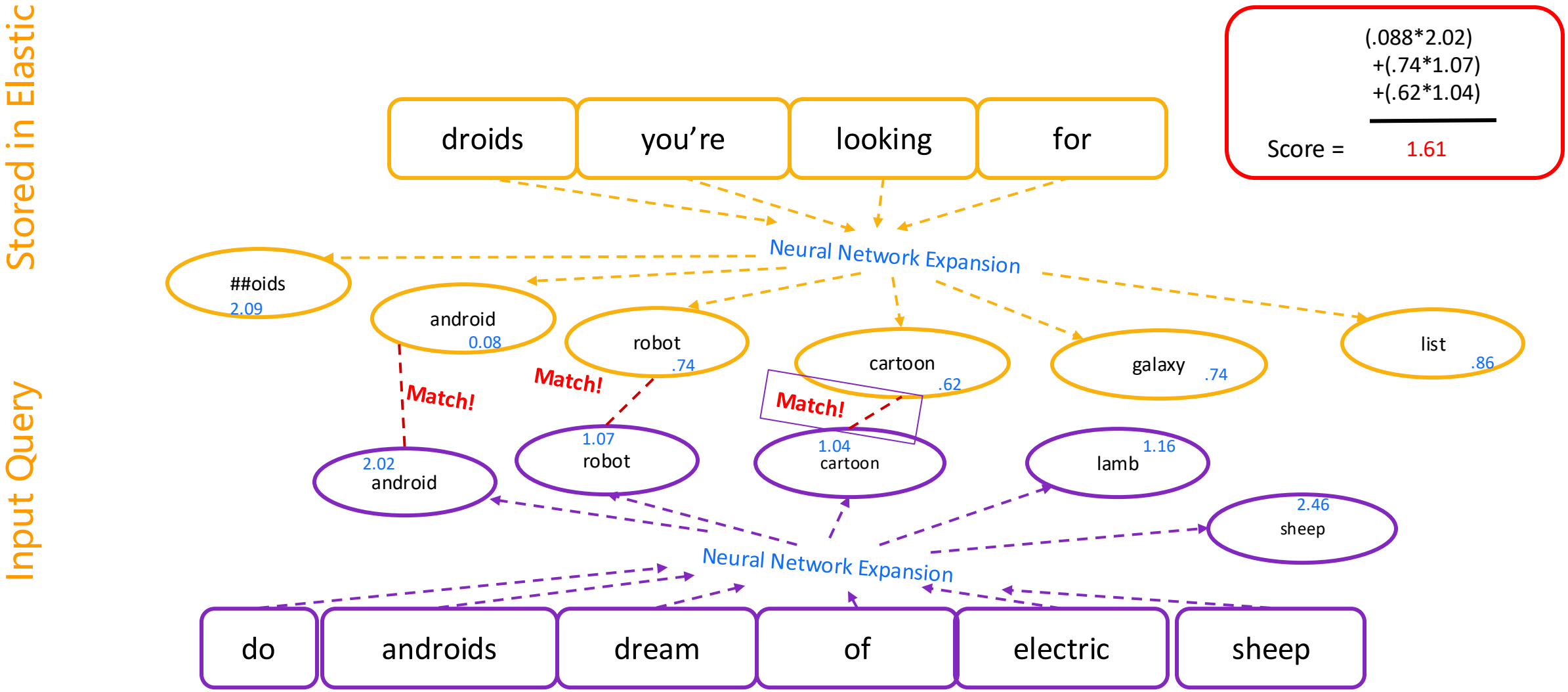
- 在数据集上进行训练，以获得较高的 “域内” 性能
- 低维（312, 512, 1536, ..）
- 捕捉语义
- 对于相似性和聚类有用
- 多模式支持
  - Text
  - Image
  - Audio
  - ...
- 较大的数据集占用大量内存
- 可解释性差

SPARSE Vector
Token Weighted Pairs

- 数十万至数百万的标记词汇量
- Token 加权对
  - Token : Weight
- 每个文档 - 仅存储 N 个最高权重的标记（其余为 0）
- 通过 DotProduct 实现语义搜索
- 与密集向量搜索相比，内存要求更低
-

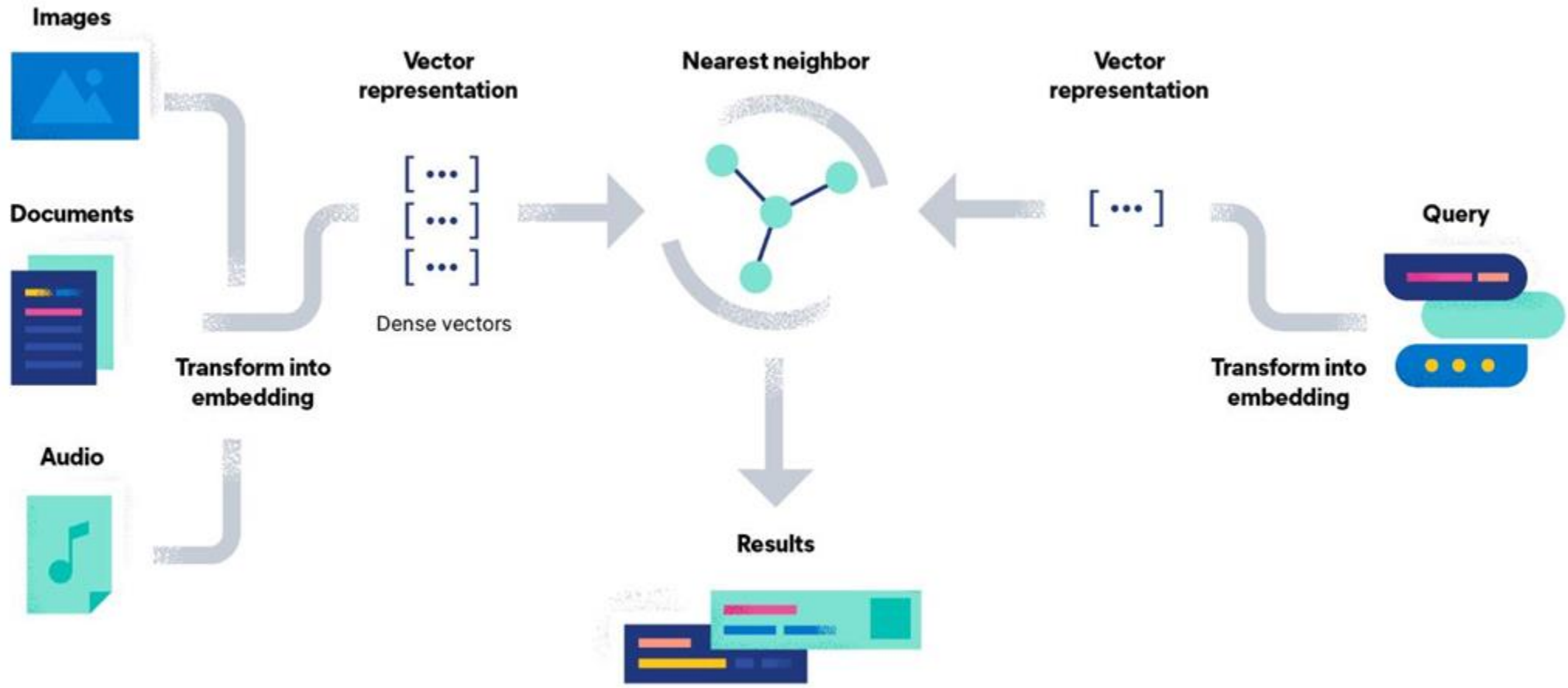


文本扩展评分


Stored in Elastic
Input Query
droids
you're
looking
for
Neural Network Expansion
##oids
2.09
android
0.08
robot
.74
cartoon
.62
galaxy
.74
list
.86
Match!
Match!
Match!
2.02
android
1.07
robot
1.04
cartoon
1.16
lamb
2.46
sheep
Neural Network Expansion
do
androids
dream
of
electric
sheep
(.088*2.02)
+(.74*1.07)
+(.62*1.04)
Score = 1.61

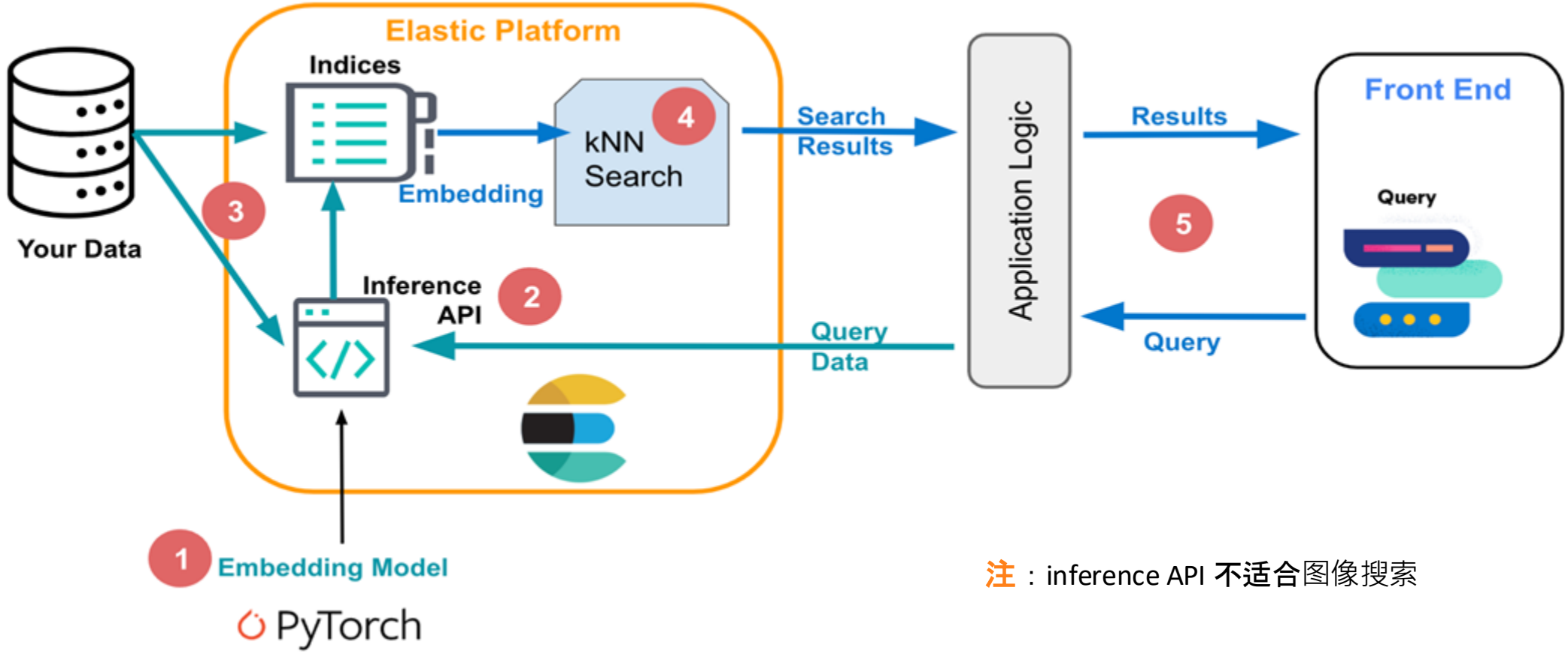
向量搜索概念架构

使用向量最近邻生成搜索排名



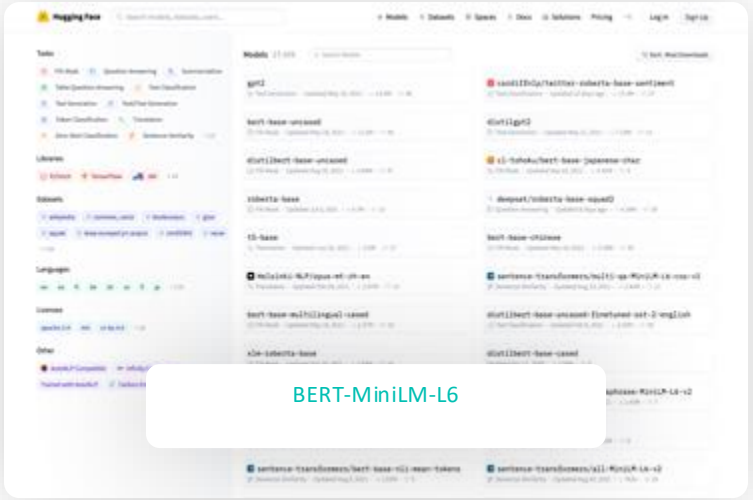
向量搜索支持的应用程序架构

了解 5 个关键组件

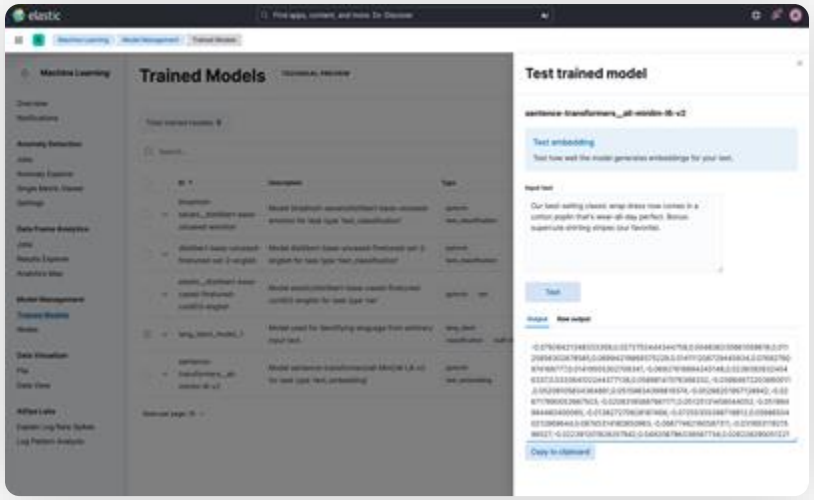


注：inference API 不适合图像搜索

步骤 1：设置机器学习模型




```
$ eland_import_hub_model
--url https://cluster_URL --hub-model-id BERT-MiniLM-L6 --
task-type text_embedding --start
```



PyTorch → eland →

选择合适的模型

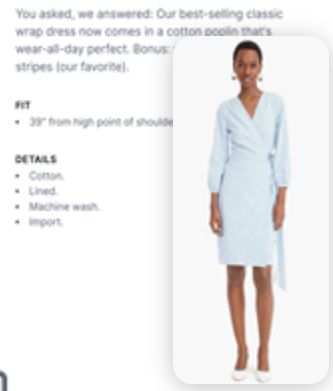
将模型加载到集群

管理模型

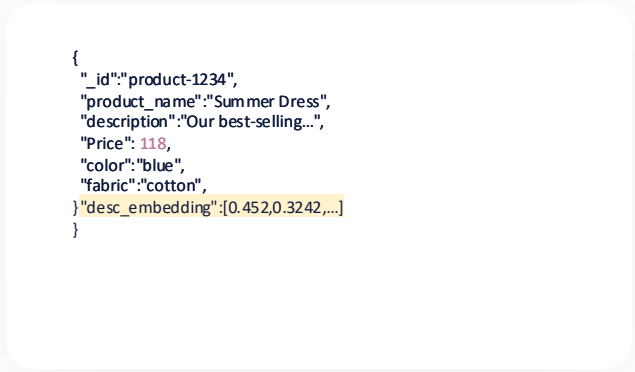
步骤二: 数据摄取和嵌入生成

POST /_doc

Standard field indexing for non-vector types

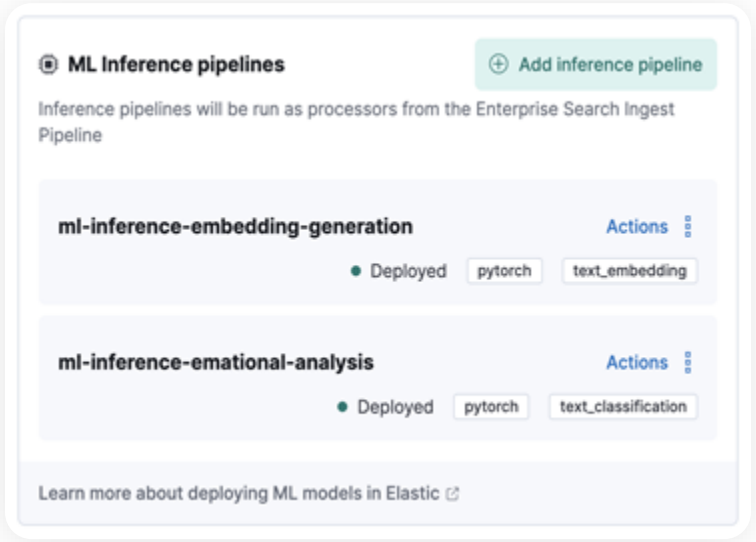


Source data



POST /_doc

Encoding via Inference Processor



步骤三： 发出向量查询

a 查询被提交给搜索驱动的应用程序

b 生成查询嵌入

```
POST /_ml/trained_models/my-model/_infer

{
  "docs": {
    "description": "summer clothes"
  }
}
```

Transformer model

PyTorch

c 使用 _search 端点发出查询，带有 kNN 子句，使用先前生成的嵌入

```
GET product-catalog/_search

{
  "query": {
    "match": {
      "description": {
        "query": "summer clothes",
        "boost": 0.9
      }
    }
  },
  "knn": {
    "field": "desc_embbeding",
    "query_vector": [0.123, 0.244,...],
    "k": 5,
    "num_candidates": 50,
    "boost": 0.1,
    "filter": {
      "term": {
        "department": "women"
      }
    }
  },
  "size": 10
}
```

需要计算后填入

把两个步骤合为一个 - 8.7 +

Transformer model

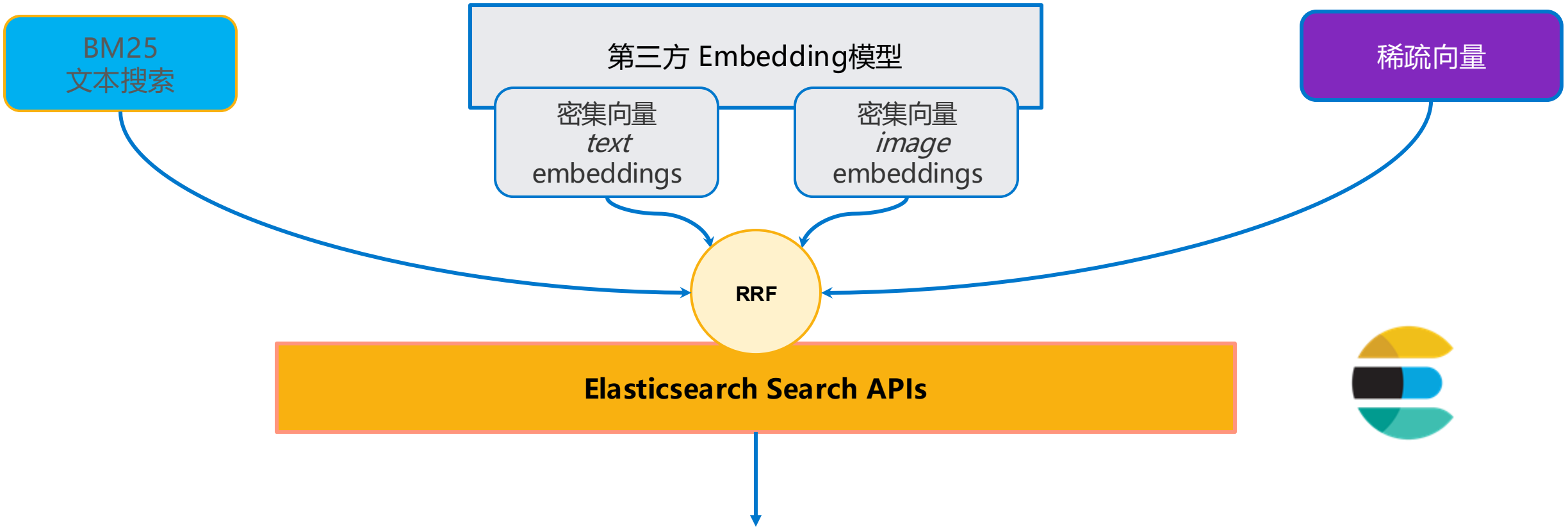
PyTorch

```
GET product-catalog/_search

{
  "knn": {
    "field": "desc_embbeding",
    "k": 5,
    "num_candidates": 50,
    "query_vector_builder": {
                    "text_embedding": {
                      "model_text": "summer clothes",
                "model_id": <text-embedding-model>
    },

    "filter": {
      "term": {
        "department": "women"
      }
    }
  },
  "size": 10
}
```

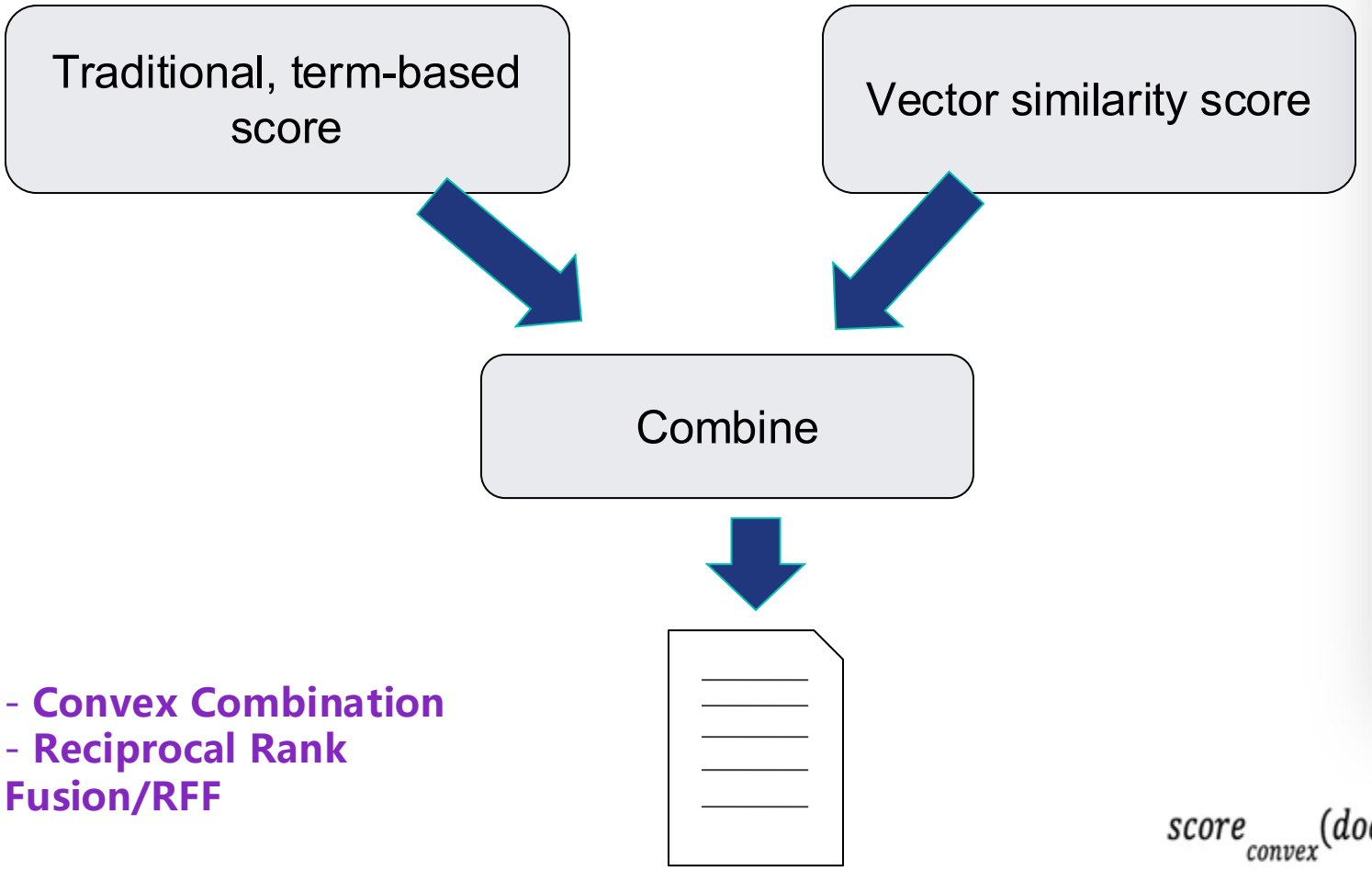
Elasticsearch 混合搜索



强大的语义和混合搜索

elastic

混合评分让你两全其美



- Convex Combination
- Reciprocal Rank Fusion/RFF

```
GET product-catalog/_search
{
 "query": {
  "match": {
   "description": {
    "query": "summer clothes",
    "boost": 0.9
   }
  }
 },
 "knn": {
  "field": "desc_embbeding",
  "query_vector": [0.123, 0.244,...],
  "k": 5,
  "num_candidates": 50,
  "boost": 0.1,
  "filter": {
   "term": {
    "department": "women"
   }
  }
 },
 "size": 10
}
```

pre-filtering

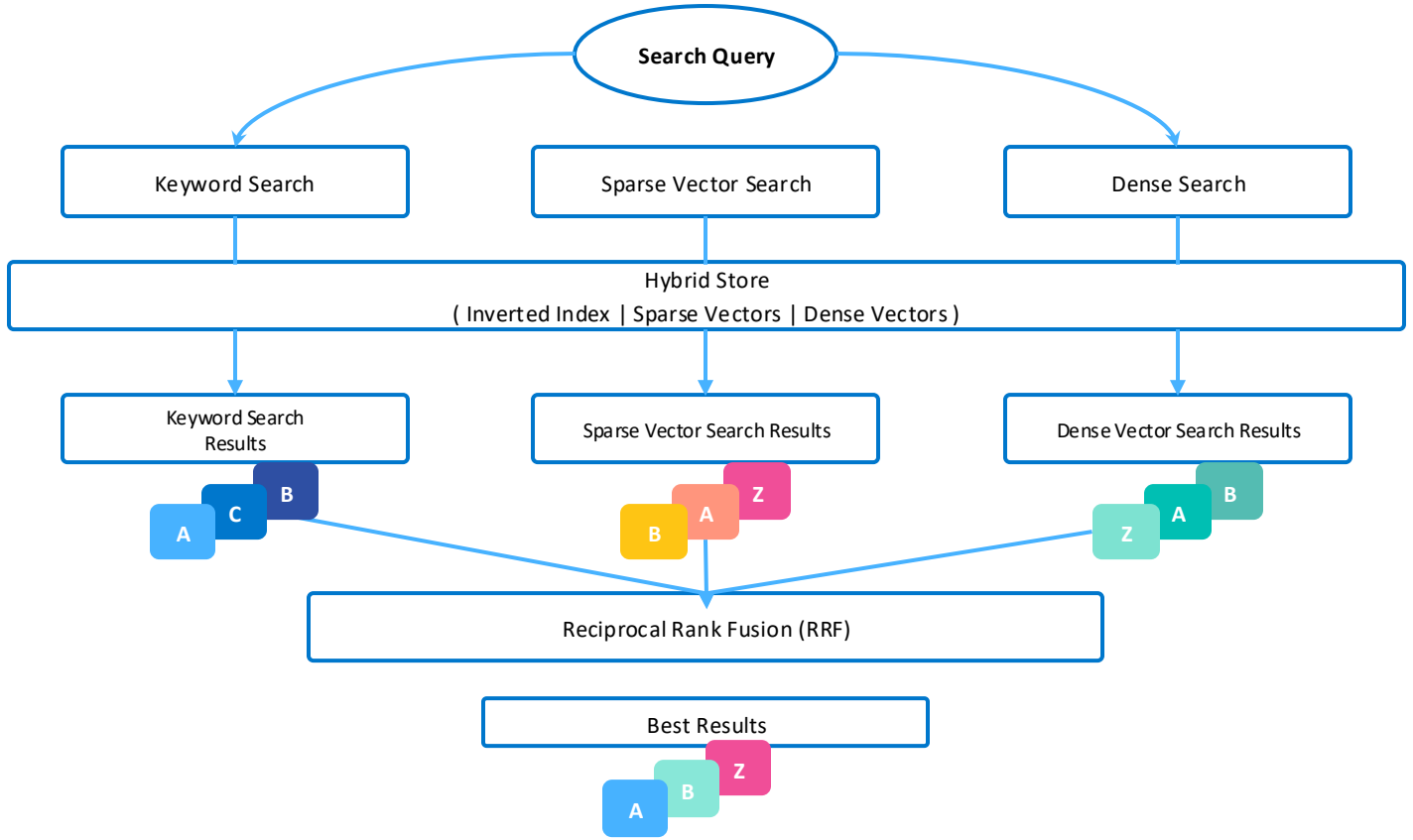
$$score_{convex}(doc) = \alpha \times score_{lex}(doc) + \beta \times score_{sem}(doc)$$

Reciprocal Rank Fusion (RRF) 算分示例

BM25 相关性排名		密集向量相关性排名		RRF结果 k=0
A	1	B	1	B : 1/2 + 1/1 = 1.5
B	2	C	2	A : 1/1 + 1/3 = 1.3
C	3	A	3	C: 1/3 + 1/2 = 0.83

$$RRFscore(d \in D) = \sum_{r \in R} \frac{1}{k + r(d)}$$

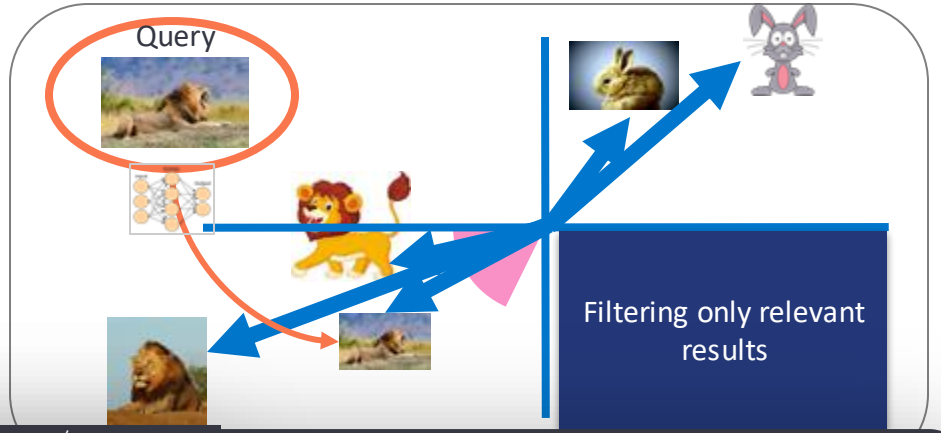
D – 文档集合
R – 排名序数的集合 1..|D|
K – 通常默认设置为60



$$score_{rrf}(doc) = \frac{1}{k + rank_{lex}(doc)} + \frac{1}{k + rank_{sem}(doc)}$$

kNN 相似度阈值

只接受足够相似的文档作为 kNN 搜索的结果



驱动

- 用户请求
- 完成 facets 的故事 - 这是我们的一个关键优势

客户价值

- 避免向用户浮动误报
- 完整实用的解决方案
  - Facets - 必须要求
  - 用户讨厌误报
  - 按排名质量阈值过滤 - 实际需要

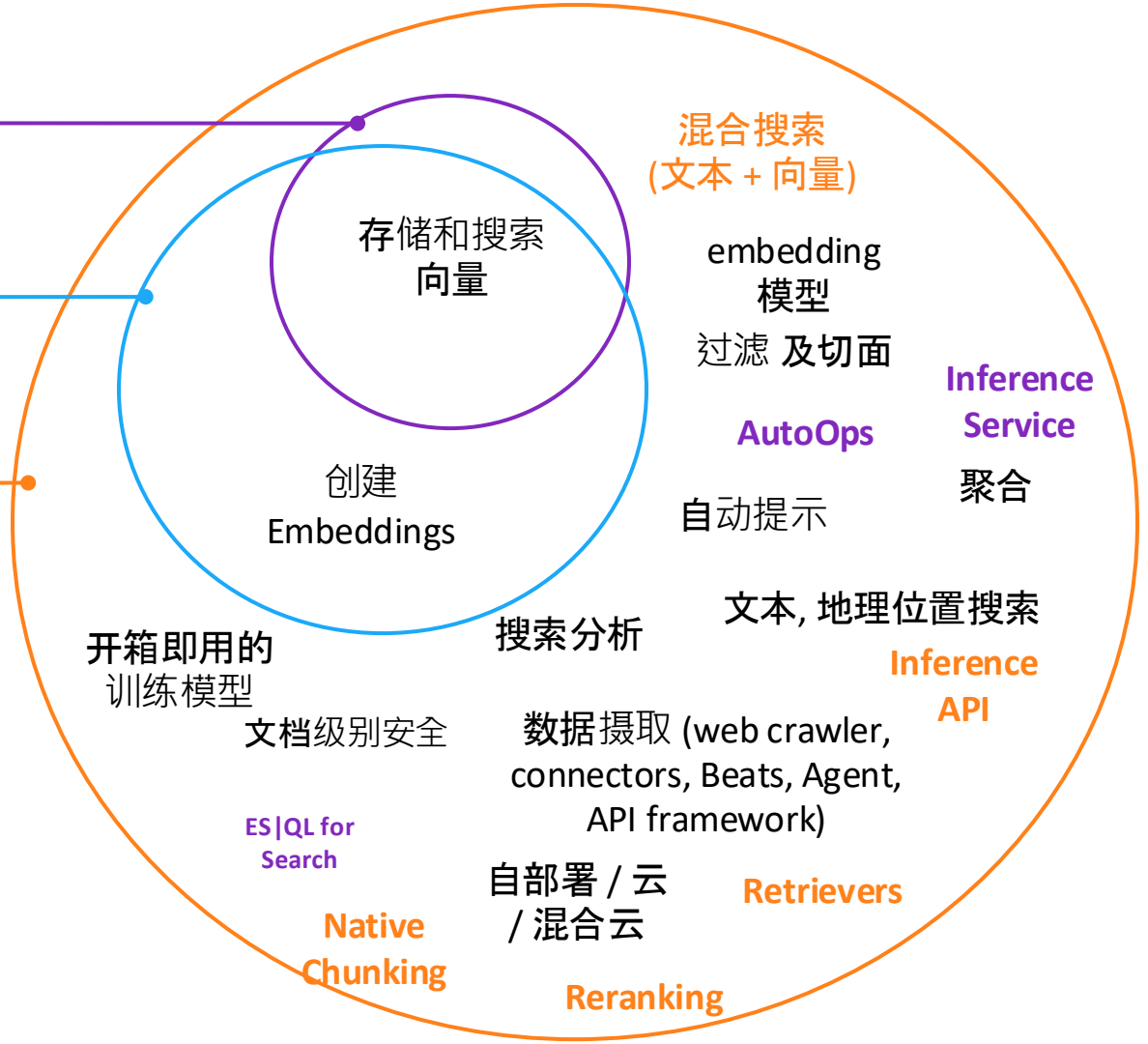
```
POST image-index/_search
{
  "knn": {
    "field": "image-vector",
    "query_vector": [1, 5, -20],
    "k": 5,
    "num_candidates": 50,
    "similarity": 36,
    "filter": {
      "term": {
        "file-type": "png"
      }
    }
  },
  "fields": ["title"],
  "_source": false
}
```

Elastic 能够提供你需要的所有功能

大多数的向量数据库

一些向量数据库

Elasticsearch

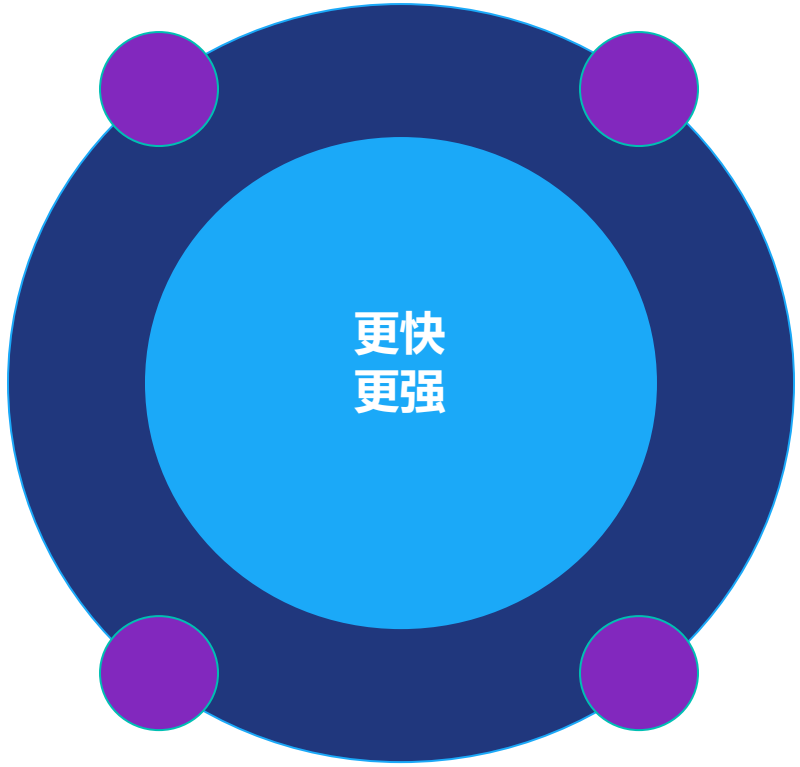


- 基于 HNSW 的索引/CAGRA
- 原生向量数据库，并非是插件
- 与现有 ES 搜索功能高度融合进行混合搜索
- 生成式人工智能应用所需的全部功能

Elasticsearch 向量引擎最新进展

硬件加速

利用 CPU 硬件指令加速向量索引和计算速度。GPU 加速/CAGRA（CUDA ANN GRAph））



增加单个查询并发

增加查询并发度，充分利用更多的计算核心

标量量化

向量有损压缩，float 到 int8、int4、bit(BBQ) 向量来平衡精度、速度和成本

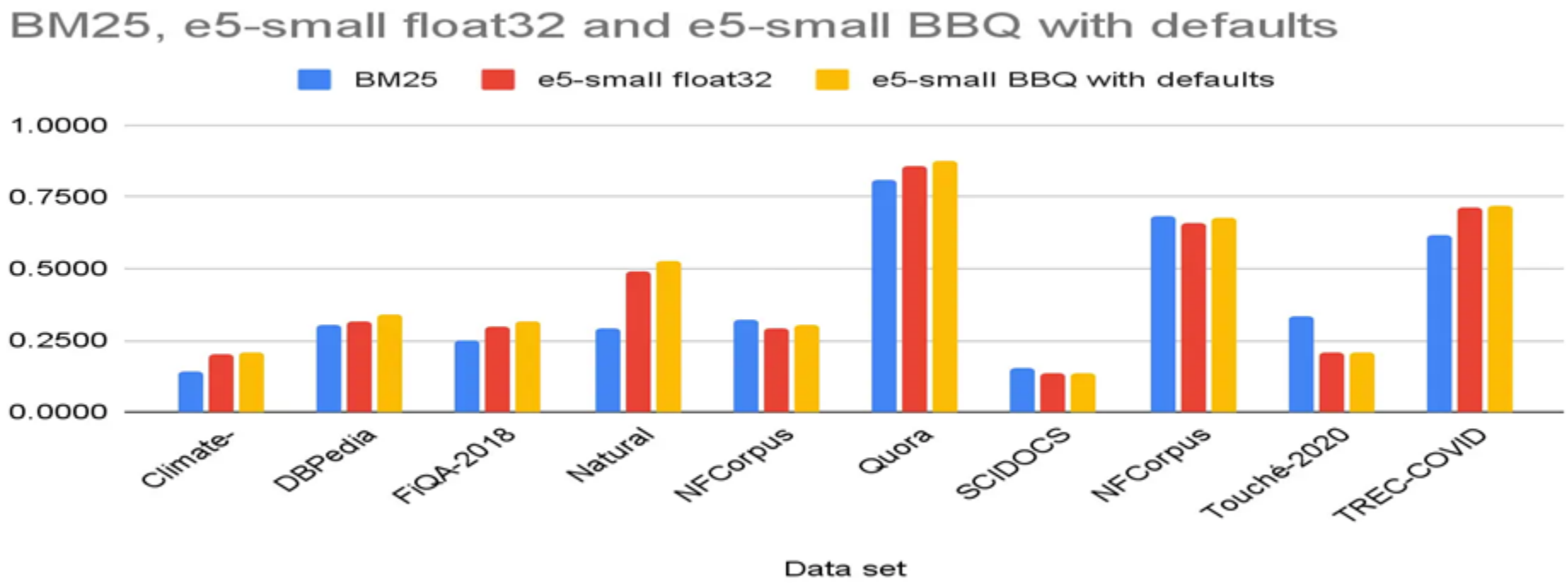
并发查询间协同

一个查询的多个并发线程间协同共享信息，提前终止一些查询线程

Elasticsearch 回顾：向量搜索创新的时间线

Better Bit Quantization - BBQ - 8.16

BBQ 是 Lucene **和** Elasticsearch **在量化方面的一次**飞跃，将 float32 维度缩减为位，**在保持高排名**质量的同时减少约 95% **的内存。**BBQ **在索引速度**（**量化**时间减少 20-30 **倍**）、查询速度（查询速度提高 2-5 **倍**）**方面**优于乘积量化 (Product Quantization - PQ) **等**传统方法，并且不会额外损失准确性。



Elasticsearch 向量引擎- 搜索并发及硬件效率改进

以前

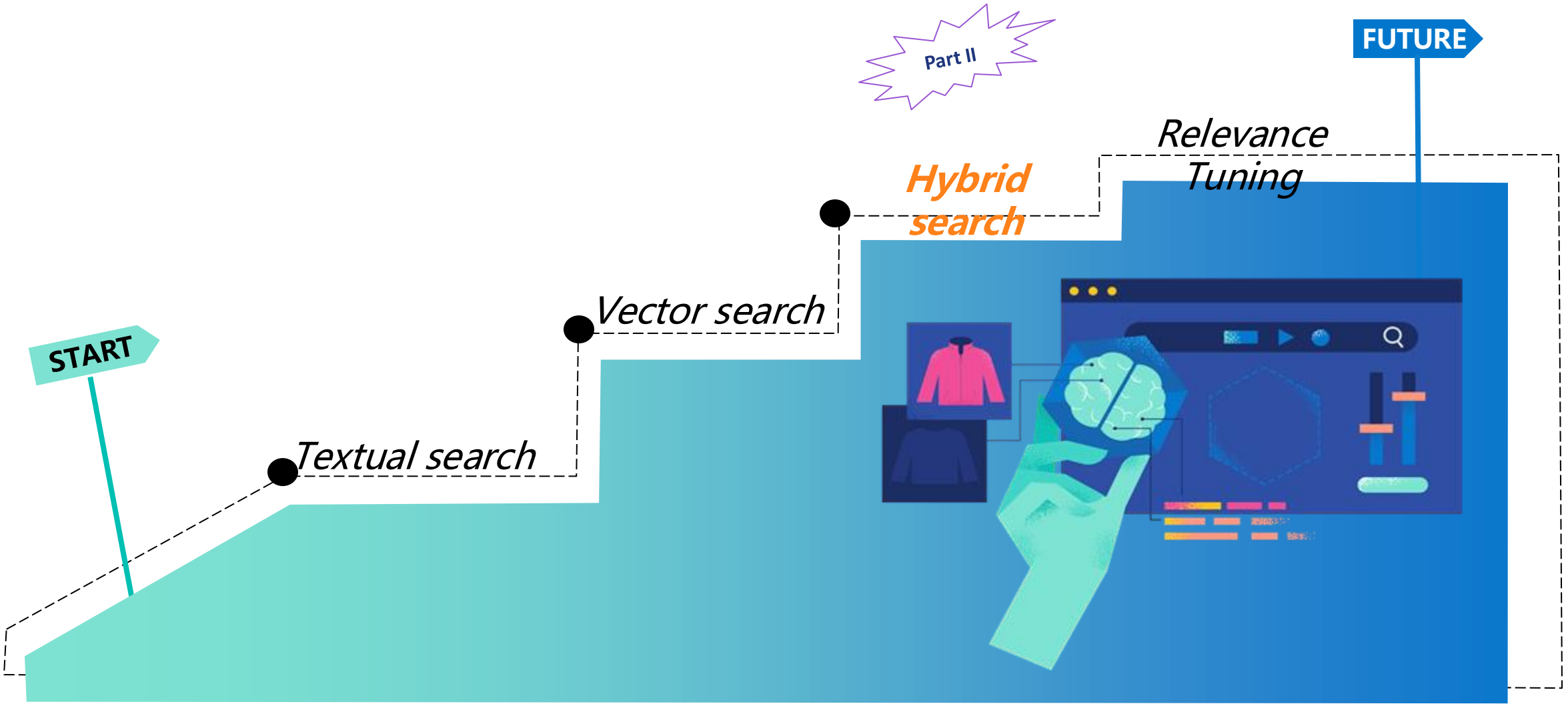
- 集群整体吞吐优先
- 限制单个查询的资源
- 每个查询每个分片一个查询线程

现在

- 每个查询每个分片中的段一个查询线程、逻辑分区
- 改进了搜索延迟
- 可以重复利用更多核心数
- 并发间协调
- IO 并发
- 稀疏索引
- 提前终止
- 快速模式

Apache Lucene 10 已发布！Lucene 硬件效率改进及其他改进

搜索发生了哪些变化 ...


Part II
FUTURE
Relevance
Tuning
Hybrid
search
Vector search
START
Textual search

搜索趋势

检索技术和查询形成

更加关注重新排序

- 系统地应用各种机器学习模型和技术
- 填补准确率与成本之间的梯度

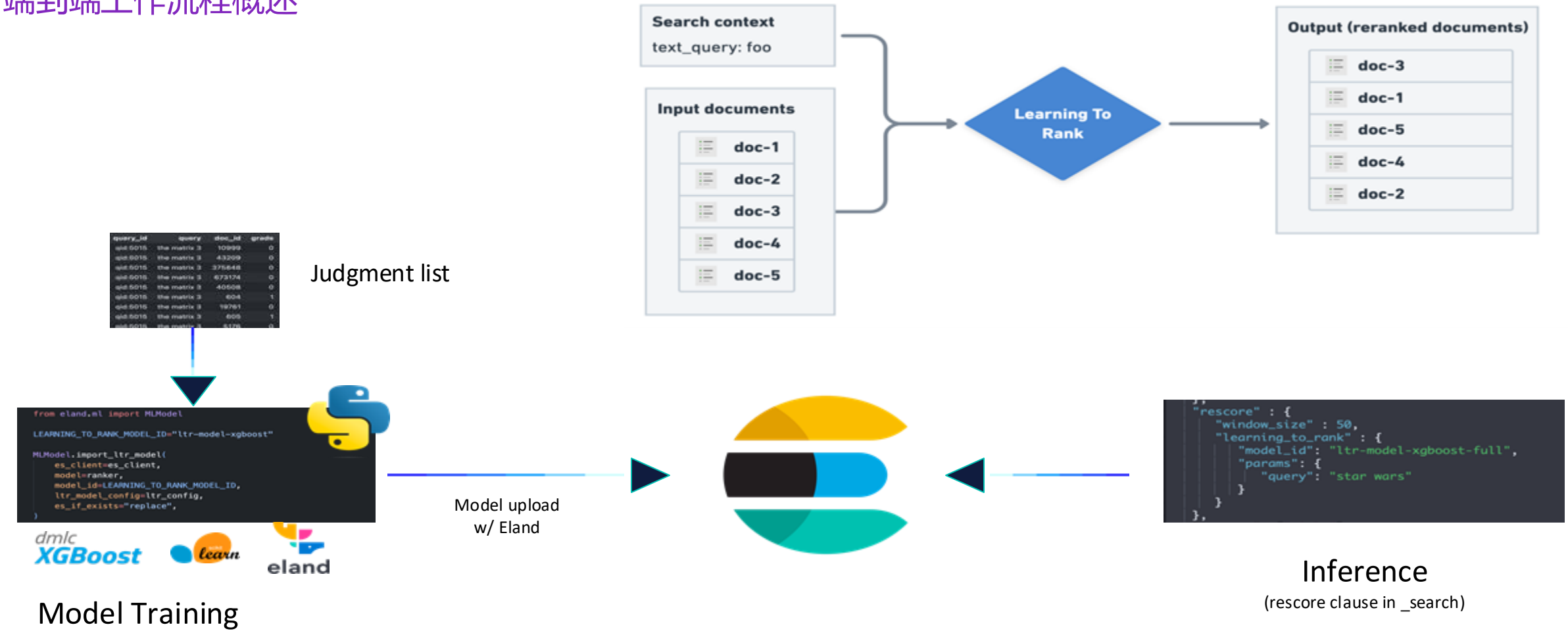
Final reranking
(10 - 100 docs)
- AI based models
- Personalization

Mid-stage rerankers
(1k - 10k docs)
- Query rescorers
- Learning To Rank

First-stage retrieval
(100k - millions of documents)
- BM25
- ANN dense retrieval
- Sparse retrieval (e.g. ELSER)

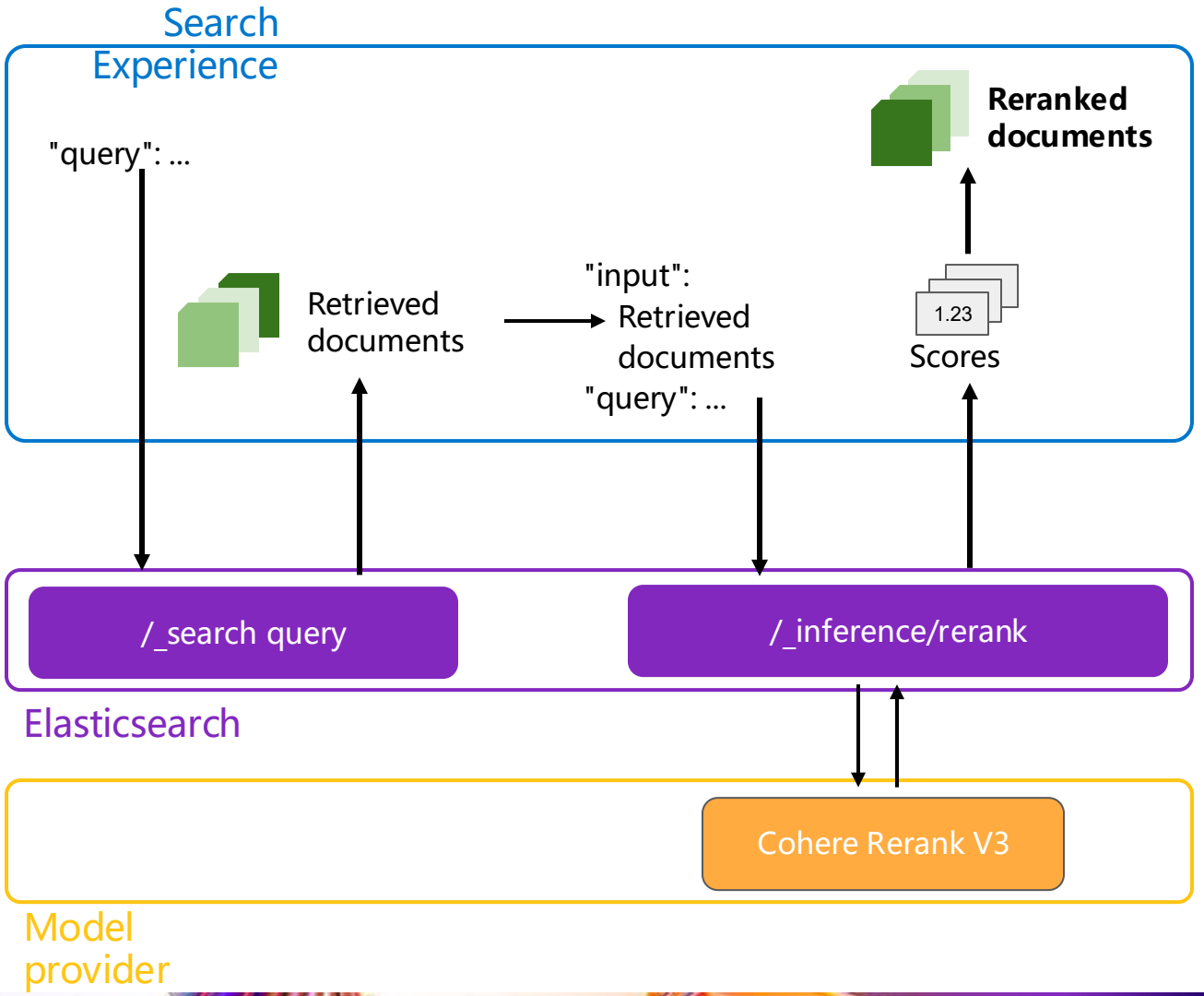
Learning-to-rank

端到端工作流程概述



介绍 Elasticsearch 中的 Learning to Tank - 学习排名

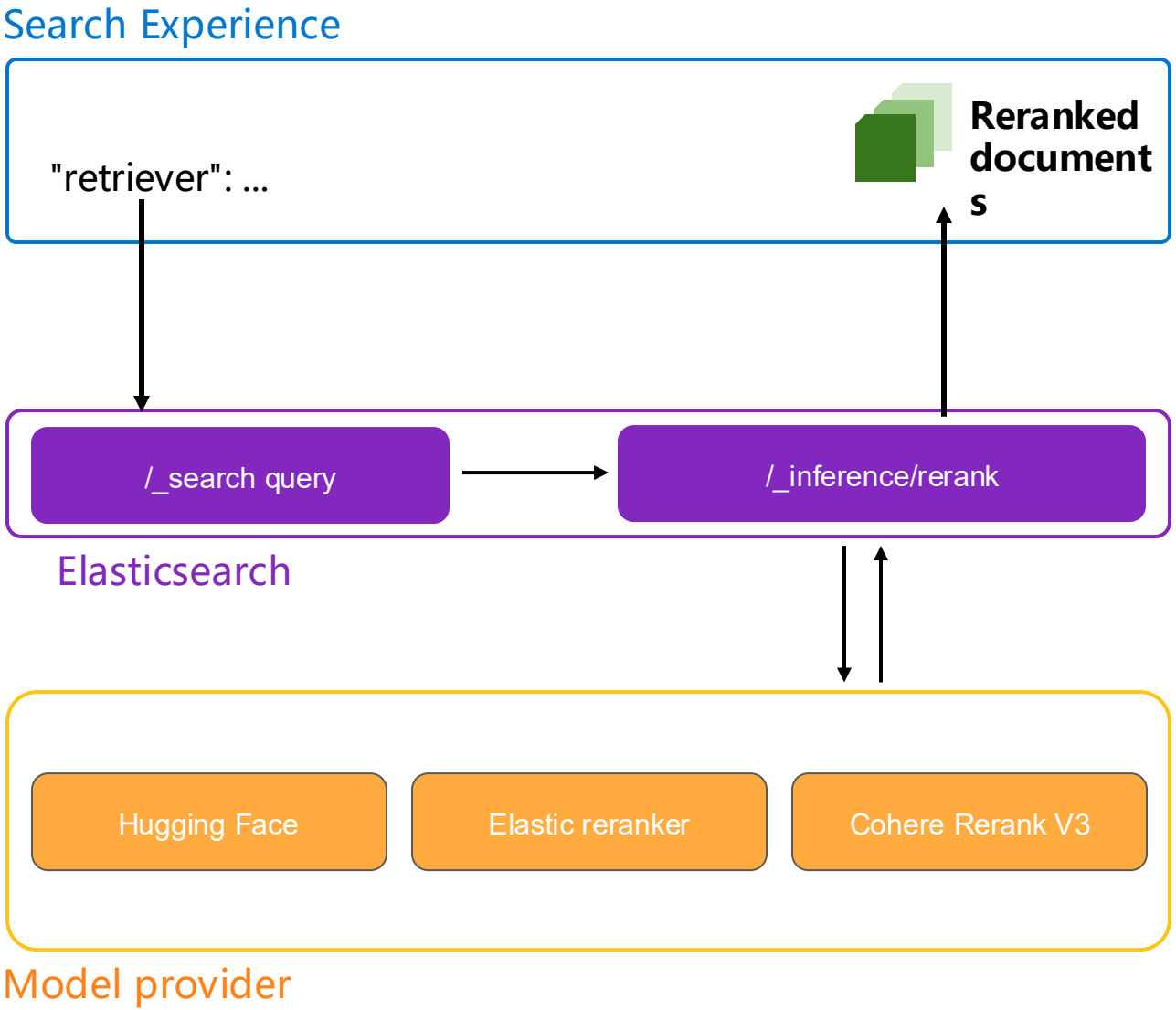
_inference/rerank




```
POST _inference/rerank/cohere_rerank
{
  "input": ["Snow Crash", "Fahrenheit 451", "1984", "Brave New World
  "query": "Snow"
}
```

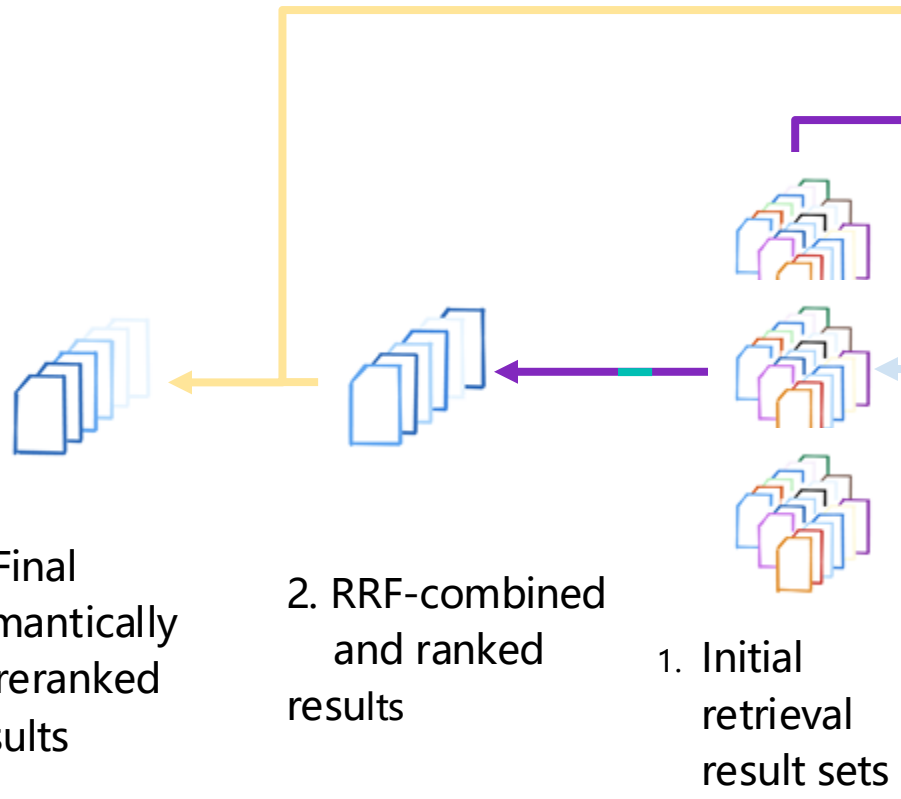
_inference/rerank

multiplied



Retrievers - 检索器

[Rescorer Retriever]



Learning to rank

```
"retriever": {
  "text_similarity_reranker": {
    // ... semantic reranking parameters
    "retriever": {
      "rrf": {
        "retrievers": [{
          "standard": {
            "query"."semantic": {
              "field": "a-semantic_text-field",
              "query": "why are retrievers fun?"
          }},{
          "knn": {
            // ... knn parameters
            // ... query_vector_builder parameters
              "model_text": "why are retrievers fun?"
          }},{
          "standard": {
            "query"."match": {
              "some-field": "why are retrievers fun?"
}}}}}}}}
```

等等 —— 我可以将我所有的私密数据向量化吗?

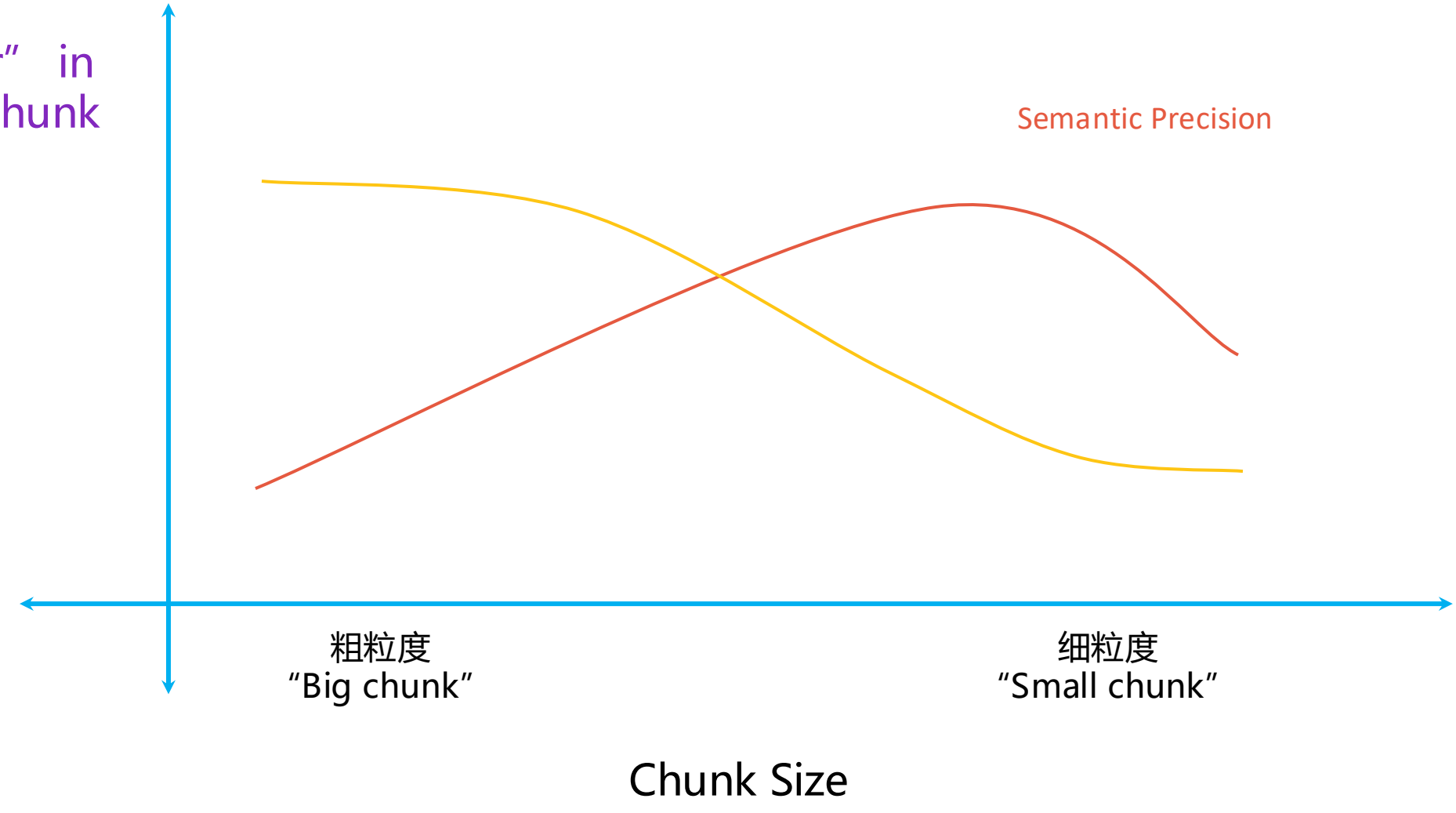
YES ... 确实需要一个分块策略

深入思考

单个向量能代表整个小说吗?

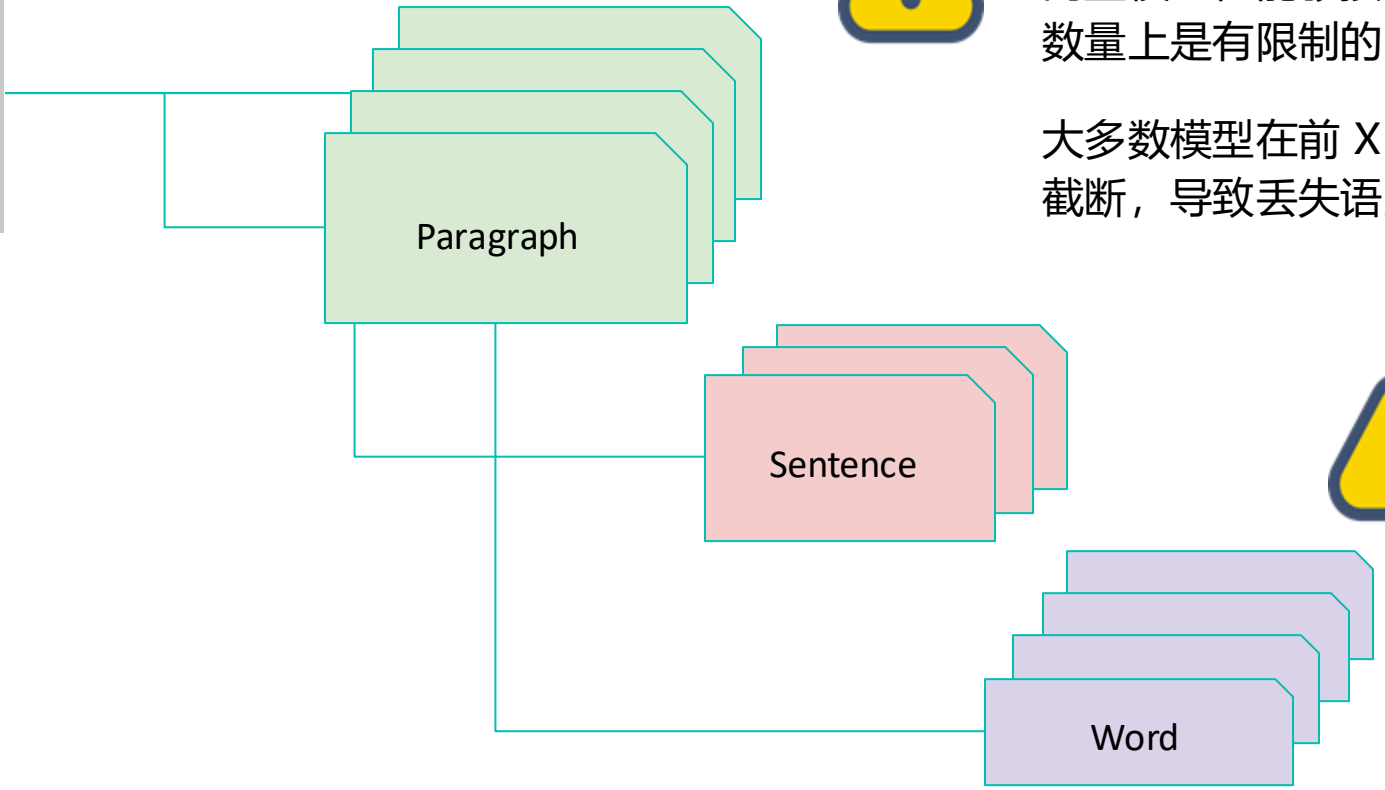
分块策略

Chance “answer” in one contiguous chunk



分块策略

几种示例



向量模型在能够接受的输入 token 数量上是有限制的。

大多数模型在前 X 个 token 之后会截断，导致丢失语义。

块越小，单个块包含回答问题所需所有信息的可能性就越小。

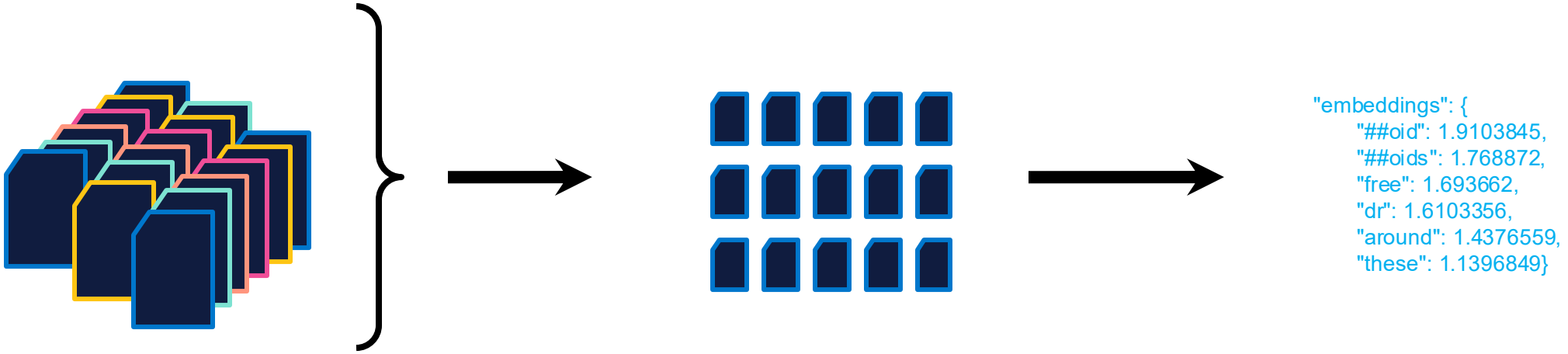
使用 semantic_text 自动在 ES 中进行分块

- 文本被分成250个单词组成的块
- 每个块包含来自前一个块的 100 个单词
- 块传递给推理 API 以进行嵌入
- 原始文本与块、块向量一起存储
- 使用 semantic 查询将自动搜索 semantic_text 字段中的嵌套块。

  - 8.16+， 默认基于句子
  - 8.16 之前，默认基于单词

Semantic Text 字段类型

存储和自动分块：简化 RAG 应用程序



Auto-chunk!

Vectorize

Elasticsearch：检索增强生成背后的重要思想

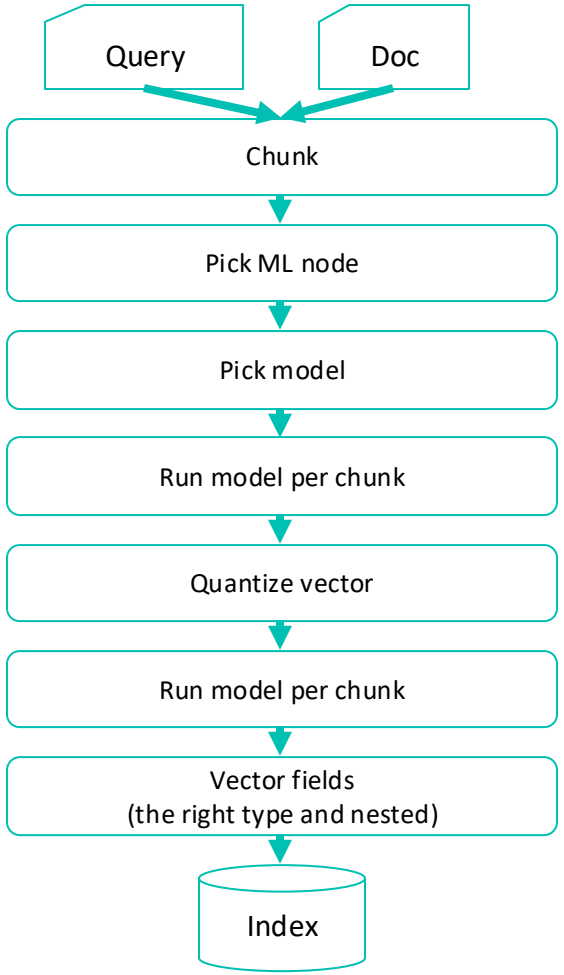
简化向量搜索使用

```
PUT _inference/sparse_embedding/my-inf-endpoint
{
 "service": "elser",
 "service_settings": {
   "num_allocations": 1,
   "num_threads": 1
 }
}
```

```
PUT test-index/_doc/doc1
{
  "my_inference_field": "my doc text"
}
```

```
PUT test-index
{
  "mappings": {
    "properties": {
      "my_inference_field": {
        "type": "semantic_text",
        "inference_id": "my-inf-endpoint"
      }
    }
  }
}
```

```
GET <index>/_search
{
    "query": {
        "semantic": {
            "field": "my_inference_field",
            "query": "my query text"
        }
    }
}
```

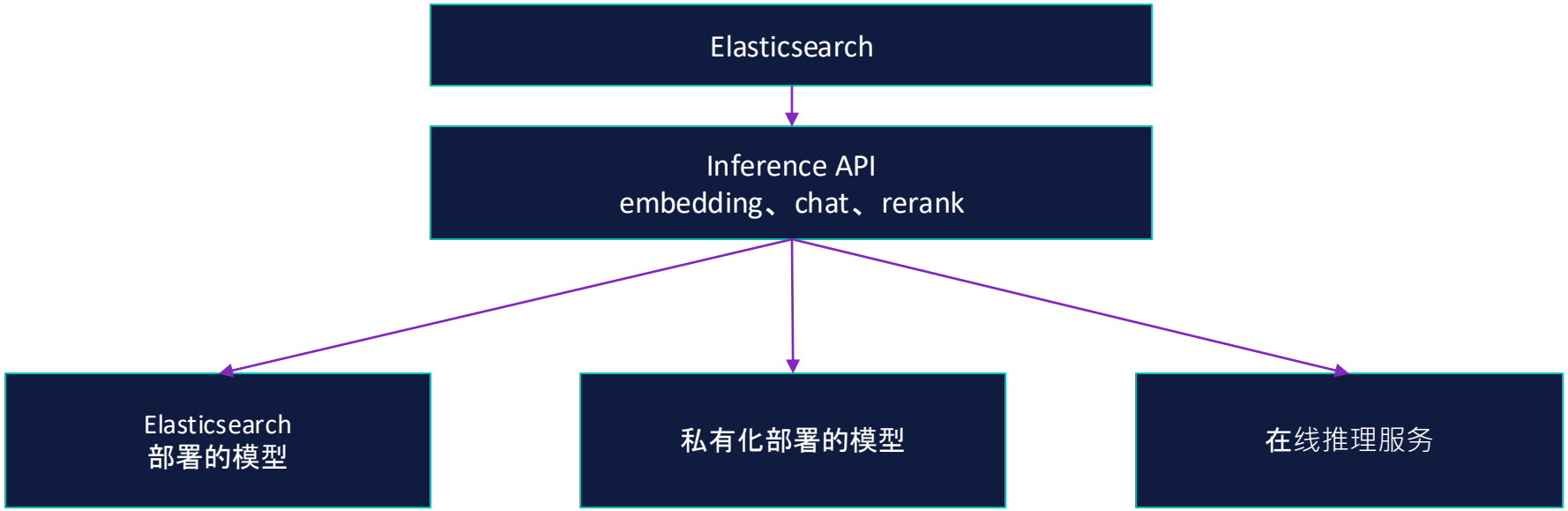


简化向量搜索使用 8.18+

```
GET test-index,index-lexical/_search
{
    "query": {
        "match": {
            "field": "my_field",
            "query": "my query text"
        }
    }
}
```

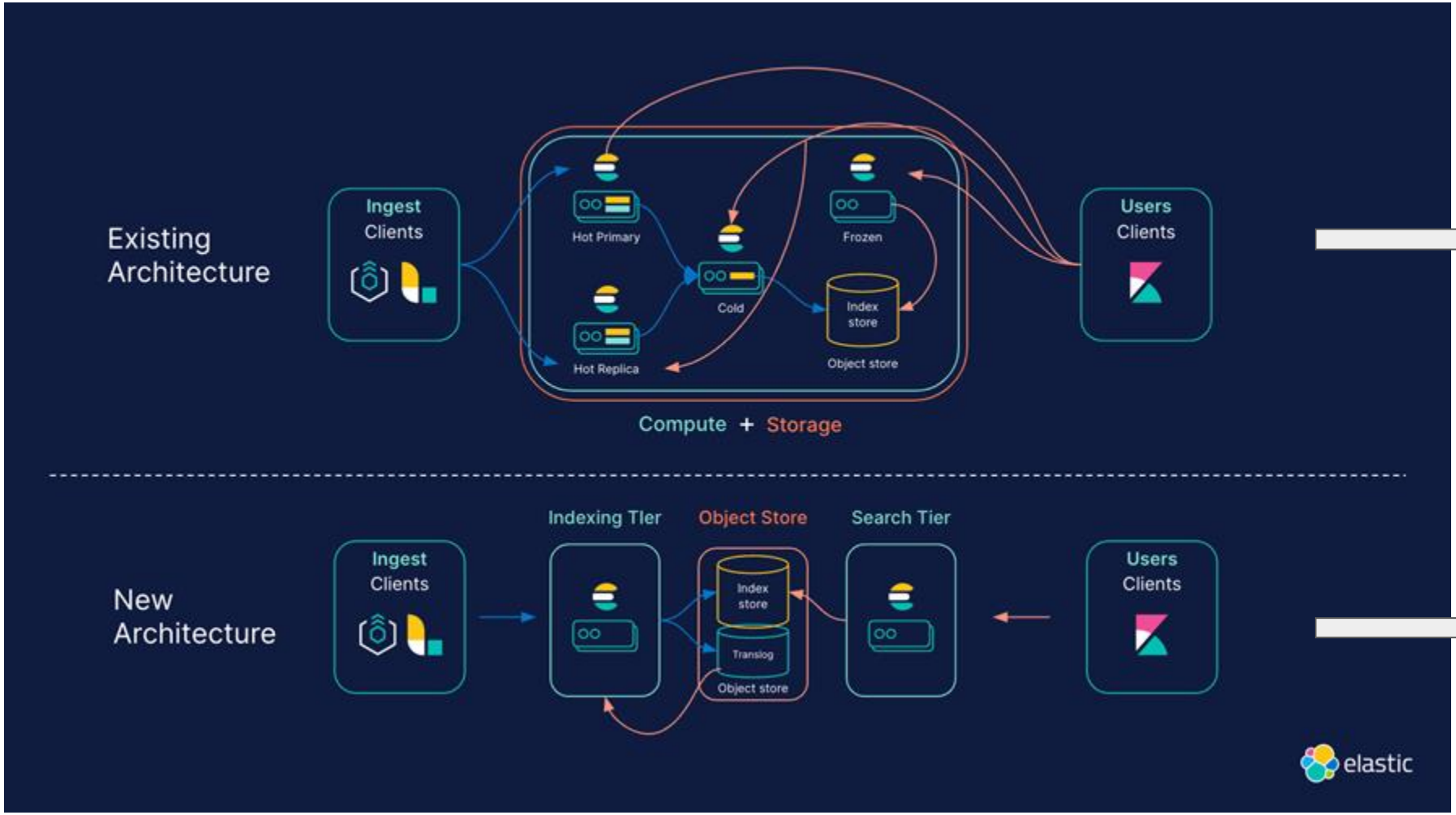
```
POST _query?format=txt
{
    "query": """
        FROM dense_vector METADATA _score
        | WHERE inference_field:"百度法定代表是谁?"
        | SORT _score DESC
    """
}
```

Elasticsearch AI 功能和集成第三方服务



阿里云、亚马逊网络服务 (AWS)、Anthropic 的 Claude、Cohere、Confluent、Dataiku、DataRobot、Galileo、谷歌云、Hugging Face、LangChain、LlamaIndex、Mistral AI、微软、NVIDIA、OpenAI、Protect AI、Red Hat、Vectorize.io 和 Unstructured。

Elastic serverless - 存算分离



Elastic Cloud Hosted

Elastic Cloud Serverless

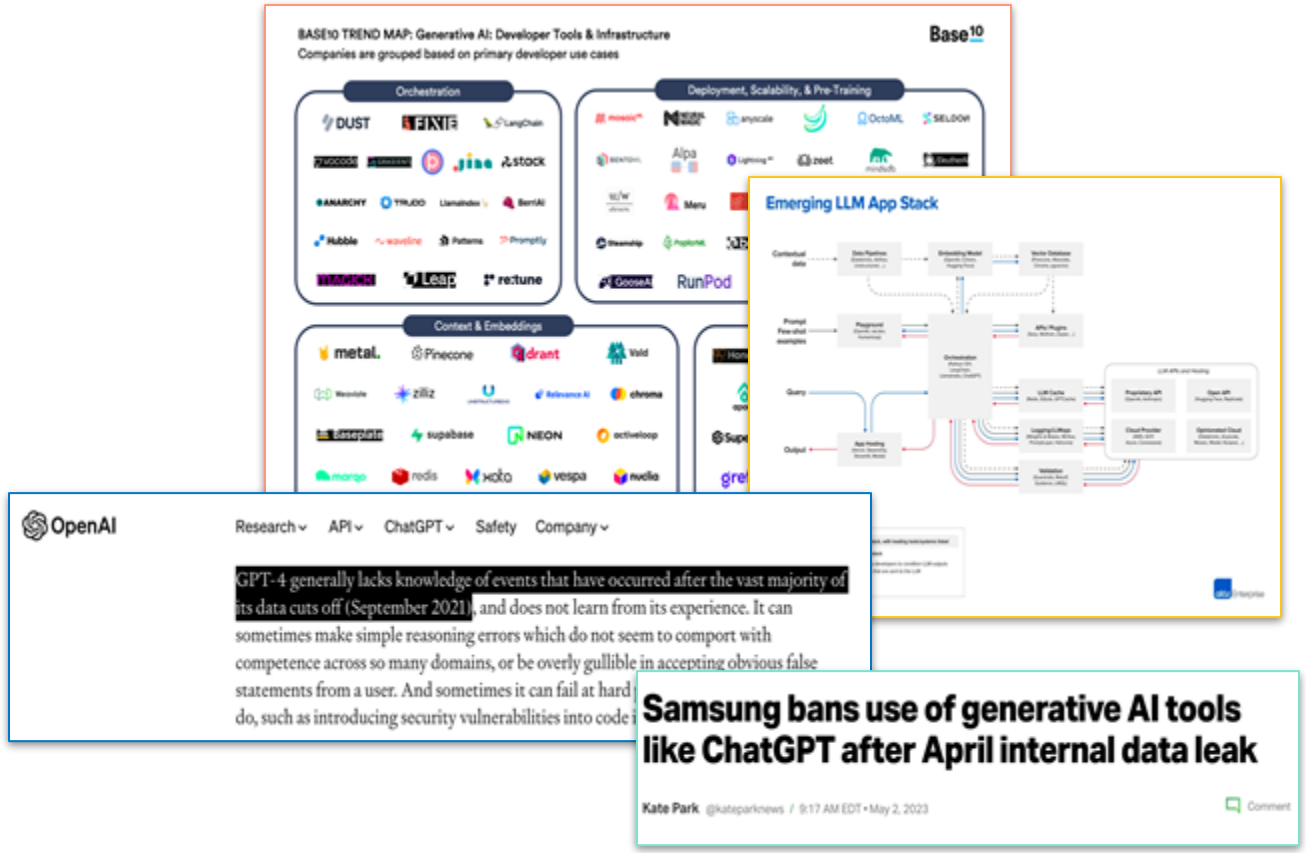
PART 03

RAG 实现原理

生成式人工智能

面临着特定的挑战

- 基础模型知识仅限于公共训练数据
- 训练和微调后数据就冻结
- 幻觉、错误答案
- 复杂的技术堆栈
- 实时访问私人数据
- 安全和隐私
- ...



LLM 变 “聪明”的三种方式

Pre-training

基础模型的训练成本高达数千万到数亿美元。 LLM 从大量公共数据集中学习语言和知识。

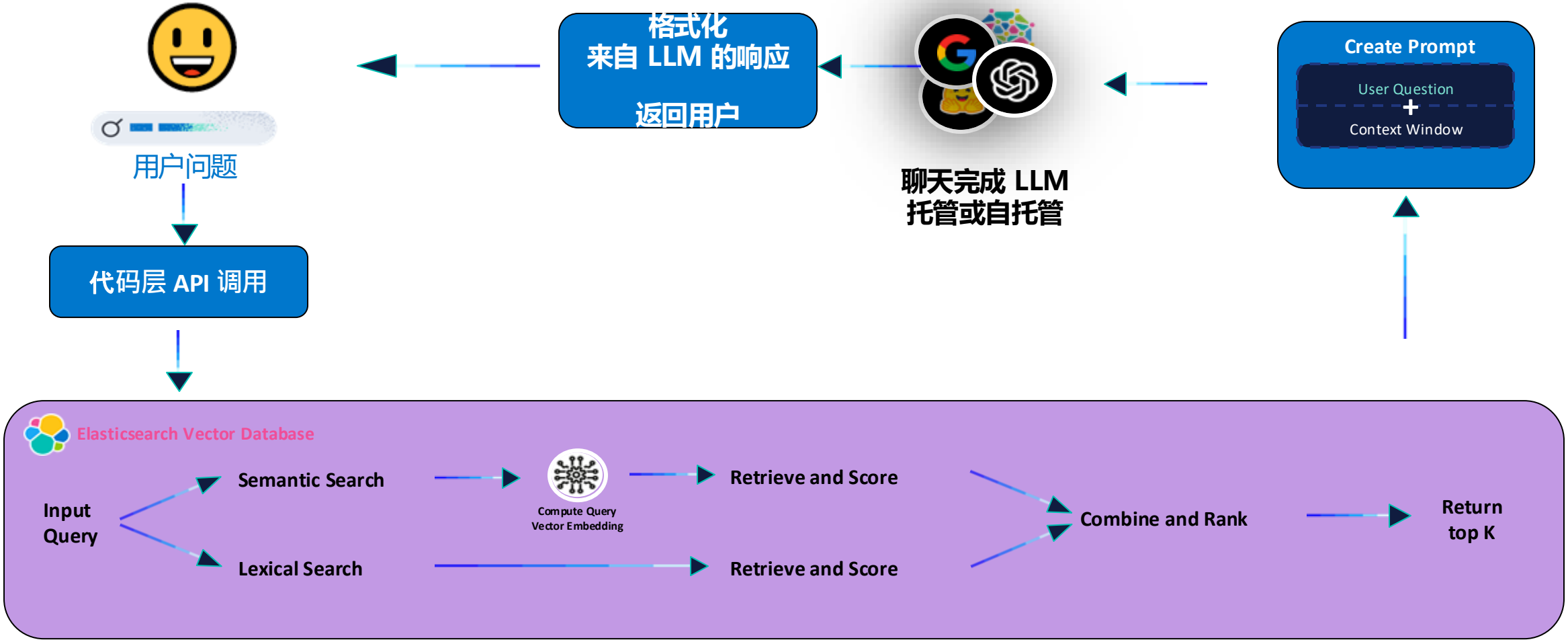
Fine-tuning

- 特定任务培训（分类等）
- 提高某一领域的响应质量
- 添加来自特定数据源的知识
- 符合保障措施和道德限制

情境学习（ prompt）

- **Prompt engineering** 技术
  Retrieval Augmented Generation - RAG

Retrieval Augmented Generation：多路召回



AiDD 7th 2025

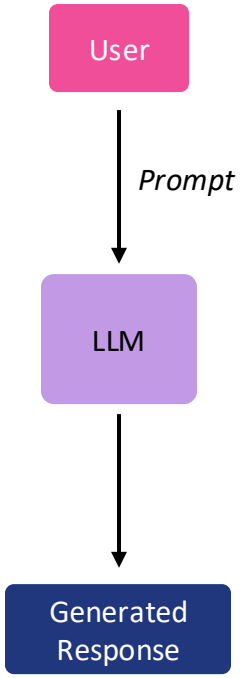
PART 04

使用 Elasticsearch 在企业搜索中的案例分享

从文本生成到决策 - Agentic RAG

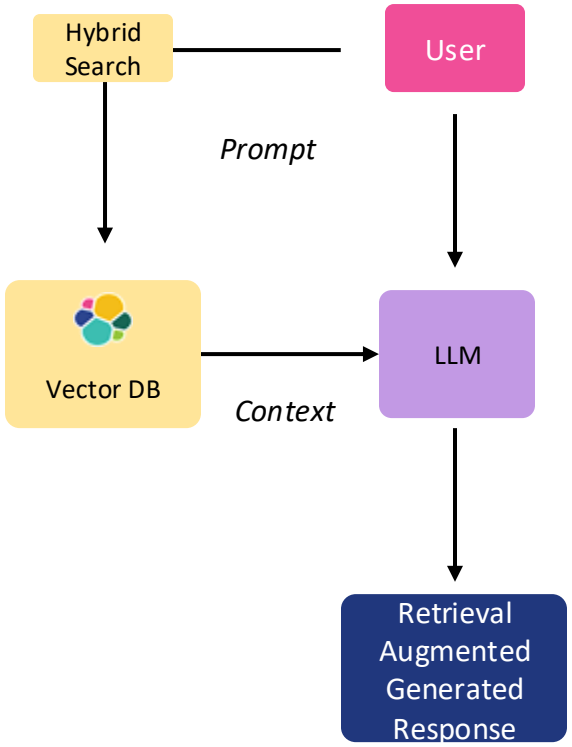
Prompting an LLM

提示 LLM 并得到回应。不需要其他工具或组件。



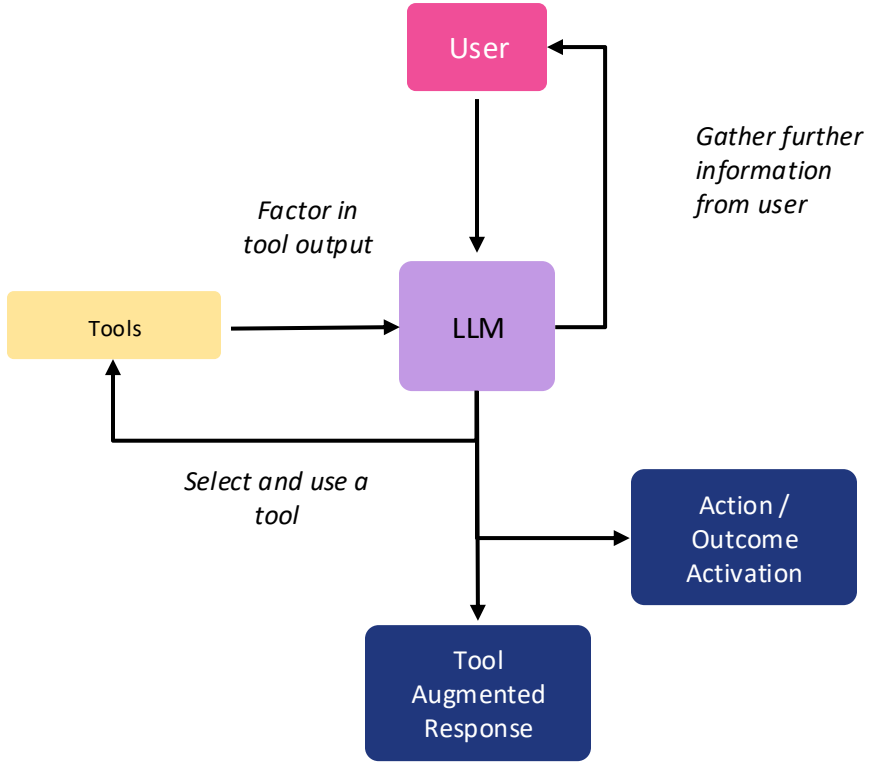
Retrieval Augmented Generation

添加知识库以提高准确性并为 LLM 添加新信息

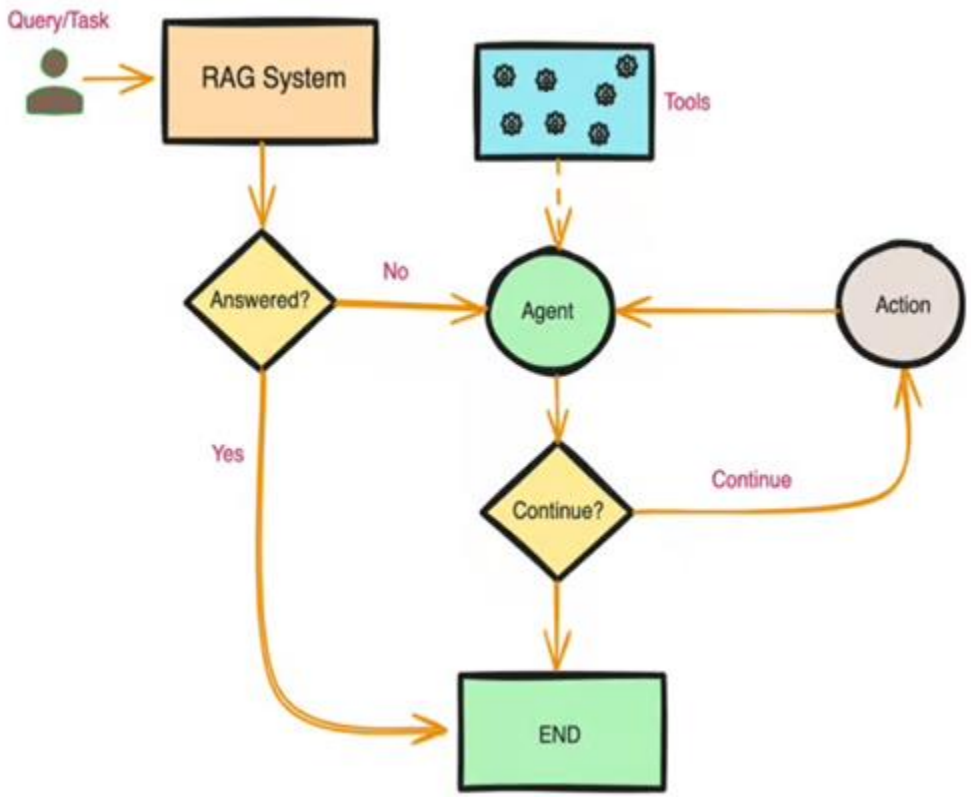


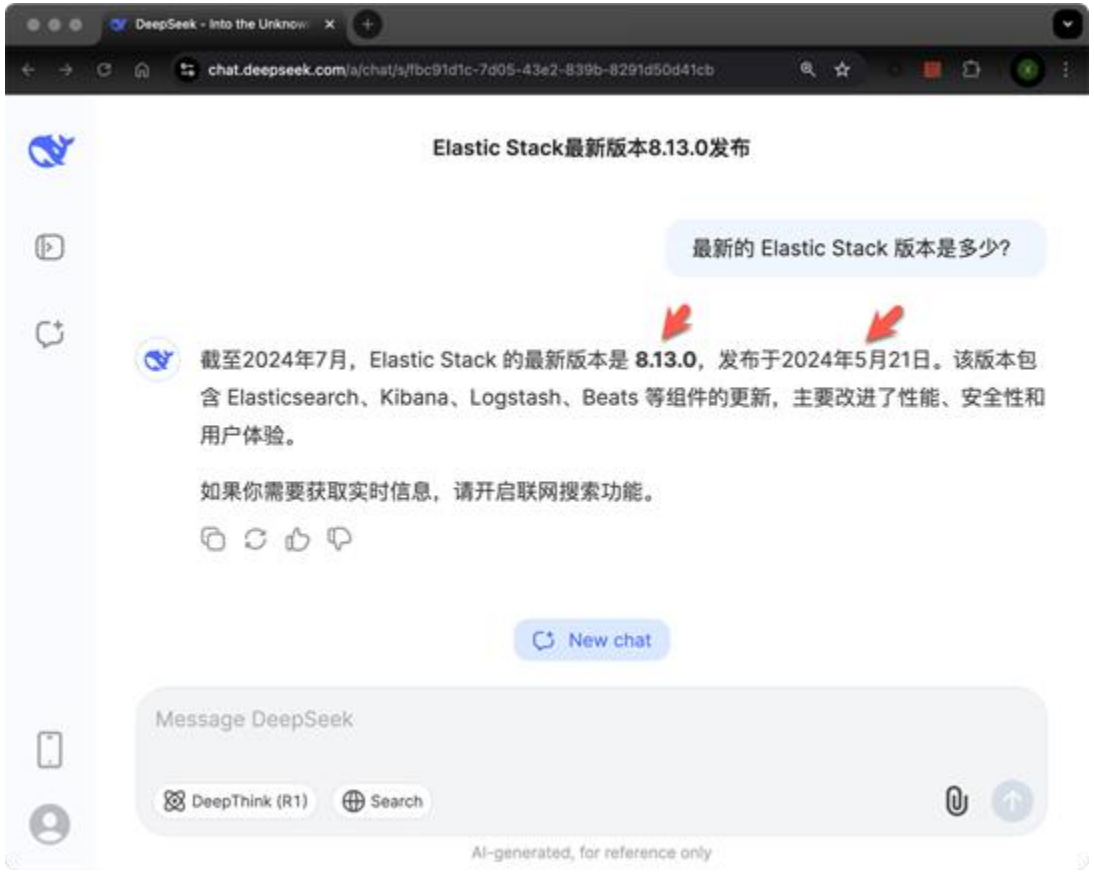
Agentic Flow

已启用决策功能。 LLM 可以提示用户信息，选择使用工具，与其他代理交互，并影响现实世界（即触发警报、发送消息等）



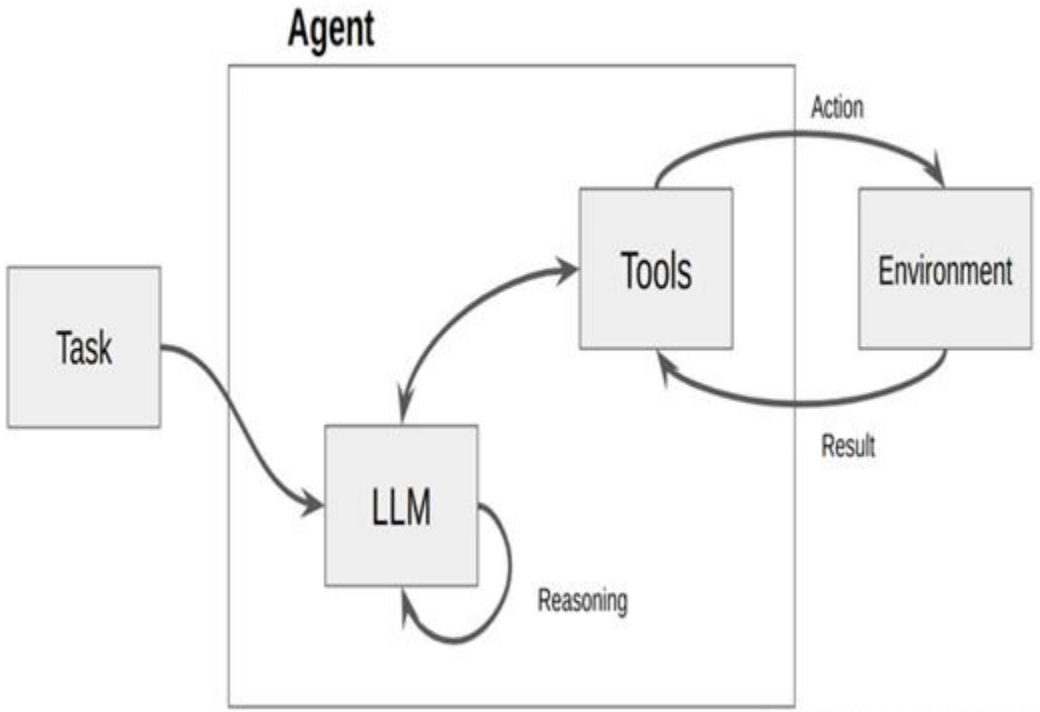
用于路由的自适应 RAG





[Agentic RAG 详解 - 从头开始构建你自己的智能体系统]

基于 Elasticsearch Agent 对文档的搜索



User 03:15:28 PM
what is the "OriginCityName? for the cheapest flight?
Took 1 step

Chatbot 03:15:37 PM
The "OriginCityName" for the cheapest flight is "Vienna". The average ticket price for this flight is approximately 100.02.

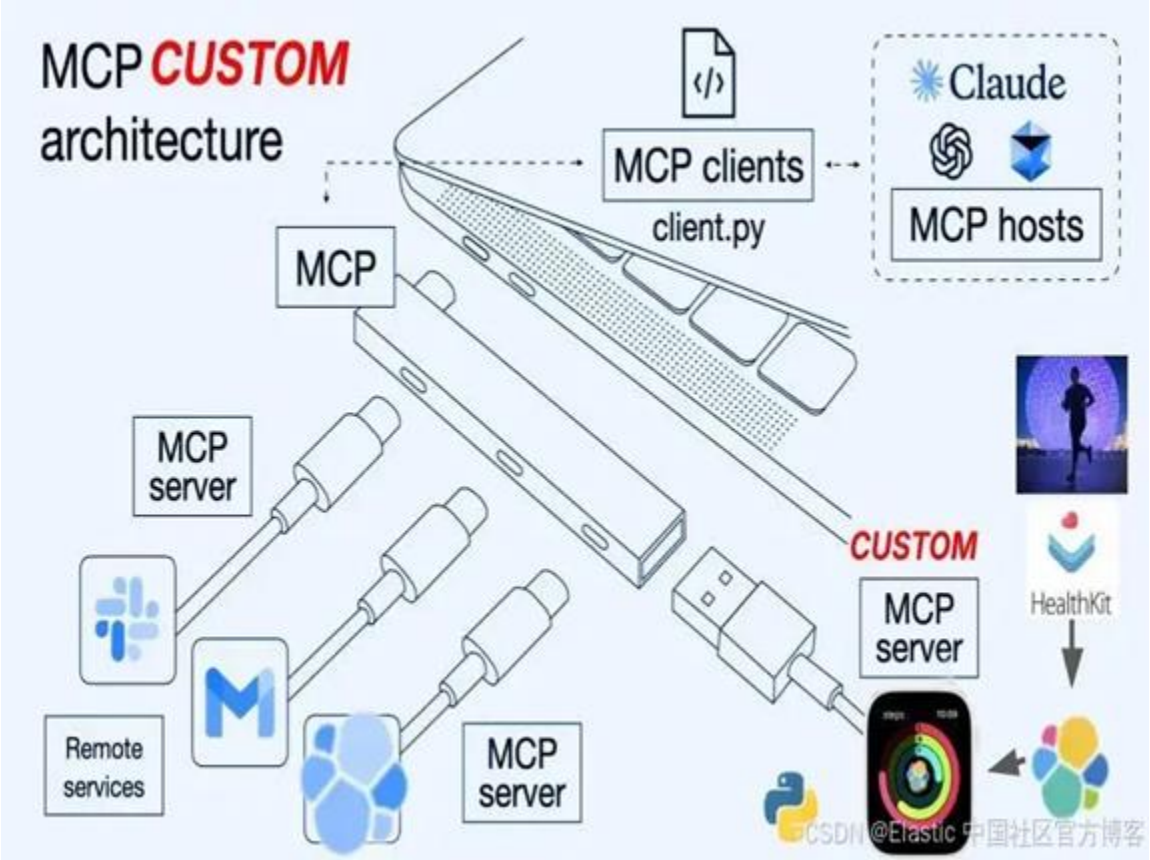
User 03:23:08 PM
What is the cheapest price from CN to US? and tell me the OriginCityName and DestCityName
Took 1 step

Chatbot 03:23:22 PM
The cheapest flight from China (CN) to the United States (US) has a ticket price of approximately $272.68. The flight originates from Guangzhou and the destination city is Tulsa.

User 03:31:23 PM
从中国到美国的最低价格是多少？请告诉我出发城市名称和目的地城市名称。
Took 1 step

Chatbot 03:31:36 PM
从中国广州到美国的最低价格是272.68美元，出发城市名称是广州，目的地城市名称是塔尔萨。

使用 MCP server 来实现 Agentic RAG



• 女性（Female）文档数量：5个

总共10个文档中，男女比例是1:1，各占50%。

他们的平均年龄是多少?

让我查询people索引中所有人的平均年龄。我会使用聚合查询来计算平均值。

search

让我再试试看具体的年龄数据：

search

现在我可以看到所有人的年龄数据。让我计算一下平均年龄：

年龄数据：30, 28, 26, 32, 29, 35, 27, 31, 34, 37

总年龄：30 + 28 + 26 + 32 + 29 + 35 + 27 + 31 + 34 + 37 = 309

平均年龄：309 ÷ 10 = **30.9岁**

所有人的平均年龄是30.9岁。

Retry

Claude can make mistakes. Please double-check responses.

HyDE (Hypothetical Document Embedding)

假设文档（Hypothetical Documents）更能反映你的文档结构。使用假设文档而不是嵌入可以显著提高相关性！

Original Query

Tell me about Fish.

LLM

Generated Hypothetical Document

Fish: A Comprehensive Overview

Fish are aquatic vertebrate animals that live in water bodies across the globe. They are characterized by several key features:

Physical Characteristics:

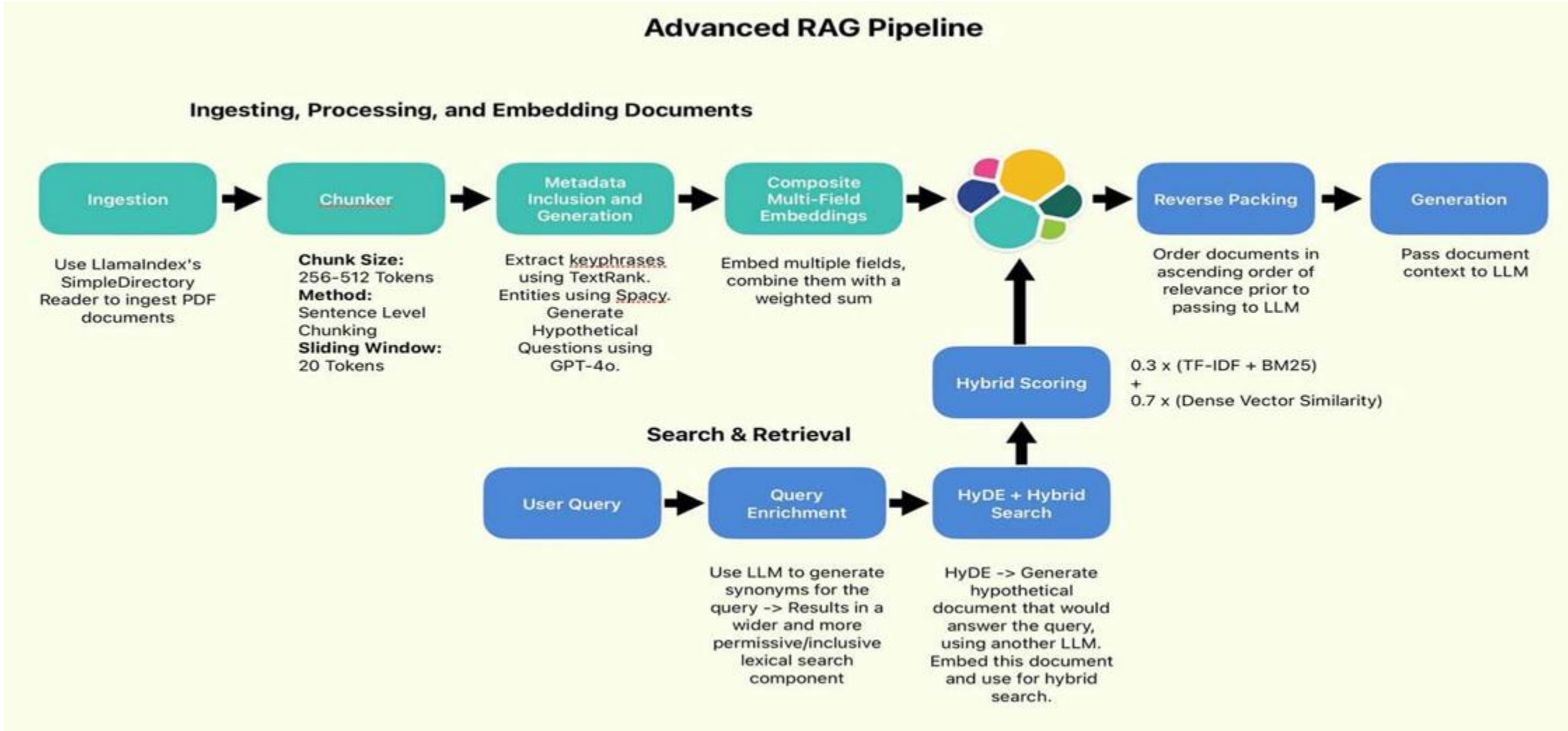
- Gills for breathing underwater
- Fins for movement and stability
- Scales covering most species
- Streamlined bodies for efficient swimming
- Cold-blooded (ectothermic)

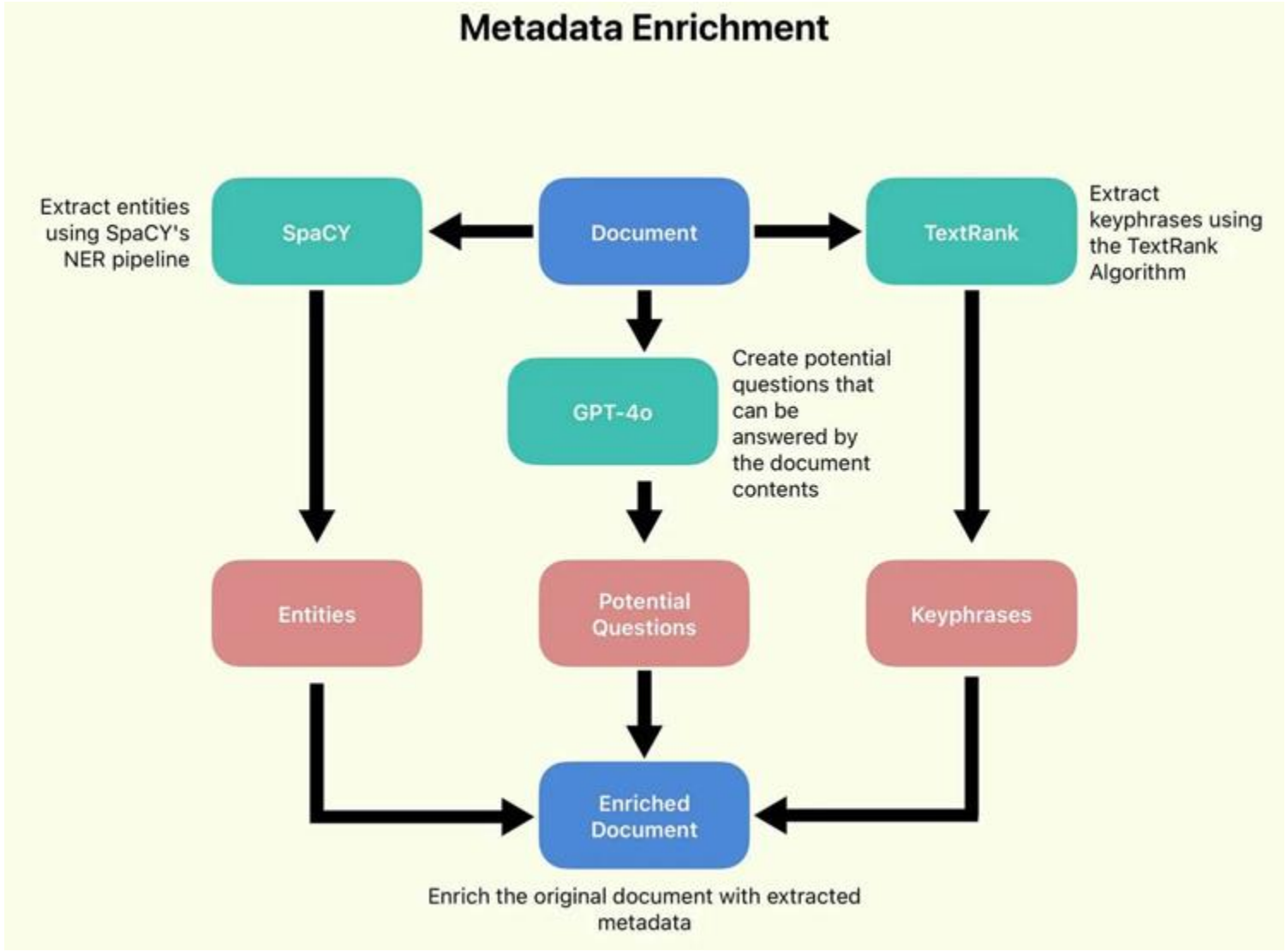
Vector Similarity Match

Search Result Document

A fish (pl.: fish or fishes) is an aquatic, anamniotic, gill-bearing vertebrate animal with swimming fins and a hard skull, but lacking limbs with digits. Fish can be grouped into the more basal jawless fish and the more common jawed fish, the latter including all living cartilaginous and bony fish, as well as the extinct placoderms and acanthodians.

混合搜索及查询重写提高搜索精度及召回率






```
embedding_cols=[
        'keyphrases_embedding',
 'potential_questions_embedding',
        'entities_embedding',
        'chunk_embedding'
 ]
```

```
combination_weights=
[
  0.1,
  0.15,
  0.05,
  0.7
 ]
```

AiDD 8th 2025

第8届AI+研发数字峰会

拥抱AI 重塑研发 AI+ Development Digital Summit

下一站预告 11/14-15 | 深圳站 12/19-20 | 上海站

查看会议详情

深圳站论坛设置

智能装备与机器人

超越“编程 Copilot”

下一代知识工程

智能网联与汽车智能化

AI 测试工具开发与应用

AI 基础设施和运维

数据智能及其行业应用

可信 AI 安全工程

大模型和 AI 应用评测

多 Agent 协同框架

从智能测试到自主测试

大模型推理优化

多模态 LLM 训练与应用

智能化 DevOps 流水线

上下文工程

AiDD

「深行·浅智」

Walk Deep, Think Light.

2025.11.16

AiDD首届麦理浩径徒步

KEYLINK ing

科技生态圈峰会 + 深度研习

——1000 + 技术团队的共同选择

AiDD峰会详情

上海站

AI+研发数字峰会

AiDD 峰会

时间：

2025.05.23-24

北京站

AI+研发数字峰会

AiDD 峰会

时间：

2025.08.08-09

深圳站

AI+研发数字峰会

AiDD 峰会

时间：

2025.11.14-15

香港站

「深行·浅智」

AiDD 首届麦理浩径徒步

时间：

2025.11.16

上海站

AI+产品创新峰会

AiDD × 联合峰会

时间：

2025.12.19-20

AiDD 7th 2025 第7届AI+研发数字峰会
AI+ Development Digital Summit

感谢聆听！

扫码领取会议PPT资料