

TECH DAY 腾讯技术开放日

腾讯云工具指南

05 AIGC时代的数据管理 -向量数据库

目录

CONTENTS

01 产品价值—向量数据库是AGI时代的数据枢纽



- 全球产业数据库具有怎样的关键发展趋势?
中国信通院人工智能创新中心负责人、云计算与大数据研究所副所长——魏凯 05
- 为什么AGI时代需要向量数据库?
腾讯集团高级执行副总裁、云与智慧产业事业群CEO——汤道生 06
- 优质的向量数据库应该符合怎样的标准?
中国信通院云计算与大数据研究所大数据与区块链部主任——姜春宇 07
- 腾讯云向量数据库的具体优势及实践
腾讯云创始团队成员、腾讯云数据库副总经理兼向量数据库负责人——罗云 08

02 行业实践—向量数据库解决各行各业智能化场景痛点



- X百川智能：解决大模型企业的数据大规模与高性能需求 10
- X销售易：向量数据库在智能CRM的实践 12

03 用户声音—开发者的产品实测报告



- 手把手带你5分钟构建 以图搜图系统
码农学习联盟 15
- 重生之我是 戏精之王
好未来 19




PART 01

产品价值

向量数据库是AGI时代的数据枢纽

SAGI时代的到来激发了数据更大的生产力——
如何更好的管理、存储、检索非结构化数据
将决定大模型在各行各业的应用前景及可能性
而为AI而生的向量数据库，或许是大模型的“最佳拍档”



全球产业数据库具有怎样的关键发展趋势？

中国信通院人工智能创新中心负责人、云计算与大数据研究所副所长 魏凯



趋势1:从类型看，非关系型数据库前景广阔

就目前全球数据库产品分布来看，非关系型数据库数量已经超过了关系型数据库。相比之下，国内的数据库市场依然以关系型数据库为主（整体占比超过65%），这其中又以图数据库、时序数据库等关键领域类型数据库为主。

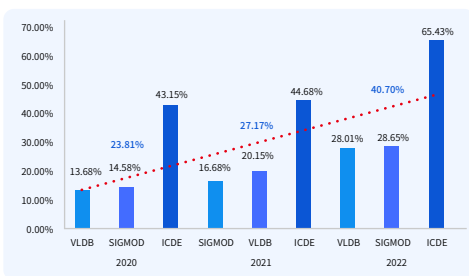


数据来源：CCSA TC601, 2023年6月

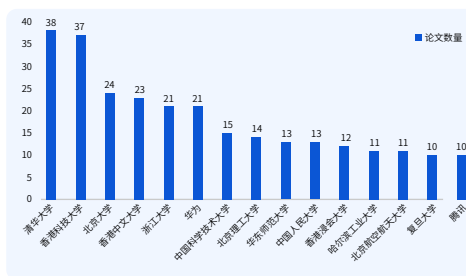
趋势2:从创新看，我国的非关系型技术实力不断增强

从VLDB、SIGMOD和ICDE三大数据库领域权威的学术会议来看，近三年，我国企业及高校平均贡献占比分别为23.81%、27.17%和40.70%，且数量呈逐年上升趋势。

2020-2022年我国中国高校及企业学术会议论文贡献情况



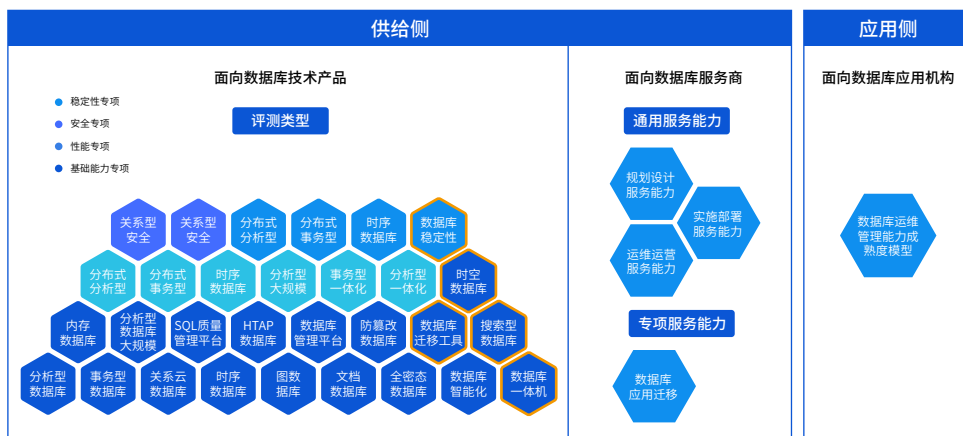
2022年我国中国高校及企业学术会议论文贡献数量



数据来源：CCSA TC601, 2023年6月

趋势3:从标准看，数据库需要从供给侧到标准侧制定相关标准

目前数据库行业玩家众多，亟需统一行业标准规范发展：主要面向三类参与方（数据库技术产品、数据库服务商以及数据库应用机构）建立不同的评判标准。



2023上半年新增5个标准（上述图中标橙色显示）

为什么AGI时代需要向量数据库？

腾讯集团高级执行副总裁、云与智慧产业事业群CEO 汤道生



最近这一年，AIGC浪潮席卷全球，很多志向高远的企业，都在思考如何拥抱新技术，以AI重塑公司的业务和产品。很多客户也和腾讯开展了紧密的合作，探索如何将大模型在实际场景中用起来，帮助业务降本、提效、增收。

众所周知，大模型应用的关键，不只是构建好模型算法，更重要的是做好数据的处理、挖掘等问题。数据贯穿了大模型从预训练到产业落地的全过程。一定程度上，智能时代，企业数据处理能力有多强，决定了业务发展的天花板有多高。

企业在搭建和使用大模型时，需要把海量数据，安全高效地接入大模型，但在企业复杂的数据中，适合关系型数据库的，结构化数据仅有20%，其余80%是文本、图像、视频、音频等非结构化数据。

向量数据库可以把复杂的非结构化数据，处理成多维逻辑的坐标值，与大模型进行连接，数据处理效率比传统方式提升10倍。

同时，向量数据库也可以作为“外部知识库”，给大模型输送最新、最全面的信息，应对有时效性的问答；并且让大模型拥有长期记忆，避免聊天时“断片”。可以说是大模型的“最佳拍档”。

AI驱动产业变革的时代正在到来，作为支撑大模型的重要基础设施，向量数据库也会从一个“领域型数据库”，变成覆盖广阔场景的“通用型数据库”，甚至是“数据枢纽”，前景广阔。

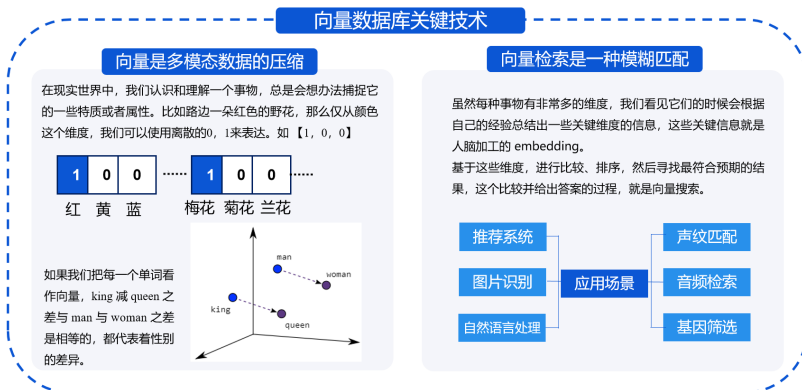


优质的向量数据库应该符合怎样的标准？

中国信通院云计算与大数据研究所大数据与区块链部主任 姜春宇



随着人工智能时代到来，一些非结构化数据需要通过机器学习算法从中提取出以向量为表示形式的“特征”，向量数据库的兴起便是为了解决对这些向量进行存储与计算的问题。



相比其他类型数据库，向量数据库具有8大关键技术能力：

分布式与并行计算能力

更高效的分布式与并行计算可以让大规模向量数据在多个计算节点间进行分配，使得查询、排序等操作能够并发进行，大大缩短了计算时间。

实时处理能力提升

许多AI应用需求求向量数据库有高效的实时处理能力，即使是对大规模的向量数据，也能在最短的时间内找到最匹配的结果。

高级查询功能

高级查询功能，如范围查询、最近邻查询，甚至基于语义的查询等，将是向量数据库的必备功能。

硬件加速

为了更高效地处理数据，硬件加速将是一种有效的解决方案。利用GPU的强大并行计算能力，或者利用定制的AI芯片，都可以大大提高向量数据库的处理能力。

不同大模型的性能优化

不同类型的大模型对数据的处理和计算需求可能会有所不同。向量数据库需要能够针对这些差异进行优化，以提供最佳的性能。

多模态数据处理能力

随着大模型向多模态发展，如图文混合模型、音视频混合模型等，对应的数据也将会更为复杂多元。向量数据库需要能够有效地处理这些多模态数据。

提升通用性和易用性

随着向量数据库的应用场景不断拓宽，提升其通用性和易用性成为一项重要任务。这包括提供更简单的数据导入导出，提供更易用的查询接口，以及提供更灵活的数据管理功能。

与大模型的深度融合

未来，向量数据库将与深度学习、大模型更紧密地结合，共同推动AI的发展。向量数据库需要能够理解大模型的需求，为其提供最合适的数据服务。而大模型也需要能够利用向量数据库的能力，以提高自身的效率和效果。

对此，信通院联合腾讯云等多家关键厂商制定《向量数据库技术要求》，针对七大能力域、三十二个能力项制定向量数据库行业标准。



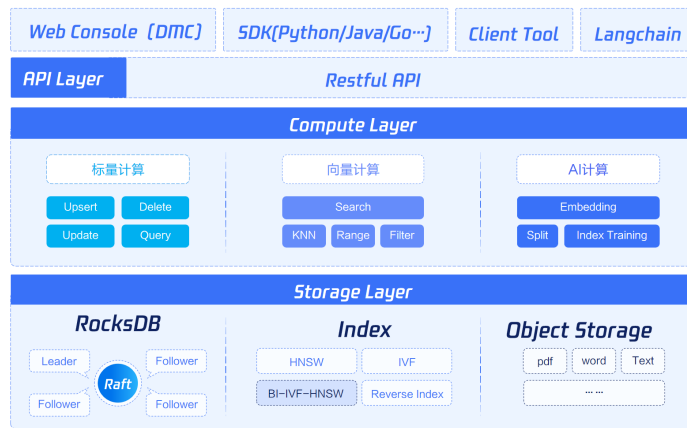
腾讯云向量数据库的具体优势及实践

腾讯云创始团队成员、腾讯云数据库副总经理兼向量数据库负责人 罗云

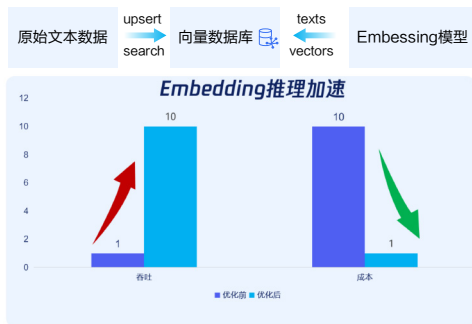


腾讯云将向量数据库定义为AGI时代的数据枢纽，其需要具备“企业化”及“智能化”两项关键能力：前者需要满足企业对分布式、高性能、高可用、安全性、可靠性及成本可控六个关键能力；后者则需要实现借口、计算、存储三个关键领域的智能化。

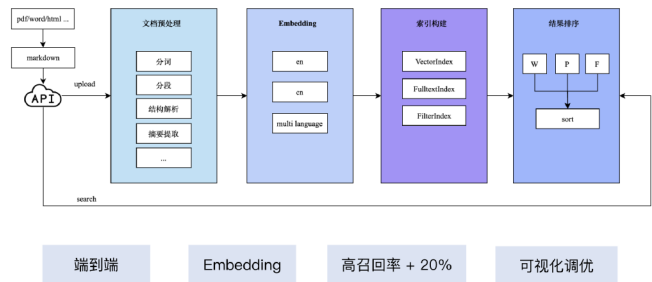
一、企业化能力：千亿级数据规模、500万QPS、99.99%可用性



二、智能化能力：内容召回率提升20%，推理速度大幅提升



集成Embedding，实现自然语言查询



AI套件：端到端的RAG应用检索方案

三、实践成果：集团内部 40+业务接入，1600亿次请求/天；1000+外部用户接入



集团内部



外部用户



PART
02

行业实践

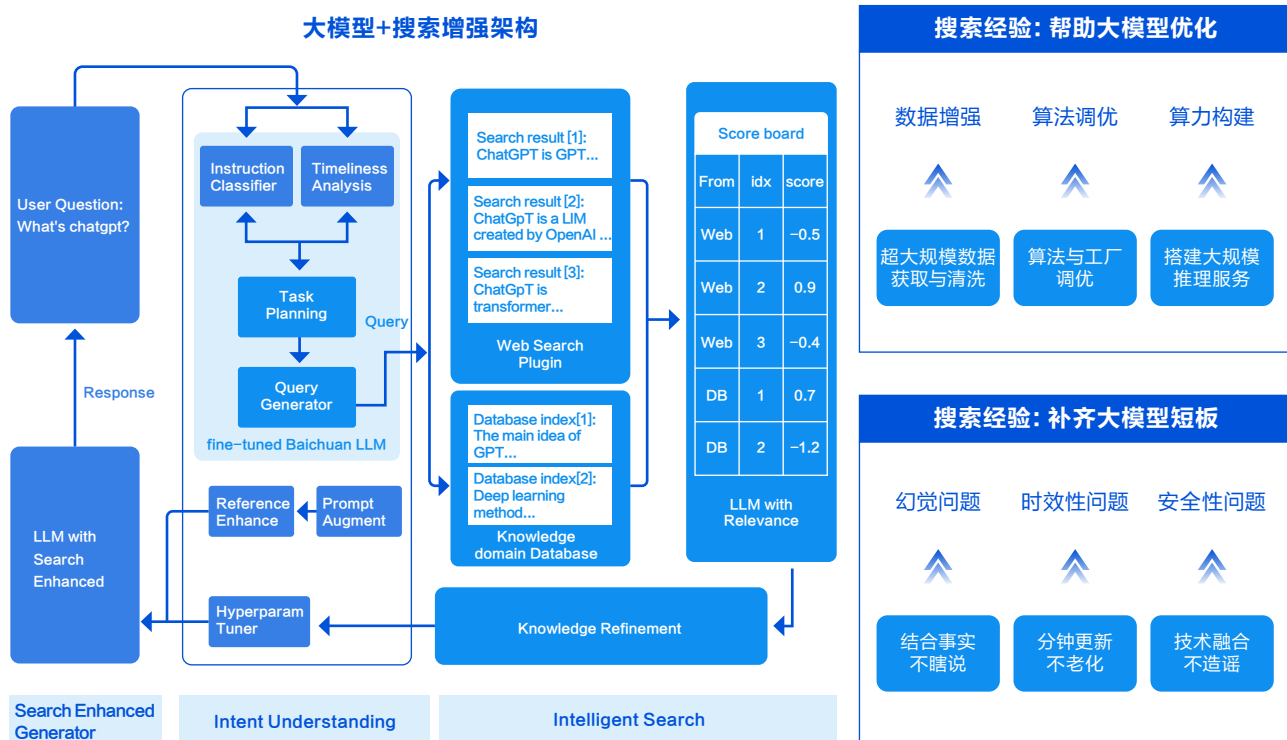
向量数据库解决各行各业智能化场景痛点



X百川智能 解决大模型企业的 数据大规模与高性能需求

客户场景：搜索增强

百川智能是一家为客户提供大模型服务的能力，基于搜索与输入法多年积累，以RAG框架为原型融合企业私有数据、实时性数据。为客户提供搜索增强的能力，一方面基于搜索经验优化大模型，另一方面基于搜索模型补齐大模型短板，解决大模型应用常见的模型幻觉与数据时效性问题。



场景痛点

百川智能作为模型服务提供商，数据规模在亿级以上。使用向量数据库方案之前主要以“开源向量算法”支撑，其面临两个主要痛点：

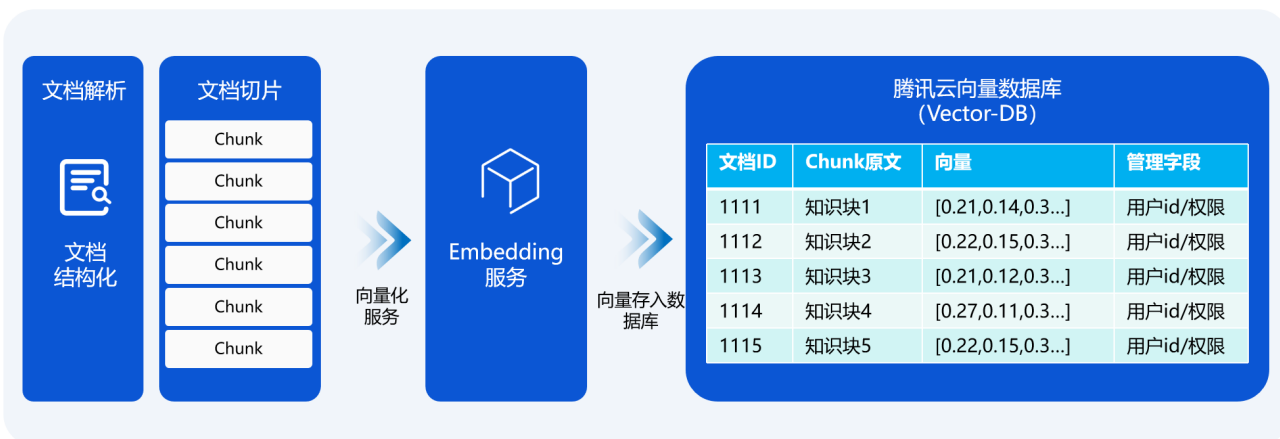
- 1.可用性较弱。**作为算法级别的功能，在分布式系统下会存在扩展性、可用性等方面问题。
- 2.消耗大量人力做二次开发。**因为没有成熟体系化，需要消耗大量人力做二次定制开发。比如说向量数据的格式组织及管理，数据分片、导入、管理、删除、索引等。

解决方案：向量数据库的统一技术栈支持

一、管理个性化知识：在企业知识文档上传时统一化存储管理

传统ES数据库是为关系型结构数据设计，向量数据库基于AI而生集成Embedding服务进行向量化处理——

- 把原始长文本内容分割为表征能力更强的短文本
- 通过Embedding模型生成向量化数据
- 将向量化数据在数据库内分区管理
- 运用可视化数据管理平台在线执行互表操作



二、问答个性化知识：在问答企业知识问题时实现搜索增强

当大模型面对企业级数据问答时，如果不能融合企业私有数据及实时性数据即会出现常见的幻觉问题。向量数据库提供端到端的RAG检索方案。当外部用户向模型起初问题时——

- 基于Embedding对问题结合其聊天历史向量化解析
- 通过在线及离线方式检索融合知识库及网页端内容
- 基于相关性模型进行精排再求解输出
- 大模型基于更全面与实时数据推理答案并召回数据



X销售易 向量数据库在智能CRM的实践

Neocrm 销售易

客户背景

销售易智能CRM主要为客户提供“智能销售”与“智能服务”功能，前者包括：智能销售助理、日程会议纪要、客户画像标签、推荐解决方案、推荐潜在客户；后者包括：客服机器人、坐席辅助、智能会话质检、生成知识条目、工单创建分配。



场景痛点

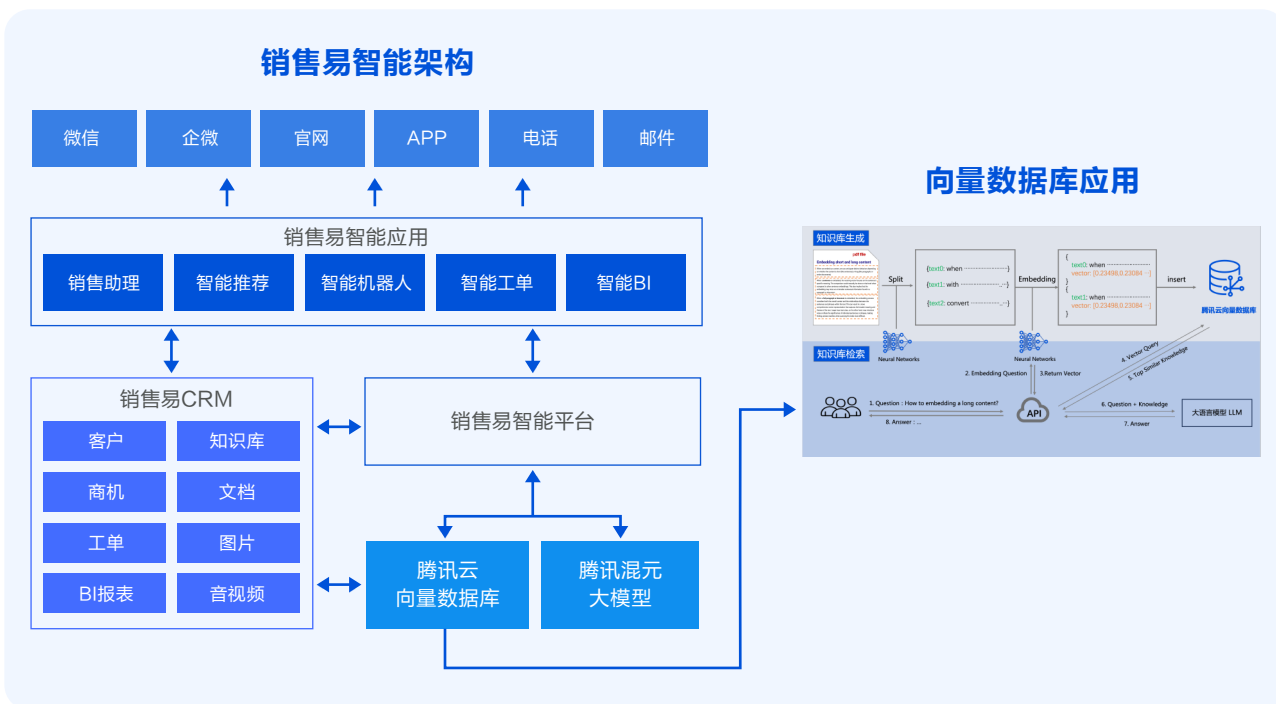
企业服务市场应用大模型的主要门槛是满足企业业务数据的安全性与时效性——

- 1.数据安全:** 企业的业务数据需要满足安全合规要求，不能直接用于大模型训练
- 2.快速变化:** 企业时刻都在产生大量的数据，如何适应业务数据快速变化的需求
- 3.实时响应:** 业务系统对实时响应要求很高，需要从海量数据中迅速检索问题答案
- 4.场景落地:** 销售和服务场景众多，如何将数据结合大模型来解决实际业务问题

原来的关系型数据库因为其数据结构原因，更多应用于表单级系统，无法支撑智能化需求，主要因为——

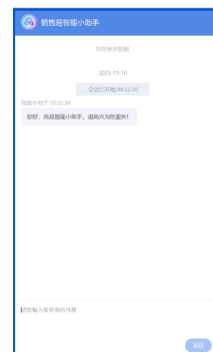
- 1.其检索时依赖于关键词文本搜索而非语义搜索，需要做大量分词的词库词法维护
- 2.其推荐时无法理解自然语言描述文本，无法做智能化推荐

销售易主要应用向量数据库于底层知识库生成与检索



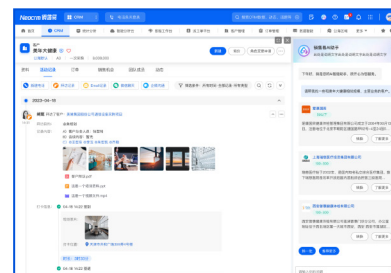
基于向量数据库，智能机器人可以实现——

- 意图识别:** 通过语义分析和识别，在问题描述模糊的情况下，精准识别用户意图，结合知识库资源，准确回答问题
- 检索信息:** 自然语言描述问题和需求，利用语义相关性检索知识库和CRM系统数据，综合内容生成结果
- 多轮对话:** 对多轮对话和上下文理解，让用户感受到拟人化的服务体验，实现了自然、流畅的对话。
- 创建信息:** 自主调用CRM系统API，创建CRM线索记录，保存潜在用户的信息；或创建服务工单，记录用户问题。



基于向量数据库，全内容语义检索和推荐相似客户——

- 知识库文档与语义搜索: 知识库、文档、附件
- CRM数据相关性搜索: 检索对象、会议纪要、活动记录
- 推荐相似客户: 基于相似特征及内容 (而不是传统表单关键词检索)



向量数据库在SaaS领域的应用优势

- 高性能大规模:** 企业的业务数据需要满足安全合规要求，不能直接用于大模型训练
- 支持向量和标量:** 支持向量和标量字段的混合存储和检索，是文档内容和结构化字段之间的桥梁。
- 运维轻量简便:** 按照帮助手册简单接入，无需安装、部署和运维，有效减少运维成本和人力成本。
- 专家级服务支持:** 腾讯云数据库团队提供专家级的技术方案指导，赋能研发团队加速产品和架构落地。



PART
03

用户声音
开发者的产品实测报告



手把手带你5分钟构建 以图搜图系统

码农学习联盟

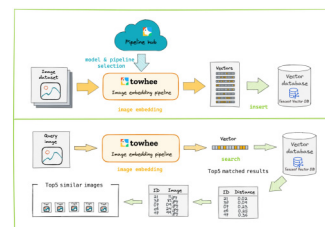
以图搜图案例

我们使用 Towhee 和 腾讯云向量数据库构建一个以图搜图 (Reverse Image Search) 系统。

该系统以图片作为输入，基于图片的内容检索出最相似的图片。其背后的基本思想是利用预训练的深度学习模型提取出每个图片的特征，并将其表示为一个嵌入向量 (Embedding)。然后，通过存储和比较这些图片嵌入向量，实现图片的检索。

工作流程如右图：

首先，使用 Towhee 对输入图片进行预处理并提取特征，得到图片的嵌入向量。然后，将这个嵌入向量存入 向量数据库 中。当需要检索图片时，同样先对查询图片进行预处理和特征提取，得到查询图片的嵌入向量。然后，将这个嵌入向量输入到 向量数据库中，向量数据库会返回最相似的图片。



Towhee 是一个用于构建强大的数据流水线的开源框架，它可以有效地处理各种数据转换任务。

构建项目

下面会对重要的代码部分做详解，最终的 demo 代码，可以在文末获取，代码拉到本地就可以运行，对新手很友好。

1. 创建一个新的项目目录：

```
Bash | 复制代码
1 mkdir -p image-search
2 cd image-search
```

2. 创建一个新的 Python 虚拟环境 (可选，但推荐)：

```
Bash | 复制代码
1 $ python -V
2 Python 3.9.0
3
4 $ python -m venv venv
```

创建一个新的 Python 虚拟环境能有效地隔离项目依赖，简化依赖管理。

3. 安装需要的 Python 包：

```
Bash | 复制代码
1 python -m pip install -q towhee opencv-python pillow tcvectordb
```

这个命令会将 Towhee, OpenCV, Pillow, tcvectordb 库安装到上面创建的虚拟目录 venv 中。

激活这个虚拟环境：

● Linux/macOS

```
Bash | 复制代码
1 source venv/bin/activate
```

● Windows

```
Bash | 复制代码
1 .\venv\Scripts\activate
```

准备数据

这里我们使用了 ImageNet 数据集的一个子集 (100 个类别)。示例数据可在 Github 上获取。

目录结构如下：

- 1.train: 包含候选图片的目录，有 100 个不同的类别，每个类别包含 10 张图片。
- 2.test: 包含查询图片的目录，与训练集同样的 100 个类别，但每个类别只有 1 张图片。
- 3.reverse_image_search.csv: 一个 csv 文件，包含每个训练集图片的 id、路径和标签。

```
Shell | 复制代码
1 curl -L https://github.com/towhee-io/examples/releases/download/data/reverse_image_search.zip
2 unzip -q -o reverse_image_search.zip
```

ImageNet 数据集是深度学习领域中广泛使用的大规模视觉数据集，用于图片分类和物体检测任务。在本文中，所使用的数据集是 ImageNet 的一个子集，这个子集为模型训练和验证提供了适当规模和复杂度的数据。

候选图片是指可能会被检索的图片，查询图片是指用于检索的图片。

连接数据库并新建 Collection

连接 Tencent Vector DB 很简单, 官方提供了多种语言的 SDK, 本文使用 Python SDK: tcvectordb 操作向量数据库。

1. 首先利用 tcvectordb sdk 编写连接向量数据库的客户端代码:

```

1 class TcvdbClient(PyOperator):
2
3     def __init__(self, host: str, port: str, username: str, key: str, dbName: str, collect
4         timeout: int = 20):
5         ---
6         初始化客户端
7         ---
8         # 创建客户端时可以指定 read_consistency, 后续调用 sdk 接口的 read_consistency 将沿用
9         self.collectionName = collectionName
10        self.dbName = dbName
11        self_client = tcvectordb.VectorDBClient(url="http://" + host + ":" + port, usernz
12        read_consistency=ReadConsistency.EVENTUAL
    
```

2. 然后调用 TcvdbClient 构建客户端:

```

1 # tcvdb parameters
2 HOST = "1b-xxxx.clb.ap-beijing.tencenttcb.com"
3 PORT = "18000"
4 DB_NAME = "image-search"
5 COLLECTION_NAME = "reverse_image_search"
6
7 PASSWORD = "xxxx"
8 USERNAME = "root"
9
10 # path to csv (column_1 indicates image path) OR a pattern of image paths
11 INSERT_SRC = "reverse_image_search.csv"
12
13 test_vdb = TcvdbClient(host=HOST, port=PORT, key="6q1v8KF0xAgZ0N7V2bcwCqrx0S436307au4R8F
14     collectionName=COLLECTION_NAME, dbName=DB_NAME)
    
```

上面的 HOST 和 PORT、USERNAME 和 PASSWORD 是申请向量数据库后获取到的。

在向量数据库中创建 DB 和 Collection:

这里代码创建一个 Collection, 并在这个 Collection 中添加了三个索引。

在向量数据库中, Collection 是用来存储和检索向量的主要结构, 创建索引的字段在检索时可以用作过滤 (filter)。

- 1.vector: 索引有 2048 向量维度。维度越高, 向量可以表达的信息越多, 但同时计算复杂度也越高, 存储需求也越大。
- 2.IndexType.HNSW 索引的类型。这是一种近似最近邻搜索算法, 用来加速高维向量的搜索。
- 3.MetricType.COSINE 是余弦相似度, 它可以衡量两个向量之间的角度, 通常用于衡量高维向量的相似性。
- 4.id 是主键索引, 用来唯一标识每个向量。
- 5.path 是过滤索引, 用来加速基于 path 字段的查询

```

1 class TcvdbClient(PyOperator):
2     def create_db_and_collection(self):
3         database = self.db_name
4         coll_embedding_name = self.collectionName
5         coll_alias = self.collectionName + "-alias"
6
7         # 创建 DB
8         db = self_client.create_database(database)
9
10        # 构造 Collection
11        index = Index()
12        index.add(VectorIndex('vector', 2048, IndexType.HNSW, MetricType.COSINE, HNSWParam
13        index.add(FilterIndex('id', FieldType.STRING, IndexType.PRIMARY_KEY))
14        index.add(FilterIndex('path', FieldType.STRING, IndexType.FILTER))
15
16        # 创建 Collection
17        db.create_collection(
18            name=coll_embedding_name,
19            shard=3,
20            replicas=0,
21            description='image embedding collection',
22            index=index,
23            embedding=None,
24            timeout=20
25        )
26
27        test_vdb.create_db_and_collection()
    
```

DMC 访问入口: <https://dms.cloud.tencent.com/>

新建之后, 可以通过 DMC (数据库管理) 方便的查看、管理向量数据库的数据。

右侧是刚刚创建的 DB 和集合:



Embedding: 图片转向量、入库

在机器学习领域中, 把文本、图片, 音频等其他类型原始输入数据转换为一种更适合机器学习的形式, 即将复杂的数据结构 (如图片、文本等) 转换为固定长度的向量的过程成为 Embedding。

右侧利用 Towhee 的 pipeline 实现图片的特征提取和向量的存储:

Embedding Pipeline 定义了一个 p_embed 的管道, 这个管道将 reverse_image_search.csv 中的图片加载后, 调用 ops.image_embedding.timm(), 使用 resnet50 模型将图片数据转换为嵌入向量。

Towhee 提供了预训练的 ResNet50 模型, 可以将图片转换为向量。ResNet50 是一种深度卷积神经网络, 它在许多图像识别任务中表现出色。此模型通过学习图片的重要特征, 并将这些特征嵌入到一个高维向量中, 称为嵌入向量 (embedding vector)。

```

1 MODEL = "resnet50"
2 DEVICE = "cpu" # If None, use default device (cuda is enabled if available)
3 INSERT_SRC = "reverse_image_search.csv"
4 # Embedding pipeline
5 p_embed = ops.pipeline()
6 p_embed.add(ops.load_image(src=INSERT_SRC))
7 p_embed.add(ops.image_embedding.timm(model=MODEL, device=DEVICE))
8
9 # Display embedding result, no need for implementation
10 p_embed.add(ops.output(img_path='img', vec='vec'))
11
12 # Insert pipeline
13 p_insert = ops.pipeline()
14 p_insert.add(p_embed)
15
16 # Insert data
17 p_insert.add(ops.insert_data(collectionName=COLLECTION_NAME, dbName=DB_NAME))
18
19 # Insert data
20 p_insert.add(ops.insert_data(collectionName=COLLECTION_NAME, dbName=DB_NAME))
21
22 # Output result
23
24
25
26 p_insert.add(ops.insert_data(collectionName=COLLECTION_NAME, dbName=DB_NAME))
    
```

Display Pipeline 代码定义了p_display的管道, 这个管道用于显示p_embed的结果。Insert Pipeline 代码定义了p_insert的管道, 这个管道用于将嵌入向量插入到向量数据库中。p_insert(INSERT_SRC)使用p_insert管道对reverse_image_search.csv文件中的图片数据进行处理。

最终会将生成的向量调用 ops.local.search_tcldb_client, 插入到向量数据库中:

```
1 = def _call_(self, *data):
2 path = ""
3 vector = []
4 for item in data:
5 if isinstance(item, np.ndarray):
6 # Convert ndarray to list and float32 to float
7 vector = list(map(float, item))
8 else:
9 path = item
10
11 # Generate a random UUID and convert to string
12 document_list = [
13 Document(
14 id=str(uuid.uuid4()),
15 path=path,
16 vector=vector),
17 ]
18 db = self._client_database(self.db_name)
19 coll = db.collection(self.collection_name)
20 coll.insert_many(document_list)
```

可以在 DMC 中, 用刚刚创建了索引的字段进行过滤, 精确查询到入库后的数据, 例如搜索: path="/train/goldfish/n01443537_1903.JPG" :



由于向量数据一般很大, 默认不会返回。如果要返回向量字段需要勾选retrieveVector。

搜索相似图

定义一个搜索管道p_search, 用于在向量数据库中搜索与输入图像最相似的图像, 然后显示搜索结果。

预搜索管道p_search_pre使用p_embed管道生成查询图片的嵌入向量vec。

后对每个元素应用一个lambda函数, 将每个元素转换为一个文件路径, 并将结果存储在pred中。

```
1 # Search pipeline
2 p_search_pre = (
3 p_embed.map('vec', ('search_res', ops.local.search_tcldb_client(
4 host=HOST, port=PORT, key=PASSWORD, username=USERNAME,
5 collectionName=COLLECTION_NAME, dbName=DB_NAME)
6 .map('search_res', 'pred', lambda x: [str(Path(y[0]).resolve()) for y in x])
7 )
8 p_search = p_search_pre.output('img_path', 'pred')
9
10 # Search for example query image(s)
11 dc = p_search('test/goldfish/*.JPEG')
12 DataCollection(dc).show()
```

调用ops.local.search_tcldb_client函数连接向量数据库, 并搜索与vec最相似的向量, 将搜索结果存储在search_res中。

```
1 = class SearchTCldbClient(PyOperator):
2 def _call_(self, query: ndarray):
3 tcldb_result = self.query_data(
4 query,
5 )
6
7 # result = []
8 # for hit in tcldb_result[0]:
9 # row = []
10 # row.append(hit["path"], hit["score"])
11 # result.append(row)
12 # return result
13
14 def query_data(self, query: ndarray):
15 # 调用 collection 搜索
16 db = self._client_database(self.db_name)
17 coll = db.collection(self.collection_name)
18 vector = query
19 # 使用 distance(query, np.ndarray)
20 # Convert ndarray to list and float32 to float
21 vector = list(map(float, query))
22
23 # search 检索数据 vector 搜索的向量
24 # 返回最相似的数据, 按照相似度从小到大返回最多 top_k 个相似性结果
25 res = coll.find(
26 vector=vector, # 指定搜索向量
27 retrieve_vector=False, # 是否需要返回向量字段, false: 不返回, True: 返回
28 limit=top_k, # 指定 Top k 用 k 值
29 )
30 return res
```

搜索管道 p_search的输出是:

- 1.img_path: 查询图片的路径。
- 2.pred: 查询到的相似图片路径列表。

使用p_search管道对图片test/goldfish/*.JPEG的相似图进行搜索。用Towhee的DataCollection组件显示搜索结果。



DataCollection.show() 函数在显示大量数据时, 为了防止在屏幕上显示过多的数据, 会省略部分数据。

同样的, 如果知道了一张图片的向量, 可以在 DMC 中用向量检索相似的图片信息, 查询到的结果默认按照score由高到低排序, 越大表示相似度越高。



集成 Gradio

觉得上述示例中的代码演示对于非技术用户来说不够友好？

我们可以使用 Gradio 提供的 Web UI, 以更直观、更互动的方式来展示上述的查询和结果。几秒钟内就可以将上述工作流程以 Web UI 的形式呈现出来。这样, 用户可以直接通过上传图片来进行搜索, 在界面上展示出相似的图片。

```

Python | 复制代码
1
2 def search_and_show_images(file_path):
3     # 使用 `file_path` 进行搜索, 返回结果的路径
4     results = p_search(file_path)
5     # 从 `DataQueue` 对象中获取数据
6     data = results.get()
7
8     # 获取结果列表
9     pred = data[1]
10    return pred
11
12 iface = gr.Interface(
13     fn=search_and_show_images,
14     # inputs=gr.inputs.File(type="file"),
15     inputs=gr.inputs.Textbox(default='test/goldfish/*.JPEG'),
16     outputs=gr.Gallery(label="最终的结果图片").style(height='auto', columns=4),
17     title='Tencent vector db 案例: 以图搜图',
18 )
19 iface.launch()

```

出于演示目的, 下面将通过输入图片路径, 查询并展示相似的图片。

search_and_show_images 会返回类似下面的数据, 最终这些图片路径会被展示到 Web UI 上。

```

Shell | 复制代码
1 [
2     '/root/image-search/reverse_image_search/train/cuirass/n03146219_11082.JPEG',
3     '/root/image-search/reverse_image_search/train/loudspeaker/n03691459_40992.JPEG'
4 ]

```

启动项目。用浏览器打开 <http://127.0.0.1:7860>即可看到成果。

输入: test/goldfish/*.JPEG返回的结果都包含鱼。

输入: test/Afghan_hound/n02088094_4261.JPEG返回的结果都包含狗。



总结

对于构建以图搜图、文字搜视频、私域对话机器人等系统, 腾讯云向量数据库由于其卓越的稳定性、性能、易用性和便捷的运维, 都展现出了显著优势。得益于大厂的背书, 腾讯云向量数据库在 AI 领域中已经成为了领军者。

演示代码地址: <https://github.com/smellersoup/image-search>

重生之我是 戏精之王

好未来



社交游戏·《重生之我是戏精之王》

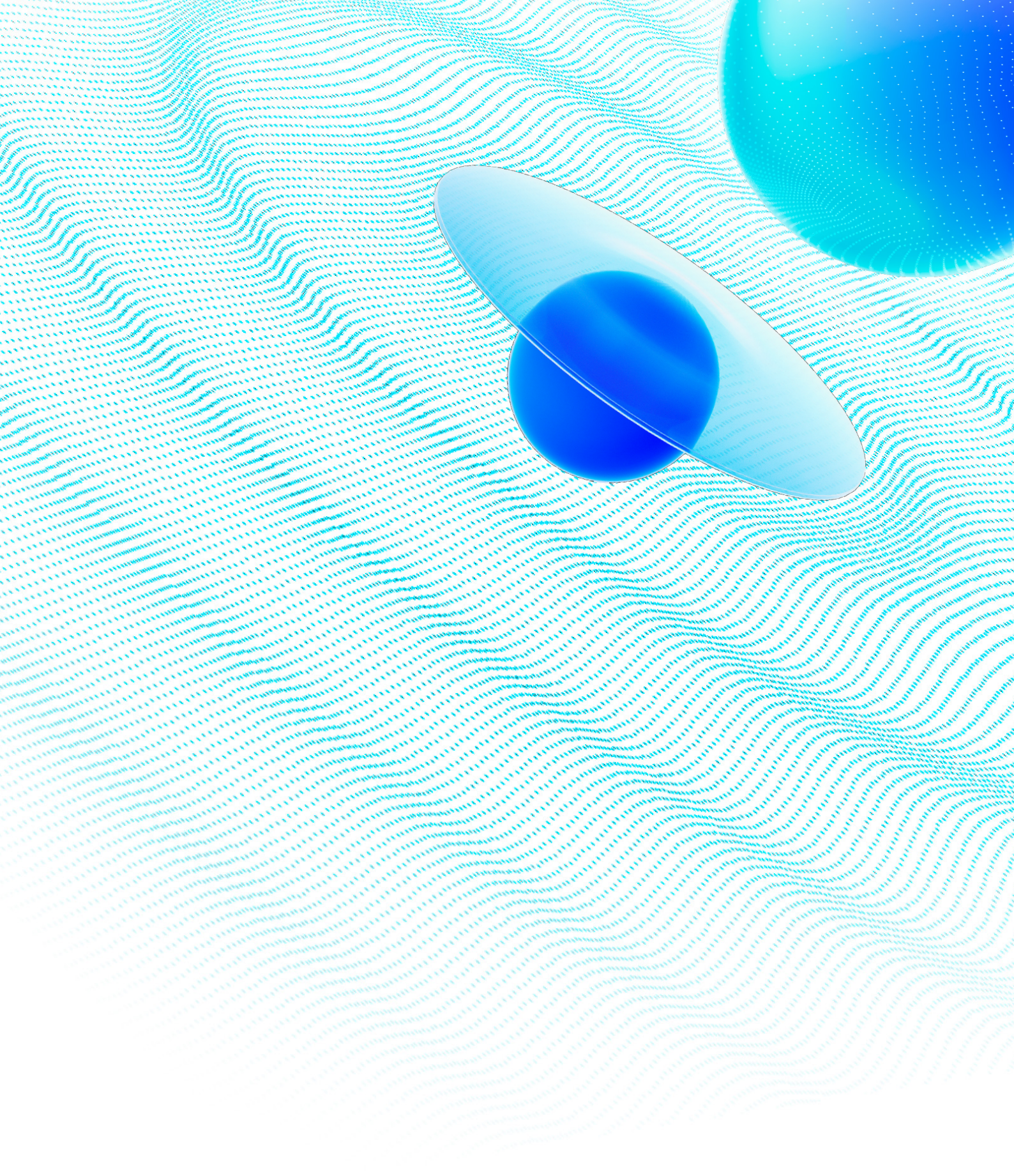
基于AI的生成式“人生剧本”

1.应用向量数据库能力
快速语义搜索

2.应用价值
基于传统文化的知识IP进行演绎可强化其传播及教育意义。

3.基础介绍
玩家可以通过多个维度的内容设定快速生成专属的个人角色，并通过文字及画面的方式演绎其独特的人生轨迹剧本。在此基础上可以与“江湖”中的其他角色进行竞技排名与社交互动。





了解更多产品详情