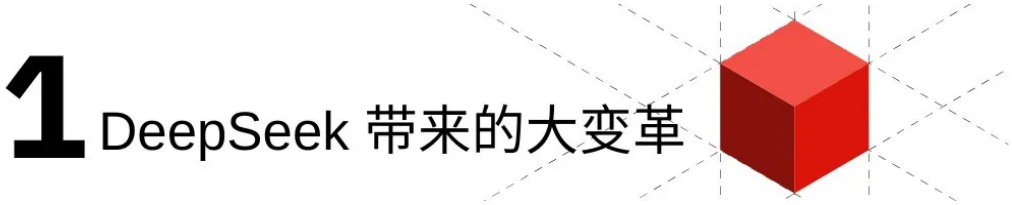
TiDB × AI ：DeepSeek 时代你需要什么样的数据基座

原创 Wish PingCAP 2025年03月20日 19:30 北京

导读

自春节以来，DeepSeek 已成为业界的热门话题。DeepSeek 的出现标志着 AI 领域的又一重要进展。在它之前，OpenAI 已在全球范围内掀起了一波变革浪潮，尤其是在国外，其影响力更为显著。然而，在国内市场，DeepSeek 的影响力更为突出，它不仅推动了 AI 技术的普及，也为相关应用开发提供了新的思路和工具。

随着 AI 技术的不断演进，数据基座的重要性愈发凸显。TiDB 作为一款高性能分布式数据库，在 AI 时代的数据管理中扮演着重要角色。本文将结合 TiDB 的技术特性，探讨在 AI 应用场景下，企业需要什么样的数据基座，以及如何构建高效、灵活且安全的数据基础设施。



DeepSeek 作为一款领先的 AI 模型，以其多样化的模型体系和广泛的应用场景受到关注。其中，V3 和 R1 是两个具有代表性的模型，它们各自针对不同的需求和场景进行了优化。

DeepSeek V3：高性价比的通用模型

DeepSeek V3 是一个以高性价比著称的模型。它被广泛认为是“价格屠夫”，主要原因是其成本远低于同类产品。与 OpenAI 的同类模型相比，V3 的价格低至 1/10 甚至 1/100，这使得它在处理不需要复杂推理的场景时，能够显著降低使用成本。这种特性使得 V3 成为许多企业和开发者在资源受限或预算有限的情况下，进行大规模部署和应用的首选模型。

DeepSeek R1：强大的推理模型

与 V3 不同，DeepSeek R1 是一个专注于推理和思考的高性能模型。它对标 OpenAI 的 GPT-o1 模型，是国内能够轻松访问的最领先的 AI 模型之一。R1 模型在国内的应用场景中表现出色，为用户带来了显著的效率提升和创新体验，甚至被形容为带来了“aha moment”（令人惊叹的时刻）。

在公共服务领域，DeepSeek 的应用经历了快速的发展。早期，DeepSeek V2 版本在小众技术领域有一定知名度，但还没有进入大众视野。随着 V3 和 R1 的爆火，DeepSeek 成为了 App Store 上全球使用率增长最快的 App（没有之一）。用户数量的增加，资源分配逐渐紧张。目前，腾讯元宝和 MiniMax 等平台已经集成了 DeepSeek 模型，为用户提供更稳定的服务。这些平台不仅支持 DeepSeek 的模型，还结合了自身的技术优势，进一步提升了模型的可用性和性能。

对于 AI 应用开发者而言，调用这些模型的 API 是最常见的方式。目前，火山方舟提供的 API 因其稳定性而受到青睐，而官方 API 由于使用人数过多，仍存在一定的稳定性问题。此外，许多领域服务也已经接入了 DeepSeek，进一步拓展了其应用场景。不过作为一个推理模型，R1 虽然有着极佳的效果，但相比自家 V3 模型来说并不便宜。

更重要的，DeepSeek 是一个开源模型，开源带来了几个显著的行业变化。

大规模廉价基础服务

开源使得 DeepSeek 能够成为一种大规模且廉价的基础服务，这一变化会对全行业产生重要影响。在 DeepSeek 出现之前，AI 模型服务市场主要被 OpenAI、Anthropic 等少数几家公司美国 AI 公司垄断，它们的模型很先进，但不开源，服务价格高昂。然而，随着 DeepSeek 的开源，这一局面被打破。如今，包括 AWS、Azure 和字节跳动的火山引擎在内的众多基础设施服务商，都开始提供类似的模型服务，极大地丰富了市场选择。

尽管 DeepSeek 的服务在未来可能不会始终保持极低的价格，但其访问方式正逐渐多样化，成本也在逐步降低。另外根据 DeepSeek 最新发布的成本定价分析，其 R1 模型的理论成本利润率高达 500% 以上，这意味着 DeepSeek 仍有较大的降价潜力。

可定制化

开源不仅降低了成本，还提升了模型的灵活性和适用性。DeepSeek 提供了蒸馏版本，支持在端侧部署，并允许用户通过 fine-tuning（微调）过程对模型进行定制化训练。通过输入特定数据，用户可以进一步优化模型，使其更好地适应自身需求。这种定制化能力使得模型能够更好地服务于特定业务场景，而无需依赖复杂的提示工程。

数据成为核心资产

开源模型的另一大优势是支持私有化部署，这不仅可以满足合规、信息安全等方面的需求，更奠定了私有数据的关键地位。以腾讯元宝提供的 DeepSeek 服务为例，该服务能够搜索微信公众号的文章内容，结合文章内容对用户的提问进行更深入的内容思考，而官方 DeepSeek 无法访问这些数据。这种差异使得腾讯元宝的 DeepSeek 服务针对特定问题（例如时事政治）能达到比官方 DeepSeek 服务更高的回答质量。过去，这类“数据孤岛”被视为是国内互联网生态不够开放的体现，而如今，正是这种私有数据的差异，在开源大模型时代形成了真正的用户体验差别。

我们还可以做出一些大胆的畅享。例如小红书在旅游攻略、美妆等领域积累了大量攻略内容，但这些内容并不公开可获取。如果小红书在未来推出类似于腾讯元宝的服务，那么结合其独有的数据资源优势，可挖掘的潜力非常大。

除了数据带来的独特价值以外，政企部门尤其关注敏感数据的安全合规问题，因此 DeepSeek 一体机正成为一个热门话题。私有化部署不仅保障了数据安全，还为企业提供了定制化和差异化的服务，进一步巩固了数据作为核心资产的地位。

接下来，讲一讲从 OpenAI 开始，PingCAP 在 AI 领域的一些实践历程。其实，我们很多工作都是在持续追随 AI 行业的发展和最新技术，通过实践和应用来学习。所以，接下来大家看到的，其实都是我们在不断学习和成长的过程，并不是说我们从一开始就知道该往哪个方向走。

Chat2Query：SQL Copilot

2023 年初，我们的海外云服务 TiDB Cloud 正式发布了 Chat2Query 功能，能够以 Copilot 的方式让用户通过自然语言直接获取到结果。对于熟悉编程的同学来说，Copilot 的

概念并不陌生，已经大规模应用在了各种代码编辑器中作为自动补全。而 Chat2Query 允许用户以自然语言向数据库提问，系统会将这些自然语言转化为可执行的 SQL 语句，并以可视化的方式返回查询结果。

这一功能经历了多阶段的演进，持续迭代，提升效果。

首先是模型的演进。OpenAI 持续推出新的模型，从最早的 GPT-3 到后续的更多版本，我们通过直接接入最新的模型，显著提升了 Chat2Query 的质量。

除了模型以外，我们还根据行业内最前沿的研究成果，落地了一系列优化措施，以下是我们实践下来有显著效果的几项成果：

1. 提示工程（Prompt Engineering）：通过改进提示词的设计，提升模型对自然语言的理解和响应能力。

2. 应用 RAG 技术：通过 RAG 技术，将表结构、数据关系等信息纳入进来，为模型补充更多领域相关知识，从而提升回答的准确性和丰富性。

3. 引入思维链：在 OpenAI 推出思维链模型之前，我们已经在实践中采用思维链的方式。最知名的“let's think step by step”其实也是思维链的一种。当然思维链并不只是加一句话一种方式，例如可以向 AI 提供一些示例，展示一个任务的处理方式，然后要求它对一个类似的任务进行

处理。这种方式（称为 One Shot）下 AI 的表现会比没有任何样例引导（Zero Shot）时更好。

4. 多步迭代、验证结果：我们让 LLM 生成的是 SQL 语句。很多时候，生成的 SQL 语句可能存在语法错误，这时我们会将错误信息反馈给模型，请其进行修复，给出新的 SQL 语句，再次尝试执行。通过这种反复修正迭代的方式，我们最终可以给到用户一个可以正确执行的 SQL 语句以及其执行结果。

从 TiDB.AI 到 TiDB AutoFlow

PingCAP 开发的第二个 AI 应用是 AutoFlow 项目（原先叫 TiDB.AI）。该项目不仅是知识库解决方案，同样也是 PingCAP 在 OpenAI 技术浪潮下，随着大模型生态不断演进而进行的一系列实践成果。TiDB.AI 驱动了 TiDB 官方文档站的搜索功能，并被应用于 AskTUG 和 Support Bot 等场景。

如今，TiDB.AI 已经从一个单一的知识库应用，进化成为 AutoFlow 框架。AutoFlow 是一套 GraphRAG 框架，不仅提供了类似于 LlamaIndex 的能力，而且还内置语义化的知识图谱构建和召回，以及我们在 AutoFlow 上实践得出的一系列行之有效的领先的 RAG 能力（这些接下来会介绍）。开发者在开发 GraphRAG 方案的应用时（无论是知识库还是其他召回类应用），可以直接使用 AutoFlow 提供的封装好的组件，组合使用，提高开发效率。在项目中引入 AutoFlow 框架很简单，pip install 一下即可。

接下来我会回顾一下 AutoFlow 的演进路线。

在 AutoFlow 的演进过程中，最基础的改进仍然是采用更强的模型。除此之外，我们还进行了以下关键演进，获得了不错的质量提升：

1. 从向量召回进化到 Graph RAG：最初，我们采用基于向量的召回机制，其本质上是语义召回，通过向量相似性来检索相关内容。随后，我们引入了 Graph RAG（基于知识图谱的检索增强生成）。当时，Graph RAG 还仅停留在论文阶段，并未广泛流行。我们较早地实践了论文中提到的方法，并验证了其在提升召回质量方面的显著效果。所谓召回效果，即用户提出问题

后，RAG 需要将一系列“相关”的材料一起提供给大模型，因而如果召回地更好、能给出更相关的材料，那么 LLM 就能有更高的回答质量。

Graph RAG 的核心在于通过知识图谱来检索相关知识，从而提升召回效果。为了更好地理解 Graph RAG 的优势，可以将其与其他召回方式对比。

在传统关键字召回中，搜索引擎会返回与输入文本直接相关的内容。这些内容被提供给大模型，由其根据参考资料和用户提问给出解答。这种方式主要基于文本匹配，召回的是文本上相似的内容。相比之下，向量召回关注的是语义相似性。例如，当提问是“水生动物”时，向量召回可以将“鱼”等相关内容召回，因为它们在语义上是相关的，即使文本并不完全一致。这种语义相关性使得向量召回能够超越单纯的文本匹配，提供更精准的结果。

在引入知识图谱后，召回能力进一步得到增强。例如，“谷歌”和“微软”这两个词在知识图谱中存在多种关联关系：它们都是美国公司、互联网公司，并且都在纳斯达克上市。这些关系构成了一个知识图谱，能够揭示它们之间的内在联系。因此，当用户提问涉及“美国公司”“互联网公司”或“纳斯达克上市”等关键词时，知识图谱可以将谷歌和微软关联起来，即使从文本上看，这两个词并没有哪一部分是相同的。这种基于知识图谱的关联能力超越了简单的文本匹配，解决了语义关系无法通过传统方法解决的问题，从而带来了根本性的变化。

当然，这里只是作为一个让大家容易理解的例子。实际上「谷歌」和「微软」作为公共领域耳熟能详的两个词，在向量上已经是相似的了，这还不能完全体现知识图谱的威力——通过知识

图谱可以对非公共领域的各种概念构建联系，例如张三和王五是某某大学的同学关系。这对于私有数据的知识库来说有至关重要的效果。

2. RAG Workflow：我们开发了名为 stackvm 的编排框架，并用在 AutoFlow 中实现了动态的 RAG 流程，这是 AutoFlow 的一项重要创新。

在传统 RAG 模式下，执行流程是固定的：用户提出问题后，系统按照预设的步骤检索相关文本，并将这些文本传递给大模型以生成回答。这种固定流程不仅缺乏灵活性，还存在不可预测、不可复现以及难以纠正的问题。

相比之下，引入编排功能后的 RAG 流程则更加灵活和动态。系统会首先要求大模型根据用户的问题生成一系列定制化的步骤，例如：第一步从知识图谱中提取相关内容，第二步对这些内容进行进一步的整理和归纳。这些步骤的数量和顺序可以根据问题的复杂程度动态调整，最终由大模型执行。这种动态流程能够适应不同问题的需求，无论是简单问题还是需要整合多个要素的复杂问题。

这种动态调整的能力更接近人类的思考方式：面对简单问题时，可能只需要查阅少量资料；而面对复杂问题时，则需要广泛查阅多种资料并进行归纳整理。动态 RAG 流程的灵活性使其能够显著提升执行效果，更好地满足多样化的需求。

面向 RAG

正如前文提到的 Fine-Tuning，它也反映了大模型数据基座的需求。首先，最基础的需求是，在未来几年内，RAG 仍然是不可或缺的。即便大模型的上下文窗口不断扩大，市场上有更多的开源模型可供选择，或者用户能够在本地进行数据更新，RAG 的必要性依然存在。

原因在于，知识本身是有局限性的。例如，Fine-Tuning 是根据用户的个性化知识对大模型进行定制化的过程，但它并非一种可以随时进行定制化的状态。新的数据会不断涌入，需要大模型进行实时解答。在这种情况下，RAG 依然是必不可少的。

此外，用户通常希望减少模型产生的幻觉现象，并要求模型明确输出数据的来源。当所有内容都被整合到模型内部，并要求其为每一条输出指定信息来源时，只有通过 RAG 才能实现这一目标。这是因为用户通过一组提示词（prompt）告知模型，哪些内容应作为数据源进行归纳。因此，这种需求始终存在。

除此之外，上下文长度限制、性能约束以及数据安全合规等问题也是用户关注的重点。鉴于这些因素，RAG 无疑是一个至关重要的需求。

RAG 发展持续对数据检索提出新的要求

这张图描绘了 RAG 生态的发展历程。最初，RAG 仅依赖于单纯的向量搜索（Vector Search）。随后，向量搜索与关键字检索相结合，这种混合检索方式被证明可以显著提升检索效率。到了大约去年年中，Graph RAG 开始兴起，人们发现引入知识图谱能够有效提升 RAG 的质量。

也许接下来会流行基于 PageRank 进一步提升召回效果？我不好说。我们现在难以预测未来 RAG 的召回技术会发展成什么样，但有一点是明确的：我们不希望每一种召回都需要维护一种新的查询引擎——这不仅增加了复杂度，更关键的是它大大减慢了我们敏捷地试验、应用新技术的速度。

相反，一个能够处理各种查询任务的通用查询引擎是更理想的选择。这也是我们主张支持向量功能的通用数据库优于仅处理向量任务的专用数据库的原因。通用数据库能够更灵活地应对多样化的查询需求，从而展现出更高的实用性和通用性。

实际上，行业发展趋势已经清晰地表明了这一点：仅依靠向量数据库已无法完全实现 Graph RAG 的功能。以 TiDB 为例，虽然它是一个典型的案例，但市场上还有许多其他数据库可供选择。例如，在本地部署中，PostgreSQL 结合 PG Vector 也是一个出色的选择。几乎所有的数据库都在不断加入向量功能，以更好地整合向量检索与全文检索能力。

当用户开发 AI 应用时，有大致这样几个层级的问题：

鉴于 RAG 依然是当前不可或缺的需求，相应的框架选择也较为流行。例如，Dify 是一个功能全面的知识库和编排框架，适用于绝大多数用户。对于少量追求更高质量回答的用户来说，Dify 会不太够用，目前尚未支持 Graph RAG，这类用户可以考虑使用我们的 AutoFlow 框架。不过需要强调的是，Dify 是一个开箱即用的非常易用的界面，而 AutoFlow 虽然功能更强却则具有比较高的使用门槛，所以这两个选择其实面向了不同的群体，用户需要依据自己的实际需求进行选择。

至于数据库部分，那其实不仅限于 TiDB，任何支持向量功能的通用数据库都可以在这个架构中发挥作用。通用是非常重要的一点，意味着它不仅能做向量搜索，也能做到图搜索，和未来更多复杂的其他类型搜索，能够非常好地支撑敏捷开发的需求。

最后，数据无疑是至关重要的。如前文所述，数据是核心资产。模型可以持续迭代，开源模型也在不断更新，私有的数据最终会成为区分不同应用体验的关键因素。

以下内容基于我们观察到的现象以及相关思考，具有一定的预测性。

大规模向量检索

目前，RAG 的应用大多集中在小规模场景中，大规模应用尚未普及。知识库的文本量通常不会太大，百万级别已算较大规模。然而，随着数据成为核心资产，未来会有更多数据被纳入检索范围，大规模向量检索将成为重要的发展方向。

关键字检索

关键字检索与向量检索、图检索相结合，能够显著提升 RAG 的效果。这种多模态检索方式可以更好地满足多样化的查询需求，提供更精准的检索结果。

混合 Ranking

混合 Ranking 是当前 RAG 应用中面临的一个关键问题。当通过向量检索、关键字检索和图检索分别获取结果时，每种方式都会返回一批最匹配的文档。然而，当这些结果合并后，如何确定哪个结果最符合用户需求，是一个亟待解决的问题。目前，通常使用 Reranker 服务例如 JINA 对多个结果进行混合和重新排序。同时，我们也正在探索如何从数据库层面原生地提供这种能力，以简化应用开发流程。

向量嵌入 Pipeline

向量嵌入的简化是当前的一个重要趋势。目前，向量搜索的使用相对复杂，需要用户提供已经经过 embedding 的文本向量（如 768 维或 1536 维的向量）。Embedding 是将文本转化为向量的过程，用户需要将向量存入数据库。然而，如果用户只有文本数据，就需要搭建一个 pipeline 将文本转化为向量，这并非易事。用户不仅需要选择合适的模型，还需要编写 Python 脚本，并使用 GPU 资源来运行。因此，如何简化这一过程成为了一个关键问题。

一些数据库已经给出了答案。例如，Chroma 数据库允许用户直接输入文本，并提供接口以支持向量相似性检索或关键字相似性检索。这种体验极大地简化了用户的操作，用户无需关心底层的向量转换过程。

除了数据库在支持向量应用或 AI 应用方面的功能演进外，数据库与 AI 的融合也在不断探索中。我目前能分享一些较为明显且可观察到的发展方向。

功能重塑

首先，数据库可以借助大模型的知识和预测能力，实现某些功能的根本性变革。以全文搜索为例，使用 Elasticsearch 时，需要为其配置分词器。对于全球化应用，数据内容可能包含多种语言（如中文、日文、韩文等），分词器的配置变得极为复杂。常见的做法是先进行语言检测，然后将数据拆分到不同字段，并分别为其配置中文分词器、日语分词器、韩文分词器等。这一过程繁琐且效率低下。

然而，借助大模型的能力，可以自然地为任何语言的文本生成高质量的分词结果。这将极大地简化分词流程——用户只需提供文本，无需关心文本的语言种类，大模型能够自动完成高效的分词处理。

此外，大模型的预测能力为数据库优化带来了新的思路。例如在数据库迁移场景中，大模型可以批量地将一种数据库的业务 SQL 语句改写为另一种数据库的 SQL 语句，降低需要人工翻译的时间，这是大模型所擅长的领域。另外，作为 SQL 语句，它还是可验证、可执行的，所以我们前边提到的 Chat2Query 中让大模型自助地对 SQL 进行修订、形成更高质量、更正确的 SQL 是一个非常可行的路径。

辅助运维

除了上述提到的应用方向，还有一些已经出现或可以预见的发展趋势，例如在数据库的辅助运维方面。以 Bytebase 为例，这款数据库软件已经集成了基于 OpenAI 模型的 Index Advisor。它并非传统的基于规则的优化工具，而是利用大模型的知识来优化不同类型的数据库。由于 Bytebase 希望支持多种数据库，为每一种数据库招募专家并编写专门的优化规则是不现实的，因此利用大模型进行处理成为了一个高效且可行的解决方案。

此外，传统上可以通过普通机器学习解决的问题，如异常检测和容量预测，在大模型时代不仅能够被解决，而且可以实现更优的效果。

随着 AI 技术的不断发展，TiDB 将继续在数据基座建设中发挥关键作用。未来，大规模向量检索、多模态检索以及混合 Ranking 等技术将成为 RAG 发展的重要方向，而 TiDB 的通用性和高性能将为这些复杂查询需求提供有力支持。同时，向量嵌入 Pipeline 的简化以及数据库功能的重塑，将进一步推动数据库与 AI 的深度融合。TiDB 将持续优化自身技术特性，以更好地满足 AI 应用对数据检索和处理的高要求，助力企业在 AI 时代构建高效、灵活且可靠的数据基础设施，推动技术生态的持续演进。

基于 AutoFlow 快速搭建基于 TiDB 向量搜索的本地知识库问答机器人

💡 点击文末**【阅读原文】**，立即下载试用 TiDB！

阅读原文 修改于2025年03月20日