

# 大模型 LLM

## 🕒 更新时间

Thursday 8th May 2025 16:22:55

## 1 LLM--AGI & AIGC

### 📄 Info

- AGI 是人工智能的终极愿景，追求构建具备人类智慧的通用系统。
- AIGC 是当前人工智能的具体应用形式，突出在内容生成领域的创新与效率提升。
- 两者的主要联系在于技术的共性与互相推动关系：AIGC 为 AGI 的发展提供技术支撑与实验验证，而 AGI 的实现会进一步提升 AIGC 的能力与潜力。

大语言模型（LLM，Large Language Model），也称大型语言模型，是一种旨在理解和生成人类语言的人工智能模型。LLM 通常指包含数千亿（765B）参数的语言模型，它们在海量的文本数据上进行训练，从而获得对语言深层次的理解。

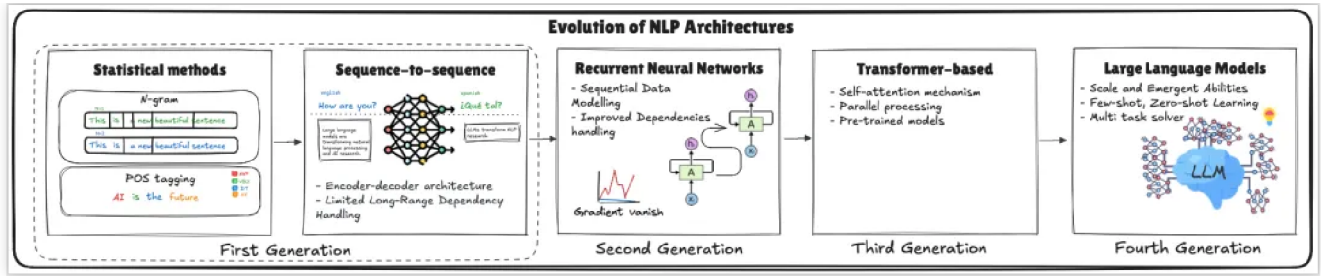
目前，国外的知名 LLM 有 GPT-3.5、GPT-4、PaLM、Claude 和 LLaMA 等，国内的有深度求索、通义千问、智谱清言等。

为了探索性能的极限，许多研究人员开始训练越来越庞大的语言模型。尽管这些大型语言模型与小型语言模型（例如 3.3 亿参数的 BERT 和 15 亿参数的 GPT-2）使用相似的架构和预训练任务，但它们展现出截然不同的能力，尤其在解决复杂任务时表现出了惊人的潜力，这被称为“涌现能力”。以 GPT-3 和 GPT-2 为例，GPT-3 可以通过学习上下文来解决少样本任务，而 GPT-2 在这方面表现较差。

因此，科研界给这些庞大的语言模型起了个名字，称之为“大语言模型（LLM）”。LLM 的一个显著应用是 ChatGPT，它实现了面向对话的 GPT 系列模型的适应。这个对话代理通过结合监督微调和从人类反馈中学习的强化学习（RLHF）的创新训练方法，在人机交互中达到了前所未有的流畅度。

### 1.1 LLM 历史

早期的统计方法，如 N-gram 模型依赖局部上下文，无法捕捉长程依赖。为解决长距离依赖问题，循环神经网络（RNN）被提出，但 RNN 面临梯度消失问题。随后，基于 Transformer 架构的大规模预训练模型（如 BERT、GPT）依靠自注意力机制，成为新一代语言模型的基础，并推动了大语言模型（LLM）的发展。LLM 通过自注意力机制和大规模预训练实现了上下文理解和多任务泛化能力。



## 1.2 NLP 的早期探索

- 20 世纪 50 年代：NLP 研究起步，尝试通过规则和统计方法解析文本。
- 规则和模板硬编码：手动编写规则处理简单任务，但灵活性差、扩展性低。
- 统计语言模型：如 n-gram 模型、隐马尔可夫模型（HMM），通过统计词频和词序预测语言。

## 1.3 深度学习的崛起

- 词嵌入技术：Word2Vec、GloVe 将词映射到低维向量空间，捕捉语义关系。
- 循环神经网络（RNN）：处理序列数据，捕捉长距离依赖，LSTM、GRU 提升性能。
- Transformer 架构：2017 年提出，通过自注意力机制高效处理长距离依赖，成为 LLM 的核心。

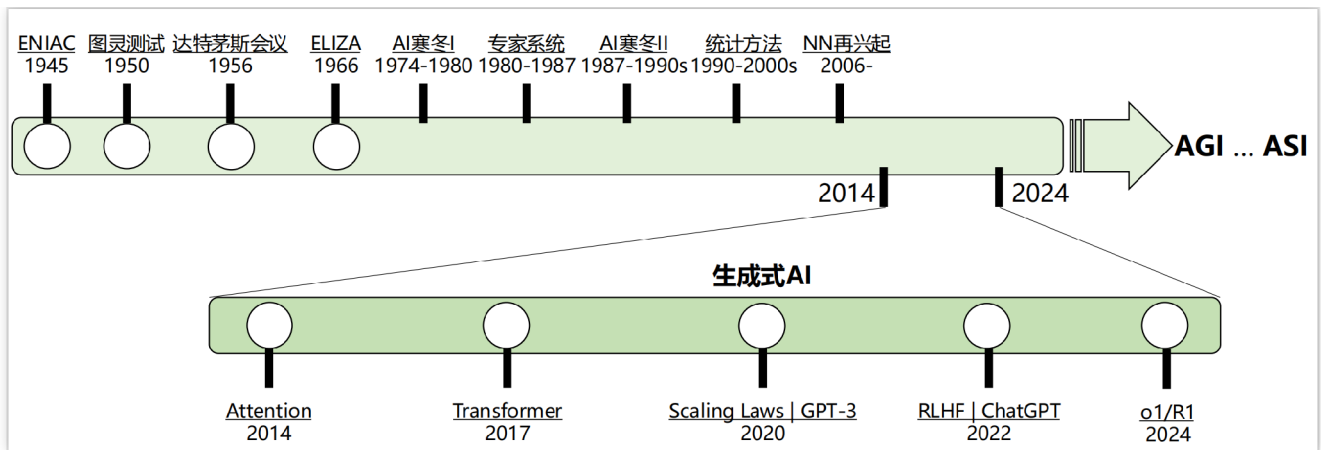
## 1.4 预训练与微调范式

- BERT：双向编码器，通过掩码语言模型（MLM）学习上下文表示。
- GPT 系列：自回归解码器，通过预测下一个词学习语言生成。
- 预训练+微调：先在大规模数据上预训练，再在特定任务上微调，显著提升性能。

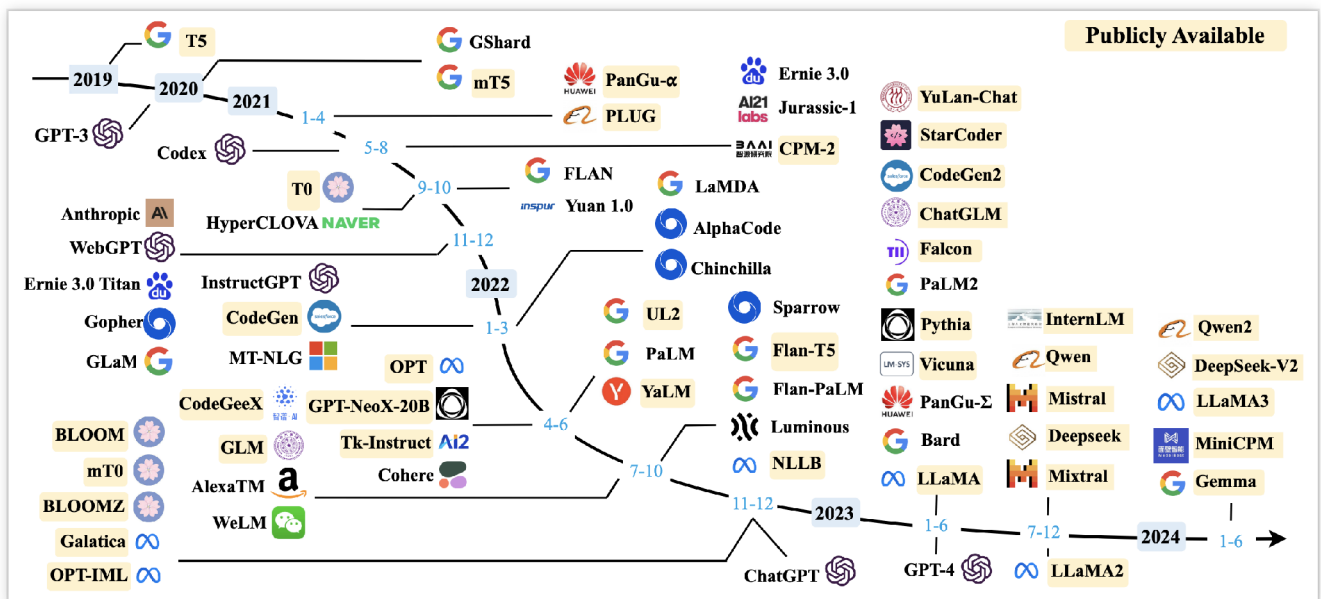
## 1.5 生成式 AI

使用生成式模型生成各类数据（语言、语音、图片、视频等）

- Attention：数据依存关系建模
- Transformer：数据生成的统一架构
- Scaling Laws：数据学习、生成的扩展法则
- RLHF：生成与人类价值对齐的数据
- o1/R1：生成式求解问题——生成问题求解的过程和答案（推理）



## 2 LLM种类



- 语言大模型（NLP）：是指在自然语言处理（Natural Language Processing, NLP）领域中的一类大模型，通常用于处理文本数据和理解自然语言。这类大模型的主要特点是它们在大规模语料库上进行了训练，以学习自然语言的各种语法、语义和语境规则。例如：GPT 系列（OpenAI）、Bard（Google）、文心一言（百度）。
- 视觉大模型（CV）：是指在计算机视觉（Computer Vision, CV）领域中使用的大模型，通常用于图像处理和分析。这类模型通过在大规模图像数据上进行训练，可以实现各种视觉任务，如图像分类、目标检测、图像分割、姿态估计、人脸识别等。例如：ViT 系列（Google）、文心 UFO、华为盘古 CV、InternLM（商汤）。
- 多模态大模型：是指能够处理多种不同类型数据的大模型，例如文本、图像、音频等多模态数据。这类模型结合了 NLP 和 CV 的能力，以实现对多模态信息的综合理解和分析，从而能够更全面地理解 and 处理复杂的数据。例如：DingoDB 多模向量数据库（九章云极 DataCanvas）、DALL-E（OpenAI）、悟空画画（华为）、midjourney。

### 2.1 框架

从大型语言模型（LLM）框架的角度来看，可以将LLM主要分为以下几种类型，其中MoE和Dense是两种重要的架构类型：

### Info

Dense架构是LLM的基础，结构简单，性能稳定，但计算资源需求高。

MoE架构通过引入专家机制，实现了参数量和计算量的解耦，提高了计算效率和模型扩展能力，尤其适合处理大规模数据和复杂任务。

## 2.1.1 Dense Transformer 架构

核心特点：这是最早被广泛应用于大型语言模型的经典架构。它采用全连接的方式构建每一层的参数，意味着每个神经元都能接收到前一层所有神经元的输出。

- 原理：基于原始的Transformer架构，包含编码器-解码器层的基本结构。通过增加模型规模（参数量），可以直接提升输出质量，这种关系相对线性和可预测。
- 优势：
  - 结构简单直观。
  - 模型质量可预期，通过增加参数量可直接提升性能。
  - 架构成熟可靠，有大量的实践经验可参考。
- 局限性：参数密集，计算资源需求高，尤其在处理大规模数据集和复杂任务时，计算开销急剧增加<sup>1</sup>。由于多头注意力机制的二次方复杂度，扩展计算成本较高。

## 2.1.2 MoE (Mixture of Experts) 架构

- 核心特点：MoE是一种创新的Transformer架构变体，旨在通过引入“专家”机制来优化资源利用，降低计算开销。与Dense架构不同，MoE模型不采用全连接方式，而是将每个神经元分配给不同的“专家”，并根据输入数据特点选择性激活部分专家。
- 原理：将一个大型网络分解为多个小型、专门化的子网络（即专家），并通过一个门控网络（Router）决定哪些专家应该被激活来处理特定的输入。门控网络根据输入特征为不同专家分配权重，实际操作中通常选择概率最高的Top-k个专家进行计算。
- 优势：
  - 可以在远少于Dense模型所需的计算资源下进行有效预训练，相同计算预算下可显著扩大模型或数据集规模。
  - 训练速度更快，效果更好。
  - 相同参数量下，推理成本低，推理速度更快。
  - 扩展能力强，允许模型在保持计算成本不变的情况下增加参数数量，可扩展到非常大的模型规模。
  - 多任务学习能力强。
- 局限性：专家选择依赖于输入数据特征，如果路由机制设计不当，可能导致某些专家未能被充分激活，影响整体性能<sup>1</sup>。如何从现有LLM的FFNs中有效地构建专家以及如何在可接受的计算成本下提高MoE模型性能是主要挑战。

- 发展：MoE思想起源于1991年。2017年Google将其应用于LSTM。进入Transformer时代后，Google在2020年将MoE应用于encoder-decoder结构的Transformer模型（GShard）。2021年，Google推出Switch Transformer，进一步简化了路由策略，训练了1.6万亿参数的MoE模型。除了Google，国内的DeepSeek团队也开源了MoE大模型DeepSeekMoE(V2)。

## 3 LLM基础概念

### 3.1 Prompt

Prompt 最初是 NLP（自然语言处理）研究者为下游任务设计出来的一种任务专属的输入模板，类似于一种任务（例如：分类，聚类等）会对应一种 Prompt。在 ChatGPT 推出并获得大量应用之后，Prompt 开始被推广为给大模型的所有输入。即，我们每一次访问大模型的输入为一个 Prompt，而大模型给我们的返回结果则被称为 Completion。

在使用LLM API 时，可以设置两种 Prompt：

- SystemPrompt: 该种 Prompt 内容会在整个会话过程中持久地影响模型的回复，且相比于普通 Prompt 具有更高的重要性
- User Prompt: 这更偏向于我们平时提到的 Prompt，即需要模型做出回复的输入

### 3.2 Temperature

LLM 生成是具有随机性的，在模型的顶层通过选取不同预测概率的预测结果来生成最后的结果。我们一般可以通过控制 temperature 参数来控制 LLM 生成结果的随机性与创造性。Temperature 一般取值在 0~1 之间，当取值较低接近 0 时，预测的随机性会较低，产生更保守、可预测的文本，不太可能生成意想不到或不寻常的词。当取值较高接近 1 时，预测的随机性会较高，所有词被选择的可能性更大，会产生更有创意、多样化的文本，更有可能生成不寻常或意想不到的词。

对于不同的问题与应用场景，我们可能需要设置不同的 temperature。例如，搭建的个人知识库助手项目中，我们一般将 temperature 设置为 0，从而保证助手对知识库内容的稳定使用，规避错误内容、模型幻觉；在产品智能客服、科研论文写作等场景中，我们同样更需要稳定性而不是创造性；但在个性化 AI、创意营销文案生成等场景中，我们就更需要创意性，从而更倾向于将 temperature 设置为较高的值。

### 3.3 思维链 (Chain-of-Thought, CoT)

思维链方法的核心思想是将思考的过程及其相关的观念和想法串联起来，形成一个连续的思维链条。这种链条可以由线性或非线性的思维过程构成，从而帮助模型不断延伸和扩展思考。相比于之前传统的上下文学习（即通过  $x_1, y_1, x_2, y_2, \dots, x_{test}$  作为输入来让大模型补全输出  $y_{test}$ ），思维链多了中间的推导提示。



### 3.4 LLM 基本训练方法



### 3.5 预训练 (Pretraining)

LLM 训练通常采用大规模无监督学习，即：

1. 从互联网上收集大量文本数据，如书籍、新闻、社交媒体等。
2. 让模型学习词语之间的概率分布，理解句子结构。
3. 训练目标是最小化预测误差，使其能更好地完成语言任务。

### 3.6 后训练 (Post-training)

- **监督微调 (Supervised Fine-Tuning, SFT):** 使用带有标注的特定任务数据对模型进行训练。
- **指令微调 (Instruction Tuning):** 使用指令格式的数据集对模型进行微调，使其能够理解并遵循人类指令。
- **基于人类反馈的强化学习 (RLHF):** 通过人类对模型输出的偏好反馈，训练一个奖励模型，然后利用强化学习算法调整大模型的行为，使其生成更符合人类偏好的输出。
- **直接偏好优化 (DPO):** 一种简化了RLHF流程的方法，直接使用人类偏好数据训练模型，无需单独训练奖励模型。

### 3.7 强化学习 (Reinforcement Learning, RL)

采用强化学习 (RL) 方法进行优化，主要通过人类反馈强化学习 (RLHF, Reinforcement Learning from Human Feedback)：

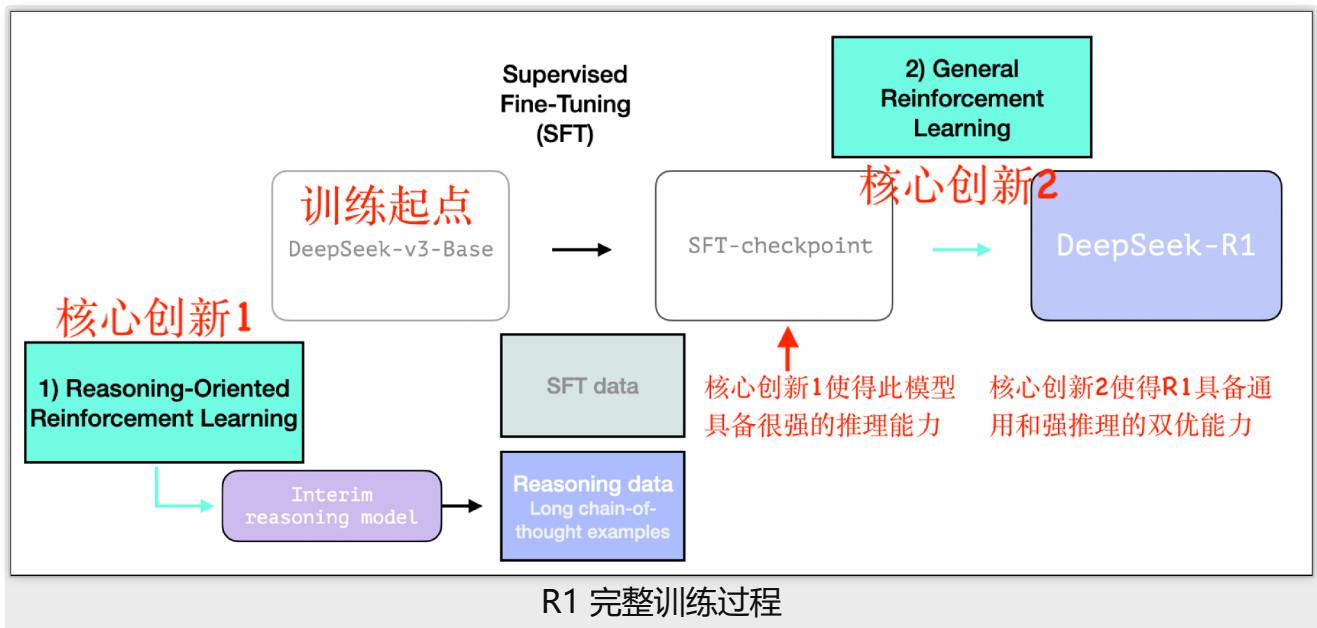
1. 人类标注者提供高质量回答。

2. 模型学习人类评分标准，提高输出质量。
3. 强化训练，使得生成的文本更符合人类偏好。

### DeepSeek-R1 完整训练过程：

DeepSeek-R1 主要亮点在于出色的数学和逻辑推理能力，区别于一般的通用 AI 模型。其训练方式结合了强化学习（RL）与监督微调（SFT），创造了一种高效训练，高推理能力 AI 模型的方法。

整个训练过程分为核心两阶段，第一步训练基于 DeepSeek-V3 论文中的基础模型（而非最终版本），并经历了 SFT 和基于纯强化学习调优 + 通用性偏好调整，如下图






训练起点。DeepSeek-R1 的训练起点是 DeepSeek-v3-Base，作为基础模型进行训练，为后续的推理优化奠定基础。

## 4 LLM应用

### 4.1 API 调用

#### 4.1.1 API 提供平台

LLM API 的提供平台分为

- 模型提供商: 这是最直接和主要的途径。许多领先的AI公司都提供其大模型的API服务，你通常需要在他们的官方网站上注册账号，创建API密钥，然后就可以通过他们的API接口调用模型。
  -  [深度求索](#)
  -  [月之暗面](#)
  -  [智谱清言](#)

-  [OpenAI](#)
-  [Anthropic](#)
-  [Google](#),
-  [Grok](#)
- 第三方API聚合平台/代理服务: 它们聚合了多家大模型的API, 用户可以通过一个统一的接口调用不同提供商的模型。
  -  [硅基流动](#)
  -  [火山引擎](#)
  -  [阿里云百炼](#)
  -  [百度智能云](#)
  -  [Openrouter](#)
- 开源模型和本地部署: 对于一些开源的大模型 (如Meta的Llama系列、Mistral AI的模型、Hugging Face上的一些模型等), 如果你具备相应的技术能力和计算资源, 可以选择将模型下载到本地或私有服务器上部署。
  -  [Hugging Face](#)
  -  [魔搭社区](#)
  -  [ollama](#)

### 4.1.2 API 调用工具

### 4.1.3 Agent 产品


-  [Manus](#)
-  [FoundationAgents/OpenManus](#)
-  [AutoGLM 沉思](#)

### 4.1.4 AI 代码开发工具





1. AI 代码编辑器:
  -  [Cursor](#) AI 代码编辑器
  -  [Windsurf](#)
  -  [Trae](#) 字节 AI TDE
2. vscode 插件:
  - GitHub Copilot
  - Cline
  - Augment
  - 通义灵码



### 3. 其他:




-  [deepwiki](#) github 仓库详细解释网页 (wiki) 生成工具

## 4.1.5 桌面工具

-  [CherryStudio](#)
-  [Chatbox AI](#)
-  [flowith 2.0](#)
-  [ima.copilot](#)

## 4.1.6 AI 搜索引擎




### 1. AI 搜索

-  [秘塔搜索AI](#)
-  [BoCha 博查AI搜索](#)
-  [天工AI搜索](#)

### 2. 搜索服务商:

-  [Epsilon AI](#) 学术搜索
-  [tavily AI](#) AI 搜索 API
- [brave search](#) AI 搜索 API
-  [Exa](#) 硅谷的 AI 搜索 API

## 4.1.7 文档处理工具

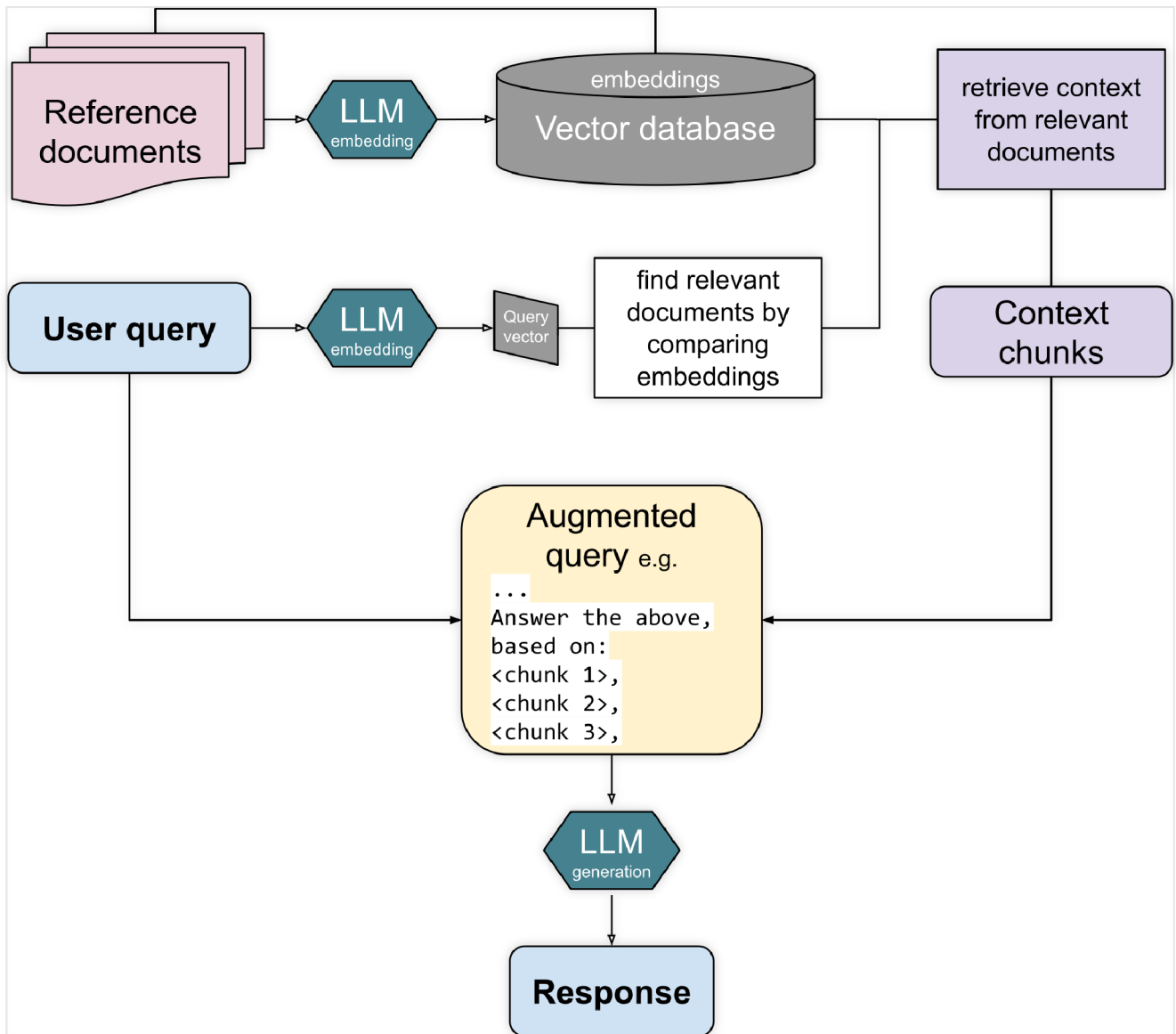
-  [info.ai PDF阅读](#) PDF 阅读
-  [NotebookLM](#)
-  [Humata](#)
- [Paper Visualizer](#) 学术论文可视化的在线工具, 支持多种来源的论文 (arXiv 和 PDF)

## 4.2 RAG

### Info

RAG (Retrieval Augmented Generation, 检索增强生成) 是目前 LLM 应用落地的重要方向, 主要的应用场景是企业客服系统和搜索结果结构化展示 (代表作是 Perplexity 和秘塔)。RAG 对数据的规范程度要求比较高, 数据越规范, 查询效果越好, 结合树形结构或知识图谱结构的数据, RAG 可以实现更好的效果。

开源 RAG 框架推荐: Cohere; Cognita



## 4.3 Agent

### Info


Agent翻译成中文是智能体的意思，是AGI的前奏。现阶段的Agent只能算**工作流**，什么时候Agent能根据用户要求直接创建好Agent，才算是真正的智能体。

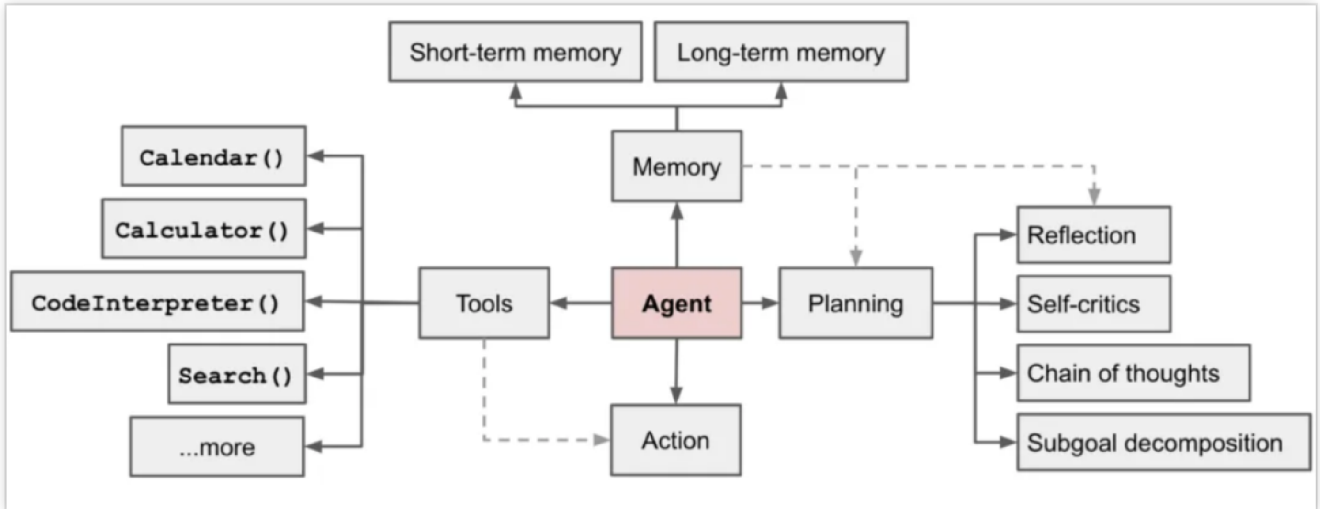
Agent 的研究可以追溯到 20 世纪 50 年代，当时被称为“智能体”或“自主体”。随着人工智能技术的发展，Agent 的概念和应用也越来越广泛。

Agent 具有以下核心概念：

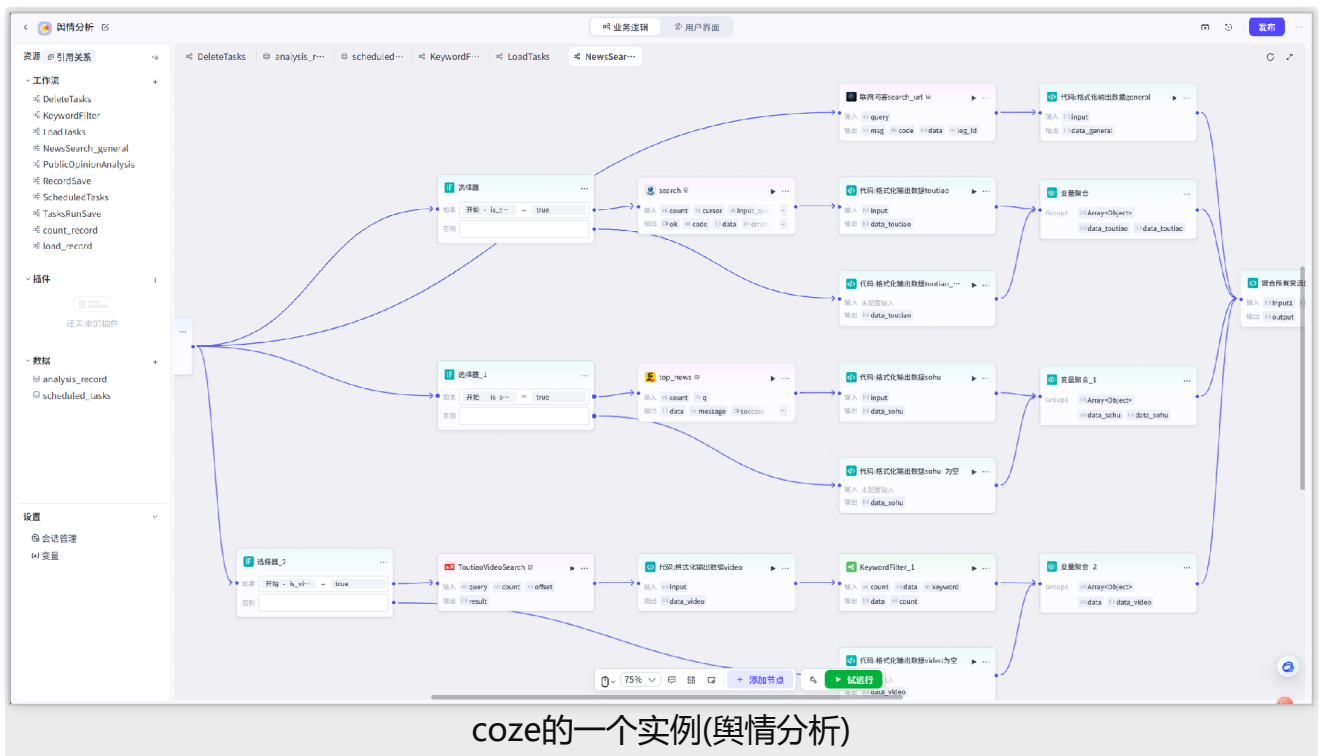
- 感知 (Perception)：Agent 通过传感器感知外部环境，获取信息。
- 决策 (Decision Making)：Agent 根据感知到的信息进行决策，确定下一步行动。
- 执行 (Execution)：Agent 根据决策执行相应的动作，影响外部环境。


- 学习 (Learning) : Agent 通过与环境的交互, 不断学习和优化自己的决策和执行策略。

AI Agent的在大模型时代由OpenAI团队重新定义。 OpenAI认为, AI Agent就是由大模型驱动, 由规划能力组件 (Planning)、记忆组件 (Memory)、工具组件 (Tools)、行为组件 (Action) 等组件所组成的“智能助手”。



目前好用的Agent平台是Coze\Diffy\n8n



还有看起来更加"科幻"Agent产品形式-- Manus, 本质是一个多 Agent 系统, 其开源版本 

[OpenManus](#)

[FoundationAgents/OpenManus | DeepWiki](#)

[OpenManus Hackathon项目投票](#)

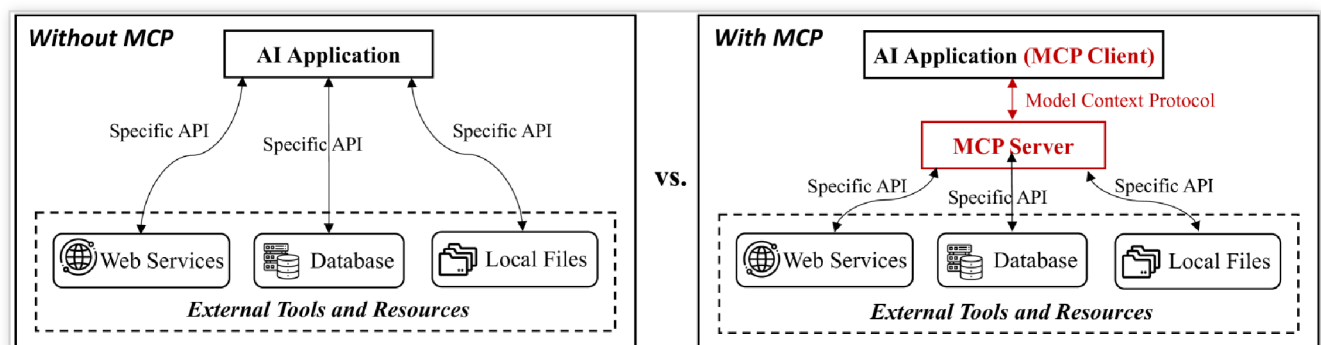
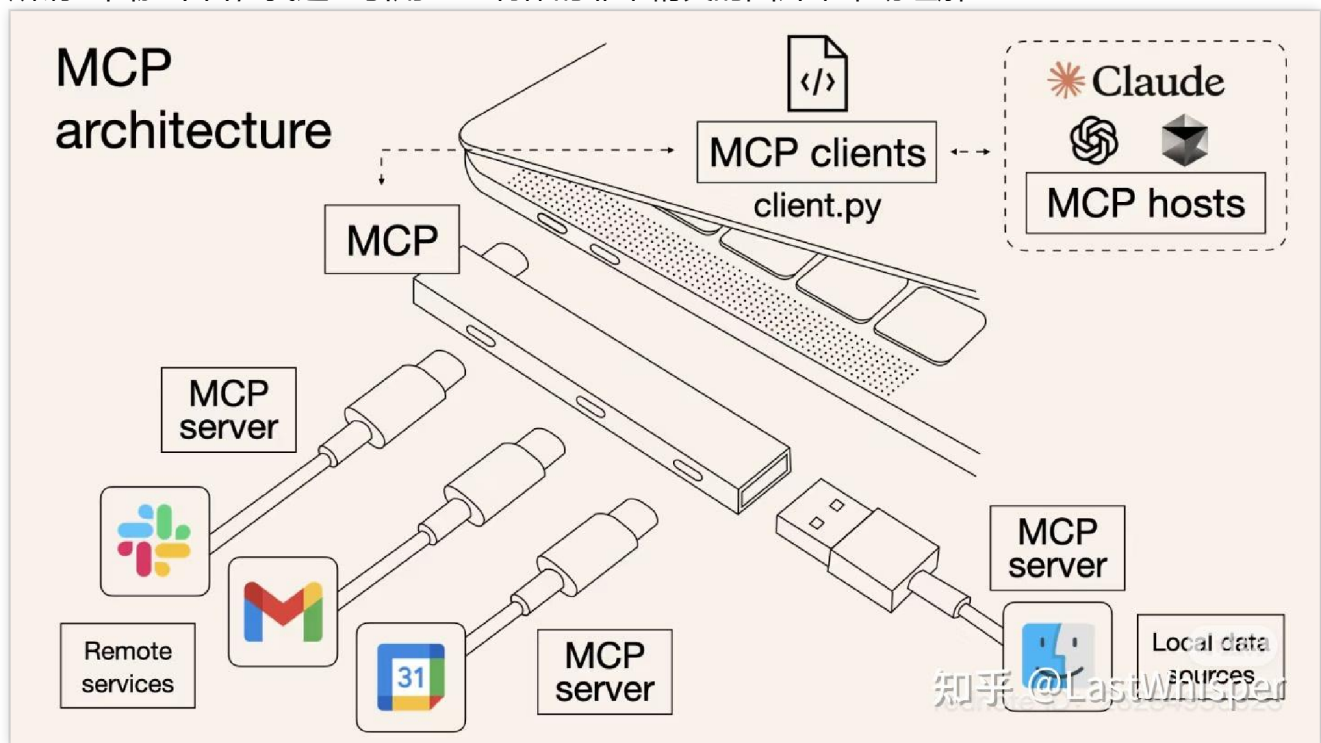
## 4.4 MCP(大模型上下文协议)

- [一文看懂：MCP\(大模型上下文协议\) - 知乎](#)
- [MCP官方](#)
- [Anthropic技术文档](#)
- [\[2503.23278\] Model Context Protocol \(MCP\): Landscape, Security Threats, and Future Research Directions](#)

MCP 起源于 2024 年 11 月 25 日 [Anthropic](#) 发布的文章：[Introducing the Model Context Protocol](#)。

MCP (Model Context Protocol, 模型上下文协议) 定义了应用程序和 AI 模型之间交换上下文信息的方式。这使得开发者能够以一致的方式将各种数据源、工具和功能连接到 AI 模型（一个中间协议层），就像 USB-C 让不同设备能够通过相同的接口连接一样。MCP 的目标是创建一个通用标准，使 AI 应用程序的开发和集成变得更加简单和统一。

所谓一图胜千言，我这里引用一些制作的非常精良的图片来帮助理解



## 1. MCP 服务收集网址 (As of March 27, 2025):

Collection	Author	Mode	Servers	URL
MCP.so	mcp.so	Website	4774	<a href="https://mcp.so">mcp.so</a>
Glama	glama.ai	Website	3356	<a href="https://glama.ai">glama.ai</a>
PulseMCP	Antanavicius et al.	Website	3164	<a href="https://pulsemcp.com">pulsemcp.com</a>
Smithery	Henry Mao	Website	2942	<a href="https://smithery.ai">smithery.ai</a>
Dockmaster	mcp-dockmaster	Desktop App	517	<a href="https://mcp-dockmaster.com">mcp-dockmaster.com</a>
Official Collection	Anthropic	GitHub Repo	320	<a href="https://modelcontextprotocol/servers">modelcontextprotocol/servers</a>
AiMCP	Hekmon	Website	313	<a href="https://aimcp.info">aimcp.info</a>
MCP.run	mcp.run	Website	114	<a href="https://mcp.run">mcp.run</a>
Awesome MCP Servers	Stephen Akinyemi	GitHub Repo	88	<a href="https://appcypher/mcp-servers">appcypher/mcp-servers</a>
mcp-get registry	Michael Latman	Website	59	<a href="https://mcp-get.com">mcp-get.com</a>
Awesome MCP Servers	wong2	Website	34	<a href="https://mcpservers.org">mcpservers.org</a>
OpenTools	opentoolsteam	Website	25	<a href="https://opentools.com">opentools.com</a>
Toolbase	gching	Desktop App	24	<a href="https://gettoolbase.ai">gettoolbase.ai</a>
make inference	mkinf	Website	20	<a href="https://mkinf.io">mkinf.io</a>
Awesome Crypto MCP Servers	Luke Fan	GitHub Repo	13	<a href="https://badkk/crypto-mcp-servers">badkk/crypto-mcp-servers</a>

## 2. 在线 MCP 服务

- [阿里云百炼](#)
- [魔搭社区MCP](#)

## 5 资源

Quote

[WangRongsheng/awesome-LLM-resources: 🌟 全世界最好的LLM资料总结 \(Agent框架、辅助编程、数据处理、模型训练、模型推理、o1 模型、MCP、小语言模型、视觉语言模型\) | Summary of the world's best LLM resources.](#)