第 1 章

智能体系统概述

智能体（Agent）技术的崛起象征着人工智能的深化发展与广泛应用。作为一种具备自主感知、决策、执行能力的系统，智能体正在引发多个行业的技术革命。小到语言翻译、文本改写，大到复杂的多智能体协同的任务处理，多样化的智能体已经成为产业智能化的主要推动力。大语言模型的集成赋予了智能体语言理解、任务规划和代码生成等多种能力，同时通过多轮对话和语义理解，实现了个性化处理、动态响应和任务拆解与优化。

本章将循序渐进地探讨智能体的基础概念，包括智能体的定义、核心构成和架构、智能体的等级划分、多智能体的介绍，以及智能体的实验环境搭建等内容。

1.1 什么是智能体

本节将详细介绍智能体的相关概念，确保本书有一致的名词定义，后续所有对智能体的介绍都基于本节的概念定义来展开。

1.1.1 智能体初识

随着AI的发展，特别是以ChatGPT（见图1-1）为首的大模型表现出超强的文本理解能力和生成能力，新一轮生产力的变革正在以一种前所未有的速度席卷各行各业，其中“智能体”这个概念正变得日益重要。智能体的技术演变经历了从简单的反应式智能体到复杂的自主智能体的发展，它们可以基于环境输入做出反应，也可以通过自我学习不断优化自己的性能和智能水平。它们不仅能够提高效率、降低成本，还能提供更加个性化和智能化的服务，其应用范围非常广泛，包括但不限于客户服务、金融交易、医疗诊断、个性化推荐、智能家居控制等多个领域。

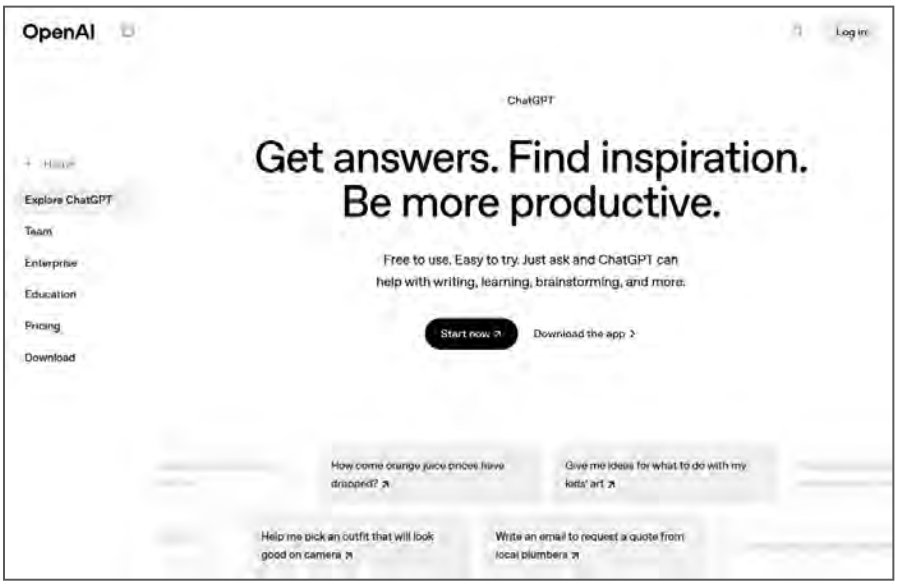

图 1-1　OpenAI 发布的 ChatGPT 官网界面

想象一下，如果你有一个智能体小助手，它不像以前那种只会按指令做事的简单机器人，而是像个小管家一样，能帮你处理各种事。比如，你在家里想放松一下，智能体就能像《钢铁侠》里的贾维斯一样（见图1-2），自动调节室内温度，播放你喜欢的音乐，甚至根据你的心情推荐一部电影。再比如，你在网上购物，有个智能体在后台分析你的喜好，然后给你推荐最适合的商品，就像有个私人购物顾问一样。又或者，你在银行办理业务，智能体能快速帮你完成复杂的金融交易，出错率低到几乎可以忽略不计，这效率比排队等人工服务高多了。

图 1-2　《钢铁侠》中的贾维斯

1.1.2　智能体的概念

AI Agent即人工智能代理，又叫智能体，或者简称代理，是一种能够模拟人类行为和决策的计算实体，它可以感知环境、自主决策、执行动作，以实现特定的目标和任务。它可以利用大语言模型（Large Language Model，LLM）来理解和处理信息，通过提示词（Prompt）来优化交互效果，借助各种工具（Tool）来增强自身能力，运用检索增强生成（Retrieval Augmented Generation，RAG）来获取更丰富的先验知识。例如，一个电信营业厅的客服智能体，当客户问“如何查询流量”的问题时，它能够利用大模型理解问题的含义和客户想表达的意图，比如是要“查询流量”还是“话费充值”，再根据提示词优化回答策略，使用流量查询的API工具查找流量相关数据，通过RAG等外部知识库确保回答的全面性和准确性，最终为客户提供满意的答案。

虽然目前业界对AI Agent还没有一个非常清晰的定义，但是研究者们对它的理解已经逐渐达成共识。业界已基本认同OpenAI的前华人研究员Lilian Weng对AI Agent的定义，即AI Agent是一个具备记忆、能够感知其环境、在给定目标下自主做出决策并采取行动的系统。这些决策是基于它对环境的理解而做出的，比如它的目的地在哪，它的位置在哪，周围有哪些障碍等；它还可以“学习”，即通过与环境交互，不断改进自己的决策过程。AI Agent的本质是那些能够感知环境、理解环境，然后做出决策、采取行动，并不断学习和进化的AI系统。

“智能体”这个概念在有些场景下表示机器人、无人机等智能体设备，为了确保术语的一致性，本书后续介绍的所有“智能体”都指代的是基于大模型的智能体，即AI Agent。

Lilian Weng在*LLM Powered Autonomous Agents*这篇博客中对智能体的定义的大体框架如图1-3所示。

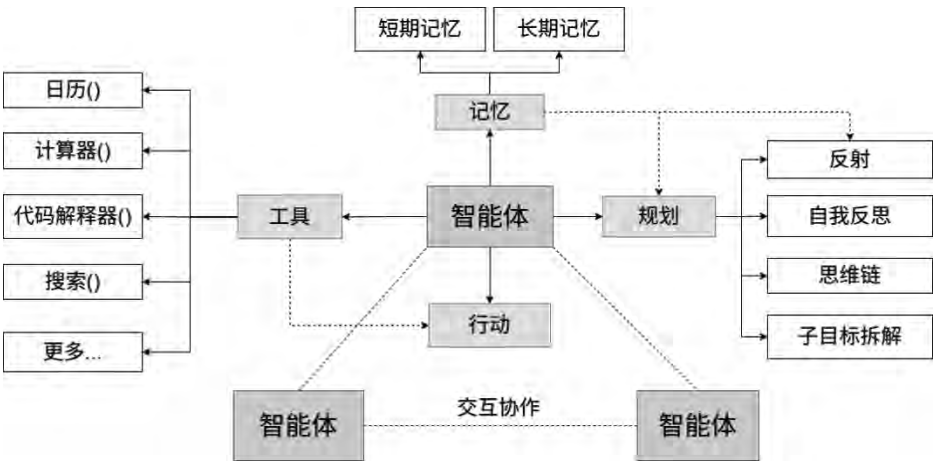

图 1-3　智能体的架构图

智能体的构建模块包括规划（Planning）、记忆（Memory）、工具（Tool）、行动（Action）等几大模块。大语言模型作为其核心控制器，辅以规划技能、记忆能力和使用工具的能力：规划技

能包括子目标拆解、反射和反思等思维模式；记忆能力则包括短期记忆和长期记忆，后者为智能体提供了长期存储和信息召回的能力；使用工具的能力则指智能体通过学会调用外部API来获取额外信息，包括网络信息检索、代码执行能力、访问专有信息源等。

AI的未来应用前景被看好，但也面临着挑战，比如如何确保AI模型不会产生“幻觉”或者提供不准确的信息，以及如何保护用户的私人信息。如果大模型能够克服这些问题，智能体可能会带领我们进入新时代。

1.2　智能体的核心组件概述

由1.1.2节的智能体架构图可知，一个设计良好的智能体架构包括工具、记忆、规划和行动四大部分。除了这四大部分外，大语言模型、检索增强生成和提示词也是构成智能体的重要组成部分。本节将综合性地介绍几个核心模块，在本书后续的章节中会逐步详细地展开各个模块的解释和技术细节。

1.2.1　大语言模型

大语言模型简称大模型，是近几年人工智能领域的一个重要发展方向，主要用于自然语言处理（Natural Language Processing，NLP）任务，也是基于大语言模型的智能体的大脑和灵魂所在。大模型的出现标志着AI从专用模型向通用模型的转变，具有广泛的应用前景。其核心原理是基于深度学习的Transformer架构，使用自回归模型（如GPT）或自编码模型（如BERT）来预测文本中的下一个单词或掩盖的词语，通过大量的文本数据进行预训练，学习训练数据中的上下文关系，捕捉语言的结构和语义，从而生成流畅且符合语法的文本。大模型的发展受到多个领域的推动，包括神经网络、自然语言处理和计算能力的提升。早期的NLP模型主要依赖于统计方法或浅层神经网络，这些方法在处理小规模任务时效果较好，但无法处理复杂的语言任务；随着模型的参数规模越来越大，模型效果也越来越好，使得模型能够理解复杂的语言模式，表现出了“涌现”的能力。

大模型具有大规模参数、多任务学习、零样本学习等特点。大规模参数赋予大模型超凡的表达能力和泛化能力，其参数量级往往高达数十亿乃至千亿；大模型卓越的多任务学习能力，使其能够在同一模型中轻松应对文本生成、翻译、摘要、问答等多种自然语言处理任务；更为独特的是，大模型具备零样本学习能力，得益于庞大的训练数据集和先进的自注意力机制，它能够在没有特定任务训练数据的情况下适应新任务，能灵活应对新的挑战。

大模型的优势在于广泛的应用适应性，能够在写作、辅助编程、对话系统等多个领域大放异彩；同时，它生成的文本不仅流畅、连贯，还能在特定上下文中保持高度一致性；此外，大模型的可扩展性也值得称赞，通过微调或多阶段训练，能够轻松适应新的任务和领域。然而，大模型也存在一些不足之处：首先在训练和运行过程中对计算资源的需求极大，尤其是对内存和算力的要求非常高；其次，由于训练数据的偏差，大模型可能会产生带有偏见的内容，甚至出现不真实的事实信

息，即所谓的“幻觉”；最后，大模型的决策过程比较复杂，缺乏足够的解释性，这在处理有争议性内容时尤为明显。

未来，大模型将朝着更高效、更安全、更具解释性的方向发展。研究人员正在探索减少计算资源消耗的方法，同时提高模型的透明度和公平性。多模态学习、多语言支持和领域自适应将成为未来大模型发展的重要趋势。

1.2.2　工具模块

尽管大模型具有强大的语言生成能力，但它们在处理需要精确计算、实时信息或特定领域的知识时表现不佳。为了解决这些问题，研究者们开发了能够集成外部工具的模块。

尽管智能体和大模型都能通过设计良好的提示词来激发模型性能，使用户得到效果更好的答案，但智能体和大模型的一大区别在于智能体能够使用外部工具拓展其能力边界。懂得使用工具是人类最显著和最独特的地方，同样的道理，为大模型配备外部工具，可以让大模型完成原本无法完成的工作。

在实际应用中，使用比较多的有这三大类工具：代码执行、API集成、搜索整合。

- 代码执行：智能体可以直接在对话中调用脚本代码进行数据处理或科学计算。例如，用户可以要求智能体计算一组数据的平均值或者生成一张数据图表。
- API集成：许多智能体可以通过API与外部系统进行集成。例如，旅行助手智能体可以访问航班预订系统，帮助用户查找和预订航班；或通过API与天气服务系统集成，提供实时的天气信息。
- 搜索整合：智能体可以调用搜索引擎接口来获取实时信息。例如，当用户询问当前的股票价格或者新闻时，智能体会执行搜索任务，并汇总相关信息供用户参考。

工具模块具有扩展性好、模块化设计、高效调度等技术特点。其中，扩展性好是指允许智能体在运行时动态扩展新工具，通过调用新的工具应对新的任务需求；模块化设计是指工具模块采用了模块化理念，各个工具以“即插即用”的方式存在，为开发者提供了极大的便利，使得功能的增减变得轻而易举，特别是模型上下文协议（Model Context Protocol，MCP）的出现，让工具调用更加方便；高效调度是指该模块擅长任务的分解与调度，不仅能够有效处理多任务并行的复杂场景，还能确保任务间依赖关系的准确无误。

工具的使用带来了以下优势。首先，在功能上，借助专门的工具，智能体能够胜任原本无法完成的任务，如精确计算、实时数据获取、图像生成等，极大地提升了其能力；其次，在适应性上，工具模块使得智能体能够灵活应对多种应用场景和任务需求，不再受限于训练数据的知识范围；在实时性上，该模块能够获取和处理最新信息，有效提高了智能体的实时性和准确性。然而，工具的使用也存在一些问题，一是增加了系统复杂性，尤其是在工具调用失败、数据传输错误或任务调度不合理时，可能会影响系统性能；二是增强了对外部工具的依赖性，一旦工具不可用，智能体的功

能将受到限制，增加了对外部服务的依赖；三是安全与隐私问题，调用外部工具时可能涉及敏感数据，这就需要采取额外的安全措施，以防数据泄露和不当使用。

工具模块为智能体提供了强大的扩展能力，使其能够在多种任务场景中表现出色。随着技术的不断发展，工具模块将继续演进和丰富，进一步增强智能体的智能化和灵活性。

1.2.3　记忆模块

记忆模块是智能体在对话、任务执行和长期交互中展现出持续性和连贯性的重要组成部分，它允许智能体在与用户交互的过程中“记住”重要信息，并在未来的任务中使用这些信息。记忆模块的发展与认知科学和神经科学的研究密切相关，这类似于人类的短期记忆和长期记忆系统，研究者们尝试将人类的这种记忆模型应用于AI系统中，以便它们能够模仿人类记忆的机制，像人类一样处理信息。

在构建智能体时，记忆模块主要分为两大类：短期记忆和长期记忆。短期记忆负责暂存当前对话或任务的相关信息，这些信息往往是临时的，一旦任务结束，它们便会被清除。长期记忆则负责存储用户或任务的持久性信息，比如用户的喜好、经常使用的指令等，这些信息能够在多次对话或任务中持续发挥作用，为用户提供更加个性化的服务。

记忆模块在技术上展现出几个显著的特点：它具备持久性，使得长期记忆能够跨越不同的会话或任务，为智能体赋予持久的记忆能力；同时，它还具备上下文敏感性，能够精准识别当前对话或任务的背景，据此检索出相关信息以做出恰当的响应；此外，记忆模块还展现了自适应性，它能够根据与用户的互动情况动态地调整记忆内容，实时更新用户的偏好或任务相关的信息。

记忆模块在应用中展现出独特的优势，主要体现在三个方面：首先，它显著提升了交互的连贯性，使得智能体能够在多轮对话中保持话题的一致性和流畅性，从而极大地增强了用户体验；其次，记忆模块能够根据用户的历史行为和偏好，为用户提供更加贴心的个性化服务；最后，在处理复杂任务时，记忆模块能够保存中间状态和上下文信息，辅助智能体更高效地完成任务。然而，记忆模块也并非完美无缺，它的引入无疑增加了系统的复杂性，尤其是在管理长期记忆时，涉及大量数据的存储、检索和更新问题；同时，用户隐私和数据安全问题也随之而来，需要我们采取有效措施确保信息的安全；此外，记忆的遗忘机制设计也是一项挑战，如何恰到好处地删除或更新记忆，以避免错误响应或信息过时，是开发者必须面对的问题。

未来，记忆模块将朝着更高效、更安全的方向发展：

（1）分布式记忆系统：智能体可以使用分布式存储技术，允许记忆模块在云端或边缘设备上协同工作，以提高效率和数据安全性。

（2）自学习记忆系统：记忆模块将更加智能化，能够根据上下文自动学习和优化记忆策略，减少人工干预。

（3）隐私保护与合规性：随着数据隐私法规的日益严格，记忆模块将在数据保护和合规性方面进行改进，确保用户数据的安全性和隐私性。

总的来说，记忆模块是智能体的重要组成部分，它赋予智能体跨时间和跨任务的连续性和个性化能力。随着技术的不断进步，记忆模块将进一步提升智能体的智能化水平和应用广度。

1.2.4　规划模块

规划模块是智能体系统中负责设计行动方案的核心部分，也是用于衡量大模型智能水平的能力之一。规划模块的思想源自人工智能研究的早期阶段，并在机器人学、运筹学、自主导航、任务调度、决策制定等领域中发挥着至关重要的作用。它通过分析当前环境和目标，制定出一组可执行的步骤或策略，以实现预期的结果。具体规划步骤如下：

01 状态评估：理解当前的系统状态，包括环境信息、资源状况以及智能体的内部状态。
02 目标设定：明确需要达成的目标，通常是具体的任务或状态。
03 行动生成：基于状态评估和目标设定，生成一系列可以执行的行动序列。
04 评估与优化：在可能的情况下，对生成的行动序列进行评估和优化，确保以最优的方式达到目标。

规划模块的技术特性表现在以下三个方面：首先，它具备处理高维度复杂任务的能力，尤其是在面对多目标和多约束条件时，规划模块能够进行大量的计算和权衡，以确保任务的顺利完成；其次，它展现了出色的自适应性，能够在环境动态变化的情况下灵活调整行动计划，例如，当遇到突发障碍物时，规划模块能够迅速重新规划路径；最后，规划模块的应用范围极为广泛，它不仅在机器人学、自动化调度领域大显身手，也在游戏AI、决策支持系统等多个领域展现出其强大的通用性和实用性。

规划模块在实际应用中展现出明显的优势，主要表现在灵活性、优化能力以及多任务处理能力上。规划模块使得智能体能够针对未知或复杂环境灵活调整行动策略，显著提升任务执行的成功率；同时，借助启发式算法和优化技术，它能制定出高效的行动计划，以最大化资源利用或者最小化成本；此外，规划模块还能同时应对多个目标和约束条件，适应各种复杂的任务场景。然而，规划模块也存在一些不足之处，例如在处理复杂规划问题时，计算复杂度会大幅上升，尤其在处理高维度空间或实时系统时更为明显；在极端环境下，如面临高度不确定性或环境剧变时，规划模块的鲁棒性可能会受到挑战，导致生成的计划频繁失效；最后，规划模块的效果在很大程度上依赖于大模型的规划能力，如果大模型存在偏差或不完整，规划模块的结果可能会不尽如人意。

规划模块未来可以结合全局规划和局部规划，既能制定长远目标，也能灵活应对短期变化。随着多智能体系统的普及，规划模块需要支持多个智能体的协调与合作，实现复杂任务的集体行动。更进一步，随着智能体协作变得越来越复杂，规划模块在面对不确定性和动态环境不断变化时，需要表现出更高的自适应性和鲁棒性，进一步提升智能体在真实世界中的应用效果。总而言之，规划模块不仅为智能体提供了实现目标的行动蓝图，还为系统在动态环境中成功执行任务提供了保障。随着技术的不断发展，规划模块将在越来越多的复杂应用中发挥关键作用。

1.2.5　RAG 模块

RAG是智能体中负责将检索（Retrieval）和生成（Generation）结合起来的一种高级模型架构。其目标是增强大模型的能力，使大模型能够在生成内容时借助外部信息来源，提高响应的准确性和相关性，解决大模型因训练数据时间或覆盖面不足而导致的错误。

RAG主要分两个阶段，即检索阶段和生成阶段。检索阶段是指在生成内容之前，首先通过检索模块（通常是向量搜索引擎）从一个大型文档集合或向量数据库中查找与当前输入相关的文档片段或知识点；生成阶段是指将检索到的内容与输入问题结合，一起输入大模型中，生成最终的回答或文本内容。

RAG模块的技术特性主要体现在动态知识注入、高效检索与生成以及灵活扩展性这三个方面。它能够实时地将最新的领域知识融入生成模型中，极大地提升了生成内容的时效性和精确度；同时，依托于尖端的向量检索技术，RAG模块能够迅速定位到与输入相关的文档片段，并通过大模型高效地创造出新的内容；此外，RAG模块的架构设计极具弹性，可以根据实际需求轻松地添加或调整知识库，以适应多样化的应用场景。

RAG模块的优势在于能够显著扩展生成模型的知识覆盖范围，通过检索机制引入训练数据之外的信息，尤其适合于那些需要即时更新内容的场景；同时，它通过检索到的高质量信息来指导生成模型，有效提升了生成内容的准确性和相关性；此外，RAG模块还能够灵活应对不同领域和任务的需求，通过定制知识库，为多种应用场景提供高效的解决方案。然而，RAG模块也存在一些不足，其性能极大依赖于知识库的质量和更新频率，一旦知识库陈旧或质量不佳，其生成效果便会大打折扣；同时，由于涉及检索和生成两个阶段，RAG模块对计算资源的需求较高，在处理大型知识库时，计算和存储的负担尤为明显；最后，相较于纯生成模型，RAG模块的架构更为复杂，这要求开发者同时优化检索和生成两个模块的性能，增加了系统的整体复杂性。

RAG模块通过引入更多的上下文理解和语义匹配技术，使检索模块变得更加智能，能够更好地理解和匹配用户的查询需求；同时集成更加庞大且多样化的知识库，包括结构化和非结构化数据，从而进一步提升生成内容的质量和深度；另外，RAG模块也会集成自动化的知识更新机制，能够实时学习和更新外部知识库内容，确保生成内容的时效性和准确性。总而言之，RAG模块通过将检索和生成能力相结合，为智能体提供了强大的动态知识获取和生成能力。随着技术的不断发展，RAG模块将在越来越多的应用场景中发挥关键作用，进一步推动AI系统的智能化和实用化。

关于提示词的内容，我们将在第3章中进行详细介绍。

1.3　智能体的发展历程

智能体技术可以追溯到人工智能研究的早期阶段，特别是20世纪50—70年代的知识表示与推理系统。最早的智能体模型是一些基于规则的系统，用于自动化推理和决策。随着计算机技术的发

展，研究者逐渐认识到创建能够独立工作并协作的系统的重要性，这促使了智能体技术的发展。在20世纪80年代，分布式人工智能（DAI）领域的研究推动了多智能体系统（Multi Agent System，MAS）的概念的出现。此时，智能体被视为能够在分布式环境中独立操作并相互协调的小型智能实体。

整体来看，可以将智能体的发展总结为下面几个阶段：

- 哲学启蒙阶段：智能代理的概念最早可以追溯到古代哲学家的思想，如亚里士多德和老子。他们探讨了具有欲望、信念、意图和行动能力的实体，这些思想为后来智能代理的发展提供了哲学基础。
- 发展阶段（1950—1970年）：这一时期智能体的概念主要集中在基于规则的推理系统和专家系统上，目的是模拟人类专家的决策过程。图灵将“高度智能有机体”的概念扩展到了人工智能领域，并提出了著名的图灵测试，这标志着智能体作为人工智能系统的基本模块开始被正式研究。
- 寒冬阶段（1970—1990年）：这段时期先经历了人工智能的第一次长达约十年的寒冬，其后到1980年左右人工智能再一次迎来热潮。这时人工智能的各项研究都有所突破，来自政府等机构的投资也开始增多，研究者们对AI Agent的探索也逐步增加。这股热潮仅维持了7年，到1987年迎来第二次人工智能寒冬。
- 智能体初期阶段（1990—2000年）：随着人工智能的复苏与进一步发展，智能体在1995年被Wooldridge和Jennings首次定义为一种能在特定环境中自主行动以实现目标的计算机系统，强调智能体应具备自主性、反应性、社会能力和主动性。随着智能体在经济学中的广泛应用，其定义被扩展为能够感知环境并采取行动以提高成功机会的系统，这使得简单的程序和能与人类对弈的棋类游戏机器人等也被纳入智能体的范畴。智能体的研究范式将人工智能的研究扩展到了智能代理研究，超越了对人类智能的研究，探索更广泛的智能形式。
- 蓬勃发展阶段（2000—2020年）：随着互联网和机器学习技术的发展，特别是深度学习技术的进步，使智能体变得更加智能和自主。智能体技术与网络技术结合，形成了各种分布式智能体系统，智能体开始在虚拟助手、智能客服、网络安全等领域得到应用。这一时期强化学习等技术也被广泛应用于训练智能体，以处理机器人控制、智能调度等复杂任务。
- 大语言模型阶段（2020年以后）：近年来，智能体技术与大规模预训练模型（如GPT）结合，形成了更为复杂和智能的系统，智能体的研究和应用进入了一个新的阶段。智能体被视为大模型、记忆、任务规划和工具使用的集合，开始具备处理多模态数据（如文本、图像、语音）的能力。借助大模型强大的语义理解和规划能力，智能体的应用场景进一步扩大。

随着更复杂的神经网络技术和大模型技术的发展，智能体将在自主性、适应性以及智能性方面有进一步的提升，同时能够在多个领域之间无缝协作，在应用之间实现更广泛的数据协同和功能协同。在人机协作方面，智能体技术的发展将推动人机协作的新模式。智能体将成为人类决策和日常生活中的重要助手，提供更加个性化和智能化的服务。值得注意的是，随着智能体能力的增强，

智能体将接管越来越多的人工操作的控制权，这对伦理和安全性提出了更高的挑战，如何保证智能体的安全性、透明性和伦理性将成为研究的重点。

1.4　智能体的等级划分

对智能体的等级划分其实是对智能体成熟度的一种评估。如今生成式AI飞速发展，智能体产品（如对话机器人、AI数字人、Agent应用程序等）成为业界首选方式之一，如何评价智能体的成熟度或者智能水平也成为实际应用落地需要关注的问题。但目前业界对智能体的等级划分尚未形成一套统一的标准，因此各家都提出了自己的等级划分标准，可谓智者见智。

1.4.1　RASA

RASA是一家比较早期的ChatBot技术领域的公司，在大语言模型出现之前，市面上大部分的智能对话聊天机器人都是基于RASA框架来实现的。这家公司按照与人交互的范围、深度和群体广度，将智能体划分为如下5个等级：

（1）L1级——基础通知助手：例如我们最熟悉的手机通知功能，仅能在微信等消息应用中显示简单通知。

（2）L2级——常见问题解答助手：是目前最普遍的助手类型，允许用户提出问题并获得直接回答，功能相较于传统的FAQ（Frequently Asked Questions，常见问题）搜索页面有所提升，助手通过一两个后续问题增强了互动性。

（3）L3级——考虑上下文的助手：正如机器人开发者所强调的，用户提出的问题往往不是孤立的。上下文（包括用户的过往对话、提问的时间地点和方式等）对于理解用户需求至关重要，这意味着AI助手能够处理多样化和非预期的输入。

（4）L4级——个性化AI助手：就像人类希望随着时间推移，他人能更加了解自己一样，这一级别的AI助手也将开始适应这一需求。例如，它会学习何时与用户沟通，并根据情境主动发起对话，它还会记住用户的喜好，提供定制化的用户体验。

（5）L5级——企业级AI助手：最终，将出现一系列AI助手，它们能够深入了解每位客户，并承担企业运营的大部分任务，包括市场营销、销售、人力资源以及财务管理等。

1.4.2　Google

谷歌根据智能体所能达到人类能力程度的百分比，对智能体的等级做了划分，并分别从特定性任务和通用性任务的视角给出了定义。特定性任务是指AI系统被设计来执行特定的、定义明确的任务或一组任务；通用性任务是指AI系统被设计来执行广泛的任务，包括学习新技能等元认知能力。其中智能体从“No AI”到“Superhuman”共分为6个等级：

（1）L0级——No AI：表示没有任何AI能力的软件或系统。从特定性任务的视角看，普通的计算器软件、编译器等都属于L0级；从通用性任务的视角看，亚马逊的Mechanical Turk这类完全依赖人工参与的计算任务就被定义为L0级。

（2）L1级——Emerging：这个等级的智能体开始出现AI技术，能力与不熟练的人类相当或略优。从特定性任务的视角看，早期的基于规则的系统（如SHRDLU），以及一些简单的AI应用，就属于L1级；从通用性任务的视角看，ChatGPT、Bard、LLaMA 2等大模型是2023年左右出现的早期通用生成式AI（Artificial General Intelligence，AGI）的尝试，属于L1级。

（3）L2级——Competent：这个等级的智能体在某些任务上能够达到至少50%熟练成人的水平。从特定性任务的视角看，如毒性检测、智能扬声器、视觉问答系统、IBM的Watson等属于L2级；从通用性任务的视角看，目前还没有实现这一水平的AGI。

（4）L3级——Expert：这个等级的智能体在特定任务上达到90%熟练成人的水平，即专家级别。从特定性任务的视角看，如拼写和语法检查器、生成图像模型等属于这一等级；从通用性任务的视角看，目前还没有实现这一水平的AGI。

（5）L4级——Virtuoso：这个等级的智能体在特定任务上达到99%熟练成人的水平，即大师级别。从特定性任务的视角看，如深蓝（Deep Blue）、AlphaGo等在特定游戏领域超越人类的AI属于这一等级；从通用性任务的视角看，目前还没有实现这一水平的AGI。

（6）L5级——Superhuman：这个等级的智能体已经达到超人级别，能够超越所有人类。从特定性任务的视角看，如AlphaFold、AlphaZero、StockFish等在特定科学或游戏领域超越人类的AI属于这一等级；从通用性任务的视角看，目前还未实现人工超级智能（Artificial Super Intelligence，ASI）。

1.4.3　类自动驾驶

对比上述等级划分，笔者比较认同另一种借鉴类似自动驾驶级别的划分方式，将智能体根据其推理和反思的能力从低到高进行划分：

（1）L0级——没有AI：这一级别的智能体并不具备人工智能特性，只能执行预定或固定的任务，没有感知、决策或学习的能力。

（2）L1级——规则符号智能：智能体开始具备基于规则的决策能力，能够根据预设的规则和符号进行简单的判断和执行。这种智能体通常只能处理特定情境下的任务，且缺乏灵活性和适应性。

（3）L2级——推理决策智能：智能体能够利用逻辑推理能力来解决问题，不再仅依赖预设的规则。它能够根据当前的环境信息和目标，进行一定程度的推理和决策，以选择最合适的行动方案。

（4）L3级——记忆反思智能：智能体不仅具备推理决策能力，还开始拥有记忆和反思的能力。它能够记住过去的经验和教训，并在未来的决策中加以利用。这种智能体能够自我优化和改进，以适应不断变化的环境和任务。

（5）L4级——自主学习智能：自主学习是L4级别智能体的主要特征，它能够自主地从数据中学习新知识和技能，无须人类的明确指导。这种智能体能够处理更复杂的问题，并在面对新情境时展现出更强的适应性和创造力。

（6）L5级——个性群体智能：智能体不仅具备高度自主的学习和决策能力，还展现出个性化的特征。它能够根据自身的特点和偏好来执行任务，并与其他智能体进行协作和沟通。此外，L5级别的智能体还能够理解和适应人类社会的复杂性和多样性，与人类实现更加紧密和自然的交互。

通过清晰定义智能体的不同等级，一方面可以标准化智能体应具备的功能特性，这种标准化有助于用户理解他们可以期待什么样的性能和响应；另一方面也可以确保只有具备相应智能等级的智能体才能访问敏感数据或执行关键任务，从而提高系统的安全性。总之，智能体的等级划分有助于推动智能体技术的发展，提高企业的运营效率，增强用户体验，并确保技术的安全性和伦理性。

1.5　为什么要使用智能体

为什么要使用智能体？这一问题需要结合智能体的特点来分析，特别是需要对比大模型的优缺点，智能体能解决大模型本身无法解决的问题。

1.5.1　大模型的缺点

大模型虽然在自然语言处理方面取得了显著的成就，但它们也存在一些缺点：

（1）会产生幻觉：大模型可能会生成与现实不符的信息，甚至胡编乱造，这种现象被称为“幻觉”。

（2）对时事的了解有限或一无所知：大模型的训练数据通常有一定的时间滞后，因此它们可能不包含最新的时事信息，这限制了它们在需要最新信息的任务中的有效性。

（3）很难应对复杂的计算：大模型在处理复杂的数学计算和逻辑推理方面存在局限。

（4）没有行动能力：大模型通常不能直接与物理世界交互，它们缺乏执行物理任务的能力。

（5）没有长期记忆能力：大模型通常不具备长期记忆，这限制了它们在需要记忆过去交互或用户偏好的任务中的性能。

智能体可以有效解决大模型上述缺点，例如智能体可以通过设计良好的提示工程和RAG技术有效缓解幻觉的产生；通过注入外部的真实数据信息，激发大模型本身的真实性；通过引入搜索工具实时查询最新的信息，例如利用搜索引擎查询最新的新闻数据，将新闻数据加入提示词中，再请求大模型，就能有效解决大模型对时事一无所知的问题；通过调用计算类的工具，将大模型与传统精确的计算系统相结合，弥补大模型难以应对复杂计算的问题；智能体还可以调用物理世界的控制类API工具，让大模型充当大脑，让API工具充当执行器，这样就具备了行动能力；智能体的记忆模块可以解决大模型缺乏长期记忆能力的缺点。

1.5.2　智能体的特点

结合智能体各个组成模块的优缺点可以看出，智能体具备一些典型特征，这些特征使其在执行具体任务和适应环境方面表现出色，也能有效解决大模型本身的一些缺陷：

（1）自主性（Autonomy）：智能体能够自主运行，无须外部干预，它可以独立执行任务、管理资源，并在必要时进行决策。

（2）感知能力（Perception）：智能体能够通过工具感知外部环境，例如通过读取传感器的方式获取信息，这使其能够了解环境变化，并根据变化调整行为。

（3）行动能力（Action）：智能体能够通过工具调用的方式执行特定的行动，这种行动可能是物理动作（如机器人运动）或虚拟操作（如发送指令、改变数据状态等）。

（4）社会性（Social Ability）：多智能体系统中的智能体可以相互通信和协作，形成一个协调的整体。它们能够分享信息、分配任务，并共同解决问题。

（5）适应性（Adaptability）：智能体能够根据经验或新的信息调整其行为，这种适应性可以通过学习算法（如强化学习、深度学习）实现。

从智能体的特点可以看出，对比传统的静态系统，智能体具备动态处理复杂任务的能力，例如，它可以将复杂问题分解为子任务逐步解决，并根据实时反馈调整策略；它还可以有效地自动化处理大量重复性任务，如客服问答、数据处理等，显著提高效率；同时通过自动化操作，可以减少人为错误，提高任务执行速度；在长期使用中，智能体也可以替代大量人力资源，节省运营成本；在个性化服务方面，智能体能够根据用户需求实时访问数据源或API，提供个性化的建议和服务，提升用户体验。总的来说，智能体技术的发展不仅有助于提升自动化水平，还能带来更加智能化的解决方案，催生出新的业务模式和服务形态，能推动各行各业的发展，是未来人工智能发展的重要方向之一。

1.6　什么是多智能体协同

多智能体协同（Multi Agent Coordination）是一种涉及多个智能体之间相互合作与协调的系统，类似于人类社会的不同角色分工协作，共同完成同一个任务。每个智能体通常是一个独立的决策实体，能够感知环境、做出决策并执行行动。通过协同，这些智能体可以实现比单智能体更复杂和有效的任务。

多智能体协同系统展现了几个显著的技术特点：每个智能体均具备自主性，拥有独立的决策和行为执行能力；智能体之间能够实现有效协作，通过共享信息和资源，共同解决问题；智能体之间通常需要通过特定的通信方式来传递信息、协调行为；智能体可能分布在不同的地点，不一定集中在同一物理位置或服务器上，体现出多智能体协同的分布式特性；智能体具备适应性，能够根据

环境变化或其他智能体的行为来调整自身策略，以适应不断变化的情况；最后，多智能体系统具有较好的容错性，因为系统功能由多个智能体共同承担，单一智能体的故障不会导致整个系统崩溃。

多智能体协同技术的优势主要体现在灵活性、扩展性、鲁棒性、动态适应能力和分布式计算能力上。这种协同模式使得多个智能体能够共同处理更为复杂的任务，这些任务对于单一智能体而言可能难以应对；系统的扩展性体现在可以通过增加智能体数量来提升处理能力和规模，以适应不同级别的任务需求；其鲁棒性得益于智能体间的冗余和分布式特性，使得系统对于个别智能体的故障具有较强的抵抗力；动态适应性是指多智能体系统能够根据环境和任务的变化进行动态调整，从而提高整体的灵活性和效率；分布式计算则允许任务在多个智能体间分配，通过并行处理提升效率。

然而，多智能体协同系统也存在一些缺点，如系统的设计、实现和维护相较于单智能体系统更为复杂，智能体间的协调和通信可能带来额外的挑战；通信过程中的网络开销可能导致系统性能下降；多个智能体间的协调和冲突解决可能颇为困难，特别是在存在竞争或利益冲突的情况下；此外，系统的行为由于涉及多个自主智能体的交互和协调，可能变得难以预测和调试，给系统的监控和调试带来了难度。

相比于单智能体，多智能体协同在处理复杂任务时展现出显著优势，能通过分工合作实现并行处理，突破性能限制。这种系统能灵活应对多变环境，具备增强的鲁棒性，个别智能体故障不会导致系统崩溃。协同工作还能产生协同效应，整体能力超越个体之和。总体来说，多智能体协同系统适合处理需要高灵活性、高鲁棒性的复杂任务，但同时也需应对系统设计和管理上的复杂性。

1.7　实验环境搭建

截至目前，从国内外大模型的通用能力来看，OpenAI的GPT-5领先国内一众大模型厂商，但GPT-5需要通过外网访问，在国内使用存在诸多不方便。国内的一些知名大模型虽然整体效果不如GPT-5，但对实际应用任务来说，其能力也基本够用。从易用性的角度对比国内的这些大模型，笔者的真实感受是智谱AI的大模型易用性最好，因此，本书主要选用智谱AI的大模型作为大模型选型。由于大模型的发展非常迅速，后续可能有不同厂商的大模型能力有长足的进步，读者可以根据实际情况来选择具体的大模型。

1.7.1　网页版对话测试

本章前面说了很多大模型的优点，但读者可能还没有一个直观的感受，因此我们首先采用网页版的对话页面，来直接使用一下大模型，体验大模型的超强文本理解能力和生成能力。

1. 登录智谱AI官网

首先登录智谱AI的官网（https://www.zhipuai.cn），自行注册账号后，在菜单栏的“AIGC产品”下拉列表中，选择第一个“智谱清言”选项，单击进去后可以看到如图1-4所示的页面。

图 1-4　智谱清言主页

从网页左边的菜单栏中可以看到，智谱清言提供了Chat对话功能；AI搜索功能，可以使用智能搜索引擎；AI画图功能，可以通过输入文本描述直接生成图像；AI生视频功能，可以通过输入文本描述直接生成对应的视频；数据分析功能，可以对数据进行分析；长文档解读功能，可以做文档总结和文档翻译，对阅读论文时非常有帮助。接下来让我们体验一下这些常用的功能，真实感受大模型的魅力。

2. ChatGLM对话功能

首先体验一下ChatGLM对话功能。假设你不知道什么是大模型幻觉，可以在ChatGLM功能页的对话框中输入“什么是大模型幻觉”，大模型会立即给予回复。实测效果如图1-5所示。

在输入框中输入的“什么是大模型幻觉”就是提示词，从大模型的回答结果来看，回答内容基本正确，说明大模型的能力已经比较强大。另外，在大模型给出的答案下面有“赞”“踩”等反馈按钮，用户可以通过这些按钮向大模型反馈信息。如果大模型生成的结果不好，通过反馈，在后续多轮对话过程中大模型会逐步改善回答结果。同时答案的最下面也给出了三条建议性问题，帮助用户继续提问，这些问题在实际体验过程中也是非常有用的功能。

3. AI画图功能

再体验一下AI画图功能。单击页面左侧的“AI画图”菜单栏，在最下面的对话框中输入提示词。例如想画一个小男孩在小路上骑自行车的场景，可以输入“夕阳下，宁静的小路上，欢快的小男孩骑着自行车”。此时大模型会调用 CogView3-Plus AI绘画工具来完成创作，实测效果如图1-6所示。

生成的画面质量还是非常不错的，非常符合输入提示词的要求。读者可以多尝试修改提示词，创作自己的精美插图。其实这个AI画图的功能就是一个典型的智能体，用户输入的文本内容作为提示词，智能体会调用CogView3-Plus AI绘画工具生成图片，并将生成的结果通过大模型总结后返回给用户。

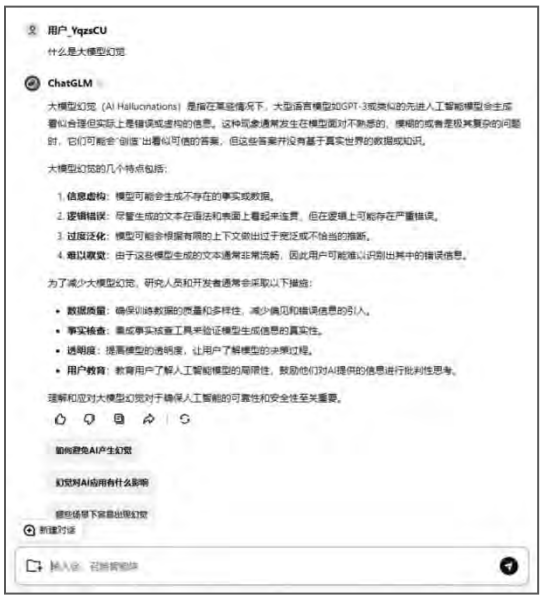

图 1-5　对话功能界面

图 1-6　AI 画图功能界面

由于篇幅有限，笔者在这里仅展示了对话功能和文生图的AI绘画功能，其他的一些功能读者可以自行体验。通过不断试用这些功能，能更清晰直观地体会大模型的能力，了解大模型的能力边界在哪里，这对读者创作自己的智能体应用非常有用，对提升日常办公效率也是非常有帮助的。比如现在有非常多的抖音短视频创作者，就会通过AI生成视频的方式，利用小说的文本内容来生成对应的短视频；游戏插画师也能利用AI绘画的方式，让AI直接生成自己想要的绘图效果，或者在AI生成的绘图结果上进行二次创作，极大地加快创作效率。

1.7.2　代码版对话测试

除了上面小节中展示的网页对话形式，对大模型应用开发工程师来说，更常用的是使用代码请求接口的方式直接与大模型对话。国内外请求大模型的形式已基本达成共识，都是通过API Key的方式鉴权。因此，我们需要先申请智谱AI的API Key。

1. 申请智谱AI的API Key

首先进入智谱AI的官网，在菜单栏的“AIGC产品”下拉列表中选择“智谱AI开放平台”，自行注册开放平台的账号后，选择最上方的“控制台”菜单项，进入控制台页面，如图1-7所示。

图 1-7　申请智谱 AI API Key 的页面

该页面右上角的“费用中心”显示当前账户余额。新用户注册时默认有20元可用，当余额不够时，需要先充值，然后才能继续使用API Key。

申请API Key需要进入个人中心，个人中心页面隐藏得比较深；单击最上面、最右边的小人图标，会出现一个下拉菜单，单击“个人中心”子菜单，进入个人中心页面，如图1-8所示。

图 1-8　个人中心页面

在个人中心页面，用户可以自行修改基本信息，进行安全设置等操作。

在申请API Key之前，必须进行实名认证，单击左侧“实名认证”菜单，用户可以利用第二代居民身份证信息自行完成个人实名认证。接下来单击左侧“API keys”菜单，进入API keys列表界面，如图1-9所示。

图 1-9　查看 API Key 列表界面

尽管系统默认设置了一个API Key，但为不同的应用设置不同的API Key，可以更好地监控各个应用消耗 token的情况，从而计算应用的活跃程度，或者判断应用内部是否发生调用异常，这对后续应用的运维非常有帮助。因此，为不同应用区分不同的API Key是一个良好的工程实践。

单击右上角“添加新的API Key”按钮，输入API Key的名称，用于区分不同的应用，界面如图1-10所示。

图 1-10　新建 API Key 界面

单击“确定”按钮后，就可以在API keys列表下看到新增的这一行API Key了，界面如图1-11所示。

图 1-11　API Key 列表界面

鼠标停留在“测试1”这行的API Key上，会出现一个复制按钮，单击该按钮，页面上会弹出一个“复制成功”的提示，表示已将完整的API Key复制到了剪切板上，供后续代码使用。

2. 安装开发环境

申请好API Key后，接下来开始安装Python安装环境。推荐使用MiniConda管理Python的虚拟环境，安装MiniConda的资料网上有很多，直接从官网下载安装即可。安装好MiniConda后，先构建Python 3.11的虚拟环境。为了使读者能复现本书中的所有代码示例，本书约定统一采用Python 3.11编程：

```
> conda create -n multi-agent python=3.11      # multi-agent为虚拟环境名
> activate multi-agent
```

将虚拟环境激活后，即可在控制台的路径前面看到虚拟环境名。调用智谱AI的API，需要先安装zhipuai的包：

```
> pip install zhipuai
```

到这里最简单的智谱AI大模型的调用环境就准备好了，接下来可以写代码来测试一下。

3. 测试智谱AI对话功能

假设我们要调用智谱AI的GLM-4-PLUS模型，完成一个简单的向大模型打招呼的示例。大模型的输出分为非流式输出和流式输出：非流式输出是指模型在完成全部推理和生成过程后，将完整的输出结果一次性返回给用户或调用方；流式输出是指模型在生成内容的过程中，边生成边逐步将结果返回给用户，呈现出“逐字输出”或“逐句输出”的效果。

1）演示非流式输出示例

【例1-1】演示如何通过调用智谱AI的Chat Completions API向大模型打招呼。

```
from zhipuai import ZhipuAI

# 请填写上面获取的API Key
```

```
client = ZhipuAI(api_key="你的API Key")
response = client.chat.completions.create(
    model="glm-4-plus",     # 填写需要调用的模型名称
                            # content为用户的输入
    messages=[
        {"role": "user", "content": "你好！你叫什么名字"},
    ],
    stream=False,           # 使用非流式回答
)
answer = response.choices[0].message.content
print(answer)
```

将代码中的api_key参数替换为上面申请的API Key。model参数为需要调用的大模型名称，例如glm-4-plus、glm-4-0520、glm-4、glm-4-air、glm-4-airx、glm-4-long、glm-4-flash等，每种模型适用于不同的应用场景，具体的差异可以在使用指南页面（https://bigmodel.cn/dev/ howuse/model）查看。messages参数为对话消息列表，每条对话消息是一个JSON对象，role表示这条消息所属的角色，有user、assistant和system：user为用户的输入，assistant为大模型的输出，system为全局设定，一般用于设定当前会话中给大模型定义的角色，例如“你是一个计算机专家，正在进行智能体相关的内容创作”。stream参数表示是否使用流式回答，由于大模型生成的内容往往比较多，如果等答案完全生成完后一起返回，中间的等待时延就会比较长，因此在实际应用过程中一般使用流式输出。

上面的代码示例使用的是非流式返回，因此可以一次性拿到生成的结果，如下所示：

```
你好！我是个人工智能助手，你可以叫我AI小助手。有什么可以帮助你的吗？
```

2）演示流式输出示例

【例1-2】演示如何通过调用智谱AI的API接口进行流式输出。

```
from zhipuai import ZhipuAI

# 请填写上面获取的API Key
client = ZhipuAI(api_key="你的API Key")
response = client.chat.completions.create(
    model="glm-4-plus",     # 填写需要调用的模型名称
                            # content为用户的输入
    messages=[
        {"role": "user", "content": "你好！你叫什么名字"},
    ],
    stream=True,           # 使用流式回答
)
for chunk in response:
    print(chunk.choices[0].delta)
```

返回结果如下：

```
content='你好' role='assistant' tool_calls=None
content='！' role='assistant' tool_calls=None
content='我是一个' role='assistant' tool_calls=None
```

```
content='人工智能' role='assistant' tool_calls=None
content='助手' role='assistant' tool_calls=None
content=', ' role='assistant' tool_calls=None
content='你可以' role='assistant' tool_calls=None
content='叫我' role='assistant' tool_calls=None
content='AI' role='assistant' tool_calls=None
content='助手' role='assistant' tool_calls=None
content='或者' role='assistant' tool_calls=None
content='直接' role='assistant' tool_calls=None
content='叫我' role='assistant' tool_calls=None
content='小' role='assistant' tool_calls=None
content='智' role='assistant' tool_calls=None
content=', ' role='assistant' tool_calls=None
content='很高兴' role='assistant' tool_calls=None
content='为你' role='assistant' tool_calls=None
content='服务' role='assistant' tool_calls=None
content='! ' role='assistant' tool_calls=None
content='' role='assistant' tool_calls=None
```

可以看到，此时的输出是一小段一小段地返回，而不是一次性返回，当大模型返回的文本比较长时，流式输出的用户体验比非流式要好。但不能说所有场景下都用流式输出，流式输出在编写代码时对编程能力要求比较高，使用起来相对复杂。在大模型需要结构化输出，并需要对结构化输出的文本进行进一步处理时，使用非流式会更方便一些。例如，需要大模型以JSON格式返回，再对JSON字符串进行解析，这个时候使用非流式输出的方式会更方便。

上述只是两个简单的示例代码，来说明如何调用智谱AI的大模型。智谱AI提供的“zhipuai”SDK中，还有其他方便的接口供开发者使用，这里不一一列举，感兴趣的读者可以自行摸索，查看接口文档（https://open.bigmodel.cn/dev/api/devguide/sdk_example），试用不同的功能。

1.8　本章小结

本章首先对智能体的概念做了定义和说明，接着对智能体概念定义中的各个组成模块做了概括性的介绍，包括它们各自的技术特点、优缺点、未来的发展方向等。通过对不同企业对智能体等级划分的介绍，使读者对智能体的发展阶段有一个了解，方便后续在实际应用中对智能体的能力边界有一个参考度。最后借用智谱AI的网页版向读者展示了如何方便地直观体验大模型，同时也提供了代码版本的示例，一步一步引导读者“跑通”大模型的API调用流程，为本书后续的演示示例做准备。

在接下来的章节中，笔者将更深入地介绍智能体各个组成部分的技术原理和框架，并通过具体的代码案例，指导读者高效地开发智能体应用。至此，我们已为智能体的学习之旅奠定了基础，让我们一起进行更深入的探索吧！