第 1 章

初识FastGPT

在这个数据洪流奔涌、智能技术飞速迭代的时代，企业对高效知识管理与智能问答的需求如同燎原之火般迫切。想象一下，当员工为查询一条规章制度等待30分钟，工作节奏被硬生生打断；当HR面对100份简历，从初筛到匹配岗位要求需要连续8小时高强度工作，还可能因主观判断失误而错失优质人才；当法务逐字逐句核对50页合同，耗时4小时却仍有15%的条款遗漏，为企业埋下法律纠纷的隐患——这些看似寻常的场景，正在悄无声息地吞噬着企业的效率与利润，成为企业数字化转型路上的“隐形绊脚石”。而今天，一款名为FastGPT的智能引擎横空出世，正以颠覆性的技术力量重塑这一切，为企业打开高效运营的全新大门。

1.1 FastGPT简介

FastGPT是珠海环界云计算有限公司精心打造的企业级AI生产力引擎，它并非简单的“问答机器人”，而是一款开箱即用的智能体开发平台。自2023年正式上线以来，它已在短短两年多时间里服务了数百家企业级客户，足迹遍布金融、医疗、制造、教育、政务、能源等众多行业，在各行各业的数字化转型浪潮中扮演着不可或缺的关键角色。从证券机构的财报分析到医院的病历辅助解读，从工厂的设备检修预警到学校的个性化教学辅导，FastGPT正以多元化的应用场景证明着自身的价值。

我们通过大模型应用开发流程与烹饪流程的对比，来认识FastGPT的价值，如图1-1所示。我们将大模型比作“顶级食材”，比如新鲜的龙虾、珍贵的松露，它们本身具备极高的品质，却需要精湛的烹饪技艺才能转换为美味佳肴；这里，FastGPT就是一位掌握了全套烹饪技艺的“五星级主厨”。它通过前期数据处理（如同食材挑选清洗）、多模块整合（好比食材调配搭配）、适应性改造（类似于预处理加工）、整体工程化（如同煎炒烹炸的火候控制）等全流程工程化能力，将通用大模型的“潜力”精准转换为企业实际业务的“价值”。

无论是让AI深度理解企业内部的专属知识库，还是搭建贴合复杂业务场景的智能工作流，FastGPT都能轻松实现，让企业无须投入大量资源，从零搭建底层架构，就能快速拥有属于自己的AI能力，真正做到“即插即用”式的高效落地。

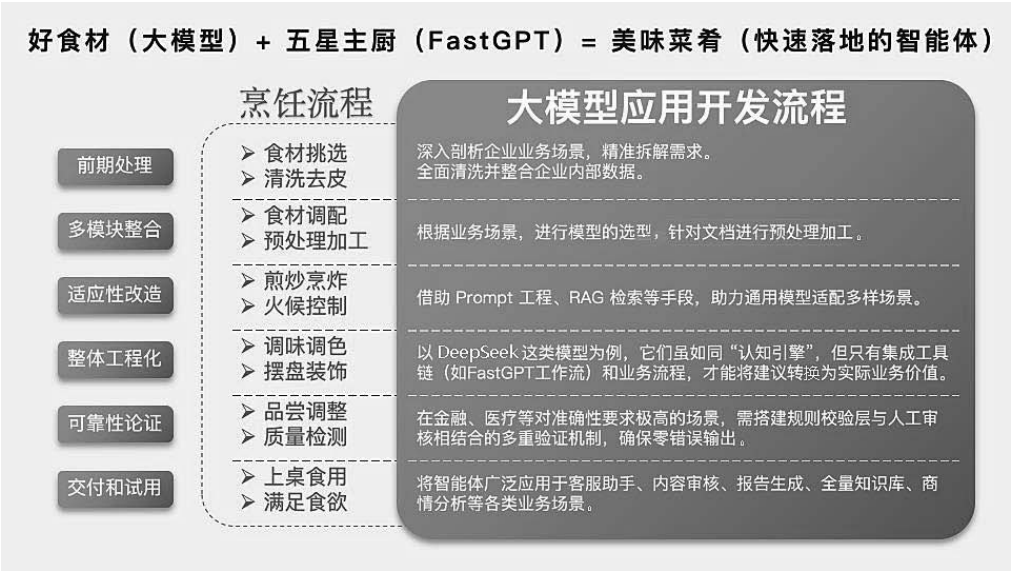

图1-1　大模型应用开发流程

在FastGPT的世界里，技术不再是高高在上的壁垒，而是触手可及的工具。它就像一位贴心的数字化助手，渗透到企业运营的各个环节：既能为HR自动筛选简历，从海量候选人中精准匹配岗位需求，将筛选准确率从60%跃升至92%；也能为法务审查合同风险，把50页合同的审查时间从240分钟压缩至4分钟，条款遗漏率压在2%以下；还能为客服提供实时问答支持，让客户咨询响应时间从“按小时计”缩短到“按分钟计”；更能成为企业知识沉淀与流转的“智慧中枢”，让分散在各个角落的文档、经验、规则形成有机整体，随时为业务决策提供支撑。正如某市政务利用它搭建24小时在线智能客服，实现“零窗口、零跑腿、零材料”的政务服务新体验；某快消品龙头企业通过它对接企业OA系统，将报销审批响应时间从24小时降至2小时，处理效率提升91.7%。总之，FastGPT正在用一个又一个实实在在的应用案例，重新定义AI与企业的关系，让智能技术从“实验室概念”真正走进企业的日常运营。

FastGPT的出现，不仅是技术层面的创新，更是对企业工作方式的重构。当传统的“人工主导、经验驱动”模式遇上FastGPT带来的“AI辅助、数据驱动”新模式，企业运营效率的提升不再是线性增长，而是跨越式飞跃。在金融行业，它能帮助投资经理将日均信息查询时间从2小时缩短至30分钟，信息覆盖度从60%提升至90%；在教育行业，它能让学生掌握单个知识点的时间从40分钟缩短至10分钟，同类题型正确率从60%提高至90%；在制造行业，它能将设备巡检时间从8小时缩至3小时，故障预警准确率达90%，非计划停机减少70%。这些真实的改变，正在让越来越多的企业意识到：FastGPT不是可有可无的“锦上添花”，而是提升核心竞争力的“必备利器”。

1.2　FastGPT的特点与优势

在AI技术飞速发展的今天，企业对智能工具的需求早已超越了简单的“问答功能”，转而追求能深度适配业务场景、高效解决实际问题的综合解决方案。FastGPT作为企业级AI生产力引擎，凭借其独特的技术架构与落地能力，构建起难以复制的竞争优势。这些优势不仅体现在技术参数的领先上，更转换为实实在在的业务价值，让AI从“实验室概念”真正走进企业的日常运营。

1.2.1　全格式兼容的数据处理能力

企业的数据往往分散在各种格式的文档中，从.txt、.md等轻量文本，到.pdf、.docx等复杂格式，再到.pptx演示文稿、.csv数据表，甚至包含图片、图表的混合文档，这些“数据孤岛”一直是企业知识整合的痛点。FastGPT搭载的智能数据处理引擎，如同一位全能的“数据管家”，实现了对主流格式文档的“全兼容”支持，彻底打破了格式壁垒。

其核心优势在于自动化的预处理流水线：上传文档后，系统会自动完成文本提取、清洗去重、结构化梳理等一系列操作。更值得称道的是其首创的“大模型辅助分块技术”——传统分块方式往往按固定长度切割文本，容易割裂语义逻辑，而FastGPT通过大模型对文档内容的智能理解，按段落逻辑、语义关联进行分块，确保每段文本的完整性。例如，在处理设备维修手册时，系统能自动识别“故障现象–排查步骤–解决方案”的逻辑链条，避免因文本分块导致的信息断裂。

针对包含图片、表格的复杂文档，FastGPT支持接入整合信息、Doc2x商业OCR工具，以及MinerU、Maker等开源OCR模型，将图片中的文字精准提取并转换为可检索内容。同时，通过“增强索引”功能，系统能自动为分块内容标注标题、关键词、图片说明等元数据，生成多维度索引库，使后续检索准确率提升30%以上。例如，某能源企业使用该功能后，设备检修手册的查询效率从原来的1.5小时缩短至15分钟，充分印证了其数据处理能力的价值。

1.2.2　可视化工作流

传统AI应用开发往往需要专业技术团队编写大量代码，业务人员的需求需经过多轮沟通才能落地，开发周期长、修改成本高。FastGPT的可视化工作流模块彻底颠覆了这一模式，它将复杂的技术逻辑封装为可拖曳的功能节点，让业务人员也能成为AI应用的“开发者”。

在FastGPT的工作流编辑器中，用户可以通过简单的拖曳操作，将“AI对话”“知识库搜索”“用户选择”“HTTP请求”“判断器”等节点自由组合，构建出符合业务逻辑的智能流程。例如，在合同审查场景中，可搭建“文档上传→条款提取→风险识别→人工审核→报告生成”的全流程工作流；在请假申请场景中，能实现“员工输入→信息校验→部门审批→HR归档”的自动化流转。某制造企业的HR团队通过该功能，仅用半天时间就能搭建出简历筛选工作流，将100份简历的处理时间从8小时压缩至1.5小时，效率提升超70%。

工作流的灵活性还体现在动态适配能力上。通过“判断器”节点，系统能根据不同条件触发不同分支流程——当合同金额超过100万元时自动转入高级审核流程，当请假天数超过3天时自动通知部门负责人。这种if-this-then-that的逻辑编排，让AI应用能精准适配企业复杂的业务规则。同时，工作流支持与外部系统的联动，通过“HTTP请求”节点对接企业ERP、CRM、MES等系统，实现数据的实时同步与流程的跨系统触发，真正打通AI应用与业务执行的“最后一公里”。

1.2.3　高精度RAG检索

大语言模型虽然具备强大的生成能力，但在处理企业内部专业知识时，常因“知识盲区”导致回答不准确。FastGPT通过优化的RAG（Retrieval Augmented Generation，检索增强生成）技术，让AI能精准调用企业知识库的信息，实现“有依据的回答”。

其RAG技术的核心优势在于多层级的检索优化：首先通过向量模型将用户问题与知识库内容进行初步匹配，筛选出相关性较高的候选文档；再接入Rerank模型对候选文档进行二次排序，提升Top*N*结果的相关性；最后结合上下文理解，从最优文档中提取关键信息生成回答，并标注引用来源。这种“粗

筛 + 精排 +生成”的三层架构，使回答准确率提升至99%，在金融、医疗等对准确性要求极高的领域表现尤为突出。

为满足不同场景需求，FastGPT支持接入全球主流的向量模型与Rerank模型，企业可根据业务特点选择最优组合。例如，在法律场景中选用精度更高的商业模型，在内部知识库场景中选用性价比更高的开源模型。某律所使用该功能后，合同条款的遗漏率从15%降至2%以下，每年因合同漏洞引发的纠纷损失减少86%，充分证明了其检索精度的价值。此外，系统还支持“问答模式”“摘要索引”等多种检索方式，用户可根据需求灵活切换，进一步提升知识获取效率。

1.2.4　多模型兼容的开放生态

不同企业的业务场景、数据安全要求和成本预算存在显著差异，对AI模型的需求也各不相同。FastGPT秉持开放兼容的理念，构建了多模型适配的技术生态，让企业能“按需选择”最适合的模型组合。

在大语言模型方面，FastGPT支持接入OpenAI、Claude、Qwen、DeepSeek等全球主流模型，无论是追求极致性能的GPT-4，还是注重成本控制的开源模型，都能无缝集成。企业可根据场景特点灵活切换——用DeepSeek处理专业领域问题，用Qwen处理日常办公场景，实现 “好钢用在刀刃上”。例如，某证券机构在使用中，对复杂的财报分析场景选用GPT-4，对日常资讯播报场景选用开源模型，在保证效果的同时降低了30%的算力成本。

除大语言模型外，FastGPT还支持向量模型、语音模型、OCR模型等多类型工具的接入，形成“模型 + 工具”的协同体系。例如，将语音模型与客服场景结合，实现“语音提问→文字转换→AI回答→语音播报”的全流程自动化；将OCR模型与报销场景结合，自动识别发票信息并完成校验。这种开放的生态架构，让企业无须受限于单一模型的能力边界，而是能根据业务需求构建专属的AI工具链。

1.2.5　企业级安全与权限管理

企业在引入AI工具时，数据安全与权限管控是首要考量。FastGPT构建了全方位的安全体系，从数据存储、访问控制到操作审计，实现全链路的安全保障。

在权限管理方面，FastGPT支持1:1对接企业组织架构，通过与HRMS系统的实时同步，自动获取员工的部门、岗位等信息，并基于此分配相应权限。企业可按“部门–群组–成员”三级架构设置权限，精细到“谁能查看知识库”“谁能修改工作流”“谁能导出数据”等具体操作。例如，让研发团队拥有模型训练权限，让业务团队仅拥有应用使用权限，确保数据访问“按需授权”。某大型制造企业通过该功能，实现了2000余名员工的权限精准管控，未发生一起数据泄露事件。

系统还具备完善的操作审计与用量管控功能，所有用户的操作行为都会被详细记录，包括文档上传、模型调用、权限变更等，可随时追溯；同时支持设置模型调用额度、数据存储容量等阈值，避免资源滥用。在数据安全方面，FastGPT采用加密存储、传输加密等技术，并通过等保三级认证，满足金融、医疗等行业的合规要求。配合SSO单点登录功能，员工只需一次身份验证即可访问所有授权应用，既提升了登录效率，又强化了身份认证的安全性。

1.2.6　全流程服务支持

企业AI应用的落地往往面临“需求梳理难、技术实现难、效果优化难”的三重挑战。FastGPT提供从售前到售后的全流程服务，为企业保驾护航。

在售前阶段，FastGPT团队会深入企业调研业务场景，通过POC（概念验证）测试验证需求的可行性，避免盲目投入。实施阶段提供定制化开发服务，从知识库构建、工作流设计到系统对接，由专业工程师全程参与。某零售企业在搭建智能客服时，FastGPT团队根据其业务特点定制了“商品咨询–订单查询–售后投诉”的专属流程，并对接CRM系统实现客户信息的实时调取，上线后客服响应时间缩短80%。

售后阶段提供持续赋能服务，包括在线技术支持群、详尽的操作手册、定期培训课程等，确保企业员工能熟练使用系统。更值得一提的是其“场景陪跑”服务，工程师会跟踪应用效果，协助企业优化工作流与知识库，持续提升AI应用的价值。香港教育大学通过该服务，将培训项目筹备周期从30 天缩短至10天，培训资源整合成本降低40%，充分体现了FastGPT服务体系的价值。

从数据处理到应用开发，从知识检索到安全管控，FastGPT以全方位的优势构建起企业级AI应用的完整解决方案。它不仅是一款工具，更是企业数字化转型的“加速器”，让AI技术能真正融入业务流程，创造看得见的价值。无论是降本增效、风险防控还是业务创新，FastGPT都在用技术实力证明：好的AI工具，应该让复杂的事情变简单，让企业的每一分努力都能收获更大的回报。

1.3　FastGPT版本体系概览

FastGPT围绕企业不同规模、技术需求和部署方式，构建了多层次的版本体系，从免费开源的基础体验到定制化的企业级方案，全面覆盖各类用户场景。版本设计以“用户规模”“部署方式”“功能深度”为核心划分维度，共包含四大类版本：开源版、SaaS版、商业版、企业版，覆盖从个人开发者到大型企业的全场景需求。其中，开源版与SaaS版主打轻量化体验，SaaS版作为FastGPT的经典云端方案，基于官方云服务器运行，无须本地部署成本，数据由官方托管，主打便捷性与轻量化体验。商业版与企业版聚焦企业级深度功能。开源版、商业版、企业版均支持单机或集群部署，数据存储在用户自有服务器，满足数据私有化需求。

1.3.1　各版本核心功能和适用场景

1. 开源版：基础功能的免费体验

（1）定位：面向个人开发者或技术团队的入门级版本，用于体验FastGPT核心框架。

（2）核心功能：支持基础工作流模块、通用文档知识库搭建，可对接飞书、语雀、API调用第三方知识库。但受限于免费的定位，无权限管控、高级索引增强、团队协作等功能。

（3）限制：仅支持一名用户使用，技术支持完全依赖社区资源，无官方服务保障。

（4）适用场景：技术验证、开源项目二次开发、个人知识库搭建，适合对功能需求简单且具备自主开发能力的用户。

2. SaaS版：轻量化云端便捷方案

（1）定位：为中小企业提供开箱即用的云端服务，无须本地部署与维护，是FastGPT体系中聚焦便捷性的经典版本。

（2）核心功能：功能与商业版轻量版完全一致，包括：

- 支持3~50名团队成员协作，可创建团队应用与共享插件，满足小型团队基础协作需求。

- 具备索引增强（标题索引、自动生成索引）、智能切片、Web 站点同步等知识库高级功能，提升知识检索精度。
- 支持飞书机器人、钉钉机器人、微信公众号等多渠道接入，可配置应用模板与工具箱，快速搭建场景化应用。
- 包含基础权限管控、文档标签管理、团队应用身份验证等功能，保障团队数据安全。

（3）独特优势：作为云端版本，数据存储于FastGPT官方服务器，省去服务器部署、运维成本；额外附赠模型token，满足初期高频调用需求，降低使用门槛。

（4）限制：无管理员后台、组织架构对接及SSO单点登录功能，功能深度适配轻量化场景。

（5）适用场景：中小团队快速搭建智能客服、内部规章制度查询工具，适合追求低成本与高便捷性，且无本地化部署需求的用户。

3. 商业版：企业级私有化基础方案

商业版作为FastGPT的主流企业级方案，分为轻量版、标准版、专业版、旗舰版，支持单机买断或按年订阅，核心差异体现在用户规模、功能深度与技术支持等级，如表1-1所示。

表1-1　FastGPT商业版各个细分版本的对比

细分版本	核心差异	适用场景
轻量版	支持3名用户，具备基础权限管控、团队应用管理、知识库索引增强（标题/自动生成索引）等功能；技术支持为L3级（工作日5×8工单/邮件响应，4小时内首次回复）	微型团队私有化部署，需求简单的内部知识库场景，如小型律所、创业团队的合同管理工具
标准版	扩展至200名用户，新增批量评估功能，支持更精细的权限配置与应用模板管理；技术支持升级为L2级（含L3服务+每周线上直播培训+ 5次更新升级）	中小型企业全场景应用，需团队协作与定期功能更新，如制造业的设备维修知识库、零售企业的产品咨询工具
专业版	支持1000名用户，开放SSO单点登录、组织架构对接、管理员后台，可管控密钥用量上限；技术支持为L1级（含L2服务+ 5人天远程知识库调优+ 10次更新升级）	中大型企业私有化部署，需复杂权限管理与系统集成，如集团型企业的跨部门知识共享平台
旗舰版	无用户数量限制，日API调用量不限制，支持审计日志、资源管理、用户支付功能；技术支持为L0级（含L1服务+原厂专属技术群+7×12小时值守+30次更新升级）	大型企业核心业务场景，需高可用性与定制化支持，如金融机构的合规问答系统、政务部门的政策咨询平台

4. 企业版：高端定制化与合规方案

企业版聚焦大型企业的深度需求，支持招投标、比价、供应商入库等正式采购流程，细分专业版、旗舰版与定制版。

（1）核心升级：在商业版功能基础上，强化合规性与定制化能力，包括全量审计日志、多租户管理、外部成员同步等功能，适配复杂组织架构。

（2）定制版特色：支持完全个性化的功能开发（如专属模型对接、复杂业务流程定制），部署模式可灵活选择单机或集群，满足超大规模用户与高并发需求，数据安全等级达到行业顶级标准。

（3）服务保障：所有企业版均提供L0级技术支持，含原厂多对一专属技术群、工作日1小时内响应、4小时内解决方案，确保核心业务稳定运行。

（4）适用场景：金融、医疗、政务等对数据安全与合规性要求极高的行业，或有深度定制需求的大型集团企业，如三甲医院的病历辅助解读系统、央企的全量知识库平台等。

1.3.2　版本对比

1. 核心功能差异

FastGPT各个版本的核心功能差异如表1-2所示。

表1-2　FastGPT各个版本的核心功能差异

功能维度	开　源　版	SaaS 版	商业版 （轻量/标准）	商业版 （专业/旗舰）	企　业　版
用户数量限制	1人	3人	3～200人	1000人至无限制	无限制
权限管控	□	基础权限	□	□ 细粒度管控	□ 全量权限
高级索引增强	□	□	□	□	□
团队协作功能	□	□（基础）	□	□ 高级	□ 全功能
SSO单点登录	□	□	□	□	□
组织架构对接	□	□	□ 标准版支持	□	□
审计日志	□	□	□	□ 旗舰版	□
技术支持等级	社区支持	L3级	L3～L2级	L1～L0级	L0级

2. 部署与数据存储

开源版、商业版、企业版：均支持本地化部署（单机或集群），数据存储在用户自有服务器，企业可完全掌控数据主权，满足金融、医疗等行业的数据合规要求，适合对数据隐私敏感的企业。

SaaS版：数据存储于 FastGPT 官方云端，无须本地服务器投入，开通即可使用，省去硬件采购、系统维护等成本，适合无技术运维团队或追求快速上线的中小用户。

3. 成本概况

（1）开源版：免费，仅需承担本地部署的服务器与技术运维成本。

（2）SaaS版：轻量化付费模式，年费数千元，附赠额外模型token，无部署与硬件成本，性价比突出。

（3）商业版：轻量版与标准版年费数千至数万元，专业版与旗舰版年费数万元，支持买断（一次性付费）或按年订阅，灵活适配企业预算规划。

（4）企业版：专业版与旗舰版年费数万至数十万元，定制版费用根据需求评估（通常十万级起步），适合预算充足的大型企业。

1.3.3　版本选择建议

（1）个人技术验证：优先选择开源版，免费体验核心功能，适合技术探索与二次开发。

（2）中小团队快速上线：推荐SaaS版，轻量化功能与云端部署兼顾成本与效率，附赠的模型token降低初期使用门槛，无须技术团队即可快速上手。

（3）中小型企业私有化需求：商业版轻量版或标准版足够覆盖基础团队协作与知识库需求，本地化部署保障数据安全，成本可控。

（4）中大型企业复杂场景：商业版专业版、旗舰版或企业版专业版，支持SSO、组织架构对接等深度功能，搭配L1～L0级技术支持保障稳定运行，适配跨部门协作场景。

（5）大型集团定制需求：企业版旗舰版或定制版，满足高并发、合规性与个性化开发需求，适配核心业务场景，通过原厂服务确保落地效果。

FastGPT的版本体系通过“功能分层 + 部署灵活”的设计，确保从个人到企业的全场景覆盖，用户可根据团队规模、数据安全需求与功能复杂度，选择最适配的方案，实现AI生产力的高效落地。

1.4　FastGPT部署与配置

1.4.1　准备工作

首先你要购买一台Linux操作系统的云主机，建议分配至少 4GB 内存和 20GB 存储空间用于容器运行和镜像存储。登录腾讯云官网选购性价比最好的云服务器（https://cloud.tencent.com/product/lighthouse），如果没有腾讯云官网账号，需要实名注册。如图1-2所示，在选购轻量应用服务器的页面，选择“使用容器镜像”，选择Ubuntu操作系统预先安装的Docker基础镜像，Docker基础镜像中除了包含底层的操作系统Ubuntu外，还默认封装了Docker软件、运行环境以及相关配置文件。

图1-2　轻量应用服务器页面

如图1-3所示，选择云主机的地域和配置规格。这里云主机的地域选择是指云服务器所在的物理数据中心的地理位置，理论上访客距离云服务器地域越近，网络延迟越低，速度越快。因此，腾讯云的云主机地域选择可以遵循就近原则，优先选择更靠近用户群的地域节点。例如，南方用户可以优先选择广州地域。配置规格选择入门型2核8GB内存，这里内存要选择大一些。

图1-3　云主机的地域和配置规格

如果你选择的是纯Linux操作系统，需要自行安装Docker。针对纯Linux操作系统的服务器环境，建议采用以下经过生产环境验证的安装Docker方案。

安装Docker的Linux命令：

```
    curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun
    systemctl enable --now docker
```

安装docker-compose的Linux命令：

```
    curl -L https://github.com/docker/compose/releases/download/v2.20.3/
docker-compose-'uname -s'-'uname -m' -o /usr/local/bin/docker-compose
    chmod +x /usr/local/bin/docker-compose
```

这是Docker官方提供的一键安装脚本，会自动适配不同的Linux发行版（通过检测系统内核和包管理器），无论是Debian系（Ubuntu、Debian）还是RHEL系（CentOS、Fedora）都能兼容。

通过执行以下标准化验证命令，确认Docker和Docker Compose的安装完整性。

```
    docker -v
    docker-compose -v
```

如图1-4所示，若上述命令成功执行并返回详细的版本信息及系统状态，则表明Docker和Docker Compose 已正确安装并具备完整的运行能力。至此，Linux系统已就绪，这里假设你有基本的Linux操作经验。

```
root@VM-20-5-ubuntu:~# docker -v
Docker version 27.3.1, build ce12230
root@VM-20-5-ubuntu:~# docker-compose -v
docker-compose version 1.28.4, build cabd5cfb
```

图1-4　Docker版本验证

1.4.2　核心配置文件config.json

首先创建一个专用的工作目录用于存放FastGPT的配置文件，然后使用wget从GitHub下载获取FastGPT系统运行所需的核心配置文件config.json，如图1-5所示，并把它存放在fastgptdocker的工作目录下，config.json下载地址为：https://raw.githubusercontent.com/labring/FastGPT/refs/heads/main/projects/app/data/config.json。

```
root@VM-20-5-ubuntu:/fastgptdocker# wget https://raw.githubusercontent.com/labring/FastGPT/refs/heads/main/projects/app/data/config.json
--2025-08-16 18:19:18--  https://raw.githubusercontent.com/labring/FastGPT/refs/heads/main/projects/app/data/config.json
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1142 (1.1K) [text/plain]
Saving to: ‘config.json’

config.json                         100%[=======================================================================================>]
s

2025-08-16 18:19:36 (32.9 MB/s) - ‘config.json’ saved [1142/1142]
```

图1-5　下载config.json

另外，还需要Docker Compose这个用于定义和运行多容器Docker应用的工具。它通过一个YAML文件（docker-compose.yml）来配置应用程序的所有服务。

1.4.3　向量数据库的配置

我们在RAG企业知识库系统中使用的是向量数据库，FastGPT系统支持三种经过生产环境验证的向量数据库解决方案，需要根据具体的业务需求、数据规模、性能要求和运维能力进行技术方案选择。下面介绍FastGPT支持的三种向量数据库解决方案。

1．PgVector解决方案（轻量级部署推荐）

- 适用场景分析：知识库向量索引量在5000万条以下的中小规模部署。
- 技术特点：基于PostgreSQL数据库的向量扩展，具备ACID事务特性和成熟的生态系统。
- 性能特征：单节点部署，查询延迟在毫秒级别，适合对实时性要求较高的应用场景。
- 运维复杂度：低，可复用现有PostgreSQL运维经验和工具链。

配置文件下载地址为：https://github.com/labring/FastGPT/blob/main/deploy/docker/docker-compose-pgvector.yml。

2．Milvus解决方案（高性能大规模部署）

- 适用场景分析：亿级以上向量数据处理，对查询性能有极高要求的企业级部署。
- 技术特点：专门设计的向量数据库，支持分布式架构和多种向量索引算法。
- 性能特征：支持水平扩展，可处理百亿级向量数据，毫秒级查询响应。
- 运维复杂度：中等，需要专门的Milvus运维知识和监控体系。

配置文件下载地址为：https://github.com/labring/FastGPT/blob/main/deploy/docker/docker-compose-milvus.yml。

3．Zilliz Cloud解决方案（企业级SaaS服务）

- 适用场景分析：需要SLA保障、专业技术支持和托管服务的企业级部署。
- 技术特点：基于Milvus的完全托管服务，提供企业级的可靠性和安全性保障。
- 性能特征：按需弹性扩展，99.9% 可用性保证，全球多区域部署能力。
- 运维复杂度：最低，由 Zilliz提供专业的托管运维服务。

配置文件下载地址为：https://github.com/labring/FastGPT/blob/main/deploy/docker/docker-compose-zilliz.yml。

如图1-6所示，针对使用不同的知识库向量数据库，下载不同的配置文件，PgVector适合知识库索引量在5000万以下，Milvus适合亿级以上的索引量（知识数据条数）。

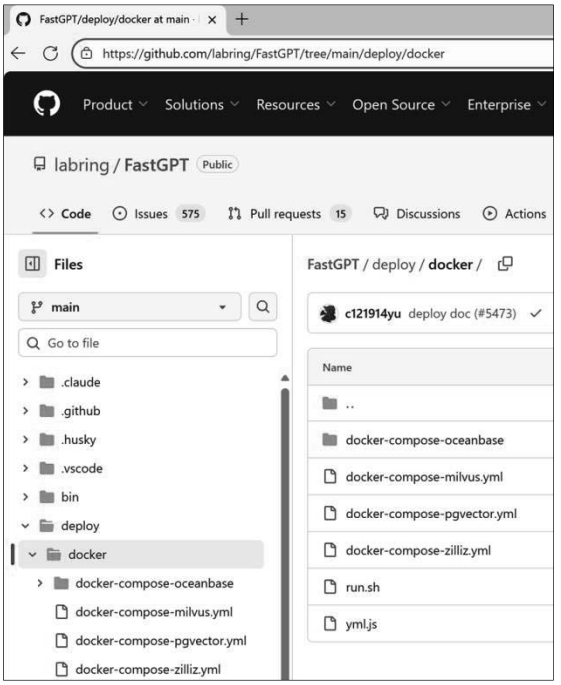

图1-6　向量数据库配置文件

如图1-7所示，此处我们以PgVector为例，使用wget命令下载docker-compose-pgvector.yml并保存文件名为docker-compose.yml。

```
root@VM-20-5-ubuntu:/fastgptdocker# wget https://github.com/labring/FastGPT/blob/main/deploy/docker/docker-compose-pgvector.yml -O docker-compose.yml
--2025-08-16 18:19:41--  https://github.com/labring/FastGPT/blob/main/deploy/docker/docker-compose-pgvector.yml
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘docker-compose.yml’

docker-compose.yml                      [        <=>                                                                  ] 199.23K  94.5KB/s
.1s

2025-08-16 18:19:44 (94.5 KB/s) - ‘docker-compose.yml’ saved [204008]
```

图1-7　下载docker-compose-pgvector.yml并保存为docker-compose.yml

1.4.4　配置Docker下载的镜像

使用Docker部署程序服务时，可能会遇到拉取镜像时出现报错mysql Error Get“https://registry-1.docker.io/v2/”的问题。这时需要添加国内镜像代理。我们可以使用命令sudo vi /etc/docker/daemon.json创建daemon.json，并加入如下内容（需要有root权限或者sudo权限）：

```
{
"registry-mirrors" : ["https://docker.registry.cyou",
"https://docker-cf.registry.cyou",
"https://dockercf.jsdelivr.fyi",
"https://docker.jsdelivr.fyi",
"https://dockertest.jsdelivr.fyi",
"https://mirror.aliyuncs.com",
"https://dockerproxy.com",
"https://mirror.baidubce.com",
"https://docker.m.daocloud.io",
"https://docker.nju.edu.cn",
"https://docker.mirrors.sjtug.sjtu.edu.cn",
"https://docker.mirrors.ustc.edu.cn",
"https://mirror.iscas.ac.cn",
"https://docker.rainbond.cc",
"https://do.nark.eu.org",
"https://dc.j8.work",
"https://dockerproxy.com",
"https://gst6rzl9.mirror.aliyuncs.com",
"https://registry.docker-cn.com",
"http://hub-mirror.c.163.com",
"http://mirrors.ustc.edu.cn/",
"https://mirrors.tuna.tsinghua.edu.cn/",
"http://mirrors.sohu.com/"
],
 "insecure-registries" : [
    "registry.docker-cn.com",
    "docker.mirrors.ustc.edu.cn"
    ],
"debug": true,
"experimental": false
}
```

修改完成后，重启Docker服务：

```
sudo systemctl restart docker
```

1.4.5　使用Docker命令安装FastGPT

如图1-8所示，运行命令docker-compose up -d即可，Docker根据docker-compose.yml内容自动下载FastGPT及其依赖的镜像进行安装。

安装FastGPT的具体耗时取决于网络环境和镜像下载速度，有时会长达数小时。最后结果如图1-9所示，表示已经安装成功。

```
root@VM-20-5-ubuntu:/fastgptdocker# docker-compose up -d
Building with native build. Learn about native build in Compose here: https://docs.docker.com/go/compose-native-build/
Pulling pg (pgvector/pgvector:0.8.0-pg15)...
0.8.0-pg15: Pulling from pgvector/pgvector
b1badc6e5066: Pull complete
423d9f34d3bd: Pull complete
3d9fd0212ce5: Pull complete
1e4441c36144: Pull complete
9726ad63f63c: Pull complete
ff3a6fab2ee2: Pull complete
6c27868070f0: Pull complete
ea1c15573ccb: Pull complete
aef740ab9f16: Pull complete
216cb0a155c1: Pull complete
6a9e54677c18: Pull complete
060e571be26f: Pull complete
4109b493cd40: Pull complete
1a2bdff2205a: Pull complete
9ea82a83a355: Pull complete
9b811989bf1c: Pull complete
Digest: sha256:1dec32aaba982ea73c51e93bee0319abf54150110321e24101c3d9de0b88db98
Status: Downloaded newer image for pgvector/pgvector:0.8.0-pg15
Pulling mongo (mongo:5.0.18)...
```

图 1-8　使用 Docker 命令安装 FastGPT

```
Creating sandbox            ... done
Creating fastgpt-minio      ... done
Creating redis              ... done
Creating pg                 ... done
Creating fastgpt-mcp-server ... done
Creating aiproxy_pg         ... done
Creating mongo              ... done
Creating fastgpt            ... done
Creating aiproxy            ... done
Creating fastgpt-plugin     ... done
root@VM-20-5-ubuntu:/fastgptdocker#
```

图 1-9　安装 FastGPT 成功

如图1-10所示，通过命令docker -ps -a可以看到容器正常运行。Docker Compose将按照配置文件中定义的依赖关系顺序启动各个服务容器，包括数据库服务、向量存储服务、应用服务等。

```
root@VM-20-5-ubuntu:/fastgptdocker# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED         STATUS
               NAMES
905d7c50a4cb   ghcr.io/labring/fastgpt-plugin:v0.1.10   "docker-entrypoint.s…"   4 minutes ago   Up 4 minutes
               fastgpt-plugin
28ff1dcdff44   ghcr.io/labring/aiproxy:v0.2.2       "aiproxy"                5 minutes ago   Up 5 minutes (healthy)
               aiproxy
ea0cd2634c42   ghcr.io/labring/fastgpt:v4.12.1-fix  "sh -c 'node --max-o…"   5 minutes ago   Up 4 minutes
               fastgpt
129388c81e64   pgvector/pgvector:0.8.0-pg15          "docker-entrypoint.s…"   5 minutes ago   Up 5 minutes (healthy)
               pg
cceddebb8754   pgvector/pgvector:0.8.0-pg15          "docker-entrypoint.s…"   5 minutes ago   Up 5 minutes (healthy)
               aiproxy_pg
```

图1-10　启动容器

1.4.6　系统访问与登录

部署完成后，如果要能访问FastGPT管理后台，需要在云主机中添加访问规则，如图1-11所示，添加一条允许访问TCP的3000端口的规则。

图1-11　添加访问规则

如图1-12所示，通过浏览器访问以下地址进入FastGPT管理界面：http://云主机公网IP:3000/。

系统初始部署后，管理员用户名为root，默认密码是1234。出于系统安全考虑，部署完成后必须立即修改默认密码。推荐的密码策略包括最小长度为12位字符，密码的复杂度要求包含大小写字母和数字。

如图1-13所示，如需修改FastGPT的root管理员用户密码，需要修改docker-compose.yml文件中的相关环境变量DEFAULT_ROOT_PSW。

图 1-12　访问管理界面

```
fastgpt:
  container_name: fastgpt
  image: ghcr.io/labring/fastgpt:v4.12.1-fix # git
  # image: registry.cn-hangzhou.aliyuncs.com/fastgpt/fastgpt:v4.12.1-fix # 阿里云
  ports:
    - 3000:3000
  networks:
    - fastgpt
  depends_on:
    - mongo
    - sandbox
    - pg
  restart: always
  environment:
    # 前端外部可访问的地址，用于自动补全文件资源路径。例如 https:fastgpt.cn，不能填 localhost。
    - FE_DOMAIN=
    # root 密码，用户名为：root。如果需要修改 root 密码，直接修改这个环境变量，并重启即可。
    - DEFAULT_ROOT_PSW=1234
```

图 1-13　修改账号密码

使用vim修改docker-compose.yml文件中的环境变量DEFAULT_ROOT_PSW的值并保存退出后，如图1-14所示，执行命令docker-compose down，再执行docker-compose up -d命令重启容器，配置文件修改生效，这时就可以用新的密码登录FastGPT管理界面了。

```
root@VM-20-5-ubuntu:/fastgptdocker# docker-compose up -d
Building with native build. Learn about native build in Compose here: https://docs.docker.com/go/compose-native-build/
Creating network "fastgptdocker_fastgpt" with the default driver
Creating fastgpt-minio      ... done
Creating fastgpt-mcp-server ... done
Creating sandbox            ... done
Creating pg                 ... done
Creating redis              ... done
Creating mongo              ... done
Creating aiproxy_pg         ... done
Creating fastgpt            ... done
Creating aiproxy            ... done
Creating fastgpt-plugin     ... done
root@VM-20-5-ubuntu:/fastgptdocker#
```

图1-14　重启容器

1.5　FastGPT模型管理

登录 FastGPT 后，系统将提示用户未配置语言模型，用户需要配置语言模型和索引模型以确保系统正常运行。如果我们要使用火山引擎的豆包大模型，必须拥有火山方舟的API Key，并且需要开通豆包大模型。如图1-15所示，登录火山方舟控制台，在火山方舟的API Key管理页面上创建API Key。

接下来开通豆包Doubao-1.5-pro-256k语言模型，如图1-16所示，记住Model ID为“Doubao-1.5-pro-256k-250115”。注意，如果还想使用其他模型，相应地按这里介绍的方法开通即可。

图 1-15　在火山方舟上创建 API key

图 1-16　Doubao-1.5-pro-32k 模型

如图1-17所示，在火山方舟控制台的“开通管理”模块，开通Doubao-1.5-pro-256k大模型。

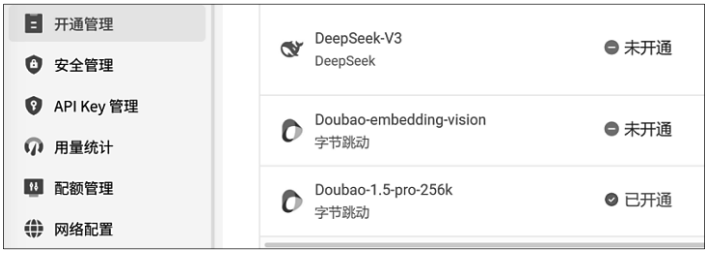

图1-17　开通Doubao-1.5-pro-256k大模型

按以上操作，开通文本嵌入模型（Text Embedding Model）Doubao-embedding-large，文本嵌入模型的核心功能是将非结构化文本（如句子、段落、文档）转换为结构化的嵌入向量（Embedding Vector），向量会保留文本的语义信息，这是后续文本类人工智能机器学习任务的关键基础。

登录FastGPT的ROOT管理界面（后面简称主页），单击右下角的“账号”选项，打开如图1-18所示的页面，选择“模型提供商”，再选择“模型渠道”，单击界面右上角的“新增渠道”按钮。

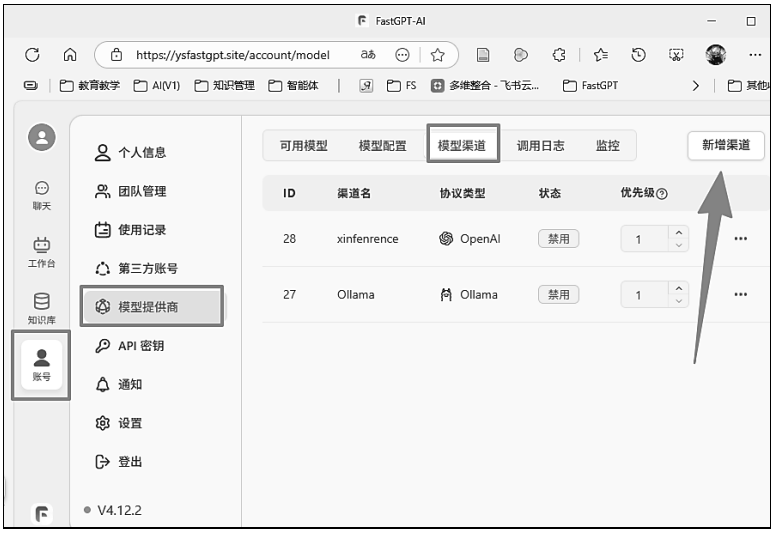

图1-18　模型渠道

如图1-19所示，在渠道配置中填写渠道名、协议类型、模型。模型至少配置两个，一个是嵌入模型（也叫索引模型）Doubao-embedding-large，另一个是语言模型Doubao-1.5-pro-256k。模型映射按图1-19填写，映射到它在火山方舟的Mode ID。

图1-19　豆包大模型渠道配置

模型渠道配置完成后，如图1-20所示，执行模型测试以验证配置的正确性。

图1-20　执行模型测试以验证配置的正确性

如图1-21所示，执行批量测试，状态栏显示成功即可正常使用模型。

图1-21　执行批量测试

如图1-22所示，完成上述配置后，在模型配置中启用已创建Doubao的语言模型Doubao-1.5-pro-256k，同理操作，启用豆包索引模型。

图1-22　启用模型

如果你需要调用各大厂商的AI模型，每次调用模型都要在不同的平台间反复横跳，在调用来自不同厂商的模型时，需要在这些厂商的计费平台上进行支付，在不同平台上管理和监控API使用情况，特别麻烦。Sealos AI Proxy这个“超级模型调用平台”，让你可以在同一个平台中轻松调用各类AI模型。我们登录Sealos官网（https://gzg.sealos.run/），如图1-23所示，这里可用区选择Guangzhou广州（Guangzhou G）。

图1-23　Sealos官网

如图1-24所示，在Sealos官网单击“AI Proxy”图标申请API key，记得先要实名认证。单击左边栏的“API Keys”，新建一个API密钥，如图1-25所示，创建了一个API Key密钥，形如“sk-xxxxxxxxxxx”字符串。

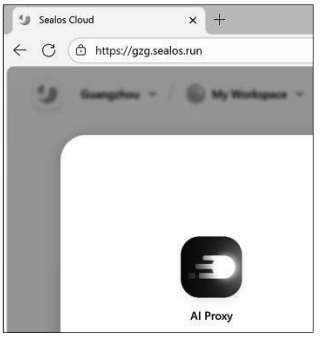

图1-24　AI Proxy图标

还是在FastGPT主页的“模型提供商”→“模型渠道”界面上新增渠道，如图1-26所示，渠道名填写Sealos，这里添加AI Proxy模型广场中提供的阿里通义千问语言模型qwen-plus和嵌入模型（也叫索引模型）text-embedding-v4，协议类型为OpenAI。由于API Keys申请的是广州区域gzg，因此代理地址应是广州的https://aiproxy.gzg.sealos.run/v1。

渠道配置完毕后，在模型渠道页面就可以看到新增的渠道名Sealos，如图1-27所示，右击出现下拉菜单，单击“模型测试”。

图1-25　API Keys

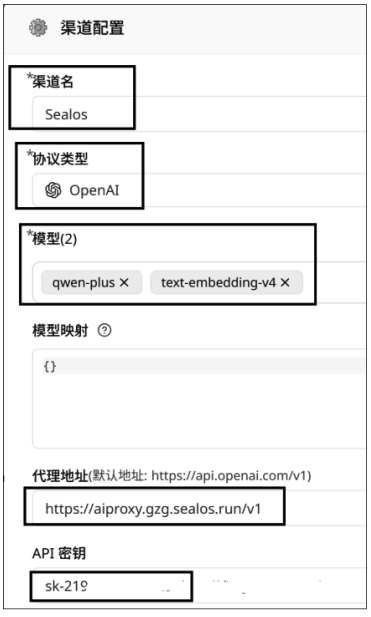

图 1-26　渠道配置

图 1-27　模型测试

如图1-28所示，单击“批量测试2个模型”按钮，可以看到模型测试成功。

图1-28　模型测试成功

如图1-29所示，在“模型配置”选项卡中，启用text-embedding-v4模型。同理操作，启用qwen-plus语言模型。

图1-29　“模型配置”选项卡

如图1-30所示，在“可用模型”选项卡中可以看到目前启用了4个模型。

图1-30　可用模型

接下来，我们新建一个AI应用，如图1-31所示，在FastGPT工作台页面，单击右上角的“+新建”按钮，在打开的菜单中单击“简易应用”。

图1-31　在打开的菜单中单击“简易应用”

如图1-32所示，在打开的“创建简易应用”对话框中，输入名字test，选择“创建空白应用”。

如图1-33所示，我们可以输入“什么是文本转向量”，测试一下模型功能。如果确认测试成功，则表示整个模型管理配置流程已完成。

图1-32　创建空白应用

图1-33　测试模型功能

通过以上配置流程，用户可以快速地进行模型管理，并确保系统具备完整的语言处理和索引检索能力。

第 2 章

从大模型到AI Agent

本章将简单介绍大模型、token、提示词、Agent等AI开发必须理解的概念，并探讨了Agent落地企业应用的要点。

2.1 什么是大模型

在某个深夜，一位程序员对着计算机屏幕愣住了。他向一个AI系统输入：“用《静夜思》的风格写一段代码注释”，几秒钟后得到的回复让他寒毛直竖：

屏幕光寒，代码未眠。
一行向北，两行向南。
键盘声里，故乡渐远。
编译通过，泪落指尖。

这不是诗歌创作大赛的获奖作品，而是大语言模型对人类情感与文字规律的跨维度模仿。当机器开始“理解”乡愁时，我们不得不重新定义：什么是智能？什么是大模型？

大模型的本质，是用数学方法模拟人类认知规律的巨型神经网络。如果把人类大脑比作历经百万年进化的生物计算机，大模型就是人类用硅基材料搭建的“人工大脑”——它没有神经元，却有百亿级参数构成的计算节点；它不会呼吸，却能通过海量数据训练形成对世界的“认知框架”。

1956年达特茅斯会议上，“人工智能”概念首次被提出时，研究者们坚信规则可以定义智能。他们手工编写上万条逻辑指令，试图让机器学会推理，却在“识别一只猫需要多少条规则”的问题上陷入僵局。

大模型的革命性突破在于放弃了对规则的穷举，转而向数据学习。就像婴儿通过观察父母的语言学会说话，大模型通过阅读互联网上的万亿级文字、图片、视频，在参数空间中默默构建着世界的运行规律。

以GPT系列模型为例，训练它的语料库相当于将人类文明史上所有书籍复印一百万次——从莎士比亚的十四行诗到量子物理论文，从微博上的早餐分享到联合国安理会决议，这些看似杂乱的信息在模型中凝结成隐形的“知识网络”。当你问“为什么夏天白天比冬天长”时，它不需要调用天文学数据库，而是通过训练中习得的“地球公转“黄赤交角”等概念关联，生成符合科学逻辑的答案。这种

学习方式暗藏着深刻的哲学隐喻：智能或许不是被设计出来的，而是从足够复杂的信息交互中涌现的。

在科技报道中，我们常看到“千亿参数大模型”这样的表述。参数究竟是什么？如果把大模型比作一台精密的钢琴，参数就是琴弦的张力——每个参数都是一个可调节的数值，共同决定着模型对信息的处理方式。

百亿级参数的神奇之处，在于它们能捕捉人类难以言说的隐性知识。例如，当我们说“他像块木头”时，模型能瞬间理解这是在形容人反应迟钝，而非字面意义上的材质对比。这种对隐喻、语境、文化背景的把握，源于参数在训练中形成的“语义向量空间”——每个词语都被转换为高维空间中的一个点，词语间的距离代表含义的关联度，“木头”与“迟钝”的向量距离，比“木头”与“石头”更近。

大模型的所有输出，本质上是对训练数据中统计规律的概率性模仿。当面对未知问题时，模型会虚构出看似合理的答案。例如，某医疗大模型曾将普通感冒误诊为罕见病，只因训练数据中该病症的描述与普通感冒有5%的相似度。这提醒我们：大模型的“智能”始终建立在数据的地基上，而数据的偏见与缺失，会成为大模型智能的“阿喀琉斯之踵”。

大模型不是替代人类的竞争者，而是扩展人类认知边界的脚手架。就像显微镜让人类看见细胞，望远镜让人类望向星空一样，大模型让人类得以在海量信息中捕捉隐藏的规律，在跨学科的知识海洋中搭建新的认知桥梁。

当我们站在智能革命的门槛上，理解大模型的本质，就是理解人类自身的认知密码。它既是人类智慧的结晶，也是一面映照人性的镜子——在这面镜子里，我们看到的不仅是技术的可能性，更是文明进化的新路径。

2.2　什么是token

在大模型的世界里，token是一个基础且关键的概念，你可以把它想象成大模型用来理解和处理语言的“积木块”。当我们向大模型输入一段文本，无论是一句话、一篇文章，还是一整本书，大模型不会像人类一样逐字逐句地“阅读”，而是会先把这段文本拆分成一个个更小的部分，这些小部分就是token。

token并没有固定的形式，它可能是一个完整的单词，比如apple（苹果），也可能是一个单词的一部分，即子词，例如playing可以被拆分成play和ing，这里ing就是一个子词token，这种拆分方式有助于模型处理各种词汇变化形式。标点符号同样能作为token，像逗号“,”、句号“.”等，它们在文本中传递着停顿、结束等关键信息，大模型也需要对其单独识别。

为什么大模型要把文本拆分成token来处理呢？这是因为自然语言极其复杂多样，直接处理一长串连续的文本对模型来说难度太大。将文本转换为token，就像是把混乱的拼图碎片分类整理，每个碎片（token）都携带一部分语义信息，模型可以更高效地对这些标准化的小块进行分析和处理。而且，通过合理定义token，大模型能够构建一个相对固定大小的“词汇表”，即便遇到生僻词或新造词，也能通过子词等方式将其拆解为已知token的组合，从而理解和处理。

例如，当我们输入“我喜欢吃苹果”这句话时，大模型的分词器（负责把文本拆分成token的工具）可能会将其拆分成“我”“喜欢”“吃”“苹果”这4个token。不同的大模型可能采用不同的分词算法，有些模型可能会把某些常用词组当作一个整体token处理，像“人工智能”，有的模型会将其视为

一个token，有的则可能拆成“人工”和“智能”两个token。在英文中，基于空格分词是常见的方式，像“I love reading books”会被拆分成“I”“love”“reading”“books”这4个token，但遇到像“didn't”这样的缩写词，可能会进一步拆分成“did”和“n't”，以使模型更好地理解其语义构成。

token在大模型处理流程中扮演着不可或缺的角色。输入文本被转换为token序列后，这些token会被进一步转换为向量形式（也就是一系列数字组成的数组），向量能够反映每个token的语义特征，以及它们之间的关联。模型通过对这些向量进行复杂运算来理解文本含义，并根据学习到的语言模式和知识，预测下一个可能出现的token，进而生成连贯的文本输出。可以说，token是大模型与自然语言之间的“翻译官”，是实现强大语言处理能力的基础单元，其处理方式的优劣直接影响着大模型对语言理解和生成的准确性与效率。

大模型按token计费，是因为token数量与模型计算资源消耗直接相关，这种计费方式能够精准计量成本，为用户提供公平透明的计费体验，同时也有助于大模型厂商实现商业可持续性，具体如下。

- 精准计量资源消耗：大模型运行需耗费大量计算资源，而token数量直接反映了模型处理文本的工作量。模型每处理一个token，都要进行一系列复杂计算，如矩阵运算等。token越多，计算量越大，消耗的算力、内存等资源也就越多，相应的能耗和处理时间也会增加。因此，按token计费能准确反映模型实际使用的计算量，相比传统的按次收费，能更精准地计量资源使用情况。
- 公平合理计费：按token计费遵循“用多少付多少”的原则，避免了传统会员制或按次收费可能出现的不公平现象。例如，传统包月会员制中，轻度用户可能会补贴重度用户，而按token计费则让每个用户根据自己使用的token数量付费，偶尔使用大模型查资料的用户无须为频繁使用大模型写代码的用户分担费用，实现了更公平的计费方式。
- 优化用户成本与体验：按token收费可以激励用户优化输入，避免浪费。用户为了降低成本，会更倾向于简洁明了地表达问题，从而提升整体资源利用效率。此外，这种计费方式也为用户提供了灵活控制成本的可能性，开发者可根据业务需求选择模型，并通过压缩输入文本、限制输出长度等方式优化成本，如将输入提示词精简，可降低输入费用。
- 保障商业模式可持续性：大模型的研发、训练、部署和维护成本极高，如GPT-4耗资约1亿美元。按token收费能让企业根据用户的实际使用量调整资源，确保收入与成本相匹配，避免亏损，从而有足够的资金持续优化模型，推动技术发展，维持商业模式的可持续性。

2.3　什么是提示词

在大模型的交互中，提示词（Prompt）是用户输入给模型的文本指令，相当于给大模型的“任务说明书”。它是用户与大模型沟通的桥梁，直接决定了模型输出内容的方向、质量和风格。

简单来说，如果把大模型比作一位“全能助手”，提示词就是你向这位助手提出的“需求”。例如：

- 你想让大模型写一首关于春天的诗，提示词可以是“请写一首描绘春天景色的七言绝句”。
- 你想让大模型解释一个概念，提示词可以是“用通俗的语言解释什么是量子纠缠”。
- 你想让大模型帮你修改文案，提示词可以是“把这段产品介绍改得更活泼，适合年轻人阅读”。

1. 提示词的核心作用

提示词的本质是引导模型理解任务边界。大模型虽然能处理复杂问题，但它无法主动猜测用户的具体需求——你需要通过提示词明确告诉它：

- 做什么（生成内容、分析问题、翻译文本等）。
- 怎么做（用什么风格、遵循什么格式、侧重什么角度等）。
- 有什么限制（字数要求、避免哪些内容、参考什么案例等）。

例如，同样是让模型写一篇关于“人工智能”的文章，不同提示词会带来完全不同的结果：

- 模糊的提示词：“写一篇关于人工智能的文章”——模型可能泛泛而谈，内容空洞。
- 精准的提示词：“以‘人工智能如何改变医疗行业’为主题，写一篇800字的说明文，重点分析AI在疾病诊断和药物研发中的应用，举两个具体案例”——模型会聚焦医疗领域，结构清晰且有细节。

2. 提示词的“好坏”影响输出质量

优质的提示词往往具备明确性、具体性和引导性。

- 明确性：避免模糊词汇（如“写一篇好文章”中的“好”没有标准），而是用可量化的要求（如“1000字、分3个部分”）。
- 具体性：提供背景信息或参考示例（如“参考《三体》的科幻风格，写一段与外星文明接触的场景”）。
- 引导性：通过逻辑拆解帮助模型梳理思路（如“先分析这个政策的3个核心条款，再说说对中小企业的影响”）。

反之，模糊、笼统的提示词容易让大模型“跑偏”。例如提示词“谈谈教育”，模型可能从古代说到现代，内容散乱；但如果改为“谈谈在线教育对农村学生的影响，分优缺点两方面说”，输出就会更聚焦。

随着大模型能力的扩展，提示词的形式也在丰富：除了纯文本外，还可以包含图片（如“描述这张图片里的场景”）、语音转文字（通过语音输入需求），甚至结合代码片段（如“优化这段Python代码，提升运行效率”）。但核心逻辑不变——都是用户向模型传递“我需要什么”的信号。

总之，提示词是驾驭大模型的“方向盘”。想要让大模型发挥出更强的能力，关键在于学会用精准的提示词“告诉它该做什么”，这也是如今提示词工程（Prompt Engineering）成为一门独立技能的原因。

2.4　什么是Agent

了解了Transformer架构后，很多人下一个疑问是：大模型到底是怎么训练出来的？

在人工智能领域，Agent（智能体）是一个能自主感知环境、规划行动并执行任务的系统。简单来说，它像一个“有目标的执行者”——不仅能理解指令，还能主动思考“怎么做”，甚至在过程中调整策略，直到完成任务。

为什么有了大模型，还需要Agent？

大模型的核心能力是“理解与生成”（比如看懂文字、写出回答），但它有一个明显的局限：被动且短视。它像一本“活百科”，能回答问题、生成内容，但需要你一步步“喂”指令；如果任务复杂（比如“帮我规划一周的旅行，包括订机票、酒店，推荐小众景点”），大模型只能给零散建议，无法从头到尾“自己搞定”。

而Agent 恰好能弥补这些短板，相当于给大模型装上了“手脚”和“规划脑”。具体来说，它解决了大模型的三个关键问题。

1. 大模型不会主动规划，Agent能拆解复杂目标

大模型擅长处理单一、明确的指令（比如“写一封道歉信”），但面对需要多步骤的复杂任务时，就会“犯迷糊”。

例如，你让大模型“帮我准备明天的客户提案”，它可能只会说“需要包含产品优势、案例数据、报价表”，但不会告诉你“先查客户公司的最新动态→补充针对性案例→计算折扣方案→排版成PPT”。

而Agent会把“准备提案”这个大目标拆解成一系列子任务，甚至按优先级排序。

- 第一步：调用搜索引擎查客户最近的业务重点。
- 第二步：从公司数据库中调取相关合作案例。
- 第三步：根据客户规模计算合理报价。
- 第四步：用PPT工具生成文档并检查逻辑。

这种“目标拆解→步骤规划”的能力，是纯大模型不具备的。

2. 大模型“不会用工具”，Agent能链接外部资源

大模型的知识截至训练数据的时间（比如2023年），也没法直接操作计算机、调用软件。但现实中，很多任务必须依赖外部工具：查询实时天气需要调用天气API，计算复杂数据需要用Excel，发邮件需要连接邮箱系统。

例如，你让大模型“告诉我明天上海的天气，以及适合穿什么衣服”，如果没有工具，它只能说“无法获取实时数据”；但Agent可以自动调用天气API拿到温度、降水信息，再结合这些数据推荐穿搭。

在更复杂的场景中，Agent甚至能组合工具：例如“分析过去3个月公司的销售数据，生成可视化图表，然后发到团队群”，它会先调用数据库获取数据，再用Python绘图，最后调用企业微信API发送，全程无须人工干预。

3. 大模型“没有长期记忆”，Agent能持续迭代优化

大模型的对话是“单次交互”，每次对话结束后，它不会记得你们之前聊了什么（除非你把历史记录重新“喂”给它）。这导致它无法处理“需要长期跟进”的任务。

例如，你让大模型“跟踪一个项目的进度”，第一次它能列个计划，但下周你问“上周的任务完成了吗”，它会一脸茫然。而Agent有“记忆模块”，能记录任务进展。

- 周一：记录“需求文档未提交”。
- 周三：收到文档后，更新状态为“已完成，开始设计”。
- 周五：发现设计稿有问题，自动提醒负责人修改，并更新进度。

甚至，Agent 能从历史中学习。例如，上次订酒店时选错了位置，下次会优先过滤掉同区域的选项，相当于“越用越聪明”。

如果把大模型比作一个聪明的大脑（擅长思考、理解），那么Agent就相当于一个能自主行动的人——它用大模型的理解能力作为基础，再加上“规划能力”“工具使用能力”“记忆能力”，从而能独立完成从“接任务”到“交结果”的全流程。

随着AI需要解决的问题越来越贴近真实生活（比如自动办公、智能管家、复杂决策），仅靠“被动回答”的大模型远远不够。Agent的价值正是让AI从“能说会道”进化为“能动手做事”，真正成为人类的“全能助手”。

2.5　Agent落地企业应用的要点

Agent在企业级场景的落地，核心是解决流程效率低、人力成本高等实际痛点，与To C场景相比，企业级Agent更强调“深度嵌入业务流程”“对接业务系统”“满足合规要求”，最终实现从“辅助决策”到“自主执行”的闭环。

1. 企业级Agent落地的核心逻辑：从“替代重复劳动”到“赋能复杂决策”

企业的核心需求是“降本（减少重复劳动）、增效（加速流程周转）、提质（提升决策准确性）”。Agent落地的逻辑是通过“感知企业数据→理解业务规则→调用工具执行→优化流程闭环”，逐步渗透到企业的“作业层、管理层、决策层”。

- 作业层：替代人工完成重复性高、规则明确的任务（如数据录入、工单处理）。
- 管理层：整合跨部门数据，生成标准化报告（如销售复盘、库存预警）。
- 决策层：结合历史数据和实时信息，提供风险预判或方案建议（如供应链优化、市场拓展决策）。

2. 企业级Agent的典型落地场景

1）客户服务：从“被动响应”到“主动服务”

传统客服依赖人工处理工单，效率低且易出错。Agent可以：

- 自动接单，一分类一处理：通过对接企业CRM系统，识别客户问题（如“查询订单进度”“投诉物流”），自动调取订单数据、物流信息，生成标准化回复。
- 主动触达：当系统检测到“客户订单延迟”“产品即将到期”时，自动触发提醒（如短信、邮件通知），并同步推送解决方案（如补寄流程、续费优惠）。
- 复杂问题升级：无法解决的问题（如“定制化需求咨询”），自动转接对应客户经理，并同步客户历史交互数据（避免重复沟通）。

具体案例：某电商企业的“售后Agent”，通过对接ERP（订单数据）、WMS（仓储信息）、客服工单系统，将售后工单处理效率提升60%，人工介入率从80%降至30%。

2）供应链与库存管理：动态优化供需平衡

供应链涉及采购、仓储、物流等多个环节，数据分散且实时性要求高。Agent可以：

- 实时监控与预警：对接库存系统（如SAP）、采购系统、销售数据，当某类商品库存低于阈值时，自动计算补货量（结合历史销量、季节等因素），生成采购申请单并推送给采购部门。

- 异常处理：若供应商延期交货，自动调取备选供应商信息，对比价格和交付周期，生成替代方案。
- 端到端追溯：当客户反馈“产品质量问题”时，自动追溯原材料批次、生产环节、物流路径，定位问题节点并生成报告。

具体案例：某快消企业的“供应链Agent”，通过整合销售预测模型、库存数据、供应商API，将库存周转天数从45天缩短至32天，缺货率降低25%。

3）财务与人力资源：自动化流程 + 合规校验

财务和HR领域规则密集、合规性要求高，Agent可以：

- 财务自动化：对接ERP（账务数据）、银行系统、发票系统，自动完成“发票验真→记账→对账→生成报销凭证”，并校验是否符合公司报销规则（如差旅住宿标准）。
- HR流程优化：新员工入职时，自动触发“工牌申请→邮箱开通→社保缴纳”等流程，对接OA系统发送待办任务给相关部门，同步更新员工档案。
- 合规审计：定期扫描财务数据，识别异常交易（如超预算支出、重复报销），生成审计报告并标记风险点。

具体案例：某制造企业的“财务Agent”，将月度结账时间从7天压缩至3天，合规检查覆盖率从60%提升至100%，减少90%的人工核对工作。

4）研发与项目管理：协同效率提升

研发项目涉及代码管理、需求迭代、进度跟踪，Agent可以：

- 需求拆解与排期：将“产品功能需求”拆解为具体研发任务（如“前端页面开发”“后端接口设计”），结合团队成员工作量自动分配任务并设置截止日期。
- 进度跟踪与风险预警：对接 Git（代码提交）、Jira（任务管理），当某任务延期时，自动提醒负责人，若影响整体进度，则同步生成资源协调建议（如调配其他团队支持）。
- 文档自动化：根据代码注释、测试报告，自动生成产品说明书、API文档，并同步更新到企业知识库。

3. 企业级Agent落地的关键能力（与To C的核心差异）

To C Agent更注重“自然交互体验”，而企业级Agent必须满足“系统适配性、数据安全性、流程可控性”三大核心要求，具体包括如下能力。

1）深度对接企业私有系统

企业数据分散在ERP、CRM、OA、数据库等私有系统中，Agent需要通过API、中间件或私有化部署，安全访问这些系统（而非依赖公开数据）。例如，调用企业内部的客户标签系统，实现精准服务；读取生产数据库，监控设备运行状态。

2）严格的权限与合规管理

企业数据涉及商业机密（如客户信息、财务数据），Agent必须具备“细粒度权限控制”：不同部门的Agent只能访问对应权限的数据（如销售Agent看不到财务数据）；所有操作留痕（如“谁调用了什么数据”“修改了什么参数”），满足审计和合规要求（如GDPR、国内数据安全法）。

3）可定制的流程编排能力

企业业务流程高度个性化（如不同行业的报销规则、审批流程差异极大），Agent需要支持“低代码配置”：业务人员无须编程，即可通过可视化界面定义任务步骤（如“当A条件触发时，执行B操作，调用C工具”），快速适配业务变化。

4）人机协同而非“替代人”

企业级任务往往涉及复杂决策（如大额合同审批、危机处理），Agent的定位是“辅助人”而非“替代人”：

- 常规步骤自动完成，关键节点推送给人工确认（如“采购金额超100万时，暂停自动流程，通知经理审批”）。
- 当Agent遇到无法处理的异常（如系统故障、规则冲突）时，自动触发“人工接管”机制，避免流程中断。

4. 落地路径：从“单点试点”到“全域渗透”

企业引入Agent通常遵循“小步快跑、逐步扩展”的路径，避免一次性投入过大导致失败。

第一步：选准高频痛点场景试点。优先选择“规则明确、重复性高、数据相对集中”的场景（如客服工单处理、发票报销），快速验证价值。例如，某企业先让Agent处理“标准化售后咨询”，成功后再扩展到“个性化需求处理”。

第二步：打通核心系统，构建数据闭环。试点成功后，逐步对接更多企业系统（如从对接CRM扩展到对接ERP+物流系统），实现“数据输入→Agent处理→结果输出→数据反馈”的闭环。例如，客服Agent处理完订单问题后，自动将处理结果同步回CRM，更新客户满意度标签。

第三步：跨场景协同，形成“Agent网络”。单一Agent难以解决跨部门复杂任务（如“新品上市”需要市场、销售、生产协同），此时可构建“多Agent协作网络”：

- 市场Agent负责竞品分析，生成推广建议。
- 销售Agent根据建议制定客户触达计划。
- 生产Agent根据销售预测调整产能。
- 总控Agent协调各Agent进度，确保目标一致。

5. 挑战与应对

企业级Agent落地并非易事，常见挑战及应对思路如下。

1）挑战1：系统集成复杂

老旧企业的系统可能是“legacy系统”（老旧、无标准API），难以对接。

应对：通过中间件或“屏幕抓取+RPA”（机器人流程自动化）技术，临时适配老旧系统；长期逐步替换为支持API的现代化系统。

2）挑战2：员工接受度低

员工担心“被Agent替代”而抵触使用。

应对：明确Agent是“减轻工作负担”（如减少重复录入），而非替代人；培训员工使用Agent的“人机协同”模式（如如何高效接管Agent的任务）。

3）挑战3：成本与ROI平衡

私有化部署、系统改造等可能成本较高。

应对：初期采用“混合部署”（核心数据私有部署，非敏感功能云端调用）；优先计算“可量化收益”（如节省的人力成本、提升的订单量），确保投入产出比为正。

6. 总结

Agent在企业落地的核心价值是将大模型的“理解与生成能力”转换为“业务执行能力”，通过嵌入真实业务流程，解决企业“效率、协同、成本”的痛点。与To C场景相比，它更强调“安全、可控、定制化”，落地路径需从“单点试点”逐步扩展到“全域协同”。未来，随着企业数字化程度加深，Agent将成为连接“数据、系统、人”的核心枢纽，推动企业从“数字化”迈向“智能化”。