计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240900022

大模型驱动的 AI 应用服务平台

梁秉豪　张传刚　袁明明
浪潮通信信息系统有限公司　济南 250013

摘　要　随着企业数智化转型的持续推进,人工智能技术已经开始应用到企业内部管理、经营分析和生产效率提升等各个方面。然而,传统的 AI 应用研发流程涉及数据采集、数据清洗、特征提取、算法建模和应用研发等多个环节。整体技术门槛高,团队成员协作难,硬件资源利用率低,难以支撑数智化业务需求的敏捷落地。针对上述问题,提出了一套基于预训练大模型的 AI 应用服务平台。该平台主要面向 AI 应用研发和运营全过程管理进行设计,大幅降低了团队协作和资产管理难度。针对预备态、设计态和运行态中的核心流程,引入了预训练大模型和低代码技术,通过构建标注大模型、测试大模型和运营大模型,提升了 AI 应用的研发效率,同时实现了对运营数据的实时分析,保障了用户的使用体验,并大幅提升了硬件资源的利用率。

关键词:AI 应用;研发服务;智能标注;自动测试;运营服务
中图分类号　TP311

Large Model Driven AI Application Service Platform

LIANG Binghao,ZHANG Chuangang and YUAN Mingming
Inspur Communication Information System Co.,Ltd.,Jinan 250013,China

Abstract　With the continuous advancement of the transformation of enterprise data intelligence,artificial intelligence technology has begun to be applied to various aspects of enterprise internal management,operation analysis and production efficiency improvement. However,the traditional AI application research and development process involves data acquisition,data cleaning,feature extraction,algorithm modeling,and application research and development. The overall technical threshold is high,the collaboration of team members is difficult,the utilization rate of hardware resources is low,and it is difficult to support the agile landing of digital intelligent business requirements. To solve these problems,a set of AI application service platform based on pre-trained large model is proposed. The platform is mainly designed for AI application research and development and operation management,which greatly reduces the difficulty of team collaboration and asset management. For the core processes in the preparation state,design state and running state,the pre-trained large model and low-code technology are introduced. By constructing the labeled large model,the test large model and the operation large model,the research and development efficiency of AI application is improved. Meanwhile,the real-time analysis of operational data is realized,the user experience is guaranteed and the utilization rate of hardware resources is greatly improved.

Keywords　Artificial intelligence application,Research and development services,Intelligent annotation,Automatic test,Operation service

1　引言

随着“新一代信息技术产业”被列入了国家战略性新兴产业体系,电信运营商作为其中的“主力军”,正在发挥着科技创新主体作用。国内三大运营商积极布局算力网络和人工智能等关键技术,同时利用自身资源和渠道优势,打造“云+网+AI”的全栈数字化解决方案,赋能千行百业。在此过程中,如何充分发挥自身云网资源优势,利用海量数据资源和用户触点,加快自身数智化转型并对外输出自身能力,面临着较大挑战。2023 年 12 月,中国信息通信研究院发布了 2024 年信息通信业十大趋势,其中提到了“AI 大模型能力持续跃升,全面构筑智能化新底座”。

当前,通信行业 AI 应用研发主要还是处于传统小模型研发阶段,研发模式以瀑布式为主,中间涉及环节多,包括数据采集、数据清洗、特征提取、算法建模和应用研发等,不同团队成员之间协作困难,研发和运营所需要的硬件、数据和算法等资源分散在各个研发团队中。此外,传统 AI 应用研发需要具备较强的专业知识,AI 应用从需求到落地验证过程需要大量人力投入,自动化程度低,一线业务专家难以自行完成 AI 应用开发,影响企业创新效率。随着大规模预训练模型的爆发式发展,以往烟囱式的管理模式以及需求和代码驱动的研发模式,更加难以适应技术发展的需求。

针对上述问题,本文主要构建了一个面向通信行业的 AI 应用研发服务平台,提供了从智算资源管理、数据资产管理到

基金项目:泰山产业领军人才项目(tscx202312006);山东省博士后创新项目(SDCX-ZG-202400307)
This work was supported by the Taishan Industrial Leading Talent Project(tscx202312006) and Shandong Postdoctoral Innovation Project(SDCX-ZG-202400307).
通信作者:梁秉豪(liangbinghao@inspur.com)

研发过程管理的全流程服务。该平台可以快速汇聚开源和生态算法，融合海量业务数据，支持 AI 应用的敏捷迭代。平台通过引入预训练大模型等技术，进一步降低了 AI 应用研发门槛，有效支撑“AI 算法”“AI 模型”和“AI 应用”等各类产品的研发和运营需求。可视化编排和向导式配置的人机交互方式，大幅降低了 AI 应用的研发门槛，帮助各领域业务专家快速验证技术方案并实现基于业务驱动的 AI 应用敏捷开发。

2 相关工作

在 AI 应用研发过程中，主要涉及硬件资源调度、数据资源管理、算法训练推理、模型应用编排等主要环节。针对各环节中存在的问题，学术界和工业界都进行了大量的技术探索。

在硬件资源调度方面，业界主流方案主要通过 docker 和 kubernetes[1] 进行实现。其中，docker 将算法运行环境打包成镜像，通过容器方式进行虚拟化和环境运行；kubernetes 则是一个可弹性伸缩的分布式系统框架，方便对容器进行管理和调度，并提供资源挂载和自动部署等能力，大幅提升硬件资源管理的便捷性。此外，为提升 GPU 资源利用效率，还需要对 GPU 资源进行虚拟化和切分，目前主要采用直通方式和 API 转发方式进行实现。vGPU[2] 是 NVIDIA 针对自身硬件提供的虚拟化方案，可以支持图像处理和 AI 算法等场景。vCUDA[3] 是腾讯通过在 CUDA 层进行拦截和转发实现的虚拟化方案，支持通过 k8s 在单个 GPU 上运行多个 Pod。上述两种技术主要针对英伟达系列 GPU 进行实现，对于异构 AI 加速卡的统一虚拟化仍然存在较大困难。

在数据资源管理方面，开源的 label-studio，labelme 和 labelimg 是常用的计算机视觉算法训练数据标注工具，支持目标检测和图像分割等常规任务。OpenDataLab 开源的首款多模态数据标注工具 Label-LLM，主要面向大模型提供文本、图像、视频和音频等混合模态的标注能力。随着大模型技术的不断成熟，通过大模型完成标注信息生成和标注信息审核[4] 是目前的研究热点之一。针对特征数据存储和知识检索增强需求，向量数据库提供了较为完善的解决方案[5]。其中，Milvus 是首个开源的向量数据库，已广泛应用在人脸识别和图像检索等场景，其云原生分布式架构大幅提升了检索效率。Chroma 向量数据库兼容多种相似度度量标准，应用场景较为广泛，通过倒排索引和 KD-树等方式提升了向量搜索速度。

在算法训练推理方面，针对深度学习算法训练主要采用 PyTorch[6] 和 TensorFlow[7] 框架。对于边缘设备推理，英伟达 TensorRT[8] 提供了高性能的推理优化器和运行时加速库，在算力较低的移动端设备可以实现低延迟和高吞吐量的推理能力。面向大规模预训练模型推理，vLLM 通过 PagedAttention 技术[9]，优化了每次请求的键值对缓存效率，大幅提升了并发请求处理能力。

在模型应用编排方面，MLFlow[10] 面向机器学习生命周期管理，提供模型注册、推理和测试等能力。百度开源的 PaddleFlow 在 k8s 基础上构建了一个云原生的 AI 工作流引擎，支持通过有向无环图（DAG）方式自定义任务流程，实现数据预处理、模型训练和模型推理。

目前，围绕 AI 应用研发和运营过程中的效率提升需求，百度、阿里、腾讯等公有云厂商陆续推出了 AI 平台产品，可以满足算法工程师对于数据管理、模型训练和服务部署的需求。然而，针对跨模态 AI 应用研发尚未有较好的解决方案，同时在各个环节上主要还是依赖人工经验，智能化程度还有待提升。

3 平台整体设计

3.1 平台总体架构

本文针对 AI 应用研发和运营全流程，构建了大模型驱动的 AI 应用服务平台。平台整体架构分为数据层、算法层、模型层和应用层（见图 1）。该平台汇聚开源及生态资源，辅助用户打造数据产品、算法产品、模型产品和应用产品，引入大模型技术进一步提升研发运营效率，对内提升企业内部数智化水平，对外协助客户打造面向泛行业需求的数智化产品。

图 1 平台总体架构

Fig. 1 Overall architecture of AI application service platform

数据层主要用于管理模型训练所需要的数据和标签，同时面向大模型应用提供知识库管理能力。在权限设置方面，构建了公共数据资产、组织数据资产和用户数据资产三级管理体系，通过指定分享对象和指定可见范围的方式，在实现数据资产便捷共享的同时，也保证了数据隐私和安全。通过引入预训练大模型和开放目标检测模型辅助完成数据标注工作，改进了传统人工标注的方式，大幅提升了标注效率。

算法层实现了对开源算法和生态算法的统一汇聚，对常用的训练框架、微调框架和推理框架进行了适配和集成；此外，提供了在线代码编辑器辅助用户完成算法研发工作；支持传统的深度学习场景，如图像分类、目标检测、文字识别和语音识别等任务，同时也兼容大语言模型场景中的中文分词和文本向量化等算法。

模型层为用户提供开箱即用的模型训练服务，基于用户数据和基础算法快速完成模型训练过程，构建个性化 AI 模型。针对模型部署需求，提供了量化加速工具提升推理速度和效率，同时提供了常用 AI 加速芯片的模型适配能力，协助用户便捷完成模型部署工作。

应用层支撑用户完成对多个 AI 模型的灵活组合编排以及参数配置，结合本地知识库和其他原子化 IT 能力快速构建场景化 AI 应用，同时对应用的状态进行实时监控，并基于监控指标数据进行任务调度和资源弹性扩缩容，保证业务连续性和算力资源的高效利用。

3.2 主要业务流程

本文所构建的 AI 应用服务平台主要服务于数据研发工程师、算法研发工程师、模型研发工程师和应用研发工程师，加强各流程间协作，提升环节研发效率。主要业务流程如图 2 所示。

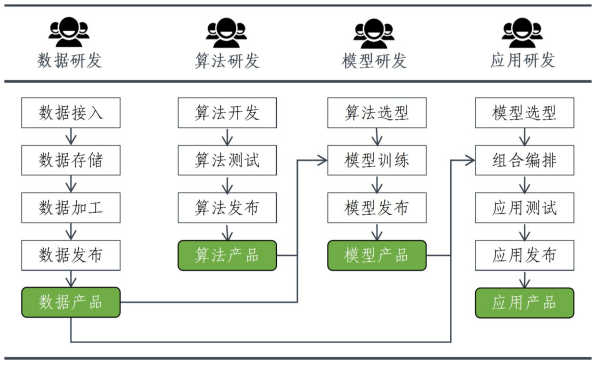

图 2　业务流程图

Fig. 2　Business flow chart

面向数据研发工程师，平台主要提供数据接入、数据加工和数据发布等能力。数据接入主要通过文件上传和数据库连接两种方式实现数据源的接入，通过内置的常用数据处理算法，可以完成结构化和非结构化（文本和图像等）数据的清洗工作。针对模型训练所需的标注数据，通过多模态大模型完成自动化预标注，结合人工校正大幅提升了数据标注效率和准确率，完成数据处理和标注，可以发布成数据产品供平台其他用户使用。

面向算法研发工程师，平台主要提供算法开发、算法测试和算法发布等能力。用户可以通过平台提供的在线编辑器完成算法代码的编写，基于大模型代码生成能力自动分析接口文档并生成和运行测试代码，最终生成测试报告。

面向模型研发工程师，平台主要提供算法选型、模型训练和模型发布等能力。用户通过自动化测试功能，可以在常用数据集上对比不同算法的准确率、召回率和运行速度等指标，快速完成算法选型工作。基于业务需求，选择合适的训练数据集、评估指标和训练参数等，完成模型训练工作，并通过模型发布模块将模型文件和推理代码打包发布成模型服务。

面向应用研发工程师，平台主要提供模型选型、组合编排、应用测试和应用发布等能力。基于业务需求，通过可视化的方式对模型进行组合和编排，结合各类判断条件和其他原子化 API 能力，形成可执行的有向无环图，测试通过后发布成最终的应用产品。

4　实验与结果分析

4.1　关键技术实现方案

平台主要采用了大模型技术对传统 AI 应用研发流程进行了优化，将其应用到 AI 应用研发过程中的预备态、设计态和运行态，提升了 AI 应用研发和运营效率。

4.1.1　预备态技术架构

预备态是 AI 应用研发过程所需要进行的前置工作流程，包括数据采集、数据预处理和数据标注等。本文主要面向预备态中的数据标注进行了优化，采用自然语言驱动方式进行标注任务设定，调用大语言模型和开放词汇目标检测模型完成数据自动标注。

如图 3 所示，传统智能标注方法通常需要先采集少量样本数据并进行人工标注，然后通过少量样本完成标注模型训练，通过训练得到的模型进行智能标注，工作量较大而且标注质量无法保证。随着多模态大模型技术的成熟，越来越多的研究开始采用大模型辅助完成标注。用户通过自然语言方式描述需要标注的目标类型，将图片和描述内容作为多模态大模型的输入，基于多模态对齐能力完成数据标注并获取标注信息。然而，多模态大模型需要消耗较多资源，标注效率较低且成本较高，难以大规模投入生产。

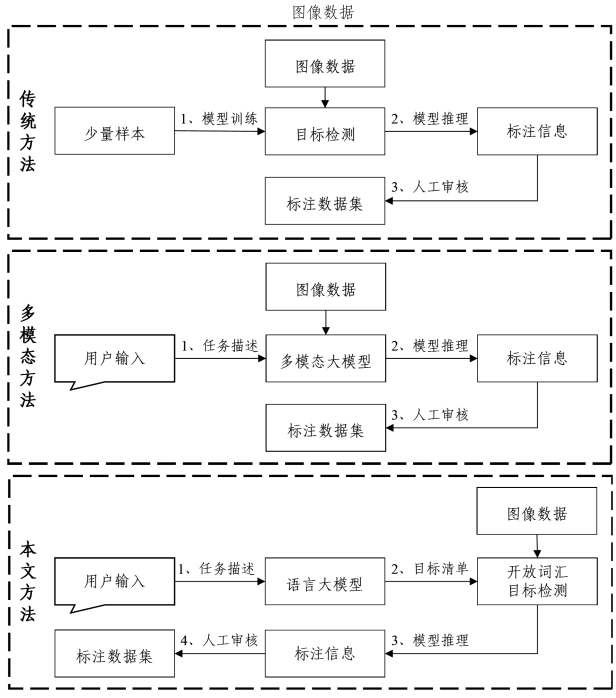

图 3　预备态技术架构

Fig. 3　Technology architecture of prepare state

针对上述两种方法存在的问题，本文基于大语言模型和开放词汇目标检测模型实现了目标检测任务的自动标注。通过自然语言方式对标注任务进行描述，利用大语言模型[11]提取和分析需要标注的目标信息，生成目标清单。将目标清单和图像数据传入开放词汇目标检测模型[12]，获取标注信息。与传统方法和多模态方法相比，本文所提出的基于大语言模型和开放词汇目标检测模型的方法可以有效提升标注准确率，同时节省算力资源。

4.1.2　设计态技术架构

设计态是 AI 应用研发的主要过程，本文所设计的 AI 应用服务平台集成了常用的人工智能算法，用户可以通过零代码、低代码和全代码 3 种方式进行模型开发。本文主要面向设计态中的模型测试，基于代码大模型和程序辅助语言模型[13]（Program-aided Language Models，PAL Models）实现模型接口的自动化测试。

目前主要通过开源大模型和提示词设计，完成单元测试用例生成。针对生成测试单元测试代码的可执行性、覆盖率和可维护性进行研究[14]。以 Chao 等提出的 CasModaTest[15]为例，通过构建测试用例池辅助大模型生成测试用例和测试预期结果，通过比对测试用例执行后的输出得到最终测试报告。此类方法主要适用于软件开发过程，特别是系统功能测试等方面，难以满足 AI 应用测试对准确率、召回率和响应时间等的要求。

本文针对上述问题，基于开源代码大模型[16]和 PAL 方式构建了面向 AI 应用的测试大模型。基于用户输入的测试目的，自动分析接口文档内容并构建测试用例，基于测试目的完成测试数据的筛选，最后调用脚本解析器完成测试用例执行和测试报告生成工作（见图 4）。

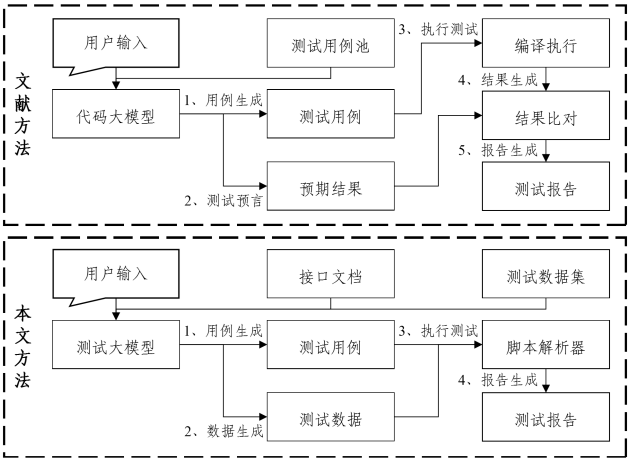

图 4 设计态技术架构

Fig. 4 Technology architecture of design state

4.1.3 运行态技术架构

运行态主要包括了 AI 应用上线后所需要的运营和运维工作，为 AI 应用使用方提供便捷的个性化服务。本文利用大模型对 AI 应用运行日志进行分析，通过 API 接口方式结合传统小模型能力完成智能化运维和运营。

传统基于任务驱动的运维方案[17-18]通常将运维过程分为异常检测、问题定位和服务恢复等原子化任务进行处理，针对不同类型的问题开发对应的运维学件，灵活性较差。本文基于大模型[19]的自然语言理解能力，对采集到的 AI 应用日志数据进行理解和分析，可以快速适应多类下游任务（见图 5）。

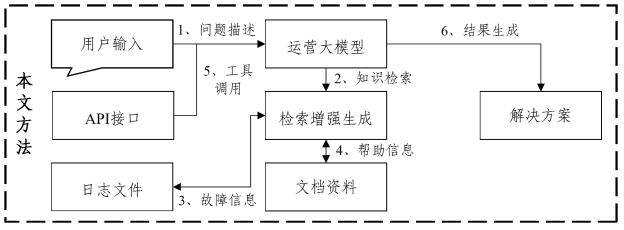

图 5 运行态技术架构

Fig. 5 Technology architecture of operating state

结束语 本文基于预训练大模型技术对 AI 应用服务平台进行了改进，针对 AI 应用研发和运营全过程管理中的问题，对预备态、设计态和运行态中的核心流程进行了改进；分别构建了标注大模型、测试大模型和运维大模型，提升了 AI 应用的研发效率，同时实现了对 AI 应用系统日志的实时分析，提升了服务可用性和用户使用体验。

参 考 文 献

[1] BERNSTEIN D. Containers and Cloud: From LXC to Docker to Kubernetes[J]. Cloud Computing, IEEE, 2014, 1(3): 81-84.

[2] NVIDIA. NVIDIA Virtual GPU Software Documentation v18.0 [EB/OL]. https://docs.nvidia.com/vgpu/18.0/index.html.

[3] GU J, SONG S, LI Y, et al. GaiaGPU: Sharing GPUs in Container Clouds[C]// 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications. 2018, 469-476.

[4] TAN Z, LI D W, WANG S, et al. Large Language Models for Data Annotation: A Survey [J]. arXiv: 2402.13446, 2024.

[5] JING Z, SU Y Y, HAN Y K, et al. When Large Language Models Meet Vector Databases: A Survey[J]. arXiv: 2402.01763, 2024.

[6] PASZKE A, GROSS S, MASSAF, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library [J]. arXiv: 2209.15428, 2019.

[7] ABADI M, AGARWAL A, BARHAM P, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems [J]. arXiv: 1605.08695, 2016.

[8] ZHOU Y, YANG K. Exploring TensorRT to Improve Real-Time Inference for Deep Learning[C]// 2022, IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys). 2022.

[9] KWON W, LI Z H, ZHUANG S Y, et al. Efficient Memory Management for Large Language Model Serving with PagedAttention[J]. arXiv: 2309.06180, 2023.

[10] BEN W. MLflow: A Tool for Managing the Machine Learning Lifecycle[EB/OL]. https://mlflow.org/docs/latest/index.html.

[11] YANG A, YANG B S, HUI B Y, et al. Qwen2 Technical Report [J]. arXiv: 2407.10671, 2024.

[12] CHENG T H, SONG L, GE Y X, et al. YOLO-World: Real-Time Open-Vocabulary Object Detection [J]. arXiv: 2401.17270v2, 2024.

[13] GAO L, MADAAN A, ZHOU S, et al. PAL: Program-aided Language Models[J]. arXiv: 2211.10435, 2022.

[14] YANG L, YANG C, GAO S T, et al. An Empirical Study of Unit Test Generation with Large Language Models[J]. arXiv: 2406.18181v1, 2024.

[15] NI C, WANG X Y, CHEN L S, et al. CasModaTest: A Cascaded and Model-agnostic Self-directed Framework for Unit Test Generation[J]. arXiv: 2406.15743, 2024.

[16] XIA Y H, CHEN Y Y, SHI T Y, et al. AICoderEval: Improving AI Domain Code Generation of Large Language Models[J]. arXiv: 2406.04712, 2024.

[17] MENG W, LIU Y, ZHU Y, et al. LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs[C]// Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19). 2019.

[18] TAO S, MENG W, CHENG Y, et al. LogStamp: Automatic Online Log Parsing Based on Sequence Labelling[J]. Performance Evaluation Review, 2022(4): 49.

[19] ZHOU X H, LI G L, LIU Z Y. LLM As DBA [J]. arXiv: 2308.05481, 2024.

LIANG Binghao, born in 1991, Ph. D. His main research interests include artificial intelligence and computing power network application.