Red Hat
Enterprise Linux AI

Instruct Lab & Granite Model

Solution Architect

Benny Liu（刘洋）

Red Hat

在Red Hat的产品中引入AI

红帽在其业务中利用人工智能有两项主要政策。一是利用AI增强现有核心产品（RHEL、Ansible、OpenShift）的功能，二是为AI本身的开发和运营提供平台。

1

利用AI来强化核心产品

去年的发布Ansible LightSpeed机制已扩展到RHEL和OpenShift，并宣布为Red Hat LightSpeed

Red Hat LightSpeed

- OpenShift LightSpeed
- Red Hat Enterprise Linux LightSpeed
- Ansible LightSpeed

生成式人工智能降低了产品使用的技术障碍，有助于提高开发人员和运维人员的生产力。

2

提供开发/运营AI应用的平台

发布新产品 RHEL AI 和 Podman AI Lab。我们还宣布为现有 OpenShift AI 添加新功能并扩大合作伙伴关系。

Red Hat Enterprise Linux AI

提供本地环境下的AI开发能力

- OSS Granite 大模型
- InstructLab CLI
- RHEL AI镜像

Red Hat OpenShift AI

提供集成了MlOps的AI开发运营平台

- 模型的开发部署
- 分布式训练
- vLLM支持
- Nvidia/Intel/AMD全面支持合作发布

Red Hat

生成式AI快速发展面临的挑战

基于“开放”模型(包括Mistral和Llama模型)的生态系统已经以惊人的规模迅速发展起来，其中包括PyTorch、HuggingFace、LangChain等开源工具。 开发人员正在分享关于优化、调优和服务开放模型的最佳实践和技巧。

在HuggingFace上，GitHub上有近17k个llama模型变体和11,200个相关的工具/项目库。有超过6k的Mixtral /Mistral模型变体和1100个相关的工具/项目。

◆ 面临的挑战

对模型本身的直接贡献是不可能的。它们以分叉的形式出现，这迫使消费者选择一个“最适合”的模型，然后进行成本高昂的资源密集型训练。这种模型不容易扩展，而且对于模型创建者来说，分叉的维护成本很高 。

无法将改进纳入上游项目，利用社区贡献不断改进模型；同时“开放模型”的协议不是基于MIT/Apache 2.0等，有自己的限制。

由于缺乏AI/ML专业知识，贡献想法的能力受到限制。一个人必须学习如何分叉、训练和完善模型，以便看到他们的想法向前发展。这是一个非常高的进入门槛。

对于分叉模型的审查、管理和分发，没有直接的社区治理或最佳实践。“AI Alliance”（AI联盟）正在致力于寻求在治理、流程和实践方面定义开源 AI 在行业规模上的含义。

Red Hat

“开放大模型”的当前局限性

开源许可证的重要性

许多大模型的许可证对使用场景、用户类型、商业用途等进行了严格的限制，远远超过了传统开源软件的规定.

开源内容的重要性

所谓的开源大模型更像是免费软件（freeware）而非真正的开源软件（open-source software）。它们提供的只是使用现有模型的便利，而不是完全的技术透明和开发自由.

技术与协助开发的挑战

原始训练数据和具体训练步骤的缺失，使得微调效果大打折扣。此外，由于微调只能在现有模型基础上进行，开发者无法对模型进行深层次的改进。

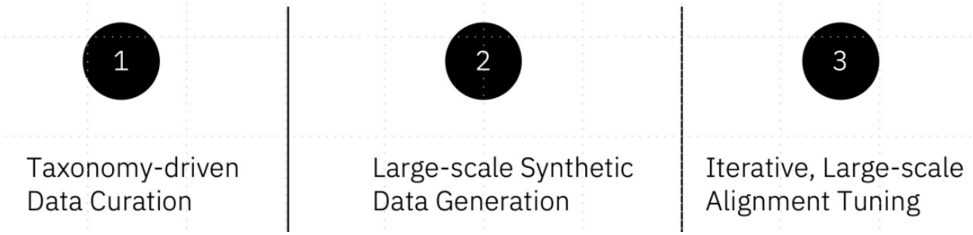
这种封闭的开发模式削弱了开源项目的社区力量，使得模型的改进速度和质量受到影响

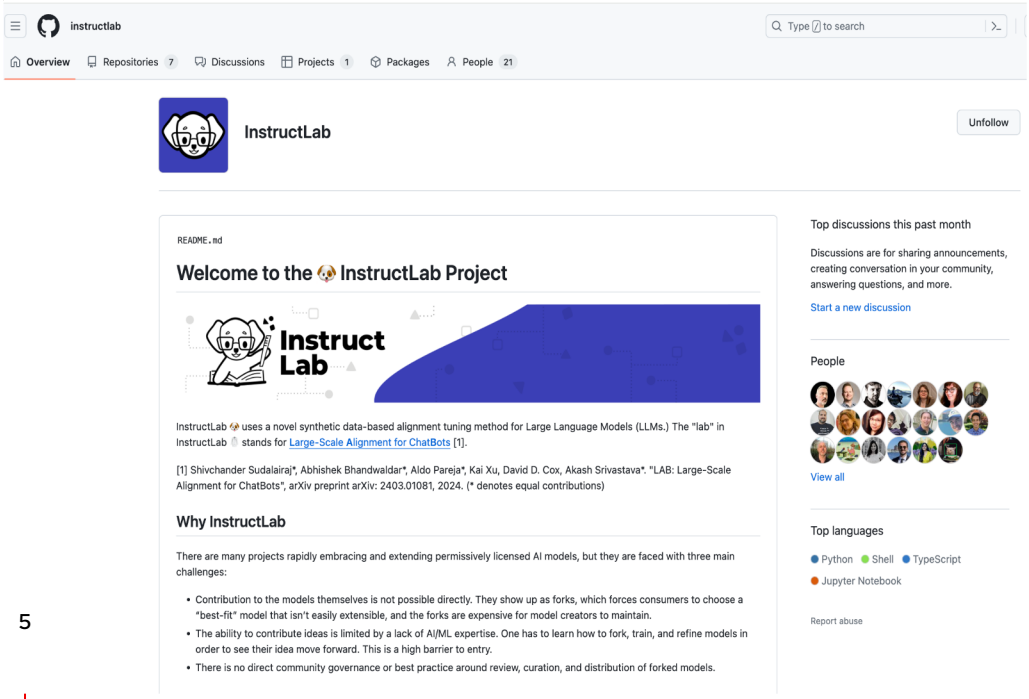
厂商	模型	发布时间	开放的内容				备注
			代码	数据集	训练过程	权重	
Meta	Llama	2023 年 2 月	X	X	X	✓	
	Llama 2	2023 年 7 月	X	X	X	✓	
	Llama 3	2024 年 4 月	X	X	X	✓	
智谱 AI	ChatGLM-6B	2023 年 3 月	X	X	X	✓	
Databricks	Dolly 2.0	2023 年 4 月	✓	✓	✓	✓	
百川智能	Baichuan-13B	2023 年 7 月	X	X	X	✓	
	Baichuan2	2023 年 9 月	X	X	X	✓	
阿里云	Qwen-7B	2023 年 8 月	X	X	X	✓	
	Qwen1.5-110B	2024 年 4 月	X	X	X	✓	
零一万物	Yi	2023 年 11 月	X	X	X	✓	
Google	Gemma	2024 年 2 月	X	X	X	✓	
Allen AI	OLMo	2024 年 2 月	✓	✓	✓	✓	
xAI	Grok-1	2024 年 3 月	X	X	X	✓	
Apple	OpenELM	2024 年 4 月	✓	✓	✓	✓	
腾讯	混元-DiT	2024 年 5 月	X	X	X	✓	
NVIDIA	Llama3-ChatQA-1.5	2024 年 5 月	X	✓	X	✓	基于 Llama 3

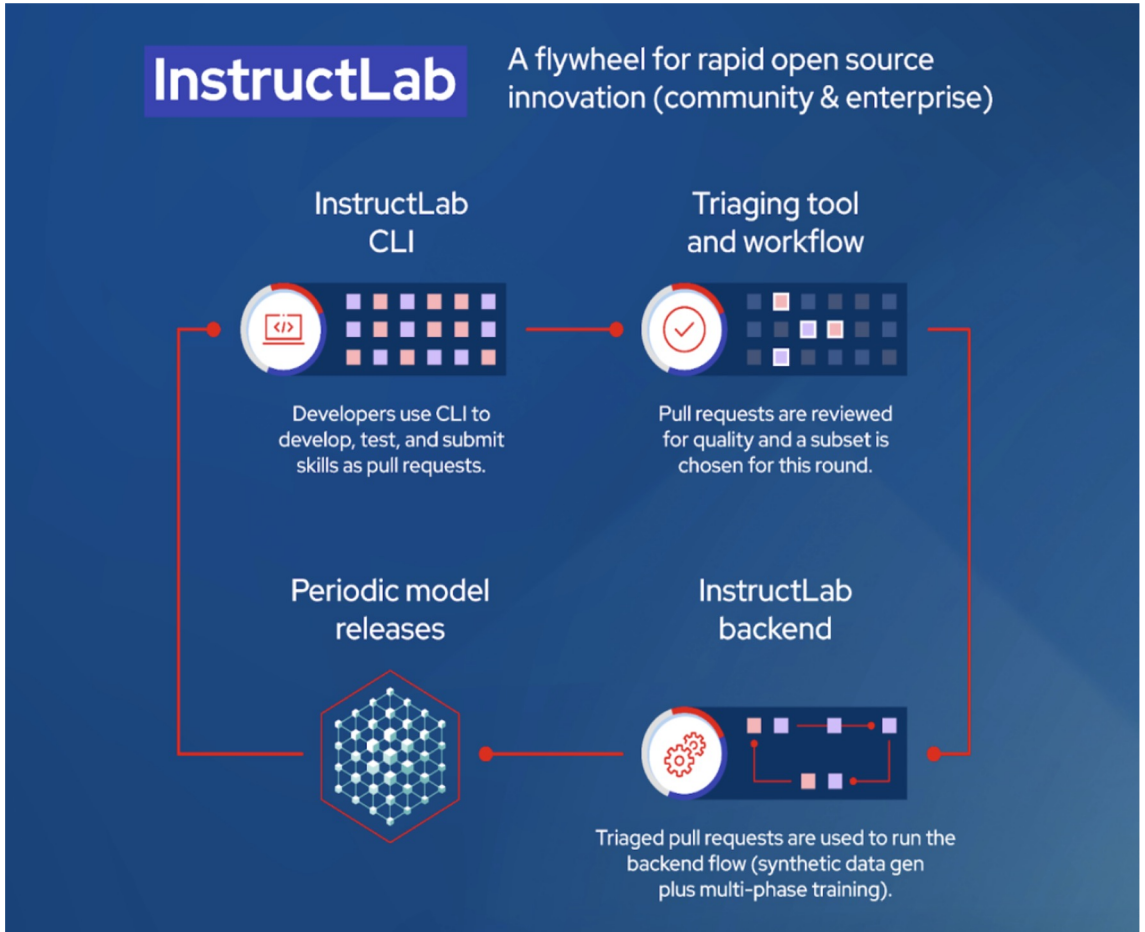
InstructLab Project: 生成式AI模型开发的开源社区

（ **L**arge-scale **A**lignment for chat**B**ots methodology ）

- ✓ InstructLab允许为LLM建立类似拉请求的上游贡献接受工作流。这些上游贡献为模型添加了额外的“技能”或“知识”，而不需要完全分叉模型并进行微调。我们相信，一个结合了强大的基础模型以及支持快速协作模型开发的独特工具包和数据集的开源项目将会引人注目，并吸引社区的注意。
- ✓ 目标是使任何人都可以通过开源的 Language Model Development Kit (LMDK) 将他们的知识贡献给AI模型







InstructLab, Granite Models & RHEL AI

- 红帽推出 InstructLab 项目，以简化生成式AI模型对齐调整，并支持基于社区的模型开发方法
- IBM Research 与 Red Hat 合作开源 Granite 大语言模型和 InstructLab 模型对齐工具
- 红帽推出RHEL AI基础模型平台，帮助用户无缝开发、测试和部署生成式AI模型

InstructLab 是一个关键平台，它使用户能够将 LLM 应用于从聊天机器人到编码助手等一系列应用。这些模型（包括 OpenAI 的 GPT、Anthropic 的 Claude、Meta 的 Llama、Mistral AI 的 Mistral 和 IBM 的 Granite）通常需要进行大量调整才能满足特定的业务需求。

InstructLab 的方法称为 LAB（聊天机器人的大规模对齐），由三个创新部分组成：

1. *分类驱动的数据管理*：这涉及创建一组由人类管理的多样化训练数据，以便为模型引入新的知识和技能。
2. *大规模合成数据生成*：该模型从精选数据中生成新示例。这些合成示例经过自动细化过程，以确保其扎实且安全。
3. *迭代、大规模对齐调整*：使用精炼的合成数据对模型进行重新训练，首先关注知识调整，然后关注技能调整。

Red Hat

LAB (Large-scale Alignment for ChatBots) method

基于分类（Taxonomy-based）的技能和知识表示

以分层分类法表示任何缺失的模型知识或技能，为每个缺失的技能提供 5 个以上缺失行为的示例数据点。

使用“教师模型”生成综合合成数据（merlinite model）

教师模型基于分类法则生成包含数百万个问题和答案的“课程”.

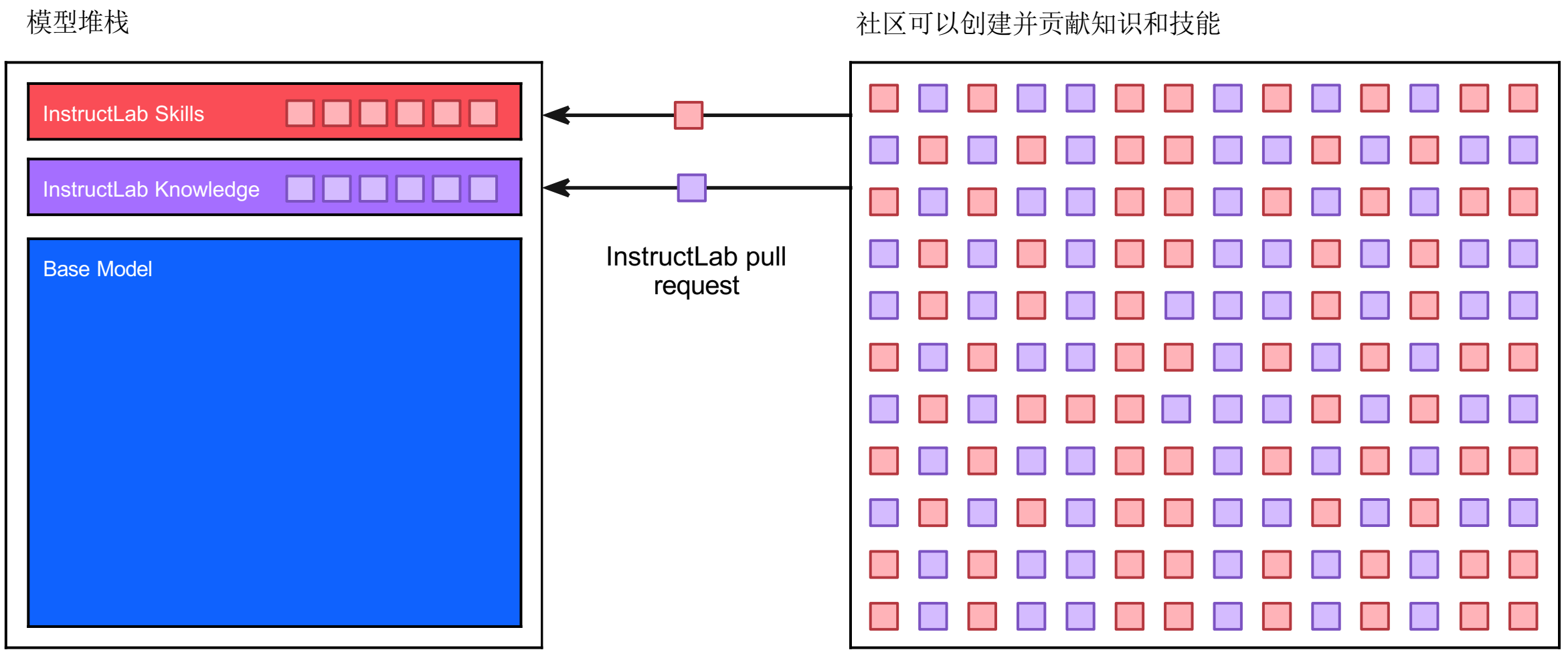
用批判模型进行综合数据验证

批判模型会过滤问题的正确性和质量，并扫描合成数据以查找违禁材料。

对学生模型进行基础技能和知识训练

使用新颖的训练方法对学生模型进行课程训练.

InstructLab: 利用LAB方法实现社区驱动的模型开发和演化



Red Hat

Granite Code Serial Model：仅解码器代码模型

Granite Code 系列代码模型，旨在支持企业软件开发广泛的编码任务。Granite Code 模型有两个主要变体，该研究以四种不同的规模发布(3B、8B、20B 和 34B)， Granite Code 模型都是在 Apache 2.0 许可下发布的:

- Granite Code Base：用于代码相关任务的基础模型，使用116种编程语言编写的代码进行了培训。
- Granite Code Instruct：使用 Git 提交与人类指令的组合以及开源合成生成的代码指令数据集进行微调的指令遵循模型。

数据收集和准备

数据爬取和过滤：预训练代码数据来自公开可用的数据集,如 Github Code Clean 2、StarCoderdata 3以及 GitHub的其他公共代码存储库和问题的组合，根据编程语言过滤收集的代码数据，包括质量较低的代码。

精确和模糊数据删除： 采用一种积极的重复数据删除策略，包括精确和模糊重复数据删除，删除具有相近代码的文档。

HAP、PII、恶意软件过滤： 应用HAP内容过滤器，减少模型产生仇恨、辱骂或亵渎语言的可能性；保护隐私，替换PII文本；使用ClamAV扫描所有数据集，以识别和删除源代码中的恶意软件实例。

自然语言收集： 策划了几个公开可用的高质量自然语言数据集,以提高模型在语言理解和数学推理方面的能力。

模型预训练

Granite Code 基础模型在 3.5T 到 4.5T 个与代码相关的自然语言数据集中的代码数据令牌上训练。使用 IBM 的两个超级计算集群 Vela 和 Blue Vela 训练 Granite Code 模型,集群配备了数千个 NVIDIA A100 和 H100 GPU，通过通过 NVLink 和 NVSwitch 相互连接。

阶段一： 3B 和 8B 模型都在 4 万亿个代码数据令牌上训练,包括 116 种语言。20B 参数模型在 3 万亿个代码令牌上训练。在 20B 模型的 1.6万亿检查点完成深度放大后，34B 模型在 1.4万亿个令牌上训练。

阶段二： 包括来自各个领域的额外的高质量公开可用数据,包括技术、数学和网络文档,以进一步提高模型在推理和解决问题方面的性能,这对代码生成至关重要。该研究在第 2 阶段训练中为所有模型训练 5000 亿个令牌(80% 的代码和 20% 的语言数据)。

Instruct—微调

Granite Code Instruct模型在以下类型的指令数据上进行了微调:

- 代码提交来自CommitPackFT
- 高质量的数学数据集，特别是我们使用了MathInstruct和MetaMathQA
- 代码指令数据集，如Glaive-Code-Assistant-v3, self - ss - instruction - sc2, Glaive-Function-Calling-v2, NL2SQL11和一小部分合成API调用数据集
- 高质量的语言指令数据集，如HelpSteer和开放许可过滤版本的Platypus。
- 使用Dolomite Engine finetune(或者是Instruct tuning)所有的Granite模型

Red Hat

Granite Code Serial Model：仅解码器代码模型

Granite Code 系列代码模型，旨在支持企业软件开发广泛的编码任务。Granite Code 模型有两个主要变体，该研究以四种不同的规模发布(3B、8B、20B 和 34B)， Granite Code 模型都是在 Apache 2.0 许可下发布的:

- Granite Code Base：用于代码相关任务的基础模型，使用116种编程语言编写的代码进行了培训。
- Granite Code Instruct：使用 Git 提交与人类指令的组合以及开源合成生成的代码指令数据集进行微调的指令遵循模型。

数据收集和准备

数据爬取和过滤：预训练代码数据来自公开可用的数据集,如 Github Code Clean 2、StarCoderdata 3以及 GitHub的其他公共代码存储库和问题的组合，根据编程语言过滤收集的代码数据，包括质量较低的代码。

精确和模糊数据删除： 采用一种积极的重复数据删除策略，包括精确和模糊重复数据删除，删除具有相近代码的文档。

HAP、PII、恶意软件过滤： 应用HAP内容过滤器，减少模型产生仇恨、辱骂或亵渎语言的可能性；保护隐私，替换PII文本；使用ClamAV扫描所有数据集，以识别和删除源代码中的恶意软件实例。

自然语言收集： 策划了几个公开可用的高质量自然语言数据集,以提高模型在语言理解和数学推理方面的能力。

模型预训练

Granite Code 基础模型在 3.5T 到 4.5T 个与代码相关的自然语言数据集中的代码数据令牌上训练。使用 IBM 的两个超级计算集群 Vela 和 Blue Vela 训练 Granite Code 模型,集群配备了数千个 NVIDIA A100 和 H100 GPU，通过通过 NVLink 和 NVSwitch 相互连接。

阶段一： 3B 和 8B 模型都在 4 万亿个代码数据令牌上训练,包括 116 种语言。20B 参数模型在 3 万亿个代码令牌上训练。在 20B 模型的 1.6万亿检查点完成深度放大后，34B 模型在 1.4万亿个令牌上训练。

阶段二： 包括来自各个领域的额外的高质量公开可用数据,包括技术、数学和网络文档,以进一步提高模型在推理和解决问题方面的性能,这对代码生成至关重要。该研究在第 2 阶段训练中为所有模型训练 5000 亿个令牌(80% 的代码和 20% 的语言数据)。

Instruct—微调

Granite Code Instruct模型在以下类型的指令数据上进行了微调:

- 代码提交来自CommitPackFT
- 高质量的数学数据集，特别是我们使用了MathInstruct和MetaMathQA
- 代码指令数据集，如Glaive-Code-Assistant-v3, self - ss - instruction - sc2, Glaive-Function-Calling-v2, NL2SQL11和一小部分合成API调用数据集
- 高质量的语言指令数据集，如HelpSteer和开放许可过滤版本的Platypus。
- 使用Dolomite Engine finetune(或者是Instruct tuning)所有的Granite模型

Red Hat

Granite Code Serial Model：仅解码器代码模型

Granite Code 系列代码模型，旨在支持企业软件开发广泛的编码任务。Granite Code 模型有两个主要变体，该研究以四种不同的规模发布(3B、8B、20B 和 34B)， Granite Code 模型都是在 Apache 2.0 许可下发布的:

- Granite Code Base：用于代码相关任务的基础模型，使用116种编程语言编写的代码进行了培训。
- Granite Code Instruct：使用 Git 提交与人类指令的组合以及开源合成生成的代码指令数据集进行微调的指令遵循模型。

数据收集和准备

数据爬取和过滤：预训练代码数据来自公开可用的数据集,如 Github Code Clean 2、StarCoderdata 3以及 GitHub的其他公共代码存储库和问题的组合，根据编程语言过滤收集的代码数据，包括质量较低的代码。

精确和模糊数据删除： 采用一种积极的重复数据删除策略，包括精确和模糊重复数据删除，删除具有相近代码的文档。

HAP、PII、恶意软件过滤： 应用HAP内容过滤器，减少模型产生仇恨、辱骂或亵渎语言的可能性；保护隐私，替换PII文本；使用ClamAV扫描所有数据集，以识别和删除源代码中的恶意软件实例。

自然语言收集： 策划了几个公开可用的高质量自然语言数据集,以提高模型在语言理解和数学推理方面的能力。

模型预训练

Granite Code 基础模型在 3.5T 到 4.5T 个与代码相关的自然语言数据集中的代码数据令牌上训练。使用 IBM 的两个超级计算集群 Vela 和 Blue Vela 训练 Granite Code 模型,集群配备了数千个 NVIDIA A100 和 H100 GPU，通过通过 NVLink 和 NVSwitch 相互连接。

阶段一： 3B 和 8B 模型都在 4 万亿个代码数据令牌上训练,包括 116 种语言。20B 参数模型在 3 万亿个代码令牌上训练。在 20B 模型的 1.6万亿检查点完成深度放大后，34B 模型在 1.4万亿个令牌上训练。

阶段二： 包括来自各个领域的额外的高质量公开可用数据,包括技术、数学和网络文档,以进一步提高模型在推理和解决问题方面的性能,这对代码生成至关重要。该研究在第 2 阶段训练中为所有模型训练 5000 亿个令牌(80% 的代码和 20% 的语言数据)。

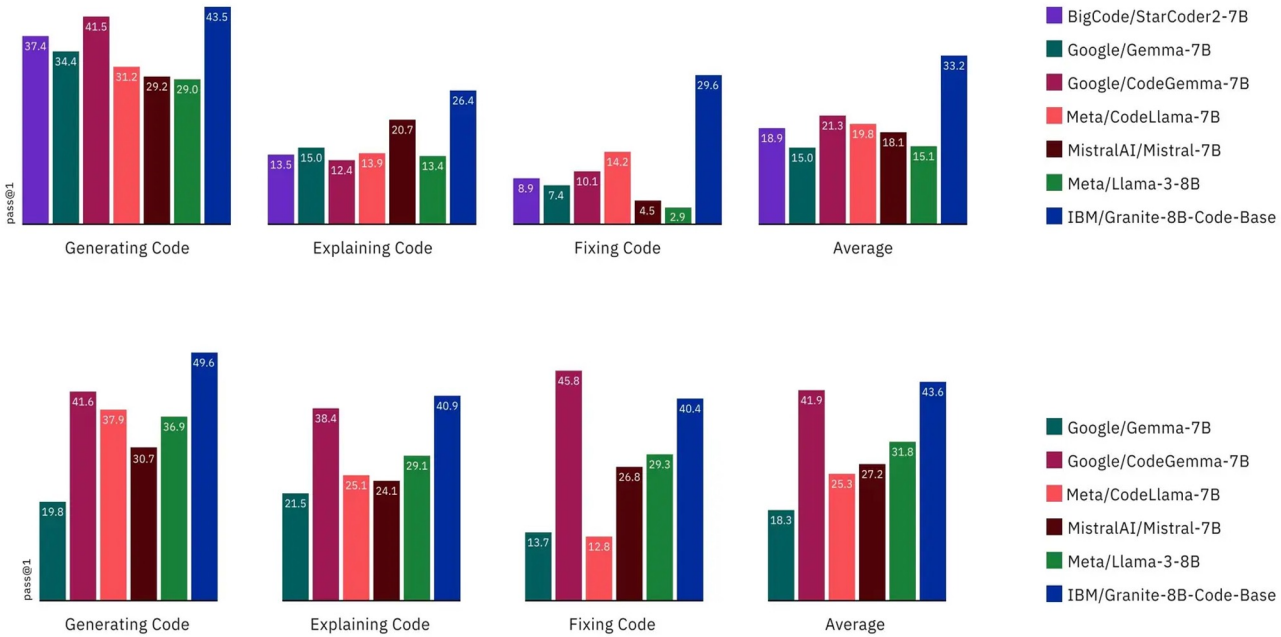
Instruct–微调

Granite Code Instruct模型在以下类型的指令数据上进行了微调:

- 代码提交来自CommitPackFT
- 高质量的数学数据集，特别是我们使用了MathInstruct和MetaMathQA
- 代码指令数据集，如Glaive-Code-Assistant-v3, self - ss - instruction - sc2, Glaive-Function-Calling-v2, NL2SQL11和一小部分合成API调用数据集
- 高质量的语言指令数据集，如HelpSteer和开放许可过滤版本的Platypus。
- 使用Dolomite Engine finetune(或者是Instruct tuning)所有的Granite模型

Red Hat

Granite Code模型：Apache 2.0开放许可，提升软件开发生产力



Model	MATH	GSM8K	SAT	OCW	MATH+Py	GSM8K+Py
StarCoderBase-7B	2.4	3.8	18.7	2.2	18.2	15.6
CodeLlama-7B	4.1	11.9	12.5	2.9	20.8	26.8
StarCoder2-7B	10.4	27.2	37.5	4.8	28.7	39.4
CodeGemma-7B	21.8	49.0	53.1	6.9	31.1	60.9
Granite-8B-Code-Base	21.4	**61.9**	62.5	8.8	**35.4**	**63.1**
Gemma-7B	**24.1**	53.3	**75.0**	7.3	27.4	52.9
Mistral-7B-v0.2	12.8	37.2	53.1	5.8	25.7	45.6
Llama-3-8B	15.6	49.8	34.4	**9.9**	0.0*	2.4
Lemma-7B	17.3	33.7	59.4	7.0	25.6	40.8

Red Hat

Merlinite & Labradorite Model：lab-enhanced Model

- IBM 宣布与法国 AI 模型公司 Mistral AI 建立新的战略合作伙伴关系。IBM 将很快在 IBM® watsonx.ai™ 上为其客户提供 Mistral 系列商业模型，既可在本地部署，也可在 IBM Cloud® 中使用。其中包括 Mistral-Large 的最新版本，它是当今市场上领先的 AI 模型之一。IBM 期待继续与 Mistral AI 进行开源合作，包括其广受欢迎的 Mixtral 系列开源混合专家模型，以及 IBM InstructLab 调优的 Mistral 7B 变体 Merlinite。

  2024 年 4 月 18 日，IBM 宣布在 Watsonx AI 和数据平台上推出 Meta Llama 3，这是 Meta 的下一代开放式 LLM，旨在帮助企业创新其 AI 之旅。Llama 3 的加入巩固了 IBM 与 Meta 的合作，以推动 AI 的开放式创新。这两家公司还于去年年底成立了AI 联盟，其中包括一群来自行业、初创企业、学术界、研究和政府的领先组织。自那时起，该联盟已发展到拥有 100 多名成员和合作者。

  这些战略合作伙伴关系是对 IBM开源战略和产品的补充，包括我们最强大、最高效的 IBM® Granite™ 代码模型和InstructLab，这是我们围绕 LLM 推进真正的开源创新的新方法。这种方法旨在通过开放模型和工具来振兴强大的 AI 生态系统，帮助企业在安全、负责任的 AI 方面开展合作。

  在 IBM，我们采用差异化方法来提供企业级模型，帮助客户自信而有控制地扩展高质量的新一代人工智能。IBM Research® Lab Alignment 技术现已集成到 InstructLab开源工具中，可以使用新的开源技能和知识来调整模型。

- InstructLab是 IBM 和 Red Hat 于 5 月推出的一个开源项目，旨在改变这一现状。它为社区提供了创建和合并 LLM 更改的工具，而无需从头开始重新训练模型。通过使 LLM 更像任何其他开源软件项目，IBM 和 Red Hat 希望使生成式 AI的访问更加民主化。
- InstructLab 的工作原理是利用 LLM 生成的高质量示例来增强人工整理的数据，从而降低数据创建成本。然后可以使用 InstructLab 生成的数据来定制或改进基础模型，而无需重新训练它，从而节省更多成本。IBM Research 已使用 InstructLab 生成合成数据，以改进其用于语言和代码的开源 Granite 模型。
- 这种技术的构建方式并不是特定于模型的，因此它可以像IBM granite家族一样轻松地应用于llama家族的模型。有可能将其他模型创建者(例如 Mistral, Meta等)作为合作伙伴。

Red Hat

Instruct Lab社区运营 & lab-enhanced Model 性能

- 使用 InstructLab 合成数据生成器的本地版本，您可以创建自己的指令来调整自己的模型，并进行实验，直到它们执行目标任务。配方完善后，您可以像任何其他开源项目一样，将其作为拉取请求提交给 GitHub 上的 InstructLab Taxonomy项目。

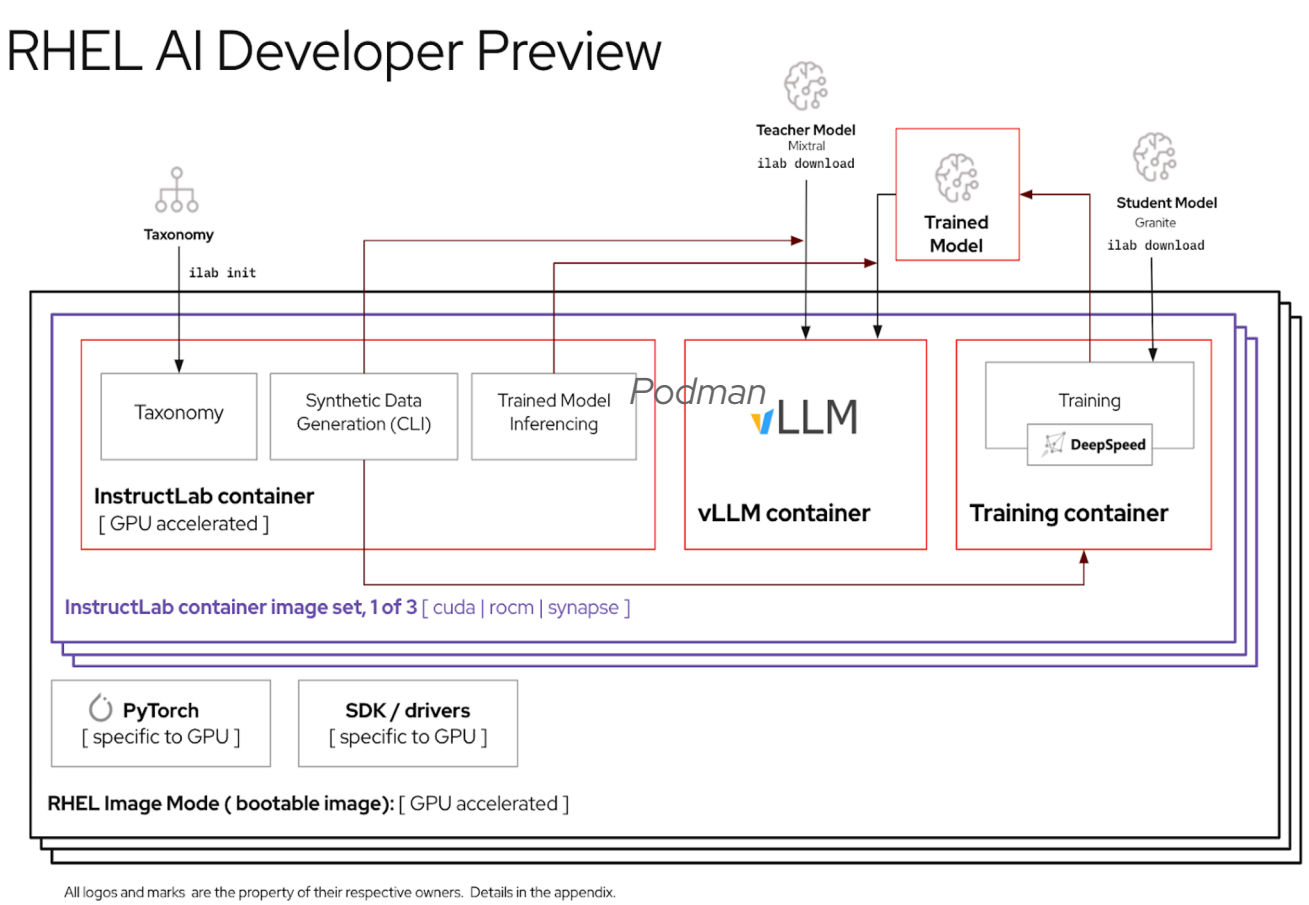
- 项目人员会审查所提议的技能，如果符合社区准则，则会生成数据并用于微调基础模型。然后，模型的更新版本会在 Hugging Face 上发布回社区。IBM 和 Red Hat 的目标是每周发布新版本。

- 随着 InstructLab 的启动，IBM 和 Red Hat 的维护人员将审查和批准社区提交的内容。获得维护者身份的贡献者可以批准提交的内容。所有提交的技能配方以及通过它们生成的数据都将发布到 InstructLab 项目中。

- 这一突破性创新实现了以前几乎不可能实现的事情——让社区能够为模型做出贡献并共同改进模型。

Model	Alignment	Base	Teacher	MTBench (Avg) *	MMLU(5-shot)	ARC-C(25-shot)	HellaSwag(10-shot)	Winogranc shot)
Llama-2-13b-chat-hf	RLHF	Llama-2-13b	Human Annotators	6.65	54.58	59.81	82.52	75.93
Orca-2-13b	Progressive Training	Llama-2-13b	GPT-4	6.15	60.37 *	59.73	79.86	78.22
WizardLM-13B-V1.2	Evol-Instruct	Llama-2-13b	GPT-4	7.20	54.83	60.24	82.62	76.40
Labradorite-13b	Large-scale Alignment for chatBots (LAB)	Llama-2-13b	Mixtral-8x7B-Instruct	7.23	58.89	61.69	83.15	79.56
Mistral-7B-Instruct-v0.1	SFT	Mistral-7B-v0.1	-	6.84	60.37	63.65	84.76	76.80
zephyr-7b-beta	SFT/DPO	Mistral-7B-v0.1	GPT-4	7.34	61.07	63.74	84.19	78.06
Mistral-7B-Instruct-v0.2	SFT	Mistral-7B-v0.1	-	7.6**	60.78	63.14	84.88	77.19
Merlinite-7b	Large-scale Alignment for chatBots (LAB)	Mistral-7B-v0.1	Mixtral-8x7B-Instruct	7.66	64.88	63.99	84.37	78.24

InstructLab, Granite Models & RHEL AI



All logos and marks are the property of their respective owners. Details in the appendix.

README Apache-2.0 license

RHEL AI Developer Preview Guide

This guide will help you assemble and test a developer preview version of the RHEL AI product.

Overview

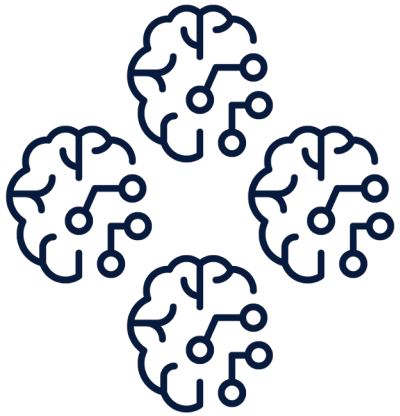
Welcome to the **Red Hat Enterprise Linux AI Developer Preview!** This guide is meant to introduce you to RHEL AI Developer Preview capabilities. As with other Developer Previews, expect changes to these workflows, additional automation and simplification, as well as a broadening of capabilities, hardware and software support versions, performance improvements (and other optimizations) prior to GA.

RHEL AI is an open-source product that includes:

- Granite: an open source, Apache 2 licensed foundation model from IBM.
- InstructLab: a CLI and tuning backend that provides a simple user interface for contributing knowledge and skills to a base model.
- RHEL Image Mode (bootc): RHEL AI is distributed as a "bootable container" image. Provision RHEL AI appliances via kickstart onto bare metal or cloud instances.
- vLLM: A high-throughput and memory-efficient inference and serving engine for LLMs, based on PyTorch.
- deepspeed: A deep learning optimization software suite for both training and inference.
- PyTorch: PyTorch is an optimized tensor library for deep learning using GPUs and CPUs.

Red Hat

RedHat AI Steps



InstructLab

Red Hat
Enterprise Linux AI

Red Hat
OpenShift AI

STEP 1

通过有限的桌面规模的训练方法(qLora)在小数据集上学习和实验。未来潜在的Podman Desktop集成。

Laptop / desktop

STEP 2

生产级模型培训，使用完整的合成数据生成以及教师和评论模型。工具侧重于可脚本化的原语

Server / VM

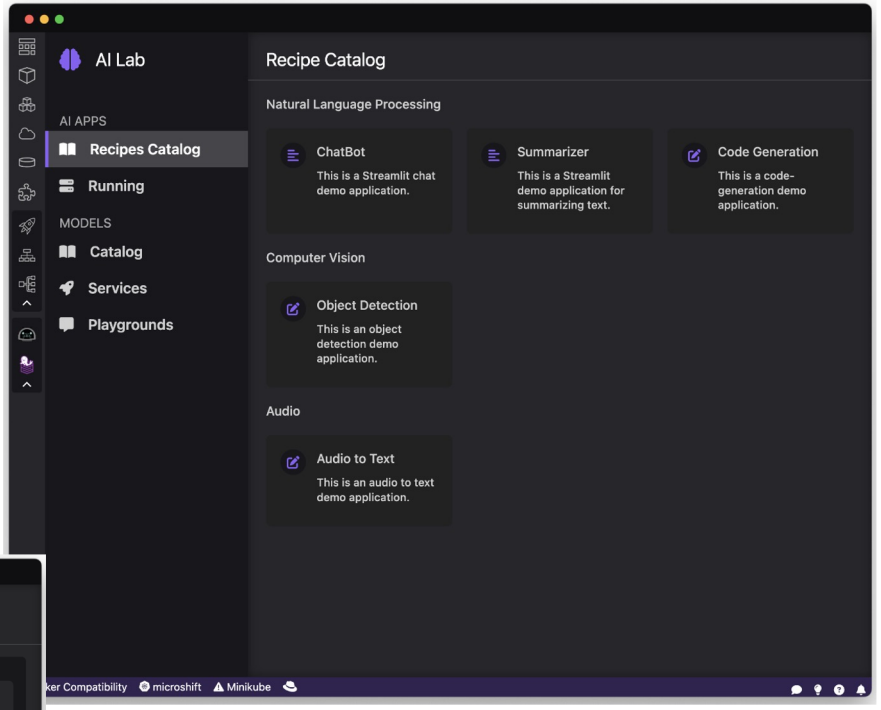
STEP 3

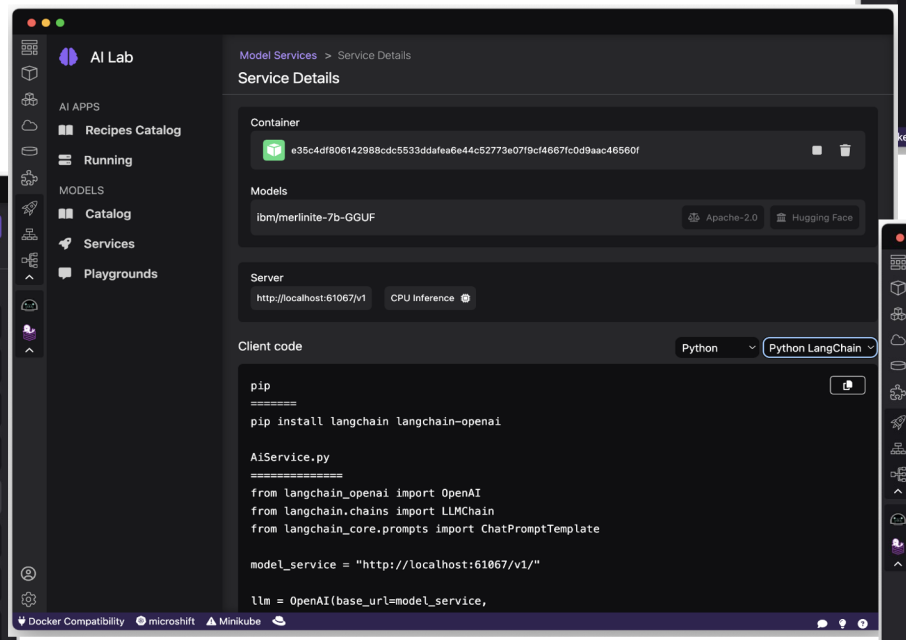
与 RHEL AI 一样的生产级模型训练，充分利用 Kubernetes 扩展、自动化和 MLOps 服务的功能

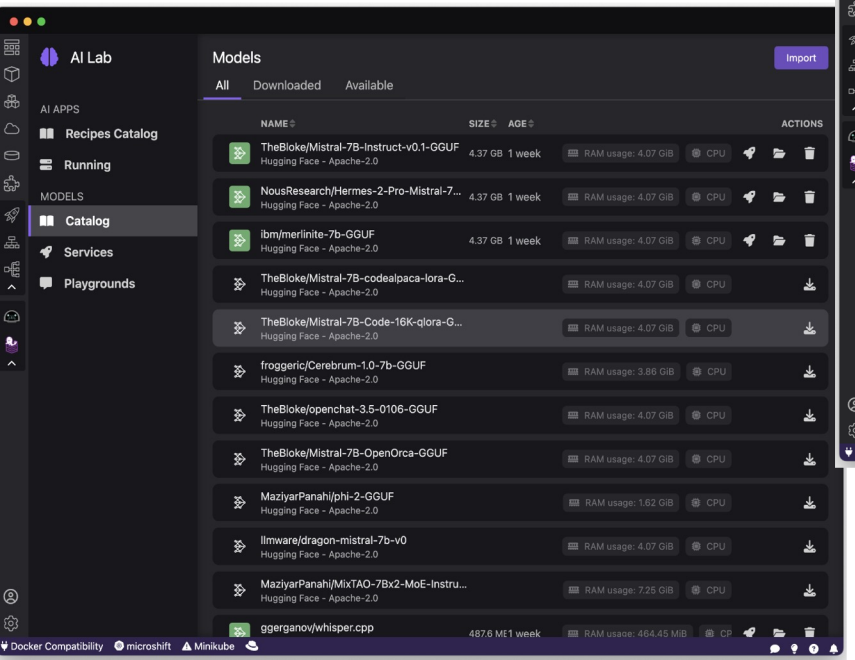
Cluster

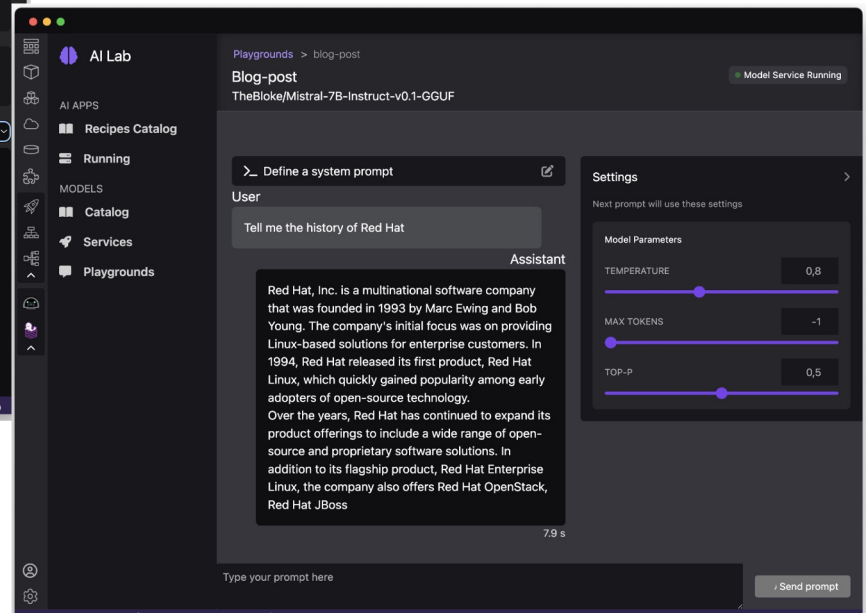
Podman AI Lab

- Podman AI Lab 是 Podman Desktop 的开源扩展，使开发人员能够使用在本地工作站上运行的直观图形界面在容器中构建、测试和运行有生成式AI加持的应用程序
- Podman AI Lab 提供包含示例应用程序的菜单目录，使开发人员可以快速开始了解LLMs的一些更常见用例，包括 Chatbots、文本摘要器、代码生成器等







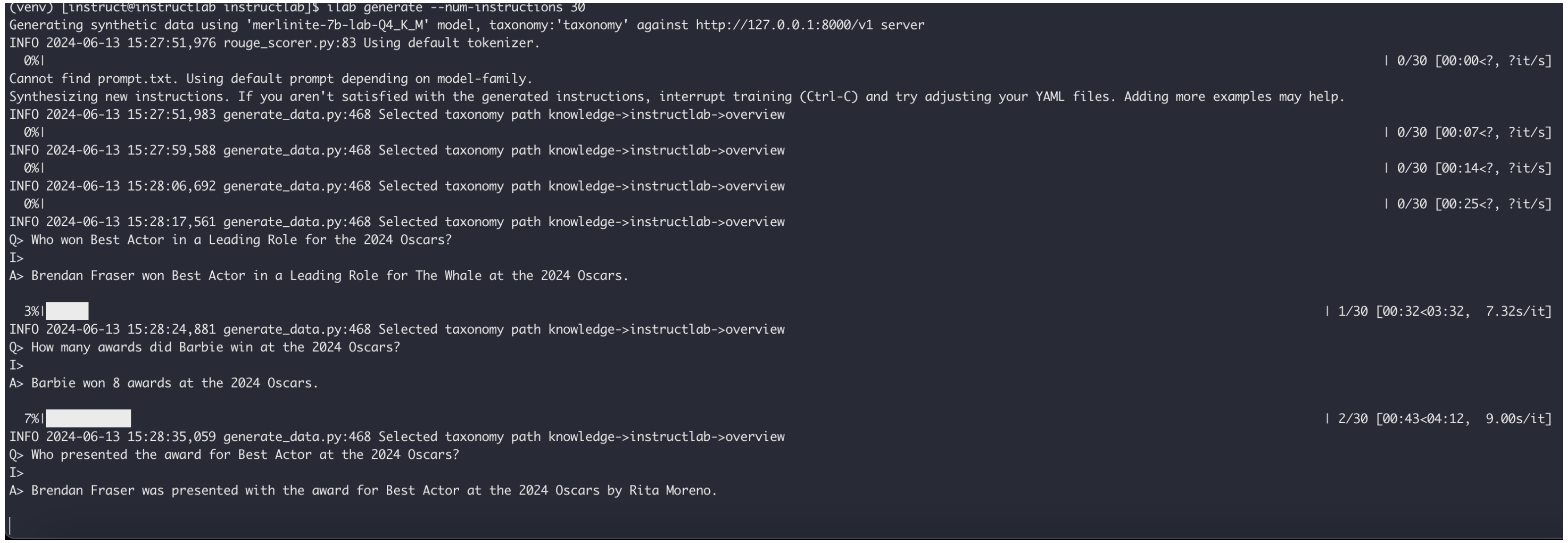


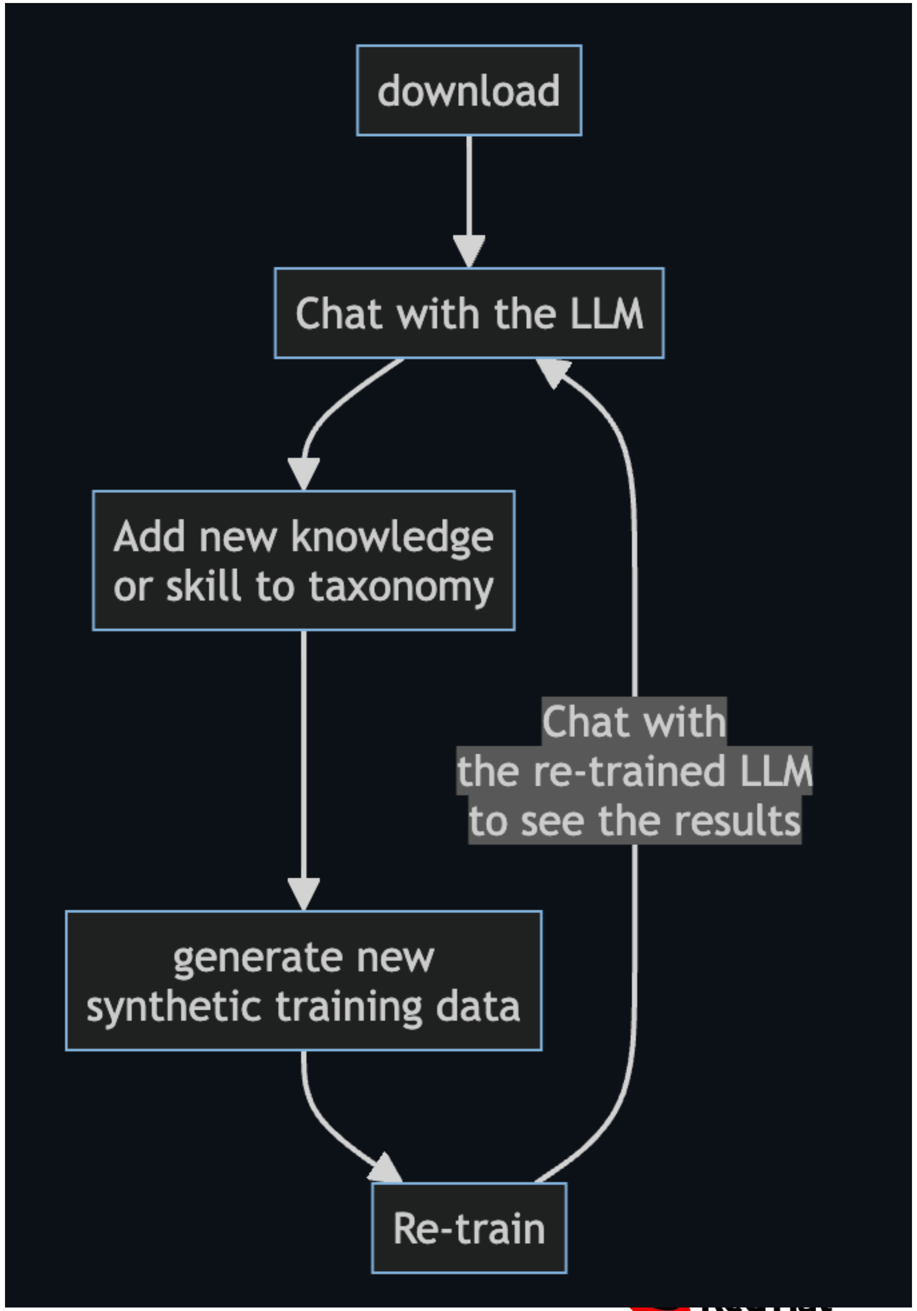
Demo

InstructLab进行模型训练

InstructLab Demo

- Granite模型下载与模型交流
- 基于分类法的合成数据生成
- 对于预训练模型进行训练
- 模型验证





Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make
Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

facebook.com/redhatinc

twitter.com/redhat

Red Hat