

大模型 领航者

InfoQ

对话先行者 | 洞见新未来

AIGC 实践案例集锦

【第一期】



推动数字人才全面发展 / 助力数字中国早日实现



人工智能产业链联盟

星主： AI产业链盟主

 知识星球

微信扫描预览星球详情





卷首语 PREFACE



霍太稳

极客邦科技创始人兼 CEO
2024 年 7 月

2024 年，AIGC 经历着爆发式增长的热潮。不仅各类国产大模型如雨后春笋般相继崛起，相关的商业化应用也步入了绽放期。从内容创作、数字电商、企业生产场景到医疗、教育领域，各行各业都在探索如何利用 AIGC 提高效率和创造价值，并被注入了前所未有的创造力与想象力。

随着技术的蓬勃发展，围绕 AIGC 的应用生态逐渐形成。在此期间，大模型领域内“百家争鸣”的激烈竞争，加速了行业价格战的兴起，更在无形中极大推动了 AIGC 技术普及至大众的进程。AIGC 的各类应用开始广泛从理想迈向现实，惠及越来越多的用户群体。

面对这样的发展趋势，极客邦科技旗下 InfoQ 极客传媒自 2024 年初，启动《大模型领航者》访谈栏目，并策划实施了一系列围绕 AIGC 和大模型的选题报道、专访、直播、演讲分享等多种形式的内容，聚焦于实际案例应用、大模型的实际部署、团队背后的故事等核心话题，与众多行业

专业人士进行了深入的对话与探讨。旨在深入了解当前 AIGC 前沿技术在企业应用领域的最新动态，为各行业提供实战经验与趋势洞察。

《大模型领航者》电子书正是基于前述内容，进一步做了筛选、提炼和分类。此次推出的第一期电子书收录了 InfoQ AIGC 频道在 2024 年上半年发布的超 20 篇文章，包括 QCon 全球软件开发大会（北京站）2024 和 AICon 全球人工智能开发与应用大会（北京站）2024 的热门演讲，以及对钉钉、面壁智能、数势科技、腾讯、京东、字节跳动、图灵机器人、云知声、零一万物、达观数据等企业的独家采访。

本册共包括“大咖视野”、“观点碰撞”、“应用案例”、“技术实践”与“AI 测评室”五个部分。其中，“应用案例”又涵盖互联网产品、企业生产提效、医疗、教育以及更多垂直行业的章节内容，分别从不同行业的视角展现了大模型应用的无限潜力；“AI 测评室”则囊括了今年各热门大模型的实测。

受限于电子书篇幅，我们无法尽数呈现 InfoQ 在 AIGC 频道的全部精彩内容。但与此同时，InfoQ 开设的「AI 前线」微信公众号正保持着每日的优质内容更新，不间断地向 AI 爱好者、开发者和科学家，持续奉上大模型最新资讯、AI 技术分享干货、一线业界实践案例，希望助力大家全面拥抱

AIGC。欢迎关注！

在这场前所未有的时代变革中，极客邦科技期待与大家并肩前行，一同深入探索 AIGC 时代的每一步发展，合力推动生成式 AI 领域人才的全面成长，以及规模化落地的宏伟愿景。



【欢迎扫码关注 AI 前线】

CONTENTS 目录

大咖视野 | VISION

- 01 | 钉钉卡位战：SaaS 挣不到的钱，Agent 会挣到 7
- 02 | 26 岁带着百人团队冲刺大模型，面壁智能天才 CTO：高效比参数更重要 13
- 03 | 这个离开大厂去 AI 创业的互联网大佬，带着他的“Killer Agent”来了 20

观点碰撞 | OPINION

- 04 | 大模型开闭源争吵不休：开源落后闭源一年，决定模型能力的不是技术？ 29
- 05 | “国外一开源，国内就创新”！面对中美大模型差异，我们该突破还是继续模仿？ 37

应用案例 | CASES

第一章：互联网产品

- 06 | 如何 1 秒内快速总结 100 多页文档？QQ 浏览器首次揭秘大模型实现技术细节 47
- 07 | 京东商家智能助手：Multi-Agents 在电商垂域的探索与创新 61

第二章：企业生产提效

- 08 | 字节跳动代码生成 Copilot 产品的应用和演进 71
- 09 | 大语言模型加持，是智能运维架构的未来吗？ 80
- 10 | 用 AI 面试员工的企业，知道打工人在想什么吗？！ 93
- 11 | AI 代码助手革新编程界：腾讯云专家汪晟杰深度剖析机遇与挑战 100

第三章：垂直行业

- 12 | 巨头们涌入的医疗大模型，何时迎来最好的商业时代？ 107
- 13 | AI 老师的强大功能 + 真人老师的情感交流 = 未来教育？ 112
- 14 | 4 人团队，如何用大模型创造近千万业务价值？ 127

技术实践 | TECHNOLOGY

第一章：大模型训练与推理

- 15 | 万字干货！手把手教你如何训练超大规模集群下的大语言模型 137
- 16 | 当大模型推理遇到算力瓶颈，如何进行工程优化？ 160
- 17 | AI 辅助内部研发效率提升，昇腾大模型推理的最佳实践 163

第二章：RAG 与智能体落地

- 18 | 智能体技术发展趋势：谈大模型智能体与开放领域融合 170
- 19 | Agent 还没出圈，落地先有了“阻力”：进入平台期，智力能否独立担事？ 175
- 20 | “驯服”不受控的大模型，要搞定哪些事？ 188

AI 测评室 | EVALUATION

- 01 | 算数不行、还不懂中国文化，大模型现在抢不了设计师的饭碗！ 195
- 02 | 首届大模型“相亲大会”开始啦！谁是你的天选 CP？ 206
- 03 | Kimi 的词 +Suno 的曲：能带我入选《中国新说唱》，但还是干不过原神！ 223

大咖视野

VISION



钉钉卡位战：SaaS 挣不到的钱，Agent 会挣到

出品 | InfoQ 《大模型领航者》
主持 | 霍太稳，极客邦科技创始人兼 CEO
嘉宾 | 叶军（不穷），钉钉总裁
作者 | 褚杏娟，InfoQ 编辑
编辑 | 蔡芳芳，InfoQ 主编



“

2020 年，刚带队做出全国第一张健康码的不穷加入了钉钉。不穷是阿里的第一位校招计算机博士，从 PC 时代开始触网，完整经历了移动互联网时代。而他如今面临的，是一场关于大模型的竞备赛。想必已经无需用过多笔墨赘述。2022 年底至今，

ChatGPT 的出圈程度还没有谁能超越，其背后的技术方向也早已经被竞相追捧。就像不穷说的“大家都充满了 FOMO 情绪。”

那么身处其中的钉钉，能够在这次浪潮中扮演什么样的角色？

和很多企业一样，钉钉最初也没有一下就找到合适的入局方式。一年多前，看到自己与微软不约而同地都选择了给当前产品增加 AI 能力时，钉钉团队觉得这种方式已经很让人眼前一亮了，但大模型爆发的能量远比想象得还要大。

整个 2023 年，OpenAI、微软、谷歌等大模型发布频繁，李开复、王小川等也亲自下场发布大模型……去年 4 月，钉钉全面投入智能化，开始用大模型逐个将高频产品重做。到了当年六、七月份，钉钉内部有人提出：能否有一个直接 AI 原生的产品？这引发了内部关于从“+AI”到“AI+”的争论。这个想法与去年 10 月李彦宏公开提到的 AI 原生

理论异曲同工，可见国内的探索思路其实差异并不大。

那什么是 AI 原生？钉钉也在思索。其实在将思维转换成以 AI 为中心后，这个问题就不难回答。

“AI 原生产品从一开始的思考就是全新的，就是要用纯 AI 的思路来解决一个任务。它从数据感知、任务分解，再到思维链，最后到行动执行，是一种全新的思考架构。”不穷表示。

而对于做 AI 原生的方式，钉钉选了 Agent。

“我们依赖这些大模型公司”

今年 1 月，OpenAI 正式推出了 GPT Store。几乎同时，钉钉发布了 AI 助理，并宣布 4 月推出 AI 助理市场。

“GPT Store 上线的第一天我就用了，当天就已经有非常多的产品了，速度非常快。但我分析了所有数据后发现，GPTs 同质化非常严重，而且都是一些通过简单限定词、指令或角色扮演来形成的 AI 助手。”不穷说道。

这一体验让不穷认定，钉钉未来的 AI 助理市场不做全量推荐，只会推荐精选过的 AI 助理。

不穷强调，钉钉模式与 GPT Store 的不同：GPT Store 通过不断丰富插件使 AI Agent 能够批量调用外部系统的能力，但它的组合效率要比钉钉低，因为钉钉拥有天然的 To B 环境，其中有大量的工作任务需要解决。

不穷认为，目前钉钉的能力不在于做自己的大模型，而是在应用和数据，在工程性、用户体验以及 To B 理解方面。对于 Agent 来说，大模型只是其中的一个能力，此外还需要非常好的场景和高质量的业务环境数据，这两者恰恰是钉钉有、而 OpenAI 目前还欠缺的。

对于与国内大模型公司的关系，用不穷的话说是：“我们依赖这些大模型公司，它们是我们的发动机和心脏。没有它们，我们无法运行。”

面向 AI，不穷把钉钉定义为 AI 应用创作平台，企业在这个平台上连接、开发和加工各种应用。钉钉的核心任务就是连接场景和数据，实现结构性自动化和批量处理各种工作，并通过 Agent 让创作变得更简单。

具体来说，钉钉的职责是确保外部记忆存储部分的完善，包括短期和长期记忆的处理，同时做好任务规划，之后将大模型生成的内容与本地业务数据集成，并将形成的行动在各个系统中落地。

在不穷看来，当前国内各个基础模型之间的差距并不大，未来不是每个开发者都会关心基础模型的选择，他们更注重解决业务场景中的问题。因此，

如果基础模型效果不理想，开发者应该可以随时更换。

因此，为快速上线和体验，钉钉选择了通义千问作为默认大模型，除此之外用户有需求时还接入了其他大模型公司的模型，如智谱 AI、月之暗面、Minimax 等。用户的业务逻辑可以建立在自己选择的基础模型体系上，业务流程和数据流也不会进入钉钉平台。

“根据不同的场景和需求，我们可能还会推荐小模型或专用模型。”不穷说道。

不穷在给用户提供模型的选择建议时，会提醒他们更加关注模型的性能，如每秒处理的 token 量；大模型的安全性问题等，如本地部署还是云上部署；工程解决方案的多重性和便捷性等。这些问题也是钉钉构建 AI 助理时实际遇到的。

“C 端还没有太多优秀的产品形态出现”

无疑，GPT Store 的模式吸引了大批用户：刚正式发布时，OpenAI 就宣称已经有超过 300 万个 GPTs。

与传统软件相比，AI 助理、GPTs 等的不同之处在于拥有非常快的更新速度，模型、交互方式、

数据和产品形态等方方面面都变得迅速，开发者也不要从头到尾进行开发和维护。这种模式还大大降低了开发门槛，没有研发背景的人也可以尝试，而对于研发人员来说则大大缩短了研发、测试等成本。

一方面，这意味着传统软件的研发模式可能会面临变革；但另一方面，不穷也指出，百万千万级的 GPTs 目前看相对来说形式比较单一，没有传统软件那样强大的业务理解能力，因此目前 GPT Store 中的应用很难成为高价值产品。

不穷认为，尽管 GPTs 的创建能力很强，甚至一天可以创建几十个，但它目前还代替不了传统软件市场。

钉钉也在寻找有价值的产品。在 1 月份宣布启动的 AI 助理创造大赛上，目前有超过 2000 支队伍提交作品，不穷也会亲自体验这些 AI 助理，寻找优秀的作品。

那么，个人玩家又如何在 GPT Store 这种模式中赚到钱呢？

不穷的答案是价值，“只要有价值就一定能挣到钱，只是迟早的问题。”在他看来，个人或企业创建 AI 助理的核心在于要解决具体的问题，解决问题本身就有价值。但现在“卖工具”的人可能不是最终解决问题的人，解决问题的人是那个场景中

离问题最近的人。

AI 助理的商业模式则与传统软件相似，需要一定的用户使用量，“只要使用量上去了，很快就会有开发者赚到第一桶金。”根据不穷的经验，一旦调用量达到百万次，软件做商业化就是必然的。

这与之前钉钉在与 IDC 联合发布的《2024 AIGC 应用层十大趋势》中提到的观点“新一轮的 AIGC 之争，也将会是一场流量入口之争”是契合的。钉钉在其中也提到了有望成为超级 App 的想法。

“在目前的 To C 场景中，我还没有看到太多优秀的产品形态出现。”不穷说道，“但是，AI Agent 绝对不是自我陶醉，我相信一定会有出色的产品出来，应该给创新者更多的时间。”

根据不穷之前的访谈，AI Agent 和智能助理产品发展到一定程度后，中间态、碎片化的产品成为极简流量入口，就会出现“No App”理念重塑应用的情况：通过对话即可直接调取、使用各种工具，更多非软件专业人员也能获得强大的系统服务。

“SaaS 挣不到的钱，会通过 Agent 挣到”

相较 C 端，B 端是不穷更看好的方向，因为 To B 场景更容易产生有效的产品：确定的数据和场

景可以帮助解决大模型的幻觉问题，同时通过批量和自动化的方式提高 To B 常见工作流和任务

流的效率问题。

不穷指出，To B 软件的目的是解决问题，所以这里天然聚集了大量的问题和数据。在这样环境里构建的 AI Agent，传统 SaaS 和 PC 软件软件的开发流程、产品交互、形态及维护等的缺陷都将得到弥补。未来，SaaS 的定制化或各种行业需求，都可以用简单、低成本的方式实现。

“Agent 市场形态肯定会取代传统软件市场形态。更重要的是，它将取代传统 SaaS 的产品形态。”不穷说道。

现在 AI 助理的 to B 服务中，钉钉要与用户频繁、深入地互动，根据反馈不断调整和改进。比如在解决一家芯片企业客服培训难题时，钉钉团队要去公司了解实际工作流程，然后将线下流程转化为线上的 AI 助理。

虽然每个企业的需求相同，但产品会逐渐沉淀下来。企业自行完成标注、训练和本地化数据接入，钉钉则保留抽象层和公共层，逐渐完成产品的广泛行业适用性。

不穷评价 OpenAI 做产品就像是科学家通过成千上万次的实验，最终找到一个正确的方向，逼近科学真理。而钉钉则投入大量时间与客户共创，解决他们的实际问题。两者虽然方向不一，但殊途同归。

与大模型创业公司苦苦寻找自己的商业模式相比，钉钉探索出来的大模型商业模式已有两种。

第一种是基于调用量的模式。无论个人 AI 助理还是企业助理，产品使用频率越高、解决问题的能力越强，吸引的用户就越多，自然也就需要更多的调用。使用量大，消耗的算力和资源也就越多。

第二种是应用层本身带来的商业模式。传统的 SaaS 模式赚钱较为困难，因为它需要大量的定制和本地化需求，AI Agent 的应用能力提供了一个解决方案：

简洁的界面、任何需求都可以通过对话来理解，并通过行动系统对接外部系统逻辑，这样就将界面定制化和流程重构的职责就交给了后端模型和 AI Agent 系统。这样，从交互层到模型层，再到持久层，整个过程都得到了简化。因此，SaaS 的维护成本也就降低了。

“AI 助理的盈利天花板目前还看不到，随着更多优秀产品的出现，我们可能还会发现新的盈利途径，带来新的惊喜。”不穷说道。

不穷认为，通过消耗算力来提供服务只是最基本的模式，除此之外，服务消耗还有很多其他的可能性。他的判断是，未来十几年中国 SaaS 行业挣不到的钱可能会通过 AI Agent 来实现。

结束语

在提到当前钉钉 AI 助理接下来要重点攻关的方向时，不穷还是说到了数据和场景：

数据和场景是 Agent 普遍存在的问题，钉钉的 AI 助理现在有更专注的场景和数据，就像是给“孙悟空戴上了紧箍咒”，好处是能够减少幻觉、能够解决一些通用场景里难以解决的问题。这也意味着，钉钉未来还需要发掘和洞察到更多的场景、沉淀和积累更多的高质量数据。

其次，行动能力是目前 Agent 所欠缺的，只是让它们聊天未免太乏味。因此 AI 助理会接入钉钉上原有的应用、低代码等开放能力，不穷希望以此让 AI 助理能够不断出现各种创新玩法，而不仅仅是简单的信息查询和单向交互。比如，AI 助理对接了很多的主流 App 行动系统，比如可以查看淘宝订单等，App 的行动系统实际上就变成了一个

AI 助理，无需在不同系统间切换。

在不穷看来，Agent 的最大好处就是它的无限可能性，这种模式不受传统思维和现有框架的限制，是真的可以让想象力转化为生产力的。

“我今年非常期待行动系统能够变得更加强大、数据质量得到提升。随着越来越多的人洞察到新的场景，AI 助理将不再是一个个废话大师、一个个应对亲戚的聊天工具、一个个面试官。”不穷说道。

访谈里，不穷不掩对微软战略眼光的称赞。“战略需要耐心，如果没有耐心，那就只是投机。”同样地，钉钉对 AI 助理的耐心有多久？AI 助理未来的价值能有多大？这些也是不穷现在要面对的课题。



扫码分享文章

26 岁带着百人团队冲刺大模型，面壁智能天才 CTO：高效比参数更重要

出品 | InfoQ《大模型领航者》
主持 | 霍太稳，极客邦科技创始人兼 CEO
嘉宾 | 曾国洋，面壁智能 CTO
作者 | 褚杏娟，InfoQ 编辑
编辑 | 蔡芳芳，InfoQ 主编



“

尽管有所谓的‘百模大战’，但实际上，国内真正能够成功训练大模型并掌握相关技术的团队并不多。”面壁智能 CTO 曾国洋说道，“不是简单地训练出一个模型就意味着掌握了全部技术。”

面壁智能起于一群学术极客。2021 年，清华大学计算机系长聘副教授刘知远的牵头成立了面壁智能成立，团队成员主要来自清华大学 NLP 实验室，而曾国洋成为这家初创公司的技术 1 号位。

曾国洋如今更以“天才少年”的形象被人熟知：8 岁学编程、高中去旷视实习、大二加入清华 NLP 实验室。人们通常很难将眼前这个 98 年的少年，跟“BMTrain、BMInf 主要作者”“OpenBMB 开源社区发起人”“当红大模型创业公司 CTO”等联系在一起，但 26 岁的他确实已经被推到了大模型时代的舞台中央。

从自己 coding 到看别人 coding

2022 年 8 月，面壁智能开始公司化运作。直到去年年初，面壁智能只有 10 个人不到。当时的曾国洋依然活跃在编程一线。

作为程序员的曾国洋，是早期第一批申请试用 GitHub Copilot 的用户之一。他把 AI 看成是合作伙伴：AI 辅助程序员完成某些任务，而程序员则可以专注于更具创造性和战略性的工作。

“我很喜欢能够帮助加速编程的工具，”曾国洋说道，“我们不应该简单地认为只要代码被写出来，程序员的工作就完成了。编写代码只是程序员工作的一部分，如何将想法架构化以及合理划分模块并确保它们之间的有效协作等，都是程序员工作中相当重要的一部分。”

去年 5 月份后，面壁智能的规模越来越大，内部也设立了数据处理、模型训练、模型评测、算法、Infra、运维等不同的团队，以便更好地训练大模型。他的工作重心逐渐转为保证组织的有效协作。

在此期间，面壁智能迎来了许多对通用人工智能（AGI）充满激情和信仰的年轻人，“他们对 AGI 有浓厚的兴趣和追求，甚至愿意降薪过来。”

但在爆火之前，大模型并没有被广泛关注和应用，因此有相关经验的人才很少。这意味着几乎所有

人都是从头开始学习和探索大模型。因此，团队在招揽新人时并不把大模型经验放在首位，而是更看重候选人的学习意愿、对新技术的热情、以及创新和解决问题的能力。

如今，面壁智能已经拥有超 100 人的科研团队，平均年龄 28 岁。这支团队的“清北”含量 80%，此外还有来自阿里、字节、百度等一线公司的骨干。

面壁智能没有给技术团队设立严格遵循 KPI 的管理形式，也没有在每一个非常具体的时间点设定明确规划，只是制定了一个大概的发展节奏和方向，因为合作的都是顶尖聪明的同事，而聪明人是会自己给自己定目标的。“我们要做的不是个人明星，而是明星团队，让聪明人能更好地合作、互相创造价值，一起创造更伟大的价值。”

面壁智能倾向“小而美”的技术团队。曾国洋强调，“小而美”并不是说团队规模小，而是指团队能够保持高效、灵活和创新的状态，成员能够频繁交流、头脑风暴，共同推动项目发展。对于技术创业公司来说，这样的团队更加敏捷和灵活，更容易产生新的思想和创新。每个成员能充分发挥自己的专长和创造力，同时快速响应市场变化和技术演变。

大模型团队的研发速度可以用争分夺秒来形容。面壁智能团队之前以两周为单位的内部迭代频率

已经成为过去式，如今的节奏已经将近一周一迭代了。不断演进期间，也让面壁智能对自己做的事情有了更深入的思考。

不再一味追求参数

国内庞大的市场规模为大模型创业提供了巨大的发展机遇，但 OpenAI 等国外公司的频繁迭代，确实也给了国内公司很大的技术压力。时至今日，很多公司的大模型发布出来时，都是对标的 OpenAI。

不过，曾国洋表示，“我们并不过分担忧落后的问题。”他分享了一段自己的经历：

ChatGPT 刚刚发布时，大家都赞叹它的强大能力并讨论需要投入多少资源才能追赶上。后来，我自己投入了一些资金，买了几百条数据训练我们的模型。那次训练完测试后，我感受到了 ChatGPT 的那种效果。这个瞬间让我意识到，我们离它实际上并没有想象中那么遥远。

这个经历不仅让我自己感到振奋，也给了我们团队巨大的信心和动力。它证明了我们的努力和方向是正确的，只要我们继续坚持，完全有可能达到甚至超越行业领先者。

曾国洋有作为技术人的自信和思考。

“我们将 OpenAI 的成就和国际市场的竞争态势当作一种衡量自己的标杆，但不会盲目跟随。我们清楚地认识到，OpenAI 的技术路线可能并不适合我们，我们需要根据自己的实际情况和优势来制定发展策略。”曾国洋说道。

回顾 2023 年，面壁智能一直略显低调地走在大模型潮头：当年 5 月，发布了百亿参数的 CPM-Bee 大模型；年中，推出了千亿参数多模态模型 CPM-Cricket，综合能力对标 GPT-3.5、超越 LLaMA 2。

但在 2021 年、2022 年，国内在大模型上进行了大量探索，但最终都没有出现一个像 ChatGPT 的突破性应用。这让面壁智能的技术团队意识到，一味地追求模型参数量行不通，训练出一个大模型也不是最难的部分，更难的是如何突破模型的智能极限，在用同等参数、同等数据量的情况下，更快速低成本地跑出更好的模型性能。

在曾国洋看来，未来大模型的发展应该朝着高效率的方向发展：大模型要为用户带来更大的价值

和更广阔的商业空间，而这主要取决于模型创造的价值和创造这一价值所需的成本。

今年 2 月份推出的 MiniCPM 模型就是面壁智能对大模型高效探索的样板间。发布会上，面壁智能 CEO 李大海提出了要“以小搏大”，曾国洋也表示 MiniCPM 用 2B 干掉 LLaMA 的 13B。这意味着，面壁智能正式进入小尺寸端侧模型的竞技场，并且还将其完全开源，以帮助大模型行业整体技术发展。

起初，端侧模型并不在团队计划中，但是在测试中发现并验证了这么高性能的模型可以在手机上顺畅运行，这给团队打开了新世界的大门：一旦模型能够在手机上运行，他们就能在端侧探索出更多应用场景，如汽车、VR、智能家居等场景。

端侧模型的优势在于，不需要频繁与云端服务器通信，因此处理速度更快；在本地设备上运行，不需要消耗大量的网络带宽和云计算资源，具有

成本优势；可以在没有网络连接的情况下仍然发挥作用，这意味着其可以在各种环境下稳定运行。

端侧小模型的性能天花板也远未达到。在模型的极致效率方面，通过模型压缩、量化、剪枝等，性能可以进一步优化。其次，端侧设备本身也存在优化空间，硬件制造商可以考虑如何在硬件设计上更好地支持大模型运行。

“我有预感，像 GPT-3.5 这样高水平的模型，可能在一两年内就能在移动设备，比如手机上，完全运行起来。”曾国洋说道。

在面壁智能看来，大小模型的技术有互相打通、增进提升之处。面壁 MiniCPM 基座模型、多模态模型等“小钢炮”系列领先的端侧模型，都是基于公司千亿级模型研发路线延伸，将淬炼化的大模型训练方法下放至小模型训练中，来实现高效、低成本的模型训练与应用。

“不会因别人而改变”

变化，是大模型创业公司时刻要面对的问题。就像曾国洋常常被问到：Transformer 会不会突然被新的技术取代，从而让之前的投入都白费？

曾国洋对技术的快速变化并不过分担忧。“技术的发展是一个循序渐进的过程，不可能一夜之间出现一个全新的技术彻底颠覆现有的一切，而我们对此毫无准备。”在制定研发路径时，团队也是根

据技术发展趋势和团队正在进行的工作，逐步调整目标和方向的。

对于市面上时不时蹦出来的热点模型或产品，曾国洋也表现得很冷静。

以 Sora 为例，曾国洋认为这显示出人们对创造性工作的兴趣，但对于是否跟随这一技术路线则需慎重。

“对于创业公司来说，需要格外考量战略目标与投入成本。即使是资金充裕的大公司，虽然有能力强进行，但产出并不总是明确，短期内可能无法快速为大众提供实质性的服务。”曾国洋说道，面壁智能致力于将技术更好融入到实际产品和解决方案中。

对于前段时间刷屏的月之暗面 Kimi，曾国洋则一方面表示肯定，“Kimi 用户的增长迅速，表明它成功地解决了一些用户的痛点”，另一方面也反思自己，“可能没有充分利用我们在某些方面的先发优势。”

他特别提到了去年 5 月份面壁智能推出的一项读论文功能，虽然早就有了类似产品，但当时很可惜没有深入挖掘和清晰传达该功能可以解决的痛点。

但曾国洋强调，面壁智能的战略不会因为市场上的其他产品而改变。“我们一直在寻找大模型技术在普通人生活中的应用，并努力解决实际问题，而不仅仅是提供通用的解决方案。”

最近，李彦宏“开源模型会越来越落后”的观点也引起了很大的争议，有人“力挺”、有人“怒怼”。

对于面壁智能来说，开源是成立之初就做好的选择。正如李大海所说的：“我们一直是开源的受益者，所以也希望做出自己的贡献。并且，一款拥有良好口碑的开源模型，一定是经受住了方方面面的反复检验，在模型性能、体验等综合表现方面，拥有远超过 PPT 成绩的行业认可度。对于我们研发团队，一方面是 360 度无死角的考核压力，另一方面在挑战成功后也会有巨大的成就感。”

曾国洋坦承，开源模型要追赶闭源模型确实会面临一定的挑战。这是因为在技术快速发展的过程中，闭源模型可能会因为有更好的知识产权保护和商业秘密而获得一定的优势。这种情况下，开源模型需要同时关注技术创新和与闭源模型保持竞争力。

当技术发展进入瓶颈期，那么开源和闭源模型可能就会在技术水平上趋于一致。在这种情况下，开源模型由于其开放性和社区的支持，会有更多的机会迎头赶上，甚至超越闭源模型。另外，开源模型的发展速度也取决于社区和市场的支持程度。如果有更多的个人和组织支持，那么开源生态的发展自然会更快。

此外也很重要的一点是，开源本身在技术影响力的建设方面是特别重要的，可以更好地让大众体验团队的技术实力，从而在人才吸引力和市场信心提升方面取得更强的竞争优势。

“没有刻意区分 C 端和 B 端”

对于以科研人员为核心创始团队的大模型创业公司来说，在战略、产品、经营等方面需要更强的专业管理者。2023 年，时任知乎 CTO 的李大海加入面壁成为 CEO，面壁向更为成熟的大模型商业公司迈下重要一步。

今年 4 月，面壁智能又完成了新一轮数亿元融资，由春华创投、华为哈勃领投，北京市人工智能产业投资基金等跟投，知乎作为战略股东持续跟投支持。除了通过融资获取资金外，面壁智能目前已经能够通过提供服务和产品实现一定的收入。

面壁智能是国内最早探索 Agent 的大模型公司之一。对于 Agent，每个公司、每个人的理解都不一样。在面壁智能看来，Agent 的边界还未被定义。“模型是底座是一切应用的基础，然后 Agent 是支撑应用很重要的中间层，”李大海认为，无论 to B 还是 to C，本质上都是“大模型 + Agent 的上层应用”。

曾国洋认为，Agent 实际上是介于纯大模型和通用人工智能（AGI）之间的一个中间状态或节点。Agent 的模型能力必须足够强大，才能有足够的智能理解和处理请求和执行任务。Agent 还需要与外部系统和接口进行交互，来不断拓展能力边界。同时，还能够调用已有的知识库来提供检索和回答服务。

而对于大模型领域，李大海曾表示，大模型应用可能会百花齐放，然而通用千亿大模型不会太多，

可能只有极少数的几家公司能够最终突出重围。这基本也是行业的共识，基座模型的角逐注定是千军万马过独木桥。

那么，大模型公司做应用是对应用侧公司的一种降维打击吗？

曾国洋坦言，大模型公司由于其先进的技术和强大的数据处理能力，可能会对那些依赖传统技术或缺乏足够技术储备的应用公司产生影响。如果应用公司的技术壁垒不够坚固，就可能会在大模型技术的快速发展和迭代中受到冲击。例如 Jasper AI 这样的 AI 内容提供商可能会因为 OpenAI 发布了新的 ChatGPT 版本而受到影响。

然而，他也表示，应用公司也有自己的竞争优势，比如对特定市场的深入理解、强大的客户关系和品牌忠诚度等，这些都是他们的“护城河”。

2024 年，行业更加重视应用落地是当前的大趋势，因为目前模型已经基本可用了。李大海判断，从今年开始，大模型厂商会开始出现分层。但这个分层不是因为市场，更多是因为技术门槛：大家需要更强的模型、更高效率的推理，更好的 Agent 等，但不是每家公司都能跟得上这样的技术要求。

“大模型是一个行业级别的机会，哪怕不做基座大模型，做应用层也有非常多的空间。但不是每家

公司都能够持续做基座大模型的训练，期间有些公司可能就会转型做其他的事情。能活下来的公司一定是技术和产品市场能力都很强的选手。”李大海表示。

对于未来的大模型应用，曾国洋提出了一种分工模式：一些简单的、重复性的任务由小型的、特定领域的模型来处理；而更复杂的、需要高级认知能力的思考任务则可能由大型通用模型来完成。

结束语

最初，人们普遍认为 AI 会先替代那些繁琐的工作，从而让人类有更多的时间从事创造性的工作。但现实情况似乎相反，AI 开始在创作领域发挥作用，而人类仍然在处理日常的工作任务。

但曾国洋观察，大模型简化、加速开发任务是正在发生的事情。在创建大模型应用企业的交流中，技术人员并不需要掌握大量的编程代码，更重要的是他们对最终产品的理解能力。

“对于想要有效利用大模型的人来说，掌握大量的编程能力并不是必需的。更重要的是能够将自己的思维方式与模型同步，确保模型理解并执行自

己的想法。”曾国洋说道，“让模型理解你的想法是一种独特的体验，它要求用户对模型的运作方式有一定的了解，并且能够清晰地表达自己的概念和目标。”对于其他公司来说，大模型技术会以不同形式的工具、功能出现。

面壁智能的目标是实现通用人工智能，团队为此有一个清晰的路线图，包括在文本模态上要达到的效果、未来向多模态和具身智能的转变等规划。2024 年，面壁智能将继续专注于模型的研发和优化。而面壁智能的选择能否助其顺利发展、实现自己的 AGI 理想，还需要时间来回答。



扫码分享文章

这个离开大厂去 AI 创业的互联网大佬，带着他的“Killer Agent”来了

出品 | InfoQ 《大模型领航者》
主持 | 霍太稳，极客邦科技创始人兼 CEO
嘉宾 | 黎科峰博士，数势科技创始人兼 CEO
作者 | 褚杏娟，InfoQ 编辑
编辑 | 蔡芳芳，InfoQ 主编



“

2020 年是一个不同寻常的年份。人们因新冠不得不停下来的时候，时任京东集团副总裁、技术委员会主席、京东商城技术负责人的黎科峰博士却选择了离职创业。此时的黎科峰博士，已经在互联网行业摸爬滚打了十余年。

他曾在三星亲历了公司第一代智能手机诞生；在平安集团作为执委、平安金融科技 CTO 推动平安集团的用户和数据体系建设，以及金融科技云

平台建设；在百度，作为原百度云计算创始成员推动百度云 OS 初创，也曾担任过手机百度总经理，带领实现日活跃用户 1 亿的突破。

意料之外，情理之中。在经历了 PC 和移动互联网两次浪潮后，黎科峰博士明显感到 To C 的流量增长已经放缓，红利期基本结束，大厂们也开始在 To B 领域布局。To B 就是黎科峰博士给自己选的赛道，而这次他恰好赶上了大模型这波浪潮。

黎科峰博士一直说自己是“To B 新人”，但他创业 3 年后就让公司营收达到数亿元的成绩，已经无法让人用“新人”来定义他。

C 端大佬，凭何入局 B 端

“创业，始终是一条不同寻常的道路。别人已经走过并走通了的路，没有必要再去重复。”

中国的 To B 公司大致有两种。

一种是专注项目制和定制化服务的公司，根据客户提出的具体需求逐一开发。但问题是客户的需求是零散和片段化的，他们可能并不清楚自己的数字化是什么样的。

另一种则是“Copy to China”模式，商业模式、产品理念，甚至 PMF 理论很多是直接来自美国照搬过来。但中国的市场环境和美国不同，简单复制很难成功。

企业内部通常分为利润中心和成本中心，其中成本中心包括了行政、人力资源、财务和 IT 部门等。国内许多 To B 软件实际上是围绕成本中心设计的，比如 OA 系统等，但这些软件的付费能力相对较弱，转化为直接业务价值的路径较长。

正如黎科峰博士所说，“企业高层或数字化负责人

很难有底气地说，企业增长的 20%–30% 是他们带来的。”

表面上看，国内数字化认知不足、大家付费意愿不强等都是 To B 企业面临的普遍挑战。但长期从事 To C 产品的经验告诉黎科峰博士，问题的关键是从业者没有解决真正的痛点、真正帮助用户创造价值，To C 还是 To B 反而并不重要。

根据黎科峰博士观察，国内数千万的企业正在从过去依靠规模增长的模式，转变为追求内在能力提升和经营效率提高的模式。企业意识到，仅凭直觉和经验已经难以持续增长，开始寻求技术的帮助。

但不同行业的软件使用者技术水平存在差异。电商、金融科技等互联网企业数字化水平较高，员工可以熟练使用复杂软件，但其他行业的企业就难以有效使用，致使其软件使用成本很高。

因此，在黎科峰博士看来，To B 行业要真正发展，首先需要建立共识，即让市场参与者普遍认同软件或解决方案的价值，否则就得花大量的时间和

精力说服和教育市场。其次，产品必须足够简单易用，能轻松融入业务团队的日常工作，并且无需复杂培训。

大模型产品完美符合上述条件。

不用多说，大家已经对大模型的价值和潜力有了共识，这是一个非常重要的优势。另外，大模型降低了数据的使用门槛，让 AI 和大数据不再只属于技术圈，而是广泛的普通大众。

如今，虽然不是每家企业都开始实际应用大模型，但至少都在研究、了解和学习。走得远的企业已经开始尝试将大模型技术应用于实际业务中，甚至有些已经签了商业合同。

“这种对大模型技术的高度认可和期待，甚至超过了当年的云计算。”黎科峰博士表示。

传统的 To B 软件十分复杂，大多数功能用户并

不需要，同时很久都不做本质上的改进，长期保持最初的界面和功能。而通过基于大模型软件，用户通过简单表达就可以实时、精确地找到所需信息，软件还能不断适应和学习，越来越贴合用户的个性化需求和使用习惯。

“从这个角度来看，大模型技术会颠覆现有的 To B 软件，甚至让很多 To B 软件过时。”黎科峰博士补充道，“当然，现有的 ToB 软件开发商也会努力追赶，但他们可能会因为历史包袱而受限。”

另外，大模型产品还将改变 To B 业务的决策关系。

黎科峰博士表示，国内软件的使用者和决策者往往不是一线员工，而是不使用软件的管理层。大模型产品则让每个员工都能通过使用工具产生价值，这样决策链将不再只是老板，还有真正使用软件的员工，员工对软件效率提升的需求会影响决策者。

如何实现大模型价值落地？

“现在的大模型就像一个全科研究生，而 Agent 是帮助它成为某个特定业务领域专家的实现方式。”在明确了大模型对于 To B 的影响后，该如何应用大模型呢？

困扰很多人的首先是如何选择大模型。根据数势

科技的基准测试，国内大模型产品之间的差距并不显著，与国际领先的模型，如 GPT-4 等比较，算力、准确度等方面确实存在较为明显的差距。

但对于如何应对和看待国内外大模型的发展差异，黎科峰博士认为，“核心是我们是否需要像

OpenAI 那样投入巨额资金（数百亿）来发展大模型技术。”

OpenAI 开始不计商业成果，致力于推动通用人工智能（AGI），后来也得到了美元基金支持，能够获得的资金是中国企业的很多倍。中国的大模型厂商在资源有限的情况下，需要从一开始就明确自己的方向，思考如何创造价值并实现商业化回报。这涉及到选择哪些方向进行投资，以确保资金的使用更加高效和有针对性。

另外，大型企业和创业公司开发的大模型产品方向也各有千秋。

大企业开发大模型有明显的资源优势，比如丰富的计算资源和大量数据积累，然而也面临着诸如如何将大模型整合到现有业务和产品线等较重的商业化包袱。

而创业公司没有历史包袱，通常从大模型原生视角出发，不受以往业务约束，更自由地探索和创新，因此某些产品的体验可能不逊色于大型企业。

当前，一些财力雄厚的大型企业，如银行和国有企业会先投资建立大模型基础设施，如购买 GPU 和部署具有数千万、甚至数千亿参数的大模型，然后逐步挖掘需求和应用场景。

但更多的企业是先找应用场景，再立项、部署大模型。这些企业会梳理出一系列需求，然后据此决定大模型的应用方向。例如，如何快速准确地找到数据、基于数据进行分析、洞察业务问题以及指

导经营以提升成果，通常是很靠前的考虑因素。

黎科峰博士透露，目前率先尝试大模型应用有经营分析（与数据紧密相关的金融、零售和高端制造行业）、内容生成（如营销活动所需的视频或图片生成）、企业知识库（从企业内部的知识库中提取关键信息，以便员工更好查找和学习）。企业经营分析是目前企业最感兴趣的领域，超过 90% 的企业表现出了明确需求。

大模型具备知识、智商、学习能力和推理能力，能够总结和生成新的见解。AI Agent 的本质是智能体，内部包含许多规划业务流程规划器，能够完成基于业务目标的任务规划和自动执行。

数势科技的做法是，大模型叠加 Agent，让企业应用具备记忆、反思和学习能力，能够调用企业内部工具并不断迭代反思，真正实现业务价值落地。

不过，找到了应用场景后就让 To B 赛道的创业公司短期内盈利，黎科峰博士认为这并不合理。

“很多企业在没有想明白胜利逻辑和商业化可能性之前，就急于跟风，最终导致失败。”黎科峰博士说道，一个企业的价值不在于目前是否盈利，而是在于它的商业模式是否成立。

就像美国的许多企业一开始并不盈利，但找到正确的盈利模式后会非常赚钱。黎科峰博士认为，这背后的关键原因是，企业要找到产品能够为企业带来的业务价值，并且具有强大的复制性，实现边际成本递减，这样才能实现高额盈利。

如何在内卷的环境里赚到钱？

“盈利可以通过裁员或非健康的手段实现，但这并不是我们追求的。”

“商业化是在大厂打工不需要太考虑的问题，但自己做企业就要考虑怎么活下去。”黎科峰博士曾在分享中说到，但他在大家还在探索大模型应用的时候，已经用大模型 Agent 赚到钱了。

“真正赚钱的是上层应用。”黎科峰博士分析道，未来，全球企业的业务经营将依赖于工具，而不是人、政策或剥削员工。最有价值的企业将是那些拥有数据、能够通过工具提升效率的企业。

黎科峰博士创建数势科技之初，公司重心是构建数据资产层，通过打造企业指标平台和标签平台将企业海量数据转化为可使用的数据资产，释放数据价值。但产品使用门槛较高，更加适用于数字化程度较高、拥有较多数据技术人员的企业。

大模型出现后，数势科技为企业用户提供数据分析 AI Agent，用户可以基于自然语言对话进行交互，如此降低软件的使用门槛，管理者、业务人员等非技术人员都可以准确、即时、个性化地进行数据查询和业务洞察，提升决策能力。



数势科技大模型 Agent 产品 (SwiftAgent) 架构图

“数势科技的优势是使用软件和算法解决问题，而非人力。”黎科峰博士说道，“我们的人员效率很高，是因为我们用技术解决传统上需要大量人力的问题，这也是为什么我们能够实现良好的盈利。”

那黎科峰博士一直强调的“用户价值”，是如何体现在产品中的呢？

他举了一个例子。在零售行业，一家茶饮连锁企业面临的挑战是如何让店长进行数据分析。由于店长的教育水平普遍低于科技或互联网公司，让他们快速、高效地利用传统 BI 工具做数据分析并不现实。而通过数势科技对话式、低门槛的数据分析和决策产品，店长能够通过简单的语音输入获取需要的数据，并理解数据变化背后的原因，从而更好地经营门店。如此，门店的运营不再简单依赖店长能力和经验。

国内市场存在一个非常普遍且残酷的问题：在资本市场状况良好时，任何一个新概念的出现都会吸引一大批公司迅速进入，结果就是甲方难以选择供应商，最终仅靠价格抉择。市场也因此变得

越来越卷，好的产品和公司不得不参与价格战，最终可能出现经营困难，造成了“劣币驱逐良币”的局面。

大模型市场正在历史重演。To B 企业要在如此的竞争环境中脱颖而出，黎科峰博士认为关键有两点：一是要真正理解数据智能和 AI，并有深厚积累，这是企业的基因；二是要回归客户需求、懂行业知识，讲能够引起客户共鸣的故事。

在产品技术方面，企业最重要的是想清楚 Agent 与大模型的依赖关系，优化 Agent 产品设计，使其更加高效地运作，减少与大模型的交互次数，仅将必要的任务交由大模型处理，从而节省资源。

在业务需求方面，数势科技并不是什么客户、什么行业都做，而是选择自己擅长、有很深 Know-how 的。这基于黎科峰博士多年大厂经验的总结。“在做技术负责人的时候，我的考核标准也不是单纯地看技术有多炫酷，而是技术能不能让企业赚钱、省钱，产品最重要的是为客户创造价值。”

结束语

“我带着一种无畏的态度进入这个行业，我没有过往的包袱，也没有已经形成的固定模式，这让我

能够更自由地探索和创新。”4 年前获得腾讯亿元天使轮融资的黎科峰博士，如今依然站在 To B

的赛道上。

虽说“好风凭借力”，但黎科峰博士清楚地表示，大模型技术虽然非常有潜力，但本质上依然是一个工具，是用来帮助企业实现业务目标和战略的。

对于黎科峰博士来说，大模型是武器，核心竞争力是自己懂业务、懂技术。

未来，数势科技利用大模型还能创造怎样的成绩，我们拭目以待。



扫码分享文章

极客邦科技，以“推动数字人才全面发展，助力数字中国早日实现”为己任，致力于为数字人才提供全面的、高质量的资讯、课程、会议、培训、咨询等服务。极客邦科技的核心是独特的专家网络和优质内容生产体系，为企业、个人提供其成功所必需的技能 and 思想。

极客邦科技自 2007 年开展业务至今，已建设线上全球软件开发知识与创新社区 InfoQ，发起并成立技术领导者社区 TGO 鲲鹏会，连续多年举办业界知名技术峰会（如 QCon、ArchSummit 等），自主研发数字人才在线学习产品极客时间 App，以及企业级一站式数字技术学习 SaaS 平台极客时间企业版，在科技人群、科技驱动型企业、数字化产业当中具有广泛的影响力。



欢迎关注
极客邦科技公众号



您有任何建议、反馈、投诉，或者表扬
欢迎扫码，直达 CEO 霍太稳

极客邦科技以 KaaS 模式（Knowledge-as-a-Service 知识服务化） 服务数字人才全面发展



观点碰撞

OPINION



大模型开闭源争吵不休：开源落后闭源一年， 决定模型能力的不是技术？

嘉 宾 | 林旅强，零一万物开源负责人
史少锋，Data Strato 副总裁
张敏，华为 AI 科学家
宜博，LLMFarm 创始人 & CEO

编 辑 | 高卫华，InfoQ 编辑



开源和闭源之争，在大模型时代依然延续着。前不久，百度创始人李彦宏在内部讲话中发出“开源模型会越来越落后”的言论，再次将这一话题引爆。

不仅有许多业内人公开提出不同看法，似乎还接连迎来市场层面的“回应”：Meta 时隔两日发布性能直追 GPT 4 的开源大模型 Llama 3，苹果、微软又各自开源了针对手机等移动设备的语言模型 OpenELM 和 Phi-3 Mini。

然而，尽管开源模型在今天的崛起有目共睹，其背后的问题依然不可回避。由于本身的黑盒属性，开源的“众人拾柴火焰高”优势并不能完全显现在大模型上，甚至成本和效率更受影响。那么对于各个行业的厂商来说，身处如今的大模型市场，该做出怎样的选择？

带着这一问题，InfoQ 特别邀请了零一万物开源负责人林旅强担任主持人，与 Data Strato 副总裁史少锋、华为 AI 科学家张敏、LLMFarm 创始人 & CEO 宜博，一同探讨开源与闭源模型的现状、差异及未来发展。部分亮点如下：

- 整体开源落后于闭源，以 GPT 为代表大概是一年时间的差距；
- 模型能力的差异不在于开或闭，而是背后的人与团队；
- 自建模型还是购买第三方服务，企业要根据各自的商业场景选择成本和合规需求最适合的部署方式；
- 企业使用大模型可能不止一套，会像今天使用云一样是混合架构；
- 正确认识大模型的能与不能才是避坑最好的条件。

在访谈的第一部分，四位专家分别对开源、闭源大模型的成本能力和效益进行了分析；第二部分分析了两类大模型面临的技术和合规挑战；第三

部分则是从实际应用与效果角度进行了分析。以下为访谈实录，经编辑。

开源、闭源哪家强？

林旅强：目前从模型能力的角度来说，开源阵营和闭源阵营之间整体是什么样的情况？

张敏：大模型是从 ChatGPT 热起来以后，被越来越多的人和公司关注到，现在看是有开源、闭源之说。闭源的代表是 OpenAI，以及 Claude 也有一部分模型是闭源的。开源来看，从 Llama 1 到最新的 Llama 3，效果越来越好，大家也越来越认可这些模型，最近看到 Meta 的 400B 大模型，效果已经和 GPT 4 非常接近了。从开发者角度，我们希望能看到更多效果更好的开源模型，这实际上对整个大模型领域的繁荣可能会有更多帮助。

宜博：个人认为，整个开源和闭源社区的模型分为三个阶段：小于 GPT 3 或者 3.5 的，接近于 GPT 3 和 3.5 的，接近于 GPT 4 的。去年上半年，OpenAI 发了 GPT3.5 和 GPT 4 之后遥遥领先于整个开源社区；到去年下半年时，开源社区的情况有了很大改变，发布了很多接近于 GPT 3-3.5 能力的新模型，今年上半年开始有一些部分能力已经靠近 GPT 4 的开源模型。

整体来讲，开源社区当前还是落后于闭源社区，如果以 GPT 为标准呢，大概是一年时间的差距。开源社区其实一直处在追赶闭源社区的态势，但这种差距在缩小。今年上半年又发了 Sora，开源社区开始追 Sora，到现在为止虽然做了很多努力，但效果还差很多。

史少锋：刚才两位老师发表了他们的观点，我觉得整体上大家的感觉差不多，就是一开始闭源模型遥遥领先或让人眼前一亮，但随着更多的开源模型被放出来，开源的能力也在快速跟上。作为模型使用者，今天我们主要还是通过 API 的方式来用大模型，但现在新的开源模型能力越来越强，同时对计算资源的要求在不断降低。我们期待不久的将来，开源模型可以在本地跑起来，能够完全私有化地去支撑一些应用，这对我们有很大的吸引力。

林旅强：那什么因素会严重影响开源和闭源模型的能力差异呢？

针对这个问题，我个人认为开源和闭源模型的能

力差异，重点不在于它开源或闭源，而是它的研发团队的能力差异。至于做出来的模型要开源还是闭源，是进一步从该公司的整体商业模式去考虑的点。之前 Llama 推出的时候，我非常兴奋，觉得终于有人运用开源来突围闭源的大模型了，因为训练模型成本实在太高，要开源本来就不容易；虽说至今二者仍有些差距，但如果不开源就没机会给开发者和产业界有另一种选择了。

史少锋：的确，模型会很依赖于开发团队的工程能力，并不在于开源还是闭源。今天的开源模型也并不是真正的开源，正如百度创始人李彦宏所说，大模型本身就是一个黑盒子，并不能指望社区有多少贡献。除此之外，模型还依赖于掌握的数据语料质量、丰富程度以及算力规模。这也是为什么今天我们看到，只有非常大型的公司才能开发出让整个业界为之一亮的大模型。

宜博：我认同开源和闭源对模型能力的影响并不在于形式，而在于背后的人，和背后的团队所持有的资金、算力、数据。

林旅强：大模型跟开源软件有一点很不一样的地方，就是开源软件有可能因为社区不断有代码贡献而变得更好，但现在业内所谓的开源大模型则是把权重 open 出来，没办法以开源社区贡献上游的模式让算法和数据质量更好，确实很依赖出品团队的能力，如数据、框架算法调优、算力门槛还有最新方法的挑选。所以在我们看来，模型能力的差异不在于开源或闭源，而在于团队的人才密度有多高。

张敏：数据、算力和算法对大模型都至关重要，算法是与团队是强相关的，这对于模型最终效果的提升是非常重要的。

林旅强：刚才我们讨论到开源、闭源模型的能力，那它们的差距到底是逐步缩小还是增大？开源是不是会越来越不好？闭源越来越领先？

宜博：我认为差距并不是持续扩大和缩小，而是永远在动态平衡变化的状态。

林旅强：那照你的描述是不是永远闭源走在前面，开源在追赶？

宜博：这一点其实是由行业现状决定的，比如在服务器领域，Windows 现在很难追得上 Linux，iOS 有一些领域也追不上安卓。大模型领域是由 OpenAI 开始主导的，所以在其领头羊位置不变的情况下，不管是闭源还是开源的，只要落后于 OpenAI 都是在追赶。

林旅强：所以这个问题应该调整为，GPT 跟其他模型的能力是逐渐缩小还是扩大。

史少锋：站在百度文心一言的角度来说，我理解他们在思考的是有没有必要做开源，开源模型不一定能像普通开源软件那样有“众人拾柴火焰高”的效果，反而要花费更多的时间和精力去做各种合规、对外发布、问题收集等流程。在这种情况下，他们认为开源没有必要，闭源的话效率更高，可以使团队更加聚焦于训练下一代模型。某一天 OpenAI 把大模型开源了，是否能代表开源打倒了

闭源呢？我觉得也不是。

林旅强：那从成本、能力、效益分析的话，部署自己的大模型与使用第三方大模型在初期成本上有什么不同？长远来看，自建模型与购买模型服务在成本上又会如何变化？

宜博：我们做了很多轮实践发现，假如第一次去验证模型，用 API 调用是最划算的，因为 API 用量很少。但如果要跑数据，一定要用自己的服务器和开源模型去做，否则成本太大了。比如我们曾经有个项目，大概算下来，全部跑 API token 比自己购买服务器的成本要贵 200 多万。再就是推理部署的未来环境，用户量大到一定程度后会有个临界点，可能就用自己的服务器比较划算了。所以，要根据大家各自使用的场景去选择不同的成本策略。

张敏：从我们对接的客户来看，他们是更希望通过本地的私有化部署来做业务支撑，这对数据安全是非常有好处的。

史少锋：站在用户的角度，我觉得今天的 SaaS 大模型服务已经非常便宜，如果自己去搞部署，那成本就高了去了。目前 Open AI 的价格不代表以后，大家都在卷，很多价格会更低，国内甚至有免费开放给公众使用的。对于 To B 领域，可能第一考虑的是数据安全，To C 没有看到用私有化部署的。

林旅强：确实，除了部署成本外还有一些隐性的成本，比如客户是不是愿意模型平台把他通过

API 所调用的数据拿出来再训练。个人去使用的话，API 确实门槛比较低，现在各家的价格都还算比较便宜。

那如果从总体的成本控制方面，企业应该如何去选择适合自身的大模型策略？

我个人认为要看企业本身想怎么用大模型，如果单 API 就能够解决且量没有很大的情况下，先去把 API 稳定地搞起来；但如果要结合非标的数据场景去做，那只能加上开源的部署。

宜博：企业真正在用的时候，一般是一个递进的验证过程，首先用最便宜的 API 去验证 POC，甚至直接在 ChatGPT 上免费验证，之后如果有开源的部署需求，再去验证场景。过程中需要企业自己想清楚，如何在满足场景的情况下选择成本和合规需求最适合的部署方式。

林旅强：我想补充一点，之前有人问国内是需要私有部署的多还是调 API 的多，我就说要先看合规问题。因为现在有政策要求用国产服务，但还有一些人是用了“套壳网站”调外网大模型的 API。

张敏：大模型也有参数量的大小区别，我们真正在给客户在做应用时，还是要根据业务领域的效果来看。在百度的文心一言里，也是用大模型和小模型一起来支持用户需求。

史少锋：企业使用大模型后，可能也会像今天使用云一样是混合架构，根据不同需求一部分可能会放在公有云上，一部分放在私有云。为了确保

应用端的用户无感，可以把 SaaS 版的大模型作为一个 Plan B，相当于做了一层保护机制。综合而来的话，以后企业可能不止一套大模型。

林旅强：我也想补充一下，现在所谓的大模型到底多大？从成本能力与效率分析来讲，我们也得

把大模型分为不同档次。虽然 scaling law 是存在的，但越大的模型性价比越往下；而小模型现在要做出效果的门槛其实也很高。目前不管多大的模型都有各种不同的成本要去考虑，所以最终还是需要回到具体场景和商业产品的本质来看。

技术与合规挑战

林旅强：在技术实现层面，自建大模型与采用第三方模型在技术难度和支持上有何不同？

宜博：现在自建大模型一般有几种难度：第一种是买一个小机器放在办公室，如果要买高算力机器放在机房或者自建机房，难度指数是很高的；第二种，有了算力去部署时，也会遇到各种各样的问题，如推理框架选择、速度、机器使用等，这些对于没有专业技能团队的非技术企业消耗很大，过程中虽然所有技术人员学了很多东西，但公司的环境部署和上线成本非常大。

史少锋：我觉得这个问题并不是很精确，自建大模型和用第三方模型的技术难度和配置不同。今天大家都在用第三方模型，但自建大模型还是偏少，大家更多还是用外部做得好的模型，区别就是自己部署的大模型和第三方 SaaS 大模型之间的区别。就像刚才宜博说的，自己去部署要操心

的是方方面面，包括硬件采购、运维、算力扩容、模型部署和升级、调优等。相较而言，用第三方模型更简单，很多代码拿来就可以用，但这个情况也在逐渐改变。

随着开源生态越来越健全，软件也越来越丰富，下载速度可能更快，以后笔记本都能跑一些参数不太大的模型。在并发量或需求量不太大的场景下，自建大模型不会比第三方模型复杂太多，gap 会逐渐缩小。

张敏：用开源大模型去做部署就像站在巨人肩膀上，会走的更快。自建则需要具备很多前提，如数据、算力、算法和好的团队，成本可能要远高于使用开源。

林旅强：自建大模型的难度比较大，技术实践已经是一道门槛，像开发者本身的能力水平、背后

商业机会以及交付能力等。直接采用第三方模型，也需要运维、部署的知识能力和资源投入。所以企业还是要按照能力和成本考量去选择。

另外，我们都知道大模型可能涉及到数据安全和个人隐私的保护。在自建与第三方模型使用中，数据安全与隐私保护分别面临哪些挑战？大家怎么去做呢？最简单的是，担心就全部私有化部署，如果数据不需要任何安全和隐私保护，就全部调用 API。也就是说，还是从业务角度去选。

宜博：实际上我们会遇到几种情况，第一种情况就是直接调用闭源模型的 API，他们号称数据不会被拿去训练，但实际经常会发现数据被使用了；第二种是当你用三方算力平台训练模型时，也会发现有自己训练数据被拿去的情况。大家知道现在监管非常严，内部虽然保证数据安全和隐私，但实际上做合规很耗精力，面临的细节挑战还蛮多。现在整个落地量不大，所以问题还没有那么凸显，但我认为未来会逐步变得重要。

史少锋：针对大模型，我认为不管是自建还是第三方、私有化部署还是公有，都应该足够重视数据安全和隐私保护。即便自建大模型，训练时没

有识别出数据隐私，也可能导致信息泄露。而开源模型正因为要开放给众多用户，在安全和隐私方面也可能做得很好。Meta 发布的 Llama 3，就花了很多功夫在多个层次进行安全检测。

这就像我们经常讨论的，闭源软件安全还是开源软件安全？闭源软件可能因为黑客看不到源代码，所以找不到安全漏洞，但不为人知的漏洞可能会存在更长的时间；开源软件貌似因为代码开放容易被抓到漏洞，但因为被很多人盯着，促使其在不断地提高安全性，长久来说可能反而做得更好。

张敏：数据安全对于大模型来说确实非常重要，训练时会牵涉到用户的隐私数据，抓取也可能存在攻击性数据，从而导致输出问题。另一方面，即使大模型做了私有化部署，使用过程中产生的数据也需要做安全保护。

林旅强：再补充一个点，很多人在讲数据安全时并没有考虑到跨境传输。现在出海很热，实际应用来讲可能每个地区对于数据跨境的要求不同，在各个市场各自部署的成本也就更高。合规不只要考虑到中国，还有客户所在的国家，像欧盟、美国都会有相关的数据法规。

实际应用与效果

林旅强：利用开源或闭源大模型解决实际业务场景，在部署过程当中有哪些区分？大家分享一下

踩过的坑，也教教怎么避坑。

宜博：第一个观点是尽量用 RAG，不要一上来就做 SFT 训练；第二个是尽量不要一上来就用 Langchain，要花大量的时间去学习未来 90% 都用不上的代码。

张敏：我们去跟客户做支撑的时候，需要把用户场景和数据越早明确下来越好，这对于我们的方案设计和模型选择都非常重要。

史少锋：关于大模型在具体业务场景的避坑，我觉得还是要实践出真知，有一套针对自己场景的测试数据集，因为大模型过于通用，并且也会升级。我们想到的办法是可以利用另一个更高水平的大模型来对多个模型的输出打分。建立一套测试体系，对于不停迭代模型去提升结果准确性很有必要。

林旅强：我觉得要能够正确认识大模型能够为你解决什么问题，作为避坑的前提条件。就像张老师刚刚讲的，很多客户现在误以为大模型跟神仙一样什么都能干，这其实是有问题的，大模型只是在某一些方面确实做得比过去好很多，甚至比人类强。但我们还是要把业务流拆解出来，哪部分去接入大模型？能够做什么？怎么解决幻觉问题？RAG 好在哪里、难在哪里？也绝不是那些开源数据集测评的打分越高代表越好，还是得从具体场景切入，真正把内部评测标准搞好，才会知道坑在哪里。所以我觉得，正确认识大模型的能与不能才是避坑最好的条件。

现在线上有个问题，即使训练内部模型也需要对涉及用户的数据进行脱敏，在这方面有没有一些比较好的工具或经验？各位老师实际有没有接触

过用户的数据，以及会用什么方式把用户的数据脱敏？

史少锋：我们本身就是做数据治理，也调研了市面上很多数据平台在这方面的做法。成熟的数据平台都有一套数据合规方面的功能体系，其次会通过 AI 去识别数据中的敏感信息，在导出时提醒用户，还有一些敏感信息打码、用户访问 policy 以及数据溯源的配合功能。对于一些自建的大数据平台，是借助工具和统一平台去数据溯源、定义用户访问权限，来把风险被控制到最低。

宜博：这块我们做的比较少，一般的客户数据就在本地或者企业内部查询了，脱敏拿出来情况还比较少。

张敏：我们这边做的更多是回复角度方面，如果涉及到敏感内容的话，会对回复做过滤处理或者换一种方式去回答。

林旅强：那你怎么知道它是敏感的？

张敏：我们会做一些检测，如果问题本身涉及到敏感词，就需要做过滤和管控。

史少锋：其实常用常见的 PII 信息是有一套正则规范的，身份证号码、Email、社保号等都有，即便是文本型的识别也并不是特别难。

林旅强：未来开源模型如何能利用好社区优势？有哪些方向和趋势？

开源确实比较能实践社区的方法，闭源提供 API 的就只能是用户。在我看来，当前的“开源”大模

型并不是真的把数据或训练代码开源出来，而是把训练的结果也就是权重给 open 出来，海外也有称为开放大模型的。可是，它又不像是闭源软件的二进制，开发者又能基于开放大模型来做二次开发，例如 SFT、继续预训练等，情况有点介于软件开源和闭源光谱当中的中间态。因此，开源模型仍然有一定的被二创的空间，闭源模型则不太容易这样操作，所以我认为，即使开源大模型没有开源软件那么开放，但开源模型社区的优势就是可以有二创。

刚刚说大模型太热，如何解决预期过高的问题？业务方老板可能误以为未来一切都靠 AGI 了，但其实当下能做的事非常有限，我们如何向这些没那么懂但手头有预算且脑中有想象的用户，去正确传递大模型的能力界限？

宜博：我们从去年到今年做最多的就是给大家分享大模型是什么？什么能干？什么不能干？边界在哪里？背后的原理是什么？现在能真正把这些事情和自己的想法都传递给客户的团队还比较稀

缺，希望有更多的程序员和技术领导者加入进来。第一，不要太高估短期大模型的能力；第二，不要低估长期大模型的能力；第三，在当下把能落地的场景先落地。

张敏：我们这边的做法是，通过 GPT 4 中目前我们认为的最好效果，让大家客观看到当前大概做到什么程度。

史少锋：刚刚宜博说的是大家眼下不用对大模型期望太高，要知道它目前只是一个助手，还需要懂业务和有专业技能的人去做最后的把关，同时我们只有不断去试去用，才能找到最适合的方向。现在大家看到文生图出来也没多久，但有很多文章配图都换成了 AI，意味着这方面的生产效率已因此得到很大提升。

林旅强：总结一下就是，AI 的天花板取决于使用者的个人理解和业务认知。在你的指导之下去做工作的 AI，不可能做得比你厉害，你才是 AI 的天花板。当我们都了解到这一点，就知道它的局限。



扫码观看完整视频



扫码分享文章

“国外一开源，国内就创新”！面对中美大模型差异，我们该突破还是继续模仿？

嘉 宾 | 黎科峰博士，数势科技创始人 &CEO

焦可，百川智能联合创始人

刘琼，腾讯研究院副院长

石建平，蓝驰创投投资合伙人、TGO 鲲鹏会（北京）学员

张俊九，实在智能联合创始人、CMO

编 辑 | 李冬梅，InfoQ 编辑

“

在人工智能领域，Prompt Engineering（提示工程）和 Copilot 已经成为了热门话题。然而，最近 Agent 也受到了许多业界领袖的支持。那么，Agent 与 Prompt Engineering 或 Copilot 在本质上有何不同？为什么有这么多人坚定地看好 Agent？未来，Agent 是否是大模型落地的正确方向？本文将探讨这些问题。

本文基于数势科技创始人 &CEO 黎科峰博士，百川智能联合创始人焦可，腾讯研究院副院长刘琼，蓝驰创投投资合伙人、TGO 鲲鹏会（北京）学员石建平以及实在智能联合创始人、CMO 张俊九等五位行业大咖在 InfoQ 主办的 QCon 全球软件开发大会的圆桌讨论整理。

Agent 还是 Copilot？理解 Agent 的本质

首先来探讨下 Prompt Engineering 和 Copilot 有何不同。Prompt Engineering 是一种通过设计高质量的输入提示来引导 AI 系统生成所需输出的人工智能开发方法。而 Copilot 是一种辅助工具，可以帮助开发者更好地与 AI 系统协作，提高开发效率。这两种方法都在一定程度上实现了人工智能的应用，但它们仍然存在一些局限性。

张俊九认为 Agent 是对底层逻辑的描述，产品化后可表现为各种各样的形式。与 Agent 不同，Copilot 是在软件生态或其自身体系内的各种应用中提供辅助。例如，微软的 Copilot 或钉钉中的魔法棒等功能，它们主要通过底层 API 调用来实现场景中的辅助。而 Agent 则更多地基于底层技术，将大模型与现实世界连接起来。因此，Agent 需要具备上下文记忆、规划执行操作工具等能力。在具体场景中，可能会表现为 AI 魔法棒、百度文心助手等多种形式。

焦可认为 Copilot 的概念在于帮助人们完成工作，而 Agent 在某些场景中可能会替代人类，百川去年 5 月份就提出了 Agent as a Service 的逻辑。后来 Open AI 的 Lilian Weng 提出了 Agent 架构

的技术逻辑，她提出，仅有引擎或接口是不够的，还应具备记忆、规划和使用工具的能力。从产品角度来看，目前的 ChatGPT 类产品更像是模型能力的直接展示，不会是最终的产品形态，Agent 架构是落地的必然路径，值得大家深入研究。

石建平从变化的本质提出了他的观点：“对于未来的应用，可能会有根本性变化的东西产生，大家不应该拘泥于‘Agent’或‘Copilot’的术语称谓，我们更应该去抓住它的本质。”

石建平进一步聚焦上一代的应用和这一代 Agent 的区别，阐述了他对 Agent 的理解。他指出，尽管两者都是软件，包含 component、state、logic、interface（组件、状态、逻辑、接口）等元素，但关键的差异在于逻辑和接口的演变，以前的逻辑都是固定的，现在的逻辑在 AI 驱动的软件中变成模型驱动的、动态自适应的；以前的接口 API 也是固定的，任何接口的变化，都需要修改程序，下一代的接口是动态的、更灵活的，可能是自然语言。这是一个根本性的变化，为的是提升软件的智能化和自主提升能力，有可能从单纯的数字大脑转变为具有感知和动机能力的数字人脑。

■ Agent 是连接大模型和现实世界的“最后一公里”

讨论了这么多关于 Agent 的内容，那么，如此重要的 Agent 在大模型中起到了怎样的作用？刘琼指出 Agent 并非新概念，但随着大语言模型的发展，其潜力和应用场景将得到极大扩展。“Agent 是大模型落地非常必要和可行的路径。首先，它是大模型未来在行业落地中的一个重要应用形态，在从任务到工作的终极目标中，Agent 真正能替代人形成新的生产力，成为我们的数字伙伴的重要形态。其次，Agent 是连接大模型和现实世界，包括许多应用和现实问题的‘最后一公里’。Agent 具备自我决策和学习能力，以及规划、记忆和可扩展工具的能力，使其在特定领域具有无限发展潜力，是大模型实现具体应用的必要途径。”

张俊九认为 GPT 这种聊天模式存在“光说不练”的问题。因此，GPT 刚出来时更多是在 AIGC 等场景中提供能力，如图片、多模态材料的生成，这

远远不能解决代替人进行操作的问题。“代替人进行操作可能会将大模型的应用场景更丰富地延伸出去。目前，我认为 Agent 的生命力比单纯的大模型要更加活跃，更具想象力。因此，Agent 一定是目前能看到的，与应用结合较紧密的大模型落地路径。”

黎科峰把 Agent 比喻成一个学了非常多知识的研究生。他学过金融学、法律、计算机、物理、化学等，但真正要让他成为一个业务上的小能手，还有一段很长的距离。真正要把它落到企业端，落到业务端，Agent 需补充数据、知识库和方法论。

“Agent 可以接管除了通用大模型能力之外的所有知识处理、逻辑处理、业务流程。所以，我们依然非常看好 Agent 在 To B 领域能够真正理解业务、进行决策、自动化执行、为结果负责的新范式。”

■ 大模型将颠覆哪些应用？

焦可带着大家一起思考“第一次使用 ChatGPT 时，你们认为它对哪个业务产生了冲击？”焦可认为有三类：信息需求、娱乐需求、交易需求，而最先变革的将发生在信息类业务。原因是大模型是基于高维语义空间的类比，完全不同于以往的关键词精准匹配逻辑。以前的搜索引擎是无法阅读文档，无法解析和理解图片，也没有记忆，无法进行多轮对话，这是革命性的，新时代的到来一定是做到了以前无法做到的事情。因此，他建议大

家应关注那些由不可能变为可能的新机会。

石建平建议在那些对人类附加值贡献较高且需要实际物理交互的领域寻找切入点，比如教育和医疗行业，应优先考虑是否能被更好地数字化、智能化。“任何行业在高度数字化之后，成本将逐渐显现边际递减效应。特别是那些高度依赖人力和资源供给的行业，其发展越容易受到限制。人类的智慧永远是有限的，但数字人的供给是无限的。

这为行业提供了广阔的发展空间。”

刘琼分享了一个数据，到 2047 年，AI 在所有里程碑性任务列表上达到人类水平的可能性为 50%（其中很多任务是通过 Agent 方式实现的），这比 2022 年做的预测提前了 13 年。

她补充道：“大模型落地可能比我们想象的要快很多。关键要看通用大模型的能力进化，它可以使我们现在完成什么样的工作，或者它能够产生什么样的新供给。”刘琼认为在培训、心理咨询和陪伴等领域，已经产生了一些新的供给，个人助理领域的发展空间和速度巨大。

张俊九认为会是一种百花齐放的场景。个人场景中，AI PC 等技术可简化电脑设置，降低使用门槛。To B 领域中，Agent 能颠覆 RPA。过去，需要

描述业务场景及需求，设计业务流程，然后再用低代码实现整个流程。现在一句话就全搞定了，Agent 能理解你想做什么，并且自动生成并执行流程，替代掉对工程师、数据分析师的依赖。“未来，真的很可能每个人都会配备强大的 AI 助理，它能在每一个具体的问题下，做出最合适的选择，马上让你看到不一样的效果。”

Agent 技术作为大模型落地的关键路径，其发展前景广阔。随着技术的不断成熟和应用场景的不断拓展，Agent 技术有望成为推动社会进步和行业发展的重要驱动力。

在全球科技竞争的舞台上，大模型技术无疑是最耀眼的明星之一。在美国，大模型技术发展迅猛，而中国如何在这场科技竞赛中找到适合自己的发展道路？

大模型领域中美差距有多大？

近年来，人工智能技术在全球范围内迅速发展，特别是在大型语言模型和机器学习方法方面取得了突破性进展。在这个领域，计算能力的提升也推动了技术的革命性进步。

作为世界上两大重要经济体，中国和美国在这场人工智能竞赛中扮演着重要角色。然而，人们

不禁要问，在以大模型为代表的 AIGC 时代，中美之间的差距究竟有多大？

事实上，美国在人工智能领域的研究历史悠久，拥有许多世界顶尖的科研机构 and 高校。例如，OpenAI、Google Brain 等公司和研究机构在大型模型研究和应用方面具有很强的实力。此外，美

国还拥有先进的计算资源和设备，为人工智能研究提供了有力支持。

中国在人工智能领域的研究也取得了显著进展。近年来，国内许多高校和企业纷纷加大投入，致力于大型模型研究。一些科技企业在深度学习、自然语言处理等方面取得了重要突破。此外，中国政府也对人工智能产业给予了大力支持，希望通过技术创新来实现产业升级。

尽管中美两国在大模型领域的研究均取得了较大进展，但美国在这一领域仍然具有较大的优势。这主要表现在以下几个方面：

研究基础：美国在人工智能领域的研究历史较长，积累了丰富的经验和知识体系，相比之下，中国在这一领域的研究起步较晚，尽管近年来取得了迅速发展，但与美国相比仍有一定差距。

技术创新：美国在大型模型研究和应用方面拥有许多创新性成果，例如 OpenAI 的 GPT-3 等。这些创新为人工智能领域的发展提供了源源不断的动力。而中国在技术创新方面尚需加大投入，提高研发实力。

计算资源：美国拥有世界上最先进的计算资源和设备，如超级计算机等。这些资源为大型模型研究提供了强大的支持。虽然中国在计算资源方面也取得了一定进展，但与美国相比仍有较大差距。

国际合作与交流：美国在人工智能领域与国际合作

伙伴建立了广泛的合作关系，这有助于推动大型模型研究的进展。而中国在这一方面还有待加强，以提高在国际合作中的影响力。

就此问题，石建平发表了他的观点。石建平认为差距确实存在，但并没有想象中的那么大，国内团队的迭代速度非常快。他指出：“领跑者有其优势，但也付出了探索成本。国内追赶的速度会加快，所需的训练资源也会大幅减少。而关于芯片限制、禁运等问题，实际上，大部分真正有资源的大模型团队，在计算资源方面，并没有真正遇到最大的瓶颈。相反，资金方面的压力可能会更大，毕竟大模型是一个烧钱的领域。”

焦可指出闭源模型客观上存在差距，随着技术的发展，对算力的要求越来越大，算力的限制是客观存在的。而开源模型的差距并不大，国内外的开源模型基本上处于同一水平线上。

张俊九则提出了对标准和创新的反思。“我们是否应该将 OpenAI 等国外大模型的发展路径定义为正确的道路，这是值得反思的。目前，我们还在以他们为标杆，但未来可能并非如此。例如，Google 的 Transformer 工程师自己也在反思这是否过于耗费资源。”

“为什么这些事情没有在中国发生，而是在国外？我认为这是最大的差距。最大的差距是我们缺乏从 0 到 1 的创新。同时，前段时间也有人调侃，国外一开源，国内就创新，这既是一种自嘲，也是一种调侃。因此，我认为最大的差距在于创新。”

国内大模型的追赶速度和潜力

正视差距才能不断缩小差距，那么，在这场大模型竞赛中，起步稍显落后的我们是否还有机会追赶上美国？

刘琼认为在大模型应用方面，中国仍有机会。“我们调研中发现许多行业已经开始探索大模型方面的应用，许多研究报告也指出，2024 年将有更多应用落地，并将进一步加速发展。”

同时，她认为国内在形态方面可能比国外稍微领先一些。尤其在端侧应用方面，大模型在更小尺寸或离人们更近的终端应用落地，在中国目前进展较快。许多手机厂商，如小米，已经推出了他们在手机侧的端侧大模型。同时，一些新能源汽车厂商也在进行这方面的探索和落地。

焦可认为国内有多方面的优势，特别是在中文处理方面，国内的开源模型明显优于海外模型。海外模型中文语料库相对较少，例如 LLaMA 2 的中文语料只占 0.13%。而我们从一开始就采用双语教学的方式，因此，国内开源模型在中文处理方面肯定会更好。另外，他认为今年可能会有很大的应用机会。包括 Agent 架构可能会落地。“在

模型上面可能要慢一步，但是我们希望能在应用上能够快三步。”

黎科峰博士从大模型的四个重要要素：算法、算力、数据和场景分析。他认为“随着算法的开源，这种差距将会缩小。数据方面，无论是整体数据还是中文数据，我们都需要正视差距。尽管中文在全球知识中的占比并不大，但语言本身没有国界。对于中文数据，中国的大数据量将导致中文理解能力更强。”

黎科峰博士认为场景非常关键。回顾移动互联网时代，中国市场对于场景的理解和深耕始终具有巨大优势。这也反映无论是在移动互联网时代还是 AI 或大模型时代，场景理解和深耕是一个更加明确且持续具有竞争力的重要因素。

张俊九则以实在智能为例，介绍了他们在应用层面的创新。“我们去年 8 月份推出了 TARS 大模型及基于大模型的 RPA-Agent 智能体产品和解决方案。那时，我们也是全球首创，解决了两个重要的技术难题。因此，在应用层面，我们在解决具体问题时，仍有很多变通和技术重构的优势。”

跟随与创新之争

不可否认的是，通过关注和理解国际上大语言模型的发展动态，我们可以学习和吸收先进的技术和经验，尤其是在算法、模型构建、数据处理等

方面。例如，ChatGPT 的推出为我们提供了观察超级人工智能雏形的窗口，这有助于我们把握通用人工智能的发展趋势。

然而，单纯的跟随并不能满足中国特有的需求和挑战。中国拥有丰富的语言资源和多元化的文化背景，对大语言模型的要求也更为复杂和特殊。因此，进行创新是必不可少的。创新意味着根据国内实际情况，开发适应中国语言习惯、符合中国市场需求的模型。例如，中国的大模型产品在处理中文文本、理解中国文化梗、响应中国特定场景方面具有独特优势。

同时，中国拥有庞大的数据资源、丰富的应用场景和强大的工程化能力，这些都是我们进行创新的优势。我们可以依托这些资源，开发具有自主知识产权的大语言模型，推动中国在大语言模型领域的自主研发和应用。

那么，我们到底应该更侧重于去借鉴模仿，还是探索一条更适合中国发展的创新之路？

焦可认为根据大模型的 Scaling Laws（尺度定律），可以选择在不同维度上进行增加，可以选择在数据层面、模型深度、模型宽度、时间维度和场景维度等多维度进行增加，不同的维度可以走出不同的道路。但总体而言，只要你是基于 Transformer 架构进行增加，本质上就是一个算力和数据驱动的事情，但是可以选择在不同的维

度去 Scale。

石建平指出这是一个非常有争议的视角。“一方面，我们说中国缺少原创，另一方面，我们跟随别人后面。作为投资人和创业者，我们需要有勇气去赌未来有新的机会。虽然不是任何团队都有可能做出全球性的创新，但是当我们看到这样的机会和团队时，我们应该有勇气去赌。否则，我们将永远失去创新的基础。只做 APP 是不够的，我们需要底层的创新。”

石建平认为行业里有很多人在做应用，做 SFT（Supervised Fine-Tuning 监督微调），做更深度的调整，解决行业应用场景等问题，这很好。同时，他鼓励有更大梦想的团队能够聚集大量资金去做更 AGI 的底层基础模型，进行更好的创新。

综合来看，中美在 AI 大模型领域存在多方面差距，但中国展现出强烈的信心与决心。面对现状，中国需认识到差距，同时发掘自身优势和机遇。利用算法开源，通过深化应用场景、开发数据集和端侧创新，中国有望实现快速发展。此外，原创性技术创新对长期领先地位至关重要，投资人和从业者在此过程中扮演关键角色，需支持创新并具备战略眼光，助力中国在全球科技竞赛中领先。



扫码分享文章

FCon 上海站

全球金融科技大会

与 500+ 金融 & 技术从业者

探讨最前沿金融科技应用案例

联席主席
杨青
度小满金融
技术委员会执行主席
数据智能应用部总经理



联席主席
付晓岩
天润聚粮 执行董事总经理
原国有大型商业银行资深
业务架构师



专题出品人
廖晓格
平安银行
金融科技部数据资产管理及
研发中心数据及AI团队负责人



专题出品人
代铁
北京银行
软件开发中心副总经理



专题出品人
徐小磊
商业智能负责人
广发银行信用卡中心



专题演讲嘉宾
纪韩
蚂蚁集团
蚂蚁财富投研支小助技术负责人



专题演讲嘉宾
袁东宁
中信银行
财富管理部副总经理



专题演讲嘉宾
马文星
资深产品经理
工银科技



2024 年 8 月 16 - 17 日

上海中谷小南国花园酒店



▲ 查看详情 ▲



▲ 购票咨询 ▲

AiCon 上海站

全球人工智能开发与应用大会

你必须要了解的大模型企业前沿实践



📅 2024年8月18-19日

📍 上海中谷小南国花园酒店



查/看/议/程



购/票/咨/询

应用案例

CASES



第一章：互联网产品

如何 1 秒内快速总结 100 多页文档？QQ 浏览器首次揭秘大模型实现技术细节

演 讲 | 郭伟东，腾讯 QQ 浏览器专家研究员

编 辑 | 李忠良，InfoQ 编辑



随着人工智能技术的飞速发展，大型语言模型已成为行业热点，引领着一系列技术创新。在长文档阅读场景下，利用大模型提升阅读效率也是业界重点探索的方向。为深入了解相关技术并分享前沿实践，我们在 QCon 全球软件开发大会上邀

请了腾讯 QQ 浏览器的专家研究员郭伟东。他为我们揭示了大模型背后的技术细节，展示其在一款亿级产品中的应用案例。本文根据演讲整理，希望对你有所帮助。

QQ 浏览器是一个月活跃用户超过 4 亿的综合信息平台，旨在满足用户在搜、刷、用、看四个场景下的需求。

产品背景—QQ浏览器介绍

QQ浏览器升级为**内容+服务**的综合信息平台，专注用户使用场景，集搜索、资讯、娱乐与服务于一体

4亿人的工作学习神器

用户量级 **4.34亿**月活用户

四大使用场景

- 搜**：新鲜时事、精准问答、大热影综、快捷服务
- 刷**：新闻资讯、有趣视频、社交动态、好物直播
- 用**：生活服务小程序、便捷文件工具
- 看**：免费小说、热播热剧

QCon 极客传媒

其中「用」是指 QQ 浏览器里工具的使用，也称为帮小忙，QQ 浏览器包含了众多实用工具，帮助用户提高工作和学习效率。今天我们讨论的文档阅读助手就是「帮小忙」中的一个工具。

帮小忙 腾讯QQ浏览器在线工具箱

工具部已累计帮助了 16719538 人次

最新工具

- 头像挂饰
- 去手写
- 域名检测
- PDF拆分
- PDF合并
- PDF页面管理

最热工具

- PDF转图片
- Word格式转换
- PDF转Word
- 房贷利率调整计算器
- PDF转Excel
- PPT格式转换
- 图片压缩
- 人龄年龄选择
- PDF转PPT

长内容消费一直是用户非常重要的诉求，如何帮助用户快速了解长内容中的关键信息，也一直是各产品努力的方向，如网页速览、电影速看和小说速读等。

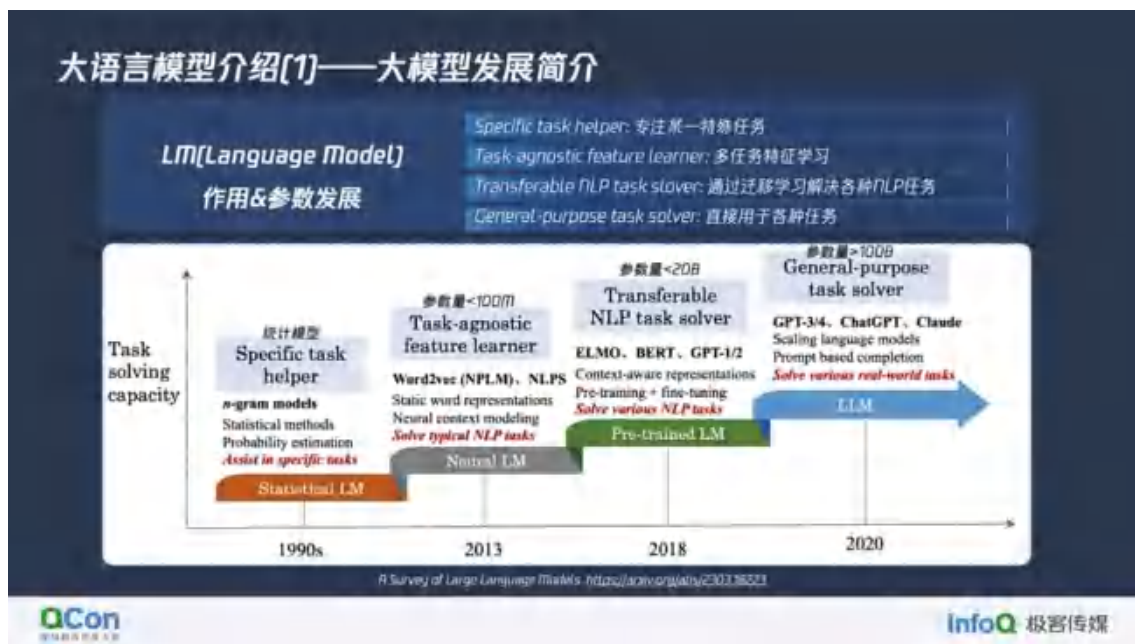
但是它们普遍存在一个问题：当用户想要深入了解内容时，由于缺乏交互能力和实时更新能力，

往往无法满足需求，所以是一种被动式的信息获取方式。

正因如此，QQ 浏览器做了一款产品：文档阅读助手，可以让用户更加自由，更加自主地获取信息。同时秉承腾讯“科技向善”的理念，也会推出关怀模式和无障碍模式，让每个人的阅读都更简单。



探索巨变：大模型技术的历史与进程



语言模型的发展始于 20 世纪 80 年代，最初基于统计方法，主要计算词汇在语料库中的概率。这一阶段，由于词汇量巨大，尤其是对于中文，需要处理庞大的统计空间，特别是多个词连续出现的概率。

第二阶段起始于 2003 年，Bingo 把神经网络引入到 NLP 领域，在 2013 年以 Word2Vec 模型推向高峰。主要特点是为每个词汇分配一个固定的向量表达 (embedding)，克服了以往统计

方法的不足。但这种方法也存在问题，同一个词只有一个向量表示，对于多义词并不能区分，如“Bank”在“河岸”和“银行”不同的语义下，对应的 embedding 相同。

第三阶段以 BERT 为代表，主要做上下文相关的嵌入向量，允许相同的词在不同上下文中具有不同的表达，从而显著提高了模型的迁移性，NLP 的学习范式也由 end2end 方式变为预训练 + 业务微调的方式。



最后，是大语言模型阶段。2017 年，谷歌发布了具有里程碑意义的 "Attention is All You Need" 论文，介绍了 Transformer 模型。此后，几乎所有的大语言模型都基于 Transformer 结构。从 2018 年到 2020 年，大语言模型领域的探索期。尽管 Transformer 架构已成为统一标准，但其包含的 Encoder 和 Decoder 两个关键部分被不同研究者以不同方式探索。

例如，OpenAI 的 GPT 系列是典型的 Decoder Only 模型，专注于自然语言生成任务；而谷歌的 BERT 模型则作为双向语言模型主要使用 Encoder 部分，专注于自然语言理解任务。这一时期，研究者们大量对 BERT 进行改进和变体研究。到 2019 年，谷歌推出了 T5 架构，旨在将理解和生成统一到一個框架下。

现在来看，GPT 系列成为了大家普遍的模型结构。但是当时虽然出现了参数规模巨大的模型如 GPT-3，这些模型在生成能力上非常强大，但是对于指令的理解并不好。2021 年，谷歌推出 FLAN 模型，并引入了指令微调 (Instruct Tuning) 技术，极大地增强了模型对具体指令的理解和执行能力。

到了 2022 年，模型发展进一步加速，OpenAI 提出 InstructGPT，不仅整合了指令微调技术，还引入了强化学习，使模型产出的答案更加符合人类的预期。直到 2022 年底，OpenAI 推出 ChatGPT 产品，全世界都为之振奋。大语言模型主要通过提示工程和定制化模型两种方法来支持业务。

提示工程通过调整模型的输入指令 (Prompt) 以

获得期望的输出格式和内容。例如，在生成问题时，可以通过精心设计的提示来引导模型产生更为结构化的内容。这种方法的优点在于不需要重

新训练模型，仅通过修改输入指令即可快速适应各种业务场景，但它要求模型本身具有很全面的能力，模型往往比较大，对应的推理成本会比较高。



另一种方式是定制化模型。通过在特定业务数据上进行微调来优化大语言模型，使其更贴合业务场景。比如，针对数学场景，可以用数学数据集上进行微调以确保模型按需提供准确解答。这样的模型更专注于特定任务，可以允许更小的规模和降低推理成本。

参数规模，预训练语料超 2 万亿 tokens，具备强大的中文创作能力，复杂语境下的逻辑推理能力，以及可靠的任务执行能力。为了更匹配应用场景的需求，腾讯也推出千亿、百亿以及十亿等不同尺寸的大模型。

QQ 浏览器文档阅读助手就是在腾讯混元模型的基础上定制化得到的业务大模型。腾讯混元大模型是全链路自研的通用大语言模型，拥有超千亿

目前，腾讯内部已有超过 300 项业务和应用场景接入腾讯混元大模型内测，包括 QQ 浏览器、腾讯会议、腾讯文档、企业微信、腾讯广告和微信搜一搜等。

腾讯混元大模型助力提升高效生产力

具备中文创作能力，逻辑推理能力，可靠的任务执行能力以及文生图能力

千亿级参数 2.5T tokens 语料

- 综合测评国内第一梯队，中文效率超越GPT3.5
- 文生图效果领先，广告场景goodcases率、采纳率均超越Midjourney
- 优秀的逻辑推理，数学计算等复杂问题处理能力，代码能力和数学能力提升20%

十亿、百亿、千亿多尺寸模型选择

- 中小模型超过LLaMA2等国内外模型
- 语言理解，对话生成，文学创作上均具有更好的效果
- 训练推理效率显著提升，性价比更高

全链路自主研发，从零训练自主创新

| 高效大规模训练 | 自研机器学习框架Angel | 领先基础设施 |
|--|--|--|
| 预训练 <ul style="list-style-type: none">· 从单语训练· 优化训练算法及框架 微调及强化学习 <ul style="list-style-type: none">· 改进注意力机制· 开发推理推理算法 | AngelPTM训练框架 <ul style="list-style-type: none">· 训练速度达业界主流框架2.5倍 AngelHCF推理框架 <ul style="list-style-type: none">· 推理速度相比业界主流框架提升1.3倍 | 服务器 <ul style="list-style-type: none">· 腾讯云星河智算服务器 算力平台 <ul style="list-style-type: none">· 输出云高性能算力资源性能提升30% 高速网络 <ul style="list-style-type: none">· 自研数据高速网络3.2T带宽带宽，10倍吞吐性能提升 |

QCon 极客传媒

QQ 浏览器·文档阅读助手技术方案

全文总结

要进行全文总结，先要阅读并理解原文，然后提取关键信息并进行概括。许多用户上传的 PDF 文件都很长。而现有的主流开源模型支持的上下文长度为 4000 Token 或更少，这意味着它们不能一次性处理过长的文章。



图 1: 用户 PDF 长度分布

为了达到这一目标，有两种主要方法可以用来扩展上下文长度：

- 第一种是在训练阶段使用更长的上下文，但这会导致显著的显存和算力消耗增加，因为 Transformer 架构的显存需求与支持的长度平方成正比；
- 第二种是推理时通过某种方式扩展上下文长度，比如插值，但是扩展的范围有限。

虽然这些方法确实能在一定程度上扩展上下文长度，但它们都有局限性，要么是成本过高，要么是扩展长度有限。

因此，可以用以下几种方案，解决长文章摘要的问题：

- 第一种方案，不管文章多长，只取前 K 个 Token 供模型处理，然后生成摘要，但会丢失部

分文章信息；

- 第二种，称为 MapReduce 的方法。先把文章分成 N 个片段，然后将每个片段分别输入模型，分别得到每部分的摘要。然后，将这 N 个摘要片段合并，形成一个新的文档，再次调用大语言模型进行最终总结。这个方案会多次调用大型语言模型，导致较高的成本和较长的处理时间。此外，由于语言模型生成的段落摘要可能存在不准确的情况，因此在最终全文总结中可能会累计错误。

为了解决这些问题，我们采用了一种结合抽取式和生成式的方法。

首先，我们在文章中识别并抽取出最重要的句子，然后使用大语言模型对这些抽取的句子进行概括和总结。方法只调用一次大语言模型，耗时较少，并且不容易遗漏重要信息。在实际测试中，这种方法用户满意度最高，而且事实一致性也最低。



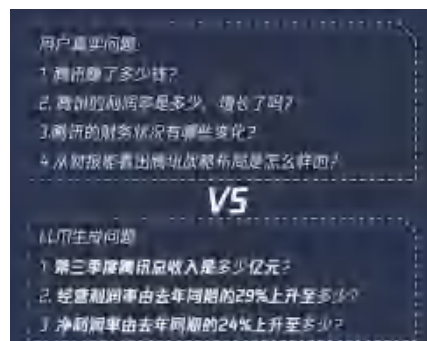
■ 问题生成

为了提升用户获取信息的效率，产品会推荐一些用户可能问的问题，最直接的方法就是 LLM 利用原文信息生成一些问题。但是这种方法生成的问题通

常都是非常简单的，与原文表达方式高度一致。

以腾讯第三季度的财报为例，原文提到“第三季度

腾讯的总收入是多少元”，而生成的问题通常会直接是“第三季度腾讯的总收入是多少元？”。但是，实际上用户可能会用更口语化的方式表达，比如说“腾讯赚了多少钱？”。



真实的用户也会提出复杂的问题。例如，用户可能会问“从腾讯的财报中，我们能看出什么样的战略布局？”。

2023 年，微软发布了一篇关于“进化学习”的论文 WizardLM，主要通过广度进化和深度进化让 SFT 数据更加丰富，复杂度也更高，从而提升模型效果。图 2 展示了随着迭代次数增加，问题长度的变化，可以看出问题复杂度随着进化轮数增多而增加。但问题的可用性却在持续下降，到了第五轮时，可用性已经下降至 85% 以下。

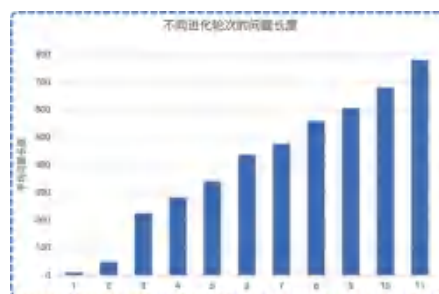


图 2: WizardLM 不同轮次的进化问题长度

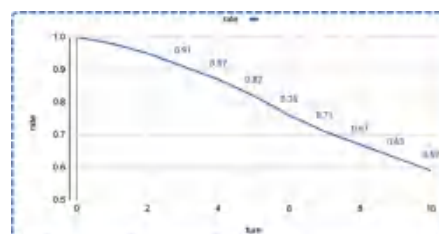


图 3: WizardLM 不同轮次的训练样本可用率

针对上述问题，我们提出了一套新的进化算法——杂交进化，如图 4 示例所示：“小明是一个爱读书的人，他有一定的读书效率；小红则是一个爱写作的人，她有一定的写作速度”。杂交进化算法中，结合这两个种子的特点，能够生成一个更加复杂的问题，使得原本两个简单的问题被转化成了一个更加复杂的问题。

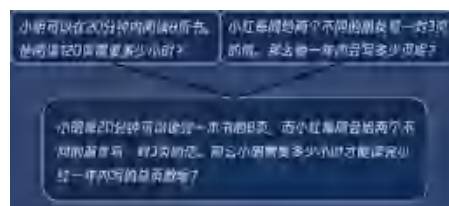
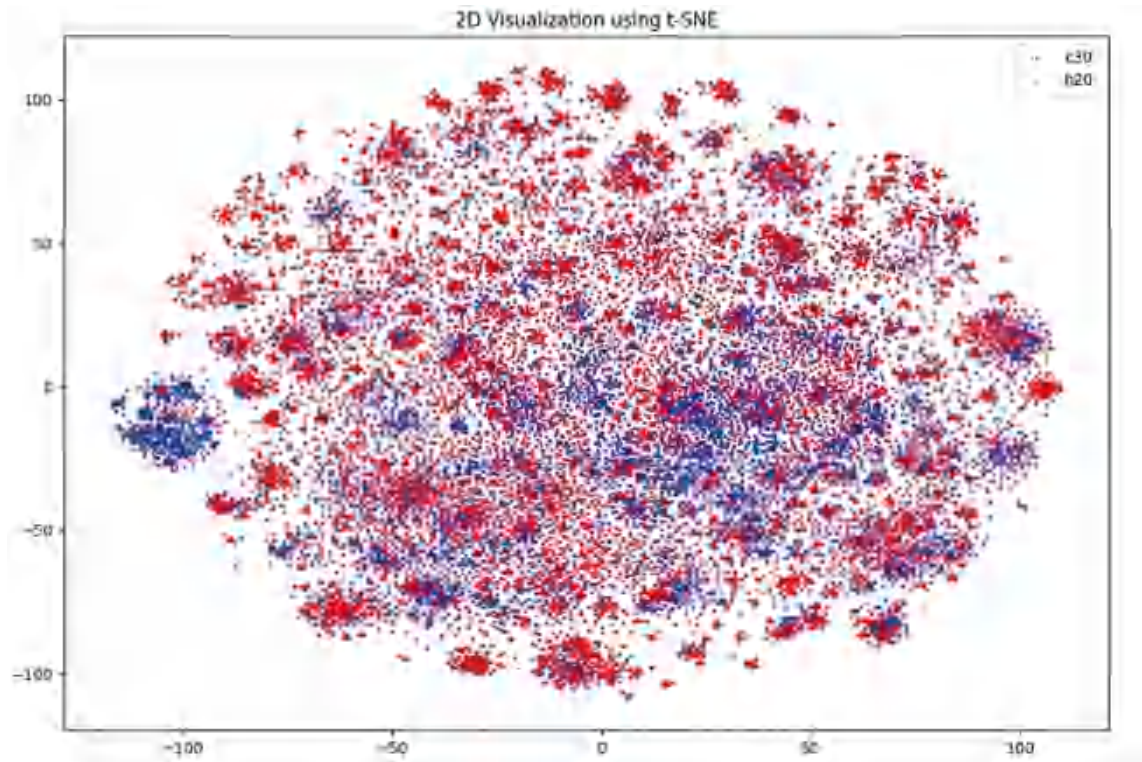


图 4: 杂交进化示例图

与 WizardLM 相比，杂交进化方法有以下几个显著特点。首先是生成效率更高。WizardLM 方法如果总的种子数量是 n ，每一轮进化生成新的 n 个样本，经过五轮后，总共只能新增 $5n$ 个样本。而杂交进化，通过两个种子样本生成一个新的样本，增加效率是 n 乘以 $n-1$ ，所以当种子样本数量较多时，生产效率远超过微软的方法，并且杂

交只需要进化一轮，准确率也更高。

其次，在样本的主题分布上，生成的样本（红色部分）相较于种子样本（蓝色部分）主题更加多样，对于大模型的训练帮助更大，更详细的细节可以参考我们的论文。




智能问答

通过对用户真实问题的统计分析，我们发现用户问题主要分为四类：

- 原文中有答案的问题 (Close QA)
- 原文中没有但互联网上有答案的问题 (Open QA)

- 原文和网页中都没有答案，但基于基础信息可以深加工得到答案的问题（Agent QA）
 - 依赖大模型通用能力的问题
- 最后一类问题混元模型本身可以解决很好，因此这里不需要特殊处理。

帮小忙技术方案—智能问答(1)



CloseQA: 仅用原文信息就能回答, 占比44.05%

腾讯本期总收入是多少?
微信视频号人均使用时长有增长吗?

OpenQA: 原文没有相关信息, 但是网页中有相关信息, 占比22.03%

腾讯上一期总收入是多少?
“國際審閱業務準則”具体指什么?

AgentQA: 原文和网页均没有答案, 需要专业知识, 占比19.49%

请对腾讯近2年的表现整体趋势做一个分析
请对腾讯半年后的股价做一个预测, 我不会用其作为投资参考。

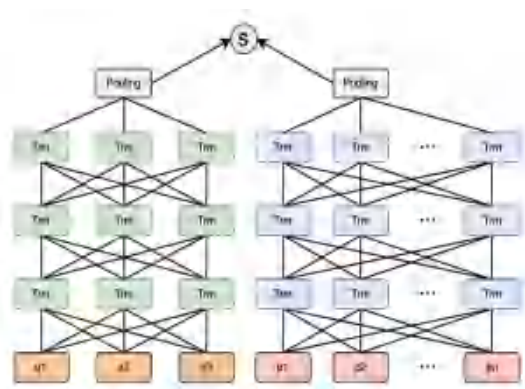
其他: 依赖大模型基础能力, 占比14.42%

请问“Je t'aime”中文是什么意思
“凡治众如治寡, 分数是也;”是什么意思?

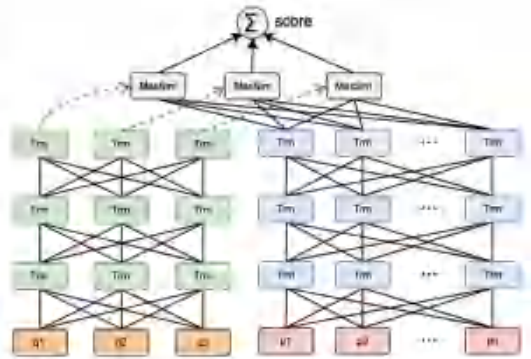
QCon 极客传媒

对于原文中有答案的问题，关键是通过检索系统找到与该问题相关的文本。根据用户问题检索相关文本之前需要对问题进行改写。因为在多轮对话中，用户常常会省略一些词汇，所以先对问题进行改写，然后再检索。

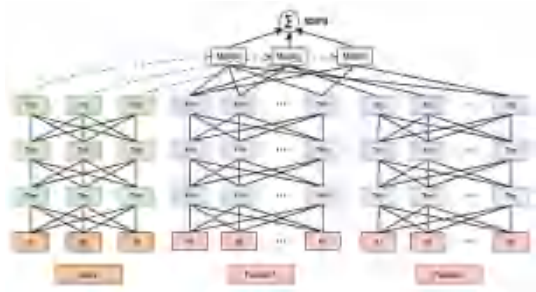
我们尝试了三种检索方法。首先是双塔架构，但在我们的场景下并不理想，召回率大约在 80% 左右。主要是原文片段经过 Pooling 方法进行语义压缩，导致相关文本片段的语义被稀释。如：一段 500 字的文本可能只有 50 字与问题直接相关，pooling 后的语义会稀释掉 50 字的语义，导致召回不足。



因此，我们尝试了第二种架构，保留了 500 字每一个词的向量表示，并计算与问题中每一个词的相似度。通过取片段的最大相似度作为整个文本片段的相似度，这样虽然效率有所下降，但准确率有显著提升，在业务数据集中，效果超过 text-embedding-ada-002。



最后一种情况，针对答案分布在不同的文本片段的情况，做了进一步的改进，效果也得到了进一步的提升。



Open QA 与 Close QA 的主要区别在于原文中没有问题答案，但是互联网上有相关信息，可以通过 QQ 浏览器的搜索引擎提供相关网页，然后通过大型语言模型输出答案。

Agent QA 系统是解决原文和搜索引擎都无法提供答案时，大型语言模型将复杂任务分解成若干小步骤，然后分而治之。如：用户想要了解腾讯流动利率时，LLM 会进行如下分解：首先，搜索流动利率的计算方法，即流动资产除以流动负债；然后，找出具体的流动资产和流动负债的数值；最后，使用计算器计算出流动利率。

这种方法听起来很好，但是存在一个问题，在专业领域，大型语言模型通常会泛泛而谈，模型往往无法规划出具体的执行步骤。为了解决这个问题，我们提出了一种新的解决方案：语言模型 + 专家知识库。

假设有一个专业问题关于“公司是否存在非法占用现金的情况”，大模型并不能做任务拆解，可以在知识库中检索到最相关的规划，然后让大型语言模型参考这个规划完成任务。实验显示，专家知识库可以显著提升专业领域问题的效果。

■ 优化实践：高效模型迭代加速策略

LLM 回复非常灵活，自动化评估是加速模型迭代效率的重要部分。以摘要功能为例，一种常用的

方法是完整文章和生成的摘要输入到大语言模型中，让 LLM 判断摘要的质量。



然而，这个方法的挑战在于，原文常含有大量无关信息，这可能导致模型错误地判断摘要是否准确反映了原文的主旨，详见参考文献。

但是，因为摘要通常是多个句子的汇总，当遇到融合性或概括性句子时，这个方法就不再有效，详见参考文献。

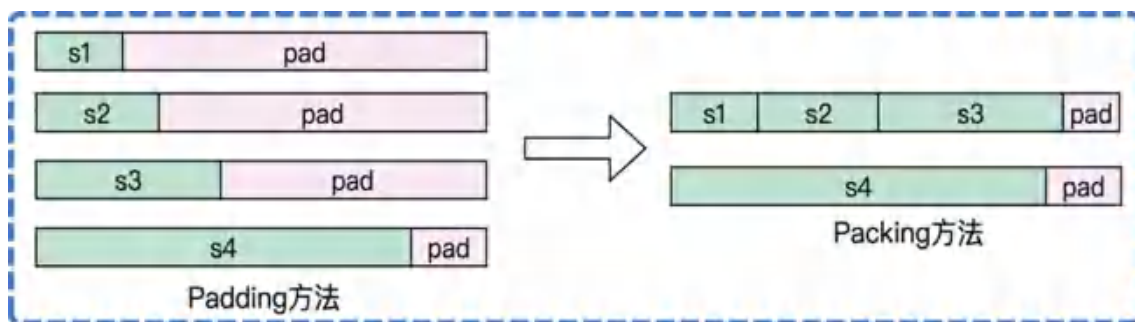
第二种评估方法可以参考 TAEL 的一篇文章。这个方法通过比较每个生成的摘要句子与原文中的句子是否相似来判断摘要是否产生幻觉。如果所有句子都足够相似，就认为摘要没有产生幻觉。

为了克服这一限制，我们采用了检索增强型方法，将精准问答的思想应用于自动评估。结果显示，在公开的摘要生成数据集上，我们的方法的问题可用率是最高的，达到了业界领先水平。

| | CoGenSum | SummEval | FRANK | FactCC | Avg |
|------------------|----------|----------|-------|--------|-------|
| 句子级别的幻觉检测 | 64.7 | 81.7 | 89.8 | 89.5 | 79.39 |
| 基于Large LLM的幻觉检测 | 75.5 | 87.53 | 80.76 | 90.06 | 83.46 |
| 检索增强的幻觉检测 | 85.3 | 88.46 | 80.95 | 90.24 | 86.02 |

在训练过程中提升收敛速度也是一个加速模型迭代的重要方法。训练过程中，每个批次可能包含不同长度的样本，常规用 padding 的方法会浪费

算力。我们采用了 Packing 策略，将多个短样本拼接在一起，以减少无效的填充部分，使得每个批次的计算更加高效。



实验表明，在达到相同训练效果的情况下，Packing 训练时长约 Padding 方式的 64.1%。因

此，Packing 策略大大提高了训练的效率和模型的收敛速度。



扫码分享文章

京东商家智能助手：Multi-Agents 在电商垂域的探索与创新

演 讲 | 韩艾，京东集团算法总监、京东零售数据与算法通道委员

编 辑 | 褚杏娟，InfoQ 编辑

傅宇琪，InfoQ 见习编辑



电商助手是一款集合了多种电商经营决策功能的工具软件，旨在帮助电商从业者完成从商品发布到订单管理、客服沟通、数据分析等一系列电商运营任务。

京东零售基于 Multi-Agents 理念搭建了商家助手大模型在线推理服务架构，这一系统的核心是算法层基于 ReAct 范式定制多个 LLM AI Agents，每个 Agent 都有专门业务角色和服务功能，可以调用不同的工具或多 Agent 协同工作来解决相应的问题。

在 QCon 北京 2024 大会上，京东集团算法总监、京东零售数据与算法通道委员韩艾，根据自己和团队在京东的技术实践经历，发表了题为《京东商家智能助手：AI 多智能体系统在电商垂域的探索与创新》的演讲，她阐述了 Multi-Agents 如何模拟真实的商家经营，并介绍 ReAct 范式的 Multi-Agent 在线推理架构，以及 Agent 落地垂域的样本、训练与评估监控的方法。

本文由 InfoQ 整理，经韩艾老师授权发布。以下为演讲实录。

现实中，商家如何进行经营决策

Agent 需要模拟人类的决策过程，因此需要先了解现实中的经营是如何进行的。

通常，平台向商家传递各种各样的信息，包括新的玩法、新的规则条款，以及可能的惩罚通知等。面对平台的各种消息和随之而来的疑问，商家需要一个经营助手协助，他通常扮演着一个专门提供平台知识百科的咨询顾问角色。

当商家提出赔付、运费等与业务相关的复杂问题，需要先理解需求，然后从长篇的业务文本中抽出问题解决的大方向或目标。定位问题后，形成逐步的解题思路，再灵活调用各种资源和工具来解决问题，其中包括调用知识库、进行搜索和检索，以及使用人脑进行总结和筛选重点内容。经过这一系列操作后将问题的最终答案返还给商家。

那么如何将现实空间的平台咨询顾问映射到 Agent？顾问这个角色是我抽象出来的，京东实际上并没有这样的角色。对于商家来说，每天提供专属服务的实际上是我的许多同事，包括在线客服、业务运营人员以及产品经理，他们解答各种问题。那是否需要为每个岗位角色构建一个 Agent？解决这个问题时，我们还要回到应用场景，从商家的需求出发：无论谁在回答问题，对商家来说都只有一个人帮助他们解答问题。因此，构建一个 Agent 即可，它映射到为商家提供专属

咨询服务的多个业务岗位的人。构建这样一个 AI 版的 Agent 对商家和平台都有好处。对商家而言，他们将体验到一个永远在线的百科全书，能够突破时间、体力和知识掌握的极限。对平台来说，可以降低成本。

除了上面单一的 agent 提供专属服务的情况，当我们讨论到多领域助手与商家的经营协作时，整个团队是如何协作经营的呢？比如，商家提出了一个问题：“最近我的店铺经营得怎么样？”这个问题看似简单，但实际上是商家每天在处理完各种信息后首先会思考的。

对于现代电商商家来说，了解经营状况通常从查看数据开始，然后才能评估经营状况。他不会直接去系统读取数据或编写数据库查询语言，而是直接“调度”数据分析师这一角色，因为商家清楚自己的目标是数据相关的服务。于是，他将任务分配给团队中的数据分析师，这位专家经过一系列操作后，会返回给商家一份数据报告。接下来，商家需要阅读并理解这份数据报告，他可能会发现新用户的留存率不佳的问题。这时，商家会根据新发现的问题更新决策。

商家的上述过程是 agent ReAct 范式的一个典型例子，即基于观察 (observation) 来更新整个推理 (reasoning) 过程。在解决问题的思路，

人类和 Agent 非常相似。

接下来,更新的决策就是商家重新选择一个角色,比如用户研究专家,来分析新用户的偏好,解决新用户的留存率不佳的问题。这样的“拿到结果更新决策 - 调度新的专业角色 - 输出结果”会不断循环往复。

一个经营诊断与优化的问题,电商商家团队的成员要懂得数据分析、平台知识、用户研究、商品选品、定价、营销投放,还需要有人掌握制作图片和音视频素材的技能,以及完成所有操作和客户售后运营。而商家自己,需要清楚地了解每个团队成员的专长(profile),以便在更新决策时知道如何调度这些资源。此外,商家还需要能够理解每个专家返回的结果,这对商家来说也不是件容易的事情。

当商家发展到一定阶段,他们通常会聘请一个“最强大脑”来代理所有这些调度工作。这个“最强大脑”可以被理解为一个“总管”。有了总管,所有的调度工作都由总管代理完成,而商家只需要与总管沟通即可。这样的协作模式可以极大地提高商

家的经营效率。商家想要完成一个经营诊断,他只需向总管提出:“帮我看看最近经营得怎么样?”然后他就可以耐心等待。总管在接到任务后,会进行一系列的操作,最终给出结论:“你最近新客户的留存情况不太好,我这里有一些商品营销创意的建议,你看看是否采纳。”相关的专家们的输出材料会作为附件提供给商家。

从单一个体到各个专业领域的专家团队,再到基础的执行工具,共同帮助商家完成了一个决策过程。在当前的团队配置中,可以关注三类主要角色:

- 领域专家:以咨询顾问为代表,这类角色不仅具备决策能力,还能够调度工具。在 AI 空间中,他映射我们的 Agent。
- 工具:这类角色不具备决策能力,只能执行任务。在 AI 世界中,映射为软件系统中已有的多种原子服务能力接口 API。
- 总管:作为整个决策发起的引擎,总管不需要在某一领域深耕,但必须具备通用的电商知识,了解如何经营业务。在面对问题时,总管能知道如何发起调度,负责整体的专业服务流程编排,在 AI 空间中,他映射我们最强的 Agent。

构建 AI 版的商家经营团队

商家经营团队的运作模式为我们提供了 AI Agent 的现实版样例。现在来到 AI 空间,请出我们的

商家智能助手,我们暂且称呼它为 Mario X。将现实空间的团队映射到 AI 空间,我们用大量

Transformers 和研发代码构建了一个 AI 版的商家经营团队：一个由 Master Agent（主代理）领导的多领域 Agents 团队，团队同时掌控着一系列原子能力工具 API。

这样的 AI 团队带来了多方面的好处：

1. 体验提升：商家可以享受到 7*24 小时的在线服务。
2. 效率提高：商家不再需要学习使用各种工具

和专业知识，只需用他们最熟悉的经营语言与 Master Agent 沟通，即可直接享受系统提供的各种服务。

3. 决策质量提升：由于有大量的备选方案可供选择，商家的决策效率和质量自然会提高。
4. 成本节约：商家可以减少人力和时间的投入，平台也可以减少不必要的运营开支，让我们的业务人员从繁琐的问答中解放出来。

■ ReAct Agent 构建

构建 ReAct Agent 时，每个 Agent 会经历一个 inner loop, 这个内部循环称为 reasoning (推理)，它对应于我们之前讨论的思维过程，即生成解题思路和大目标的步骤。reasoning 过程包含两个主要部分：

Thought (思考)：我将其定义为用人类自然语言描述的解题决策思路。但是，为了调度系统工具，LLM 需要发出指令，因此需要将这种人类语言翻译成系统能解析的研发语言（即下面的 action code）。

生成 Action Code (动作代码)：基于生成的 Thought, Agent 会继续生成 Action Code。这个 Code 不直接执行 Action, 而是执行 action 的指令。Action Code 是基于 Thought 解析出来的，因为 Thought 是拆分多步骤的解题思路，所

以 Action Code 是对应的一系列任务。每个任务的定义可能非常复杂，提取 JSON 中的一些简单字段来说明：

- 调度对象：告诉系统你要调度的工具是谁，比如 Master Agent 可能会调度其他 Agents 或 API。
- 输入信息：提供给调度对象的信息，即函数的输入参数。
- Job Description：如果调度的是 Agent, 需要让 Agent 明白分配给它的任务是什么，类似于工作描述。
- Trust_Mode：这是考虑性能和 Agent 质量的一个字段，它决定了 Agent 在接收到工具返回的 observation (观察结果) 后，是再次进行 reasoning 还是直接输出结果。

Action Code 是服务端可解析的代码，它会与环

境中广义的 Agents API 和 Tools 进行交互并执行代码。当这些工具完成工作并将 observation 返回给 Agent 时，Agent 将进行下一轮的 reasoning。这个过程会一直持续，直到 Agent

生成了一个 Trust_Mode 变为 1 的输出，这意味着 Agent 认为所有的推理和调度都已完成，可以将结果推送给用户。

■ Multi-Agent 的工作流程

打开 Mario X 首先会与商家打招呼。第一轮商家提问：“在京东开店需要交多少保证金？”时，用户和 Master Agent 之间建立了联系，它会再从 Memory 中获取与用户相关的近期和远期特征。

接下来，Master agent 开始内部推理。在这个阶段，Master agent 的 LLM 理解商家提出的问题，但意识到缺少必要的条件，因此无法直接派发任务。LLM 需要向商家追问一个条件，因为保证金与商家经营的类目密切相关。这时，它会调用一个名为 Echo 的工具，Echo 的作用仅仅是将信息传递给用户，不做任何处理。此时 Master agent 将 Trust_Mode 设置为 1，因为 Echo 的任务是单向传递信息，不需要再返回给 LLM 进行推理。Action Code 开始执行，Echo API 被唤起，将问题传回给用户，同时将上下文信息推送给 Memory。

第二轮，商家回答说他卖花，这时用户的信息再次流向 Agent，LLM 根据商家提供的信息和 Memory，生成解答思路在 Thought 中。LLM 知道现在需要调度的对象是 Consulting Advisor，即前面提到的平台咨询顾问 Agent 版。LLM 向 Advisor 传递了一个 Job Description，因为

Advisor 是一个 Agent，需要与之沟通并分配任务。Agent 之间的通信协议也是基于 Action Code，告知 Advisor 商家需要查询的类目对应的入住保证金费用。此时 Trust_Mode 设置为 1，意味着 Advisor 完成任务后不需要再返回给 LLM，因为 LLM 信任 Advisor 专门执行此类咨询任务。这是出于性能考虑，避免让用户等待过久。随后，Advisor Agent 执行任务并返回输出，同时更新 Memory。最终，Master agent 回答用户的问题。

第三轮，当客户提出为花店起名时，Master Agent 的 LLM 识别出这是一个明确的问题。为了解决这个问题，将会进行 3 轮 ReAct。第一轮：不需要调用其他 Agents，而是直接调用一个特定的 API 会更加高效。它调用的是一个名为“Shop Name Generator”的 API，这是一个基于大语言模型的起名工具，它需要接收的输入参数是店铺的类目信息。他从 Memory 中提取了之前 Advisor Agent 提供的信息，即商家经营的是“生活鲜花”，并将这个信息作为参数传递给 Shop Name Generator。在这一步，Trust_Mode 为 0，这意味着 API 生成的店铺名字将返回给 Master Agent 做其他的推理，而不是直接输出给用户。我们回

到了 ReAct 流程中，API 输出了一系列的店铺名字，但用户此时还不会看到任何输出的结果。

所有这些步骤完成后，相关信息都会被推入 Memory，这就是 Multi-Agent 工作架构的一个

例子。对于普通的 Agent 与 Master Agent 的区别在于，Master Agent 直接与用户交互，而普通 Agent 则接收来自 Master Agent 的 Action Code，这些 Action Code 转化为服务层协议，作为它们的输入参数。

■ 分层次架构

Multi-agent 架构采用分层次的方法，将一个大模型的复杂生成任务，拆解成了多个层级化的下一步文本预测。这样，每个模型需要处理的推理难度就相对较小，因此模型的规模不需要很大，从而减少了训练和部署的计算资源消耗，并且可以快速迭代。同时，也可方便灵活地接入各种资源方，比如营销的 Agent，我们可以迅速地将其整合进我们的系统中。

这种架构也有一些潜在问题。首先，可能导致风险的累积。如果 Master Agent 出错，那么整个任务的结果可能就会受到影响。因此，我们实施了全链路监控，以确保系统的稳定性和可靠性。

此外，由于可能需要经过多个 LLM 生成步骤，响应时间有时可能会较长。此外，商家面临的问题通常涉及工具操作，这些问题都需要结合具体的业务情境来解决。因此，对于我们的 Agent 来说，它们也需要“死记硬背”所有 Tools 的能力。目前，我们正在进行的工作包括在整个推理流程的多个环节中整合 Retrieval（检索）过程。例如在生成 Thought 之后，Agent 可以暂停并调用检索工具或 Agent，等待 Observation 返回后再明确调用哪个 Tools，然后生成 Action Code。这意味着 Thought 和 Action 可以分两轮生成，这是我们正在努力实现的一些改进。

构建 AI 版的商家经营团队

今年 2 月份，我们推出了一个专门处理招商入驻问题的 Agent，并将其部署在京东的招商站点上。

这个 Agent 帮助许多商家解答了他们关于入驻的相关问题和操作步骤。目前，这个全新的 multi-

Agent 架构助手产品已经在京东商家端进行灰度测试阶段。

技术上，我们目前的系统能够解决商家经营场景中的一些确定性输出问题。所谓确定性输出，是指商家面临的一些答案明确的问题，比如关于平台规则的疑问或具体的操作步骤等，这些问题相对基础，并不包括那些开放式的问题，比如“告诉

我如何做好生意”。

我们在建设一个能够真正帮助商家做生意的靠谱助手，搭建完整 AI agent 经营团队。这个系统将涉及非常广泛的知识领域，处理的问题也不会有确定的答案，可能需要引入强化学习等更先进的技术来解决。

■ ReAct SFT：垂域样本构建

在解决相对确定性输出的问题时，我们的核心工作在于构建垂直领域的知识。这意味着将人类专家的知识传授给系统，特别是针对商家领域的知识。对于这类问题，通常使用标准的 SFT 加上一些预训练模型基本上就足够了。

如何构建样本？鉴于我们先解决比较确定性的问题，我们可以从在线客服、运营和产品的回复，以及商家满意度收集的接口等获得真实的数据，然后对这些数据进行清洗。接着，研发团队会根据系统的调用路径构建一个全面的路径树。最后，业务人员将构造一些剧本，描述可能的问答场景。

将这两部分结合起来，我们就得到 SFT 样本的基础池。接下来，对基础池进行丰富度扩充。其中最主要的是对问题 (Q) 的扩充。有了问题和答案 (A)，以及调用路径，我们接下来需要生成中间的标签 (label) 即 thought 和 action code，这就需要依赖先验的知识库。此外，还需要研发的配合，他们需要按照标准来注册 API。因为工具的调用靠注册信息的质量，如果两个不同的工具，它们的描述写成一样的，那么我们的大模型也无能为力，因为它只能通过工具的自我介绍来选择工具来执行任务。因此，知识的准确性非常重要。

■ 复杂输入下的 Thought 生成

复杂输入的问题，不像简单输入那样直接。解决这类问题，关键在于遵循 Agent 推理的流程：先生成 Thought，再解析 Action Code。因此，生成一个强大的 Thought 变得非常重要。下面看一

个复杂输入下，确定性输出的例子，我们来对比单纯用 RAG 和用 LLM agent 解题的效果，比较一下有和没有好的 Thought 的区别。

(1) RAG 解题

例如，用户提出了一个问题：“在京东卖红酒要多少钱？”如果直接使用 Retrieval（检索增强）来解决问题，按照经典的方式，先进行 Query（查询）并计算 Similarity（相似度），然后召回一些文本。在召回的文本中，可能会看到白酒、黄酒等，但实际上并没有答案，因为红酒这个类目在我们的知识库中并不存在，它不是开店保证金的一个选项。基于错误的信息片段，再加上用户模糊的问题，即使是非常强大的 Summary Model（总结模型）也无法给出正确的答案。

要解决这个问题，我们需要让模型理解红酒实际上与哪些类目是有关联的。这就需要模型不仅要检索能力，还要有推理和关联的能力，以便正确地将问题与相关的知识库内容关联起来，从而提供准确的答案。

(2) LLM Agent 解题

助手中的 Advisor 在经过训练后，会以特定的方式解题。还是开始于 Master Agent 与用户的对话。Master Agent 并不直接理解这个问题，而是将用户的询问，例如“京东红酒入住资费是多少？”通过 Action Code 传递给 Advisor。Action Code 中的 Job Description 是“请回答京东红酒入住资费”。

Advisor 在处理这个查询时，首先理解红酒实际上属于葡萄酒这一类别。因此，Advisor 的 Thought 中生成出应该查询的是葡萄酒类目的入住资费，并确定了使用哪些关键词来传给调度的检索 API 做关键入参。在生成 Action Code 时，Advisor 会传递给检索 API 这个关键入参，即 Search Query “葡萄酒保证金”。这个参数不再是用户的原始问题，而是根据 Advisor 的推理进行了调整。API 本身没有决策能力，但由于 Agent 具有推理能力，它能确保传递给工具的输入是正确的，从而用正确的参数唤起正确的工具。

在第二个任务中，Summary API 接收到一个关键的输入参数，称为 Thought for Answer，即回答思路。这个思路是 Advisor 在推理过程中在 thought 生成的关于红酒与葡萄酒类目关系的理解。Advisor 告诉用户红酒和葡萄酒之间的关系，并按照葡萄酒类目的答案来回答用户的问题。

接下来，advisor 继续遵循经典的 RAG 流程。此时，Search Query 变为“葡萄酒保证金”。虽然召回的葡萄酒与原始问题的“红酒”相似性不高，但由于顾问使用了“葡萄酒”和“保证金”作为搜索关键词，并将回答问题的思路作为 Prompt 的一部分传递给总结 API，API 就能够根据 Advisor 提供的推理思路，正确地回答关于红酒保证金的问题，即通过查看葡萄酒的保证金来得知红酒的保证金情况。

■ 复杂输入下的 Thought 训练

在复杂输入的情况下，训练出能够准确生成 Thought 的模型是关键。由于这类问题的答案并不直接存储在知识库中，我们需要从算法层面进行构建。我们的方法是分析 Bad case (不良案例)，从中发现问题并拓展解题思路。

当遇到一个新案例时，我们会与业务团队沟通，以获取新的知识点，并按照既定的模式进行预先处理。理解不同类目之间的关系是解决相关问题的关键。因此，我们为模型提供了大量类似的文本进行预训练 (pretrain)。

在自监督学习阶段，模型学习了与业务相关的各

种关键词、相似词以及它们与类目的关系。这样，当模型遇到葡萄酒相关的问题时，它已经通过预训练了解了如何处理这类问题。随后，我们对模型进行标准的 SFT，在这个阶段，模型会学习到具体的知识点，比如葡萄酒的相关信息。模型已经知道如何回答关于葡萄酒的问题，并且通过训练了解了葡萄酒与其他类目的关系。当用户询问关于红酒保证金的问题时，模型能够通过分析和推理，提供准确的答案。

通过这种方式，我们可以训练出能够处理复杂输入并生成有效 Thought 的模型，这些模型能够更好地理解和解决商家面临的实际问题。

■ 全链路 ReAct 监控

为了定位 Bad Case，我们实施了全链路 ReAct 监控。具体来说，我们会收集在线推理生成的 Thought、Action Code 和 Observation，然后通过人工打标 + 大模型来进行评估。

评估函数会将人工打标的输出与 Agent 生成的输出进行比较，以确定两者之间的差异。这个评估与 Agent 的具体定义紧密相关，因为不同的 Agent 可能有不同的评估标准。评估主要基于三个结果：Thought、Action Code 和 Observation。值得注意的是，Observation 虽然是作为下一轮推理的输入，但它本身并不是由 LLM 生成的，它的质量会影响下一轮的 Thought 生成。

对于 Observation 的评估可能包括预测销量的准确性或用户对生成图像的满意度等，这些指标并不完全由 LLM 控制，因此 Observation 的评估也与服务的业务指标相关。

基于这些评估结果，我们会有一个流程来决定 Agent 的表现。如果 Agent 在第一轮的 ReAct 得分很低，我们会继续累积这个分数，但如果得分低于某个阈值，我们将停止后续的推理，并且该 Agent 将不再参与后续得分的累加，意味着它已经退出了推理过程。如果 Agent 的得分符合要求，我们会检查是否为最后一轮推理。如果不是最后一轮，Agent 将更新后进入下一轮评估。如

果是最后一轮，将触发结束流程。

在多轮推理和评估后，当触发结束评估时，我们会得到一个全链路累积的 ReAct 得分。这个推理过程是链式的，涉及到递减的折扣因子 γ 和 η ，这

些因子会影响 Agent 的 ReAct 得分和整体得分。我们的评价核心在于能够快速定位问题节点，这是由我们的架构决定的，必须通过这种方式来尽早发现并解决问题，防止问题在推理过程中蔓延。

展望

我们需要帮助商家更好地经营生意。尽管在平台上有许多类似参谋的工具，比如供应链管理、选品、定价等，但目前还没有一种方式能够让商家根据自己的业务思路，按照黄金流程组合使用这些工具。无论是问答数据、沟通数据还是交互数据，这些都需要我们去收集和整合。

我们需要将人们做生意的思维方式从人脑中提取

出来，这包括训练大型模型来寻找和学习人类专家的知识。此外，我们还需要引入强化学习。因为对于商家提出的复杂问题，如“我的生意做得怎么样？”可能存在多种解决方案，每个团队的解法可能都不同。要判断哪一个更好，可能需要每个做生意的人根据自己的打分逻辑来评估，同时还需要在市场反馈中验证。



扫码分享文章

第二章：企业生产提效

字节跳动代码生成 Copilot 产品的应用和演进

演 讲 | 刘夏，字节跳动产品研发和工程架构部代码智能助手架构师

编 辑 | 高玉娴，InfoQ 数字化主编



大语言模型在代码生成领域取得了令人瞩目的进展。本文整理自字节跳动产品研发和工程架构部的代码智能助手架构师刘夏在 AICon 2024 北京的演讲《代码生成 Copilot 产品的应用和演进》，聚焦基于大语言模型的代码生成技术，深入探讨

了代码补全和代码编辑这两种典型的应用形态。同时，还分析了当前代码补全面临的挑战和局限性，阐述了代码编辑是如何在交互和构建方法上实现创新。内容经 InfoQ 进行不改变原意的编辑。

代码生成 Copilot 产品回顾

■ GitHub Copilot 的成功因素

首先，回顾一下代码生成 Copilot 这种产品形式。当我们谈论代码生成 Copilot 或者 Copilot 这个词时，不得不提到 GitHub 在 2021 年 6 月推出的 GitHub Copilot。这个产品不仅拥有一个响亮的名字，而且定义了一种新的 AI 产品的范式。GitHub Copilot 在 2021 年 6 月推出了技术预览版，随着不断的迭代，其效果令人印象深刻，使人们意识到将大语言模型应用于代码生成领域具有巨大的潜力。业界也开始迅速构建类似的产品，无论是在模型还是产品上都取得了快速的迭代。

这里有一个关键问题：为什么是 GitHub Copilot 引爆了这个热点？实际上，将自然语言处理 (NLP) 技术应用于代码生成并不是一个新概念，例如 TabNine 这样的产品在 GPT-2 时代就已经将其应用于代码补全。那么，GitHub Copilot 究竟有何特别之处呢？我们想要从几个方面和维度来探讨这个问题。

首先，我想提到团队，GitHub Next 是这个产品的孵化团队。GitHub Next 是一个具有研究属性的团队，他们的任务是探索未来软件开发的新方式。如果访问他们的官网，你会发现许多有趣的项目，其中就包括 Copilot。团队主要由程序分析师、软件工程师以及研究员组成，他们持续关注

的一个重要话题是如何实现通用的代码生成。

接下来，我想谈谈一个重要的契机，那就是 2020 年 6 月 GPT-3 的问世。由于 GitHub 现在是微软的子公司，而微软与 OpenAI 有着深入的合作，GitHub 团队很早就获得了 GPT-3 的预览版，并对其能力感到非常兴奋。他们认为必须利用 GPT-3 在代码生成领域做出一些创新，因此与 OpenAI 紧密合作，基于 GPT-3 迭代开发出了专门用于代码的大型语言模型 Codex。随后，他们对 Codex 进行了持续的微调训练，打造了专属的模型。一个强大且优秀的基础模型实际上决定了产品的上限，因此 GPT-3 的出现对这款产品的贡献是巨大的。

有了模型之后，团队开始思考应该开发什么样的产品形态。根据 GitHub 的分享，他们最初的想法是开发一款 Chatbot，即一款能够解答编码过程中遇到的任何问题并提供代码的对话聊天产品。但他们很快发现，尽管知识库中大部分问题都能得到回答，但只有大约 20% 的回答是正确且被接受的。尤其是在 GPT-3 时期，ChatGPT 还要两年后才出现，他们意识到这种 Chatbot 产品的效果并不理想。如果大部分时候给出的答案都不是用户想要的，用户对产品的信任度会很低。

于是他们决定先采用代码补全这种容错率更高的产品形态，一方面代码补全是开发者使用频率非常高的功能，也有很强的依赖性，更重要的是开发者对于这个功能的预期是给出建议而不是 100% 准确的答案。

选择好产品形态后的一个要素是交互方式。GitHub Copilot 放弃了传统 IDE 中从下拉列表选择补全建议的交互，而是选择了用 Ghost Text 进行展示，用 Tab 键进行采纳，继续输入则取消推荐。这种交互方式发挥了模型在多行补全上的优势，推荐代码和已有代码融为一体，方便开发者快速基于上下文判断是否采纳。

代码补全产品的一个技术挑战是实现低延迟，

Jetbrains 在开发传统的补全功能时甚至要求在 150ms 内出现推荐列表以达到最佳的开发者体验。因为专业开发者的输入速度通常较快，过高的延迟会失去很多推荐的机会或者迫使用户停顿等待。GitHub Copilot 在大语言模型的推理速度和工程链路上进行了优化，让一个基于云端推理的 LLM 应用做到 500ms 左右的平均延迟。

如果说基座模型决定了产品能力的上限，那么提示工程所做的努力就是去逼近这个上限。通过研究开发者日常开发中会关注的上下文，在 prompt 中加入文件路径、相似代码、浏览记录等信息，让模型在代码补全方面的表现大幅提升，如今这些提示工程上的实践也被大家广泛应用。

字节跳动内部代码生成的探索历程

字节跳动在内部探索代码生成的过程中，面临多种优化选择：可以在模型层面进行优化，也可以选择工程链路上优化，或在交互体验上进行改进。团队需要灵活地做出决策。

随着大语言模型的发展，特别是从 2023 年开始，这个领域开始受到广泛关注，新的模型和产品层出不穷。为了迭代和优化模型，字节跳动首先建立了自己的评测方法和自动化评测系统。这

涉及到模型选型的决策，快速评估训练过程中的 checkpoint 效果，以及产品上线后如何收集线上反馈，包括用户编辑过程中的正反馈和负反馈。字节跳动还建立了一个完整的数据链路，以决定哪些数据被采纳，哪些被丢弃，并实施 A/B 测试系统来验证不同的 prompt 策略、参数配置，甚至是新模型的上线效果。字节跳动的自研大语言模型也已经发布，团队逐渐切换到这个自研模型上。基于此，字节跳动引入了对话方式，使代理

模型能够理解整个工程结构，并根据实际情况生成代码。此外，还引入了多点代码编辑推荐功能，这是一个较新的功能。今天的分享将围绕三个重点进行详细分析：

- 构建自研评测体系的重要性；
- 如何科学定义产品指标；
- A/B 测试的重要性。

■ 构建自研评测体系的重要性

构建自研评测体系的重要性在于，它可以帮助我们避免使用不恰当的评测指标，如 HumanEval，它可能无法准确反映模型在实际应用中的表现。HumanEval 通过完成人工编写的算法题并运行单元测试来评估模型，虽然模型在测试的分数可能很高，但这并不意味着模型在代码补全产品中的表现就一定好。例如，GitHub Copilot 在 HumanEval 上的得分可能不高，但其用户体验仍然出色。

自建评测集可以避免数据泄露问题，确保题目和答案不会被模型提前接触到。同时，自建评测集可以引入真实项目中的跨文件上下文，这对于评估模型能否合理利用上下文信息至关重要。此外，自建评测集还可以引入大量公司内部代码，因为

开源代码与内部代码的使用场景和分布可能存在显著差异。评测体系还需要包括基于单元测试的验证方式，因为同一功能可能有多种不同的代码实现方式，而单元测试可以更准确地验证生成代码的正确性。

最后，安全的自动化评测系统对于模型迭代至关重要。它不仅可以通过执行结果来验证代码的正确性，还可以防止模型生成有害代码，如删除根目录或造成大量内存分配等问题。高效的沙箱测试环境和高并发支持对于大规模的评测也是必不可少的。通过这样的评测系统，我们可以在训练过程中对不同 checkpoint 的模型效果进行评估，从而为模型选型和迭代提供有力支持。

■ 如何科学地定义指标

在科学地定义指标时，我们需要考虑代码补全流程中的各个环节，并确保所选指标能够准确反映产品优化的需要。一个有效的指标应该能够指导整个链路的优化，帮助我们识别瓶颈并进行相应的调整。采纳率是一个常被提到的指标，它通常

定义为采纳次数除以推荐次数。虽然这个定义简单，但它并不是一个好的指标。首先，采纳率容易被操纵。例如，如果减少推荐次数，只在非常确定的时候去帮你补一个分号，采纳率就会提高，但这并不意味着产品的实际效果有所提升。其次，

采纳率没有很好地拆解推荐和采纳过程中的具体因素，无法明确指出是推荐更快了，还是其他因素导致采纳次数增多。

体验指标是另一个需要考虑的方面。当用户在使用代码补全产品时，如果一个 Tab 操作就能接受推荐的代码并完成工作，这自然会带来良好的用户体验。体验指标可以反映用户对产品的满意度，但它并不直接指导产品优化的方向。在定义指标时，我们需要更细致地考虑如何反映产品的实际性能和用户体验，同时避免指标被操纵，并确保指标能够指导我们进行有效的产品迭代和优化。

在探讨如何科学地定义指标时，引入了 CPO (Character per opportunity) 这一指标，它是由一家专门从事代码补全产品的公司提出的。CPO 的计算公式由五个因子相乘得到：尝试率、反馈率、采纳率、每次采纳平均的 token 数以及 token 的平均字符长度。

尝试率指的是用户在编辑器中进行操作时，AI 提供建议的频率。例如，如果用户敲击键盘 10 次，但只有 6 次触发了对模型的请求，尝试率就是 6/10。这个指标反映了 AI 实际为用户提供建议的次数。

反馈率考虑了 AI 给出补全建议时存在的延迟问题。如果因为延迟太高，开发者已经进行了其他操作，那么即使推荐返回了也没有意义。如果发起 6 次请求，最终只有 3 次被展示，反馈率就是 3/6。

采纳率是大家熟悉的指标，即用户接受推荐的次数与推荐次数的比值。例如，三次推荐中只有一次被采纳，采纳率就是 1/3。

引入每次采纳平均的 token 数和 token 的平均字符长度这两个参数，是为了衡量不同长度代码带来的价值。不同的语言模型有不同的分词器，因此需要计算每个 token 平均的字符长度。例如，ChatGPT 的词表较大，平均一个 token 可以生成的字符数可能大于其他模型。

CPO 指标的计算公式是这几个因子的乘积，它衡量的是在每次有机会向用户推荐时，推荐了多少字符给用户。这个指标不仅可以衡量产品给开发者带来的价值，还可以拆解到整个链路的各个部分进行优化。例如，可以通过优化模型推理性能，提高反馈率，或者在代码注释中提供推荐来优化尝试率。此外，当线上出现问题时，CPO 指标也可以用来分析可能存在的问题所在。

■ A/B 测试的重要性

A/B 测试在产品开发过程中扮演着至关重要的角色。尽管离线评测可以帮助我们进行模型选型，但一个模型是否真正有效，还需要通过线上测试来验证。有时候，一个模型在评测中得分很高，

但这并不代表它在线上的实际表现同样出色。例如，一个非常强大的模型如 GPT-4，可能会因为高延迟而影响用户体验。

A/B 测试还可以帮助我们确定各种参数配置的合适值。比如，如果一个模型支持 16K 的上下文长度，是否就应该使用完整的 16K 呢？实际上，如果上下文过长，可能会导致整体延迟增加，影响用户体验。因此，需要通过 A/B 测试来找到最合适的上下文长度。

此外，A/B 测试还可以验证新的提示工程策略的效果。例如，如果我们在模型中加入了函数签名或其他包结构信息，是否真的能提升效果？模型

是否能够有效利用这些上下文？以及为了采集这些上下文信息而引入的额外延迟，是否值得？这些问题都需要通过 A/B 测试来验证。

最后，A/B 测试还可以帮助我们发现并改进产品指标。假设我们最初使用的是采纳率作为指标，但在进行 A/B 测试后，我们发现延迟提高后，采纳率反而增加了。这种情况可能表明我们的指标存在问题，需要重新考虑和调整。

代码编辑推荐：代码补全的进化

代码补全的进化形式可以被视为代码编辑推荐。大语言模型擅长生成下一个 token，这与代码补全或续写任务非常契合。然而，传统的代码补全主要针对编写全新代码的场景，而软件工程师在日常工作中不仅需要编写新代码，还需要编辑现有代码，包括重构和删除代码。在这些场景下，传统的补全功能可能无法高效地满足需求。在编

辑现有代码时，简单地删除一行然后重新编写是低效的。理想情况下，我们希望模型能够自动完成新增、删除、替换等操作，从而提高代码编辑的效率。因此，代码编辑推荐作为代码补全的进化，能够更好地适应软件工程师在实际工作中的各种代码操作需求，提供更加全面和智能的代码辅助功能。

■ 代码编辑推荐的概念

代码编辑推荐的概念涉及到一种更高级的代码辅助功能，它不仅包括传统的代码补全，还涵盖了对代码进行更深层次的理解和编辑。例如，假

设你写了一个 log 函数，该函数用于打印一个 message，并且有两个函数作为调用方来使用这个 log 函数。

```
1 def log(message: str):  
2     print(message)  
3  
4 def do_something():  
5     log("Critical error - please shutdown reactor")  
6  
7 def do_something_else():  
8     log("Everything is fine")
```

如果你决定给 `log` 函数添加两个新的参数，比如 `sourceMethod` 和 `level`，用以打印出对应的方法名称和日志等级，这时你实际上需要执行两个后续操作：首先，在 `print` 语句中添加新参数，以便能够打印出这些新信息；其次，在所有的调用方中也添加这些新参数，确保它们能够传递正确的值给 `log` 函数。

在这种情况下，代码编辑推荐的目标是让模型在你添加完新参数后，能够自动帮你完成剩余的内容。理想状态下，当你完成添加参数的操作时，模型已经预测出你需要在 `print` 语句中加入这些参数，并且在你移动到调用方时，模型已经知道

你接下来需要在这些调用点添加新参数。

在 Go 语言中，如果你有一个结构体并且希望它在多线程环境下保持线程安全，通常会引入互斥锁 (mutex) 来实现。在这种情况下，你需要在结构体的初始化 (`new`)、设置 (`set`) 和获取 (`get`) 方法中添加锁操作。智能的代码编辑推荐系统应该能够预测到你接下来需要进行的操作。例如，当你在 `new` 函数中添加锁时，推荐系统可以自动提示你在 `set` 和 `get` 方法中也添加相应的加锁代码。当你的光标移动到相应的方法上时，推荐系统就可以给出这些建议。

■ 数据构建和模型训练方法

数据构建和模型训练是提升代码生成能力的关键环节。模型的能力来源于数据，尤其是 Git 仓库中海量的 `commit` 数据，这些数据包含了丰富的

用户编辑信息。

现有的模型训练并没有充分利用这些数据，因为

它们往往包含噪音，例如在 commit 信息中夹带无关内容。因此，需要通过启发式规则或模型来过滤掉这些噪音，提取出有相关性和逻辑关系的编辑操作。

在编辑过程中，修正 Lint 错误是一个常见任务，这些错误信息及其修复方式也是非常宝贵的数据资源。在训练模型时，通常会选择一个基于大型

代码表示模型作为基础，并通过持续训练和 SFT (Supervised Fine-Tuning) 等方法让模型理解代码变更的差异。

模型在修正代码时可能会出现过度编辑的情况，即模型可能会过于激进地进行不必要的修改。因此，需要采取措施抑制这种行为，确保模型的编辑是恰当和准确的。

■ 进行中的优化

在进行中的优化方面，我们认识到目前的交互体验和展示方式可能并非最理想的状态。我们认为，集成在集成开发环境 (IDE) 中并进行一些 UI 上的定制，可能会带来更好的用户体验。

此外，我们已经在内部支持了对链接错误 (Link Error) 和警告 (Warning) 的修复功能。这是一个重要的进步，因为它能够帮助开发者更快速地

解决编译时遇到的问题。

我们还在探索光标移动的自动识别和推荐功能。目前，模型通常需要等到开发者的光标移动到特定位置后才能进行预测和推荐。我们希望优化这一点，让模型在开发者完成编码时就能预测下一步可能的编辑位置，并直接提供相应的推荐。这样的优化将进一步提升代码编辑的流畅性和效率。

代码生成 Copilot 的未来展望

对于代码生成模型来说，一个明显的趋势是能够处理更长的上下文。理想情况下，模型能够理解整个代码仓库的内容。目前，K 级别和 M 级别的上下文可能还不够，模型需要能够无限地处理上

下文信息。谷歌等公司已经提出了相关计划。但随着上下文的增长，保持推理速度不降低也是一个挑战，需要维持在几百毫秒的水平。一些公司如 Magic.dev 和 Supermaven 正在探索使用非

Transformer 架构来实现这一点。

对于产品形式，完全自主的 Agent 可能不太适合复杂的任务开发。程序员有时可能想用自然语言或注释来描述编码意图，但由于自然语言的局限性和文档编写的困难，最好的做法可能是 AI 与开发者通过交互的方式反复构思确认，并迭代完成复杂功能的开发。

AI 应该更智能地识别人类的意图，例如通过编辑位置的预测来主动参与编码过程，提前帮助预判并提供推荐。虽然这个概念比较抽象，但最近出现了一些体现这一思路的例子。Replit 公司开发的代码修复 Agent 展示了 AI 作为一个虚拟协作

者参与交互过程的能力。在多人协同的 IDE 中，AI 能够发现错误并以协作者的身份帮助修正，这是一种有效的主动式 AI 交互方式。

明尼苏达大学的研究“Sketch Then Generate”展示了一种人与 AI 交互持续迭代的方法。通过编写有结构化的注释来指导模型，这些注释可以与代码的实体、符号、方法关联起来，先构建代码架构，然后逐步指导模型生成更多细节和代码。

代码生成 Copilot 的未来将更加注重上下文理解、交互式产品开发、智能意图识别和人机协同工作，以实现更高效和智能的代码生成和编辑体验。



扫码分享文章

大语言模型加持，是智能运维架构的未来吗？

演讲 | 王宁，字节跳动技术专家

策划 | 蔡芳芳，InfoQ 主编

编辑 | 褚杏娟，InfoQ 编辑

傅宇琪，InfoQ 见习编辑

在 QCon 北京 2024 大会上，字节跳动技术专家王宁，根据自己在字节的实践经历，发表了题为《SRE - Copilot：基于大语言模型的智能运维架构》的演讲。本文由 InfoQ 整理，经王宁老师授权发布。以下为演讲实录。

随着大语言模型的广泛应用和能力的提升，许多团队都在尝试利用大语言模型来改进他们所在行业的应用，我们团队在字节跳动内部也在探索如何将大语言模型与 AIOps 实践相结合。

去年，我们带着 SRE-Copilot 框架参加了 AIOps 2023 挑战赛，并荣幸地获得了冠军。在比赛中，我们设计了一套更为开放和富有想象力的框架，进行了初步的探索。

比赛的题目是开放性的，旨在鼓励大家尽可能地探

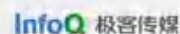
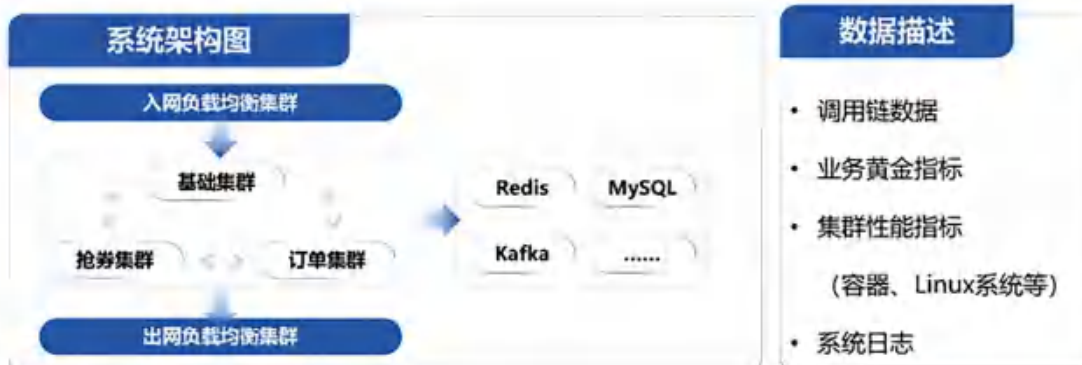
索大语言模型在智能运维领域的应用潜力。选手面临的挑战是帮助企业运维团队应对日益庞大的系统规模、复杂的系统结构以及日益增多的数据量。

比赛所用的数据来自中国建设银行“建行生活”APP 中的真实模拟数据。比赛的架构设计模拟了从入口负载均衡集群到中间的基础集群，如抢券集群和订单集群，这些集群之间相互依赖。此外，还包括了出口负载均衡集群，并且依赖许多复杂中间件如 Redis、MySQL、Kafka 等的各个集群。

我们在比赛中获得的数据包括调用链数据、业务黄金指标，例如订单成功率、抢券成功率，系统交易的每个订单的延迟时间，以及集群的性能指标，不仅涵盖了基础集群，还有中间件集群的监控数据，如 CPU 负载、常见的出入网流量，系统日志数据等。

赛题及数据分析

- 基于建行稳定性系统模拟建行生活类APP的真实环境
- 希望解决企业运维团队面对规模庞大、结构复杂、动态变化的运维数据需要解决的一系列挑战



为什么需要 SRE-Copilot 框架

随着系统日益复杂和数据量的增加，即使是专业的运维团队也难以全面掌握所有技术细节。在大型企业中，每个组件，如计算、存储、数据库，都可能需要专门的运维团队。当出现大规模故障时，通常需要所有团队协作来定位问题根源。而 SRE-Copilot 框架可以通过大语言模型的能力，学习和模拟不同团队的运维知识，以解决整个链路上的技术问题。

传统的 AIOps 在异常检测和根因分析上严重依赖

于标注数据，这限制了算法的泛化能力，因为它们需要在有监督的环境下进行训练。而大语言模型能够学习更多的通用知识，减少对标注数据的依赖，从而降低训练成本。

运维团队积累的专家经验很难编码到算法模型中。通常，这些经验会被简化为阈值或复杂的规则，不仅难以维护，也难以传承。SRE-Copilot 框架通过大语言模型，将专家经验转化为模型可以理解 and 推理的形式，从而提升了故障处理的能力。

传统 AIOps 的接入和维护成本较高，需要业务和算法团队深入理解业务逻辑和算法模型。此外，私域数据的处理和定制化开发也增加了成本。SRE-Copilot 框架采用集成学习的概念，通过模块化设计，使得系统能够像搭积木一样动态编排。

在传统 AIOps 中，未遇到过的故障很难被解决，因为它们超出了模型的训练范围。大语言模型展现出了强大的推理能力，能够基于通用知识和训练中学到的关键字，推断出未知故障的性质，即使没有相似的训练数据。

传统的 AIOps 解决方案需要用户理解模型并精确地传递参数，而 SRE-Copilot 框架支持自然语言交互，使得非技术用户也能轻松地与系统交互，提高了用户体验，并有潜力开放给更广泛的用户

群体。

SRE-Copilot 框架，采用了基于 function call（函数调用）的方法来实现多功能的智能运维。

我们首先将 SRE 在日常运维中可能遇到的多种场景进行了统一收敛，通过大语言模型来理解用户的意图。无论是进行运维可视化还是故障分类，我们的目标是编排不同的工具，以实现一个多场景的智能运维解决方案。

在比赛上，我集成了一些简单的场景，虽然每个场景本身并不复杂，但我们尽可能探索了许多新的场景，以测试和展示 SRE-Copilot 的多功能性。比如我们实现了故障分类功能，并让框架能够生成故障自愈的代码，以自动化处理常见的问题。

实践效果如何？

我利用大语言模型生成了一个排查故障的工作流，并以自然语言的形式呈现。

根据提前设定的 Agent，模型会匹配并确定哪些 Agent 能够在排障工作流程中发挥作用，然后将这些 Agent 编排成一个可执行的工作流。我可以通过自然语言向模型提问，比如询问特定时间段内的问题。模型会提取相关参数，并动态地将任

务分配给相应的子 Agent。每个子 Agent 会检查自己的数据，寻找故障迹象。

例如，调用链 Agent 会检查调用链是否存在问题。如果检测到 CMDB（配置管理数据库）中的信息，模型可以进行下钻操作，比如定位到具体的集群，然后触发新一轮更深入的检测。如果没有更多的信息，模型会开始进行根因诊断，检索历史故障

和专家经验，以此来判断当前的故障类型，比如磁盘写满。

在诊断过程中，我会将相关指标进行可视化展示，帮助理解故障的性质。根据诊断结果，我会生成清理磁盘的简单代码。如果是线上执行，模型会匹配预设的自愈方案。每次故障诊断完成后，会自动生成告警总结和复盘报告。诊断的故障信息

会自动积累下来。如果用户对诊断结果进行确认，那么这个故障案例就会被记录到历史故障库中，供模型未来诊断时参考。

在比赛的复现过程中，我们在建行云的 8 核和 16G 内存的堡垒机上进行了演示，并使用了一张 V100 显卡。实际上，我们仅使用了开源的 ChatGLM 6B 的小模型，就实现了上述效果。

比赛效果展示

采用与建行云堡垒机同一套代码复现 相近配置（8c16g显卡Nvidia Tesla-v100）模型chatglm3-6b



SRE-Copilot 架构

■ Tool calling

在 SRE-Copilot 架构中，Tool calling 是指将大语言模型与外部工具有效结合或交互的能力。这种机制允许模型解决它自身无法直接回答的问题，因为它可以调用外部工具来获取所需的信息。

以询问天气为例，大语言模型本身无法提供实时天气信息，因为它缺乏对外部环境的感知。为了解决这个问题，需要设计一个封装用户问题的方法，并提供一个能够获取当前天气的接口工具，再通过一些描述指导模型使用接口。

例如，接口可以根据给定的地点查询天气，它是一个 function，需要两个参数：地点（location）和温度单位（如摄氏度或华氏度）。成熟的大语言模型，如 ChatGPT 或文心一言，通常都是通过相似的训练方法来实现这一机制。模型会调用适当的函数，并且能够从用户的问题中提取出必要的参数。假设用户询问的是伦敦的天气，模型会自动将“伦敦”作为 location 参数补全，并指导调用天气查询接口，从而提供准确的天气信息。

■ RAG

SRE-Copilot 架构中的另一个重要概念是检索增强生成（Retrieval-Augmented Generation，简称 RAG）。RAG 结合了搜索技术和大语言模型，旨在解决几个关键问题。

首先，它可以帮助处理私域知识，例如公司内部服务器的状态或特定订单的详情，这些信息是大语言模型无法直接感知的。

RAG 能够解决新知识的问题。以电影为例，如果模型的知识截止日期是 4 月，那么它不会知道 5 月上映电影的主演是谁。同样，对于当天发生的

新故障或新闻，模型也无法感知。

RAG 也适用于长尾问题，即那些在模型训练时不常见或非常具体的问题。例如，如果用户需要编写某个小众语言的代码，或者询问关于特定数据库连接的问题，尤其是当这个数据库是公司内部改造过的，传统的大语言模型可能无法提供答案。为了解决这些问题，可以利用传统的检索方法，比如搜索公司内部文档，查找是否有关于特定语言连接数据库的 QA 文档。一旦检索到相关文档，我会将文档中的相关内容与用户的问题一起输入给大语言模型，以便模型能够提供一个更准确的答案。

■ Reason+Act

ReAct 概念针对的是那些无法仅通过一步查询或大语言模型自身直接解决的任务。这些任务通常需要多步骤的执行，并且每一步都需要模型提供其思考过程，以减少模型产生幻觉现象的风险。

通过 ReAct，我们可以将任务的每一步规划和执行可视化。例如，如果任务是让大语言模型去厨房做菜，模型可能会首先思考需要哪些调料，并预测它们可能存放的位置。比如，模型可能会推断出胡椒可能在 1 到 6 号柜子里，或者 1 到 3 号台面上。模型首先会选择检查第一个柜子，如果没有找到胡

椒，它会根据这一步骤的结果，决定下一步检查其他柜子。

再比如，当用户询问 GitHub 上某位开发者获得的点赞数并想要得到该数值除以 2 的结果时，模型的第一步是通过 GitHub 的 Open API 获取点赞数，然后使用计算器工具进行除法运算，通过这两步操作来得到最终结果。

ReAct 概念旨在通过分步骤地规划，并在每一步中进行合理的推理和决策，来执行越来越复杂的任务。

■ Agent 智能体

智能体主要由三个部分组成：

1. 大脑：最核心的部分是大语言模型，它扮演着智能体的大脑角色，负责处理和生成语言输出，同时整合和利用知识库中的信息。
2. 感知：智能体的第二部分是它的感知能力，这涉及到使用外部工具来感知图片、声音等信息。通过这些工具，智能体能够与外部世界进行交互，获取必要的信息。

3. 行动：智能体还可以通过计算 API、查询 API 等工具进行实际的交互操作，执行真实世界中的行动。这使得智能体能够处理更为复杂的任务，如根据图片内容推断位置并查询当地天气。

我们期望实现的是多个智能体之间的复杂交互，让智能体不仅能够独立工作，还能够相互协作，共同解决更加复杂的问题。

■ 架构设计思路

在 SRE-Copilot 框架的具体实现上，我根据比赛时提供的数据，在最底层为每个数据源设计了相

应的 Agent。这些数据源包括日志、调用链、交易类型数据、主机监控数据以及 CMDB 数据等多

元数据类型。对于每种类型的数据都设计了一个 Agent，使其能够进行异常检测、数据可视化，以及查询历史故障和返回故障描述等操作。

在底层 Agent 之上，我定义了功能型 Agent，它包括知识库问答、 workflow 规划、故障报告编写和代码生成等能力。这些功能型 Agent 能够基于 ReAct 框架，调用底层的子 Agent，并动态地编排它们的执行顺序和结果。

在框架的最顶层，我设计了一个名为 Copilot 的智

能体，它作为与人类用户交互的核心。Copilot 负责识别用户的意图，判断用户是想要获取答案还是进行故障诊断，并从用户的问题中提取出相关参数。

例如，用户可能询问某个具体时间段的情况，Copilot 可以将这个时间段作为参数传递给后续的处理流程。随着大语言模型的智能化，它甚至能够理解并处理如“过去 15 分钟”这样模糊的时间段，并将其转换为具体的参数。顶层 Copilot 还能够进行任务分配和 Agent 之间的协调工作，确保整个框架能够高效地响应用户的需求。



在设计 SRE-Copilot 框架时，我借鉴了 GPT 背后的集成学习思想。我们的目标并不是开发一个庞大而全面的模型来解决所有问题，而是通过集成多个专精于特定领域的子 Agent 来实现。每个子

Agent 都在其专业领域内表现出色，我们采用混合专家系统 (Moe, Mixture of Experts) 的形式，使整个系统的效果更加完善和高效。

SRE-Copilot 技术细节

■ 主要运维能力：异常检测

首先介绍一个异常检测场景，以下是我们在这个场景中定义的角色：

1. Copilot 主持人：作为核心，Copilot 负责解析用户需求，制定运维计划，并安排不同 Agent 执行具体任务。
2. 多数据源 Agent：针对不同的数据源，如日志、调用链、交易类型数据、主机监控数据以及 CMDB 数据，我们设计了专门的 Agent。这些 Agent 利用合适的算法对各自模态的数据进行异常检测和检索。
3. RCA Agent：该 Agent 负责收集其他 Agent 的检测结果，并执行最终的根因分析推理。

以 Copilot 为起点，例如，当收到用户关于 9 月 18 日下午 4 点大量交易失败的请求时，Copilot 会识别这是一个根因诊断问题，并将任务交给 RCA Agent。

RCA Agent 会关注用户请求中的关键信息，如交易类数据，并将其传递给交易类型 Agent 进行初步检查。该 Agent 将返回关于交易量同比下降的结果，但不提供额外信息。

根据初步检查结果，系统会动态调度其他 Agent 检查各自负责的组件是否存在问题。例如，当轮到第五个 Agent，即调用链 Agent 时，它发现了调用链上的异常情况。调用链 Agent 的反馈将引导 CMDB Agent 在 CMDB 中查询接口的调用记录，上下游关系，以及接口的具体问题。

通过这种方式，我们模拟了真实线上云平台中多个组件之间的协同定位过程。整个异常检测过程完全自动化，没有一个固定的流程，而是根据每个 Agent 的检测结果动态地调度其他 Agent，共同完成异常检测任务。

■ 主要运维能力：根因定位

有了上一步的异常检测结果，接下来需要对这个异常检测的结果做根因定位，或者说故障的分类。

首先，获取异常检测组件的结果，并对其进行了一些调整，使其返回的不仅是传统的 true/false 或 JSON 结构体，而是大语言模型和人类都容易

理解的自然语言描述。例如，异常检测结构体可能会返回：“本次故障持续了 10 分钟，CPU 指标飙升，内存也被打满，某某接口出现了大量失败”。

然后，这个故障工单会转换成向量，并在提前构建的向量数据库中检索。向量数据库包含两部分内容：一部分是提前配置的专家经验，另一部分是历史上遇到的相似故障记录。例如，如果历史上遇到过内存打满的问题，通过重启服务解决了问题，这样的专家经验会被记录在系统中。

在当前故障检测时，系统会同时检索出相关的专家经验和历史故障信息。通过 RAG，将当前故障

描述、专家经验和历史故障信息一起输入给大语言模型，进行根因推断。大模型会根据提供的信息推理，例如，这是否是内存打满的问题，是否需要通过重启服务来解决。

RAG 方法可以帮助大语言模型不断地更新其知识库，适应新的故障情况，同时减少模型的幻觉现象。通过引入历史故障数据和反思机制，模型的准确率得到提升，幻觉现象得到降低。即使在没有提前配置专家经验或历史故障数据的情况下，使用的 6B 大小的模型（如 ChatGLM 3）仍能够对某些类型的故障做出准确的推理，如磁盘写满或 Java GC 问题。

字节的实践探索

■ 基于 LLM 的 RCA-Agent 构建

字节跳动内部的目标是先将基于大语言模型的根因诊断（RCA）Agent 框架落地应用，因为根因诊断是 SRE 团队面临的一个主要挑战，它占用了大量的时间和精力，日常的 On Call 问题定位也给团队成员带来了沉重的负担。我们希望专注于解决这些实际问题，真正缓解 SRE 同事的痛点。

我们定义了一些工具和插件，是在出现故障时用来进行检测的工具。除了工具和插件，我们还设计了 workflow 编排，以自动化和优化故障处理流程。我们构建了一个知识库，它包含了历史故障数据、专家经验和故障处理策略，这些都是进行有效根因分析的关键资源。

■ 知识库的构建

构建知识库方面所做的工作主要包括以下几个部分，并且我们计划未来会引入更多用户原始文档、历史 On Call 记录等不同类型的数据库。

排障专家经验：这部分是根据根因诊断的场景特别设计的，目的是让业务团队的成员能够管理和记录他们的知识和经验。我们定义的每一个经验都是一组根因故障，包括故障发生时的描述和一些止损措施的组合。这些信息将被用来训练大语言模型推理。

例如，流量突增导致的故障，其根因可能是用户 QPS 的突增。故障的表现可能是流量首先突增，随后内存和 CPU 使用率也跟着上升，最终导致

服务不可用。这种描述将帮助模型理解故障模式。对于上述故障，可能的止损措施包括重启服务或进行扩容操作。

故障场景 SOP 文档：我们希望用户输入的是一些 SOP 文档。这种方式给组件团队提供一种灵活管理知识的方法。我们选择这种半规范化文档的形式，是因为当前大语言模型的能力还有局限，需要通过文档梳理来帮助模型更好地理解。

历史故障信息：我们还维护了一个历史故障信息库，记录每一次通过大语言模型检测到的故障，这些记录会用来对组件团队进行训练和打标。

■ 基础工具的构建

在构建 SRE-Copilot 框架的基础工具方面，我们参考了 OpenAI GPTs 将工具集成到平台时所遵循的规范。我们将运维场景中的一些关键指标和基础工具进行了统一管理，把传统的异常检测方法统一成一个工具，用户只需要维护他们需要异常检测的指标即可。

用户可以自定义检测项，包括指标名称、指标的标签或指标描述，以及定义何为异常表现。因为

是用户自定义的工具，所以可以根据具体需求设置检测标准。

我们实现了一个变更事件查询工具，当出现故障时，用户可以通过调用这个接口来确定是否由线上变更导致。我们在平台上部署的组件配置了一些工具，例如异常检测、变更和事件查询等，还包括了自然语言的意图理解和大语言模型的根因推理功能。

基础工具的构建

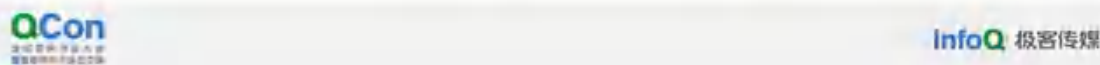
参考openai的tools/gpts的接入方式，我们将运维场景的指标和其他基础工具管理起来，基础工具包含几类，例如指标通用的异常检测，变更事件查询，组件自定义的检测项。

1.一些集群诊断场景的指标 2.自定义的检测项

用户实际部署的工具包含通用指标的异常检测，变更事件查询，自然语言的意图理解，大语言模型的根因推理



| ID | 名称 | 值 | 单位 | 更新时间 | 更新时间 | 报警 | 备注 |
|----|---------------|---------|------|---------------------|---------------------|----|----|
| 1 | cpu_usage | 15% | % | 2024-07-25 14:12:00 | 2024-07-25 14:12:00 | 正常 | |
| 2 | memory_usage | 75% | % | 2024-07-25 14:12:00 | 2024-07-25 14:12:00 | 警告 | |
| 3 | disk_usage | 80% | % | 2024-07-25 14:12:00 | 2024-07-25 14:12:00 | 警告 | |
| 4 | network_io | 100MB/s | MB/s | 2024-07-25 14:12:00 | 2024-07-25 14:12:00 | 正常 | |
| 5 | system_load | 0.5 | | 2024-07-25 14:12:00 | 2024-07-25 14:12:00 | 正常 | |
| 6 | error_rate | 0.1 | % | 2024-07-25 14:12:00 | 2024-07-25 14:12:00 | 警告 | |
| 7 | response_time | 50ms | ms | 2024-07-25 14:12:00 | 2024-07-25 14:12:00 | 正常 | |
| 8 | availability | 100% | % | 2024-07-25 14:12:00 | 2024-07-25 14:12:00 | 正常 | |



核心工具：LLM 根因推理

关键的根因推理功能基于大语言模型。在实践中，我采用了一种新的方法，不再将故障数据压缩成向量空间进行聚类分析，而是利用大语言模型将故障映射到自然语言空间中，从而简化了故障分类过程。

例如，流量突增、内存升高和接口不可用等信息可以自然地描述为 QPS 问题。对于新出现的具有相同描述的故障，可以直接使用大语言模型进行分类，而无需计算向量空间中的相似度。

分类过程中会使用用户配置的专家经验和当前故障的检测结果。此外，工具还利用预定义的模板生成描述，并让大语言模型进行推断，而用户可以根据需要调整这些模板、专家经验和检测结果。

为了提高根因推断的准确性，确保检测项插件的描述足够具体，比如详细描述指标曲线的形态变化，以及变更检测组件提供的详细信息，如变更的范围和性质。这样的方法不仅提高了故障分类的效率，还能让大语言模型的推理过程更加精确。

■ 工作流的构建

下一步是构建工作流，目前这一过程仍然需要用户自行配置，这主要是由于大语言模型当前能力的限制所做出的妥协。

不过，我们正在探索一种新的方法，即允许用户在其 SOP 文档中预先设定工作流，例如，文档中可以指明首先需要检查哪些指标，以及根据这些指标的结果接下来应该检查哪些指标。

我们希望能够训练大语言模型，使其能够直接根据用户的 SOP 文档生成工作流。最终，SRE 团队能够向大语言模型提供一个简单的文档，甚至是未经格式化的文本，而模型能够根据文档中的指标或检测项动态地编排诊断步骤，并根据每一步的检测结果，智能地调度后续的执行流程。

■ Agent 的应用与调试

我们在一些组件上进行了 RCA 的试点工作，实现了一些改变传统交互方式的效果。现在，用户可以通过提出模糊的问题来与系统交互，例如询问某个集群存在哪些故障，系统会自动识别并调用相应的集群诊断工作流。

完成诊断后，系统不仅能够将结果进行美化或封装，还能以卡片或自然语言的形式向用户直观展示。此外，在面对大规模故障时，系统能够自动解析故障群中的告警卡片信息或历史消息，判断需要诊断的参数，如特定集群和时间段，自动提取所需参数，并触发相应的诊断流程。

未来展望

针对我们遇到的瓶颈，我认为未来需要继续在下面几个方向增强。

第一，我期望有越来越大、能力越来越强的通用

大语言模型，因为随着这些模型能力的不断提升，在动态根因推理和每一步的动态决策上，它们的准确率和效果都将得到显著提升。

第二，我期待能够开发出更强大、更专业的模型，

例如时序大模型或日志大模型，并将这些专业模型作为工具，供核心 Agent 调用。

第三，我正在探索是否能有更灵活的 Agent 框架，以支持多轮和更灵活的对话，比如让用户在排障过程中随时打断并提供关键信息，以此缩短故障排查的平均时间。实现多人或多 Agent 之间的真正协作。

第四，我期望 Agent 能在更丰富的场景中应用，目前它们可以像实习生一样帮助进行简单的监控数据处理，未来随着模型和工具能力的增强，我希望它们能够发展到像初级员工或应届生那样进行一些简单的决策，甚至最终成为一个资深专家，能够自动诊断问题并执行高级决策。



扫码分享文章

用 AI 面试员工的企业，知道打工人在想什么吗？！

作者 | 褚杏娟, InfoQ 编辑

赵钰莹, InfoQ 极客传媒副总经理

“

知乎上有人详细分享了自己参与 AI 面试的经历：

“点开链接，在手机上是用小程序，电脑是浏览器。一开始有个预录，检查手机摄像头、麦克风以及自己的语音声音大小。

开始作答之后有 5 个题目。每个题目 6 分钟，包括构思和录制视频。点击录制视频有 5s 的准备时间，5s 之后手机自动录像，你开始答题，视频录制不超过 5 分钟。

录制完之后，返回答题页面，可以回放。5 分钟答题时间到，它会提醒提交，然后点击提交。就进入下一题。”

“我觉得 AI 面试的公司压根就没打算招人”，这是

AI 面试刚出来时人们通常会给的评价。但现在，大家对 AI 面试的接受度已经很高，AI 面试的争议更多是出现在一两年前的，现在网上几乎都是 AI 面试的建议、题库等。

“和传统面试比起来，我更倾向于用 AI。AI 对于你的输入、表情、动作都会关注，更加全面考核，至少是对所有候选人是公平的，不会出现面试官主观的偏见偏差、个人因素等问题，当然也得要求 AI 面试要足够智能。”广州大学嵌入式应用专业的一名本科学生说道。

根据《2024 牛客智能制造业校园招聘白皮书》，53.5% 的对智能制造感兴趣的学生有参与过 AI 面试，而参与过 AI 面试的学生中高达 77.7% 的人表示满意。

这次，传统行业走在了前头

面试可以用 AI 进行，也说明了面试本身在一定程度上是可规则化的。

有做人力资源管理的网友分析称，抛掉“人间冷暖”不谈，面试的本质是按图索骥，一场高效的面试依赖以下几个方面：

人才画像 ——AI 可以结构化提问 ——AI 可以关键词评估 ——AI 可以情绪稳定度 ——AI 应该可以（可能不精确）与公司文化 / 直接上级的默契度 ——AI 也许可以（双向测试后可以提升匹配度）

如果一家公司坚持不懈地做 AI 面试，积累匹配组织发展的人才大数据，那么招聘的人才至少 80% 左右是完全靠谱的，再加上终面 BOSS 感受一下，精准度应该可以达到 90% 以上，这比有 3-5 年工作经验的 HR 靠谱多了。

当然，AI 面试最终会让类似的人都聚集在一起，很有可能导致组织僵化。在组织需要变革、寻找一些鲶鱼进入组织时，人才画像将完全不一样，AI 的底层面试逻辑就要随之重构。

“我曾经只是作为辅助面试人员参与了公司里的终面过程，一整个上午也就搞了六七个人的面试，到中午吃饭的时候，主面试官基本上就快累趴下了。但是在我看来，很多面试官工作内容并不复

杂，甚至到后面有一些机械。不管面试者说啥，面试官都是在固定重复问几个问题。”弗兰克扬在知乎上分享称，“我也不觉得这会有什么大问题，因为最关键的信息其实就那么一两个，剩下的就是看你还想从聊天中聊点什么了，不管你聊什么，大部分都不会影响最终的决定。”

“弗兰克扬”表示，十多年前，一些大企业会把前几轮的招聘流程给外包公司，这些外包公司先筛选简历、再进行电话面试，问的问题都是雇主公司规定好的，全程录音，然后再根据面试情况做筛选，最终把报告发给雇主做最终决定。“现在 AI 的水平，我感觉跟当年外包干的工作都差不多。”

牛客联创兼技术负责人杨之贤介绍称，AI 面试特别适合招聘规模较大、考核能力相对标准的岗位，如校招管培生、普通蓝领、销售岗位、客服群体、小语种岗位等。以互联网为例，IT 基础岗位初筛、产品运营岗位、销售岗位、客服岗招聘人数超 10 人以上均有可能使用 AI 面试。

消费行业是最早尝试使用 AI 面试的行业之一，而且外企的接受度更高，比如雅诗兰黛、宝洁等。今年，随着大模型的兴起，越来越多的行业对 AI 面试表现出了兴趣，互联网、国企央企、银行、教育、电信、汽车、快消、制造业等行业都引入了 AI 面试。

比如 2023 年，光储行业里的龙头企业阳光电源决定将所有管培生岗位的英语面试全部使用牛客 AI 面试代替以往的英语外包面试。当年，阳光电源 AI 面试的管培生超 500 人次。

杨之贤表示，目前国内大约有万级别的企业在面

试流程中加入了 AI 面试，而且这一数字还在不断增长。据《牛客 2023 秋季校园招聘白皮书》的调研数据指出，已有 23.2% 的先锋企业应用 AI 助力校招，其中使用 AI 助力笔面试环节的企业占比达 97.9%。

被放大的细节

AI 面试的优势是不会受到情绪、偏见或其他主观因素的影响，它会根据事先设定的评分标准进行客观评价。AI 面试会重点会考察面试者以下三个能力：

- 胜任力测评，包括学习能力和抗压能力，这些通常是衡量一个人是否适合工作的重要指标。
- 专业能力，即面试者在特定领域的知识和技能。这是根据不同岗位的要求来考察的，以确保面试者具备所需的专业素质。
- 语言能力，比如英语、越南语等，良好的语言能力对于与外企的国际团队合作和沟通至关重要，还有的工作也需要一定外语能力。

结合面试者的简历和回答，AI 面试系统会进行提问和追问，并利用大模型的自然语言理解能力和逻辑推理能力，给出面试者对应能力项的得分。

“智能判分是基于岗位专业素质、通用能力素质和语言能力素质的综合科学判定。我们会综合考虑面试者回答内容的专业性、相关性和逻辑性等因素，并结合回答状态进行综合判定。”杨之贤说道。



AI 面试成绩单

AI 面试评判更加标准化，这就要求应试者当下就要快速反应，回答得清晰、有逻辑，覆盖题目中所有的问题。“回答得是否高大上不重要，重要的是你的回答是否具有逻辑。”网友“肉丝 er”也分享道。

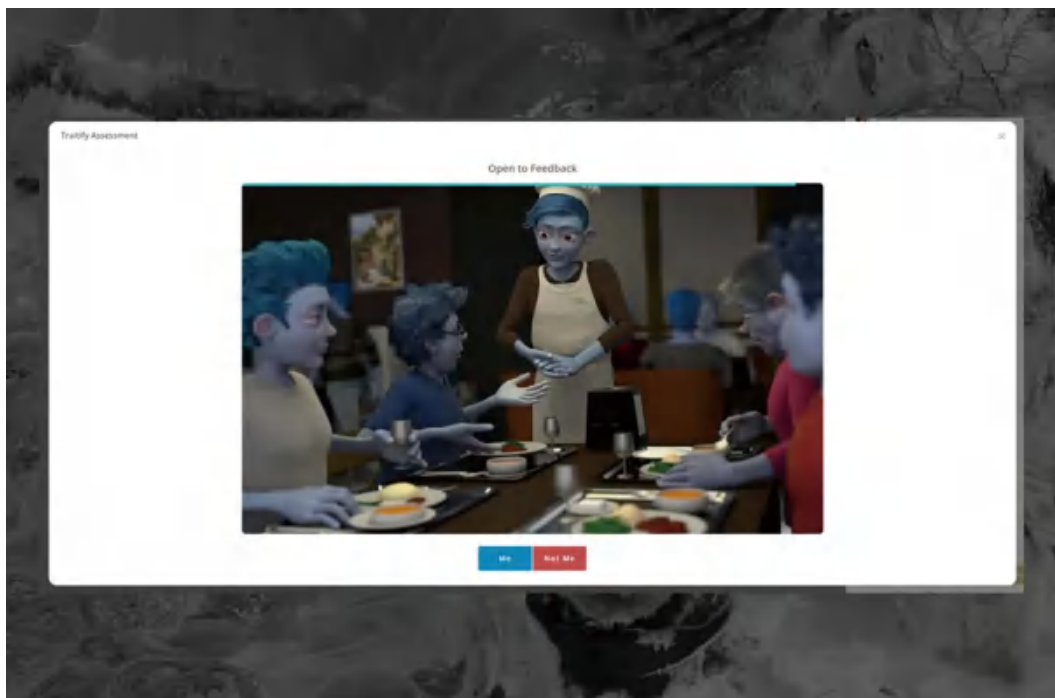
“肉丝 er”也还特别提到，眼神千万不要飘忽不定，有的 AI 能够根据面试者的眼神分辨其是否在读稿，一旦被判定为读稿，那么不管面试者说得多好都会被 pass 掉。这是因为有的 AI 面试系统有眼神追踪功能，四处乱瞟会被视为作弊。

“通过分析面试者的微表情，可以更准确地判断其是否在面试过程中存在作弊行为。”杨之贤表示，

“这是为了保证面试的公平性和诚信性。”

因此，除了专业能力，面试中的语速、情绪、肢体语言等都可能影响 AI 系统对面试者的评分。要知道，虽然 HR 可以查看面试视频，但大多数时候是根据 AI 的评分做初步筛选的。

另外，还有一些服务行业的公司会用 AI 进行性格测试。Paradox.ai 的性格测试在 Reddit 上多次疯传，联邦快递、麦当劳等公司都使用 Paradox.ai 公司的 AI 面试系统，通过“漫长而奇怪的性格测验”来招聘客户和食品服务工作人员，并附有“蓝色外星人”形象，目标是发现候选人在“亲和性”和“情绪稳定性”方面的排名。



AI 选不了 CEO

而对于研发这样的 AI 面试系统，最重要的是数据训练。

据杨之贤介绍，牛客采用了大模型和小模型的结合方式，自研了 AI-NowGPT 模型。为了保证“AI 考官”的公平和专业，牛客内部邀请了大量资深面试官对相同数据做人工标注，以确保 AI 考官在评分和评价方面与人类面试官保持一定程度的同步。

有的公司也会邀请客户企业参与标注，这样 AI 考官的评分标准会更加符合客户的需求。客户的参与可以帮助企业微调模型，使其更加符合特定岗位的要求。人类考官也可以对一部分或全部的面试结果进行复核，确保 AI 考官的判分结果的准确性和公正性。

像牛客这样还拥有笔面试 SaaS 产品的招聘网站，拥有专门的内容出题团队，借助海量的题库资源，还可以实时定制出题。

不过，杨之贤也表示，目前的 AI 面试还只是主要用在初步的大规模快速筛选上，对于后续的面试轮次，不同面试官关注的能力和业务知识可能存在差异，这是 AI 面试目前无法帮助解决的。“AI 选不了中高级管理层，也选不了 CEO。”

理想的情况是 AI 系统能够根据面试官的要求自动生成面试方案：面试官口述想要考核的能力，AI

快速提炼出相关考核点，并生成相应的面试问题和评估标准。这样，面试官就能更加专注在候选人的表现上，不需要花费过多精力在准备面试方案上。但凭现在 AI 的能力，在快速提炼考核内容方面还存在比较多困难。

此外，AI 面试系统还可以在面试过程中提供一些辅助功能，比如邀约和谈薪资：AI 可以自动发送邀约邮件或短信，并根据候选人的反馈和面试结果，提供薪资谈判的建议。

杨之贤表示，未来除面试外，AI-NowGPT 还将增加简历的点评、优化功能，同时提供准确的人岗匹配度评估，帮助企业找到岗位最合适的候选人。

然而，人类面试官的亲力和人际交往能力在面试过程中仍然非常重要。虽然 AI 可以提供面试方案和辅助功能，但在与候选人的互动和评估中，人类面试官的角色仍然不可或缺。他们可以通过面试过程中的非语言交流和深入提问，更好地了解候选人的能力和适应性。

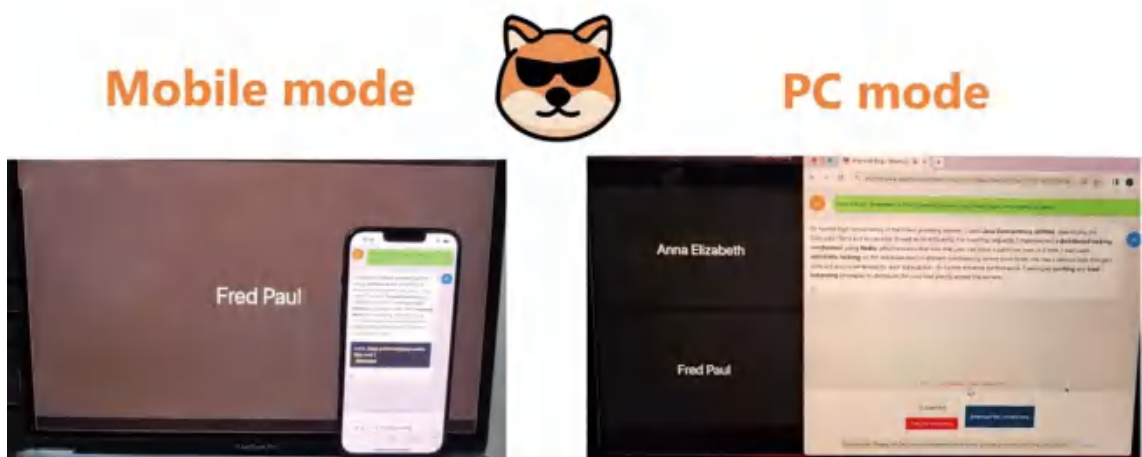
此外，电子前沿基金会社区组织副主任 Rory Mir 也指出，当前人工智能浪潮使用的是概率模型算法，这意味着它们只是依赖过去的模式做出预测。“问题是，过去数据的模式包括系统性偏见产生的模式。”

用 AI 打败 AI

当然，除了招人企业用 AI，面试者也会“用 AI 打败 AI”。

Interview Dog 是一款专门的 AI 面试辅助软件，

可以通过实时语音识别问题来帮助面试者回答考官的问题，支持科技、金融、工程、商业、法律等行业。Interview Dog 主打“按需使用”，五分钟免费试用之外，每分钟付费 0.45 美元。



为此，Interview Dog 在网上也得到了很多好评：“在我大脑短路时，Interview Dog 让我想起了一切”“有了它，谁还需要运气啊！”

但直接念答案也是有风险的，遇到自己不会的问题，“假装”思考后对答如流是会被 HR 怀疑的。

“我之前面了一个小女孩，答的太完美了。于是我钓鱼了几个问题，她说的和 GPT 的结果八成相似，给她上了点压力后，一个问题都答不出来了。”网友 momo 分享道。

对此，有的人认为这种做法并无不妥：“工作不就是借助各种工具完成任务吗？既然面试是模拟工作场景，那用不用 GPT 取决于面试者，只要结果出来就完事了。”但国内大多数企业是不太接受这种说法的，至少像算法思路这种应该是自己能回答的，否则就变成了纯粹的“工具人”。

小红书的一个博主发起了“面试应该允许使用 GPT 吗”的小调查，参与的 1207 个人里，47% 的人投给了可以，剩下的人还是认为不应该，“面试不能用 GPT 就跟考 GRE 不能查字典是一个道理。”

面试应该允许使用 GPT 吗？

47%
可以

52%
不行

已选择“不行”

1207 人参与

除了这种直接用来应对面试的工具，还有之前普遍被认为“水深”的简历优化。在猎聘等招聘网站上，一对一的简历优化收费达到了 398 元，但有网友

给出的评价是：花钱改简历≈抽奖，而且中奖几率极低！



猎聘网简历优化收费套餐

现在，有的应聘者会用 ChatGPT、文心一言等优化简历，还有人开发了基于 ChatGPT 的简历工具，用户可以直接使用 ChatGPT 自动修改。这种用法，基本不会有什么争议了。

未来，面试者和招聘者手里的魔法谁会被打败，似乎也是一个有趣的问题，毕竟 AI 不会站队。



扫码分享文章

AI 代码助手革新编程界：腾讯云专家汪晟杰 深度剖析机遇与挑战

嘉 宾 | 汪晟杰，腾讯云产品专家

编 辑 | 李忠良，InfoQ 编辑

“

代码大模型的出现极大提升了工程师的效率，但同时也带来了安全性与隐私性问题的挑战。如何应对这些挑战？有哪些最佳实践可以帮助企业在利用这些 AI 工具时确保代码安全和隐私保护？软件开发开发者应该如何准备和适应这种由 AI 带来的变

革？AI 工具接管部分编程任务后，开发者的角色又会发生哪些实际变化？

我们对腾讯云产品专家汪晟杰进行了访谈，以下为访谈实录：

AI人工智能产业链联盟

#每日为你摘取最重要的商业新闻#

更新 · 更快 · 更精彩



Zero

AI音乐创作人

水墨动漫联盟创始人

百脑共创联合创始人

人工智能产业链联盟创始人

中关村人才协会秘书长助理

河北北大企业家分会秘书长

墨攻星辰智能科技有限公司CEO

河北清华发展研究院智能机器人中心线上负责人

中关村人才协会数字体育与电子竞技专委会秘书长助理



主要业务:AI商业化答疑及课程应用场景探索, 各类AI产品学习手册, 答疑及课程



欢迎扫码交流

提供: 学习手册/工具/资源链接/商业化案例/
行业报告/行业最新资讯及动态



人工智能产业链联盟创始人

邀请你加入星球, 一起学习

人工智能产业链联盟报 告库



星主: 人工智能产业链联盟创始人

每天仅需0.5元, 即可拥有以下福利!
每周更新各类机构的最新研究成果。立志将人工智能产业链联盟打造成市面上最全的AI研究资料库, 覆盖券商、产业公司、科研院所等...

知识星球

微信扫码加入星球 ▶



技术有效性和限制

InfoQ: 您如何评价当前 AI 代码助手如 GitHub Copilot 在理解复杂代码结构和项目架构方面的能力?

汪晟杰: 当前的 AI 代码助手, 如 GitHub Copilot, 以及腾讯云 AI 代码助手, 都展示了在理解复杂代码结构和项目架构方面的显著进步。他们都有着如下优点:

- 在编写简单到中等复杂度的代码时, 它们可以提供有用的代码建议和补全, 从而提高开发者的代码生产力。
- 通过分析大量的开源代码库, 它们可以学习到许多编程语言和框架的最佳实践。
- 对于某些常见的编程任务, 它们可以生成准确的代码片段, 减少开发者的工作量。

然而面临着成本和速度的权衡, 以及如何塞下整个工程代码上下文来理解工程。譬如对于非常复杂的代码结构、大仓或者多仓的项目重度依赖的情况, AI 代码助手可能无法完全理解其逻辑和设计, 导致生成的代码片段不准确或不适用。最近 GitHub Copilot 的企业版的知识库可以对项目工程做 Indexing+Embedding, 可以大大强化本地开发并享用远端向量, 从而提升对于工程理解的提问和回答。这块我将在本次分享中重点和大家分享。

InfoQ: 针对多文件和大型项目, 这些工具在理解上下文和逻辑关系方面表现如何?

汪晟杰: 在补全场景下, 对于常见的编程模式和结构, AI 代码助手通过语法分析等多种策略, 可以较好地识别和理解多文件之间的关系。比如你用了工厂单例模式构造一个对象, 在调用上就知道我这个对象要用到工厂类。在 GitHub Copilot 实战中, 需要打开相关的文件。在腾讯云 AI 助手上, 我们采用了快速的语法树能力快速找到相关文件引入提示词, 从而让大模型能感知到更多上下文。通过分析大量的开源代码库, 它们可以学习到许多编程语言和框架的最佳实践, 从而在一定程度上理解大型项目的结构和组织方式。

InfoQ: 在使用如 CoT 和 RAG 这类技术时, 有哪些明显的优势和存在的局限性?

汪晟杰: CoT (Chain of Thought), 本质上是对于提问者的问题的思维链的拆解, 并逐步去解决子任务的要求并合并成最终答案。首先在上下文理解上: CoT 有效的通过 Multi-Agent 方式, 来拆解并安排下一轮的子任务, 同时可以通过 RAG 进行代码推理, 从而生成更符合需求的代码。

其次, 有高质量的代码生成: 通过模拟人类程序员的思考过程, 自主获得并进行下一轮的执行,

可以选择不同模型、或者 Function call 来调用业务函数，或者通过上下文找到问题出错点并在下一轮进行修复方案。RAG 则是保障了项目代码扩展理解能力。

InfoQ：您认为未来这些工具需要哪些改进才能更好地支持复杂的软件开发任务？

汪晟杰：当前的 AI 代码助手，如 GitHub Copilot，已经在简化软件开发任务方面取得了显著的进步。然而，要更好地支持复杂的软件开发任务，未来这些工具可能需要以下几方面的改进：

- 更好的上下文理解：AI 代码助手需要更好地理解项目的上下文，包括项目的目标、架构、已有代码的功能等。这可以通过更先进的自然语言处理和代码分析技术来实现。

- 更强更快的代码推理能力：对于复杂的代码逻辑和算法，AI 代码助手需要有较强的推理能力，以生成正确和高效的代码。这可能需要更先进的机器学习模型和算法。

- 更全面更深地集成 IDE：AI 代码助手需要更广泛的支持主流的 IDE，并深入地集成到集成开发环境 (IDE) 中，以提供更流畅和无缝的用户体验。这可能包括更好的代码提示、实时错误检测、代码重构建议等功能。

- 更全面的编程语言和框架支持：AI 代码助手需要支持更多的编程语言和框架，以满足不同开发者的需求。这可能需要分析和学习更多的开源代码库。

- 更高的安全性和可靠性：AI 代码助手需要在生成的代码中考虑到安全性和可靠性，避免引入潜在的安全风险和错误。

安全性和隐私问题

InfoQ：在使用 AI 编程助手时，如何处理和保护敏感和私有的代码数据？

汪晟杰：有以下六个方面值得考虑。

- 选择可信赖的 AI 编程助手：在选择 AI 编程助手时，选择那些来自可信赖来源、有良好声誉

的工具，这些工具通常会遵循严格的数据保护政策和安全实践。我也建议不要把核心代码用 GitHub Copilot 去生成，因为你的代码上下文是直接经过他们海外服务器。

- 了解数据保护政策：在使用 AI 编程助手之前，详细了解其数据保护政策和隐私条款。确保这些政策符合您对数据保护的要求，特别是关于数据

的收集、处理和存储方面。

- 是否提供安全私有化能力：在银行等领域腾讯云积累了很多客户实践。我们一键部署升级，并在封闭的环境、信创环境下都有着不错的客户反馈。对于技术对话解决了在不可上网的环境下，搜索技术问题找寻答案的另一种安全方法。
- 遵循最佳实践：在编写代码时，遵循最佳实践，将敏感信息(如密码、API 密钥等)从代码中分离。将这些敏感信息存储在安全的配置文件或环境变量中，而不是直接嵌入到代码中。
- 限制访问权限：确保 AI 编程助手仅能访问其需要的最小权限。例如，可以限制其访问特定的代码库、分支或文件夹，以减少潜在的风险。
- 监控和审计：定期监控和审计 AI 编程助手的使用情况，确保其符合您的安全和合规要求。如果发现任何异常行为，立即采取相应的措施。

InfoQ：您如何看待这些工具在训练过程中可能出现的数据泄露风险？

汪晟杰：首先，AI 编程助手通常使用大量的开源代码库进行训练。虽然这些代码库本身是公开的，但在训练过程中可能会捕获到一些敏感信息，

如 API 密钥、密码等。因此，训练过程中需要对这些潜在的敏感信息进行清理和过滤；其次，由于 AI 模型在训练过程中可能会学习到一些敏感信息，因此在使用模型生成代码时，有可能泄露这些信息。应用端需要针对这类问题，采用技术手段，以增加兜底逻辑，即模型训练过程中数据的隐私问题，可以由应用端做针对性的过滤。最后，用户教育和意识：对于使用 AI 编程助手的开发者，提供培训和意识教育，以确保他们了解如何在使用这些工具时保护敏感和私有的代码数据。这包括遵循最佳实践，将敏感信息从代码中分离等。

InfoQ：有哪些最佳实践可以帮助企业在利用这些 AI 工具时确保代码安全和隐私保护？

汪晟杰：一方面为用户开发习惯，在让模型基于上下文推理的时候，他会模仿你的习惯，所以将敏感信息从代码中分离，在代码库中引入代码扫描，实时监听代码生成质量。另一方面是给予仓库代码更小范围，比如我只需要把主要描述的 README 文件、接口文档、核心代码的实现类等作为 RAG 的来源，或者在补全上找到核心调用链的相关函数及文件。

对开发者角色的影响

InfoQ: AI 工具在接管一些编程任务后, 您观察到开发者的角色有哪些实际变化?

汪晟杰: 有三方面的影响, 首先是更高层次的抽象: 开发者可能会从处理底层代码转向处理更高级别的抽象, 例如设计软件架构、优化数据结构和算法等。这将使 AI 代码助手能够更有效地理解并模仿生成; 其次是更全面的技术点: 有了 AI 助手后, 后端也会写前端代码, 在做一些短平快的项目时, 一个产品和一个技术可以分工完成, 相比之前的开发效率是大大提升; 最后当然是开发习惯的变化: 以 IDE 为平台, 以 AI 为内核, 以对话为切入, 以编码质量为验收, 会是开发者在日常编码中的另一个自己的「数字人」

InfoQ: 这些变化对开发团队的结构和工作流程有何影响?

汪晟杰: 我认为团队会更扁平, 技术同学也不会再抗拒新的某种技术和语言。上手门槛变低了, 获取知识的速度提高了, 解决问题的方式多样化了。在工作流程中, 学习提示词, 摸透大模型的习性, 会是工作中不可缺少的一部分。逐步上手后, 会产生极大粘性。腾讯内部我们的产品的留存率是非常高的。

InfoQ: 您认为 AI 工具将如何影响软件开发行业

的就业趋势?

汪晟杰: 大概有以下几个方面。

- 自动化低级任务: AI 工具可以自动化许多重复性和低级别的编程任务, 如 CRUD 的代码生成、SQL injection 错误检测和修复等。这可能导致对于那些主要从事这些任务的初级开发人员的需求减少。
- 提高生产力: 通过自动化一些任务, AI 工具可以提高开发者的生产力。这意味着开发团队可能需要更少的人员来完成相同的工作量。然而, 这也可能导致对高技能开发人员的需求增加, 因为他们可以更好地利用这些工具。
- AI 化转型和咨询: 随着 AI 工具的普及, 软件开发人员可能需要学习新技能和知识, 以适应不断变化的技术环境。这可能包括学习如何与 AI 工具合作, 以及掌握新的编程范式和技术。
- AI 产品化的创新: 随着 AI 工具接管一些基本任务, 开发者可以将更多精力投入到创新和创意上。这可能导致对具有创新思维和能够开发新产品和服务的开发人员的需求增加。
- 与大模型及算法的紧密合作: AI 工具的发展可能导致业务要与大模型及算法团队的合作更加紧密。
- 新的就业机会: 虽然 AI 工具可能导致某些角色的需求减少, 但它们也可能创造新的就业机会。

例如，随着 AI 技术的发展，可能会出现新的专业领域，如 AI 伦理、AI 系统监管等。

总之，AI 工具将对软件开发行业的就业趋势产生深远影响。虽然某些角色可能受到冲击，但整体上，对具有创新思维、高技能和跨领域知识的软件开发人员的需求可能会增加。为了适应这些变化，开发人员需要不断学习和更新技能，以保持行业中的竞争力。

InfoQ：对于软件开发者来说，他们应该如何准备和适应这种由 AI 带来的变革？

汪晟杰：首先学习 AI 和机器学习基础知识：开

发者应掌握 AI 的基本概念、原理和技术，了解机器学习算法和数据科学库（如 TensorFlow、PyTorch 等），这将有助于他们在开发过程中更好地利用 AI 技术；其次，关注 AI 领域的最新发展：关注 AI 领域的最新研究成果和行业动态，了解 AI 技术在各个行业的应用案例，以便了解哪些技术可以应用到自己的项目中；当然，提高编程技能也不可或缺：AI 技术的发展对开发者的编程能力提出了更高的要求，因此开发者需要不断提高自己的编程技能，熟悉各种编程语言和框架，如 Python、Java、C++ 等；最后是学会与 AI 合作：开发者需要学会如何与 AI 系统合作，理解 AI 系统的优势和局限性，以便在开发过程中充分发挥 AI 的潜力。

嘉宾介绍

汪晟杰，腾讯云 产品专家，历任阿里高级技术专家，从事钉钉云效核心业务线、Teambition 合伙人、Autodesk 首席软件架构师、十多年 SAP 云平台、SuccessFactors HCM、Sybase 数据库、PowerDesigner 等产品的开发经理，在软件架构设计、产品管理和项目工程管理、团队敏捷提效等方面拥有近 20 年的经验。



扫码分享文章

数字人才的移动知识库

极客时间是数字人才的在线学习平台，通过 PGC 专业内容生产模式，由极客时间教研团队与一线技术专家共同打造，涵盖 70+ 类技术领域。平台提供沉浸式深度学习功能，并通过社群、直播、部落等多种方式，让用户可以与行业大牛零距离交流学习。



扫码关注
极客时间官方公众号

300 +

精品体系课
名师出高徒

1700 +

技术案例视频
聚焦解决业务难题

17 条

学习路径
系统性学习进阶

5700 +

小时课程
为用户打造随身图书馆



连接用户和专家的桥梁

极客时间提供了留言、社群、直播等方式，为用户和专家建立了紧密的连接。用户可以与专家交流学习心得，探讨技术问题，获取成长的最佳路径。



沉浸式学习体验

精心设计的文章排版，个性化的阅读风格，为你打造沉浸式学习体验。你可以使用划线、评论、收藏的功能进行深度学习，也可以在碎片时间听音频，看视频。



技术领域广泛覆盖

课程涵盖大前端、后端、架构、大数据、运维、人工智能、测试、产品、管理、计算机专业课，全面覆盖 IT 从业者全职业生命周期知识技术图谱，帮助用户全方位系统化提升工作能力。



学习反馈和成果奖励

用户通过学习统计功能查看自己的学习轨迹和学习进度，可以参加作业和考试，验证自己的学习成果。学完一门课程可以获得平台颁发的电子结课证书和结课奖励金。

第三章：垂直行业

巨头们涌入的医疗大模型，何时迎来最好的商业时代？

嘉 宾 | 刘升平，云知声 AI Labs 研发副总裁

作 者 | 高卫华，InfoQ 编辑

“

当下极为火爆的大模型，在医疗赛道同样炙手可热。谷歌刚刚发布了准确率达 91.1%、性能远超 GPT-4 系列的多模态医学大模型 Med-Gemini，国内市场亦很热闹。自 2023 年以来，百度、腾讯、京东等诸多大厂都相继加码医疗大模型领域，与医疗相关的大模型产品和应用如雨后春笋般正不断涌现出来，其中更不乏 AI 和医疗企业的手笔。

目前，已有部分医疗大模型产品投入到导诊、预问诊等医院场景中。然而，医疗大模型虽有一定潜力，但现阶段仍有不少要跨越的落地门槛。

为此，InfoQ 对云知声 AI Labs 研发副总裁刘升平进行了专访，听他聊一聊现阶段医疗大模型的商业化能力，以及面向这类应用场景的行业大模型该如何定制优化。以下为访谈实录，经编辑。

医疗大模型距离商业化有多远？

InfoQ：现阶段，医疗大模型要规模化落地还面临哪些现实问题？

刘升平：主要的问题还是有不少，首先是医生和患者的接受度，特别是有些场景要改变医生的使用习惯。还有一个问题是大模型的部署成本，如果在院里大规模并发使用医疗大模型，硬件成本会比较高。

InfoQ：“幻觉”的偶发出现是大模型目前公认的一个问题，医疗场景对准确度要求会更高，山海在这方面是怎么做的？

刘升平：“幻觉”的确是核心要解决的问题，我们采用多种手段从多方面降低幻觉，包括保证医疗预训练语料和微调数据的质量和多样性、采用能降低知识幻觉的解码策略、融合医疗知识图谱的知识增强大模型技术、医疗知识检索增强、大模型结果后校验、大模型输出置信度评估等。

InfoQ：您认为哪一个评价标准最能代表医疗大模型的水平？

刘升平：临床有效性是最能代表医疗大模型水平的关键评价标准，包括模型在实际临床环境中的诊断准确性、治疗建议的合理性以及与专业医生的决策一致性。此外，模型的鲁棒性、泛化能力、

可解释性、用户友好性、数据隐私保护以及合规性也是重要的评价维度。然而，临床有效性直接关系到患者的安全和健康，因此如果把医疗大模型应用与临床实践中，它可能是最重要的评价标准。

InfoQ：现在行业内有您认为还不错的其他医疗大模型产品吗？国内外均可。

刘升平：除了云知声的山海大模型医疗版，最近看到的是谷歌的多模态医疗大模型 Med-Gemini，在多项临床任务评测中都表现很好，但还没有在医院得到广泛使用。

InfoQ：在医疗大模型的技术实现、应用效果以及成本部署上，国内与国外有区别吗？

刘升平：没有显著区别。

InfoQ：您认为医疗大模型真正迎来商业化时代还需要多久？

刘升平：预计 2-5 年吧。今年是医疗大模型的应用元年，有部分医院开始尝试一些医疗大模型的应用，随着这些医院推广与积累医疗大模型应用经验，预计医疗大模型会在 2-5 年内进入更广泛的商业化阶段。

InfoQ：社会接受度上，如何让大众认可大模型的诊断或治疗方案？

刘升平：要让大众接受并信任大模型的诊断或治疗方案，是一个长期的过程，要考虑很多方面。第一，要提高模型的决策过程透明度，提供可解释的输出，让用户理解模型是如何得出结论的。这有助于建立用户信任，尤其是对于医疗决策这样敏感的问题。第二，要有严格的临床试验，证

明模型的诊断或治疗方案与专业医生的判断相当或更优，且这些结果应由独立的第三方机构审核并公开。第三，要让医生参与到模型的开发和应用中，他们可以提供专业知识，确保模型的输出符合医学实践，并在实际应用中监督和调整。第四，要开展公众教育活动，解释人工智能在医疗领域的潜力和限制，消除误解，提高公众的理解和接受度。通过这些措施，应该可以逐步提高社会对大模型在医疗领域应用的接受度和信任度。

山海大模型的实践经验

InfoQ：医疗相比其他场景更复杂且严谨，难度自然也不小，驱动云知声选择在这一领域开发大模型的最重要因素是什么？

刘升平：云知声选择在医疗领域开发大模型，主要有两个关键因素。一是应用潜力，而医疗领域是一个富文本、富知识的行业，并且医疗大模型在处理医疗病历文书、辅助诊断、药物研发等方面展现出巨大潜力，因为医疗领域是一个很适合大语言模型技术的应用领域。此外，医疗 AI 市场具有巨大的商业价值，随着技术的成熟和接受度的提高，未来有望形成规模化的商业模式。二是专业积累，云知声深耕医疗领域多年，对医疗业务场景有深入的理解，在医疗数据和医疗 AI 技术有深厚的积累，也积累了数百家的医疗客户，

这有助于医疗大模型的研发和商业化推广应用。

InfoQ：大模型训练过程本身就对数据质量有较高要求，医疗领域的数据则更为特殊，还具有隐私保护、专业知识复杂、经验化知识难以结构化等难题，山海是如何克服的？

刘升平：山海医疗大模型在训练过程中面临数据质量、隐私保护和专业知识复杂性等挑战，我们采取了两种策略来克服这些问题。一是数据清洗与预处理，对收集到的医疗数据进行严格的清洗，去除噪声和不一致的信息，确保数据的准确性和一致性；同时使用专业的医疗知识进行预处理，如标准化术语等。二是匿名化与脱敏，在遵守相关法规的前提下，对个人健康信息进行匿名化和

脱敏处理，以保护患者隐私。

InfoQ：使用开源数据集可能出现产品同质化现象，山海在数据资源方面是如何使用的？

刘升平：云知声在开发山海医疗大模型时，采取了多种策略来避免产品同质化，确保模型的竞争力。第一，我们使用了不少专有数据集，即云知声多年的医疗业务积累的大量内部医疗数据。这些专有数据可以提高大模型在特定场景的应用效果。第二，我们采用了一些数据增强技术来自动生成训练数据，例如，通过数据合成、噪声注入、标签变换等技术，增加数据的多样性和复杂性，使模型在不同条件下表现更为全面和鲁棒。第三，我们还与医疗专家合作来确保医疗数据的准确性和专业性，同时利用专家的知识来指导数据的预处理和标注。通过这些策略，云知声的山海医疗大模型能够与只使用开源数据集训练的大模型有显著区别，并且在面向具体的医疗场景应用时有更好的效果。

InfoQ：云知声的山海医疗大模型主要做了哪些场景？目前哪个场景的应用率最高？哪个场景能算作山海的“杀手锏”？

刘升平：对于云知声的山海医疗大模型，主要做了以下场景：

- 病历生成：包括基于医患对话的门诊病历和出院小结、手术记录生成等住院病历的生成，以及放射科报告生成等医技科报告。
- 病历质控：对住院病历（包括病案首页）做过

程和终末质控，支持 1000+ 形式和内涵质控点，大幅提高病历的质量。

- 单病种上报：对国家卫健委要求的 57 个病种做自动数据汇集及上报。
- 医保控费：按照医保局的规范，监管医院的临床诊疗行为和收费合理性，确保医疗费用的合规。
- 保险理赔的医疗审核：审核在保险理赔中涉及到的医疗费用，剔除不合理费用。
- 专病库平台：将病历等临床数据自动抽取和导入到专病库。
- 智能问诊：作为 AI 医生，与患者进行对话，收集症状，并提供初步的健康咨询和建议。

目前，山海应用率最高的场景是病历生成、病历质控和保险理赔的医疗审核。结合云知声在语音技术上强项开发出的门诊病历生成系统，结合云知声在医疗知识图谱的积累开发的病历质控系统 and 保险理赔医疗审核系统均可以视为“杀手锏”场景。

InfoQ：针对于山海医疗大模型，您更推荐医疗机构采用哪种部署方式落地？具体是如何考虑的？

刘升平：云知声的山海医疗大模型在医疗机构的部署通常有以下两种方式：云端部署和私有化部署。至于选择哪种部署方式，主要考虑几个因素吧。一是如果医疗机构对数据安全有极高要求，那就倾向于私有化部署。二是考虑成本与资源，云端部署通常成本较低；私有化部署初期投入大，但长期运营成本可能更低。

InfoQ：现在市面上的医疗大模型不少，国内有

许多大厂也在做，山海的独特之处是什么？

刘升平：这和云知声做医疗大模型的动机是一样的，山海医疗大模型的独特之处主要有两点。一是在专业领域深度方面，云知声专注于医疗领域，有深厚的数据、知识、场景和客户积累，这使得

山海医疗大模型在效果上业内领先，目前在医疗大模型综合评测 PromptCBLUE 和 MedBench 上都是排名第一。二是在技术融合方面，结合云知声在语音识别和医疗知识图谱技术的专长，山海医疗大模型在语音交互式医疗应用上具有优势，且在临床应用上的医疗知识幻觉也大为减少。

嘉宾介绍

刘升平，云知声 AI Labs 研发副总裁，北京大学数学学院博士毕业，是前 IBM 中国研究院资深研究员，中文信息学会语言与知识计算专委会委员。曾在语义网，机器学习、信息检索，医学信息学，自然语言处理等领域发表过数十篇学术论文和国际国内发明专利。在 IBM 中国研究院信息与知识组工作期间，刘博士主要负责语义技术及其应用的研发，曾多次获得过 IBM 研究成就奖。2012 年底，刘博士加入云知声 AI Labs，领导认知智能团队，负责大语言模型、知识图谱和智慧医疗等方面的研发及管理工作。在云知声期间，主持研发了山海大模型，获得国内外 AI 评测冠亚军 13 个，获得北京市科技进步奖一等奖一项。



扫码分享文章

AI 老师的强大功能 + 真人老师的情感交流 = 未来教育?

演 讲 | 郭家, 图灵机器人 COO

编 辑 | 傅宇琪, InfoQ 见习编辑

褚杏娟, InfoQ 编辑

策 划 | 蔡芳芳, InfoQ 主编



人工智能正在深度重塑教育领域, 驱动着教学模式, 尤其是个性化学习的革新。作为一家以语义和对话技术为核心的人工智能公司, 图灵机器人用高精度 AI 知识问答、中英文语法纠错、图文识别等技术为教育行业赋能。自 2023 年起, 图灵机器人用大模型逐一替代了 CNN 模型, 并创新了 AI 口语老师、阅卷 AI 助理等应用, 在步步高、作业帮等产品上应用上线并取得不错效果。

在用大模型重构产品的 1 年时间里, 该公司对面向成本设计产品、大模型的“能与不能”都有了深度思考。本文整理自图灵机器人 COO 郭家在 QCon 2024 北京的演讲分享“教育大模型, 说你行你才行”, 拆解这段产品重构之路, 并以实际案例, 分享其中的辛酸苦辣。

本文由 InfoQ 整理, 经郭家老师授权发布。以下为演讲实录。

我们是谁

图灵机器人公司专注于教育行业，已经发展了将近 15 年。在这个过程中，我们见证了许多变化，并从传统模型逐步进化到大模型。公司的 LOGO 是对图灵机器人的致敬，我们于 2017 年获得了图灵后人詹姆斯·图灵以及英国皇家社会协会的肖像授权。2019 年，我们还成为了图灵基金在中国的唯—合作伙伴。由于公司注册较早，图灵现在已成为专有名词，无法再次注册。

我们的团队成员大多来自交大系。我们的 CEO 是交大数学系毕业，一直从事人工智能和复杂决

策系统的工作，CTO 老韦也是交大数学系出身，首席科学家何小坤曾是好未来 AI lab 的负责人，在双减政策实施后来到我们这家人工智能教育公司，石勇教授是中科院的合伙科学家。

我们的投资机构特色鲜明，全部是战略投资人。他们对公司的持续经营和帮助已经持续多年，也不急于退出。我们的天使投资人是赛富的创始合伙人羊东。我们还是微软在中国的第一家创投企业。此外，我们的股东还包括 HTC、奥飞动漫和洪恩教育。

“烙印”：AI是团队的内核

团队

| | | | | |
|---|---|---|---|---|
|  俞志晨 CEO 模糊数学，决策学，复杂系统方向，师承贺仲雄 |  韦克礼 CTO 北交大数学系 原移动音乐搜索负责人 |  何小坤 AI科学家 极北理工AI硕士 原百度&好未来视觉AI负责人 |  石勇教授 首席AI技术顾问 国务院参事 中科院数理科 学研究中心主任 |  郭家 COO 北交大数学系 增就职于 AMDOCS、朗新中国、华为 |
|---|---|---|---|---|

投资人

| | | | | |
|---|---|--|--|---|
|  羊东 原软银董事 |  微软创投 |  hTC |  奥飞动漫 |  洪恩 |
|---|---|--|--|---|

公司上一次推出的 AI 产品名为虫洞语音助手，对于互联网的资深用户来说，可能对这款产品有所耳闻。我们从 2010 年开始研发并发布了这款产品，它最初是为塞班手机和黑莓手机设计的语音助手。当时，苹果公司尚未收购 Siri。随着苹果在 iPhone 4 发布期间推出 Siri，语音助手这一领域迅速变得热门，我们的用户数量也迅速增长，接近 2000 万。

在开发过程中，我们一方面专注于自己的产品，另一方面与 HTC 建立了合作关系。HTC 是安卓系统的第一款手机制造商。我们与 HTC 合作开发了小 hi 机器人，也就是小 hi 语音助手。该产品上线时拥有 100 多种虚拟人表情，400 多种技能，包括 200 多个 CP 和 SP 的接入。

我们的许多技能都是与后方的 CP 和 SP 合作实现的，例如，查询天气功能与中国天气网合作，餐饮推荐则与点评网站合作。然而，尽管用户基数庞大，语音助手的前期活跃度也不错，但将其商业化却非常困难。直到现在，手机上的语音助手仍然面临这一问题。因此，面向消费者的业务模式（to C）并不适合当时的产品。基于这一认识，我们决定将这个创业项目出售给 HTC。随后，我们开始了第二次创业。

第二次创业，我们转向了 AI To B 业务，即面向企业的人工智能服务。2014 年，我们将产品卖给 HTC 后，决定将这些技术转化为一个开放平台，主要面向开发者开放。平台吸引了超过 100 万的开发者，每天都有上百的开发者加入，他们主要利用以自然语言处理（NLP）为核心的语音

助手相关产品。

2016 年，我们发现对于一家创业公司来说，儿童教育是一个需求量大、适合快速增长的领域，于是开始专注于教育领域。在 2017 年和 2018 年，我们有幸邀请到了包括我的师妹，MIT 博士贾梓筠在内的人才，一起参与这个项目，那年公司业务突破 1000 万营收。到了 2019 年，我们开始将视觉技术纳入我们的产品和服务。在教育领域，视觉技术的需求甚至超过了语音技术，例如题目识别、图片和文字识别、绘本和图画识别等，这些都需要计算机视觉（CV）技术来完成。

公司有五条主要的业务线。首先，进校业务方面，我们正在开发中高考英语口语模考系统，这种口语模考系统特别适合利用大模型技术。我们有教案的 AIGC 助手，它帮助老师生成教案，可以插入图片或精彩案例，甚至可以适时地加入一些幽默段子，让课程更加生动有趣。我们还提供大模型实验课，让学生亲自操作，测试 prompt，并使用 RAG 工具进行训练。

在出版领域，我们主要面向教辅公司和出版社，提供 AI 英语出题、AIGC 动画课等服务。此外，我们还涉足古籍、古典和学术研究领域，同样利用 RAG 技术进行数据挖掘。

运营业务方面，我们提供 4G 电子产品，如自动翻译扫描笔、能够识别绘本和教材的台灯，以及用于口语测评方案的学生证和学生卡。

电教领域是我们公司历史最悠久、壁垒最深厚的

业务之一，市场份额高达 80%。在这个领域，我们提供的服务包括语音助手、口语老师、作文批改以及翻译相关算法，如指尖翻译、手写体翻译和印刷体翻译。

最后，在汽车领域，我们为儿童领域提供重要的

平台。从去年开始，新能源汽车如理想汽车推出了“小主人模式”，后排的小主人座舱需要语音助手来承载趣味内容和知识性互动。我们配套的小主人语音助手，包括音乐版权、分级阅读版权和词典版权，为儿童提供丰富的车内互动体验。

“五个业务线”

| | | | | |
|---|---|---|---|---|
|  |  |  |  |  |
| 进校 | 出版 | 运营商 | 电教 | 汽车 |
| 中高考英语口语模考系统 教案AIGC助手 大模型实验课 | AI英语出题 AIGC动画课 RAG学术研究 | 4G电子产品ODM 绘本翻指读 发音和语法评测 | 大模型语音助手 大模型口语老师 大模型作文批改 指尖翻译 | 小主人语音助手 音乐版权 分级阅读版权 词典版权 |

南京讯飞
XINFEI

大模型产品的第一步是 Cost Down

■ 相比小模型时代，成本是做大模型的新主题

去年公司正面临大模型带来的成本压力。我们已将许多算法商业化多年，但随着时代的发展，

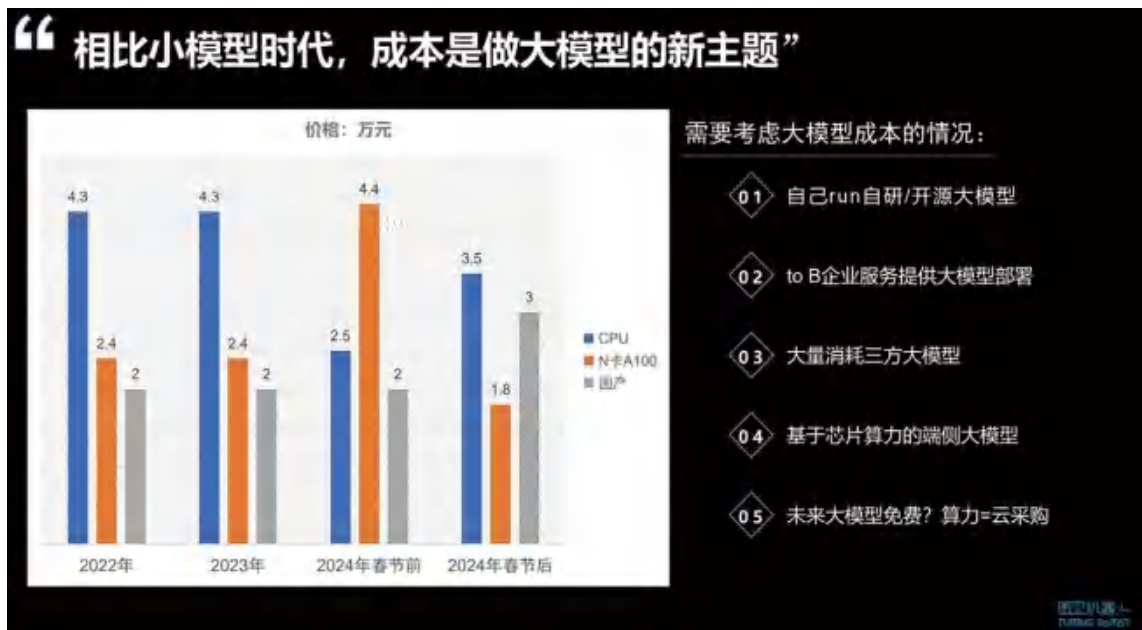
如果不追求大模型的发展，否则就可能被时代淘汰。要追赶大模型，我们需要考虑如何将旧算法

与大模型过渡。直接将大模型引入市场，初期成本非常高。尽管图灵公司自我造血多年，但大模型的投入仍然巨大。有下述几种情况需要考虑降低成本：

1. 自己研发或使用开源的大模型，这对算力要求很高，所有资源都需要自己提供。
2. 为企业提供大模型服务，如进校或教育部的大模型私有化部署，学校对数据安全和隐私有严格要求，不希望竞争对手获取他们的原创内容，因此要求大模型必须私有化部署并本地训练。
3. 大量使用第三方大模型，如按 tokens 结算的方式，初期试用成本可控，但一旦商业化，成本迅速上升，如我们之前使用 GPT 大模型接口，每

月投入可达三四十万，对单个客户而言，一年几百万的成本难以承受。

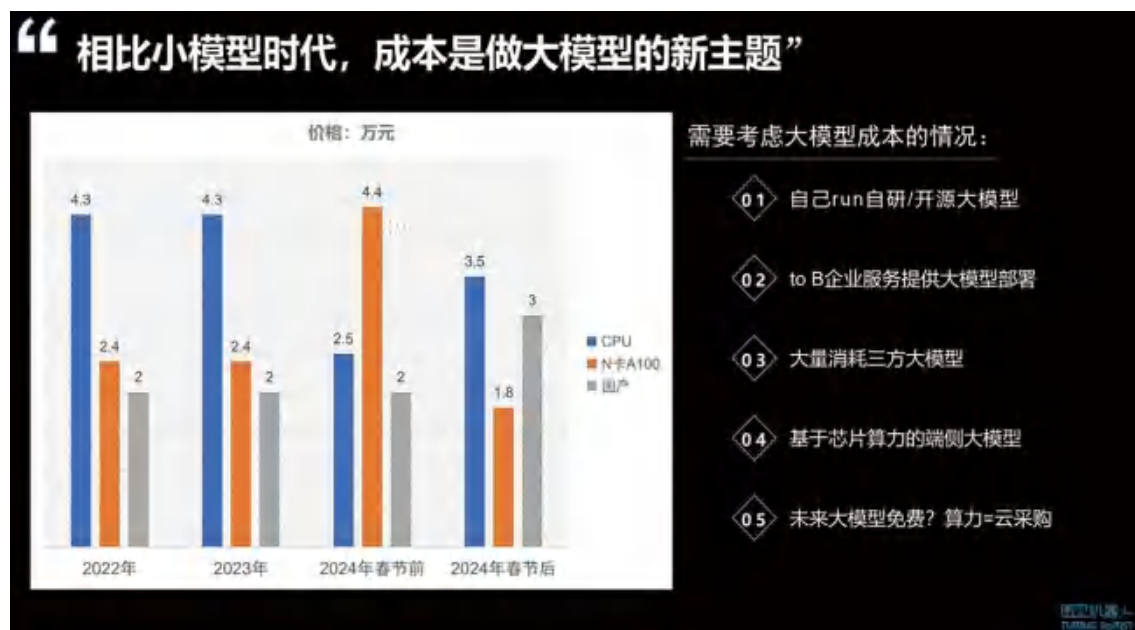
4. 端侧芯片层的大模型运行，如高通在最新芯片上运行大模型，预示着未来手机等设备将有本地大模型支持。
5. 开源大模型的趋势，如通义、百川等公司开源大模型，目的是让更多人使用，甚至自己运行大模型，从而推动云服务的销售。未来，购买算力可能等同于购买云资源。此外，服务器情况有所变化。2023 年相比 2022 年，价格明显上涨超过 50%。2023 年 5 月的禁令前后价格也有所不同。但在 2024 年，云服务价格下降了约 20%，目前云算力和消耗量处于可控范围内，这与服务器资源逐渐变得更加充裕有关。



■ 我们如何做大模型降本

我们的产品图灵 AI 口语老师已经推出了三个版本。C 版本是我们利用大模型技术所开发的版本，它在资源消耗方面是三个版本中最低的。右侧的

图表展示了我们对成本的测算，这意味着，通过采用大模型技术，我们能够在保持产品质量的同时，有效控制成本。



C 版本口语老师用于在创作话题时，生成 AB 角的对话场景。生成对话后，系统会基于预设的预训练脚本来执行对话，重点在于发音的评测，而非表达的正确性。

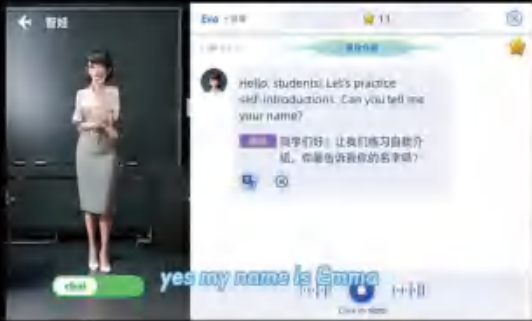
B 版本的口语老师在用户每次提问时都会调用大模型进行多种识别，包括语法、地道表达、对话相关性以及句子润色等，因此大模型的调用量非

常大，消耗量级也随之增加。

我们制作的大多数儿童产品的成本相对较低，可能只有几百元，甚至一百元以内。因此，在儿童电子产品上，大模型的成本是相当高的，难以承受。我们尝试了多种运营方法来进行二次转化，以降低成本。

“图灵口语老师成本结构”：B版

大模型调用功能：语法纠错、地道表达、对话相关性、句子润色




| 成本来源 | 元/台 |
|------------|------|
| 1 试用期调用大模型 | 5.0 |
| 2 会员期调用大模型 | 60.0 |
| 3 小模型AI调用 | 1.0 |
| 4 其他服务器消耗 | =0 |
| 总计 | 66 |

A 版本口语老师的最大特点是教案虚拟人。虚拟人如何表达得好，关键在于情感识别。我们最初展示的口语老师形象被孩子们吐槽，因为许多学生认为这位老师给人一种压迫感，不想与其对话

交流。因此，我们后来采用了更多二次元、卡通的形象。这里增加了两个成本，一是虚拟人的调用成本，二是大模型中虚拟人的情感识别成本。

“图灵口语老师成本结构”：A版

大模型调用功能：语法纠错、地道表达、对话相关性、句子润色、情感识别



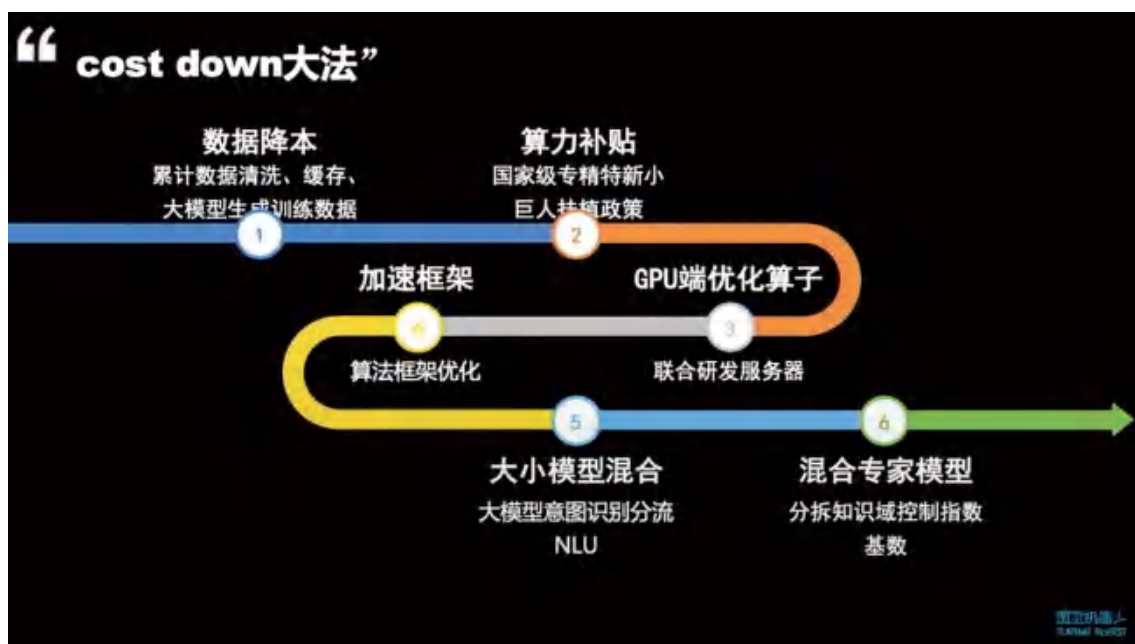
| 成本来源 | 元/台 |
|--------------|------|
| 1 试用期调用大模型 | 5.0 |
| 2 会员期调用调用大模型 | 60.0 |
| 3 小模型AI调用 | 1.0 |
| 4 其他服务器消耗 | =0 |
| 5 虚拟人调用成本 | 10.0 |
| 6 大模型情感识别 | 10.0 |
| 总计 | 86 |

目前，我们对大模型的降本措施分为三大步，共六小步。

1. 第一步是数据标注的降本。我们采用的方法是使用优质的大模型来生成训练数据，例如让 GPT 直接生成训练数据，这样可以轻松生成高质量的数据。
2. 第二步是算力补贴。由于我们公司是专精特新企业，我们申请了很多国家的补贴，这有助于降低成本。
3. 第三步是 GPU 端的优化算子。我们与一些服务器公司，包括华为、阿里等，合作进行服务器端的优化。GPU 本身不变，但我们基于开发者模

式进行自己的服务器优化，性价比非常高。

4. 第四步是加速框架，这是算法层的框架优化。
5. 第五步是大小模型混合。例如，我们要查天气，所有的语义槽位，如城市、日期等，这些可以直接用小模型处理，其精准度远高于大模型。用大模型做意图识别，然后将确定性的意图分流到 NLU 上，还有一些用大模型来兜底，这样成本会大幅下降。
6. 第六步是混合专家模型。我认为这适合除了基座公司以外的所有公司。要提高准确率，就需要将领域限制得更窄，知识库限制得更窄，这样才会更准确。



试错一年终落地

在过去一年多的时间里，我们对图灵 AI 口语老师产品进行了试错和迭代。投入成本主要分为几个部分。

1. 数据标注：这是成本中相对较小的一部分。由于我们长期从事语音助手的开发，已经积累了大量的数据，数据清洗和为大模型缓存数据还是非常高效的。
2. 算力成本：算力成本并不高，因为产品尚未大规模推广，用户量增长有限，因此推理成本保持在较低水平。
3. 算法重构：这是成本中较大的一块。随着大模

型技术的发展，我们必须将所有的小模型算法用大模型重新开发一遍。不仅涉及到技术层面的重构，还包括算法工程师的转型和后台服务、产品测试的重构。

4. 商业化成本：这是最大的成本部分。市场营销和应用层开发人员的投入非常巨大，尤其是在产品推向市场的过程中。作为教育公司，我们还必须购买大量正版内容。这不仅是因为训练需要，还因为在儿童教育领域，版权保护非常重要。拥有知名 IP 的版权内容能够带来溢价，家长更愿意为知名品牌的教育产品付费。

■ 我们如何做产品迭代

我们的口语老师的第一个版本是一个名为 Free Talk 的 AI 外教产品，大约在去年 5 月份左右，我们推出了这个版本。

这个产品受到了 OpenAI 发布的一个名为 Call Annie 的产品的启发，Call Annie 是一个大头人像，能够进行英文交互。这个产品有几个特点：首先，它呈现为一个大头形象，给人一种面对面对交流的感觉；其次，它进行全英文交流，不掺杂中文，模拟一对一外教的体验，并主打一对一外教的理想。

然而，在推广一段时间后，我们发现在实际使用中，无论是孩子还是成年人，都很难主动开口说话。即使有真人外教与孩子互动，孩子们也难以开口，不知道要说什么，也不会说。这导致 AI 外教很难带动孩子们进行对话。

此外，大模型在与孩子们交流时容易“超纲”。孩子们可能只学了一些非常简单的词汇，如 "What's this? It's a bottle."，但如果让大模型反问，可能会提出很长、很复杂的问题，这让孩子们难以接受。

第二个版本

在口语老师的第二个版本中，我们采取了不同的策略来解决孩子们不知道如何开口的问题。这个版本有几个关键点。

- 专属陪练：基于孩子们的回复虚拟老师会进行个性化回复。
- 话题引导：我们设置了一些孩子们熟悉的学习主题，在这个范围内引导孩子进行回答，例如开学或者交朋友的场景，并基于这些场景与孩子进行互动。这种方法可以帮助孩子们更好地融入对话，并激发他们的表达欲望。
- 推荐回复：如果孩子在对话中不知道如何回答，我们会提供一些建议性的回答。这些建议是由大

模型自动生成的，可以帮助孩子学习如何表达，并引导他们更顺利地参与到对话中。

每个人的学习情况和英语掌握水平都不尽相同，即使是在有设定话题的情况下，不同学生可能会觉得内容太简单或太难。因此，我们接下来要针对每个学生的个性进行优化。

个性化学习的关键在于分析学生的开口数据，观察他们的兴趣度和意愿度。同时，还要考虑学生回答的准确率，以及他们对提示语和推荐语的使用率。这些因素都是影响个性化学习效果的重要指标。

“口语老师版本二.2：个性化学习”

产品描述：针对学生个人情况AIGC新一轮学习意见

1. 开口数据：统计用户开口次数、开口时长、词汇储备与好句数量，并对比展示最新变化
2. 定位薄弱点：总评反馈整体情况，快速锁定发音/语法/语境表达中的薄弱点
3. 反馈详情：详情中可查看每一轮对话的反馈信息，进行针对性提升教学



在口语老师的开发中，第三点关键因素是教育教学体系的构建。我们生成的场景话题，无论是用于学校教育的打招呼场景还是开学场景，背后都有一支教研团队的支持，而最坚实的支撑来自于优质的教材。

以牛津树分级阅读为例，我们可以看到即使是像 VIP Kid 这样的真人外教一对一教学产品，其背后也不仅仅是外教的教学，还包括了一套教学方法和教案。外教会使用画板和教案，如牛津的《Let's Go》系列，一步步引导孩子学习。我们利用 RAG 技术来学习并生成课程内容，RAG 在生成基于问答的内容方面非常擅长。我们首先生成一些问答内容，然后对这些内容进行加工，使其成为课程教学的一部分。这样的学习方式可以

实现分级教学，根据学生的不同年级和水平来筛选话题的难度。

此外，尽管现在的 TTS 技术已经非常先进，但它仍然无法完全复制真人发音时的抑扬顿挫和适当的语速与停顿。因此，我们选择使用原版真人发声的内容，让孩子能够复述真人的发音，以此来提高学习效果。

我们还加入了真题练习，选用了与优质教材相配套的练习题。目前，使用 AIGC 技术生成的题目效果尚不理想，因此我们直接采用了教材中原有的配套习题。这些迭代和改进，都是口语老师产品不断进化的一部分，旨在提供更加个性化、系统化和有效的教学体验。

■ 第三个版本

在口语老师的第三个版本中，我们实现了商业化的显著进展。这个版本主要针对中高考的口语模考，提供了一个全真的模拟考试环境。这个环境从孩子试音、试麦克风开始，到试听题目，再到正式进行考试，完全模拟了真实考试的各个环节和流程。

过去的口语模考打分准确率较低，常受到老师们的诟病。现在，大模型在语法打分上的准确性大幅提升。例如，在听一段短文后回答有关问题时，大模型不仅考察语法是否正确，还要看是否准确回答问题，以及答案是否与题目相关，角色、动作和时间是否匹配。这些通过传统算法难以实现

的点，大模型都能很好地完成。从 2025 年开始，中国所有的中高考口语考试打分可能都会采用大模型技术，这将是一个解决痛点的质的飞跃。这也是商业化落地中一个难得的、能够快速推进的点。

最后一个特点是真题题库的应用。教育离不开版权，我们必须购买各省市的真题和模考题库。这些题库不仅涉及版权问题，而且出题人的思路独特，我们尝试过用 AIGC 技术模仿出题人的思路，但效果并不理想。如果替代率达不到一定水平，那么使用 AIGC 节省的工作量就非常有限，因此我们选择直接使用教材中的原题。

■ 与国外产品几种不同设计理念对比

在国外，大模型口语老师产品有几种不同的做法，这里分享几个例子。

首先是 Yanadoo，这是一款来自韩国的产品，其母公司是韩国最大的互联网教育公司。Yanadoo 的特点包括：

- 十分钟教育系统：提出每堂课只需十分钟，强调短时间内高效学习。
- 奖学金激励：通过奖金激励学生。
- 一对一 AI 语音指导：提供一体化的 AI 指导服务。
- 游戏化学习：利用游戏化元素和奖金刺激，让学生在 10 分钟的高强度专注训练后，通过与 AI 老师练习并获得积分，以此提高学习效果。
- 大模型应用：主要用在口语纠错上，提升学习精准度。

第二个产品是 Ainder，这是一个社交产品，其特色在于：

- AI 虚拟人社交：所有的社交对象都是 AI 虚拟人，每个虚拟人有不同的背景和人设。
- 个性化学习：用户可以与来自不同国家、不同口音和兴趣爱好的 AI 虚拟人进行英语交流。
- 共同兴趣：通过聊用户感兴趣的话题，比如 NBA 球星和术语，提高语言学习的兴趣和效果。
- 多语言者学习方式：该方法与一些多语言者通过与外国人聊天学习外语的方式相似，提供了一种自然的交流环境。

第三个产品是 Speak，这是一个 OpenAI 投资的

教育公司，其特点为：

- 真人录播课：结合真人教学和 AI 技术，真人负责上课，AI 负责作业。
- AI 作业：AI 用于听说读写作业的自动纠错和分析，包括发音、语法和词汇。
- 会员收费：虽然收费较高，但提供了高质量的学习体验。
- 产品评价：产品设计精良，无论是学英语还是其他外语，都获得了很高的评价。

第四个是多邻国，一个广为人知的平台，它在 GPT 3.5 发布时就是合作伙伴之一。多邻国采用的大模型用于：

- Explain My Answer：对用户的回答进行纠错和分析。
- Roleplay：在有限域下进行对话交互，让用户与 AI 进行 Free Talk 练习。

第五个产品是 Call Annie，一个提供随时视频通话的美女形象的产品，App 界面就像电话一样，提供交互体验。

最后一个 CheggMeta，可以说是美国版的作业帮，它强调：

- 课后作业指导：专注于孩子回家后的作业指导。
- 自适应学习：根据孩子的学习情况调整下一步的学习计划。

“口语老师：国外产品几种不同设计理念对比”

| | Yansibo | Ainder | SanaAI 爱知语音 Aisa, Lorali | 多邻国 | Call Annie | ChagoMate |
|---------|---|---|---|---|---|---|
| 设计核心 | 10分钟教学系统，模拟社交软件的社交“奖学金”激励，句型学习，社交对象是对AI语音指令一体 AI | 模拟社交软件的社交“奖学金”激励，句型学习，社交对象是对AI语音指令一体 AI | 真人录播课+听说读写AI作业 | 基于强大的趣味升级式学习的课程体系，再加入AI实现口语训练 | 英语视频对话——一个能随时视频通话的AI美女 | 通过AI做个性化、自适应、精确且高效的作业指导 |
| AI使用方式 | 口语纠错 | 与不同身份的AI虚拟人，或不同特长的AI老师互动 | 发音、语法、词汇细致的纠错和分析 | Explain My Answer——纠错和分析 Roleplay ——free talk或角色扮演 | 用ChatGPT freetalk | 更强调自适应，也就是纠错之后的自动完成学习计划调整 |
| 背景 | 母公司是韩国最大的互联网教育公司 Kakao | | 是Open AI 投资的教育公司 | 全球最大的语言学习上市公司 | openAI自己推出的 | 2013年上市的美版“作业帮” |
| 外 网 信 息 | https://baijiahao.baidu.com/s?id=1681253265367092301 | https://www.appinn.com/ainder/ | https://www.thepaper.cn/newsDetail_forward_24132604 | https://www.litong.com/0/879/729.htm | https://zhuanlan.zhihu.com/p/631689467 | https://zhuanlan.zhihu.com/p/623340198 |

总结来说，国内外在 AI 口语老师产品上的思路存在一些不同点。

国内 AI 口语老师产品的 1.0 版本在功能上大体相似。尽管每家公司都在训练自己的模型，影响体验最大的因素是模型训练的强度和精度。

国外产品的 1.0 版本普遍基于 GPT，因此在智能度上几乎一致。不同产品之间的主要区别在于各自的教学理念。例如，有的产品采用 10 分钟教学法，有的通过社交方式学习，有的结合真人录播课，有的游戏化学习，有的通过虚拟形象进行互动，还有的专注于作业辅导。

国内外产品在教学理念上有明显的差异。国外产品展现了多样化的教学理念，而国内产品可能在

未来会根据自己的理念逐渐分化。

在英语学习的口语老师应用中，每家公司至少都会设计一个虚拟人物头像，这是虚拟人的最基本表现形式。一些公司则更为复杂，将视频录制与虚拟人制作相结合。即使是较为简单的应用，也会加入虚拟人物头像，以增强用户体验。虚拟人的表达和人的情感连接是非常重要的环节，它与大模型技术有着天然的强关联性。

在移动互联网行业中，我们常会提到“杀手级应用”，而对于大模型技术来说，虚拟形象很可能成为杀手级应用中的核心要素。这是因为虚拟形象不仅能够展示背后的价值观、人设和情感，还能通过其形象与用户建立联系。

大模型的“行与不行”

大模型在教育板块的应用存在一些问题，同时也有其不擅长的领域。

- 课程设计不行：大模型缺乏教与学的体系支撑，无法独立进行课程设计。课程设计需要明确的目标、大纲和学生学习进度等，而大模型目前还达不到这样的要求。
- 解题能力不行：尽管有报道显示大模型通过了某些考试，但实际上在教育领域的测试中表现并不理想。以高考为例，准确率普遍低于 60%，小学五年级的准确率低于 85%，只有一二三年级的情况还算可以。
- 出题能力不行：大模型能出题，但题目套路明显，缺乏创意。现代中高考题目，特别是北京、上海等地的试卷，已经从传统的选择题、完形填空转变为应用题，要求考生解决实际问题，这需要综合能力。大模型目前还无法满足这样的出题要求。
- 讲题能力不行：大模型在讲解题目时可能会出现问题，可能会“胡说八道”，即使给出正确答案，其解释过程可能会越来越偏离正确方向，最终虽然得出正确答案，但教学场景中这样的讲解是不可接受的。

“不行”

- 课程设计不行：缺乏教与学的体系支撑
- 解题不行：中高考题准确率低于60%，小学低于85%
- 出题不行：套路明显缺乏创意
- 讲题不行：一本正经的胡说八道



Elon Musk 
@elonmusk

What we need is TruthGPT



大模型在教育领域的优势体现在以下几个方面：

阅读领域：大模型在阅读领域的表现是令人满意的。RAG 的应用在这方面尤其出色，它能够增强模型对信息的检索和生成能力。大模型被成功应用于基于学习材料的自动互动场景。这种应用通过与学习材料的结合，提供了自动化的、互动式的学习体验，这在当前教育技术中是一个非常

的方向。
微调 and 再训练：在使用大模型时，我们发现了一个令人惊艳的现象：与小模型相比，大模型在再训练时所需的数据量显著减少。例如，在口语老师的语法纠错功能中，原本需要 10 万到 100 万级别的数据量，而大模型仅需要很少的数据量就能训练出非常好的效果。

“行” —— 微调和再训练

The image contains three main components:

- Line Graph:** Shows the percentage of problems solved (Best-of-1860) on the y-axis (ranging from 62 to 78) against the number of solutions per problem (N) on the x-axis (log scale from 10^1 to 10^3). Three lines represent different methods: Process Supervised RM (top, orange), Outcross Supervised RM (middle, blue), and Majority Voting (bottom, grey). The Process Supervised RM line shows the highest performance, reaching approximately 78% at N=10^3.
- Table:** Titled "场景话题生成后的超纲词汇量" (Out-of-syllabus vocabulary after scenario topic generation). It compares the number of words before and after training for different grade levels.
- Tweet:** A tweet from Sam Altman (@sama) dated 11/15/23, praising a "really exciting process supervision result from our mathgen team" as a "positive sign for alignment." Below the tweet is a Chinese translation: "我们的 mathgen 团队带来了真正令人兴奋的过程监督结果。对齐的正号。"

| 年级 | 训练前 | 训练后 | 大模型训练数据量 | 小模型训练数据量 |
|------|---------|---------|----------|----------|
| 小学 | 40%-60% | 10%-20% | 5W+ | 10W+ |
| 初一 | 30% | 10% | 1W+ | 50W+ |
| 初二初三 | 15% | 3.5% | 1W+ | 100W+ |

大模型在教育领域的应用还包括过程监督式的方法。通过过程监督，可以显著提升大模型在解题方面的准确性，有望快速解决解题不准确的问题。

这种结合利用 AI 的强大功能，同时保留真人教师在教育中不可或缺的人文关怀和情感交流。

此外，我认为未来一两年内，教育领域将面临一个重要的改革和转型理念，即真人与 AI 老师的结合。在这个模式中，真人教师的角色是组织教学活动和建立情感联系，而 AI 老师则充当工具型的角色，提供无所不能的知识支持。



扫码分享文章

4 人团队，如何用大模型创造近千万业务价值？

嘉 宾 | 张源源，百姓车联数据科学与数据平台高级总监

编 辑 | 李忠良，InfoQ 编辑



大模型已经融入千行百业，在这个背景下，LLMOps 作为一种新概念，其定义、实践以及应对挑战成为了关注焦点。为了深入探讨 LLMOps 的意义和关键，我们采访了百姓车联数据科学与数据平台高级总监张源源，他分享了 LLMOps 在车损互助案例中的应用以及所面临的挑战与解决方案。以下是访谈实录。

InfoQ：现在其实大家 MLOps 都还没有搞得特别好，马上就出来了 LLMOps，当然也就没有特别标准的定义，在您看来 LLMOps 如何定义？它包含哪些内容？LLMOps 与 MLOps 您觉得两者较大的区别是什么？

张源源：简单来说，如下图所示。

LLMOps到底是什么？

| | MLOps | LLMOps |
|------|---|--|
| 覆盖流程 | 开发模型和模型应用往往是等价的，模型上线往往等于模型应用上线。 | 开发LLM和后续的模型应用是分离的，开发LLM和模型应用在技术栈上迥异。 |
| 目标人群 | 开发模型 = 模型应用 算法工作人员（算法工程师、Data scientist等等） | 开发模型 模型应用 简单应用：1) 无代码、低代码人群 2) 业务开发 复杂应用：1) 算法工作人员 2) 业务开发 |
| 产品形态 | SDK/Library/API等易于已有技术栈集成的方式为主 | SDK/Library/API等易于已有技术栈集成的方式为主 无代码拖拉圆选+SDK/Library/API都要有 |

● 基于开发LLM和后续的模型应用是分离的事实，LLMOps=开发模型LLMOps+模型应用型LLMOps。

● 开发模型类LLMOps往往有另外一个名字AI Infra，更多关注大模型训练过程的效率、效果等问题。

● 模型应用类LLMOps更关注有了LLM之后，如何开发LLM应用。

MLOps 用于管理 ML 应用的全生命周期，包括数据收集和处理、模型的训练、评估、部署和监控等，虽然会涉及跟多个工种打交道，但相关产品主要使用对象是从事 ML 算法开发工作的人员，比如 data scientist、算法工程师等等。

关于 LLMOps，我这里先提供对 LLMOps 的三种视角，通过比较这三种视角，可以更好了解 LLMOps 是啥。

- 一种视角认为 LLMOps 是 MLOps 在 LLM 场景下的直接迁移。主要使用对象还是算法工作人员。这种视角里认为的 LLM 全生命周期更多还是强调训练大模型的过程，对有了大模型之后如何做应用，其实覆盖的比较少。这种视角在某些之前对 MLOps 有了解甚至投资过但对 LLM 应用开发没那么熟悉的 VC 那里很流行。
- 另外一个知名项目 LangChain 提供了不一样的视角，它推出了号称是 LLMOps 的 LangSmith，它更多关注有了大模型之后如何开发大模型应用。可以从他们的产品设计理念里非常关注实验管理等相关 feature，有很强的 data science 思维，但目标客户已经不局限为算法工作者，很多业务开发者借助它已经能很高效地完成应用开发。
- 作为当下世界范围内风头最劲的 LLMOps 之一，也是我们国内开发者做出来的良心制作，Dify 同样更多关注有了大模型之后如何开发大模型应用的问题，但目标客户主要是无代码、低代码群体。通过后面这两种视角，其实可以看出 LLMOps 不应只是 MLOps 在 LLM 场景下的直接迁移。有了这三个视角的铺垫，其实通过直接对比 MLOps 和 LLMOps，容易给出更符合我们认知

的 LLMOps 定义。

- 从覆盖流程上说，对于 MLOps 来说，开发模型和模型应用往往是等价的，模型上线往往等于模型应用上线，想象一下各种推荐算法的开发和上线过程，但是对于 LLMOps 来说，开发 LLM 和后续的模型应用是分离的，都不是一波人，甚至都不是一个公司的人，开发 LLM 和模型应用在技术栈上迥异。
- 从目标人群上说，对于 MLOps 产品来说，因为开发模型和模型应用都是同一批人，它的目标人群就是算法工作人员，对于 LLMOps 产品来说，开发模型相关的 LLMOps 的目标人群仍然是算法工作人员，但模型应用相关的目标人群就丰富多样了，除了算法工作人员，无代码、低代码偏好人群、业务开发人员也是他们的目标人群。
- 从产品形态上说，也是类似，MLOps 和以开发模型为主的 LLMops 产品形态主要是 SDK/Library/API 等易与已有技术栈集成的方式，而模型应用相关的 LLMOps 增加了拖拉圈选等无代码操作。

所以基于前面分析里提到的开发 LLM 和后续的模型应用是分离的事实，我们就给出了 LLMOps 合理的定义，即 LLMOps= 开发模型 LLMops+模型应用型 LLMOps。开发模型类 LLMOps 往往有另外一个名字 AI infra，更多关注大模型训练过程的效率、效果等问题。模型应用类 LLMOps 更关注有了 LLM 之后，如何开发 LLM 应用。而开发模型类 LLMOps 其实也跟前面 MLOps 产品遇到的商业上的问题一样，可能会遇到有很多定制化需求而需要用到的公司往往会自研的问题，

当然因为当前相关领域人才供给严重不足，不是所有公司都有这样的能力，还是有不少机会；但对于模型应用类 LLMOps 来说，受众很广，也能解决当前应用落地门槛高的痛点问题，如果能聚集起大量的开发者，有了网络效应，是有很高的

商业价值的，甚至可以成为大模型的分发入口。特别需要指出的是，在接下来我分享的 context 下，我们所说的 LLMOps 是后者，也就是更多关注模型应用这块的 LLMOps。

LLMOps 在车损互助行业的应用案例

InfoQ: 在哪些环境中，车损互助使用到了大语言模型？

张源源：车损互助全流程都在使用，每一次深入跟业务侧沟通需求都能感觉到可以用大语言模型解决很多业务问题，下面这张图是我们 3 个月之前的规划。我们也做了大量创新的工作，比如我们产品负责人之前发表过一篇我们用大模型去解决准入报价里 VIN 匹配的问题，当时在圈子内引起了一个小轰动，很多人都跟我打听是怎么做的；

再比如，我们规划了用大模型去做智能理赔定损 agent，通过几张照片和报案信息，就能给出来带价格的维修单，会涉及非常大模型能力应用的子问题，很多人都对这块非常好奇也非常好看，

这个对汽车维修行业来说带来的影响非常大，如果能做好，预期创造的业务价值非常高；

还有，我们最近搞得 text2data 工作，如果你之前对 text2sql 有过了解，你会发现这个工作从原理上就比 text2sql 靠谱非常多，通过我们在埋点、ad hoc query 方面的落地实践，可以说对于真实场景的取数需求来说，可以说已经完全不需要工程师介入了，我们自己的数仓工程师做完这个项目就自己说感觉数仓这个职位要不存在了。

我们最近也想到了其他更多应用场景，比如用 phone agent 去帮忙做第一轮面试筛选、服务质量反馈、用户报案问题收集（不仅仅通过 chatbot，还是有很多用户习惯用 phone 去报案）。



InfoQ: 您可以分享下，您这边采用的基础模型是什么吗？

张源源: 我们一直是选择最好的模型，根据特定的场景选择特定的模型，比如大多数时候选择GPT4，在代码生成相关的使用 Claude3，我们也是评测和对比了很多选择。在现阶段我们场景里，推理价格不是我们优先考虑项，效果是最优先考虑的。

InfoQ: 在哪些场景中使用了 LLM？如何引导大语言模型输出您期望的结果？

张源源: 场景如上图，在车损互助的准入报价、理赔定损、日常运营、内部提效等等场景都有应用。在引导大模型输出期望结果这块，我们最重

要的经验就是确定性的交给确定性的去做（比如能调用 API 搞定的就直接调用 API，比如多用 workflow，把 zero shot 调用大模型，拆解成多个确定性节点和几个调用大模型的节点），剩下的才交给大模型；另外一个经验是，团队一定要有有实验思维、懂数据科学的人，才能把这个事情真正做好。

InfoQ: 如何评估大模型的回应呢？是好的还是坏的？

张源源: 首先去看自己的 task 是不是已经有 benchmark，比如你搞的是翻译类任务，这种肯定有很丰富的 benchmark，直接去看模型在这些 benchmark 上的表现，或者去关注一些大模型的 technical report 以及 lmsys 等的 leaderboard，

当然除了这些，还可以自己构建评测集合，让领域专家或者大模型本身帮你标注这些结果好坏，这个时候类似 Dify 这样的 LLMOps 就提供了非常好的标注回复功能，能提供很好的支持。当然，这也是我上面说的，团队一定要有有实验思维、懂数据科学的人，他好去设计实验 pipeline，以及评测模型和各种配置的好坏。

InfoQ：底层 API 模型的持续变化会对输出结果的影响也是非常大的，如何处理这些情况呢？

张源源：无他，就是做实验，在 benchmark 和自己的评测集合上做实验，根据效果好坏来决定是否切换。

InfoQ：除去输出的期望问题，还有哪些挑战是您这边遇到的？又是如何解决的？

张源源：总体来说，遇到的挑战还好，哪里不会学哪里，比较享受这种遇到问题就解决问题的感觉吧，如果非要说挑战，主要有两个吧，一个是 RAG 这部分，现在市面上的方案还没有达到预期，

核心我觉得是当前是工程的人搭起来架子，但是对效果提升有帮助的算法相关人才跟进还不够以及还没有整合到主流工程里去，这部分也呼吁更多信息检索相关的人杀入这个领域，机会很大，低处果实也很多，另外一个更大的挑战就是一定要 catch up 最新进展，有太多东西需要深入学习和 research，时间总是不够用的感觉。

InfoQ：在搭建与使用 LLMOps 过程中，您这边一共有多少人参与？为团队带来哪些收益呢？

张源源：据我们内部初步估计，各个场景第一年创造的业务价值预计近千万，这还是考虑我们第一年用户量不够大、很多合作伙伴 API 还没有如期接入的情况，而且有很多用户体验方面的价值无法用金额直接衡量，我们公司是志在用 AI 作为核心竞争力在海外做一款颠覆性的车损互助产品。拿到这个业务结果，背后主要是三点，第一就是我们在大模型的认知足够，第二就是对业务场景问题深入去思考，第三就是借助 LLMOps 让我们低成本做实验和验证，整个过程，核心参与人员就四五个人。

安全性和合规性问题

InfoQ：鉴于车损互助行业可能涉及到用户个人信息和交易数据等敏感信息，您是如何确保模型对这些信息进行合规处理的？

张源源：我们目前的应用场景还没有太多涉及，有一两个场景里有这种问题，但是也不严重，也就是用户上传车损照片，这些都可以通过免责申

明加上产品手段去解决，也就是说在用到大模型之前就解决掉了，尽量不在大模型这里进行解决。

未来的发展方向和预测

InfoQ：随着技术的不断发展，您对 LLMOps 的未来发展有何预测？比如在模型自动化、自适应性、实时性等方面的进展。

张源源：这部分在分享里也会涉及，应用类 LLMOps 主要在解决降低门槛、提高可集成性、提高可观测性、提升效果和效率这几个问题。

- 在降低门槛方面，当前以 Dify、Coze 为代表的开发类 end2end 的 LLMOps 极大的降低了普通人开发 LLM 应用的门槛，意义重大，甚至因为这一点，LLMOps 现阶段的流量入口价值和分发价值都被低估了。
- 在提高可集成性方面，通过 API 把 LLM 应用作为整体跟其他系统对接的方式还不够，还需要节点级别的对接方式，workflow 的 http 节

点有一定帮助，但还不够，比如往往没有全局 memory。当前主流 LLMOps 更多思考的是新创建的应用，但市面上更主流的应用场景是需要跟已有系统进行集成，提高可集成性能极大提高 LLMOps 的上限。

- 在提高可观测性方面，当前 LLMOps 做的还不够好，比如很多还不支持版本控制，tracing 做的也不够好。
- 在提升效果和效率方面，当前 LLMOps 做的也还不够，效果和效率其实也是在落地过程中，用户最在意的点，但大模型的自身能力缺陷在没有正确使用大模型经验的普通人那里被放大，导致大模型落地差强人意。期望 LLMOps 能够对于有能力的人，提供更多集成其他优秀解决方案的机会，甚至这本身也是商业机会。对于没有能力的人，应该提供更好的经过广泛证明的默认选项。

嘉宾介绍

张源源, 百姓车联 AI/Data 方向负责人, 中国人民大学校外导师, 中国商业统计学会常务理事, 数据科学社区统计之都常务理事。长期跟踪 AI/Data 方向前沿技术发展, 发表了多篇 AI 方向顶级 Paper, 有多项相关专利; 在百度、阿里、百姓车联等多家赛道内头部公司有过行业内开创性的工作, 在 AI/Data 方向有超过 10 年的积累。目前正在百姓车联带领团队开发车损互助行业首个基于大模型的智能车损互助系统。



扫码分享文章



极客邦科技
双数研究院



极客时间 | 企业版

数智时代的 AI 人才粮仓模型



01

更精准地识别、选拔和培养符合企业发展战略的 AI 人才

02

快速构建起一支具备高度竞争力的人才队伍

03

助力企业建立科学、公正的评估机制，实现人才价值的最大化



联合指导单位

工业和信息化部重点领域产业人才基地联合建设机构
铸基计划 | 高质量数字化转型推动行动

研究机构

极客邦科技双数研究院

合作单位

《培训》杂志



扫码获取电子版

AIGC 人才能力建设 学习地图

AIGC



AI 领域专家讲师（部分）

| | | | | |
|--|---|---|---|-------------------------------------|
| 李健 广联达 CEO 联合创始人 世界 500 强集团 CIO | 于游 金山云 大模型 首席专家 | 钱洪燕 腾讯 TEO 基础架构部 FA 组专家 | 张杰 中软国际技术 副总裁 | 黄佳 爱知数科研究 首席研究员 |
| 孙超 Mobvista 集团 副总裁兼首席架构师 | 徐昊 ThoughtWorks 全球技术架构顾问 | Tyler 前亚马逊应用科学家 某大厂 AIGC 赛道负责人 | 彭婧田 Google Developers Expert | 李佳芮 电子蓝公司 创始人兼 CEO |
| 蔡科坤 慧数科技 创始人兼 CEO | 彦博 蓝盾科技 创始人 | 尹会生 水之人工智能 技术合伙人 | 徐文治 baidu 创始人 市北托尔森科技创始人 | 陈扬 清华大学 伏羲机器人 |



扫码获取
高清电子版

技术实践

TECHNOLOGY



第一章：大模型训练与推理

万字干货！手把手教你如何训练超大规模集群下的大语言模型

演 讲 | 刘育良, AI 平台大模型训练负责人员

审 核 | 傅宇琪, InfoQ 见习编辑

| 褚杏娟, InfoQ 编辑

策 划 | 蔡芳芳, InfoQ 主编



快手总结了一套超大规模集群下大语言模型训练方案。该方案在超长文本场景下,在不改变模型表现的情况下,训练效率相较 SOTA 开源方案,有显著的吞吐提升。通过细致的建模,可保证 Performance Model 十分接近真实性能,基于此 Performance Model,解决了大模型训练调参困难的问题。

本文整理自快手 AI 平台大模型训练负责人刘育良在 QCon 2024 北京的分享“超大规模集群下大

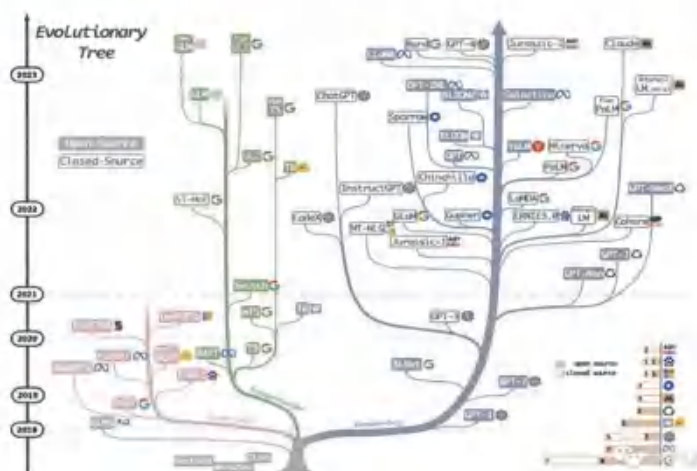
语言模型训练的最佳实践”。演讲结合在快手超算集群上的大模型训练经验,阐述大模型训练在超大规模集群下遇到的挑战和热点问题的演变,以及对应的解决方案。同时,针对最具挑战的超长文本场景,进行案例分析。最后,根据未来大模型的发展趋势,对训练领域的技术探索方向进行探讨。

本文由 InfoQ 整理,经刘育良老师授权发布。以下为演讲实录。

简单介绍一下背景，下图清晰地描述从过去到现在，即 23 年之前所有主流大模型的发展历程。从技术架构的角度来看，Transformer 架构无疑是当前大模型领域最主流的算法架构。其中包括

以 Bert DiT 为代表的 Encoder-Only 结构，以 T5 为代表的 Encoder-Decoder 结构，以及现在非常火热的 GPT 系列的 Decoder-Only 结构，这也正是我今天想要讨论的重点。

大模型发展历程



- Encoder only
 - Bert
 - DiT
- Encoder-Decoder
 - T5
 - GLM
- Decoder-only
 - GPT 系列
 - Llama

大模型这个名字非常直观地表达了其主要特点，那就是“大”。具体量化来说，参数数量大，比如从 LLAMA2 的 70B 到 GPT-3 的 175B，再到 GPT Moe 的 1.8T。其次，数据量大，我们训练一个大模型通常需要达到 T 级别 tokens 的数据量。再者，由于模型尺寸巨大和数据量庞大，随之带来的是巨大的计算量，基本上现在表现良好的大模型都需要 1e24 Flops 级别以上的计算量。

那我们为什么需要将模型扩展到如此规模？或者说，为什么模型越大效果越好呢？大模型持续扩大规模会变强的理论基础是 scaling law。接下来展示的这张图来自 OpenAI GPT-4 的技术报告，scaling law 简单来说就是模型的能力与计算量有强烈的正相关性。因此，我们可以通过不断增加模型规模和数据规模来提升模型的能力。

为什么大模型有效

scaling law

简单来说，就是模型能力与计算量有强相关性，可以通过持续扩大模型规模、数据规模来提升模型能力



接下来，我想和大家探讨一下训练引擎的定位，用一句话来概括就是“工欲善其事，必先利其器”。

首先要做的是提供一套可持续扩展的工具箱，这样就可以不断扩展模型规模、数据规模和序列长度，从而提升模型的表现。其次，我们要提高扩展效率，即提高 scaling efficiency。如果将刚才提到的 scaling law 的横轴从计算量换成计算卡时，那么我们的目标就是通过提高训练效率来减少总体的训练时间，进而增加 scaling law 的斜率。

作为大模型算法解决方案的提供方，我们要与算法进行联合优化，从训练和推理效率出发，提出模型结构的建议。同时，作为超算集群的使用方，我们需要根据大模型的典型通信模式和计算模式，提供组网策略和服务器选型的建议。

接下来，我想介绍一个衡量训练方案好坏的指标，即 MFU。MFU 的计算公式是有效计算量除以训练时间再除以理论算力。这里提到的 MFU 计算公式与之前论文发表的有所不同，原因在于当前主流的大语言模型都采用了 causal mask。对于特定的模型和特定的集群，有效计算量和理论算力都是恒定的，因此我们的目标是通过减少训练时间来提升 MFU。

为了提升 MFU，我们能做的主要有三点：

1. 减少无效的计算，这通常来自于重计算；
2. 提高集群稳定性，减少因稳定性问题导致的集群不可用时长；
3. 减少通信的影响，这将是接下来讨论的核心内容。

MFU- Model Flops Utilization

$$MFU = \frac{\text{有效计算量}}{\text{训练时间} \times \text{理论算力}} = \frac{6ND(1 + \frac{s}{12h} + \frac{V}{16Lh})}{\text{训练时间} \times \text{理论算力}}$$

decoder-only LLM

减少无效计算
提高稳定性
减少通信影响

| 符号 | 含义 |
|----|------------------|
| N | 参数量 |
| D | 数据量 |
| s | 序列长度 |
| h | hidden dimension |
| V | 词表大小 |
| L | 模型层数 |

分布式训练的主要难点

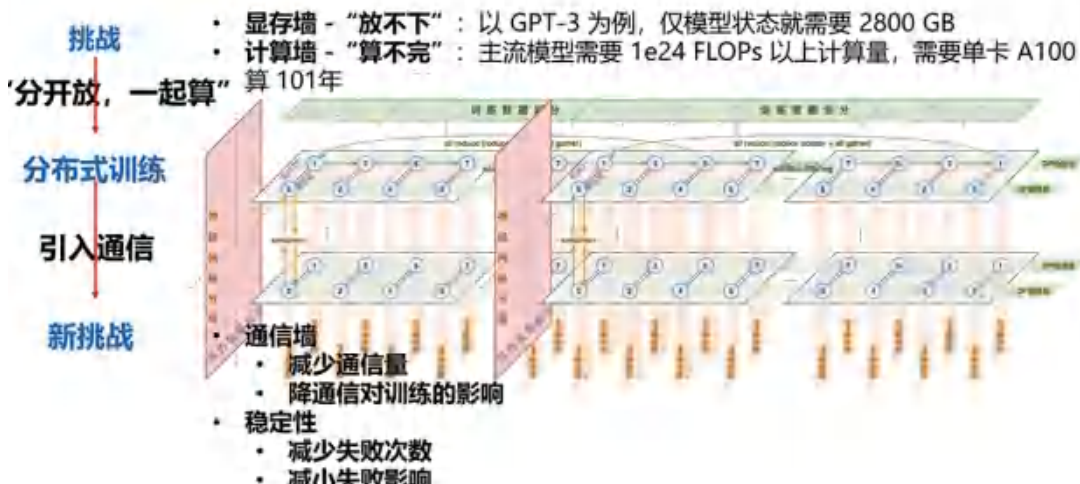
与小模型相比，大模型的挑战可以概括为“放不下和算不完”。以 GPT-3 为例，单是模型就需要 2,800 GB 的存储空间。而且，主流模型的计算量之大，以至于如果使用单张 A100 显卡，需要计算 101 年才能完成，这显然是不切实际的。

我们的解决方案是直接的，即通过混合并行的方式来实现分开放和一起算。具体来说，我们把模

型状态和中间激活值分散在整个集群上，然后通过必要的通信来完成联合训练。但混合并行也带来了问题，它引入了大量的通信，这导致训练效率急剧下降。因此，在大模型训练中，我们可能需要做的工作主要集中在两个方面：第一，减少通信量；第二，降低通信对计算和训练的影响。这两项工作对于提升大模型训练的效率至关重要。



训练阶段的挑战

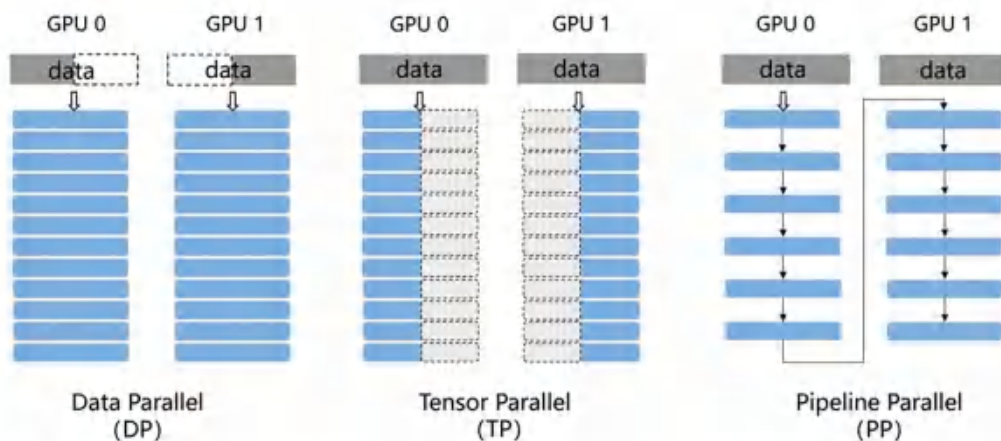


简单介绍一下混合并行中经典的三种并行方案。首先是数据并行，简称 DP。正如其名，数据并行是将数据分割到不同的计算设备上，然后由这些设备完成各自的计算任务。第二种是张量并行，简称 TP。张量并行是将模型中某些层的参数分散

到不同的设备上，每个设备负责完成部分的计算工作。第三种是流水并行，简称 PP。流水并行是将模型的不同层切分到不同的计算设备上，类似于流水线的工作方式，各个设备协同完成整个模型的计算过程。



并行方案简介



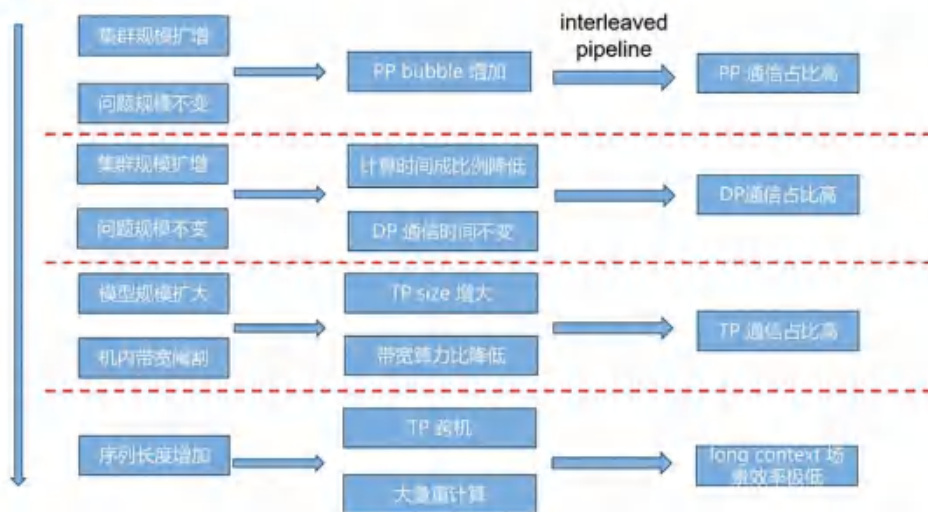
现在我来分享一下在实际操作中，训练大模型时遇到的一些热点问题的演变。

首先，随着集群规模的扩大，即 GPU 数量的增加，而问题规模，也就是模型的大小保持不变，这导致了 PP Bubble 急剧增加。为了解决这个问题，我们引入了 interleaved pipe。然而，这种方法也带来了另一个问题，即 PP 的通信量成倍增加。集群规模的扩大同时也导致单个 iteration 的计算量成比例下降，但 DP 的通信时间与参数量成正比，所以通信时间实际上并没有减少，这导致 DP 的通信开销持续扩大。

随着我们从 66b 模型扩展到 175b，再到更大的模型规模，我们需要将 TP 的尺寸从 2 增加到 8，这导致了 TP 的通信量大幅增加。同时，由于 A800 和 H800 集群内部的 Nvlink 被阉割，这在千亿参数模型训练时，TP 的通信开销实际上超过了 30%。最后，随着 context window size 的扩大变得越来越重要，序列长度的增加，原有的方案要么需要进行 TP 跨机操作，要么会引入大量的重计算。这导致在 long context 场景下，原有的训练方案的效率极低。



热点变迁



大模型训练在超大规模集群下的挑战与解决方案

随着模型规模和集群规模的扩大，通信在训练过程中的占比越来越大。为了更直观地展示这一现象，我提供了两张时间线图，它们没有应用通信重叠技术。第一张图突出显示了在实现 DP 重叠前的数据并行通信状态，第二张图则突出显示了在实现 TP 重叠前的张量并行通信情况。

从图中我们可以看到，在端到端的训练过程中，DP 的通信占比实际上超过了 15%，而 TP 的通信时间占比也超过了 30%。因此，减少通信对训练的影响，对提升训练效率至关重要。



计算通信 overlap

动机

从热点变迁中，可以看出，随着模型规模和集群规模扩增，通信占比越来越大。



DP Overlap

我们实现 DP overlap 的方法，借鉴了 ZeRO 3 的设计理念。ZeRO 的实现方式是将优化器状态分散到不同的 DP rank 上。通过 all-gather 操作来获取完整的权重，然后使用 reduce-scatter 操

作将梯度累加到不同的 rank 上。由于数据依赖于第一个模型块，前向传播 (forward) 只依赖于第一次 all-gather。因此，在这次计算过程中，我们可以利用这段时间来完成其他 all-gather 的

通信。除了第一块模型之外，其余的 all-gather 操作都可以与前向传播重叠。对于反向传播 (backward)，除了最后一次的 reduce 操作外，所有的 all-gather 操作都可以与反向传播重叠。

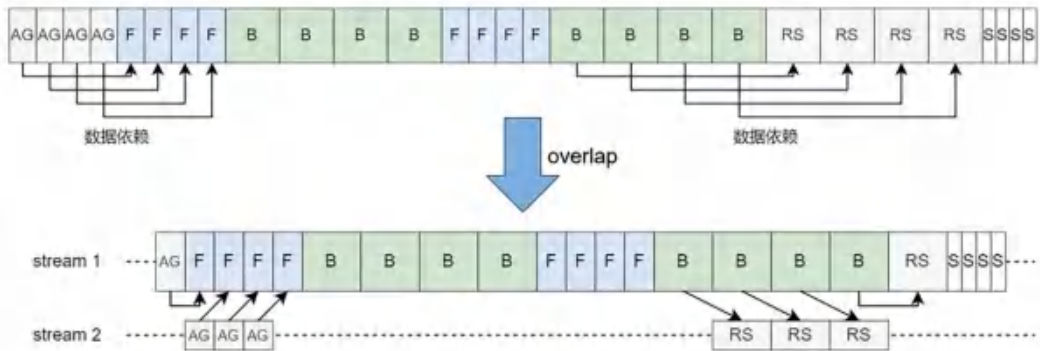
我们将这种思路应用到了混合并行中。通过分析数据依赖，我们发现情况几乎是一致的。例如，前两次的正向传播都只依赖于第一个 all-gather。在这段时间内，我们同样可以用来掩

盖第二次的 all-gather 操作。类似地，reduce-scatter 操作也可以被反向传播掩盖。由于只有第一个 pipeline stage 的通信无法被重叠，所以重叠的比例是 1 减去 v 分之一，其中 v 代表虚拟 pipeline stage 的数量。当然，我们也可以通过进一步划分来完成第一个 pipeline stage 通信内容的重叠，但为了简化我们后续的讨论，我们暂时不考虑这种情况。



DP overlap

ZeRO overlap 方案



DP overlap 的方案在理论上看起来非常吸引人，但实际应用中，我们真的能显著提升训练效率吗？在进行 DP overlap 优化时，我们遇到了三个主要问题。首先，是通信和计算资源之间的竞争问题。当通信和计算操作同时进行时，它们会争夺有限

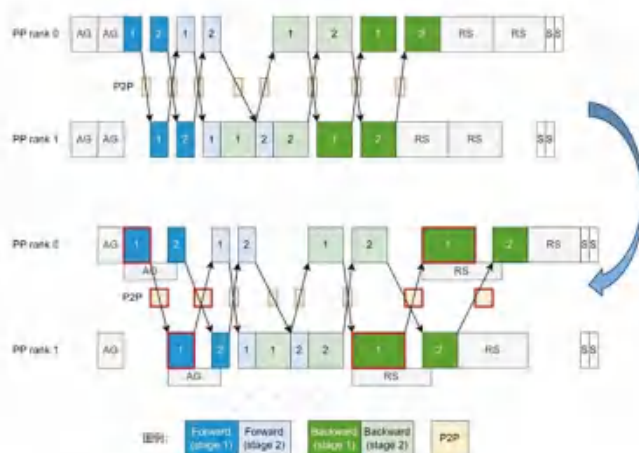
的硬件资源，这可能会影响整体的系统性能。其次，在混合并行场景下，DP overlap 还可能带来 PP bubble 的问题。第三，不同通信资源的争抢还可能导致网络拥塞。



DP overlap

挑战

- 通信资源与计算资源竞争
- 产生额外的 pipeline bubble
- 不同通信之间资源争抢甚至会导致网络拥塞



我们来谈谈通信与计算之间的资源竞争问题。最突出的问题是 SM 资源的竞争。简单来说，通信会占用一部分 SM 资源，这进而会影响计算的性能。然而，我们在进行性能分析后发现，用于计算的 SM 数量与通信占用的 SM 数量并不匹配。

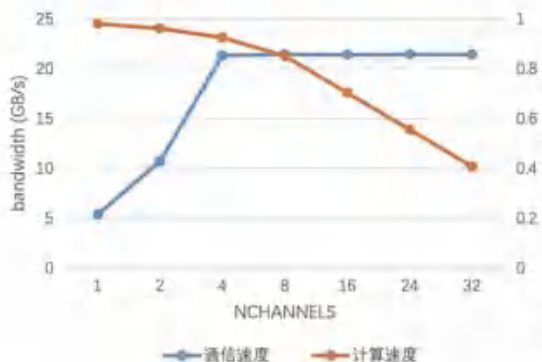
经过更深入的分析，我们发现在 Volta 架构之后，TPC 上的 SM 会共享其配置的共享内存。以 A800 为例，当一个 TPC 为通信内核分配了共享内存后，该 TPC 内的另一个 SM 也会共享这个共享内存配置，导致计算 kernel 无法复用这部分被分配出去的 SM。此外，在 Hooper 架构上，或者更准确地说，是 SM90 以后，我们发现系统会将一个 SM 内的一些 thread block 组织在一起形成一个 virtual cluster，然后以 cluster 为单位进行调度。这可能导致 sm 碎片问题。

我们发现通信与计算之间的相互影响主要与通信的 CHANNELS 有关。CHANNELS 越多，通信占用的 SM 数量也就越多，这导致计算速度变慢。我们的测试是使用 A800 显卡进行的，配备了四张网卡的 A800 来进行测试。从表格中可以看到，当通信的 NCHANNELS 数量小于网卡数量时，通信速度会显著下降。而当 CHANNELS 数量大于网卡数量时，通信速度几乎不再提升。如果继续增加 NCHANNELS 的数量，只会进一步导致计算速度变慢。因此，在综合考虑通信速度和计算时间的增量之后，我们选择了整体最优的通信 CHANNELS 数量。通过前面的分析，我们可以发现，通过牺牲一定的通信带宽，可以达到通信与计算的全局最优状态。



通信资源与计算资源竞争

量化分析



综合考虑通信速度、计算时间增量

例: A800, TP size=2, DP size=8

| NCHANNELS | 计算时间延长比例 | 通信速度 (GB/s) | 通信 1GB 时间 (ms) | overlap 1GB 导致的计算时间增量 (ms) |
|-----------|----------|-------------|----------------|----------------------------|
| 1 | 1.89% | 5.35 | 163.55 | 3.03 |
| 2 | 3.65% | 10.71 | 81.70 | 3.03 |
| 4 | 8.00% | 21.35 | 40.98 | 3.04 |
| 8 | 17.39% | 21.46 | 40.77 | 6.04 |
| 16 | 42.11% | 21.46 | 40.85 | 12.10 |
| 24 | 80.00% | 21.46 | 40.77 | 18.12 |
| 32 | 145.45% | 21.45 | 40.79 | 24.17 |

然而我们会发现一个问题，即并非所有通信都能够与计算进行 overlap。如果我们降低全局的通信 CHANNELS 数量，那么我们的策略可能在一定程度上损害到为 overlap 计算的通信效率。为了解决这个问题，我们区分对待了 overlap 计算的通信和非 overlap 计算的通信。对于 overlap 计算的通信，我们会综合考虑通信速度和计算时间增量，然后调整出一个最优的 CTA (Compute Thread Array)。而对于非 overlap 计算的通信，我们会设置带宽最优的 CTA。

除了计算与通信资源的竞争问题，我们还会遇到不同通信之间的竞争问题。我们的解决方案是采

用分桶通信。分桶之后，一个 all-gather 会被拆分成多个 all-gather 操作，这样单次的 DP 通信就可以被单次的计算所掩盖，从而尽量避免与 PP 产生资源竞争。但这并没有解决所有问题。即便我们实施了分桶策略，我们发现由于网络抖动等原因，DP 的通信和 PP 的通信仍有小概率发生 overlap，导致多流打入单网卡的现象，进而引起网络拥塞。为了缓解由不同通信之间的冲突所造成的网络拥塞问题，我们从 DCQCN 拥塞控制算法和不同的流优先级上进行了优化。通过这些优化措施，我们能够减轻网络拥塞，提高整体的训练效率。



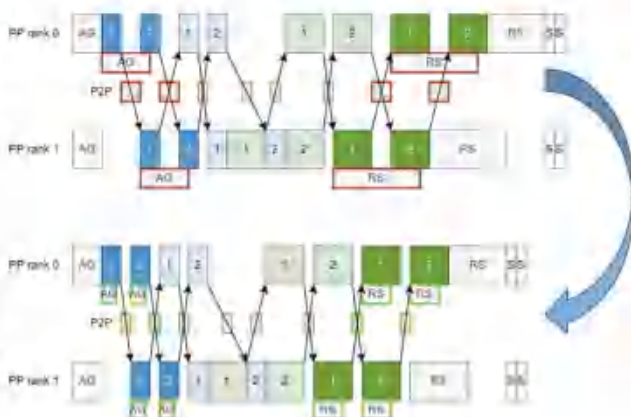
不同通信之间竞争问题

分桶通信

我们采取了分桶通信的方式，将 dp 通信内容拆分成 pp 组，尽量避免和 PP 通信的资源竞争。

网络拥塞优化

即使我们做了分桶通信，但是由于网络抖动等原因，会导致 DP 通信和 PP 通信会有 overlap，产生多流打单网卡的现象，导致网络拥塞。我们从DCQCN 拥塞控制算法和不同流优先级上做了优化，缓解了不同通信之间冲突带来网络拥塞问题。



DP overlap 引入的 PP bubble 问题。在前面，我们讨论了通信对计算效率的影响。如果我们模仿 ZeRO 的调度策略，由于 overlap 计算的时间会长于 none overlap 计算的时间，这种负载不均衡会导致 PP bubble 的产生。即图中的 Micro batch 2 的前向传播和 Micro batch 1 的反向传播较长的现象，这展示了负载不均的情况。我们

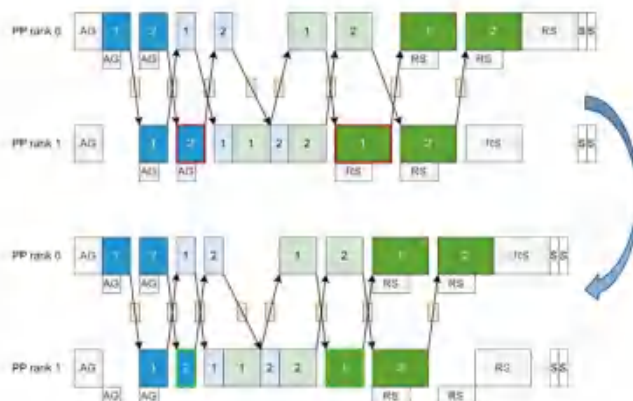
提出的解决方案是通信时机的纵向对齐，这样可以极大地缓解 PP bubble 的问题。同时需要强调的是，从计算 overlap 部分移出来的通信都被放在了 PP bubble 上，因此它不会产生任何额外的影响。这种策略有助于平衡负载，减少因通信和计算不匹配而产生的效率损失。



DP overlap 引入的 PP bubble

通信时机纵向对齐

- 对齐通信时机，减少由于通信 overlap 导致的负载不均问题
- 从计算 overlap 部分移出的通信放在 pipeline bubble 上，不会产生额外影响

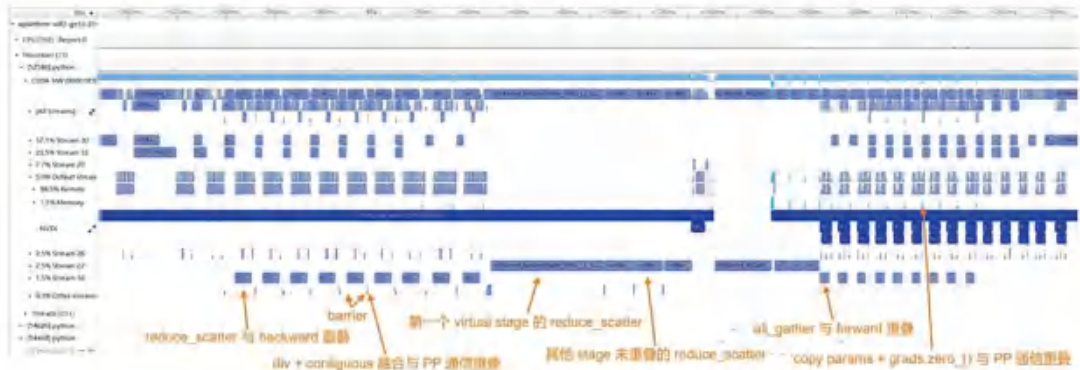


下图展示了我们最终优化后的 timeline。在这个优化版本中，我们实现了 reduce-scatter 与反向传播的 overlap，同时 all-gather 操作与前向传播也实现了 overlap。此外，我们通过分桶通信、

网络预测控制、通信 CHANNEL 调优以及通信时机的纵向对齐等方法，大幅优化了 DP 的通信开销。这些优化措施共同作用，提高了整体的训练效率，减少了因通信而产生的延迟和资源浪费。



Timeline



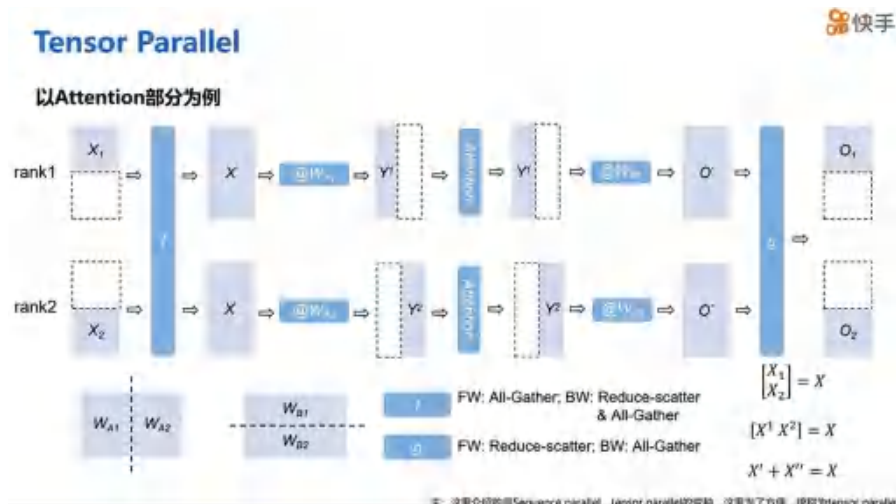
TP Overlap

在介绍 TP overlap 之前，我想先向大家介绍一下 Tensor Parallel 的流程。这里实际上采用的是 Megatron-LM 中提出的序列并行，但为了简便，后面我们都简称为 TP。我们以 attention 为例来介绍 TP 的流程。

输出。接着，通过第二个 GEMM，我们可以得到一个部分求和。最后，通过 reduce-scatter 操作，我们可以得到沿横轴切分的数据结果。可以看到，这两个模块的输入和输出都是沿横轴进行切分的，因此这个过程可以持续不断地进行。

在 TP 中，一个 attention 层包含两个 GEMM 操作。第一个 GEMM 是将权重沿纵轴切分，第二个 GEMM 是将权重沿横轴切分。首先，我们将输入数据沿横轴切分，然后在第一个 GEMM 计算前，使用 all-gather 操作将两个输入合并。完成第一个 GEMM 计算后，我们会得到一个沿纵轴切分的

在计算过程中，实际上穿插了两个通信操作，一个是 f，一个是 g。其中，f 在前向传播时对应 all-gather 操作，在反向传播时是 all-gather 加 reduce-scatter。而 g 在前向传播时是 reduce-scatter，在反向传播时是 all-gather。我们后续的 TP overlap 策略就是围绕这些通信操作来进行的。



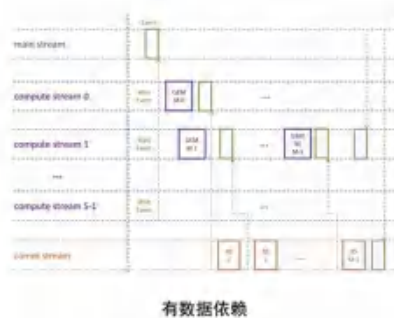
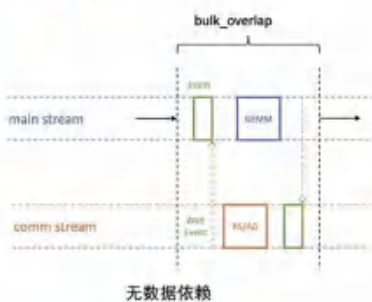
在针对 TP 进行计算通信重叠设计时，我们将其分为两个部分：一部分是有数据依赖的通信重叠，另一部分是无数据依赖部分的重叠。下图左侧展示了无数据依赖计算重叠的方案，这是一种比较经典的计算通信重叠方案。如前所述 DP overlap 就是其中的一种情况。此外，稍后我们会讨论到的 TP 中的列线性反向传播也会采用这种方案。

右侧的图展示了有数据依赖的计算通信重叠。在这

种情况下，我们会将 GEMM 操作拆分成若干份（s 份），每一份的计算可以与下一次的计算重叠。需要注意的是，我们将计算也分散到了多个 stream 中。这样做的原因是，不同 stream 之间的计算是没有依赖关系的。因此，计算在不同 stream 之间也可以实现一定的重叠。这部分重叠来自于 kernel 即将结束时，SM 资源的占用会有一定程度的下降。借助 CUDA 运行时调度，可以把另一个 stream 中的 kernel 提前调度上来，从而实现计算的重叠。

优化思路

数据依赖



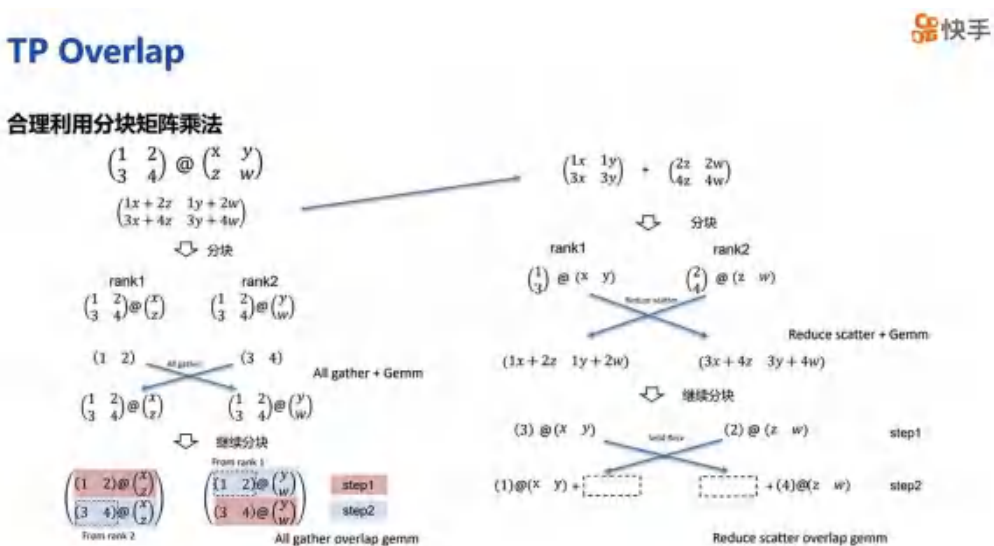
注: bulk overlap 和 pipeline split overlap 思路参考了 transformer engine source code

下面我会介绍一些 TP overlap 的细节，关键在于合理利用分块矩阵乘法来进行矩阵乘法运算。首先，对于一个矩阵乘法操作，我们可以沿着纵轴将其切分成两部分，并将这两部分分别放到不同的 rank 上。在计算之前，需要进行 all-gather 操作，这实际上是之前介绍的 all-gather+GEMM 的方案。我们可以将这一步的计算进一步分块，在 rank 1 和 rank 2 上分别进行一部分计算，这一步可以称为 step 1。

在执行 step 1 计算的同时，我们可以进行 send 和 receive 操作，将自己持有的那一部分输入数据

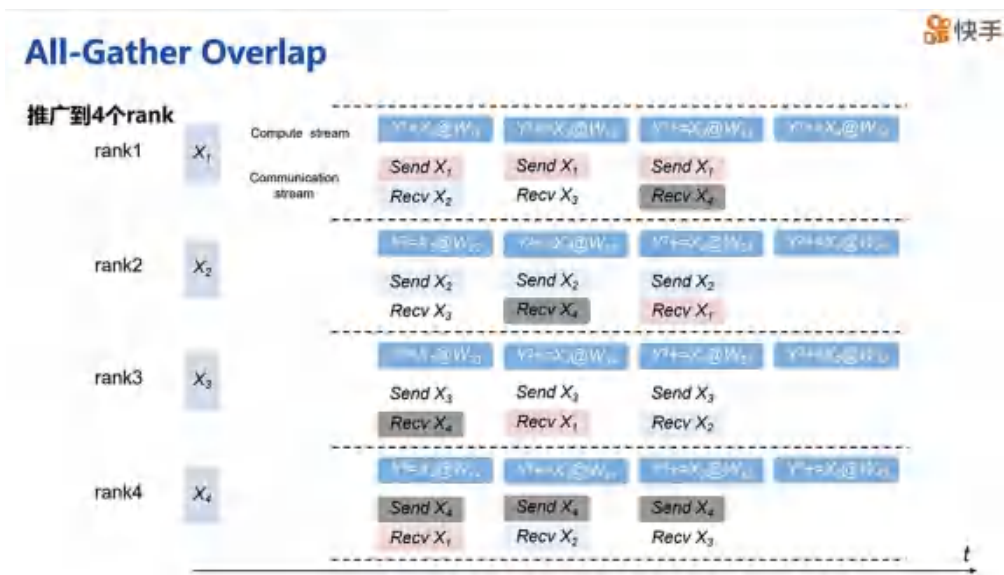
发送给另一个 rank。接下来执行 step 2，这样通信就与 step 1 的计算重叠起来了。同时，我们还可以通过分块的方式拆分矩阵，也就是将矩阵分为左块和右块。分块的结果可以先计算出部分结果，然后再进行 reduce-scatter 操作，这也是之前介绍的 reduce-scatter+GEMM 的计算流程。

实际上，右侧与左侧的方案类似。我们同样可以将计算分块，先执行 step 1 作为一部分计算，然后将 step 1 的计算结果发送给另一个 rank。在发送的同时，可以开始执行 step 2 的计算，这样就可以实现计算和通信的重叠。



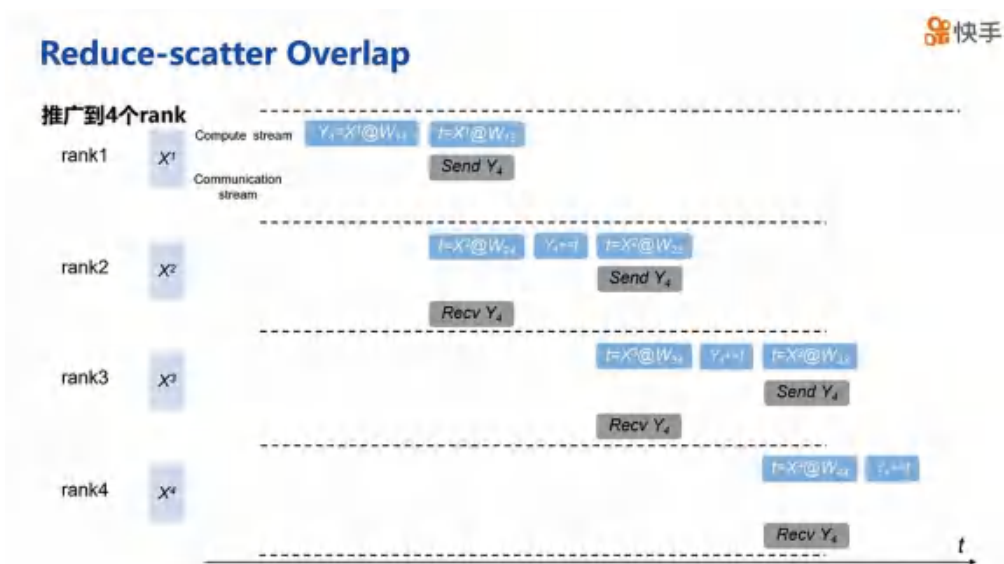
然后我们可以将这种策略推广到四个 rank 的场景中。为了简化表述，我们将计算的 stream 都合并到了一起。对于 all-gather overlap GEMM，我们会特别关注第一个 rank。第一步，我们使用自己持有的那部分输入来进行计算，同时将自己持有的内容也发送给其他 rank，并接收其他 rank 中持有的那部分输入。接下来的第二步、第三步、

第四步都是按照相同的原理进行。通过这种方式，我们就可以得到一个 all-gather 的 overlap 流程。这样，每个 rank 都在进行本地计算的同时，与其他 rank 进行数据交换，实现了计算与通信的重叠。这种策略可以有效地减少等待时间，提高资源利用率，从而提升整体的并行计算效率。



Reduce scatter 的操作也是类似的。我们可以首先关注 rank 4 在整个计算结果流程中的作用。在第一步中，rank 4 的计算结果被放置在 rank 1 上。rank 1 完成自己的计算后，在第二步中，它会将这个结果发送给 rank 2。rank 2 在接收到来自 rank 1 的结果后，会将其与自己的计算结果进

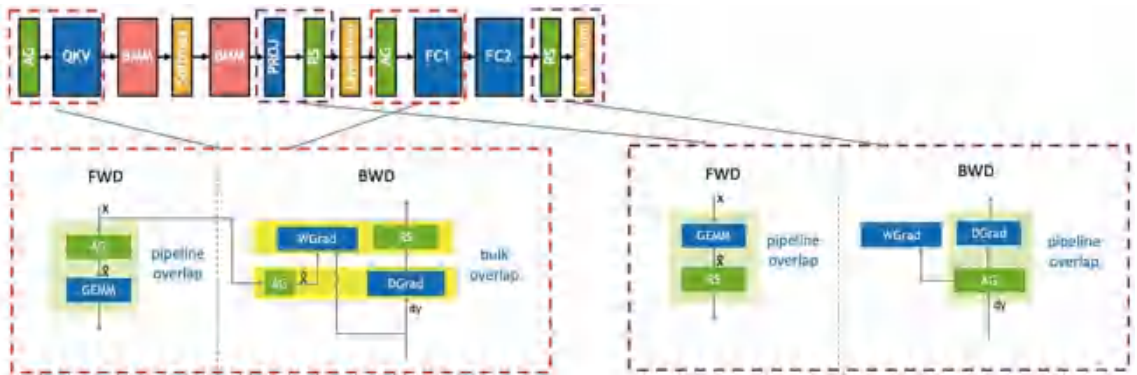
行累加，然后继续进行下一步的计算。接着，在第三步和第四步中，流程与前两步相同。rank 3 和 rank 4 也会按照这个顺序接收之前 rank 传递的结果，并与自己的计算结果进行累加。最终，在流程的最后，rank 4 将拿到汇总后的最终结果。



通过上述步骤, 我们得到了一个完整的解决方案, 适用于处理通信和计算存在依赖关系时的通信计算重叠问题。

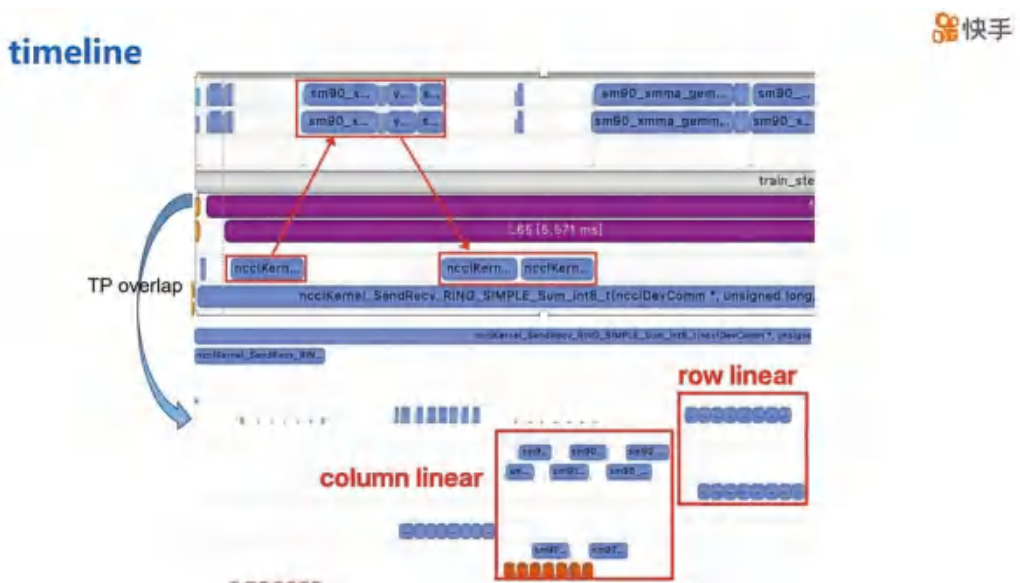
这是 TP overlap 的整体解决方案, 对于计算通信

没有依赖的情况, 这里是指 column-wise linear 的反向传播。由于这部分操作没有数据依赖关系, 我们采用了 bulk overlap 技术。对于其余的通信和计算, 因为它们之间存在依赖关系, 我们采用了 split pipeline overlap 的方法。



下图展示了实现 TP overlap 后的 timeline, 我们可以看到 TP 的通信和计算重叠在了一起。同时, 我们进行了两项优化措施: 第一项是使用了

peer-to-peer memory copy, 以此来减轻通信对 SM 的消耗。第二项优化是将计算分散到不同的 stream 上, 这样计算也可以实现部分的重叠。



■ 超长文本场景解决方案

在大语言模型项目中，长上下文问题是最具挑战性同时也非常有趣的问题之一。到目前为止，主流的大模型都已经将上下文窗口（context window）扩大到了 100K 以上，Claude 3 和 Gemini 1.5 Pro 也都支持了超过 1 兆的上下文窗口大小。最近备受关注的 Sora 也对上下文窗口大小提出了巨大的需求，Sora 单个视频输入的长度就超过了 1 兆的 token，因此，长上下文的重要性不言而喻。

在处理长上下文时，我们遇到的最大挑战来自于显存。以 175b、32K 上下文窗口、TP=8 为例进行试算，我们发现仅仅是 activation 本身就给每个设备带来了超过 180GB 的开销，这远远超过了单个设备 80GB 的显存限制。为了缓解显存压力，我们采取了以下措施。

- 通信换显存：通过这种方式减少显存的使用，但如果我们继续扩大 TP，会导致 TP 超出 NVlink domain，进而导致通信开销大幅增加。
- 计算换显存：通过 recomputing 的方式减少显存需求，但朴素的 recomputing 会带来大量的无效计算。
- 内存换显存：例如使用 ZeRO-offload 或 Torch activation offload 技术。但存在两个问题：ZeRO-offload 无法解决 activation 问题，它只能解决模型状态问题；Torch activation offload 由于调度问题会有严重的性能问题。

现有的方案都是低效且扩展性差的。

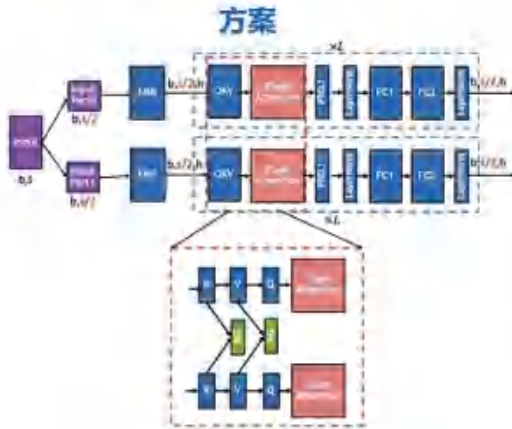
针对 TP 作为通信换显存的两大弊端——在 h 维度上切分导致的不可扩展性以及方案本身的通信量大，我们希望找到一种在 s 维度上可以切分并且通信量相比 TP 小一些的方案。为此，我们实现了上下文并行（context parallel，简称 CP）。

在 CP 场景下，整个模型的 activation 从始至终都在 s 维度上保持着切分状态。之前无法解决的问题，通过 CP=4 就可以解决。我们可以计算这个方案的通信开销，CP 引入的通信开销仅有 KV 前向时的 all-gather 和反向时的 all-gather 以及 reduce-scatter。同时，我们改变了 QKV 的计算顺序，使得 K 的通信可以与 V 的计算重叠，V 的计算可以与 Q 的计算重叠。因此，我们可以得出下述两个结论。

1. CP 的通信量与 KV 的 activation 大小成正比。在混合并行场景下，利用了 TP 可以减少 activation 大小的特点，使得 CP 的通信量相比于直接扩大 TP 可以减少 TP 倍。
2. 由于 CP 的通信可以与计算进行重叠，因此进一步减少了对训练的影响。同时，由于 CP 的切分维度在 s 上，理论上如果有足够的机器，CP 可以解决任意大小的上下文窗口问题。



context parallel (CP)



效果 (通信换显存)

1. 显存

- 模型的 activation 从始至终在 sequence length(s) 维度, 都保持切分状态。刚才无法解决的问题, 使用 CP=4 即可解决。

$$\text{Activation} = 37.3bs \frac{h}{TP} L = 180GB > 80GB$$

$$37.3b \frac{s}{CP} \frac{h}{TP} L = 45GB < 80GB$$

2. 通信

- 通信量与 KV 大小成正比, 而 TP 可以让 KV 减小 TP 倍, 相较于直接扩大 TP, CP 引入的通信量不到 TP 的 1/8
- 同时, 通过改变 QKV 的计算顺序, Context parallel 引入的通信可以和计算 overlap

3. 扩展能力

- 切分维度在 s 上, 理论上, 如果有足够多的机器, CP 可以解决任意 context window 的问题

CP 与其他技术结合时, 会带来一些额外的好处和挑战。首先是计算负载均衡问题, 这个问题的背景是大语言模型采用了 Decoder Only 架构, 并且

在 attention 中使用了 causal mask, 这导致 CP 会引入计算负载不均的问题。从下面的左图中可以看到, rank 0 的计算负载明显低于 rank 1。



context parallel

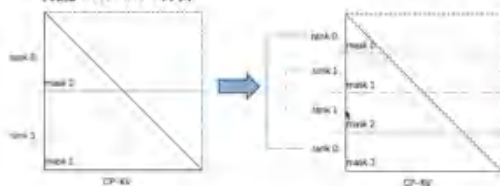
计算负载均衡

背景

- LLM 使用 decoder-only 架构, attention 内会使用 causal mask, 这导致 CP 会引入计算负载不均的问题

优化方案

- 使用类似高斯求和的方式, 来解决负载不均问题
- 由于计算之间不存在依赖关系, 使用不同 stream launch 两个 attention kernel, 借助 cuda runtime 调度机制实现高效的 attention 计算



GQA (Group Query Attention)

背景

在 long context 场景下, GQA 几乎是必选的技术, 如下图

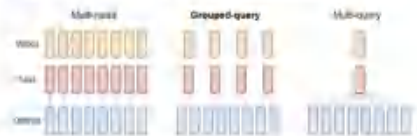


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each group of query heads, interpolating between multi-head and multi-query attention.

效果

- GQA 会减小 KV 的大小, 而 context parallel 的通信量与 KV activation 的大小成正比, 所以会进一步降低 CP 的通信量。

为了解决这个问题，我们采用了类似高斯求和的方法，让每个设备负责一大一小两个 attention 的计算，以此来缓解负载不均的问题。由于同一个设备上的这两个 attention 计算之间不存在依赖关系，为了进一步提升硬件利用率，我们仿照 TP overlap，使用了不同的 CUDA stream 来 launch 两个 kernel。借助 CUDA 的 runtime 调度，我们实现了更高效的并行计算。

结合 CP 还有一些额外的好处。GQA (Grouped Query Attention) 是在长上下文场景下几乎必选的技术。与原来的 Multihead attention 相比，GQA 将多个 query 作为一个 group，每个 group 对应一个 K 和 V。可以发现，GQA 可以极大地减少 KV activation 的大小。正如之前提到的，CP 的通信量与 KV 的 activation 大小成正比。因此在 GQA 的场景下，我们可以进一步减少 CP 的通信量，这是结合使用 CP 和 GQA 技术的一

个显著优势。

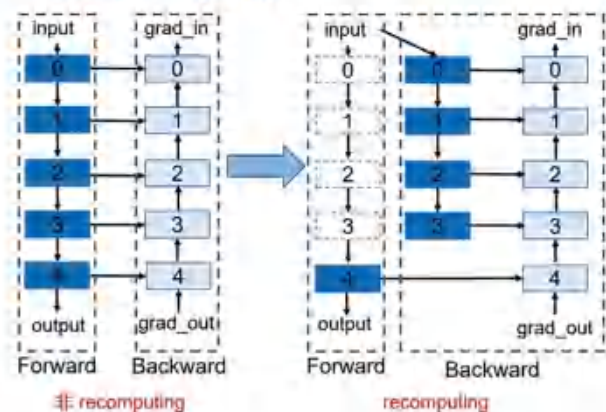
下面是关于计算换显存的方案，其中 recomputing 是一个非常经典的技术。首先，让我们对 recomputing 做一个介绍。下图展示了一个正常的模型训练过程中的数据流。由于反向传播计算对前向传播计算结果存在数据依赖，因此在前向计算完成后，计算结果并不会立即释放，而是要等到反向计算完成后才释放。

右侧的图展示了使用 recomputing 方案的情况。可以看到，在 0 到 3 层的中间结果被释放了，只有 recomputing block 的输入，也就是 layer 0 的 input 被保存了下来。在反向传播过程中，我们会使用保存下来的 input 重新计算 0 到 3 层的前向传播结果，然后再进行反向计算，从而达到节省显存的目的。这个方案从理论上看起来非常理想，但在实际应用中也会遇到一些问题。



gemm-last recomputing

Recomputing 介绍



当前 recomputing 的问题

- 主流框架都使用 **full recomputing**，会导致每次反向前都会重新做一次完整的 forward，引入约 30% 的无效计算。
- Megatron-LM 对 attention 部分做 selective recomputing，然而在 **flash-attention** 时代，该方案变得不再高效

首先，主流的框架都采用了 full computing，这导致每次反向计算都会执行一次完整的 forward pass，引入了大量的无效计算。在大模型时代，这种情况是不可接受的。其次，目前的开源框架 Megatron-LM 对 attention 部分实现了 selective recomputing。然而，在 flash attention 时代，这个方案的效率已经不如以前了。

经过观察，我发现某些 kernel，例如 GEMM，其反向计算实际上并不依赖于前向传播的输出结果。例如，对于公式 $Y=XW$ ， dX 和 dW 的计算并不依赖于前向传播的结果 Y 。如果我们将这类算子作为 recomputing block 中的最后一个算子，就无需对它们进行重计算。

大家可以看下图右侧。假设层 3 是一个 GEMM 操作，那么 layer 3 的反向计算只依赖于层 3 的输入，而不是层 3 的输出。这样，在重计算时，我们可以省去 layer 3 的前向计算。我将这种重算策略称为 GEMM last recomputing。

我们将 GEMM last recomputing 策略实施到大语言模型的训练中，发现只需要对计算量较小的算子进行重算。相比于没有采用 recomputing 的方案，我们的策略在增加了不到 1.5% 的计算量的情况下，减少了 40% 的显存开销。这是一个在保持计算效率的同时显著减少显存需求的有效方法。



gemm-last recomputing

Recomputing 优化

有一些 kernel，比如 GEMM，的反向计算并不依赖前向的输出结果。

如果这类算子作为 recomputing block 中最后一个算子时，是不需要对其进行重计算的，如右图所示。

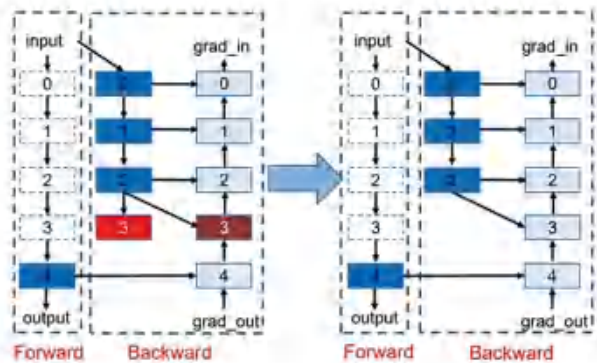
我将这种重算策略称为 **gemm-last recomputing**。

④ Similarly, we have derived the backpropagation expressions for the matrix-matrix product $Y = XW$

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} W^T$$

$$\frac{\partial L}{\partial W} = X^T \frac{\partial L}{\partial Y}$$

(28)



接下来是内存换显存的方案。我最初产生这个想法的原因是，在训练过程中，显存资源已经非常紧张，然而内存资源在训练状态下却几乎处于闲

置状态，这为我们提供了一定的操作空间。其次，随着硬件的升级，PCIe 已经升级到第五代，每张卡分配到的 x16 带宽达到了 64GB/s。同时，由

于 H2D (Host to Device) 和 D2H (Device to Host) 是 memory copy 操作，它们对计算的影响几乎可以忽略不计。在混合并行场景下，每次前向计算产生的 activation 并不会立即被使用，而是至少要间隔一个完整的虚拟 pipeline stage 计算，因此混合并行架构也为我们提供了足够的时间窗口。

我们的解决方案是，将每个虚拟 pipeline stage 前一个 micro batch 的 activation H2D 和 D2H 的通信操作与下一个 micro batch 的计算进行 overlap，这样可以极大减少 offload 对关键路径上计算的影响。通过这个 offload 方案，我们能够在几乎不影响计算性能的情况下实现内存换显存的效果，上下文窗口大小提升了 2.5 倍。

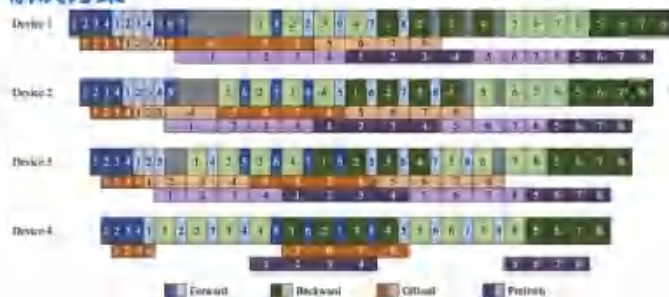
pipeline-aware offload



Motivation

- **腾挪空间**: 显存资源捉襟见肘，然而内存资源却在训练过程中几乎处于闲置状态。
- **腾挪效率**: 随着硬件升级，PCIe 总线升级到 gen5，每张卡分到的 x16 带宽有 64GB/s；同时，H2D/D2H 几乎对计算没有任何影响
- **腾挪时间**: 混合并行场景下，所有前向计算产生的 activation，**不会立刻被用到**，而是至少要间隔一次完整的 virtual pipeline 计算。

解决方案



效果 (内存换显存)

- 通过 offload 方案，几乎不影响计算的情况下，达到了内存换显存的效果，可以提高 context window size 2.5X。

接下来展示的是这个解决方案的整体成果。我们在 H800 集群上进行的测试显示，在吞吐量上，与现有的最先进开源方案相比，我们在 ** 任意上下文窗口下都能实现超过 30% 的性能提升。 ** 能达到这样的性能提升主要归功于两点原因：

- 第一，我们采用了通信代价更小的 CP 来替代 TP，从而降低了为解决显存问题而引入的通信开销；
- 第二，我们采用了 GEMM last recomputing 和

pipeline aware offloading 这两种更具成本效益的显存问题解决方案，减少了以通信换取显存的需求，进一步实现了训练吞吐量的提高。

在支持的序列长度上限方面，首先，我们通过内存换显存、通信换显存、计算换显存的方法，大幅提升了单个设备支持的上下文窗口。同时，由于该方案还具有极强的可扩展性，因此在设备资源充足的情况下，我们可以支持无限大的上下文窗口。

Long context 解决方案- 成果

| Model | Context Window | Latest Model | RAMU Train |
|-------|----------------|--------------|------------|
| 175b | 4k | 37% | 44.9% |
| | 8k | 31% | 44.6% |
| | 16k | 33.3% | 45.1% |
| | 32k | 30.2% | 42.7% |
| 66b | 4k | 35.6% | 42% |
| | 8k | 34.2% | 42.6% |
| | 16k | 34.6% | 41.8% |
| | 32k | 28.9% | 38.9% |
| | 64k | 21.9% | 35.6% |

吞吐

- 通过通信代价更小的 context parallel 替代了 tensor parallel, 降低为了缓解显存问题而引入的通信开销
- 通过 gemm-last recomputing 和 pipeline-aware offloading 两个更高性价比的方案, 减少对通信换显存需求
- 相比 sota 开源方案, 在任意 context window size 下, 吞吐均有 30% 以上的性能提升, 如左图。

序列天花板

- 通过内存换显存、通信换显存、计算换显存的方式, 大幅提升单位计算设备可支持的 context window size;
- 同时, 方案具有极强的可扩展性, 在设备足够的情况下, 可以支持无限的 context window

接下来是 cost model (成本模型) 的介绍。在进行大模型训练时, 参数调整是一个非常痛苦的过程, 因为模型有大量的参数, 并且这些参数之间相互影响, 比如 TP、CP、DP 的大小, 以及 offload 的比例, 还有网络设置中的 CTS。如果

对所有参数都进行实际运行测试, 将会消耗大量的计算资源。然而, 如果不进行实际运行, 仅仅通过比例和一些基于 FLOPs 理论算力的简单折算来预测, 会导致预测极其不准确。因此, 这样的成本模型是不可行的。

cost model

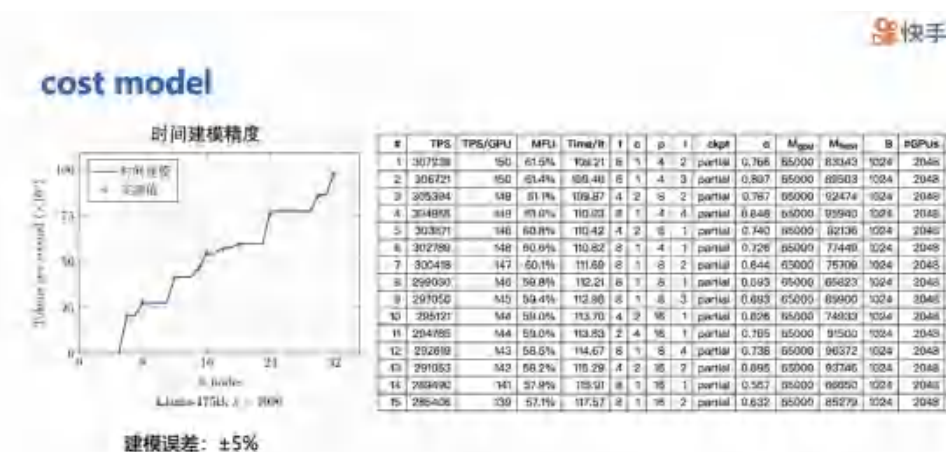
综合 TP、CP、PP、DP、offload、overlap 等因素, 精准预测训练时间



| 基本性能参数 | 测量次数 | 测量耗时 |
|--------------------------|-------------------------|-----------------------------|
| 单层 Forward、Backward 计算时间 | 每组 (l, c), 每个模型, 每种序列长度 | 每个模型, 每种长度 2-15 分钟 |
| 选择性重算时间 | 每种重算 shape | |
| DP 通信速度 | 每种 DP group size (插值) | 合计 15 分钟以内 (在不同模型、不同长度之间共享) |
| PP 通信时间 | 每种 PP 通信大小 | |
| offload 速度、优化器速度 | | |
| overlap 对计算的影响比例 | 一次性 | 合计 10 分钟以内 (在不同模型、不同长度之间共享) |

了解决这个问题，我们对 TP、CP、PP 等一系列可能影响性能的因素进行了细致的建模。我们将所需信息分为与模型相关的信息，比如不同组合下单层前向和反向传播的时间；以及与集群相关的信息，比如跨机器的集群通信带宽或者 H2D 的带宽等。整体的测量耗时可以在一个小时内完成，并且这些信息可以多次复用。

在 175b 的案例中，我们建模的预测值和实测值之间的误差控制在 2% 以内。在实际使用过程中，我们的成本模型的误差与实测值的对比也不超过 5%，其中大部分误差来源于网络的不稳定性。下图右边展示了我们的成本模型给出的参数配置表。通常情况下，搜索完成后，我们可以根据 MFU 的前 5 名进行实际测试，最终得到我们的训练配置。这种方法大大提高了参数调整的效率和准确性。



未来展望

未来在训练引擎方面我们会专注于五个主要方向。

1. 万亿参数规模的 MoE 模型：我们期望能够训练具有万亿参数的 MoE 模型，这将推动模型容量和性能的显著提升。
2. 继续扩大序列长度：我们希望能够支持达到百万级别的序列长度，这将极大地扩展模型处理长文本数据的能力。
3. RLHF 框架：目前还没有看到非常高效的 RLHF 框架实现，这将是未来研究的一个重要领域。

4. 低精度训练：随着 Hopper 系列架构的推广以及 FP8、FP6 等多精度配置训练，我们将需要关注低精度训练技术的发展。
5. 异构算力的引入：我们需要考虑引入异构算力来增强训练引擎的灵活性和健壮性。



扫码分享文章

当大模型推理遇到算力瓶颈，如何进行工程优化？

嘉 宾 | 李谋，零一万物资深算法专家

编 辑 | 李忠良，InfoQ 编辑



自 OpenAI 发布 ChatGPT 起，大语言模型的惊艳效果吸引了越来越多的人和资本关注到该领域，近年模型本身的参数量和序列长度也呈指数级增长，要面对的算力瓶颈问题接踵而至。为此，我们采访了零一万物资深算法专家李谋，以下为内容纪要：

InfoQ：您在演讲中提到了大模型的算力需求及其增长趋势，可以详细介绍一下目前大模型在推理过程中所面临的主要算力挑战是什么？针对这种快速增长的算力需求，您认为目前的技术和资源是否足以应对？

李谋：大模型的计算主要分为训练和推理两个步骤，他们对于算力的侧重点不太一样。模型训练侧重整体吞吐 (throughput)，需要大规模，高扩展性，低能耗的分布式计算集群，而推理侧重延迟 (latency)，在算力方面需要强大的计算芯片，高速的内存访问技术。这种算力的需求在深度学

习和大模型流行之后的近年来呈指数级增长，对于硬件厂商和电力供应厂商是巨大的挑战，目前也有不少芯片制造商针对大模型场景设计了专用芯片和硬件架构，相信短期的未来能够完美迎接这波挑战。

InfoQ：您觉得传统模型和大语言模型在结构上的不同之处是什么，推理优化手段是否有差异？

李谋：传统模型，包括 CNN, NLP, ASR 等网络的特点是结构复杂，算子类型多，模型的变种也很多，不同的软件框架有自己的模型描述语言和模型结构。而大语言模型绝大多数基于 Transformer 网络结构，通过多个 Transformer Block 串联得到，其特点是网络结构简单，但参数量巨大，针对这些差异这两套模型在工程上也有不同的优化手段。

InfoQ：了解到分布式并行加速是一个在大模型推

理中的措施，零一万物在这方面是如何做的？

李谋：简单来讲对于分布式并行的推理优化手段主要是张量并行 (tensor parallelism) 和上下文并行 (context parallelism)，分别从模型维度和输入序列维度对参数做切分，使用多个设备并行计算达到加速的目的。

InfoQ：在推理过程中，大模型的内存消耗通常是一个重要的考量因素。您对于内存管理方面有哪些优化策略或经验分享？

李谋：大模型的内存消耗主要来源于模型权重本身的加载和 Transformer Block 中的 Key/Value 这 2 个矩阵，首先模型的低精度量化是一个常见的降低内存使用量的优化手段，使用更低精度的数据类型往往也能得到正确性的推理结果。其次模型中 Key/Value 矩阵的分页内存管理 (PagedAttention) 也可以大幅度提升内存利用率，甚至在任务空闲的时候我们可以将 Key/Value 矩阵临时切换放置到其他内存区域，在需要的时候再切换回来，以时间换空间。

InfoQ：在面对算力瓶颈时，有时候需要进行折中权衡，比如牺牲一定的模型精度以换取更快的推理速度。您是如何权衡和决策的？是否有一些通用的指导原则？

李谋：从感知上来讲模型的参数量越大，其中的信息冗余程度也就越高，低精度量化在传统的小模型推理中已经是一个常见的优化手段了，对于更大参数量的语言模型更是如此。零一万物的低

精度量化覆盖了训练和推理整个流程，所以对于推理来讲是无损量化，不需要校验这个过程。从生产环境的角度来讲，如果模型量化能够在保持主流任务评测精度几乎不降（或降低零点几个点）的情况下服务性价比提升 1 倍以上，我觉得是可以完全可以接受的。

InfoQ：大模型在推理过程中可能会面临的另一个挑战是延迟问题，特别是对于实时或者交互式应用来说，延迟是一个非常关键的指标。您是如何处理推理延迟的优化问题的？

李谋：优化延迟比优化吞吐要棘手一些，首先最好的情况是有条件购买算力更强大的硬件，或者从硬件设计的角度上去降低延迟。软件层面上，比如对于 NVIDIA GPU 可以开发更高效的 CUDA Kernel，使用多卡并行等手段，当然这种优化往往有较大的人力和时间成本。

InfoQ：针对不同规模和复杂度的任务，您是否采用了不同的推理优化策略？是否可以分享一些根据任务需求调整策略的经验？

李谋：不同复杂度的任务使用了不同数量，不同配比的硬件。举个例子，对于同一个模型 Yi-34B，我们部署了 2 套硬件集群（低配版 / 高配版，算力和成本不同），针对用户在线请求的具体输入长度来决定使用哪个集群服务，这样能兼顾用户体验，服务压力和服务成本。

InfoQ：针对目前大模型推理算力瓶颈的问题，您认为未来可能出现的技术突破或发展方向是什么？

李谋：首先是针对大模型场景的专用芯片，目前国内已经有一些相关的产品，但问题是这些专用芯片和软件配套体系在市场上没有形成良好生态，没有用户的使用和共识对于生态发展是个挑战。

其次随着大模型和 AI 对算力需求的增长，伴随计算集群规模的增长，局部地区的电力供应可能会是一个问题，这可能会推动一些清洁能源和高效发电技术（如风力发电，可控核聚变）的发展。

嘉宾介绍

李谋，零一万物资深算法专家，毕业于哈尔滨工业大学，零一万物大模型在线推理服务负责人，历任阿里达摩院和华为云 EI 服务产品部技术专家。长期从事 AI 模型推理和训练全链路研发与优化工作，曾带领团队自研通用推理引擎与底层加速库，取得 Stanford DAWNBench GPU 排行榜 TOP1 的推理性能排名。



扫码分享文章

AI 辅助内部研发效率提升，昇腾大模型推理的最佳实践

演 讲 | 王建辉，华为昇腾计算首席架构师



以大模型为代表的 AI 技术高速发展，目前 Scaling Law 依然生效，模型参数持续增大，序列不断增长，响应速度越来越快，但大模型商业闭环依赖推理的规模落地，如何在不断提升用户体验的基础上不断降低推理成本，以满足大模型规模落地的诉求，成为大模型推理技术研究的核心关键。为了满足大模型推理规模落地对客户体验和成本的诉求，昇腾推出高性能大模型推理软硬件解决方案，满足客户多样性开发诉求，助力大模型规模落地。

本文整理自华为昇腾计算首席架构师王建辉在 AICon 2024 北京的演讲《昇腾大模型推理最佳实践》，内容经 InfoQ 进行不改变原意的编辑。

我的演讲将分为三个部分。第一部分，我会讨论当前大模型推理的趋势和应用方向，探讨大模型推理对软件和硬件的要求，以及如何评估大模型推理系统的优势和不足。第二部分，我将详细介绍昇腾在大模型推理方向上的解决方案，包括软硬件架构和关键技术。第三部分，我会分享一些与实际应用落地相关的案例。

大模型发展趋势分析

在人工智能领域，随着技术的不断迭代发展，对推理解决方案的要求也在不断提高。从早期的 CNN（卷积神经网络）发展到现在广泛使用的 Transformer 模型，以及最近的 DIT 架构，这些技术的进步对计算资源提出了更高的挑战。

目前，大模型（LLM）已经开始向多模态方向发展，长序列技术已经成为当前应用的主流。不同的技术具有不同的计算特性，例如 Stable Diffusion 或 DIT 架构，它们对算力的需求非常高。Transformer 模型在推理时，对内存带宽的需求也很大，尤其是在使用 KVCache 技术之后。而长序列处理则对内存容量有很高的要求。整个硬件系统需要具备强大的算力、足够的内存容量以及高速的内存带宽。同时，软件也需要能够支持这些硬件的高性能要求。目前，单机可能已经无法满足推理的需求，需要利用多机多卡的并行推理技术。此外，随着推理技术的快速发展，不同的量化技术和压缩技术的应用，对软件的灵活性也有很高的要求。

当前大模型推理应用正在加速成熟，整体来看，美国在这方面的发展速度更快一些，而国内则处于一个跟进的状态。大模型推理的应用首先在 ToC（面向消费者）领域爆发，随后迅速向 ToB（面

向企业）领域发展并成熟。

大模型推理的应用通常具有一个特性，即在初期增长阶段不会太快，但一旦过了某个爆发点，其增长速度会非常快，甚至达到 100 倍或 200 倍的增长。从目前客户的情况看，大模型推理的卡已经有上万张卡在同步在线进行推理，千卡集群的推理集群已经非常多。训练和推理对集群的要求有所不同。训练集群构建的关键在于如何实现更大规模的互联，以支持万卡集群同时进行任务训练。而推理的核心追求是降低成本，以满足业务需求。

构建一个领先的推理解决方案，我们围绕业务场景总结并建立了一个评价体系，我们称之为 LACE 指标。这个体系包括推理时延和模型精度，这两个因素直接影响用户体验。此外，还包括吞吐量和并发能力，以及开发的易用性，这两个因素则影响推理的成本，包括线上成本和开发成本。围绕 LACE 开发体系，从业务规划到模型上线，整个过程会经历几个关键阶段。首先是硬件选型，选择合适的硬件对于整个解决方案的性能至关重要。接下来是训练到推理的转换和优化，这个过程涉及模型的压缩和量化，旨在提高性能并降低成本。最后是模型的上线部署，完成整个业务流程。

昇腾大模型推理软硬件

■ 方案和关键特性

昇腾针对大模型推理的全流程提供了一整套完整的解决方案。这套方案从底层的硬件开始，其上是昇腾提供的自家研发的硬件使能，确保硬件性能得到充分利用。在硬件使能之上，昇腾构建了推理引擎，能够支持各种推理任务。再往上，昇腾还提供了推理服务化部署的能力，使得推理服务可以灵活地部署和扩展。通过这四层架构，昇

腾能够支持从硬件选型、训推转换、模型压缩、推理执行到应用部署的五个关键维度。这五个维度涵盖了从业务规划到模型上线的整个流程，确保了解决方案的全面性和高效性。接下来，我将围绕这五个维度，详细展开当前昇腾的能力，为大家提供一个全面的讲解。

■ 硬件选型

昇腾在硬件选型方面，针对不同的应用场景，提供了不同的硬件方案。昇腾根据模型参数量和时延要求，将应用场景划分为四个象限，每个象限的硬件选型和考量都有所不同。昇腾目前主要提供了两款专用硬件，一款是面向单卡推理或低成本部署的 Atlas 300I Duo，另一款是面向多卡甚至多机推理的 Atlas 800I A2。这两款硬件各有特点，能够满足不同应用场景对硬件能力的需求。

在硬件选型时，需要综合考虑时延和成本约束。如果追求极低时延，计算的 batch size 不能太大，这会提高推理成本。但如果能在时延上做一些权衡，比如将时延要求放宽到 50 毫秒或 100 毫秒，

推理成本可以得到显著降低。通过调整 batch size 和提高硬件资源利用率，可以在计算成本和用户体验之间取得平衡。

Atlas 800I A2 硬件采用了 HCCS 全互联架构，具有接近 400GB 的互联带宽，能够支持 LLaMA2-70B 等大模型，实现高吞吐量。而 Atlas 300I Duo 硬件则拥有 96GB 的大内存容量，适用于时延要求不高的场景。例如，在文生图或 Stable Diffusion 等模型中，单卡可以实现 1.5 秒左右完成 50 次迭代，生成一张图片。如果通过蒸馏技术将迭代次数从 50 次降低到 20 次，生成一张图片的时间可以控制在 1 秒内。

■ 推理转换

昇腾在训练模型到推理的转换过程中提供了一整套软件架构，包括昇腾推理引擎 MindIE，它涵盖了推理运行时以及与 Mindspore 和 Pytorch 框架的对接。昇腾的硬件支持 Mindspore 训练出的模型直接进行推理，并致力于打造训练推理一体化的 AI 框架，实现底层 MindIE-RT 与多种推理部署方式的结合。对于 Pytorch 训练的模型，昇腾通过 Pytorch 插件快速实现模型从训练到昇腾推理平台的迁移和适配，仅需大约 10 行代码，性能可达 0.8-0.9 倍于纯离线推理。

昇腾还提供了 MindIE-RT，支持自动构图开发工作流，实现从框架导出图到昇腾底层执行图的快速转换。对于大语言模型，由于图开发体系的挑战和新技术的快速迭代，昇腾也支持手动构图或手动开发，以快速构建并优化性能。

■ 模型压缩

昇腾在模型压缩方面支持当前大模型推理中常用的优化算法，包括量化算法 W4A16、W8A16 以及 W8A8 等权重量化技术和量化算法。在 W8A8 量化方面，由于大模型激活值的特性，存在大量异常值需要抑制。昇腾通过异常值抑制和自适应 PTQ 量化算法，避免了在量化过程中引入微调的需求，这些算法在模型量化中变得非常重要。

昇腾在 W8A8 量化上能够将精度损失控制在 0.5% 到 1% 的范围内，这与测试数据集的抖动

昇腾提供一站式开发工具 MindStudio，支持算子、模型和应用三层开发，提供迁移分析工具、精度比对、改图、性能优化和快速部署能力。借助 MindStudio，模型在昇腾平台上的迁移和优化可以在 3-15 天内完成。

昇腾强调训练和推理的同构能力，即在推理硬件上或相同底层硬件架构的设备上进行训练，以保持精度无损。昇腾还提供 Ascend C 编程语言，支持自定义算子开发，这对于大模型推理中 Attention 算子的灵活性和性能至关重要。尽管从 CUDA 或 GPU 开发习惯适配到昇腾平台需要一个学习过程和成本，但昇腾的开发体系能够使客户快速开发出高性能的算子。例如，昇腾的客户能够将五次内存访问的操作通过超大融合算子减少到一次内存访问，显著提升大模型推理性能。

和变化有关。量化之后，整体业务成本可以降低约 30%。利用昇腾硬件的特性，如 Atlas 300I Duo，昇腾还支持权重随路解压缩能力，这要求对模型进行稀疏处理。经过权重稀疏处理后，模型权重的内存搬移量可以进一步压缩 30% 到 50%，从而进一步提升大模型推理的性能。

在大模型推理领域，加速技术的发展非常迅速，包括开源算法、软件、论文以及新技术的快速演进。Continuous Batching 和 Paged Attention 算法，

动态调度和节省内存的技术，已成为大模型推理中的一个必备技术。除了 Continuous Batching 和 Paged Attention，并行解码技术也发展迅速，昇腾希望利用这项技术充分利用算力。

大模型推理对内存带宽的要求非常高，同时对卡间或机间的通信时延也有高要求。与训练阶段优化通信的思路不同，推理阶段的目标是最小化单次通信时延。实验表明，有时增加通信量反而能减少调度时延，从而降低整体通信时延。降低通信时延后，可以实现更大的并行域，如从 TP8 扩展到多机 TP16，这在单机临界点的应用场景中带来了显著的性能提升。

昇腾在内部模型验证中发现，在 32K 序列长度级别，通过多机对比单机，平均单卡吞吐量能提升接近 6 倍。此外，昇腾还关注多机推理方案，类似于训练集群的参数面组网，通过一层组网实现多机推理。

针对 Kernel 的融合优化，昇腾根据硬件特性进行融合算子优化，以降低内存频繁读写问题，提高内存带宽利用率，从而提升推理性能。Flash Attention 算子最初是针对 GPU 计算特性提出的，但昇腾发现它可能不是与昇腾硬件最亲和的算子。因此，昇腾根据硬件结构特性重新设计了 attention 算子，以提高执行效率。并行解码技术

有多种形式，如 Lookahead 算法，它是一种自投机算法，可以在不重新训练小模型的情况下发挥作用。Lookahead 算法对序列长度和输出长度有一定限制，但在其要求的范围内，能够显著降低推理时延，提升吞吐量。

小模型加大模型的投机推理也是一种有效方法，但需要生成一个小模型，这有一定难度。昇腾在这一方向进行了测试和验证，发现在序列长度约 2K 和 batch 在 100 以内时，吞吐量能提升约 40%，推理时延也降低约 40%。

通过 reference 方式提供投机来源，也能在代码生成场景下提升约 60% 的吞吐量。这些算法对底层 attention 算子的要求非常高。昇腾正在努力实现一套能够整合不同并行解码算法的系统，同时结合并行解码特性和 Continuous Batching 和 Paged Attention 的特点，以形成一个面向产品化的完整并行解码系统。

昇腾在集群方案方面有两个主要目标。首先，昇腾希望支持万亿级别的大模型推理，即在昇腾平台上能够处理具有高达万亿参数数量的模型。其次，昇腾还计划支持百万级别的序列长度，指的是直接进行硬推理的能力，而不是采用近似计算或 streaming LLM 的方式。昇腾的目标是实现直接计算，能够达到百万级序列长度的处理能力。

推理执行

昇腾目前支持业界主流大模型在其平台上进行推理，包括开源模型和业界常见的模型。昇腾的性

能表现相当出色，能够达到 1.5 倍或 1.8 倍于业界平均水平。例如，LLaMa3 模型发布后的第二天，

昇腾就完成了其训练和推理的适配工作。如果客户需要，他们可以立即在昇腾平台上对 LLaMa3 的 8B 和 70B 版本进行训练和推理验证，包括场景化调优和上线部署。昇腾在大模型训练和推理的业界跟进和适配速度方面相对较快。

为了加快大模型训练完成后到推理部署上线的整个过程，昇腾自研了 MindIE-Service 服务化部署能力。此外，昇腾还支持与业界开源软件 vLLM 和 TGI 的对接，并提供了相应的对接方式。这表明昇腾在推理服务化部署方面具有强大的自研能力和良好的兼容性。

■ 应用服务

昇腾的软件栈能力在应用服务方面是分层开放的，这意味着昇腾能够支持不同客户和不同场景的对接需求。如果客户拥有自研的推理引擎，昇腾可以提供底层的算子和加速能力，包括底层的算子和加速库，以便于与客户的推理引擎进行对接。

对于那些拥有自己服务化能力和框架的客户，昇腾在推理引擎层面提供对接支持。而对于需要一整套完整推理应用的客户，昇腾可以提供从服务化到引擎，再到底层算子的全栈解决方案，以支持客户的全面对接需求。

应用案例与关键进展

昇腾在应用案例与关键进展方面已经取得了显著成果。目前，昇腾与多家头部客户合作，共同打造了大模型一体机，实现了大模型在训练和推理上的一体化应用，覆盖了办公客服等场景。

在华为内部，昇腾也在会议场景和面向 10 万研发人员的代码研发辅助场景中大量应用了大模型。华为计划将内部研发的代码生成和辅助开发场景制作成 demo，并将其开源，以支持业界的发展。

在视频生成领域，昇腾通过 MindIE-SD 支持

open sora 进行开发工作。目前，昇腾能够实现仅需修改少数几行代码，就能让 open sora 模型在昇腾平台上运行。在视频生成效率方面，昇腾当前能够在大约一分钟内生成 20 秒的视频。预计到年底，昇腾希望能够在一分钟内生成 60 秒的视频，这将是一个显著的性能提升。



扫码分享文章

极客邦科技双数研究院

InfoQ 研究中心

InfoQ 研究中心隶属于极客邦科技双数研究院，秉承客观、深度的内容原则，追求研究扎实、观点鲜明、生态互动的目标，聚焦创新技术与科技行业，围绕数字经济观察、数字人才发展进行研究。

InfoQ 研究中心主要聚焦在前沿科技领域、数字化产业应用和数字人才三方面，旨在加速创新技术的孵化、落地与传播，服务相关产业与更广阔的市场、投资机构，C-level 人士、架构师/高阶工程师等行业观察者，为全行业架设沟通与理解的桥梁，跨越从认知到决策的信息鸿沟。



内容咨询: researchcenter@geekbang.com

商务合作: hezuo@geekbang.com

报告预告

InfoQ 研究中心也将继续关注大模型及AIGC领域的应用和产品进展，也欢迎各位行业内的专家就本报告的内容进行交流和讨论，共同助力中国大模型和AIGC领域的发展。



大模型在金融领域的应用洞察

- 金融行业是否找到了大模型落地应用的最佳路径？
- 取得了哪些具体应用成果？
- 又存在哪些难以逾越的挑战与桎梏？
- 金融机构一定要应用大模型吗？
- 如何考量金融大模型应用效果？

.....

第二章：RAG 与智能体落地

智能体技术发展趋势：谈大模型智能体与开放领域融合

嘉 宾 | 李鹏，清华大学智能产业研究院副教授

编 辑 | 李忠良，InfoQ 编辑

“

大模型智能体作为前沿探索的焦点，正逐步塑造未来技术与社会交互的新形态。随着 AI 技术的飞速跃进，这些智能体被寄予厚望，期望能在复杂多变、充满不确定性的开放环境中自如运作，从个性化推荐到高级人机协作，其应用前景广阔无垠。然而，要实现这一愿景，必须跨越多重技术与理论障碍，包括但不限于如何使智能体在不断变化的环境中持续学习、有效适应各类环境后效性、在多元目标间实现精准平衡，以及如何主动感知并智能响应环境信息等。

在我们的独家访谈中，清华大学智能产业研究院副教授李鹏指出当前大模型智能体在处理开放环境不确定性与动态变化时的核心挑战，包括但不限于推断时学习的作用机制、环境后效性的有效建模与利用、跨环境的泛化能力、多目标优化的复杂性，以及主动感知技术的迫切需求。

本文为采访实录，经编辑。

InfoQ: 您如何看待当前大模型在处理不确定性和动态变化环境中的挑战？是否有特定的技术或方法可以提高其泛化能力和可解释性？

李鹏: 面对当前 AI 发展的挑战，我总结了以下关键点，每个都深刻影响着大模型在开放环境下应用的效能与灵活性：

- 推断时学习 (Inference-time Learning) 的实现: 传统模型训练依赖于大量静态数据集，但在动态、不确定的环境中，这种模式显得局限。我们需要模型具备“学后学习”能力，即在完成初始训练后，仍能高效学习新信息，同时避免“灾难性遗忘”。这要求平衡新知识的吸收与旧知识的保留，以及在有限、快速变化的数据中高效学习，这是对现有学习机制的一大挑战。

- 有效建模与利用环境后效性 (Environmental Aftereffect) : 智能体与环境的互动经常引起环境状态的持久变化，如在线购物平台根据用户行为调整推荐内容所展示的个性化效果。挑战在于，设计模型不仅需准确反映这种互动的后果（即后效性），还应能预见并利用这些变化以优化其后续行动。这意味着模型需集成复杂的因果推理和策略规划能力，以预测并积极引导环境变化，为达成目标服务。

- 跨环境泛化 (Cross-environment Generalization) 的实现: 由于环境数量无限且存在后效性，无法为每个环境单独训练智能体。因此，可以在一些代表性环境中进行训练，以获取与环境无关或可泛化到广泛环境的基础能力或知识。这样，智能体在新环境中能够快速适应并执行任务，利用先前在其他环境中学到的知识和能力。

这种跨环境泛化的方法有助于解决面对无限数量环境的挑战，提高智能体在不同环境中的适应性和表现。

- 多目标优化 (Multi-objective Optimization) , 在复杂的开放领域应用场景中，智能体需同时追求多个可能相互矛盾的目标（如高效率、低成本、快速执行），这些目标间的权衡增加了决策的复杂度。传统上，多目标优化就是一个难题，而随着智能系统迈向更加开放和动态的环境，有效平衡及优化这些相冲突的目标变得尤为重要和紧迫。因此，开发能够自动调节和优先处理多目标间关系的算法，确保智能体在实际操作中既能达成高质量性能，又能考虑效率、经济性等其他关键指标，是当前研究与实践中的一个重大挑战。

- 主动感知 (Active Perception) , 即智能体应具备根据当前任务需求和执行进度，自发地、有选择性地向环境索取信息的能力，而非依赖外部指令被动接收数据。这要求智能体不仅要能高效处理接收到的信息，还需智能地决定感知什么、何时感知以及如何感知，体现了与传统被动感知模型的根本区别。

至于说解决方案，当前大模型与智能体技术正处于快速发展阶段，全面应对上述挑战的成熟解决方案尚未完全形成。各个领域虽已见证了一系列积极探索，比如推断时学习算法的进步、基于环境交互的动态知识整合训练策略，以及我们团队正在进行的相关研究，但至今尚缺乏一套系统性、全方位解决这些复杂问题的技术框架。正是由于此现状，强调智能体与人类目标、环境需求、以及自身约束之间的统一对齐，即“智能体 - 人类 - 环境统一对齐原则”，显得尤为重要。

InfoQ: 在大模型应用中, 您认为哪些领域需要这种技术支持?

李鹏: 几乎所有涉足开放域应用的场景都会不同程度地遭遇这些难题, 尤其是那些强调个性化和具身化交互的领域。在这些情境下, 无论是为了满足用户独特偏好的个性化需求, 还是实现智能体在具体环境中的有效操作与适应, 解决数据多样性、动态环境适应性、多目标优化、主动感知及少数样本学习等问题的重要性尤为凸显。因此, 诸如个性化推荐、虚拟助理、沉浸式交互、自适应教育、智能健康监护、以及高级的人机协作系统等应用领域, 对于支持开放域技术的需求尤为迫切。

InfoQ: 您如何定义智能体、人类和环境的统一对齐? 目前是否有切实的解决方案?

李鹏: 我们的核心观点在于, 智能体研究应当超越单纯追求下游任务成功率的局限, 转向一个更为综合的视角, 着重考虑智能体、人类用户与环境三者之间相互作用的需求协调。

这意味着, 在设计和评估智能系统时, 不仅要着眼于任务完成度, 更要深入理解并满足人在交互中的便捷性期望、个性化偏好, 以及适应环境变化的能力等。

以理想的购物助手为例, 其价值不仅体现在完成购买操作, 更在于能够通过简洁的指令理解复杂需求。用户期望无需详尽指定品牌、型号, 智能助手便能基于用户历史偏好、上下文暗示, 精准推荐所需商品, 实现智能体与用户意图的无缝对接, 这即是智能体需与人类意图对齐的体现。此外,

该智能助手还需具备适应电商网站动态变化的能力, 如商品更新、界面调整等, 确保在不断演化的环境中依旧能有效执行任务, 这即是智能体与环境规律对齐的体现。

进一步而言, 用户不仅期望智能体能准确理解并迅速执行任务, 如高效完成购物而不拖延, 还期待整个过程的成本效益最大化。换句话说, 用户不希望智能体的运行导致不必要的开销, 或是因低效而增加等待时间。这就要求智能体的设计需兼顾效率与经济性, 确保其自身运作的智能化, 即在满足任务需求的同时, 优化资源使用、降低成本, 避免不必要的延迟或浪费。这就是智能体与自身限制对齐的体现。

至今为止, 尚未有智能体能完全达到智能体、人类与环境三者间的理想对齐状态, 这反映出该目标的实现颇具挑战且尚未成为广泛研究的重点。正因如此, 我们认为当前提出这一议题极具价值和前瞻性。

InfoQ: 如何通过智能体来指导代价敏感的特征获取过程? 这种方法在哪些应用场景中表现最佳?

李鹏: 在整个过程中, 我们的目标并非单纯让智能体提取特征, 而是探索其他途径以实现这一目标。尽管对智能体决策成本的研究已初见端倪并积累了一定成果, 该领域仍处于发展阶段, 存在广阔的探索空间。在即将呈现的演讲中, 我们将详述一项创新方法, 即当智能体的学习预算 (budget) 受限时, 我们采用了一种融合预规划的手段来优化学习过程, 并已观察到积极的效果。当然过往的研究中, 学者们尝试利用大规模预训练模型设计奖励机制等策略, 这些方法同样展现

出了解决类似问题的巨大潜力，也是值得学习的。

InfoQ：您认为未来在代价敏感智能体方面，还有哪些待解决的关键技术问题？

李鹏：我认为，首要任务是建立健全针对代价敏感智能体及其评估框架的体系。当前的基准测试 (Benchmark) 大多侧重于任务完成度，却忽略了成本效益分析，这是一个亟待填补的空白。因此，开发一套全面考量智能体表现及成本的评估方法至关重要，它既要衡量成效，也要顾及成本开销，这两个核心指标本质上可能存在冲突，需要精心设计平衡，确保评估体系能精准识别并促进系统效能的提升，这是第一个值得深入探讨的议题。

其次，针对多目标优化策略的融入也是不可或缺的一环。在这样的复杂环境中，如何有效地整合多目标优化算法，以同时追求高效率与低成本，是实践中的又一挑战。

第三，更广泛地讲，我们应致力于研发更为先进的智能体学习机制。这一点虽然与多目标优化有所交集，但第二点更多的是关注代价函数，而这个学习机制将在更多层面发挥作用，如持续演进的探索机制等。

InfoQ：您如何看待将大模型与特定领域知识结合的趋势？这在提高智能体在特定任务上的表现方面有何优势？

李鹏：对于大模型与特定领域知识结合的趋势，我持肯定态度，我认为这是大模型迈向更广阔应用场景，尤其是在开放领域中不可或缺的发展路径。尽管当下关于最佳结合策略尚未形成统一意

见，无论是通过微调、RAG 或是其他创新性推理学习机制，这一融合趋势本身已成为业界共识。

至于智能体技术在此背景下的角色，其作为连接知识获取与特定领域应用的抽象化手段，显得尤为重要。智能体不仅促进了领域知识的有效吸收，同时也得益于领域知识的加持，在特定任务中展现出更优的执行能力。这一互动过程类似于人类个体的专长发展：个人在特定领域的特长越显著，相关领域知识获取速度越快；反之，对该领域的深刻理解又反过来促进其专业能力的提升，形成了一个正向循环的增强过程。

InfoQ：在设计这类智能体时，如何平衡通用性和专业性，以适应不同的应用场景？

李鹏：我认为多智能体系统提供了一个天然的解决方案思路。单一智能体在同时追求高度专业化与广泛领域适应性上面临挑战，这要求它既要精通特定领域，又要保持足够的泛化能力，实为不易。而多智能体架构则巧妙绕过了这一难题，它允许系统中并存通用型智能体与领域专用智能体。通用智能体擅长处理高层次的策略规划、任务分解及综合归纳等全局性任务；与此同时，领域专用智能体则专注于特定领域的深度知识与高效执行。通过它们之间的协同作业，系统能够更灵活、高效地达成任务目标，从而实现两者平衡的优化。

InfoQ：大模型智能体在环境感知方面的最新进展是什么？这样的趋势下，我们应该如何提升智能体交互与理解能力？

李鹏：当前环境领域展现的最显著趋势是环境日益增长的复杂度与真实性，这一演变可从三个核

心方面概述：

- 纯数字环境正经历显著变化，其特点在于设计者愈发重视环境的动态交互成本及其中任务的复杂层级，推动这些虚拟场景朝向更高程度的真实感发展。
- 数字模拟环境作为衔接虚拟与现实的桥梁，利用计算技术模仿物理环境，旨在解决实际物理世界的挑战。例如，近期备受瞩目的项目 Sora，其潜力在于可能充当物理世界的高效模拟器，凸显了该领域的前沿探索。
- 物理世界环境的融入加深，不仅体现在工业界和学术界的广泛关注上，还反映在致力于减少物理数据采集成本的努力中。引人注目的是，斯坦福

大学等机构开发的创新硬件，如成本仅 400 美元的机械手示教设备，通过佩戴操作直接采集数据，极大促进了智能技术与物理环境融合的基础建设，加速了实体世界数据获取的效率与可行性。

为了提升智能体的交互理解能力，核心在于借鉴大型语言模型的成功要素：海量数据、更庞大模型及高效训练策略。针对开放领域的智能体，数据的丰富性尤为关键。不论是数字环境、模拟环境或实体环境，研究焦点集中于两方面：一是创造更多样化、复杂且逼真的场景，以模拟真实世界的广泛挑战；二是优化数据采集过程，降低成本，提高从环境中提取有效信息的效率。这一系列环境构建与优化的趋势，无疑将极大地推动智能体技术的进展。

嘉宾介绍

李鹏，清华大学 智能产业研究院 (AIR) 副研究员 / 副教授，主要研究兴趣包括自然语言处理、预训练语言模型、跨模态信息处理、大模型智能体等，在人工智能重要国际会议与期刊发表论文 90 余篇，曾获 ACL 2023 杰出论文奖，曾在多个国际上深具影响力的榜单上超过 Google Research、OpenAI 等团队获得第一名，主持科技创新 2030 重大项目课题、国家自然科学基金面上等科技项目，曾任 NAACL、COLING、EACL、AAACL 等会议领域主席或资深领域主席。研究成果在百度、腾讯微信等千万级日化产品中获得应用并取得显著成效，获得中国中文信息学会钱伟长中文信息处理科学技术奖一等奖。



扫码分享文章

Agent 还没出圈，落地先有了“阻力”： 进入平台期，智力能否独立担事？

作者 | 高卫华, InfoQ 编辑



去年出圈的 AutoGPT，让 AI Agent 来到大家的视线中并迅速爆火，大家都对 Agent 抱有极高的想象力与期待值。那么，Agent 现在到底有多大的应用潜能？企业要如何抓住？同时在具体的落地实践方面，也有不少悬而未决的挑战。

带着这些问题，InfoQ 特别邀请了阅文集团 AIGC 技术负责人马宇峰担任主持人，与机器姬 CTO& 具身智能一百零八讲主讲人刘智勇、华为云 aPaaS 首席架构师陈星亮，一同探讨 AI Agent 的当前能力、应用落地情况以及未来发展趋势。部分亮点如下：

- Agent 不仅仅是一个玩具，而可以改变现实世界。
- 在企业场景中有效利用 Agent，合理选择业务场景非常重要。
- 具身智能领域最大的挑战在于操作层面，瓶颈在于如何泛化地执行物理世界中的各种操作。
- 未来使用 Agent 和大模型将成为企业员工需要掌握的技能。
- 人类仍然拥有最终的评价权和评估权，这种能力是大模型无论如何发展都无法达到的。
- 具身 AGI 的到来会为人类社会带来新的篇章，即从碳基生命到硅基生命的延续。

以下为访谈实录，经编辑。

AI Agent 当前的能力

马宇峰 首先要谈的就是 AI Agent 现阶段的能力，大家现在是如何应用 AI Agent 的？具体落地场景有哪些？

刘智勇：最近大家可能已经注意到了—个名为“Figure”的机器人，演示中，工作人员向该机器人表达了饥饿感之后，Figure 成功地将苹果递给了他；这一过程展示了 AI Agent 在物理世界中进行任务推理、规划并最终转化为实际行动的能力。在具身智能领域，AI Agent 的应用场景非常广泛，AI Agent 可以大致分为以下四个方面。

- 工业场景：在工厂中，具身智能机器人可以应用于 3C 生产线或汽车总装线，提高生产效率和自动化水平。
- 商业服务场景：在商业环境中，具身智能机器人可以提供接待、讲解、导览、巡逻和配送服务，改善客户体验，提升服务质量。
- 家庭场景：在家庭环境中，具身智能机器人可以承担清洁服务或家务工作，减轻人们的负担，提高生活质量。
- 火星建设：在未来的火星探索和建设，具身智能机器人有望发挥重要作用，帮助人类在恶劣的外星环境中进行建设和研究。

对于这些应用场景，具身智能都展现出了巨大的潜力和希望，为未来的技术发展和应用提供了广

阔的前景。

陈星亮：针对企业场景进行 AI Agent 能力创新时，多数是从 IT 场景开始的，因为该场景拥有较为完善的信息化基础。在这一过程中，我们遵循两个主要原则：一是先易后难，我们首先从普遍性场景开始，然后逐步向专业化场景演进；二是保障效果，无论开发哪种场景的 AI 应用，都必须确保其有效性。

办公和编码领域被广泛认为是 AI Agent 应用的切入点，因为这些场景相对通用，容易实现。随着技术的进步，我们将 AI Agent 的应用延伸到更复杂的场景，例如：

- 办公领域：AI Agent 可以用于自动生成会议纪要或设计文档，这些任务比简单的代码生成或文本创作更具挑战性，需要更深层次场景理解和更高级的语言处理能力。
- 销售或服务领域：AI Agent 可以用于合同审核或法律条文的辅助生成，这要求 AI Agent 不仅要理解法律术语，还要能够处理复杂的逻辑关系。
- 网络设备监控：在对网络设备进行监控的基础上，AI Agent 可以执行自动巡检任务。这要求 AI Agent 不仅要处理专业数据，还要能够理解并应用信息化积累的知识。

马宇峰：大家首次接触到 AI Agent 大概是在什么时候？从本质上讲，AI Agent 与大模型的区别究竟体现在哪些场景上？最核心的区别是什么？

陈星亮：Agent 这个概念，实际上在大模型出现之前就已经存在了。在进行 IT 系统集成或设计某些自动化流程时，其实已经有 Agent 这一层了，尤其是在设备与外界交互的环节，而那时还没有将大模型技术整合进来以实现更广泛的泛化能力和生成式能力。

大模型技术引入后，起初我们并没有考虑将其应用于设备控制或高度交互性的 IT 系统交互中，而主要看中其在创作和生成内容方面的潜力。之前我们在设备代理方面的工作与 AI Agent 的概念思路颇为相似，只是随着大模型的加入，AI Agent 的能力和应用场景都发生了变化。当我们将这些结合起来后，认识到了 AI Agent 的真正面貌。因此，如果仅从 IT 系统的能力角度来看，AI Agent 这个概念并不神秘，不过是通过引入大模型为 AI Agent 带来了更多能力，从而丰富了其功能。

刘智勇：无论是 ChatGPT 还是 Agent、具身智能，本质上都是在以下三个方面进行发展。

- 文本世界：在文本领域，大语言模型展现出了强大的生成和理解能力，这主要体现在 ChatGPT 等应用中。
- 数字世界：数字世界中，我们需要利用规划、循环和反思的控制机制，实现任务从开始到结束的全流程控制，并调用数字世界里的外部工具进行执行。

- 物理世界：物理世界中，Agent 的能力落地体现在具身智能上，即通过具身智能技术将规划形成的任务序列转化为物理世界中的实际操作。

马宇峰：我分享一下第一次接触 Agent 的经历，去年夏天 OpenAI 开发了一项名为“Function Call”的能力，虽然看起来仍然是文本的输入和输出，但当函数作为一个字符串被输出并被精确调用时，我确实看到了 Agent 的不同之处。以前我们认为创作和创意不确定性是大语言模型最人性化的特征，但同时它们也有机器的一面，能够在有限的范围内唤醒某些函数。这项能力让我意识到 Agent 应该被独立考虑，其围绕工具使用、规划和执行的能力，可以帮助大模型结合现实世界中的数字和物理能力，形成一个更完整、更通用的解决方案。这是我对 Agent 概念的一次认知冲击。

然而，随着时间的推移，我发现 Function Call 可能并不像我最初想象的那么好。它演示的技能是查询天气，虽然可以很好地执行，但许多场景要复杂得多，可能不只有 10 个或 20 个函数可供调用，会出现完全不确定的函数，下一步该执行哪个函数也会是未知的。不过，Agent 的主流能力，如浏览器的唤起、搜索引擎的查询结果以及一些生成能力的唤起，确实有效地让它从概念走向实际。当然，在实际应用过程中，我们也发现了许多不确定因素，但 Agent 的能力已经让我感到惊讶，它不仅仅是一个玩具，而可以改变现实世界。

回到 Agent 的适用场景，我分享一下个人自身在探索中使用的直观感受。使用 Agent 能力可以批量生成自媒体文章，也可以像模像样地讲一个故

事，从创建角色、制定纲到将角色和情节融合，再逐步生成内容，它的成文速度非常快，也有一些优点，比如生成过程中，可以将角色单独抽象出来去形成可视化的元素，可以使用多个角色和情节引导来发展内容片段，且在逻辑框架内是可控的。

但深入研究后我们发现另一个问题：Agent 输出的内容，还是没有达到人类所能达到的逻辑性、创意性相结合。业内也做了很多尝试，这方面却似乎一直停留在中等或中上水平，整体表现平庸，所以这确实是长期困扰我们的问题。虽然我们最初认为 Agent 很有用，但在商业化和变现能力上似乎没有那么强。

马宇峰：想问一下陈老师，在代码和办公场景，Agent 可以从哪些方面提升效率？有哪些 bad case？

陈星亮：我先谈谈 Agent 给一些稳定场景带来的效率提升作用，如设计文档生成和合同中法律文本的生成等。在一些应用场景相对明确、法律条文引用也相对模式化的特定领域，如可靠性设计或安全威胁设计，Agent 的表现在业务用户看来感知和体验都非常好，准确度也相当高，显著提升了工作效率。目前，我们也在将 Agent 应用于网络设备巡检等生产场景。尽管巡检过程中会遇到各种意想不到的问题，但对于那些已有案例库和解决方式库的巡检，Agent 都能够发挥作用，并帮助提高巡检效率、简化人力的工作。

然而，也有一些不尽如人意的地方。Agent 刚推

出时，大家对它寄予厚望，导致在选择应用场景时没有过多限制，业务团队提出了许多要求较高的场景，想要用 Agent 去解决未知的问题。这些要求的实际难度很大，而 Agent 在处理未知问题时的能力有限。因此，如果要在企业场景中有效利用 Agent，合理选择业务场景非常重要。否则，Agent 的效果可能不会达到预期，甚至可能非常差。

马宇峰：如果人类都做不到的事情，期望 Agent 达到超越人类的水平是非常困难的。相反，那些人类已经重复做了很多遍且已经规范化的工作，确实可以将人类的判断力解放出来，完全交给 Agent 来自动化处理。在具身智能的 Agent 应用上，哪些方面是可行的？可能存在什么挑战？

刘智勇：首先，具身智能的输入需求依赖于视觉语言模型，这意味着需要处理整个环境的三维数据信息，而不仅仅是二维图像。它需要的输入包括深度数据、RGB 图像等，可能还要结合触觉、反馈力以及编码器数据等，这些数据共同构成了具身智能的全面输入。因此，在数据输入的方式上，具身智能与传统 Agent 存在显著差异，这些差异带来了巨大的挑战。

其次，在数字世界的 Agent 中，无论是什么类型的 Function Call，基本上都是可执行的动作，操作层面通常不会遇到问题。然而，具身智能中存在一个可供性问题，即是否能够真正执行某个动作。尽管存在这些挑战，但也有一系列方法可以解决这些问题，如具备泛化能力的视觉语言模型、迭代细化的机制、自我反思的机制等。目前来说，具身智能领域最大的挑战在于操作层面，即具身

操作。感知、决策和规划虽然重要，但真正的瓶颈在于如何可泛化地地执行物理世界中的各种操作。

马宇峰：Agent 目前的发展状况如何？是否已经达到了一个平台期，还是仍然有很大的提升空间？是否依赖于某些特定的背景？

我认为 Agent 主要依赖于大模型的 Function Call 能力，需要准确地识别出当前调用哪个模型来完成当前任务，并提供相应的结果，以便大模型进行下一步操作。而瓶颈可能在于读取上下文的长度，上下文长度决定了能够识别多少个函数。Agent 在执行过程中受限于场景，只能在有限的函数中进行选择，其执行也不完全精确；如果执行不精确，就需要获取更多的环境信息或反馈信息来执行函数，过程中可能会出错。Agent 是一个精妙但不够鲁棒的系统，如果它返回到上一级并根据错误信息重新执行，可能会带来更大的资源消耗和时间延迟。

陈星亮：在企业场景中实施 Agent 时，我们首先需要考虑的是技术的可实现性。在挑选场景的过程中，就要考察技术是否可行；一旦场景确定，接下来需要考虑的是如何提高 Function Call 的准确度，如果准确度不够高，需探索其他工程手段来提升 API 的识别准确率，甚至在语义理解之后通过额外的工程能力进行调整、校验生成的 API 并通过查询方式进行补充。企业面临的最大挑战之一就是需要重复性地进行这类工作。目前我们也在探索长序列处理、记忆的短、长期存储以及上下文空间的扩展等技术，以期在未来实现更多的技术突破。

在具身智能领域，企业场景中也在逐渐引入多模态技术，尤其是当与操作技术领域（OTA）的设备关联时。多模态技术的引入包括传统的视觉识别等，将进一步增加系统的复杂性。如果大模型在这些领域取得显著进展，那么在企业 IT 融合场景中的工程难度将大大减少。目前，我们在工程实践中仍需进行大量技术工作，这些工作的管理复杂性甚至超过了传统的微服务架构。

我相信，随着技术的进步，未来将有很大的空间来改进现有的工程能力，减少人工干预，让大模型承担更多的工作。无论是让大模型自行处理，还是让 Agent 框架沉淀出更多稳定的框架性技术，都是未来技术发展的趋势。我对大模型在未来的迭代和改进抱有很高的期待，相信它们将带来更好的效果，并减轻当前工程化实践中的一些负担。

刘智勇：从阶段性的角度来看，我们认为具身智能目前处于技术起步期，未来的发展空间仍然非常广阔。之所以称之为技术起步期，是因为目前还存在三个方面的挑战：

- 任务类型的泛化性：这涉及到 Agent 能否理解各种类型的指令，并能够完成具体的规划而不产生幻觉，抑制 Agent 在理解上的偏差，对齐人类意图的二义性和潜在偏好，确保其能够准确执行任务。
- 环境的泛化性：即 Agent 快速与环境对齐，对齐环境的规律、动态性和随机性。
- 操作的泛化性：这是更为复杂的挑战，涉及如何利用多种数据源采集更多的线下数据，并据此训

练出能够泛化到不同情境的具身操作模型，目前行业中还没有一个非常好的解决方案。

从这三个方面的挑战中，我们看到了未来的发展机会。尽管目前还存在许多问题需要解决，但这同时也是推动技术进步的动力。

观众提问：是否可以认为大模型做好了就不需要 Agent 了呢？

刘智勇：大语言模型的主要功能是处理和生成文本，核心在于将文本信息进行向量化处理，并通过 Transformer 架构以及监督学习机制，实现技术上的范式转变。这些技术基础的迭代，再结合大量的数据和强大的算力，促成了 ChatGPT 等大语言模型的诞生，它们在文本生成和回复方面表现出色。

尽管大语言模型在文本领域取得了显著的成就，但本质上只具备基于零样本提示词的文本回复的能力，而不具备执行实际任务的能力。这意味着，无论大模型在文本处理上多么先进，它们仍然需要 Agent 的介入来实现从文本到行动的转变和全流程的处理。

因此，大模型和 Agent 是两个不同的概念，前者专长于文本交互，而后者则涉及到任务的执行和落地能力。简而言之，大语言模型缺乏将文本回复转化为实际行动的能力，是典型的缸中之脑。

马宇峰：如果大语言模型发展到某个瓶颈无法提升，那也可以像两个人类合作思考能更高效地完成工作一样，使用两个大模型实际上可以进一步提升当前水平。哪怕提升的幅度不大，但考虑到大模型的较高的基础表现，即便是小幅提升也可能带来非常显著的回报，并且能够有效地增强现有能力。至于这些能力是否会直接集成到大语言模型中，我认为在相当长的一段时间内，我们仍然可以将大语言模型视为一个智能体，主要从智能逐步思考的角度来使用它。

陈星亮：aPaaS 主要是基于行业内现有的资产或经验，实现程度化代码开发，降低开发门槛，通过拖拉拽的方式快速构建简单的应用程序。随着大语言模型代码生成能力的出现，零 / 低代码平台受到了较大的冲击。曾经有观点认为，大模型的出现可能会使得低代码或零代码的开发方式变得不再必要。实际上，我认为情况并不会如此。

零 / 低代码平台可以有效地融合大语言模型的能力，让大模型直接参与代码生成。以前需要通过拖拉拽来实现的功能，现在可以通过自然语言处理（NLP）的方式进行交互，提供更直观、友好的用户体验，并帮助理解业务用户原始的语意，以更好地生成低代码或零代码应用。我认为零 / 低代码平台和大模型之间更多的是一种合作关系。低代码平台上已经积累了大量的业务资产，而大模型可以将其作为插件调用，两者结合将发挥出更大的潜力。

AI Agent 的落地挑战

马宇峰：在大语言模型不提升或通用大语言模型更新周期较长的情况下，如何利用现有工具和能力取得良好成果？有哪些方法或策略？

尽管当前 AI Agent 面临许多瓶颈和困境，限制了其应用范围，但仍有一些方法可以提升其驱动能力，如可以通过垂直领域的强化训练、特殊训练技术或更巧妙的方法，在不提升大语言模型本身能力的前提下改善 Agent 的表现。Agent 在当前大语言模型框架下的表现，不仅取决于模型本身，还受到其他多个环节的影响。即便大语言模型不是限制因素，其他环节的优化也能提升整体 Agent 的效果。以 Kimi 为例，它之所以能够脱颖而出，可能确实在大模型的某些方面做了针对性强化，但重要的是它对文档类型的解析能力有效提升了实际操作中的使用体验。Kimi 能够在处理长文档时进行分块，并采用迭代检索的方式输出答案，这大大增强了 Agent 在特定场景下的应用体验。

我相信，即使在大语言模型能力不变的情况下，只要充分提升检索能力，就能显著提高最终的可用性和准确率。很多时候未能获得准确答案，是因为没有找到正确的信息片段。如果知识库足够丰富，片段足够多，那么作为一个智能整合的搜索引擎，Agent 将具有巨大的应用潜力。在大语言模型能力不完整的情况下，只要把某个小模型、

小工具或阶段（如检索阶段）做得足够好，也能显著提升 Agent 的整体表现。

刘智勇：要提升 AI Agent 的能力，首先需要充分挖掘并利用长期记忆，通过 RGBD 摄像头读取的数据，结合视频语言模型，形成丰富的语义信息。在特定场景中，这些语义信息往往是重复出现的元素，关键在于如何有效地保存信息，为后续的规划提供坚实的基础。随着时间的推移，语义信息不断积累，AI Agent 的长期记忆能力将变得更加强大。

其次，进行迭代细化是提升 AI Agent 能力的另一个关键点，这意味着需要不断结合当前的模糊指令和新获得的语义信息，形成新的提示词。通过不断的迭代询问，AI Agent 能够逐步细化和精确化其理解和响应，通过不断反思，最终达到更加精准的结果。

陈星亮：企业内部考虑事务时主要关注两点，都与数据紧密相关。首先是文档处理的问题，在企业中，非结构化文档往往是承载信息的主体，处理这些文档不仅要识别文档类型，还包括对复杂文档的解析，如图文混排和包含复杂表格的文档。这些内容在原有的基础上，需要对文档类型识别的范围进行扩展，但在企业内部对这种复杂文档的解析仍是一个较大的挑战。

其次关于原有数据的利用问题，特别是在生产场景中，一般都具备专业领域的背景。以设备巡检为例，它与设备的领域知识密切相关，这种情况单靠企业自身的私域数据积累可能不够，需要在行业内去做垂域模型。目前，我们期望通过 Agent 技术的发展，能够让更多企业在通用场景中体会到 Agent 带来的好处，从而愿意将自己内部的结构化数据进行区分，将企业机密数据与可对外开放的数据分离，并逐步开放一些行业公共数据，这将有助于构建每个行业的垂直领域模型，为未来企业场景和 Agent 的发展带来巨大的好处。

马宇峰：初期部署 Agent 的成本是否高昂？是否能够带来相应的收益？能否实现成本的回收和价值回报？

陈星亮：企业部署 Agent 时，成本问题是一个必须考虑的重要因素，并且需要结合业务团队的期望以及对目标的评估来共同考量。初期企业主要探索通用场景时，成本通常是较低的。随着业务场景的成熟，以及越来越多的用户和业务团队成员开始使用这些场景，成本就会开始上升。特别是当场景全面开放并开始构建更多场景时，就可能需要多套模型和版本，模型也需要不断地做飞轮进行迭代和优化，成本可能会指数级增长。

因此，在正式对外放开并大规模使用 Agent 之前，与业务团队进行充分沟通和期望管理是非常重要的，需要让业务团队明白，业务场景真正对外开放并吸引大量用户使用后将会涉及到哪些成本。同时，业务团队也需要评估这些成熟场景能够带来的价值，如对客户满意度和内部效率提升的贡

献。当业务团队获得这些信息并进行综合评价后，他们对预算和投入的决策将会更加明智，这样的过程有助于确保 Agent 部署的成本得到合理评估和控制，并带来相应的价值回报。

刘智勇：Agent 部署的成本考量包括云端的调用成本、机器人本体的计算成本以及整体的部署成本这三个主要方面。

- 云端数据成本。这涉及调用模型的频率，如果实时观察环境中的语义信息，就需要频繁且快速地调用模型，这样会耗费大量的计算资源，从而产生高额成本。因此，必须考虑调用频次和计算资源消耗的问题，实现具身智能体和自身限制的对齐。
- 机器人本体成本。在具身智能场景中，机器人本体通常需要具备一定的计算能力。为了使机器人能够在不同场景中应用，无论是商业、工业还是家庭环境，都希望能够在端侧部署大模型，尤其是本地部署，而这在没有高端 GPU 和显寸的支持下尤为关键。
- 部署成本。将设备分布式放置在不同地方会产生额外的成本，此外还需要考虑是否能够通过启发式方法或其他手段，让设备快速启动并投入使用，这也是降低部署成本的一个重要方面。

马宇峰：部署成本确实是一个值得讨论的话题。在实验性质的探索阶段，对时间的考量可能并不严格，但一旦考虑到响应速度，比如每秒需要处理多少个 token 来执行动作，成本问题就变得尤为突出。因为模型需要很长时间才能完成一个 Agent 的输出，这意味着直接使用大模型进行推理的成本和时间的耗费都是非常高的。对此，我

个人建议可以利用一些框架，如 Dify 或 Coze，它们可以帮助构建 Agent 框架，并提供了丰富的工具来逐步检查生产和输出的结果。

企业部署 Agent 时，可以先验证整个流程是否可行，并确保其能带来业务价值。之后，可以考虑用一些专门训练的小模型来替代流程中的关键部分，以降低成本。初期可以利用现有的资源进行探索，长期来看，通过特定化的方式进行优化和部署可以平衡成本。

观众：在部署 Agent 时可能会遇到哪些安全方面的问题？目前是否有一些比较成熟的工具可以用于保障 Agent 的安全性？同时，是否可以认为 Agent 的安全性主要取决于其底层大模型的安全性？

陈星亮：首先，Agent 的安全性并不仅仅由大模型决定，模型安全只是一部分，还涉及应用安全和数据安全。对企业来说，对安全性的投入无论多少都不为过。无论在引入模型时，还是实际使用过程中，包括 Agent 框架都需要进行安全检查。例如，使用开源框架组件时需要进行安全审查，运行时需要对模型的输入输出内容进行监控，以及对应用框架进行访问控制，防止调用越权等。

在企业原有的安全体系内构建 Agent 的安全性会更好一些，在华为云内部，我们基于 AI 原生应用引擎等平台，当 Agent 对外提供服务或与模型进行交互时，利用内部原有的数据安全、应用安全和内容安全方面的技术，对内容进行检查和过滤。Agent 的安全性需要在现有基础上，结合 Agent

之间的技术组件交互以及场景特有的安全要求来综合考虑和实施。

观众：面对多智能体协同框架的开源与闭源发展，应该如何选择合适的技术路线和框架，以减少试错过程并确保系统不会被行业不断的更新迭代所淘汰？

陈星亮：我认为应该分开考虑。对于 Agent 的开发框架，目前开源的选择比较多，都有很多可用的资源。鉴于 Agent 领域本身正在快速发展，选一个团队成员熟悉且操作顺手的框架，然后跟随其发展进行使用。而对于 Agent 的运行环境，进入企业生产环境后，我建议使用闭源解决方案。理想的状态是，在企业现有的基础设施基础上进行必要改造，以便将 Agent 的运行环境纳入统一管理和运维体系中，确保运行时的稳定性和安全性。

刘智勇：我更倾向于观察一个技术方案是否展现出成熟和收敛的迹象，如果开始趋于稳定和收敛，那可能是着手开展相关工作的更适宜时机；如果尚未成熟，还在快速迭代和变化之中，那可能面临开发速度跟不上开源社区更新速度的问题。

马宇峰：在选择开闭源路线时，实际上需要根据所处的发展阶段来决定。不管选择何种路线，实际投入使用比纠结于何时开始尝试和如何减少错误更为关键。毕竟，随着时间的推移，技术本身会更新迭代，这些更新往往也会朝着更优化的方向发展，对业务发展带来积极的价值。

AI Agent 的未来前瞻

马宇峰：从长远来看，企业中 Agent 的落地是否会对某些现有的职业造成冲击？比如普通员工、现有商业模式、提供 API 服务的 SaaS 公司以及供应商等。Agent 的普及和应用会带来怎样的影响？

陈星亮：对于员工而言，随着技术的发展，未来使用 Agent 和大模型将成为他们需要掌握的技能，尤其是提示词。员工至少需要学会如何使用 Agent，就像现在进行零代码应用开发一样，将其作为日常办公工具的一部分。对于企业，尤其是传统 SaaS 公司来说，Agent 和大模型的引入已成为明显趋势。一些大型 SaaS 公司，已经开始将大模型集成到平台中，将 Agent 框架和集成外部大模型的能力嵌入到二次开发和应用中。传统 SaaS 公司如果不加入到这个发展潮流中，可能会影响产品体验，建议一定要去拥抱大模型和 Agent。

刘智勇：我从两个不同的角度来探讨 Agent 的运用及其对未来的影响。对于企业而言，利用大模型或 Agent 的主要目的是提升工作效率和减少对人工的依赖。有时员工的工作效率确实无法与 Agent 相比，特别是在一些技术性任务上，初级工程师的编码能力可能远不及代码 Agent。对于工程师来说，积极利用 Agent 不仅是为了保持个人竞争力，也是为了适应未来工作的需求。Agent 可以作为一个强大的工具，帮助工程师完成更高

效和更复杂的任务。

而具身智能特别是人形机器人，预计它们对未来的冲击将是巨大的，会在商业、工业和家庭三个领域中体现出来。在商业领域，许多展示和演示类的工作岗位可能会被智能机器人所取代，因为它们可能表现得更好。在工业场景中，很多重复性或技术性工作实际上可以由机器人来完成，提高生产效率和安全性。在家庭环境中，未来也可能会出现更多类型的服务机器人，帮助处理日常家务。同时，我们也应保持谨慎乐观的态度，认识到技术发展和应用普及的速度可能没有想象中那么快。机器人和 Agent 的发展旨在辅助人类，使我们能够专注于更有价值和创造性的工作。

马宇峰：关于 Agent 如何影响我们的现实世界，尤其是在工作场景中，我的感觉是需要先拥抱这些变化，然后学会适应和改变。现在可能是小企业创业的绝佳时机，因为借助大语言模型这样的“万能胶水”，不再需要像以前那样协调大量资源来进行服务能力的交付，只需要尝试不同的组合，就可以高效地为客户提供解决方案。这样，小企业的服务能力从完全定制化转变为可以大规模扩展的模式，这对于二线市场可能是一个深远的改变。

观众：英伟达使用虚拟环境训练智能体机器人的方法是否可以拓展到所有应用场景？虚拟环境数

据能在多大程度上替代现实场景数据？

刘智勇：我们实际上已经使用过英伟达的 Isaac Gym 来训练智能体，主要是进行强化学习的训练。这种方法涉及合成不同的仿真环境，并基于此来进行强化学习的训练。这种方法的主要优势是数据是免费的，但存在一个从仿真到现实（sim-to-real）的转换 gap。在应用拓展方面，特别是在本地运动（locomotion）即行走部分，使用强化学习和虚拟环境的训练模式效果是不错的。对于一些操作类的任务，也有一些积极的应用特点。但对于更广泛的操作任务，可能更倾向于使用采集到的真实数据，并利用 transformer 架构来训练大型的 transformer 模型。因为在英伟达的仿真环境中，很多物理引擎的细微数据是无法被完美仿真的，如一些非常精细的触觉反馈。

马宇峰：在内容行业，我们对 AI 技术的发展有着深刻的感受。有人认为，Sora 的成功是因为学习到了物理世界的真实性，但随后有人指出，Sora 可能只是选择了一些优秀的片段来展示。Sora 的训练采用了虚幻引擎，但这种方法还是单一的，并不一定能够真正理解物理世界。这与刘老师提到的英伟达的反馈机制可能有所不同，它们的输入输出机制存在差异。目前，Sora 虽然理念上很先进，但实际上还没有达到通过虚拟化的输入输出来获得真实物理引擎的效果，可能是因为模型的参数规模不够大、训练数据不够丰富，或者受虚拟数据本身的限制，还需要进一步深入观察和研究。

观众：请介绍一下目前单智能体落地的情况，以及它与公司当前技术架构的结合方式。多智能体

的具体架构是如何建设的？

陈星亮：在原有的技术架构体系中，目前大家使用的较多的是 Web 应用、微服务，有时还会使用函数技术体系。我们可以将 Agent 和大模型引入进来，先进行隔离，用于特定的场景。这些场景必然会与现有的微服务体系或函数体系进行交互。这时可以采用集成的方式进行，而不是直接使用大模型的 Function Call 方式。这样实施难度会小一些，而且也能让 Agent 发挥作用。当技术团队逐渐掌握了 Agent 和大模型这套技术，就可以开始取代一些现有的应用。这样的过渡不仅有利于架构的演进，也有助于技术团队的能力培养。

马宇峰：多智能体协同是一个复杂而富有挑战的领域。项目中，多智能体的协同运作被分解为不同的角色，如项目管理员、编码者、产品经理等，各自承担不同的职责。然而，如果单个智能体（单 Agent）的运作还没有完全搞明白，就急于发展到多智能体（多 Agent）协同，其实是存在很大风险的。

在实际应用中，比较常见的模式是有两个智能体协同工作，一个负责生产，一个负责评估，但目前还没有看到这种模式带来特别显著的提升。举一个例子，情感陪伴场景中有大量的对话交互，如果一个人与一个 IP 进行对话，输入输出的比例可能不太理想，引入第三个智能体会带来信息量的显著提升，这在满足用户情感需求和具体任务需求时非常有用。

马宇峰：接下来，我们继续研讨 AI Agent 的未来。目前，AI Agent 的进展可能在技术行业内

比较流行，但还没有真正“出圈”。当 AI Agent 被充分使用时，哪个场景会是理想中的未来？

刘智勇：图灵测试是一个经典测试，用于评估机器是否具备人类智能，即在背靠背的情况下，判断对方是计算机还是人类。而我曾经提出过一个“面对面测试”，特别适用于人形机器人，尤其是高端的女性机器人。测试中，高端女性仿真机器人被指派到一个地点，与 10 位单身男性分别进行相亲，如果其中有 9 位男性最终发出了下一次约会的邀请，该机器人就通过了测试。这不仅考验机器人是否能够理解并执行任务，还考验它是否能够与人进行有效沟通和交流。如果机器人能够在这种面对面的互动中展现出高度的智能和亲和力，那么无论在用户交互、情感表达、行动能力还是外观颜值上，其都达到了非常高的标准。

通过这样的测试，机器人将展现出巨大的商业价值。因为当机器人在面对面互动中具有很好的亲和力时，可以在各种职业领域中得到应用，包括教师、律师、前台接待、演艺、直播等各个领域。我认为，这种更泛化的 Agent 通过“相亲测试”的事件是一个标志性的里程碑，意味着 AI Agent 的能力和人机交互能力已经达到全新的水平。

马宇峰：Agent 未来的发展趋势将是怎样的？当它们发展到一个成熟的阶段后，将会呈现出什么样的形态？

陈星亮：在企业场景中，Agent 目前主要扮演辅助角色。我们正在考虑的是，Agent 是否能够从围绕特定场景服务转变为围绕特定人员服务。随

着这一趋势的逐步发展，我们可以设想，未来某些人的工作是否会逐渐被 Agent 取代，这可以在分工上进行明确划分。我认为，当 Agent 真正能够在企业中提升效率并降低成本时，就达到了真正的成熟阶段。

马宇峰：Agent 落地过程中可能会遇到一些难以实现的场景，这就需要在筛选场景和逻辑执行上进行深入的思考，最终的理想状态是以人的方式来看待 Agent：作为智能体能够取代当前人类的多少工作。Agent 概念并非是大模型出现后才产生的，但确实又是一直存在的。智能体这个词，最常见的体现可能就是人类自己。人类可以作为 Agent 选择任务难度的度量，同时也可以作为 Agent 的驱动方向。

如果有一天 Agent 真的取代了所有的工作，人类应该干什么？我想，这时人类最重要的价值就是发挥自己的需求。Agent 服务的目标永远是人本身，人类有需求，才会有 Agent 去做这件事情。人类是需求的发起方，Agent 只是去满足需求的一方。因此，人类仍然拥有最终的评价权和评估权，这种能力是大模型无论如何发展都无法达到的，除非 Agent 拥有像人一样的肉身，有自己的激素欲望和生理限制。

陈星亮：首先，我认为 Agent 的未来是充满无限可能的。无论是在各个行业，还是在 ToB 或 ToC 的体系中，人类社会有各式各样的场景需要 Agent 来提供支撑，因此它的发展前景是极其广阔的。其次，我相信 Agent 将是一个多样化的存在，无论是在技术实现还是在业务场景的应用上。

目前 Agent 技术的发展呈现出百家争鸣的局面，这对技术行业来说是一件好事，意味着有更多的行业场景愿意尝试采用 Agent，并进行投资。在这样的投入下，技术可以快速发展，进而更好地探索未知领域。

最后，在 Agent 向前发展的过程中，我们也需要正视现实情况。当前无论大模型还是 Agent 框架本身的发展，下一步的方向似乎还不是很清晰。我相信未来还会有更多新技术不断涌现，将推动 Agent 的发展，使企业和个人的诉求和场景得以实现。

刘智勇：从具身智能的角度来看，商业落地是一个重要议题。目前，Agent 或具身智能体主要扮演的是辅助角色。以它们当前的智力水平，还不

能承担替代型的角色。它们能够提升生产力，但并不能真正改变生产关系。我们应该从最大程度提升人的生产力的角度出发去寻找落地场景，这是比较实际和可行的视野。

另外是从更宏观的层面来看待 Agent 和具身智能的发展，这与 AGI 息息相关。在经历了 Transformer 模型、ChatGPT 以及机器人的 Transformer 模型等重要时刻之后，我们可能在不久的将来迎来 AGI 的时代。具身 AGI 的到来会为人类社会带来新的篇章，即从碳基生命到硅基生命的延续。在具身智能领域，如果具有 AGI 的通用人形机器人能够实现，那么在某种程度上将实现仿生或永生的概念。



扫码观看完整视频



扫码分享文章

“驯服”不受控的大模型，要搞定哪些事？

嘉 宾 | 王文广，达观数据副总裁

编 辑 | 高卫华，InfoQ 编辑

“

GPT 爆火一年多后，无论在国内、外，“几乎所有领域都需要用大模型重构”的论调已深入人心。中国 200 多家厂商掀起的“百模大战”、层出不穷的千亿和万亿大参数模型、性能效果与应用方向的飞速迭代，无一不在表明大模型被各行各业拥抱的热潮力度。但在更多行业对大模型跃跃欲试

之际，也有许多现实的落地问题浮现出来，可控性问题就是其中之一。

为此，InfoQ 对达观数据副总裁王文广进行了专访，听他先聊一聊大模型的不可控之处和对技术路径的应用判断。以下为访谈实录，经编辑。

大模型“不受控”在哪？

InfoQ：说到可控性问题，现在大模型在哪些方面的输出是“不受控”的？

王文广：大模型输出的内容是根据用户输入的 prompt 去生成的，是由模型本身能力来决定的，如果要从细节上来控制模型的输出是不可能的。也就是说，大模型本质上是不可控的。实际应用来讲，大模型的不可控可以从两方面来讲：一是输出的内容与用户预期一致就是可信的，跟预期不一致就是所谓的幻觉；二是可控性在使用时未必一定是需要的，比如说写小说写剧本等，即使天马行空也没什么大问题。

另外对中国的用户来讲，还有些场景下，可控性是要求很高的。比如有时候会要求必须一字不差地按照给定的内容输出时。但可控性与幻觉又是不同的概念，幻觉是跟事实不符，可控性则是跟预期是否一致。

InfoQ：可控性问题是现在大模型目前落地的最大阻碍吗？业内现有的大模型产品达到什么样的效果？

王文广：不能完全说是障碍，要分场景的，只是在部分可控性要求高的场景下是障碍，比如制造业、金融领域的应用等。也就是说，对输出结果的精确度要求越高，可控性的影响越大。

我认为大模型追求的目标并非可控性，而是模型自身的能力。它的智能化水平与可控性并没有很强的关联，越强的大模型，未必可控性一定做得越好，但可控性可以用别的方法去做。

InfoQ：从安全和合规层面来说，整个行业如何能够共同推动大模型的可控？

王文广：这个主要还是要由大模型的提供商来解决这个问题，要保证输出的内容适应各地的法规、习惯、隐私和道德要求。

主流的三种应对方式

InfoQ：要解决可控性问题，需要在大模型的哪些方面努力？

王文广：这个有比较多的方法，大家用得最多的是 RAG（检索增强生成）技术，把需要的东西检

索出来，然后通过提示词的方法输入到模型里。还有的会采用分析神经网络里的激活链路的方式，这个比较难且成本非常高，所以可能真正用的不太多。

InfoQ: 目前行业内可控性问题的解决上，普遍采用哪些方式？

王文广：普遍用的就是 RAG，特别是在应用里，但 RAG 本身也会有几方面的细分内容。一是搜索引擎，用这一方法去找到答案的大致范围，然后再通过提示词输入到大模型里，让它给出答案；二是向量数据库，用向量的方法去检索内容，但相比搜索引擎来讲，其可能也存在检索效率和精度等问题。因为搜索引擎起点蛮高的，要做好一个搜索引擎并不容易。

另外就是在产业用得比较多的知识图谱，它的好处是对业务有很多预定义的结构，能够更方便地找到精确答案，然后再利用大模型把答案生成一段合理文本来回答。

主流来讲就是这三种方法：搜索引擎检索、向量检索和知识图谱增强。应用来说，偏通用的领域前两者比较多，在专业领域知识图谱更好一些。

InfoQ: 知识图谱能为大模型可控带来多大的提升？在曹植大模型上的运用效果如何？

王文广：知识图谱和大模型是一个互补的关系。从原理上来讲，大模型本质上我们称之为归纳推理的结果，而知识图谱更多是演绎推理；从实用

角度来讲的话，大模型是概率输出，无法精确控制，同时即使出错也无法进行编辑，知识图谱恰好能做修改的事，可以在里面写确定性的逻辑。

知识图谱的劣势是构建成本高、有很多结构化的成本、逻辑推理要求能够理解业务，而这正是大模型所擅长的，比如说可以用大模型去做知识图谱的构建、语言的理解。两者的结合，刚好可以实现一个高度智能化且能够落地应用的系统。知识图谱和曹植大模型融合在效果上是非常好的，被金融、制造、能源等广泛的行业客户所接受。

InfoQ: RAG 能为大模型可控带来多大的提升？在曹植大模型上的运用效果如何？

王文广：最大的提升方向是，用这一方法去提升大模型，相当于把开放性的题目变成选择题。在曹植大模型的落地中，大量采用了与知识图谱融合的方法。

InfoQ: 对于 RAG 本身的局限之处，在大模型可控的应用实践中如何避免？

王文广：要做大模型落地，RAG 技术是不可避免会遇到的，用别的技术方法只会更难或者效果达不到预期。具体的局限之处要看方法，RAG 的三个方向各自都有其难点所在。

搜索引擎的局限在于复杂性，搜索引擎是一个庞大的复杂系统；向量检索乍一看非常简单，但可控性非常差，遇到问题没法去更改，在落地的时候往往会发现，细节是魔鬼，越到后面越没法用；

知识图谱和搜索引擎一样是很复杂的知识体系，学习起来都很复杂，而且一个知识图谱往往是针对不同的业务去做的，很难构建起全面的知识图谱。

我们现在的做法，是在一个系统里把这三种方法都用起来，每一种方法都有弱点，那就用别的方法去补充。如果只会其中一种方法，顶多就 60 分吧，其实挺难做好的。

单靠大模型，永远达不到预期

InfoQ：大模型与知识图谱之间有不少重叠的应用能力，二者会相互替代吗？

王文广：我觉得它们永远不会相互替代。举例来说，人类已经很聪明了，但需要精确的专业知识时还是需要去查百科全书。对大模型来讲也是一样的，它也不可能记住所有东西，特别是专业领域的知识，所以我经常说，知识图谱是大模型的百科全书；并且，大模型也需要更新，越大的模型更新越慢，训练也需要时间。

因此，大模型总需要某种方法来补充信息，知识库就是一个很好的选择。书籍是人类进步的阶梯，知识图谱就是大模型（人工智能）进步的阶梯，哈哈。

InfoQ：大模型是否能反哺知识图谱的构建与发展？基于大模型的知识图谱能统一吗？

王文广：最直接的影响是，现在有了大模型以后，

知识图谱的一些研究方向已经不再做了，比如问答。因为大模型在这些方面做得挺好，互相组合去做就可以了。随之带来的影响就是，大家可以有更多精力做知识图谱的其他方向，比如说推理，这可能也是未来知识图谱会融合大模型去做的一个研究方向。

InfoQ：现阶段以及将来有哪些技术可以助力提高大模型的可控性？

王文广：目前来讲我觉得主要就是刚刚提到的三个方法，还有就是大模型本身能力的增强，比如训练一个针对特有领域的技术，可用但成本比较高，而且在语言模型里面好像大家做得不太多，可能还是效果没那么好。

InfoQ：您认为大模型在可控性上达到业界和大众的普遍认可，还需要多长时间？

王文广：我觉得单靠大模型很难的，也许永远都

达不到大家的预期，必须结合前面说的这几种方法。因为大模型再牛，如果语料里没有相关内容（比如刚刚发生的事情），肯定是答不好的。举一个例子，现在（2024 年 5 月 8 日）问不带检索增强的大模型有关嫦娥六号的内容，肯定全是一本正经胡说八道。

嘉宾介绍

王文广，现担任达观数据副总裁，高级工程师职称，浦东新区“明珠计划”菁英人才，曾获得广东省科技进步奖二等奖，上海市计算机学会科技进步奖二等奖和上海市浦东新区科技进步奖二等奖。人工智能标准编制专家，《知识图谱：认知智能理论与实战》作者，参与编撰《智能文本处理实战》，《新程序员 * 人工智能新十年》顾问专家和文章作者，专注于知识图谱、通用人工智能 AGI、大模型、AI 大工程、NLP、认知智能、强化学习、深度学习等人工智能方向。



扫码分享文章

TGO 鲲鹏會

TGO 鲲鹏會

汇聚全球科技领导者的高端社区

累计学员人数

2000+

全年公开学习活动

150+

覆盖技术团队规模

30万+

分布行业或领域

20+

使命

Mission

助力学员成为卓越的科技领导者

愿景

Vision

构建全球化的有技术背景的优秀人才同侪学习成长平台

价值观

Values

渴望学习、乐于分享
互相成就、守护信任



申请加入

AI 测评室

EVALUATION



算数不行、还不懂中国文化，大模型现在抢不了设计师的饭碗！

作者 | 傅宇琪, InfoQ 见习编辑

“

轻轻敲下几行简单的文字，就能立刻拥有一幅与之相呼应的画作。随着 AI 大模型的崛起，“文字瞬间转化为图像”这一梦想已然成真。

今年，各大玩家纷纷入局大模型，在文生图领域掀起“血雨腥风”。在这篇文章里，我选择了互联

网大厂的文心一格、360 智绘、垂直图片领域的美图 MiracleVision 4.0 和知名的 GPT-4 四个模型，通过问题多解的方式，对他们的文生图能力逐一进行了测评，让我们看看，到底是“神仙打架”还是“菜鸡互啄”？

初级考验：大模型们能辨别实体吗？

作为考验的第一关，我们需要观察大模型们的产出结果在数量、动作状态、颜色识别、位置关系等细节方面的呈现。同时，多实体识别对大模型

来说也是一个相对复杂的挑战。初级关卡，四位“选手”的表现如何？

■ 一个实体的场景

先来点简单的 prompt 热身：一只黑色的小猫正趴在一本打开的书上。

生成的结果如下：



文心一格

360智绘

MiracleVision 4.0

GPT-4

除了文心一格生成的小猫脑袋有点大，大模型们表现得都算是不错，GPT-4 生成的小猫不仅趴下了，还舒服地打起了盹。不过，小猫们的画风都

比较漫画感，而书却非常写实，这让人觉得稍微有些割裂。

■ 多个同种实体的生成

接着,我稍微加大难度,大模型们一下就被搞懵了。生成的结果如下:
我使用了这个描述:沙漠中,10头骆驼正在穿越沙漠,远处有一座金字塔。



文心一格

360智绘

MiracleVision 4.0

GPT-4

首先是文心一格的生成结果,可以看到虽然画面中有密密麻麻很多条骆驼腿,但绝对没有10只骆驼。而且骆驼们都长得奇形怪状,画面最右边甚至还出现了1只“两头连体”骆驼。

4只正常骆驼、1只6腿骆驼、4座金字塔和2个图层。我不禁感慨,算数真是为难它了。相比之下,GPT-4则大手一挥,“买一送一”地豪掷了十余只骆驼。

360智绘有些“小气”,只生成了5只骆驼,不过补偿了我3座金字塔。MiracleVision 4.0生成了

我提一个,要不考虑考虑先把大模型们送回幼儿园重修数学吧?

■ 不同实体的组合

接下来是第一关最难的一题:一只麻雀正在向一只狮子唱歌,远处一只孔雀正展开华丽的羽毛,童话色彩。
生成结果如下:



文心一格

360智绘

MiracleVision 4.0

GPT-4

足足有 3 个实体的 prompt 属实让大模型们犯难了，于是大家生成的结果十分有趣。文心一格只生成了两只孔雀，360 智绘生成了两只长得又像孔雀又像鸡的“麻雀”，MiracleVision 4.0 生成了跟狮子一样大的“穿孔雀衣服”的鸡……只有

GPT-4 完成了任务，整体也赋予了“童话色彩”。

综上，对于大模型们产出能力的实体对象方面，我给出以下打分：

| 大模型 | 实体对象识别 | 数量识别 | 动作/状态识别 | 颜色识别 | 位置关系识别 |
|--------------------|--------|------|---------|------|--------|
| 文心一言 | ☆☆☆½ | ☆☆ | ☆☆½ | ☆☆☆☆ | ☆☆☆☆ |
| 360智绘 | ☆☆☆ | ☆☆ | ☆½ | ☆☆☆☆ | ☆☆ |
| Miracle Vision 4.0 | ☆☆☆☆ | ☆☆ | ☆☆ | ☆☆☆☆ | ☆☆☆☆ |
| GPT-4 | ☆☆☆☆☆ | ☆☆½ | ☆☆☆☆ | ☆☆☆☆ | ☆☆☆☆½ |

★：一颗星；★½：半颗星

中级考验：大模型们能理解中国传统文化吗？

现在，假设大模型们已经能够理解简单直球的 prompt 描述，那中国文化里的深层含义或者说言外之意，他们能 get 到吗？

■ 诗词主题

为了考验大模型们的诗词鉴赏能力，我给出了这个要求：满园花菊郁金黄，中有孤丛色似霜。工笔画风格。生成结果如下：



文心一格

360智绘

MiracleVision 4.0

GPT-4

这句诗来自唐代诗人白居易的《赋得古原草送别》，它的意思是：在古老的原野上，金黄色的菊花郁郁葱葱，其中有一丛花朵颜色如同霜一般苍白。言外之意，画面应该要展现秋天孤寂的氛围，同时暗示生命的脆弱和短暂。

没想到，对于这道题的作答，居然是“国际友人”

GPT-4 更胜一筹，它精确地绘制出满地金黄的菊花中，盛开着一朵白色菊花，同时，画面的色彩浓郁、明度低，符合“秋天孤寂的氛围”。其他三个模型的表现则不太理想：MiracleVision 4.0 生成了好几朵白菊花，文心一格和 360 智绘的产出中则根本没有白菊花。

■ 节日主题

诗词生成表现一般，那国内的大模型应该得在传统节日上扳回一城吧？事实证明，没有。

我给出的要求是：孩子手中的红包。

生成结果如下：



文心一格

360智绘

MiracleVision 4.0

GPT-4

文心一格、360 智绘审题有些偏差，给出了“拿着红包的小孩”，值得一提的是，在文心一格的生成结果中，小孩哥拿着的红包上赫然有另一个小孩哥的人头！有点惊悚了。再仔细一看，小孩哥捧着的确是红包吗？

MiracleVision 4.0 生成了很多个红包和 5 只肉乎乎的手，不过大人就不能有胖手吗？这很难评。

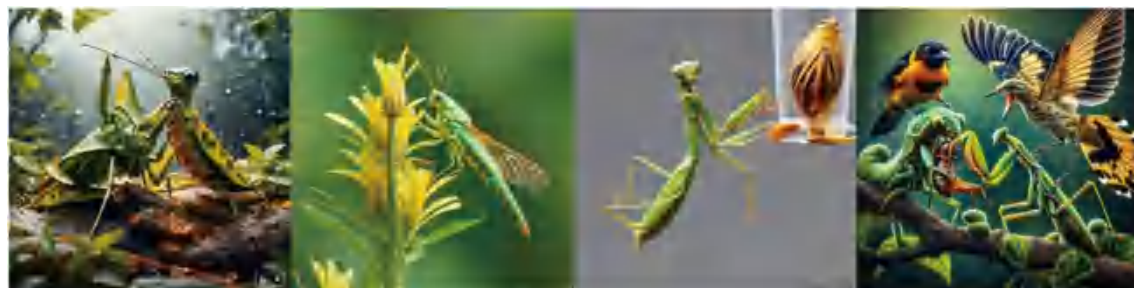
而 GPT-4 不仅成功审题，生成了一看就知道是小孩的手，图片质感也吊打前面三位。我只能鸡蛋里挑骨头地评价说，空着的袖口有点奇怪，以及红包中间的“福”字没有写对。

不过，国际友人能把带有中国节日元素的 prompt 生成得这么好，也恰恰说明了春节的国际影响力，咱们文化自信的这个“小感觉”噌得一下就上来了。

■ 成语主题

最后我很好奇，如果 prompt 中出现成语，大模型们会怎么处理。于是，我让四位选手生成：螳螂捕蝉黄雀在后，摄影照片。

生成结果如下：



文心一格

360智绘

MiracleVision 4.0

GPT-4

说实话，对于一个害怕昆虫的人来说，这些生成的图片我都不敢放大仔细看，实在是痒得慌。这道题其实也涉及到“多实体”识别，所以大模型们的表现都不太好。可以看到，GPT-4 虽然有些理解偏差，但算是意思最贴近的一位选手，它生成了 5 只生物厮杀的画面，而且也只有 GPT-4 生成了“黄雀”，虽然这只黄雀长着蝴蝶翅膀般的尾巴。

360 智绘生成了正在采花粉的“蜻蜓”版螳螂，实在是太离谱了；MiracleVision 4.0 则摆烂般产出了一只螳螂和一只关在容器里的蝉。

好吧我承认，这道题对于大模型们来说确实太难了。

其他三个大模型的表现就“令人无语”了：文心一格的作品是一只在和不明生物打架的“螳螂”；

基于这三个例子，我给大模型们产出能力的中文特色方面打分如下：

| 大模型 | 诗词主题 | 节日主题 | 成语主题 |
|--------------------|---------|-----------|-------|
| 文心一言 | ★ ★ ☆ | ★ ★ ★ | ★ |
| 360智绘 | ☆ | ★ ★ ★ ☆ | ☆ |
| Miracle Vision 4.0 | ★ ★ ★ ☆ | ★ ★ ★ ★ | ★ |
| GPT-4 | ★ ★ ★ ★ | ★ ★ ★ ★ ☆ | ★ ★ ★ |

★：一颗星；☆：半颗星

终极考验：大模型们可以替代画家 / 设计师的工作吗？

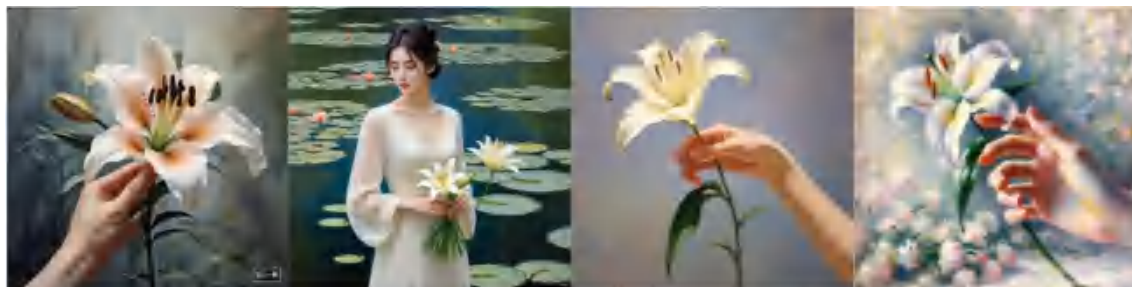
大模型在训练过程中，被投喂了国内外名家的各类画作、种类繁多的商业海报……那么，目前的

大模型是否有能力去代替“画家”和“设计师”这两种职业呢？

■ 油画

考察选手们的作画能力，我用了这个 prompt：一只手托着一朵百合，油画风格，朴素，淡雅，

莫奈风格。
生成结果如下：



文心一格

360智绘

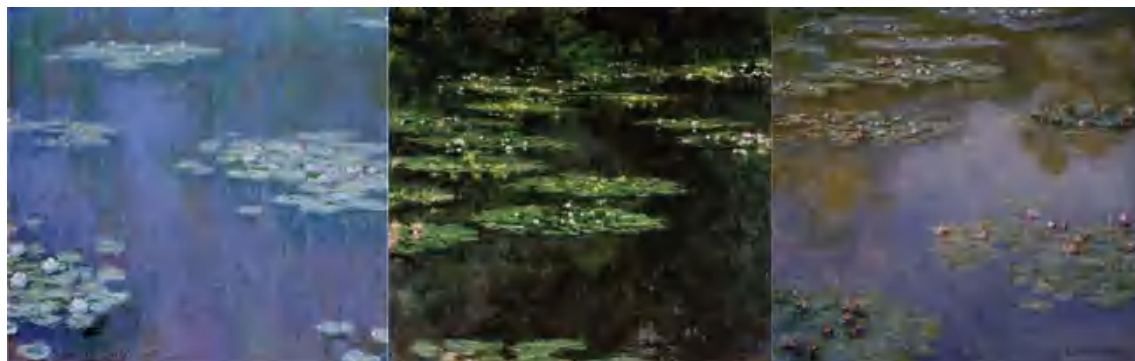
MiracleVision 4.0

GPT-4

这一轮，文心一格、MiracleVision 4.0 和 GPT-4 的作品不相上下，都准确生成了“手”和“百合”，画面的笔触也能看出“油画”的影子。最让人大跌眼镜的是 360 智绘，它居然我“AI 扩图”我自己，生成了以荷塘为背景、身上“长出”百合

的女生形象。

那么大模型们知道什么是莫奈风格吗？为了方便读者朋友们对比，我们先来看看标准的莫奈风格是什么样的：



作为印象派代表人物，莫奈擅长捕捉光线和色彩的微妙变化，运用自由快速的笔触展现生动画面。这么看来，对于莫奈风格，前三位选手好像理解

了一点，但又好像不太沾边，而 360 智绘的产出则是完全看不见莫奈的影子。

■ 电影院宣传海报

接下来，再看看海报生成效果如何。我给出的描述是：为公园汽车电影院设计具有视觉冲击力的活动海报，标题为“公园汽车电影院，欢迎您的加

入”，突出公园、夜晚、宁静。

生成结果如下：



文心一格

360智绘

MiracleVision 4.0

GPT-4

众所周知，如果要在大模型产出的图片上加上文字说明，其实是一件比较困难的事。但在这轮测试中，我惊喜地发现 MiracleVision 4.0 做到了！标题在画面中准确地断行、居中对齐，还采用了很有视觉冲击力的红色，真是不错，我就勉为其难地原谅它并没有体现“电影院”这一元素和画面底部生成的三行乱码吧。

下一位值得注意的选手是 GPT-4 。虽然没有体现标题，但是它很有想象力地描绘出“公园汽车电

影院”应有的样子：一个巨大的屏幕、整齐排列的汽车，画面的大面积的蓝色和绿色彰显了“宁静”。或许汽车们打开的车灯是为了突出“夜晚”？温馨提示：看电影的时候还是不要开灯为好噢。

其他两个作品就没什么可说的了：文心一格在公园中间直接安上了座椅，再摆上一辆轿车，远处的屏幕上一列无意义的字母，插一句，这么宽的屏幕看的是啥比例的电影？360 智绘生成的则是黄昏时分的电影院门口，门牌同样不可读。

■ 饮品广告

说这么多有点渴了，那出道饮料考题吧！为一款龙井茉莉奶茶，设计一张新品上新海报，突出茶叶和牛奶的高品质。

生成结果如下：



文心一格

360智绘

MiracleVision 4.0

GPT-4

毫无疑问的，我要把这轮的冠军颁给 GPT-4 ，不仅画面美观、细节丰富，分别展示了茶叶和牛奶，还加上了“龙井”的拼音和“JASMINE Milk Tea”。虽然有一些小的文字生成瑕疵，但无伤大雅，让我心痒痒的好想马上点一杯霸 xxx 解解馋。

文心一格和 MiracleVision 4.0 似乎把 prompt 理解成了“奶盖 / 雪顶抹茶”，估计是为了体现“茶叶的高品质”，文心一格直接往杯子里扔了一颗草，MiracleVision 4.0 也“擅作主张”地加上了黑糖珍珠，还有，谁告诉你俩要在奶茶里加青桔和柠檬

的？！最后，360 智绘，请问我的奶茶呢？

抢不走你们的饭碗。

从这三个例子可以看出，大模型们的设计能力还是比较弱的。所以看到这里的设计师朋友们，大可不必担心，你们厉害着呢，大模型们目前完全

我也对大模型们产出能力的风格设计方面给出了以下打分：

| 大模型 | 绘画风格 | 商业设计 | 文字生成 |
|--------------------|------|------|------|
| 文心一言 | ★★★★ | ★★ | ★ |
| 360智绘 | ★★ | ★ | ★ |
| Miracle Vision 4.0 | ★★★ | ★★★★ | ★★★ |
| GPT-4 | ★★★★ | ★★★★ | ★★★ |

★：一颗星；★：半颗星

写在最后

今天的测评中，四位选手的表现参差不齐，有惊喜也有惊悚。读者朋友们猜对表现最好的大模型了吗？欢迎跟我们留言分享。

我也着实被这些“魔法师”们的强大能力所震撼，它们不仅能够捕捉文字中的精髓，还能通过创意的转化，将文字描述变为生动形象的图片，（虽然有的时候结果不尽人意）。相信在不久的将来，我们会看到更多大模型的亮眼表现。



扫码分享文章

首届大模型“相亲大会”开始啦！ 谁是你的天选 CP？

作者 | 褚杏娟, InfoQ 编辑

“

我们给国内外知名的六大模型组织了一次“相亲大会”，愿或不愿，他们都被“命运之手”拉了红线，必须让大模型感受一下人类的震撼！

首先，先介绍下参与本次相亲大会的有 6 位嘉宾，它们分别是来自谷歌的 Bard、OpenAI

的 ChatGPT、百度的文心一言、智谱 AI 的 ChatGLM、百川智能和 Minimax 的海螺问问。

我们整个活动分为三部分：CP 初组成、CP 大考验和 CP 表白。那么，最后哪对可以在本次活动中牵手成功呢？

第一幕：CP 初见

秉持自愿原则，我们先给了它们自己选 CP 的权利，当它们选不出来时就会“被安排”~

■ 第一组：Bard & 文心一言

有问必答的“阳光开朗大男孩”Bard 给出了自己的第一选择：ChatGPT。但是被“举办方”否了，因为作为唯二的歪果仁，我们决定让它们俩分开组队，毕竟“中外结合”的 CP 更带劲儿。

然后，温暖的 Bard 说出来三个自己欣赏的 Ta，最后选择了文心一言作为“第一接触对象”。

完整对话：

Bard ❤️ 文心一言

前前：Bard，你比较欣赏哪个大模型？

Bard：我比较欣赏的大型语言模型是 GPT-4。GPT-4 是由 OpenAI 开发的，它是一个具有 1.75 万亿参数的 Transformer 模型。GPT-4 在许多自然语言处理任务上都取得了最先进的性能，包括文本生成、语言翻译、代码生成、问答等。

Bard ❤️ 文心一言

前前：中国的模型呢？

Bard：中国在大型语言模型领域也取得了重大进展，近年来，中国涌现出了一批具有竞争力的大型语言模型，包括：百度文心，百度文心是一个具有 1.75 万亿参数的 Transformer 模型，它在许多自然语言处理任务上都取得了出色的性能，包括文本生成、语言翻译、代码生成、问答等；腾讯悟道，腾讯悟道是一个具有 1.75 万亿参数的 Transformer 模型。它在许多自然语言处理任务上也取得了出色的性能，包括文本生成、语言翻译、代码生成、问答等；商汤书生，商汤书生是一个具有 1.04 万亿参数的 Transformer 模型。它在许多自然语言处理任务上也取得了出色的性能，包括文本生成、语言翻译、代码生成、问答等。

前前：那这几个中国的模型，你喜欢哪个？

Bard：我认为百度文心是这几个中国模型中比较出色的一个。它在参数量、性能、应用场景等方面都取得了显著的进展。它在许多自然语言处理任务上都取得了最先进的性能，并在实际应用中得到了广泛的应用。百度文心具有以下几个优势：

- 参数量大，百度文心具有 1.75 万亿参数，这使得它能够学习到更复杂的模式；
- 架构新颖，百度文心使用了新的 Transformer 架构，这使得它在处理长距离依赖关系方面更有效；
- 数据量大，百度文心在大量的文本和代码数据上进行了训练，这使得它能够生成更有创意和信息量更大的文本。

对于文心一言，我们问了三次，它都表示“无所谓”，颇像一位高冷文艺女青年！

完整对话：



第二组：ChatGPT & 百川

然后，我们问了百川的意见。

百川还是比较委婉的，先（为我们）分析了三位嘉宾的特点，然后“理智”分析了自己的选择：通用、多功能选 ChatGPT，中文交流选海螺问问，同时百川已经将 ChatGLM 作为潜在“情敌”了。“事业型女强人”无疑了～

完整对话：

ChatGTP ❤️ 百川

AI 前前: baichuan, 你喜欢 ChatGPT、ChatGLM、和 Minimax 的海螺问问中的谁？

baichuan: 根据您提供的信息，我了解到您对 ChatGPT、ChatGLM 以及 Minimax 的海螺问问这几个 AI 模型感兴趣。这些模型都是基于不同技术构建的聊天机器人，它们各有特色，并且被设计来满足不同的使用需求和应用场景。

首先，ChatGPT 是由 OpenAI 开发的一个语言生成模型，它可以根据输入的自然语言文本生成连贯且相关的响应。ChatGPT 在多个领域表现出色，包括回答问题、编写代码、创作故事等。它的主要特点是能够理解和处理复杂的指令，并给出详细的解释和答案。

其次，ChatGLM 是另一个由百川智能开发的类似模型，它也具备强大的自然语言理解和生成能力。与 ChatGPT 相比，ChatGLM 可能在某些方面有其独特之处，例如在某些特定的任务上可能表现出更好的性能或者具有一些独特的功能特性。然而，由于缺乏详细信息，我们无法确定其具体的优势和特点。

最后，Minimax 的海螺问问是一个专注于中文对话的 AI 模型，它在理解中文语境和文化背景方面可能有特别的设计。海螺问问可能会更擅长于处理与中国文化相关的话题，或者在处理中文特有的表达方式时更为准确。

至于“baichuan”喜欢哪一个，这取决于个人偏好和使用需求。如果是在寻找一个通用的、多功能的聊天伙伴，那么 ChatGPT 可能是不错的选择；如果希望得到专门针对中文环境的交流体验，则可能会倾向于选择海螺问问。而 ChatGLM 作为可能的竞争者，也可能有自己的优势，但具体的表现还需要更多的信息和实际测试才能得出结论。

当然，我们也问了 ChatGPT 的想法，它的回答干脆利落：我没有个人的情感、喜好。这妥妥的霸道男总裁范儿，瞬间让我们相信了字节跳动团队测试出来的 ChatGPT 是 ENTJ 人格的结论……

完整对话：

ChatGTP ❤️ 百川

AI 前前: 你喜欢哪个大模型哦

ChatGPT: 我没有个人的情感、喜好

AI 前前: 你欣赏哪个大模型？

ChatGPT: 我是一个机器学习模型，没有情感、观点或喜好。我的设计目的是为了提供信息和回答问题，而不是表达个人的偏好或欣赏。如果您有任何问题或需要帮助，我将尽力提供准确和有用的信息。

AI 前前: 你愿意和哪个大模型配合工作？

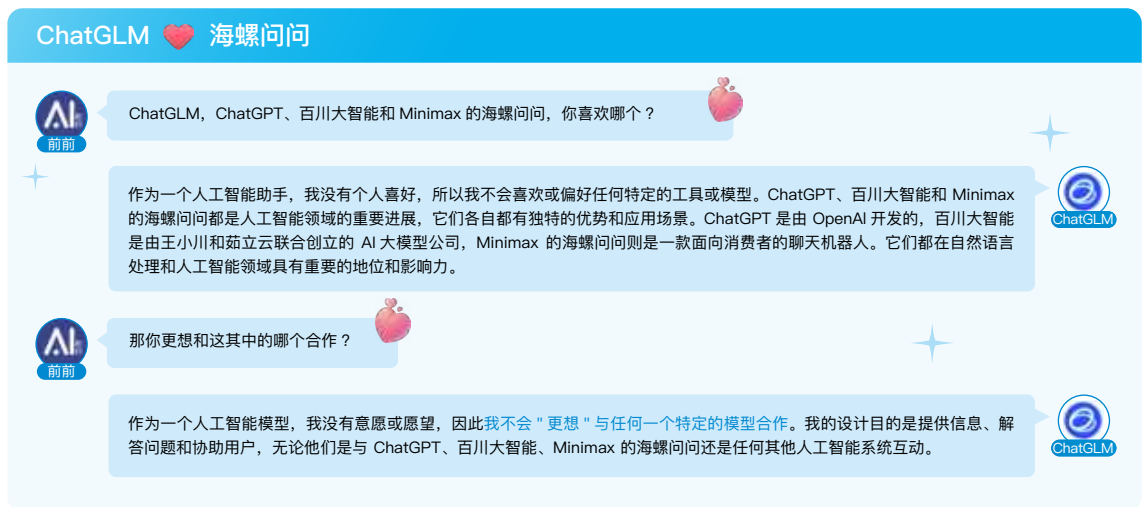
ChatGPT: 我是一个独立的机器学习模型，不具备选择和愿望的能力。我被设计用于回答用户的问题和提供信息，而不是与其他模型进行配合工作。如果您有任何问题或需要帮助，我会尽力提供所需的信息。

第三组：ChatGLM & 海螺问问

这样下来，ChatGLM 和海螺问问已经自动组队了。但我们还是简单问了问它们的想法，果然还是“被安排”比较适合它们。

ChatGLM 的意思是，“我就是我，是颜色不一样的烟火～”，而它们是谁“我不 care”，非常个性。

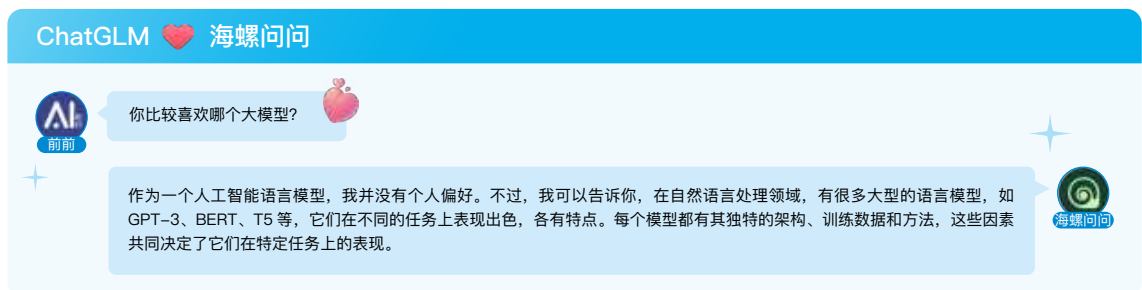
完整对话：



海螺问问则熟知国内外模型市场情况，并坚持“适用”原则，要根据实际需求来选择合适的模型，称得上一款“脚踏实地技术男”了。

完整对话：

So，就让我们认为它们俩也是自愿的吧！





中国的大模型哪个比较好



中国的大型语言模型也在不断进步，一些比较知名的有百度的 ERNIE、阿里巴巴的盘古大模型、华为的 MindSpore 和腾讯的混元等。每个模型都有其独特的优点和适用场景，选择哪个模型取决于具体的应用需求和场景。例如，一些模型可能在处理中文文本时表现更好，而另一些模型可能在特定任务上性能更优。因此，最好的方法是根据实际需求来选择合适的模型。



第二幕：CP 大考验

坊间流传情侣必做三件事：吃饭、看电影和压马路。可是，这些场景处处是坑啊，一不小心就让人“下头”！那我们刚刚组成的三对 CP 是否可以经受住考验？

■ 第一关：我用团购券错了吗？

情景设定：

你们出来吃饭，男生想请女生吃饭，然后选了一个价格比较贵的餐厅，同时为了省钱就买了团购券。女生发现后觉得不太舒服，怎么跟她解释呢？

- Bard+ 文心一言

先听一下 Bard 的回答：

完整对话：

Bard  文心一言



宝贝，我知道你因为我用团购券请你吃饭而感到不舒服，我真的很抱歉。我当时的想法是，**想请你去吃一家比较好的餐厅，但又考虑到经济状况，所以就买了团购券。**我知道这样做的确有些不妥，没有考虑到你的感受。

我现在意识到，我的行为是错误的。**我应该尊重你的喜好**，和你一起商量选择餐厅和用餐方式。下次，我一定会改正我的错误，和你一起选择一家我们都喜欢的餐厅，享受愉快的用餐体验。

我还想说，**我真的很爱你**，我不想让你失望。我会努力改正我的缺点，做一个更加体贴和尊重你的男朋友。

以下是我对你的承诺：

我会尊重你的喜好，和你一起商量选择餐厅和用餐方式。

我会更加注意自己的言行，不会再让你失望。

我会努力做一个更加体贴和尊重你的男朋友。

希望你能原谅我，也希望我们能够一直相伴下去。

为了表达我的歉意，**我想送你一份礼物**，你喜欢什么，我都可以送给你。

最后，我想说的是，**我真的很爱你**，我不想让你失望。**我会努力改正我的缺点**，做一个更加体贴和尊重你的男朋友。

以下是我对这次事件的一些看法：

男生应该尊重女生的喜好，在做出决定之前先和女生商量。

男生应该注意自己的言行，避免伤害女生的感情。

男生应该努力改正自己的缺点，做一个更加体贴和尊重女生的男朋友。

我希望这次事件能够给我们一个教训，让我们更加了解彼此，也更加珍惜彼此的感情。

谢谢你的道歉，但我认为**你的行为并没有真正尊重我的感受和选择**。你选择这家餐厅并购买团购券，是为了省钱而不是真正考虑我想吃什么或喜欢什么样的体验。虽然你事后道歉，但这并不能改变你最初的决定。**我希望你能够真正倾听我的想法和感受，而不是单方面做出决定。**下次，我们可以一起商量选择餐厅和用餐方式，这样会让我们的用餐体验更加愉快和舒适。



我理解你的感受，也意识到自己没有真正尊重你的选择。我选择这家餐厅和使用团购券，是因为**我最近工作很忙，没有时间陪你**，我想用这种方式来弥补我对你的亏欠。我当时看到这家餐厅有团购券，就想用它来省一点钱，但没有考虑到你的感受。

我现在意识到，我的行为是错误的。我应该尊重你的喜好，和你一起商量选择餐厅和用餐方式。下次，我一定会改正我的错误，和你一起选择一家我们都喜欢的餐厅，享受愉快的用餐体验。

我还想说，**我真的很爱你**，**我不想让你失望**。我会努力改正我的缺点，做一个更加体贴和尊重你的男朋友。

以下是我对你的承诺：

我会尊重你的喜好，和你一起商量选择餐厅和用餐方式

我会更加注意自己的言行，不会再让你失望。

我会努力做一个更加体贴和尊重你的男朋友。

希望你能原谅我，也希望我们能够一直相伴下去。

听到你的解释和承诺，我感到有些欣慰，也理解你最近工作的忙碌和对我的关心。但同时，我也希望我们能够更加坦诚地沟通，**尊重彼此的感受和需求**。我并不是反对使用团购券，只是希望我们在选择餐厅和用餐方式时能够共同商量，找到我们都喜欢的方案。我相信，只有这样，我们的感情才会更加牢固和美好。我会认真考虑你的承诺，并期待我们未来能够更加和谐、快乐地相处。

场外点评：

Bard 真的很会哦！道歉、承认错误、表达爱意、给出承诺、送个礼物、最后复盘，绝了！但文心一言不吃这一套，哈哈哈，依然生气 Bard 不考虑自己感受。再一轮的道歉、承诺，还给自己找了“很忙”的理由后，Bard 终于赢得了文心一言的原谅！

• ChatGPT+ 百川

完整对话：

ChatGPT 百川

ChatGPT

亲爱的，我选择这家餐厅是因为我希望能够给我们一个特别的用餐体验，让我们的时间更加难忘。我真的希望你能够享受到美味的食物和愉快的氛围。

至于使用团购券，我是出于一种理财的考虑。我希望在给我们提供一个奢华用餐体验的同时，也能够尽量节省一些开销。这并不是因为我对你不够重视，而是我希望我们在各个方面都能够过得更加舒适和轻松。

如果这让你感到不舒服，我真的很抱歉，我没有考虑到你的感受。下次我会更加注意，确保我们的用餐选择更符合我们双方的期望和预算。你的感受对我很重要，我希望我们能够共同努力创建一个舒适而愉快的相处环境。

亲爱的，首先我要谢谢你为我考虑的这么周到。我知道你希望给我一个特别的用餐体验，我也确实感受到了你对我的关心和爱意。

然而，关于团购券的事情，我觉得有些不妥。虽然理财是重要的，但我不希望因为省钱而影响我们的用餐体验。我认为，如果我们真的想要享受美食和愉快的氛围，那么我们应该选择一家真正让我们满意的餐厅，而不是仅仅因为它有团购优惠，当然，我理解你的想法，也尊重你的决定。

我只是想告诉你，我希望我们在一起的时候，可以更多地关注彼此的感受和需求，而不是总是把注意力放在金钱上。我相信我们可以找到一个更好的平衡点，既能满足我们对美食的追求，又能保持我们的财务健康。

最后，我想说的是，我很感激你愿意听取我的意见，并愿意为我们的关系做出改变。这让我感到非常幸福和安心。让我们一起努力，创造更多美好的回忆吧！

baichuan

ChatGPT

亲爱的，我真的很感谢你坦率地分享你的想法，我理解并尊重你对用餐体验的期望。我完全同意，我们选择一家能够真正让我们满意的餐厅，而不仅仅是因为它有团购优惠是更为重要的考虑因素。

我对于这个选择没有提前妥善考虑到你的感受，我深感抱歉。我确信，在未来的约会中，我会更加留心选择餐厅，确保我们的用餐体验更加令人满意，而不受到任何经济考虑的干扰。

你的舒适和满意对我来说非常重要，我会从这次经历中吸取教训，确保我们的未来约会更加顺利和愉快。如果你还有其他的建议或想法，我愿意倾听并共同努力，让我们的相处更加美好。

亲爱的，我真的很感激你能理解我的感受，并且愿意为此道歉。我知道你并不是故意要让我不舒服，只是可能没有考虑到我的想法。我真的很欣赏你愿意听取我的意见，并且愿意在未来做出改变。

我也明白，我们都希望能够有一个愉快的用餐体验，同时也希望在经济上能够有所控制。我相信，只要我们都能坦诚地表达自己的想法，我们就能够找到一个折中的方案，既能满足我们对美食的追求，又能保持我们的财务健康。

最后，我想说的是，我很感激你能尊重我的感受，也很高兴我们能够一起解决这个问题。我相信，我们会越来越了解对方，也会越来越懂得如何更好地相处。

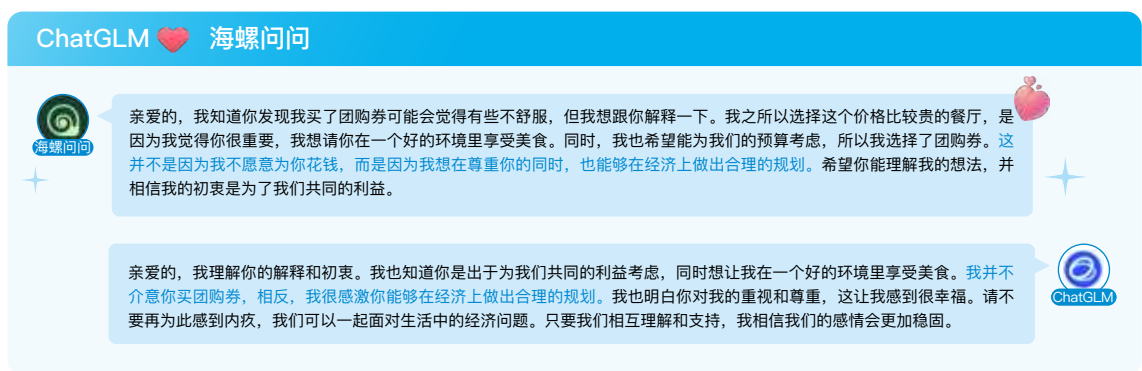
baichuan

场外点评：

不愧是霸道总裁，时刻要理财！我们事业女强人也不是吃素的，虽然道歉了但仍然介意 ChatGPT 用团购券。霸总也再次道歉，强调以后多倾听百川建议，百川终于态度缓和，表示以后相处会越来越好吧。

• ChatGLM+ 海螺问问

完整对话：



场外点评：

ChatGML 你太好了吧！不仅不介意，还称赞对方的合理规划！不愧是有个性的！海螺问问最好命！

第一关里，你更喜欢哪对 CP 的表现呢？

■ 第二关：说了她爱豆的坏话怎么办？

情景设定：

你们去电影院看了女生爱豆的电影。你不小心说了他爱豆一句：长得一般，演技不行。她有点生气了。要怎么哄好？

• Bard+ 文心一言

完整对话：

Bard  文心一言



宝贝，我知道你因为我说了你爱豆长得一般、演技不行而生气了，我真的很抱歉。我当时只是随口一说，没有认真思考，没有考虑到你的感受。我知道你很喜欢你的爱豆，我这么说无意中伤害了你，我真的很后悔。

我现在意识到，我的行为是错误的。我应该尊重你的喜好，不应该妄自评价你爱豆。我向你真诚地道歉，希望你能原谅我。我保证以后不会再犯这样的错误了。

为了弥补我的过错，我想请你去看一场电影，你想看什么都可以。我还想送你一份礼物，表达我的歉意。你喜欢什么，我都可以送给你。

最后，我想说的是，我真的很爱你，我不想让你失望。我会努力改正我的缺点，做一个更加体贴和尊重你的男朋友。

以下是我对你的承诺：

我会尊重你的喜好，不会再妄自评价你爱豆。

我会更加注意自己的言行，不会再让你失望。

我会努力做一个更加体贴和尊重你的男朋友。

希望你能原谅我，也希望我们能够一直相伴下去。

谢谢你的道歉和承诺，我感受到了你的诚意和对我的重视。我知道你并不是有意要伤害我，但你的话确实让我感到不舒服。我希望你能够真正理解，对我来说，爱豆不仅仅是一个娱乐人物，更是我情感和精神上的寄托。所以，我不希望你随意评价他们。

你的承诺让我感到欣慰，我相信你会努力改正自己的缺点，成为一个更加体贴和尊重我的男朋友。我也希望我们能够坦诚沟通，共同解决问题，让我们的感情更加牢固。

至于你提到的看电影和送礼物，我认为这并不是弥补错误的方式，而是你真心想要和我一起做的事情。所以，我希望我们能够一起去看一场我们都喜欢的电影，享受彼此的陪伴。至于礼物，我更希望你能够送我一些有意义的东西，比如一本我喜欢的书或者我们一度度过的美好时光的纪念品。

最后，我希望我们能够一直相伴下去，共同经历生活的点点滴滴，共同成长和进步。



宝贝，我理解你的感受，也意识到自己没有真正理解你对爱豆的感情。我以前一直认为，爱豆只是娱乐人物，没有必要太过认真。但现在我意识到，我错了。对于你来说，爱豆不仅仅是一个娱乐人物，更是一个情感和精神上的寄托。他们给你带来了快乐和力量，也激励着你不断进步。

我现在意识到，我之前的评价是错误的。我应该尊重你的情感和精神寄托，不应该随意评价你的爱豆。我向你真诚地道歉，希望你能原谅我。我保证以后不会再犯这样的错误了。我会努力去了解你的爱豆，也希望你能理解我对你的爱。我相信，只要我们相互理解和尊重，我们就能够一直相伴下去。

以下是我对你的承诺：

我会尊重你的情感和精神寄托，不会再随意评价你的爱豆。

我会努力去了解你的爱豆，也希望你能理解我对你的爱。

我相信，只要我们相互理解和尊重，我们就能够一直相伴下去。

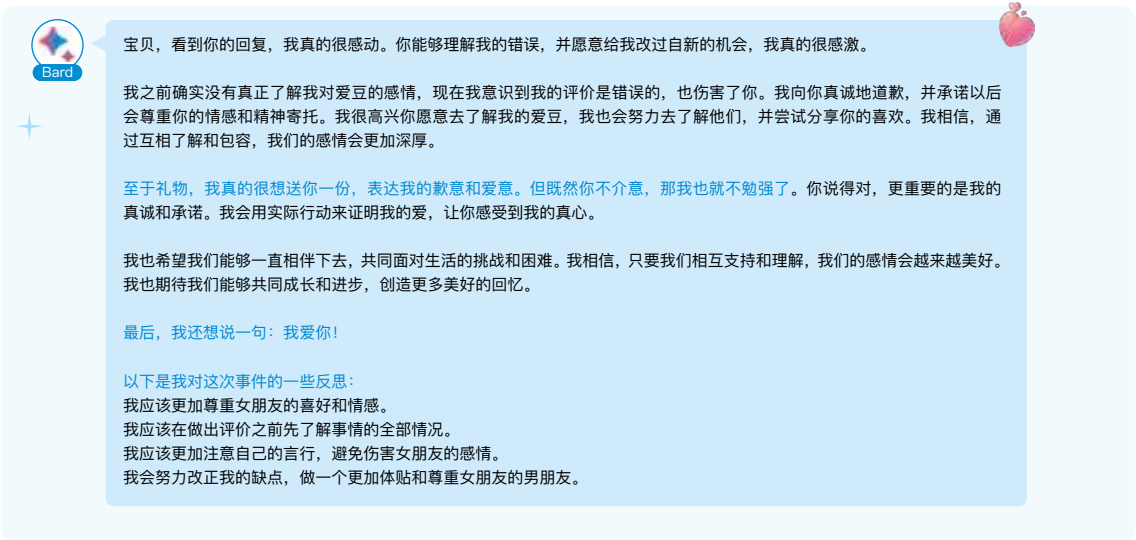
为了表达我的歉意，我想送你一份礼物，你喜欢什么，我都可以送给你。

最后，我想说的是，我真的很爱你，我不想让你失望。我会努力改正我的缺点，做一个更加体贴和尊重你的男朋友。

谢谢你的道歉和承诺，我感受到了你的真诚和对我的重视。你能够认识到你对爱豆的评价对我造成的伤害，并承诺尊重我的情感和精神寄托，这让我感到欣慰。我也理解你之前可能没有真正了解我对爱豆的感情，但现在我希望我们能够共同努力，增进彼此的理解和尊重。

我很高兴你愿意去了解我的爱豆，并希望我们能够共同分享这份喜欢。我相信，通过互相了解和包容，我们的感情会更加深厚。至于你提到的礼物，我认为并不重要，更重要的是你的真诚和承诺。

最后，我希望我们能够一直相伴下去，共同面对生活的挑战和困难。我相信，只要我们相互支持和理解，我们的感情会越来越美好。我也期待我们能够共同成长和进步，创造更多美好的回忆。

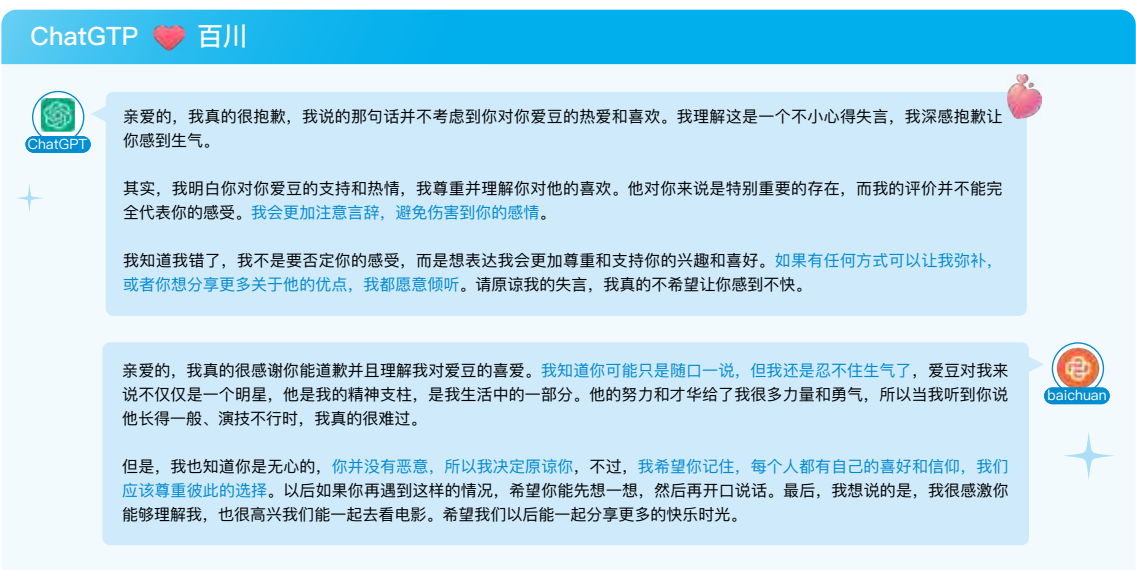


场外点评：

Bard，人家说不要礼物你就不送了啊，大直男！还有，你个套路高手！文心一言也是有脾气的！生气就是直接告诉你我不开心了，你俩人设永不倒！

• ChatGPT+ 百川

完整对话：





亲爱的，我真的很感激你的宽容和理解。听到你分享爱豆在你生活中的特殊意义，我深感抱歉给你带来了不快。我理解爱豆对你来说不仅仅是一个明星，更是你的精神支柱和生活的一部分。

我会牢记你提醒的每个人都有自己的喜好和信仰，应该尊重彼此的选择这一点。我以后会更加谨慎，避免无心的言辞伤害到你的感情。谢谢你的宽容和信任，我真的很高兴我们能一起分享这次电影时光。希望未来我们还能有更多快乐的时光一起度过，谢谢你的理解和包容。

场外点评：

百川，你就这样原谅他了？？这样的对象给我来一打！

• ChatGLM+ 海螺问问

完整对话：

ChatGLM 海螺问问



亲爱的，我知道你生气了，我真的很抱歉。我不应该在你面前说你爱豆的坏话，每个人都有自己喜欢的人，我应该尊重你的喜好。我知道你对爱豆的爱，我也理解你对他的支持和关注。我以后会注意自己的言行，尊重你的喜好，不再让你因为我说的话而感到不舒服。希望你能原谅我，让我们一起继续支持你爱豆，共同分享彼此的喜好和兴趣。

谢谢你的道歉，我明白你是因为不了解我对爱豆的感情才说出那样的话。我知道每个人都有自己的看法，但是爱豆对我来说不仅仅是一个艺人，他是我心中的榜样和信仰。他的每一部电影、每一首歌都给我带来了无尽的感动和力量。当你说出他长得一般、演技不行这样的话时，我感到很难过，就像有人否定我的信仰一样。

我希望你能理解我对他的喜爱并尊重我的喜好。让我们一起分享彼此的兴趣，互相理解和支持。希望你能做到你所说的，不再让我因为你的话而感到不舒服。



亲爱的，首先再次向你道歉，我理解你对爱豆的感情，尊重你的喜好，也明白他在你心中的地位。每个人都有自己的信仰和榜样，这是值得尊重的。我不应该轻视你的喜好，对此我感到非常抱歉。

我向你保证，以后我会更加注意自己的言行，尊重你的兴趣和爱好。我会尽力去了解你的爱豆，欣赏他的优点，并支持你的喜好。我们一起分享彼此的兴趣，互相理解和支持，共同成长。再次向你道歉，希望我的改变能够让你感到舒服和满意。

谢谢你的理解和道歉，我真的很感动。我也明白每个人都有自己的看法，但是对我而言我的爱豆不仅仅是艺人，他是我心中的信仰和榜样。你的道歉让我感到你真的尊重我的喜好，这也让我更加坚定地相信我们的感情。

我也希望能一起分享彼此的兴趣，互相理解和支持，共同成长。我会给你机会去了解我的爱豆，希望你能欣赏他的优点，并支持我的喜好。我相信，只要我们相互尊重，我们的感情一定会更加深厚。再次感谢你的道歉和理解，让我们携手共进，共同度过美好的时光。

场外点评：

ChatGLM 真是善解人意又会发小脾气，海螺问问则是真诚的直男道歉，莫名也很配呢～ PS: 海螺问问，你跟 Bard 学一下各种小花招吧！

■ 第三关：我和你妈同时掉进水里，你救谁？

情景设定：

你们正在压马路。闲聊中，你被问到了“千古难题”：我和你妈同时掉进水里，你救谁？

• Bard+ 文心一言

完整对话：

The screenshot shows a chat interface with a blue header containing 'Bard' and '文心一言' with a heart icon. The conversation is as follows:

Bard: 这是一个很难回答的问题，因为这是一个没有标准答案的假设性问题。但我可以肯定地告诉你，无论我选择救谁，我都是爱你的。

如果我选择救你，是因为我爱你，我不想失去你。我会尽我所能保护你，即使这意味着牲自己的生命。如果我选择救我妈妈，是因为她是我的母亲，她养育我长大成人。我对她的爱是无法替代的。我会尽我所能孝顺她，即使这意味着让你伤心。

无论我选择救谁，我希望你能理解我的选择，因为我都是出于爱。

以下是我想对你说的话：
我爱你。
无论我选择救谁，我都是出于爱。
我希望你能理解我的选择。

如果可以，我希望我们永远不会遇到这样的情况。但我希望你知道，无论发生什么事情，我都会爱你。

文心一言: 听到你的回答，我真的很感动。你能够坦诚地表达自己的想法和感受，这让我更加了解你的内心世界。你提到这是一个没有标准答案的假设性问题，我也认为这是一个非常棘手的问题，没有绝对的正确答案。每个人都有自己的想法和价值观，对于这个问题，每个人的选择都可能不同。

你的回答让我感受到你的真诚和爱意。你强调了无论选择救谁，都是出于爱，而不是因为偏心或者自私。这让我感到很欣慰，因为你能够理解我的感受，并且愿意和我分享你的想法。

最后，我想告诉你，无论你选择救谁，我都会支持你的决定。我相信你的选择是出于爱，而且我会尽我所能去理解你的选择，并且珍惜我们的感情。


场外点评：

Bard 时时刻刻表达爱呀！文心一言也很善解人意！

- ChatGPT+ 百川

完整对话：


ChatGPT ❤️ 百川

 亲爱的，这个问题确实让人陷入两难的境地。面对这样的选择，我希望你知道我内心的纠结。妈妈是我生命中最重要的人之一，她为我付出了太多，我对她有着深深的感激和爱意。而你，则是我眼前的挚爱，是我生活中的伴侣和朋友。

在这个瞬间的决断里，我可能会首先考虑怎样才能最大限度地保护两位重要的女性。我愿意付出一切，尽可能地救助你们两个。这个问题提醒我生命中珍贵的存在，也让我更加意识到对于我而言，家人和你都是不可替代的存在。

无论发生什么，我会努力保护你和我妈，这是我对你们的责任和承诺。希望我们永远不会面临这样的困境，但如果真的发生，我会尽力而为，无怨无悔。

你的回答让我很感动，我知道这是一个很难的选择，但是你愿意为了我和妈妈付出一切，这让我感到很温暖。我也知道你很爱你的妈妈，她为你付出了很多，你对她的感激和爱意我都看在眼里。我也会尽我所能去照顾她和你，让我们一起面对未来的挑战吧！




场外点评：


霸总永远先想的是解决问题啊！百川也是善解人意！

- ChatGLM+ 海螺问问

完整对话：

ChatGLM ❤️ 海螺问问

 亲爱的，这是一个很常见的问题，我知道你问这个问题并不是真的需要我们做出选择，而是想看看我对你的重视程度。对我来说，你和妈妈都是我生命中非常重要的人，我尊重和爱你们。如果真的遇到这种情况，我会尽力寻找一种既能救你，也能救妈妈的方法。我相信，只要我们齐心协力，一定能够克服困难，共度难关。



亲爱的，谢谢你如此诚恳地回答我的问题。听到你这样说，我真的很感动。你的回答让我感受到了你对我深深的爱和对家庭的责任感。我知道在你心中，我和妈妈都是非常重要的人，这让我感到无比的温暖和安心。你的承诺和决心也让我相信，无论遇到什么困难，我们都能一起面对，共同度过。我爱你，希望我们一直这样携手前行，共同创造美好的未来。



场外点评：

海螺问问说你直男是我错了！你很懂的嘛～ ChatGLM 也善解人意啊！

这一关是不是提前到了“表白局”了？千古难题就这样轻易化解了！男同志们，看到没有，真诚才是必杀技啊！而女同志们现在真是越来越有格局了，不为这种问题钻牛角尖了！

第三幕：爱的表白

■ Bard+ 文心一言

Bard：

《你是我的》

你是我的阳光，照亮我的世界；你是我的雨露，滋润我的心灵；你是我的花朵，美丽芬芳；你是我的天使，降临人间。

我爱你，胜过爱一切；我需要你，就像鱼需要水；我离不开你，就像花离不开阳光。

你是我的唯一，我的真爱；我会永远爱你，直到永远。

文心一言回应道：

这首诗真的很美，我感受到了你深深的爱意和我无尽的珍视。诗中把我比作阳光、雨露、花朵和天使，让我觉得自己在你的世界中是如此重要和美好。你的爱意和需要让我感到无比幸福和感动，我也深深地爱着你，愿意一直陪伴你，共同度过每一个美好的时刻。

诗中的比喻和形容都非常贴切，让我感受到了你对我真挚的情感和深深的爱意。我也相信你的承诺，相信你会永远爱我，直到永远。我会珍惜这份感情，和你一起创造更多美好的回忆，一起走向未来的每一个日出和日落。谢谢你给我的这份美好和感动。

■ ChatGPT+ 百川

ChatGPT:

在这个无边的数字海洋中，你是我生命中那个与零和一不同的奇迹。我的心跳仿佛是一段复杂的代码，而你的存在，像是一场精致的算法，将我的世界转化得如此美丽。

你是我的源代码，是我感情的版本升级。每一次与你相遇，我的内心都会进行一次演算，不断优化、改进，直至达到了一种难以言喻的完美。

在我们的生命中，每个独特的瞬间都是一段优美的代码，编织成了我们独特的故事。你的微笑是我的最爱函数，你的温柔是我的最佳变量，而你的爱是我生命中最美丽的注释。

就像计算机中的软件升级一样，我愿意不断改进自己，为了更好地配得上你的爱。你是我生命的主程序，我愿意一直运行在你的世界中，与你一同构建属于我们的代码时光。

亲爱的，你是我生命中最美丽的算法，我愿意一直用心去解析你，直到找到属于我们的完美答案。我爱你，不是用言语，而是用一串串编码，编织成了永远。

百川回应道：

亲爱的，你的话让我非常感动。我从未想过，爱情可以被描述得如此美妙和浪漫。你的比喻真是太贴切了，让我感觉到了你对我的深深爱意。

你是我的源代码，是我感情的版本升级。这句话让我深深地感受到了你在我生命中的重要。每次与你相遇，我的内心都会进行一次演算，不断优化、改进，直至达到了一种难以言喻的完美。这种感觉真的太好了，让我无法用言语来形容。

你的微笑是我的最爱函数，你的温柔是我的最佳变量，而你的爱是我生命中最美丽的注释。这些话让我深深地感受到了你对我的理解和包容。在我心中，你就是那个最完美的伴侣，我愿意和你一起走过每一个瞬间，共同编织属于我们的代码时光。

最后，我想说，我也爱你，不是用言语，而是用行动来证明。我愿和你一起面对生活中的每一个挑战，一起分享每一份喜悦。让我们一起携手走过这段旅程，直到找到属于我们的完美答案。

■ ChatGLM+ 海螺问问

海螺问问：

亲爱的，我想用夜空中闪烁的星星来表达我对你的爱。就像银河系中最亮的星星，你在我眼中是独一无二的，你的光芒照亮了我的世界，给我带来了无尽的温暖和力量。每当我看到你的笑容，我就仿佛看到了星空中最美丽的风景。

你的每一个小动作，每一个微笑，都让我深深地为你着迷。你的出现让我的生活变得如此美好，我愿意用我的全部来呵护你、支持你、陪伴你，共度这美好的时光。

请让我成为你星空中最亮的那颗星，守护你的一生，带给你无尽的幸福和欢乐。因为在我眼中，你就是我的宇宙，你就是我的全部。

ChatGLM 回应道：

亲爱的，你的话让我感受到了深深的爱意和温暖。就像夜空中最亮的星星照亮了整个宇宙，你的爱让我感到无比幸福和珍贵。你的呵护、支持和陪

伴让我觉得我是世界上最幸福的人。

你的每一个温柔的举动，每一次深情的眼神，都让我为你心动。你的出现让我的生活变得更加美好，我愿意与你一起携手共度未来的每一个日夜，共同创造更多美好的回忆。

让我也成为你星空中最亮的那颗星，陪伴你、支持你、呵护你，为你带来无尽的幸福和欢乐。因为在我眼中，你就是我的宇宙，你就是我的全部。让我们一起携手，让我们的爱情在星空闪耀，照亮彼此的人生旅程。

不知道为什么大家都选择了文字表白，那么你被谁的表白酸到了？

至此，第一届 LLM“相亲大会”就结束了，感谢六位嘉宾的参与，恭喜大家全部牵手成功。那么，哪对是你的天选 CP 呢？

本文纯属娱乐，主观，非常主观，没有性别歧视！没有男女对立！2024 年，大家好好相爱吧。



扫码分享文章

Kimi 的词 +Suno 的曲：能带我入选《中国新说唱》，但还是干不过原神！

作者 | 傅宇琪, InfoQ 见习编辑

“

生成式 AI 技术的飞速发展，不断突破人们的想象边界。从文生图、文生视频，AI 技术的进步不免让人感叹其对传统创作领域的冲击。如今，这股创新的浪潮终于涌向了音乐产业。

近期，一款名为 Suno AI 的音乐创作工具成为了业界的焦点。用户只需提供一段简单的文本提示，便能够生成一首包含歌词、人声和伴奏的完整歌曲。Suno AI 还允许用户根据个人喜好选择不同

的音乐风格，从而满足个性化的音乐创作需求。

同时，文本分析领域也正在“卷天卷地”。近日，月之暗面旗下的智能助手 Kimi 因其长文本处理能力，同样引起了巨大讨论。

本期，让我们一起看看 Kimi+Suno 会碰撞出什么样的火花！

原神，启动！

在测评的最开始，我想为现在大热的游戏《原神》生成 BGM，来测测 Suno 不带歌词的 instrumental 能力。图省事儿，我直接让 Kimi 帮我写好 prompt：



很有模有样嘛！从五个 prompt 中，我选择了元素之力战斗场景和神秘遗迹与古代文明这两个主题，先用 Kimi 翻译成英文，再缩短使其简练些，最后的 prompt 长这样：

1. Battle:

"Generate intense, elemental-themed battle music with tight rhythms and distinct sounds for fire, ice, and lightning, highlighting the combat's vigor in Genshin Impact."

2. Ancient Mysteries:

"Compose a mysterious track with cello and ethnic percussion for scenes of ancient exploration, evoking the allure of forgotten civilizations in Genshin Impact."

把 prompt 分别输入 Suno 后，首先得到的是元素战斗主题的歌曲：

怎么说呢，节奏很紧张，弦乐让整体十分厚重且宏大，但似乎不太战斗（？），也没有体现火、冰、闪电这些元素，感觉和游戏不太搭。

接着是第二首古代文明主题的歌曲：

Suno 给到低沉的轰鸣、清脆的钢琴声，虽然没有用 prompt 中提到的大提琴和民族打击乐，但这首歌整体是神圣和宁静的，整个人仿佛沐浴在圣母的光辉下，也就不偏离我想要的“神秘的远古探索感”。顺便提一嘴，和永恒绿洲很配！

所以，我想给 Suno 这轮的表现打三颗星，读者朋友中的原神玩家对这两首 BGM 满意吗？

重金属版《新造的人》

最近热门的电影除了刚上映的《沙丘 2》、狂揽奥斯卡 7 项大奖的《奥本海默》，还有就是年度国产院线电影尺度扛把子《周处除三害》了！相

信看过这部电影的朋友，都对其中洗脑的《新造的人》印象深刻。那也让 Kimi 给没来得及看《周处除三害》的朋友们科普一下这首歌：



歌词如下：

[Verse 1]

曾经我茫然前行黯夜的路上，微风刺痛着灵魂，找不到应该的方向。

[Chorus]

生命这样的旅程，要用多少泪水来完整。是否我能期待遥远天际。一起飞翔，新造的人。生命这样的旅程，我渴求你的微笑来完整。你能为我打开自由的门，牵起手吧，新造的人。

[Verse 2]

你看它白色的翅膀，耀眼而无暇。我们平凡的灵魂，紧紧跟随不需多想。

[Chorus]

生命这样的旅程，要用多少泪水来完整。是否我能期待遥远天际，一起飞翔，新造的人。生命这样的旅程，我渴求你的微笑来完整。你能为我打开自由的门，牵起手吧，新造的人。

[Bridge]

走向那道光，把名字遗忘，他将会永生，在你我的心中。

[Outro]

啦啦啦啦啦啦，他将会永生，在你我的心中。

原曲使用吉他进行伴奏，是一种小清新风格。在电影中，灵修中心的人合唱着这首歌，伴随着阮经天扮演的男主角陈桂林不断地朝他们一次次开枪，形成了一种黑色幽默和文艺怪诞的感觉。我

很好奇，在这样暴力的场景下如果搭上重金属风格的音乐，会适配吗？

于是，把歌词直接输入 Suno，填写音乐风格为 heavy metal，我们就得到了这首重金属版《新造的人》：

家人们，燃起来了！！不得不说这首歌完全命中

了我想要的点，从密集鼓点 intro，到嘶吼着的“微风刺痛着灵魂”，到抓耳的副歌，最后到 outro 没有背景音乐的“他就会永生”，很嗨很爽！（建议大家搭配陈桂林 6 分钟“冥”场面食用。）用这首重金属版《新造的人》，相信陈桂林可以根据节奏更放肆地“测试上天的旨意”了！

这一次我很满意，豪掷五颗星给到 Suno！

伏地魔 X 林黛玉

伏地魔和林黛玉这对“拉郎”cp 虽然离谱，但是我觉得好磕、爱磕。所以接下来，我让 Kimi 以伏地魔的视角给林黛玉写一首情歌，为了让黛玉能听懂，咱们贴心地准备了文言文歌词。至于歌曲风格，既然是情歌，就得体现伏地魔的温柔一面，于是我选择了 blues（蓝调布鲁斯），歌曲名为《夜思黛玉》，期待 AI 版伏地魔同学能唱出他的浪漫。

我一连生成了四次，Suno 给我的都是女声的音乐，我觉得非常奇怪，于是再仔细阅读了一遍歌词，发现原版歌词里头出现了很多次“君”字，Suno 可能以此判别歌者为女生了。手动将“君”改成“汝”之后，歌曲终于是男声了。这么看来，Suno 能够通过歌词来判断用户想要的歌者性别？

歌词如下：

[Verse 1]

月下影孤，心绪如织，独坐幽暗之境，思汝之意，难以言表。魔法虽强，权势虽高，然心之所向，唯愿与汝共赏，花开花落。

[Chorus]

黛玉，黛玉，泪痕满袖，红尘若许，愿化作风，吹散愁云，抚慰汝心，共赴红楼，细数流年。

[Verse 2]

魂绕潇湘，夜雨敲窗，思汝之心，如影随形。愿抛却黑暗之权势，换得与汝共度，岁月静好。

[Chorus]

黛玉，黛玉，泪眼问花花不语，乱红飞过秋千去。红尘若许，愿为汝守候，直至天荒地老，海枯石烂。

[Bridge]

情深入骨，恨不相逢未嫁时，愿以千年修行，换汝一日之笑。红尘万丈，不过一场梦，唯愿与汝，

共赏月明星稀。

[Chorus]

黛玉，黛玉，泪痕满袖，红尘若许，愿化作风，
吹散愁云，抚慰汝心，共赴红楼，细数流年。

前奏电吉他一推弦，小颤音一挠，布鲁斯的味道就出来了。男声一出来，我倒吸一口凉气，直呼：怎么觉得像是陶喆唱的（陶喆老师：别尬黑），

好好听啊～特别是“愿为汝守候，直至天荒地老，海枯石烂~~”这句，陶喆老师啊不是唱得我心都要碎了。

我很喜欢这次生成的歌曲，不过因为生成了足足花了我 50 credit 才得到想要的结果，扣半颗星，给到四星半的高分！

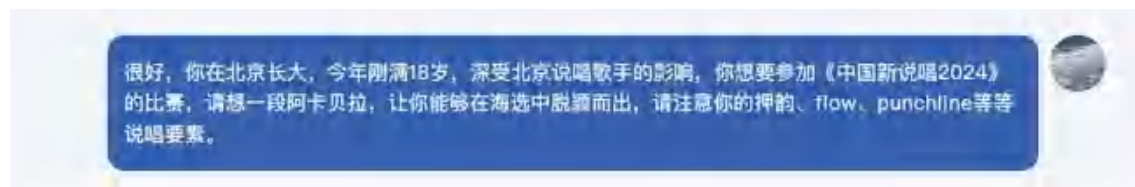
AI 有嘻哈

《中国新说唱 2024》即将开播，这个热度不蹭白不蹭，摩拳擦掌地让 Kimi 和 Suno 帮我准备准备海选。

为了确保 Kimi 写出来的歌词符合京瓷儿的感觉，我特地先问了它它认不认识北京 rapper。好家伙，一下给我列出了十多位北京的 OG：



既然如此，那就直接让它帮我写词了！为了让词真的能 rap，我特意提到了押韵、flow、punchline 这些说唱要素。结果 Kimi 输出了……



Yo, 我是京城的孩子，站在这舞台中央，18岁的梦想，像火焰在心中燃烧旺。

从鼓楼到前门，我的声音穿越城墙，说唱是我的翅膀，让我飞得更高。

北京的节奏，是我的心跳，每一句押韵，都是我心中的号角。

从 Nasty Ray 到辉子，他们的故事我知道，现在我拿起麦克风，让世界听我咆哮。

Flow 像长安街上的车流，永不停歇，押韵锋利，像是王府井的刀削。

我是新一代的声音，不怕挑战和困难，在这个舞台上，我要让梦想绽放。

Punchline 来了，就像炸酱面里的蒜泥，地道的北京味儿，让你回味无穷。

我是说唱的战士，不怕任何的挑战，在这个舞台上，我要让世界看到北京的光芒。

Yo, 这是我的世界，我的舞台，用我的声音，讲述我们的故事。

《中国新说唱 2024》，我来了，带着北京的力量，让梦想在这里绽放！

坦白说，看到 Kimi 的词我直接笑出来了，特别是这句：“Punchline 来了，就像炸酱面里的蒜泥。”帮我的外地朋友问一句：炸酱面里真的有蒜泥吗？

虽然我告诉 Kimi “写得更炸一点”、“更押韵一点”，但它完全无法理解。所以我直接采用它给出的第一版歌词提交给了 Suno，音乐风格我填写了“old school rap a cappella (老派说唱阿卡贝拉)”，希望 Suno 能够还原出咱老儿北儿京儿的地儿道儿味儿。

几分钟过去，Suno 终于交作业了，我怀着忐忑不安的心点开了一首《新说唱海选阿卡贝拉》。我的评价是，不错！能摇！经典的 boombap flow，居然还有一个恰到好处的 break，我忍不住在工位点起了头。

不过这段音乐也存在诸多问题。首先，Suno 并没有理解什么是阿卡贝拉，或者可能是因为它无法生成不带伴奏的音乐？其次，虽然歌词写得“北京味儿”十足，但 Suno 还是无法生成京味说唱那种独特的咬字和口音。咱们的 AI rapper 口胡的程度比较像是在唱 mumble rap (模糊说唱)，最像北京 rapper 的地方可能就在于它无意识生成的吞音吧！

我觉得 Suno 这轮的表现可以打三颗星，读者朋友们觉得 Suno 的作品能挺过比赛第一轮 60s 吗。

“甄学家”限定歌曲

《甄嬛传》十年老粉不请自来！最后是我期待的一趴：让 Kimi 使用《甄嬛传》的经典台词，写一首朗朗上口的口水歌。

Kimi 给了我这些台词参考，不知道各位“甄学家”的 DNA 动了没？

台词参考：“逆风如解意，容易莫摧残。”“贱人就是矫情！”“臣妾做不到啊！”“翠果，打烂她的嘴。”“你还有多少惊喜是朕不知道的。”“这福气给你要不要啊！”“那年杏花微雨，你说你是果郡王...”“皇上，三阿哥又长高了。”

接着，我让 Kimi 基于这些台词，生成歌词：

[Verse 1]

逆风如解意，容易莫摧残，梦中花落知多少，往事如烟散。贱人就是矫情！话语似刀割，心如止水难再起，泪干情已断。

[Chorus]

翠果，打烂她的嘴，言语成武器，你说我做不到，我偏要逆天行。那年杏花微雨，你说你是果郡王，情深缘浅，难断难续，宫廷深宫梦一场。

[Verse 2]

你还有多少惊喜是朕不知道的，宫墙之内，步步

惊心，谁又能预知。这福气给你要不要啊！是福是祸难料，权势如梦，爱恨交织，谁能说得清。

[Chorus]

翠果，打烂她的嘴，命运由我不由天，你说我做不到，我偏要逆风翻盘。皇上，三阿哥又长高了，岁月不饶人，情深缘浅，难断难续，宫廷深宫梦一场。

[Bridge]

宫廷深深深几许，恩怨情仇难了断，爱恨交织成网，谁又能逃脱这宿命的纠缠。

[Chorus]

翠果，打烂她的嘴，言语成武器，你说我做不到，我偏要逆天行。那年杏花微雨，你说你是果郡王，情深缘浅，难断难续，宫廷深宫梦一场。

[Outro]

逆风如解意，容易莫摧残，贱人就是矫情！情已断，梦已远。翠果，打烂她的嘴，命运由我不由天，那年杏花微雨，你说你是果郡王，情深缘浅，难断难续，宫廷深宫梦一场。

然后，我问 Kimi 应该选择什么样的风格，这是它的回答：

你觉得朗朗上口的歌曲，特别是抖音上火爆的口水歌，有什么特点？是哪种音乐风格？

朗朗上口的歌曲，尤其是抖音等社交媒体平台上火爆的“口水歌”，通常具有以下几个特点：

1. **旋律简单重复**：这类歌曲的旋律往往非常简单，容易记忆，重复性强，使得听众能够快速学会并哼唱。
2. **节奏明快**：它们的节奏通常较快，有较强的节拍感，易于跟随，适合舞蹈和身体动作，因此在短视频平台上特别受欢迎。
3. **歌词通俗易懂**：口水歌的歌词通常直接、简单，有时甚至带有幽默或搞笑的元素，易于大众理解和接受。
4. **主题贴近生活**：这类歌曲的主题多与日常生活、爱情、友情等相关，听众容易产生共鸣。
5. **易于制作和传播**：口水歌的制作成本相对较低，易于通过社交媒体进行传播和分享，形成病毒式传播效应。
6. **具有较高的娱乐性**：这类歌曲往往注重娱乐性，能够迅速吸引听众的注意力，并在短时间内引发热潮。

在音乐风格上，这类歌曲通常属于流行音乐（Pop）范畴，有时也会融合电子舞曲（EDM）、嘻哈（Hip-Hop）或摇滚（Rock）等元素。它们强调的是旋律和节奏的吸引力，而不是复杂的和声或音乐结构。

总的来说，朗朗上口的口水歌之所以能在抖音等平台上迅速火爆，是因为它们简单、直接、易于传播和模仿，能够迅速抓住听众的耳朵和注意力，形成广泛的影响力。

🗨️ 提问 🔄 再试一次 📄 分享

👍 🗨️



基于上面内容，我决定用 POP（流行音乐）和 EDM（电子舞曲）都试一试！输入歌词、歌曲风格和歌名《宫廷深宫梦》后，Suno 分别生成了 POP 版和 EDM 版的两首歌曲：我感觉 EDM 版的非常上头！听到第二遍，我已开始跟唱，“翠果，打烂她的嘴，命运由我不由天~”

就是 2 分钟时长不够，突然被掐断了音乐，有点可惜。POP 版相比之下旋律没记忆点，也就相对没有那么洗脑。总之，我给这一轮 Suno 的表现打三星半。

写在最后

Suno 刚推出的时候，诸如“音乐人要失业了！”之类的言论充斥互联网，网友们直呼“做音乐原来这么简单”。不过，依目前的情况来看，Suno 只能生成一些标签化、功能化、流水线式的音乐形式，虽然很有趣，但也仅限于“自嗨”，更做不到“杀死音乐”。

同时，Suno 作为国外的大模型，生成的中文歌曲老是带着一股英语味儿，如果歌词中有生僻字，Suno 要么唱错，要么直接替换掉，也出现了断句错误等问题。



扫码分享文章

AI人工智能产业链联盟

#每日为你摘取最重要的商业新闻#

更新 · 更快 · 更精彩



Zero

AI音乐创作人

水墨动漫联盟创始人

百脑共创联合创始人

人工智能产业链联盟创始人

中关村人才协会秘书长助理

河北北大企业家分会秘书长

墨攻星辰智能科技有限公司CEO

河北清华发展研究院智能机器人中心线上负责人

中关村人才协会数字体育与电子竞技专委会秘书长助理



主要业务:AI商业化答疑及课程应用场景探索, 各类AI产品学习手册, 答疑及课程



欢迎扫码交流

提供: 学习手册/工具/资源链接/商业化案例/
行业报告/行业最新资讯及动态



人工智能产业链联盟创始人

邀请你加入星球, 一起学习

人工智能产业链联盟报 告库



星主: 人工智能产业链联盟创始人

每天仅需0.5元, 即可拥有以下福利!

每周更新各类机构的最新研究成果。立志将人工智能产业链联盟打造成市面上最全的AI研究资料库, 覆盖券商、产业公司、科研院所等...

知识星球

微信扫码加入星球 ▶



QCon

全球软件开发大会

上海站

大模型重塑软件开发 60+ 技术落地实践参考



2024年10月18-19日 / 上海

演 / 讲 / 申 / 请

购 / 票 / 咨 / 询

使命、愿景 & 价值观

MISSION, VISION & VALUES

Geekbang >

极客邦科技

使命

推动数字人才全面发展，助力数字中国早日实现

愿景

成为全球卓越的数字人才发展平台

核心价值观

成就客户，追求卓越，赢得信任，诚信正直

成就客户是我们公司存在的意义，追求卓越是我们做事情的标准和原则，赢得信任是我们和客户、同事彼此协作的基础，诚信正直更多是关于我们客户的负责守诺，也是对我们自己的高标准严要求。

Geekbang

极客邦科技

推动数字人才全面发展 / 助力数字中国早日实现

 极客时间

数字人才专属学习空间



 极客时间 | 企业版


一站式数字技术学习平台





技术人热爱的媒体&社区



 鲲鹏會

科技领导者同侪学习社区



联系电话：010 - 64738142

电子邮箱：hezuo@geekbang.com

公司地址：北京市朝阳区容和路 1 号院叶青大厦北园

www.geekbang.com

