90 后 AI 天才的大模型首战

https://mp.weixin.qq.com/s/-uy01U2g0tym2rOtrTn9Pg

站在核爆中心圈，是一种什么样的体验？

在这次 ChatGPT 引发的 AI 大爆炸中，做了十年堪称冷门的 NLP（自然语言处理）的杨植麟，就处在这样一个位置。这位保送清华、程序设计课程满分的“少年天才”，在卡耐基梅隆大学读博士时，就已经作为第一作者发表的关于 Transformer-XL 与 XLNet 的两篇论文，成为本次 AI 大模型技术能够突破的重要一环。

“先是非常激动，好像被苹果砸中一样，”杨植麟对 36 氪说，随即又陷入沮丧，再想到可干的事情还很多，又“兴奋起来”。

这也是他新创办的第二家 AI 公司“月之暗面（Moonshot）”的由来。Moonshot 这个名字，则来自英国著名摇滚乐队 Pink Floyd 的专辑《Dark Side of the Moon》。

杨植麟认为，做大模型如同登月工程一样，“月之暗面”意味着神秘，令人好奇和向往，同时又极具挑战难度。

事实上，月之暗面的核心团队曾参与到 Google Gemini、Google Bard、盘古 NLP、悟道等多个大模型的研发中——这是一支在“登月”道路上已探索多年的队伍。而 AI 大模型，目前还在一个以技术能力定成败的阶段。

在这半年的国内大模型市场中，Moonshot 显得尤为沉默，但并不妨碍投资人的蜂拥而至。36 氪最新获得的消息是，月之暗面已经完成一轮超过 2 亿美元的融资，目前身处中国大模型创业公司融资额第一梯队。

成立半年多后，10 月 9 日，Moonshot 终于推出了首款大模型产品：智能助手 Kimi Chat。这是 Moonshot 在大模型领域做 To C 超级应用的第一次尝试。

Kimi Chat 支持输入 20 万汉字，是目前全球大模型产品中所能支持的最长上下文输入长度。

这也代表着，Moonshot 在长文本技术的探索突破到了一个新高度——对比当前市面上几家主流模型，Kimi Chat 的上下文长度是 Claude 100k 的 2.5 倍（实测约 8 万字），GPT-4-32k 的 8 倍（实测约 2.5 万字）。

如今市面上的大模型产品繁多，拓展了上下文长度的 Kimi Chat，在使用上有什么不同？

最明显的是，你可以一次性给模型输入大量的信息，由模型理解进行问答和信息处理，有效减少幻觉问题。

比如，公众号的长文也可以交给 Kimi Chat ，让它帮你总结分析：

发现了新的算法论文时，Kimi 能够直接帮你根据论文复现代码：

快要考试了，直接把一整本教材交给 Kimi，就可以让它陪你准备考试：

甚至，也可以只用一个链接就让它来扮演你喜爱的游戏角色，和你对话：

目前，Moonshot AI 的智能助手产品 Kimi Chat 已开放了内测。访问 Moonshot.cn（或于文末扫描二维码），即可加入内测计划。

长文本：大模型落地的另一瓶颈

值得关注的一点是，不同于其他大模型公司拼参数、展示各种各样的行业案例，在 Moonshot 的发布会上，“长文本”成了绝对的主角。

“无论是文字、语音还是视频，对海量数据的无损压缩可以实现高程度的智能。而要有效提升大模型的性能，不仅要扩大模型参数，更要提升上下文长度，两者同样重要。”杨植麟表示。

大模型之所以能在智能水平有质的飞跃，是因为通过扩大参数规模，突破到了千亿级别，才能够让智能“涌现”（Emergence，指模型自主产生出复杂行为或特性）。

但如今，大模型落地更重要的瓶颈不是模型大小，而是在于上下文不够，文本长度不足会带来对模型能力的严重束缚。

一个典型问题是，如果遇到多轮对话或者需要复杂步骤的场景，往往会出现模型记不住的情况——讲了具体设定，但下一回合就忘记。比如，Character AI 的用户就经常吐槽模型记不住关键信息：

这与计算机运行的原理类似：计算机依靠 CPU 进行计算；内存则存放了临时计算的数据，决定其运行速度。“如果说参数量决定了大模型支持多复杂的‘计算’，而能够接收多少文本输入（即长文本技术）则决定了大模型有多大的‘内存’，两者共同决定模型的应用效果。”他解释道。

这也是 Moonshot 在保持模型拥有千亿级参数的同时，首先将上下文长度先“拉满”的原因。

要想做到拓宽上下文长度（Context），在模型训练和推理侧都存在算力+显存的双重挑战。

比如，计算量会随着上下文长度的增加呈平方级增长——比如上下文增加 32 倍时，计算量实际会增长 1000 倍；而在推理方面，即使是将单机显存配置拉到目前的最高水平（如配备 8 张 80GB 显存的 GPU 芯片），最多只能在千亿级模型上处理约 5 万汉字的长度。

但在 Kimi Chat 上，Moonshot 团队通过创新的网络结构、改进算法策略等等，对模型训练的各个环节进行了上百项的优化，从而在千亿级参数下可以实现对超长文本的全文理解。

简单而言，Moonshot AI 并不通过当前滑动窗口、降采样、小模型等对效果损害较大的“技术捷径”来实现长文本，而是研发基于大模型的长程注意力，以实现真正可用的超长文本技术。

让模型“记性”更好，会让大模型未来的应用场景拓宽不少。比如，律师、分析师等职业，就能让大模型分析长篇报告；像狼人杀这样需要基于大量信息来推理的游戏，大模型也能够胜任。

而在本次产品发布前，36 氪曾与杨植麟进行过一次深谈。作为站在这次技术核爆中心圈的人，杨植麟谈起 AI 大模型，有种笃定感。他会不时用轻松的语气，抛出一些让人一愣的断言。

比如，“Next token prediction（预测下一个字段）是唯一的问题。”“只要一条道走到黑，就能实现通用泛化的智能（AGI）。”

比如，“五年之内，大模型将持续保持较强的技术壁垒，不会 commoditize（变成平价的、没有壁垒的商品）。”

从 LLM（大语言模型）到 LLLM（长文本大语言模型），Kimi Chat 只是 Moonshot 的第一步。不过，如今的 Moonshot 已经寄托着杨植麟一些很“黑镜”的预想：在未来，如果机器能够掌握一个人一生的信息，人们就会拥有自己的 AI 分身，这个 AI 分身共享了你的所有记忆，无异于另一个你。

以下为 36 氪与杨植麟的对话实录，经 36 氪编辑整理：

时隔七年，两次 AI 创业

36 氪：先来聊聊这次产品发布吧。很多大厂、创业公司都会选择先发一个具体的大模型，开源或者闭源的都有。大模型已经火了半年后，Moonshot 如今选择先发一个 To C 的智能助手产品。为什么？

杨植麟：因为我始终坚信以终为始，只有当大模型被多数人使用时，才会涌现出最多的智能。Moonshot 会秉承以应用为导向的模型开发，我们并不想只是发布一个模型，以迅速获得科技圈可能的短期技术关注。

比如，“长上下文”技术的价值，可能很难第一时间让用户感知到。但通过 Kimi 智能助手，就可以直接触达用户。我们希望让技术成为用户日常生活中一旦接触就不可或缺的助手，以真实的反馈做来迭代模型，尽早地创造实际价值。

36 氪：ChatGPT 出来之后，这半年你的心情是怎么样的？

杨植麟：这一年来，我是百感交集。如果是什么可控核聚变的突破，那其实跟我也没什么关系，但这个事情（大语言模型）是我做了十年的事情，我觉得就好像是被苹果砸中一样。

ChatGPT 刚发的时候，我非常激动，我好奇这个世界到底能做什么样的 AI，我能多大程度去复制、甚至做得比人脑更好。

同时，我也陷入到非常沮丧的状态——因为这个事情也不是你做出来的对吧？我会开始想在这个浪潮里我还能贡献什么，又开始兴奋起来：现在是非常好的 timing，不管发生什么，一定要做。

36 氪：ChatGPT 算是直接促使你创立新公司“月之暗面”？

杨植麟：对。从一开始的激动到沮丧，再决定创业之后，我逐渐恢复理性思考，思考想要什么样的团队来做，现在是技术演进过程里的什么阶段，我们要做什么？

然后再开始焦虑——铺天盖地地，所有人都说要做大模型，那大模型到底能不能做？是不是做不了？

最后又会回到理性。我会去更长期地看这些个事情，短期内的大模型进展，东边发一个模型，西边发一个，其实都是噪音。

GPT-4 的水平在这儿（高一截），其他模型都是在下面，其实大家现在说“我比你高”“你比我高”，没什么意义。我这半年都在思考底层逻辑，最后发现这件事还是很适合我们来做。

36 氪：适合在什么地方？

杨植麟：每一次技术突破里会有三层的机会。

第一层机会，是被第一个找到第一性原则的人抓住，那就是 OpenAI。这需要很强大的 vision，非常高瞻远瞩，是靠经验所支撑的。

第二层机会就是在技术创新期，能解决一些技术方向性的问题——比如 long context（长上下文）怎么做？能把技术做好的团队可以抓住。

第三层是纯应用的机会，就是技术已经全部清楚了，不再需要考虑技术层面的事情，只做应用。

我们可以抓住的是第二层机会，在这个层面我们拥有很好的积累和优势。

36 氪：月之暗面想做的大模型，是怎么样的？

杨植麟：我们希望先把模型能力做到世界领先水平，同时也会聚焦 C 端的超级应用，通过产品连接技术与用户，从而共同创造通用智能，Kimi Chat 只是我们的第一个产品尝试。

我们现在做的模型已经到千亿级，未来会是一个多模态大模型，当前会先把语言模型做好。

36 氪：在做应用上，你们大概思考的方向是怎么样的？

杨植麟：我们还处在技术创新的阶段，所以我们会先持续追求世界级的技术突破，比如长上下文、多模态等。

而在产品层面，我们肯定是坚定在 To C 这一侧，希望能做头部的 Super App。以 ChatGPT 和 Character.ai 为例，这两个产品已经积累了大量的数据和用户反馈，有大量的迹象证明已经通过这种的产品产生了新的入口，新一代 AI 在“有用“和”有趣“两个方向上，都会有巨大潜力。

我相信，无论是智能助手还是情感陪伴，我们都能通过技术为更多人解决工作和生活中的实际问题。

36 氪：什么样的是真需求？

杨植麟：比如 Character.AI 的情感更多元化，他其实底层满足的是人的征服欲，我觉得征服是一个真正的刚需。

AI 最后不会是一个完全同质化的东西。它不像电，在新加坡充电和中国充电是一样的。所以像 Character.AI 最后所实现智能可能比其他公司会更强，因为他们有数据能一直积累，后面可以做一些专业化，这也导致以后 AI 的毛利率会比以前的云计算要高。

36 氪：好多大模型公司忙着在硅谷挖人，比如从 OpenAI、Google、微软。你是怎么组建起月之暗面的团队的？
杨植麟：我们很多人还是重新招的。我们更多是找这种 30 岁左右，有很多一手实践经验的人。从去年 12 月开始，我就去了一趟海外，开始为招人做储备了。
36 氪：海外的 AI 人才愿意回来吗？
杨植麟：我们在海外有 office，其实两边还是可以相结合的。
36 氪：现在月之暗面团队有多少人？你预想中的团队，会是什么样子？
杨植麟：我们的团队约 60 人，有很多技术专家，每个月都有在全球某个领域有显著影响力的人加入，我们在努力打造大模型公司里产品人才密度最高的团队。
互联网时代的技术和产品已经成熟分工，但我们希望产品团队能更直接地参与模型优化，大幅缩短创新周期。智能时代无论技术、产品、增长还是商业化，都存在创新的机会。我们的愿景是建立一个全新的组织，能与用户共情，也能用客观数据来定义美和智能标准，将科技与人文融为一体。
36 氪：OpenAI 会是这种组织的理想状态吗？
杨植麟：我觉得他们提供了很多很好的实践。比如他们就不搞赛马，这是非常重要的例子。
这并不是因为他们资源或者人不够。他们资源挺多，但是会把资源放到一个统一的 scope 下面。比如，他们希望花 10%的精力去探索一些新的东西，那会有一个团队在做这个事情，主线永远就只有这一个——这是非常重要的。并且，他们鼓励底层创新，每个人贡献想法。
36 氪：现在不少人关注成本问题，这直接关系到工程化的成本，还有后续的商业化进展。现阶段，你最关注的是什么因素？
杨植麟：就是能不能尽快找到 PMF，这是第一优先级。
36 氪：现在不少大厂、创业公司都在发开源模型，Moonshot 有开源计划吗？你怎么思考这个问题？
杨植麟：我们目前没有开源计划。我认为，开源和闭源在整个生态里面会扮演不同的角色，开源很大一个作用是在 To B 端的获客，如果想做头部的 Super App，大家肯定都是用闭源模型去做的，在开源模型上做 C 端应用很难做出差异化。
36 氪：你从博士阶段就已经开始创业，之前创立第一家 AI 公司“循环智能”的经验，会给你什么启发？
杨植麟：现在月之暗面还是处在第一阶段，更重要的任务是降低不可预测性等偏技术上的工作，其实不会太受到外部因素的影响。

但从大环境上来说，不可预测性肯定是要比之前更多了。几年前的年景更好，可以顺着市场做扩张，做营收；但市场不好时，反而是需要做成本控制、降低烧钱速度。这也是我从上一段创业经验学到最多的。

大模型很烧钱，把握好投入的速度，同时还要保证自己还是要拿出东西，有产品数据，是非常关键的问题。

预测下一个 token 是唯一问题

36 氪：AI 领域有几大方向：图像识别（CV）、自然语言处理（NLP）、机器学习（ML）。前几年 CV 更热闹，上一波 AI 四小龙（商汤、旷视、云从、依图）都是这个方向。你一直在做 NLP，为什么？

杨植麟：抛开偶然因素，还是有一些必然的原因。我觉得，Vision（视觉）方向其实更早地看到一些产业成果，但 NLP 可以去解决更多认知类的问题，让 AI 真正实现价值。

36 氪：NLP 怎么让 AI 真正发挥价值？

杨植麟：NLP 相当于是从视觉的感知层面，进化到更有认知的层面。

像 Midjourney 这种 AI 绘画产品，它可能生成的图片特别好看，但它本质是一个没有大脑的画家——你不知道中美关系怎么样，不知道印第安人以前是怎么被奴役的。你需要知道这些历史，才有可能成为一个顶级画家。甚至最后不光只是画画，你还要做很多画画之外的事情。

从这个点来说，NLP 会解决更难的、更有挑战性的问题，比如推理，它的存在会让 AI 的版图更加完整。

36 氪：Transformer 是你主攻的研究方向，它也是 ChatGPT 诞生的基础。Transformer 的革命性意义在什么地方？

杨植麟：我比较幸运的地方在于，我博士有一半时间是在 2017 年之后。因为 2017 年 Transformer 出来了，这是一个超级巨大的分水岭。

Transformer 架构的出现让整个 NLP 领域都发生了巨大的认知变化。有了这个东西之后，你就发现这里面可以做的东西实在太多了，突然一下子就给大家指明了方向。有很多之前完全无法实现的东西，它现在变得有可能了。

36 氪：怎么理解这个“认知层面的变化”？

杨植麟：AI 领域对语言模型的认知，存在三个阶段的变化：

2017 年前，大家觉得语言模型有一些有限的作用，比如在这些语音识别、排序、语法、拼写等等小的场景里面可以做辅助，但用例（Use Case）都很小；
第二个阶段：Transformer、Bard 出现后，语言模型可以做绝大部分的任务，但它还是一个辅助的角色——我有一个语言模型，AI 工程师微调一下任务就好了；
到第三阶段，整个 AI 领域发展到最后，大家的认知会变成：所有东西其实都是语言模型，语言模型是唯一的问题，或者说是 next token prediction（预测下一个字段）是唯一的问题。
这个世界其实就是一个硬盘模型，当人类文明数字化之后，所有人类文明之和就是硬盘的总和。输入的 Token 是语言，或者也可以是别的东西——只要能预测下一个 Token 是什么，那我就能实现了智能。
从思想到系统的层面，其实技术发生了非常大的变化，这里面有很多变量。然后你就可以在这个空间里面去看，怎么把这些技术做的更好。
36 氪：从 2017 年 Transformer 出现到今年 ChatGPT 爆火，中间还有五年的时间。这五年里，你的重要工作——有关 Transformer-XLNet 的论文，其实也有被拒稿过。中间有过对自己研究路线的怀疑吗？
杨植麟：这个很有意思。当因为行业发生认知变化，而变化还没有调整过来的时候，会存在非共识。
部分人觉得非共识是错的，但其实他实际上是对的。OpenAI 在这里面绝对是一个先驱，因为他们最早有这种正确的非共识，最早看到“语言模型是唯一的问题”这一点。
我们当时的研究效果非常好，能实现当时全世界最好的效果。但评审就问我们一个问题：就是说语言模型有什么用？你们好像没有证明他有用。
但是这个时候其实你要做的事情并不是说去寻求认同，而是说你要把真把那个事儿给做出来。
36 氪：你说“唯一重要的问题就是预测下一个字段。”这个事儿在当时如果是非共识的话，你是怎么意识到这一点，并且坚信的？
杨植麟：坦白说，我在那个时候还没有完全坚信这个事情，直到现在我觉得它也不一定是个共识，而是在逐渐变成共识的过程中。
36 氪：什么叫“预测下一个字段”，应该要怎么理解？
杨植麟：本质上，做下一个 token 的预测，其实等价于“对整个世界的这个概率去进行建模”，就是现在给你任何一个东西，你都能给他估算一个概率。
这个世界本来就是一个巨大的概率分布，里面有一些是不可建模的不确定性，你不知道下面会发生什么。但有一些是你能确定的，能排除掉一些东西的，这是一个通用的、对世界去进行建模的模型。有很多历史学家来对这个事情做过研究，比如 Density Estimation（密度统计），大模型本质是在做这样一个事情。
但当时我只意识到这是个重要的问题，而没有意识到是唯一要解决的问题。
36 氪：那是什么时候让你改变主意了？

杨植麟：2020 年 GPT-3 出来的时候，那个时候有了更明确的证据。OpenAI 的人最厉害的点是，他们观察到了更多的数据，再更早的时候真正去把模型参数、训练规模扩大，所以他们更早地知道只要一直 scale（扩大规模），就可能解决所有的问题。
36 氪：知道它是如此重要之后，这会怎么影响你的技术路线？
杨植麟：回到刚刚那一点，如果这个世界只有一个问题：要预测下一个字段，那么输入和输出其实是一样的——也就是“理解”和“生成”其实也是同一个问题。
几年前，我们自己也会区分，到底是要做理解模型还是生成模型，但现在不需要了。
36 氪：不过，现在有很多团队的技术路线，可能会先做文字理解，在理解这一端做得更多些，生成可能会靠后一点。
杨植麟：这些思考方向不够本质。现在任何说“只能做理解而非生成”都是错误的方向。正确的方向应该是：理解和生成就是一个问题。如果能做很好的理解，那能做很好的生成，这两个应该是完全等价的。
36 氪：相当于这两者无法分开来。
杨植麟：对的。现在就只有一个问题。比如说我能够去生成接下来 10 秒钟的视频，我那我必须对之前的这个视频有很好的理解，你得知道他发生了什么，这是一个什么样的 story，接下来很有可能是什么样的演进，它是分不开的。
36 氪：你对实现 AGI（通用泛化的智能）有信心吗？
杨植麟：有没有信心取决于它的第一性原理，我觉得大家现在已经明白原理了，只有一个问题：就是预测下一个字段。一条道走到黑的话，我觉得就能实现。
但确实还存在一些“第二层面”问题，也就是具体的技术方向难题。但是这些都是小问题，并非原则性的，第二个层面就是我们要去攻克的。

人的一生不过是大量的信息

36 氪：用一句简短的话来描述月之暗面的目标跟远景，你会怎么说？
杨植麟：长期的几个目标是：探索智能的极限、让 AI 有用，以及让每个人都能拥有真正普惠的 AI。
36 氪：“普惠的 AI”怎么理解？
杨植麟：现在的一个问题是，很多时候 AI 的价值观是被一个处于中心的机构控制。一个模型表现成什么样子，完全是由平台来决定——TA 觉得什么是“好的”，什么是价值观正确的答案。
但每个人会有自己的价值观。价值观是更底层的东西，它其实还包含很多可能——你的偏好，也就是你认为什么是对的，什么是错的。
每个人都应该要有这种个性化定制的机会，所以以后的 AI 也应该要拥有“对齐”的机会。（Alignment，指确保 AI 系统的行为匹配预期的人类价值观和目标的过程）。
当然，我们肯定要去设置安全底线，以及监管层面的东西。在这个基础上，可以有很多个性化 AI 的机会。
36 氪：个性化的 AI，它的实现路径是什么？每个人都能训练一个代表自己的 AI 模型吗？
杨植麟：你刚说的训练是一种方式，但我认为可能后面也许不需要去训练，也许直接设置就可以了。
最终的一个可能形态是，AI 会数字化的所有东西全部记录下来，你的手机、电脑上会有一个和你共生的 AI Agent（AI 代理、AI 分身），它会知道所有一切你能知道的东西。
36 氪：你在你的个人主页上写，你的所有的工作目标都是“让 AI 价值最大化”。这指的是什么？
杨植麟：最大的价值就是，最终每个人不用做自己不想做的事情，保留人性里面最精华的部分。比如，我们这次谈话也可以不用面对面，而是有更高效的方式——比如由我们的 AI Agent 直接对话。在公司也是一样，现在的组织要花时间去定绩效、考核。其实这都会非常花时间。以后我们也许就不需要公司了，一个人的效率会高很多，也不用为了赚一点钱就非得要去上班，可以用 AI 来做很多工作。
要达到这样的效果肯定很难，但最终人类有可能实现生产最大化。最后，也许真正的共产主义会出现。
36 氪：如果让你现在对未来做一个预测的话，你觉得十年之后我们这个社会会有什么样的变化？或者说 AI 对这个社会最大的变革，你觉得会来自什么方面？
杨植麟：十年有点难，五年可以说一说。

我觉得至少五年内大模型技术不会 commoditize（指技术还会有壁垒，不会变成廉价的商品）。因为至少还有一大批模型没有出来，我们还没有真正看到视频大模型。
我觉得这两年可能是文本模型持续迭代的窗口。后再过三年，是视频模型持续迭代的窗口，这里始终是有技术壁垒的。
36 氪：所以，视频大模型会是关键性的节点？
杨植麟：对的，这些节点都迈过后，会出现一个巨大的变革。
美国有一个公司叫 Rewind（主打“记录一切”，让人类搜索一切在上看见过的所有内容），现在的产品能实现的效果，可能只是能问它：我上个月做了什么？它会记录下来，现在的效果还是比较浅层的。
以后的 AI Agent 会更加深度地实现个性化。比如，大模型会和你有共享的记忆，知道你所有的价值偏好，所有的价值取向。如果你让他写一个 Q3 的规划，他会基于已知的这些东西直接去写规划，而不需要知道 Q2 做了什么东西。
36 氪：从文字到图片，再到视频大模型、Agent，要实现的关键是什么？
杨植麟：是 context（上下文长度，也可以理解为模型单次能处理的信息量），这基本决定了 AI 能产生价值的上限。
如果大模型的 context 就是你的全部记忆，理论上，那它就可以做你现在做的全部事情。
对于大模型来说，最关键的一点就是，你到底能有多少 context 被捕捉到。这取决于视频模型的能力，如果模型能力很强，理论上你的手机和电脑加起来就差不多是你完整的 context。
人的一生也不过是如此，我们每天就活在数字世界里面。可能除了我们现在这种线下对话，他可能捕捉不到，其他大部分都是都 ok 的。
36 氪：如果真的达到这种状态，人类应该要怎么和机器共存？
杨植麟：我自己是比较乐观，就是说他在提供更多生产力的同时，他应该会创造很多新的岗位。视频现在是大家花时间最多的地方，所以他肯定会对生产关系产生很大的影响。所以每个人可能都可以生产（视频），很多价值会被重新分配。
但这是一个反馈闭环时间比较长的事情。挑战在于，当前替代现有岗位的速度比创造新岗位的速度更快。核心问题在于，在理想的岗位没有被创造出来之前，我们如何解决一些社会问题。
36 氪：普通人怎么去面对这次技术变革？这种变化继续下去，普通人应该做点什么？
杨植麟：我觉得最重要还是学习。不光是普通人，我觉得所有人，拥有最强终身学习的能力的人，以后才能够实现自己真正的价值。
另外一点是要 open minded。我四五年就找过很多人说，要不要来一起做大模型，当时他们说我现在要做数字人，你不要跟我讲这些东西（笑）。所以人确实有时候还是会被自己认知所局限。无论我们对技术的态度如何，历史的发展都是超出个人意志的。因此，我们要不断的自我迭代，适应这个世界唯一不变的，就是变化本身。

久等了，欢迎与 Moonshot AI 共同开启 Looooooooooong LLM 时代
https://mp.weixin.qq.com/s/stKAB8wX7xWj2Js8FSoJpA
今天，Moonshot AI 带着首个支持输入 20 万汉字的智能助手产品 Kimi Chat 与大家见面了。
据我们所知，这是目前全球市场上能够产品化使用的大模型服务中所能支持的最长上下文输入长度，标志着 Moonshot AI 在“长文本”这一重要技术上取得了世界领先水平。
为什么说大模型的“长文本”能力很重要？
因为从技术视角看，参数量决定了大模型支持多复杂的“计算”，而能够接收多少文本输入（即长文本技术）则决定了大模型有多大的“内存”，两者共同决定模型的应用效果。支持更长的上下文意味着大模型拥有更大的“内存”，从而使得大模型的应用更加深入和广泛：比如通过多篇财报进行市场分析、处理超长的法务合同、快速梳理多篇文章或多个网页的关键信息、基于长篇小说设定进行角色扮演等等，都可以在超长文本技术的加持下，成为我们工作和生活的一部分。
相比当前市面上以英文为基础训练的大模型服务，Kimi Chat 具备较强的多语言能力。例如，Kimi Chat 在中文上具备显著优势，实际使用效果能够支持约 20 万汉字的上下文，2.5 倍于 Anthropic 公司的 Claude-100k（实测约 8 万字），8 倍于 OpenAI 公司的 GPT-4-32k（实测约 2.5 万字）。
同时，Kimi Chat 通过创新的网络结构和工程优化，在千亿参数下实现了无损的长程注意力机制，不依赖于滑动窗口、降采样、小模型等对性能损害较大的“捷径”方案。

目前，Kimi Chat 已开放内测。
访问 https://www.moonshot.cn 或扫描下方二维码，即可加入内测计划。

大模型输入长度受限带来的应用困境
在我们看来，当前大模型输入长度普遍较低的现状对其技术落地产生了极大制约。例如：
目前大火的虚拟角色场景中，由于长文本能力不足，虚拟角色会轻易忘记重要信息，例如在 Character AI 的社区中用户经常抱怨“因为角色在多轮对话后忘记了自己的身份，所以不得不重新开启新的对话”。
对于大模型开发者来说，输入 prompt 长度的限制约束了大模型应用的场景和能力的发挥，比如基于大模型开发剧本杀类游戏时，往往需要将数万字甚至超过十万字的剧情设定以及游戏规则作为 prompt 加入应用，如果模型输入长度不够，则只能削减规则和设定，从而无法达到预期游戏效果。
在另一个大模型应用的主要方向——Agent 中，由于 Agent 运行需要自动进行多轮规划和决策，且每次行动都需要参考历史记忆信息才能完成，这会带来了模型输入的快速增加，同时也意味着不能处理更长上下文的模型将因为无法全面准确的基于历史信息进行新的规划和决策从而降低 Agent 运行成功的概率。
在使用大模型作为工作助理完成任务的过程中，几乎每个深度用户都遇到过输入长度超出限制的情况。尤其是律师、分析师、咨询师等职业的用户，由于常常需要分析处理较长的文本内容，使用大模型时受挫的情况发生频率极高。
而上述所有的问题在大模型拥有足够长的上下文输入后都将会迎刃而解。
长文本打开大模型应用的新世界
那么拥有超长上下文输入后的大模型会有怎样的表现？下面一起来看一些 Kimi Chat 实际使用的例子：
公众号的长文直接交给 Kimi Chat，让它帮你快速总结分析：
新鲜出炉的英伟达财报，交给 Kimi Chat，快速完成关键信息分析：
出差发票太多？全部拖进 Kimi Chat，快速整理成需要的信息：
发现了新的算法论文时，Kimi Chat 能够直接帮你根据论文复现代码：
只需要一个网址，就可以在 Kimi Chat 中和自己喜欢的原神角色聊天：
输入整本《月亮与六便士》，让 Kimi Chat 和你一起阅读，帮助你更好的理解和运用书本中的知识：
通过上述例子，我们可以看到，当模型可以处理的上下文变得更长后，大模型的能力能够覆盖到更多使用场景，真正在人们的工作、生活、学习中发挥作用，而且由于可以直接基于全文理解进行问答和信息处理，大模型生成的“幻觉”问题也可以得到很大程度的解决。
不走捷径，解决算法和工程的双重挑战
其实长文本技术的开发，存在一些对效果损害很大的“捷径”，主要包含以下几个方面：
“金鱼”模型，特点是容易“健忘”。通过滑动窗口等方式主动抛弃上文，只保留对最新输入的注意力机制。模型无法对全文进行完整理解，无法处理跨文档的比较和长文本的综合理解（例如，无法从一篇 10 万字的用户访谈录音转写中提取最有价值的 10 个观点）。
“蜜蜂”模型，特点是只关注局部，忽略整体。通过对上下文的降采样或者 RAG（检索增强的生成），只保留对部分输入的注意力机制。模型同样无法对全文进行完整理解（例如，无法从 50 个简历中对候选人的画像进行归纳和总结）。
“蝌蚪”模型，特点是模型能力尚未发育完整。通过减少参数量（例如减少到百亿参数）来提升上下文长度，这种方法会降低模型本身的能力，虽然能支持更长上下文，但是大量任务无法胜任。
我们相信，走这些捷径无法达到理想的产品化效果。为了真正做出可用、好用的产品，就应该直面挑战。
具体来看。训练层面，想训练得到一个支持足够长上下文能力的模型，不可避免地要面对如下困难：
如何让模型能在几十万的上下文窗口中，准确的 Attend 到所需要的内容，不降低其原有的基础能力？已有的类似滑动窗口和长度外推等技术对模型性能的损害比较大，在很多场景下无法实现真正的上下文。
在千亿参数级别训练长上下文模型，带来了更高的算力需求和极严重的显存压力，传统的 3D 并行方案已经难以无法满足训练需求。

缺乏充足的高质量长序列数据，如何提供更多的有效数据给模型训练？
推理层面，在获得了支持超长上下文的模型后，如何让模型能服务众多用户，同样要面临艰巨挑战：
Transformer 模型中自注意力机制（Self Attention）的计算量会随着上下文长度的增加呈平方级增长，比如上下文增加 32 倍时，计算量实际会增长 1000 倍，这意味着如果只是用朴素的方式实现，用户需要等待极其长的时间才能获得反馈。
超长上下文导致显存需求进一步增长：以 1750 亿参数的 GPT-3 为例，目前最高单机配置(80 GiB * 8)最多只能支持 64k 上下文长度的推理，超长文本对显存的要求可见一斑。
极大的显存带宽压力：英伟达 A800 或 H800 的显存带宽高达 2-3 TiB/s，但面对如此长的上下文，朴素方法的生成速度只能达到 2~5 tokens/s，使用的体验极其卡顿。
在过去半年多的时间里，Moonshot AI 的技术团队进行了极致的算法和工程优化，克服上述重重困难，终于完成了大内存模型的产品化，带来了首个支持 20 万字输入的千亿参数 LLM 产品。
“登月计划”第一步：欢迎来到 Long LLM 时代
Moonshot AI 创始人杨植麟此前在接受采访时曾表示，无论是文字、语音还是视频，对海量数据的无损压缩可以实现高程度的智能。
无损压缩的进展曾极度依赖「参数为王」模式，该模式下压缩比直接与参数量相关，这极大增加了模型的训练成本和应用门槛，而 Moonshot AI 认为：
大模型的能力上限（即无损压缩比）是由单步能力和执行的步骤数共同决定的。单步能力与参数量正相关，而执行步骤数即上下文长度。
我们相信，更长的上下文长度可以为大模型应用带来全新的篇章，促使大模型从 LLM 时代进入 Long LLM (LLLM)时代：
每个人都可以拥有一个具备终身记忆的虚拟伴侣，它可以在生命的长河中记住与你交互的所有细节，建立长期的情感连接。
每个人都可以拥有一个在工作环境与你共生（co-inhabit）的助手，它知晓公域（ 互联网）和私域（企业内部文档）的所有知识，并基于此帮助你完成 OKR。
每个人都可以拥有一个无所不知的学习向导，不仅能够准确的给你提供知识，更能够引导你跨越学科间的壁垒，更加自由的探索与创新。
当然，更长的上下文长度只是 Moonshot AI 在下一代大模型技术上迈出的第一步。我们计划凭借该领域的领先技术，加速大模型技术的创新和应用落地，不断取得更多突破。
听听“登月计划”的伙伴是怎么说的：
Monolith 砺思资本创始合伙人曹曦：
“杨植麟是全球大模型领域里最被认可的华人技术专家，其团队在人工智能技术，特别是大语言模型 LLM 领域拥有深厚的技术积累，并已在国际上获得了广泛认可。眼下，美国硅谷的 OpenAI 和 Anthropic 等公司获得了多方关注，实际上在国内，拥有足够多技术储备的 Moonshot AI 也正成长为全球领先的 AGI 初创公司。多模态大模型是各家 AI 厂商竞争的关键领域，其中长文本输入技术更是其核心技术之一，Moonshot AI 团队最新发布的大模型和 Kimi Chat 在这方面实现了重要突破，并已成功应用于多个实际场景。砺思将继续加码并支持 Moonshot AI 团队在 AGI 领域大胆创新和技术突破，引领中国人工智能技术的未来发展。”
真格基金合伙人戴雨森：
“我们认为近期 AI 应用的爆火只是一场革命的序幕，AI 技术要想真正改变世界创造巨大价值，在智能程度上还需要大的突破，这需要具备顶级技术能力的团队，以坚持追寻 Moonshot 的勇气，持续挑战智能提升的边界。杨植麟作为 XLNet 等多项知名科研工作的第一作者，具备非常丰富的科研和实践经验，多年来他一直坚信通过大模型实现对高维数据的压缩是人工智能发展的必经之路，也团结了一支人才密度超高，配合默契，又充满挑战巨头摇滚精神的创业团队。真格基金非常荣幸能够再次从天使轮开始支持杨植麟的新征程。”

Long Context 解决 90% 的模型定制问题，Moonshot AI 发布开放平台
https://mp.weixin.qq.com/s/TWNFX5xSegjVSGcq20SEjQ

云栖大会是一年一度的全球顶级科技盛会。2023 云栖大会于 10 月 31 日至 11 月 2 日在杭州举办，吸引了全球 44 个国家和地区的 8 万多名从业者参会。今年最受关注的话题莫过于大模型技术。在 11 月 1 日上午举办的“AI 大模型新势力”论坛上，主办方邀请了国内知名 AI 创新的负责人分享大模型的当下与未来。

Moonshot AI 工程副总裁许欣然受邀在该论坛发表了题为《Moonshot AI：寻求将能源转化为智能的最优解》的主题演讲，以下是现场演讲视频和实录。

Moonshot AI：寻求将能源转化为智能的最优解

大家好，我是许欣然，Moonshot AI 公司的工程副总裁，很高兴能在这里给大家分享 Moonshot AI 在大模型发展路线方面的思考。

Moonshot AI（月之暗面）是一家专注于通用人工智能领域的公司。我们团队中很多人曾作为核心成员参与了国内外知名大模型的研发，并且发明了像 Transformer XL、RoPE 这样的关键算法。

我们的愿景是，致力于寻求将能源转化为智能的最优解，通过产品与用户共创智能，实现普惠 AI。

上个月我们的第一款产品 Kimi Chat 开启了内测，可能很多朋友就是通过几张图了解到我们公司的。

左边是自动整理发票场景：一次性给它可能几十张发票，让它帮忙整理一下，然后统计一下满足条件的一些发票。中间是给他一个十几万字的长报告，让他帮忙分析。

最右边的例子是直接把 arXiv 上的一篇论文交给它，让它根据论文里面的伪代码去直接编写对应的一些 Python 的示例代码，非常好用。

在这里非常感谢参与内测的用户对我们的认可和鼓励。整个过程中我们也非常惊喜地发现，大家的认可和各种创造性的使用方式，都跟我们最初的想法不谋而合。那就是要充分利用 Long Context 技术。

Long Context 走向“最优解”的第一步

Long Context 是我们本次产品中最核心的能力，也是我们认为迈向公司愿景——“将能源转化为智能的最优解”的第一步。

Kimi Chat 目前支持长达 20 万字的上下文处理能力，我们认为这个长度是可以满足绝大部分场景使用诉求的。当有了这么长的上下文处理能力之后，我们就发现，它可以直接解决很多以前由于上下文窗口不够大带来的问题。

我们来看一些例子。比如过去我们如果要翻译一篇非常长的文档，这份文档可能会超过一个上下文的窗口。以前的做法就会把它切割成若干个小段，分别送给模型。这个时候同一个词，比如说 chair，可能在前文被翻译成了“主席”，而到了后文可能就被翻译成了一把“椅子”，出现很尴尬的情况。更不用提在一些比较专业的词汇上，可能前面有一些缩写的解释，那个时候它可以正常翻译，而到了后文就翻译不对了。

之前，大家为了解决这些问题想了很多工程优化的方法。比如说在切段的时候，尽可能让前文和后文重合一些，让模型尽可能的多了解一些，或者说人工总结一些关键词的词表，保证模型在这些比较关键的词汇上不要犯错。这些都是一些临时的解决方法。

但是还有很多场景可能不是很好这么绕过去。

比如在 Agent 场景下，可能一些比较复杂的任务，根本就装不到 Context 里，还没有描述清楚，上下文就已经超出长度限制，就没办法做了。

再比如说像一些比较角色扮演的那种情况下，可能没聊几十句，模型就忘了最开始的角色定义，就开始放飞自我，又变回了一个普通的 AI 小助手。我们发现在上下文足够长了之后，这些问题就全都迎刃而解了。模型的表现会变得非常的好。

我们的内测用户的反馈也印证了我们的想法。大家把 Kimi Chat 玩出了很多新奇的花样，很多人在跟我们讨论一些比较有意思的想法。比如说一种 Lifetime 的 Chat，比如说可以做到像微信里你的一个朋友一样，你一直跟它聊，它可能记着你长达过去一年甚至几年的这种聊天的内容。那你可以随时跟它提，上个月跟你提到的什么东西，怎么样了，会有非常好的体验。再比如说，可能有些人直接把一些连载小说放到我们的模型里面去，让模型帮忙续写，效果也非常好。

刚才讲到了我们公司的愿景是寻求将能源转化成智能的最优解。同时这也是我今天演讲的主题。

Long Context 是我们迈出的第一步。它的确直接解决了很多痛点，也释放出了很多大家没有想到的想象空间。不过我们之所以把 Long Context 列成第一步，还有一个非常重要的理由。

那就是我们认为 Long Context 是解锁模型定制与模型迭代之间矛盾的钥匙。

AI 行业一直有一个很严重的问题就是碎片化，也就是说不同的场景会有非常强烈的定制化诉求。

我们跟每一个客户去聊起来，都会迅速的变成定制化，想要 fine-tune 一个自己的模型。

现在想要 fine-tune 一个模型成本是很高的。不管是要去搞数据，还是去做一些工程上的开发。但是大家仍然会咬着牙去做这件事情。那就说明模型现阶段的通用性还不能解决这个问题，定制化还是一个非常强的诉求。

为什么呢？我们跟大量的客户进行了沟通，可以把大家的需求和痛点分成这三类。

首先，客户有非常强烈的差异化诉求。大家最强的诉求是希望“我有你没有”。比如一些投行的客户跟我们去聊。他们希望模型拥有一些独到的见解、独特的分析方法，能帮助他们对公司、对财报做分析。但是这件事情里边重要的是他们要有这个能力，更重要的是别人不能拥有他这样的分析能力。再比如说，有一些客户希望用我们的模型去做二次的封装，去做一些创意或者营销类的工作，他们也会希望有一些别人没有的独特的特色和风格，这些都是差异化的诉求。

第二，客户希望融入自己的领域专有知识。企业客户天然存在非常丰富的领域知识，举一个非常简单的例子，一个公司内部开会，他们用什么样的方式，用什么样的格式去记内部的会议纪要，其实就是一种模型再怎么聪明也没有办法直接知道的信息，必须要用那个公司内部的知识才能得到这个信息。

第三，客户希望模型可以兼容已有的业务逻辑。在绝大多数的现有业务系统里，算法模块的边界已经确定了，这些模块并不能因为要接入大模型，就要配合大模型的行为去调整，这样会跟存量系统无法兼容。

在大模型出现了之后，这几项其实并没有因为模型变聪明了就直接解决了，或者说其实是解决不了的，反而会随着算法能力的增加，大家对模型想象力的放飞，进一步去增强定制化的诉求。

也就是说，客户为了提升自己的行业竞争力，就一定会有这种很强烈的定制化的诉求。而大模型的供应方因为想要迭代模型，也希望控制成本，就会抵触这种过度的定制化。这里就存在一个非常尖锐的矛盾。

而我们认为 Long Context 就是解决这个问题的钥匙。

我们意识到，用 Long Context 可以在上下文里放足够多的信息，当模型接收到足够多的信息时，它的行为就足以达到定制化的水平。比如说：

在角色扮演的场景，我可以把非常详尽的人物世界观定义，这个角色的特点，包括它的一些说话风格，跟其它角色的关联，全部放入模型的上下文当中，那么模型的行为方式和说话风格就非常满足定制的要求了。

在常见的客服场景， 当我尝试把家里汽车的产品手册，还有客服的服务手册完整放到 Context 里面，这个时候模型就能用一种非常标准的客服方式跟我去做沟通交流，能够回答我对这辆汽车的各种各样的问题，可以是一些比较具体的，比如说车灯怎么开这种文档里有的。也可以是一些比较模糊的，比如我觉得灯太暗怎么办。

还有我们公司内部的一种做法。在筛选简历的时候，我们会有一份内部的文档。这个文档是讲解我们如何去筛选候选人的，描述了我们的人才偏好是什么。当我们把这个文档和以往选中和没选中的人才简历都提供给模型时，模型就能直接帮我们筛选简历，而且我们的测试效果非常好。

有了 Long Context 技术，AI 的定制化问题就有了不同的解法。

Long Context：解决 90% 定制问题

现在，我们几乎可以判断，有了 Long Context，在很多情况下就不再需要做 fine-tune 这么麻烦的事情了，Long Context 可以解决 90% 以上的定制化问题。

这个结论可以从很多个层面来验证，我把它简单总结成“多、快、好、省”。

第一个是“多”。我们自己的经验是用 5 万字足以详细地定制一个模型的能力了。一般提示词（prompt）里边可能可以包含像角色的定义，你希望它作为一个什么角色去完成什么样的事情，然后再额外提供给它三到五个类似的样例，这个时候模型的行为就可以被非常充分地定制了。而在做完了这件事儿之后，还剩很多字。因为我们的上下文窗口足够的长，这样日常的使用其实就足够用了。

第二就是“快”。提示词这种定制的方式的效率会远远高于大家用 fine-tune 先要构造数据，然后再从数据中训练的过程。大家可以想象一下，如果今天你要定制一款模型，或者提出一种特殊的需求的时候，一般你手中有的都是一些需求的指令，而不是具体的一些 demo 或者一些样例的数据。那么如果你直接把这个原始的需求直接交给模型，你发现模型它就已经可以做到很多了。

如果要通过 fine-tune 的方式，你可能需要找一名有算法经验的人，根据这些材料去总结生成一份对应的数据集，再用数据集进行训练。而且这个数据集还有一个很重要的要求，就是一定要符合你的业务场景的数据分布。因为万一数据分布不对，或者你生成数据的时候没有考虑到某些情况，模型最后训练出来遇到没有考虑到的情况就应付不了。

更不用提，假设今天突然想临时给线上的系统增加一个新的规则，比如说某些东西我希望它额外提到一下，或者某些内容不要提，那么加一条额外的指令，要比大家完整的去做一次 fine-tune ，再加上最后训练，效率要高得多。

第三个是“好”。提示词（prompt）是一种人类的语言，大家可以用一种可以跟人类交流的自然语言去跟模型交流，它会有天然的兼容优势。这个兼容的优势在于未来可以随着模型的能力提升而提升效果。甚至我们在切换大模型供应商的时候是非常轻松的，你不会被完全绑定。

第四个是“省”。因为在 Long Context 的情况下，模型是同一个标准的模型，所以在调用方不需要去摊销固定的部署成本。比如说像 OpenAI 要 fine-tune 一个模型，单位 token 的成本其实是你

调标准模型的十倍之多。那就更不用提写一个 prompt 的成本，还是要比构造数据集去做训练要成本低得多，人员要求和成本也远低于 fine-tune 所需的成本。

通过 Long Context 来做定制化的方法，其实在一些比较著名的国际大厂里面已经验证了，他们有很多的算法工作直接用 Prompt Engineering 这种方式来实现，效率非常高，而且成本也显著地低了很多。

我们自己内部的大量“算法需求”，也会转化成“写一段 Prompt”就可以解决，一般来说迭代的周期也就是五分钟，就可以快速验证一轮，然后去调整细节。

在“多、快、好、省”这几个好处之中，促使我们最终把 Long Context 做成最高优先级的其实是“好”，prompt 更好写，也更好向前向后兼容。

我们意识到，当模型有这种兼容性了之后，Moonshot AI 在模型快速迭代、成本持续下降过程之中，我们的用户可以几乎不增加成本地共同享受到成果。大家用 prompt 去专注于自己的业务场景，可以去定制自己的需求，然后去持续的优化。而不会因为用了 fine-tune 而被永远的锁定在我们的某一个版本比较早期的模型上。

也就是说，我们用户可以伴着 Moonshot AI 的模型能力迭代而持续获得收益。

这让我想到了塑造整个计算机行业的摩尔定律。

在几十年前，CPU 的单核性能的发展速度其实是非常快的，就跟现在的大模型一样日新月异。那个时候其实有很多公司会针对某一款、某一代 CPU 去写汇编，去特殊的做优化。他们就会很快发现，这些优化在很短的时间内就变成了一个负资产。因为一旦“特化”了之后，代码就永远锁定了那一代的 CPU 可能没有办法升级，或者升级之后效果表现会不好。不出半年到一年的时间，他们都会被一些比较通用的代码加上最新的 CPU 给远远的甩在身后。这些公司的竞争对手可能什么都没有做，就等了一段时间，还省了固定的成本，就吃到了这个红利。

我们认为这些“特化”工作恰如今天大家想要做 fine-tune 一样，也会遇到相同的问题：跟不上模型的进化速度。

所以我们希望通过 Long Context 来做到这一点，依靠更长的上下文让大家高效地定制模型的行为和特点，从而让每一个用户能够愿意看到我们 Moonshot AI 的模型，每隔一段时间又提升了，又迭代了，效果更好了，成本更低了。

而不是跟我们说，你们又升级了，我老的模型，我老的 fine-tune 那个怎么迁移……

这就是我们把 Long Context 作为迈向公司愿景（寻求将能源转化成智能的最优解）第一步的另一个关键原因。

Kimi Chat：一个好用的助手

最后我再回顾一下 Moonshot AI 基于 Long Context 技术打造的第一个产品 Kimi Chat。

Kimi Chat 拥有非常强的上下文的处理能力，前文提到了有 20 万字的处理能力，同时它支持各类文档的解析功能，可以解析 PDF、Excel、CSV 等各种各样的格式，这些文档你都可以放进去，一次可以放很多条。同时，在缺乏信息的时候，它会像人类一样去调用搜索引擎去看前五到前十个网页。因为 context 非常长，所以它可以把所有的这些网页里的每一个细节都读完，而不是只是读一个摘要。

大家会发现，有了超长上下文之后，长文档的解析和 Web Copilot 这两个功能几乎是非常直接，也非常自然就能想到并快速应用的。

我们通过内测阶段用户的反馈，还发现有些用户非常聪明，他们用 Kimi Chat 完成了一些非常成功的尝试。比如说：

把整个源代码放到我们的 Kimi Chat 里，然后跟它说请你帮我根据这份代码编写一个流程图，我要用来去写专利或者软著。他们的专利或软著材料有一多半都可以直接用模型生成，效果非常好。
有一些用户在做标书的时候，他把三四封以前自己写过的标书放给模型，说仿照这个标书，你要注意什么？不注意什么？今天请你照着那个写一份新的标书。他们发现模型的效果很好。
最有意思的是把整个全文，就是一个很长的英文文档都放到模型里边，但是只贴其中一小段，就跟它说请你只翻译这一段。我们发现在这种情况下，不管是多么复杂的技术文档，模型翻译质量都特别好。因为它有非常充分的上下文，它知道每一个缩写，知道每一个细节跟其它的组件的交互，所以它的翻译质量会非常的高。大家可以想象，非常爱追剧的朋友们可能知道字幕组翻译水平不稳定，也许以后字幕组翻译的时候，也可以利用这种方式，让翻译质量更高。
Moonshot AI 开放平台发布

还有一些用户在内测之后觉得 Kimi Chat 的 Long Context 能力是非常好的，一直在催促我们希望有面向开发者的 API。

从今天开始，我们的开放平台也启动内测申请了。

大家可以扫描上方右侧的二维码申请。我们非常希望有想法、有创意的用户能够跟我们一起探究将能源转化成智能的最优解。我们会持续提升模型的能力，并且持续地降低使用成本。

谢谢大家。

月之暗面杨植麟：大模型需要新的组织范式，场景摩尔定律能催生 Super App
https://mp.weixin.qq.com/s/499NG03U3jC-S_9K6ek8pA
月之暗面 Moonshot AI 是一家神秘且特别的大模型创业公司。

公司目前只发布了一款产品，基于千亿大模型的 chatbot 产品 Kimi Chat。发布之初，就打出了「长文本」、「自研闭源」、「toC」等清晰的标签。

创始人杨植麟饱受期待，他毕业于 CMU，师从苹果 AI 负责人 Ruslan Salakhutdinov，曾在 Meta 和 Google Brain 任职，是 Transformer-XL 与 XLNet 等爆款论文的第一作者。

但在这次直播中，他更多地聊了聊战略、组织、产品和人才等等——除了技术以外，作为创业者需要关注的一切。

大模型的产品经理，需要具备怎样的特质？

以下是极客公园创始人 & 总裁张鹏与月之暗面 Moonshot AI 创始人 & CEO 杨植麟的对话，经 Founder Park 编辑。

01

大模型时代，对组织形态提出了新的要求

张鹏：大家都说你们公司有点神秘，尤其这个名字——「月之暗面」，先来给我们揭秘下「月之暗面」这个名字背后有什么讲究？

杨植麟：我们的名字其实来源于一张摇滚专辑，Pink Floyd（摇滚乐队）的《The Dark Side of the Moon》（月之暗面）。因为我们（创始人们）都比较喜欢摇滚，以前也是玩乐队的，今年也刚好有个契机，是月之暗面发行 50 周年。

我们平时看月亮，都只能看到发光的一面，看不到背后，但是你会有一种很强的冲动，想要去探索神秘的月球背面。

这和大模型很相似，你很想去探索一个神秘的、未知的东西。它很难，很有挑战性，同时又可以结合很多摇滚的底层精神，不断地创新，不断地挑战事物已有的形状，去想象接下来可能会是什么样。

我们当时想了各种名字，最终选择了月之暗面和 Moonshot（登月计划）的中英文组合，它可以比较好地去反映我们对于做 AGI 的决心，以及——它或许可以定义，我们是怎样的人。

《The Dark Side of the Moon》是 Pink Floyd 最受好评的专辑，专辑以以疯狂为重点，探讨了冲突、贪婪、时间、死亡和精神疾病等主题丨图片来源：Wikipedia

张鹏：我也是 Pink Floyd 的粉丝，从我的角度来看，这张专辑虽然用了个很天文的名字，但本质上讲的东西可能更接近人的潜意识。

杨植麟：对，里面的那个主打的歌曲叫 Brain Damage，讲的就是一个人出现了 hallucination（幻觉）。我们现在就是穿越了 50 年，要去拯救大模型的 hallucination 问题。

张鹏：八卦一下，你原来在乐队里是哪个位置？

杨植麟：原来是鼓手。张鹏：鼓手在乐队里大概是一个什么样的定位？

杨植麟：我觉得是掌握节奏，为整个乐队的演奏提供一个框架。

构建新的组织形式是通往 AGI 的必经之路

张鹏：投身到大模型赛道里。你当时是怎么做出这个决心，并选择要做一个组织来投身其中的？能不能分享下当年的决策逻辑？

杨植麟：我的认知在过去几年里面发生了非常大的变化。

一开始觉得语言模型可能是个工具，可以去提升很多不同场景的效果；第二个阶段，认为语言模型可能对很多任务都有用。后来大家认为语言模型可能成为 AI 唯一的一个（要解决的）问题——所有的问题，都可以通过把语言模型做得更好，把 next token prediction 做得更好来解决。

2018、2019 年，在 Google 开始用几千张卡，基于 Transformer 训练语言模型的时候，过程中会观察到非常多的现象，这些现象进一步提供了更多的证据，证明这条道路是正确的。可能只需要往这个路径一直走下去，然后不断地去寻找更高效的 scale 它的方式，就可以得到一个非常好的结果，能够解决很多以前很难解决的问题——不管是记忆的问题、推理的问题，还是很多常识，甚至是更复杂的多链路问题。

这个经历给了我很深的冲击，或者说创业的铺垫。

2020 年开始，我去找很多机构合作，一起去训练大模型，也是最早在国内训练了很多像盘古、悟道大模型，这个过程也一直在酝酿一个真正的时机。

同时在这个过程中，我也看到很多大模型面对的挑战。这种挑战一方面是来自于技术，另外一方面可能是来自于组织。我们发现，如果你还是用传统的组织结构，对训练大模型来说，可能很难成功。我们今天看到 OpenAI 的成功，本质上，也是因为它的组织做了极大的创新。

所以我觉得之前的经历，可以理解成我一直在寻找一个机会——怎么样能够去从零去建造一个新的组织。我认为这是通往 AGI 的必经之路，它甚至比我们今天接触到的各种细枝末节的技术还要更重要。因为组织是一个更底层的东西，只有你把这个组织做好，才有可能真正地在 AGI 这条路上好好走下去。

去年开始，不管是资本市场，还是人才市场，都发生了极大的变化。在这种情况下，我觉得时机更加成熟，我们有机会去从零搭建一个组织去做这件事。

02

大模型时代的创新很难被规划

底层逻辑变了，做事的方式也要变

张鹏：是什么让你认为组织是一个核心的问题，以至于要利用这个契机去构建这样的组织？

杨植麟：更多的还是实践。

今年之前我们也尝试过很多，用不同的方式，不同的组织。一种就是传统的，在企业内部去做事的方式。还有就是我也参与了一些独立研究机构（的工作）；甚至还有其它一些高效的模式。然后就发现这几种不同的模式，其实都很难成为组织上的根本创新。

举个很简单的例子，就是（你）很难用一种规划的方式去创新大模型。如果是在移动互联网时代，我可以规划我接下来要做哪些需求，然后这些需求只要一旦被定义出来，是可以被确定性地生产出来的，基本上很少存在今天这种 APP 突然不知道怎么开发，或者我这个定义了需求，结果没有被实现的情形。因为它是一个确定性的事件，只需要经过人在计算机上用编码让计算机理解，就可以被实现。

但是 AGI 是不一样的，我很难去规划，今天要去完成一个什么样的需求、然后完成到什么程度，因为它是没有办法被硬编码、被规则表示的。AGI 做事情的方式不是这种前置的规划性的创新，而是后置性的，我可能要去试一下才知道。AGI 得有一个底层的机器，它在一个更系统的方式下做很多东西。

这两者是一个 fundamental 的区别。因为你的组织要跟你做事的方式相匹配，当你做事情的底层逻辑发生了变化的时候，就需要新的组织形式。在互联网时代产生了很多非常好的组织，他们可能在某些领域比如跟推荐系统相关的产品都非常擅长，但是有可能在新的时代，会有一些非常擅长 AGI 的组织出现。

我觉得这个是大概率会发生的事情。

张鹏：OpenAI 在你眼里是一个比较好的 AGI 的组织形态的样板吗？它有什么是你觉得对的？有哪些可能也未必是最好的？

杨植麟：首先从结果上来讲，OpenAI 肯定做出了非常大的突破。如果没有这个公司，我觉得可能人类的进程都会不一样。

如果再深入下讨论的话，一个好的组织首先需要很高的人才密度，然后所有人都应该有一个共同的 vision（愿景），能够很高效地聚焦围绕一个目标去做事。我觉得这些点是他们做得非常好的。

但这里面最核心的一个点，也是从外界可能不那么容易看到的一个点，就是当你有了这些前提之后，你怎么样找到一个系统性的方式去做事情。我觉得这个点是所有技术的一个前提条件，也是我们现在可能最想去迭代做得更好的一个点。

张鹏：你说的系统性方式是指它可以被复制、被放大吗？

杨植麟：对。但这里的复制指的是你能把它用在做不同的事情上，它不一定能够复制到别的地方。因为这是你在一个公司里面形成，要复制到别的地方可能很难。但是一个公司可以反复地去利用这个系统去做不同的事情。比如说我今天可以用它去克服长文本的挑战，明天可以去做一个自主的 AI 能力，后天我还能用它去攻坚多模态……

这应该是一个可复用的一个系统，本质上它会沉淀下来，成为你的核心资产。我觉得这应该是每一个 AGI 公司最需要花时间去打磨的一个东西。

Google 的组织涌现出 Transformer，OpenAI 的组织涌现出 ChatGPT

张鹏：以前大家聊 OpenAI 和组织，面对不确定的创新，好像经常会二选一，是 bottom-up 还是 top-down？你在构想这套创新的组织时，是怎么去定义的？

杨植麟：我觉得 top-down 这个大的框架肯定还是适用的，特别是对于大模型来说，有一个 top down 的框架是非常重要的。

Top-down 讲究的其实是 leadership 的 vision，就是你能不能判断什么是对的、要做的事情，然后什么可能是你现在不要做的事情。

AGI 就像登月工程，它是一个需要长时间、很多个人互相耦合的一个巨大的系统，所以这种 top-down 的设计我觉得肯定是非常必要的。

然后在这个框架下面，重要的就是系统能不能把这个事情做好。组织之中会有很多很小的单元，每个单元能在做不同的事情。如果你有一个好的系统，让每个单元都可以很高效地去产出一些东西。最后有一个 top-down 的框架，把这些东西去整合起来。

张鹏：其实某种程度上我感觉你是在一个大框架下，在一个组织里，模拟不同的方向上能否涌现合适的创新，通向那个 AGI 的目标。今天我们无法精准定义哪一条路，因为里边还有大量不确定的东西，所以它是需要一些涌现的。所以这个时候需要组织支持这种涌现，对吧？

杨植麟：是的，我觉得这个是非常好的理解。

比如我们可以去看 Transformer 是怎么产生出来。它本质上是 Google 给这帮人提供了一个涌现的环境。在 Transformer 出现之前，已经存在像注意力机制、残差网络、LayerNome 这样的技术，有 SGD 这些训练的基础配套，然后有 learning rate schedule，就是所有的东西都提前准备好了。

然后这个时候 Google 提供了这样的一个环境，能够让这些人其中自由地去组合，突然涌现就产生了。

但是不同的环境能涌现出来的东西不一样，Google 的环境只能涌现出来一个科学的结果，但是它没有办法涌现出来一个伟大的系统工程——也就是 ChatGPT，这样一个把东西做到极致的同时又在产品上能够精准抓住需求的跨时代作品。因为 Google 的组织跟 ChatGPT 就是不配对的。

OpenAI 其实没有发明任何新的东西，但是它涌现出来了一个工业化的杰作。它结合的东西跟谷歌就不是在同一个维度了。

OpenAI 结合了什么？

1、Transformer 的架构。

2、计算中心，能支持 10 的 25 次方浮点数运算。

3、整个互联网积累了 20 年的数据，这个可能是互联网最大的价值。

所以 OpenAI 看到的是这三个因素，然后它提供了一个环境，使得这三个因素能够被组合起来，那它就涌现出来了一个一个 AGI 的里程碑。

这就是我刚刚讲的，不同的组织会允许涌现出来不同的东西，但你想让它涌现出来什么，你就应该把这个组织往什么方向去调整。

03

AGI 的技术路径已经确定，但产品方向仍然有很多未知

张鹏：上半年有一个流行的说法叫「明牌重注」，意思是 AGI 的技术路径已经是确定的了，接下来只要比拼资源的投入了。你会怎么看？因为从你对组织的看法来看，它还有很多的不确定性。

杨植麟：现在 AGI 的第一性原则是清楚的，就是只要能把无损压缩继续做得更好，就能产生更高程度的甚至超越人类的智能，这个本身是已经有大量的证据证明了。

就像力学三定律之于经典力学一样，大方向已经是基本确定的。

所以「明牌重注」的观点，我认为有一定的合理性。剩下的就是去推演第二个层次的东西，在大的原则下面有一些具体的事情到底应该怎么做。比如说怎么样做一个真正能无损压缩的长的上下文？这个问题它可能就没有那么简单。

我觉得即使是 OpenAI，它也只是做了第一步。后面每一步到底应该怎么做，仍然是存在着一些不确定性。

然后就是技术层面之外，在产品层面上的思考。今天我们离很多科幻电影里展现出来的超级聪明的 AI 其实还有蛮大的差距，并且现在的产品也未必是在往正确的方向发展。

每个时代都会有最伟大的人和次伟大的一些人。最伟大的人去发现正确的第一性原则，但是也会有一批同样很伟大的人。他们可能没有第一个人那么伟大，但是这一批人解决了很多技术的挑战，产品的挑战，甚至商业的挑战。

所以在第二个层面，我能想象它未来肯定是一个巨大的空间，是一个星辰大海。但是这里面具体怎么去玩，我觉得其实还是有挺多的未知和挑战。

Transformer 是新的计算机，上下文长度就是「内存」

张鹏：长文本是月之暗面模型的一个专长。因为像你所说的，其实有非常多的创新方向，那为什么选择了长文本？

杨植麟：首先我觉得其实每个人都应该去问一下自己，你希望这个 AI 未来能帮你做什么事情，或者说人与 AI 应该是一个什么样的关系。然后会发现，在一个很终极的形态下，它相比于今天欠缺了一个很大的能力，就是拥有一个更长的输入窗口。

长的输入窗口和短的输入窗口之间的区别，其实要比我之前想象的更加本质。我认为 AI 一个很终极的形态就是能够跟人建立长期的情绪价值，就是每一个人都可以有一个几乎无限长时间的终身陪伴。

因为时间是一个很重要的维度，只有时间长了之后，你所有的信任、更复杂的情感，以及能够横跨几十年的交互它才会展现出来它的力量，然后这个时候它才有可能给人提供很深刻的精神上的价值。这种情况肯定不能是一个需要每天重启上下文窗口的 AI 可以满足的。

张鹏：就是它明天就忘了你今天干的事，对吧？

杨植麟：对，所以本质上来看，如果说 Transformer 是一个新的计算机，它有两个最重要的维度。一个维度就是参数的数量，决定计算的复杂度，有点像旧式计算机里面的 CPU。另一个维度是上下文长度，它其实是这个新计算机的内存，这个内存决定了你有多少东西能参与计算。

所以在计算足够复杂的情况下，内存越大，能解锁的应用空间就会越大，所以它其实是一个非常本质的东西。如果我们去看过去几十年计算机系统的发展，40、50 年前所有人都觉得 500K 的内存可能就够了，但今天看起来这显然是一个谬论。所以我觉得一样的事情会发生在这个新的计算机系统，而「长文本」作为「新计算机」的「内存」，绝对是一个非常非常重要的东西。

04

闭源路线是为了打造 AGI 时代的 Super App

张鹏：这一波大模型创业里，我们能看到不少开源模型，开源模型也是体现团队能力和生态构建的一部分。Moonshot 是一个闭源模型，而且最近应该也没有开源的计划，想知道你们背后对这件事的思考？

杨植麟：我们是非常支持开源的。

我认为开源和闭源接下来在大模型领域里会是互补的关系，开源可以支持开发者去尝试各种创新的应用，而且在开发过程中可以对数据、训练过程、环境部署等合规性有更高的要求，场景也会更灵活。

而闭源的话，也会有自己的价值，比如说像未来的很多超级应用的入口，不管是生产力端还是娱乐消费端，都会有以闭源为核心的超级应用出现。这两种不同的模型其实是一定程度的互补，而不是冲突的关系，如何取舍其实是看每个公司不同的策略。

我们的策略是希望去打造超级应用，这是我们目前专注的地方，所以会把时间都花在上面。

以终为始，ToC 方向匹配 AGI 的终极目标

张鹏：听起来你们接下来不是要做 api 或者帮助企业训练自己的大模型落地到他们的业务里，而是要做 toC 的业务。其实在上一波 AI 浪潮里我们可以看到，没有在 toC 领域里有太多的突破，

基本上做的还是 toB 的事情。为什么这一波 AI 里，你们会坚定选择 toC？毕竟，toB 的事情看起来还是至少比较确定能带来收入的。

杨植麟：ToB 我们也不是说完全不做，但主要聚焦和发力的还是 C 端。

对我们来说，这是一个新的技术变革产生的新机会，因为在过去很长一段时间，AI 的技术在 toC 领域还没有任何成功案例。

但我觉得随着新的技术变量的出现，很多 AI 技术可以实现更好的效果，这些更好的效果就可以帮助我们做到之前没有办法做的事情。这些事情可以用新的应用、新的入口的方式呈现出来，收入呈现指数的增长，用户量也在快速增长。

不管是 Midjourney、Character AI 还是 ChatGPT，都在很大程度上证明了 AI Native 的 app 是完全有机会的。

还有就是，既然做 AGI，就要选择与之能匹配的业务模式——强调极高的创新效率，我觉得也只有 toC 的模式才能去完成快速闭环，组织才有可能形成一种快速迭代的文化。以日为单位去更新模型、调整组织和满足用户的诉求，所有东西以数据为核心快速运转。只有在 C 端才有可能产生这样的能量，才有可能与以 AGI 为目标的公司匹配。

这也是我们以始为终地去思考这个事情，我们认为，与 C 端用户共创也是在做 AGI，这可能本身也是一个必要的前提。AGI 不能闭门造车，这里面核心的一个点是数据，如果不跟用户共创，很难有足够高质量的数据，就没办法知道模型真正被用起来之后会产生什么问题，很难跟用户一起去在很多场景里做更深入的挖掘和优化。我甚至觉得，这在很大程度上也是一个必要的前提条件。

张鹏：所以这件事又回到了对于目标的第一性上。如果不去做一个 C 端的 super app，有足够多的用户和数据，其实最终很难去真正实现通向 AGI 的目标。也就是说，不想做 super app 的公司其实不算是 AGI 的信徒。如果你真的要推 AGI 的话，可能需要去做这样一款 toC 的 app，而不是在 toB 领域去解决一些确定性、工程性的问题。

大模型技术让新的 Super App 成为了可能

张鹏：在你看来，Super app 的定义是什么？比如大家会认为微信肯定是 super app，因为用户可以在上面做很多事情，淘宝应该也是一个 super app，因为它有非常大的用户量，能产生很大的价值。

在瞄准 AGI 目标的时候，我们应该如何定义 super app？是应用创造了很大的价值，还是应用背后的思路和以前的定义不一样。

杨植麟：从定义上来说，倒没有特别创新的地方。就是可能实现的价值不太一样，因为能提供以前提供不了的价值，才会有新的入口出现。但最终肯定是要有很多用户、以很高的频率在用，并且在使用的过程中产生很大的价值。另外就是，AGI 本身有一个很好的属性，使得 super app 是有可能的。那就是 AGI 的 G——general，通用人工智能，可以不只是解决一类问题，而是能解决的问题越来越多。

当能解决的问题越来越多的时候，产品边界一直是在拓展的。AI 会逐渐深入到生活的各个方面，应用的价值也会越来越强，也就更符合我们对于 super app 的指标和定义。AGI 的通用性和能成为 super app，这两者是兼容的。

张鹏：我觉得你说了一个非常关键的点。移动互联网时代，我们一般看到的是，一个应用先达到某个规模，然后才去不断拓展服务的边界，逐渐成为通用的应用。但是在新的技术范式下，

如果要做一款 super app，是因为先具备了通用的生产力引擎，然后自然而然变成了 super app。这是两个时代引擎驱动的基因的不同，上一个时代大家是跑马圈地，赶紧把规模做起来。现在是因为有技术的引擎作为驱动，天生就具有 super app 的基因。

杨植麟：总结得特别好，补充一点就是，即使技术上很通用，肯定也是要从一部分的场景开始，然后去不断的泛化，而且泛化的速度可能是指数级而不是线性的。

张鹏：今天很流行聊 AI Native 的概念，但是好像没有特别精准的定义。以前我们开发产品，一般是有明确的目标下，产品经理、前端、后端彼此配合，按照周期去迭代交付，观察用户数据进行 A/B Test，找到最好的路径。

但今天站在 AGI 的视角，做 super app 的开发，开发范式到底应该是怎么样的？还会是原来的开发形态吗？

杨植麟：产品开发方式会随着底层技术的变化而变化。

移动互联网时代的开发，是有了明确的需求，对应确定的操作和完全确定性的事件。背后对应的是旧式计算机的技术和确定性的编码。这些非常确定的逻辑运算，衍生出各种前端确定性的交互。本质上是基于确定性的 Graphic UI，加上确定性的系统，这是过去二十年看到的互联网产品的开发范式。

但是在今天，技术范式发生了很大的改变。首先是前端变成了对话式的 Conversation UI，未来可能会有越来越多的产品采用这种 UI，后端也被极大程度的统一了，统一到了一个「语言模型」上。这个模型处理的不光是语言，它能处理世界上所有的信息，本质上是对世界上所有信息进行编码和无损压缩。

这两个都确定之后，大部分应用层的产品开发其实都不涉及后端的计算构架或者前端的 Lauguage UI，可能会有一些 GUI 和 LUI 的结合，但整体的构架是被基本确定下来了。

今天所说的大部分的开发，实际上是做中间层的事情，就是数据。交互和模型可能会一样，但用不同的数据，就会出来不同的产品。比如可能是 ChatGPT、Github Copilot 或者 Midjourney，本质上其实没有什么不同，主要是定义的数据不同。

所以我觉得这是极大的范式创新，产品经理越来越多需要想的事情是怎么通过两个数据集去开发一款产品，定义好了数据集，其实产品就定义完成了。一个是训练数据，一个是测试数据，训练数据决定了模型能提供什么能力，测试数据决定了模型的实际可用程度。

以前没有 AI Native 的产品，只有 AI feature，所以这种开发范式还不是很流行，很多强产品 sense 的人，也不一定知道怎么去套这种东西。

但是现在有越来越多的产品需要 AI Native，比如今天想开发一个跟 Character AI 一样的产品，或者在上面做优化，那你就要考虑怎么做优化。如何定义你的两个数据集，可能需要你有很好的数据的生产和处理技术，如何获取数据，以及什么样的数据是有效的等等。

我们需要在不断的探索过程中，把这些流程和开发范式具象化，AGI 的新的开发方式，可能需要一个新的组织形式才有可能做到。

张鹏：说白了，在新的开发范式下去开发产品，需要考虑的是怎么让模型和设定的目标是匹配的，而且要知道怎么训练模型和调试模型。

开源和闭源并不冲突，会长期存在

张鹏：现在大家很多人说开源挺好的，长期来看，开源的技术也会水涨船高，Llama 2 现在可能跟 OpenAI 有差距，但本身基础素质也不错，未来也会有持续的发展。

你现在要做 Super-App，想要运用大模型作为引擎，那我买一个引擎，改一改，是不是也行？未必自己要有一个引擎的专利，一个引擎的团队。我发现你在模型层面是要较劲的，为什么你会认为，必须端到端地做这件事？

杨植麟：基于开源做应用，肯定也有很大的机会，两者（开源和闭源）我觉得并不太冲突。

如果最后是一个超级入口、超级应用，我觉得大概率还是闭源，因为你可以通过闭源去形成产品的差异化，可以从制造模型的第一天，当你规划它长期演进方向的时候，你就有绝对的空间，让你的 App 能够产生非常大的差异化优势。

开源模型做应用，也许不是一个超级入口，但我觉得也有机会做很多增量的价值，更多是产品化的价值，专门的数据，通过微调产生一些人无我有的增量的东西，这也是成立的。

这两者我觉得并不冲突，在生态里面两种路径都会存在。

张鹏：你有一个特别确定的要解决的问题，这个问题用今天的技术也能工程化地解决好，也可以创造价值。如果你想要追求 super app，通向 AGI，那么你的模型就需要跟随应用共同成长，你的数据集、测试集要不断变化，引擎要不断变化，生命力得掌握在自己手里。

杨植麟：没错。而且，很多模型的基础能力，也需要跟市面上的 commodity（行活）有差距，现阶段还处于技术驱动的阶段，通过更好的基座模型，可以转化成产品优势。

但最终肯定不是，比如再过 10 年 20 年，技术上会陷入一种 commoditized（行活化）的情况，那你可能就需要利用先发优势，把壁垒转化为更可持续的壁垒，比如更强的网络效应。

05

新时代产品经理需要具备的素质：快速迭代

张鹏：产品经理很有意思。移动互联网早期真的是靠一波产品经理，想象、定义未来的场景，推动产品落地。之前我跟张小龙在视频号聊过，他开玩笑说，他是古典产品经理了。后来的产品经理，14、15 年之后，越来越多的数据（驱动）型，做 A/B test，但在最早期、蛮荒期是需要一些想象力的。我常说这是一种「神性」，「我对这个事是怎么定义的」，他们可能没有一个完整的逻辑能给所有人都讲通，但是这个想象、设定可能就就是对的。

这种不可知，有点像大模型，不可解释，给你一个结论，是直觉，这个直觉也来自人生经历等等，也是一种无法被反向解释的模型。直到后来就开始变得很科学，赛道确定，做事方法都确定了。

现在你要做 Super App，你肯定很关注产品经理，对于产品经理的要求，现在是要「神性」多一些，还是科学性多一些？

杨植麟：本质上，会在「神性」和系统性之间寻找一个平衡。我个人觉得，长期来看，系统可能会以碾压的方式成为主流的开发范式。但并不是说「神性」就不重要了，只是它需要一个很好的系统作为支撑，或者说，系统应该是主力军。

我经常说一个比喻。张小龙在一张巨大的地图上指了一个点，说在这种一棵树，后来证明他是「神」，因为他指的地方是对的，这棵树长成了巨大的森林。这就是「神性」的体现。一个神枪手，指哪打哪，判断非常准。

但是 AGI 不是这样工作的，比如你想做一个 ChatGPT。引用一个很有名的设计师说过的话，不是根据设计来完成制造，而是通过制造完成设计。（注：日本工业设计大师柳宗理，喜欢先手工制作模型再画设计草图。）

东西做完了，就设计完了，而不是先设计好了，再去做它。设计好了之后再去做，有点像以前的种树，找地方找半天，种一棵树，「我靠成了！」有点这个意思。

现在，不用那么细，大概看一下，「诶，这块地不错」，有个「神枪手」划一块地，AGI 是你的主力军，直接开过去整块推品，这里面有什么机会，哪些地方能种树，都能找出来。

这个过程需要一个系统。有了强大的系统以后，你会发现让很多天赋型、运气型的（产品经理）判断在哪里种树，就像今天几千万、几亿的场景里，选哪些场景比较好？它本身是极其低效的。

在 AGI 时代寻找 PMF 的过程，就应该用 AGI 的方式做，发挥它通用的价值，发挥用户生态的力量，发挥系统的力量。你应该一口气全推过去。我觉得这可能就是与古典的产品经理的做法最大的区别。

张鹏：最开始的时候，直觉更多用在对于问题的定义、目标的选择上，而真正在这个目标上，怎么能更快更好地实现，更多是靠系统。

看增量不看存量，可能是下一代产品经理的特质

张鹏：更具象地说，你招了不少产品人，团队有很多年轻人，他们身上有比较统一的特质吗？是不是要大厂？是不是都要做过产品？从已经招来的人身上反向总结的特质，能不能给我们透露透露？

杨植麟：我觉得很重要的，一个是开放的心态，二是学习的能力。它指向的一点就是：一个人能不能快速地迭代。这是我们觉得价值非常大的潜质。

不光是产品经理，每一个角色，甚至每一个人，不管做不做 AI，（这个点）都非常关键，因为现在变化太快，基本上你很难预测明年底，AI 能给我们提供什么，我觉得基本不太可能预测。

一年都很难预测，更不要说接下来三五年、五到十年。所以我觉得需要我们每一个人都用非常开放的心态，快速地学习，然后具体地去做。

具体到产品上，我觉得需要很强的 C 端 sense。你可能对它很感兴趣，但是到底怎样是有效的方式，能够让产品持续演进下去？做出第一版的产品，可能是容易的，因为你还没有经历让它持续迭代变好的过程，或者说去更加精细化地定义你想要一个怎样的产品的过程。它远远没有之前的产品那么直接、简单。

因为，你可能有一个产品，然后你怎么让它变得更好？「好」是非常抽象的。你要怎么让 ChatGPT 变得更好？怎样算好？往哪个方向好？好多少算是好？这些都很难（定义）。

很多产品经理容易陷入一个误区，定义一堆 feature 功能，像以前一样，这可能是不对的。因为（现在）你的功能是通过数据定义出来的，这才是 AI Native 的方式。

所以，需要很多的学习，它不光是静态的理论，而是学习之后要去试。我今天讲的东西可能是错的，没事，你去试，试完发现，好像又收获了一些，自己又迭代了一个梯度，在这个过程中，逐渐加深对它的理解。我觉得在这个过程中会出现我们这个时代的张小龙。

张鹏：不一定是这个时代的张小龙。龙哥是那个时代的经典，下一个时代可能会有一个全新的人。这个时代里，一定会出现这个时代的新的产品之神，它的「神性」体现可能是不一样的，这才是时代进步最让人期待的东西。

你刚刚说的，这个时代的人才，我们很难定义说，今天你身上一定要具备怎样的特质，但有一点是确定的，看增量。也许不看你的历史，但看你的历史和今天之间，今天和明天之间，你的增量如果足够大，在一个新的、不确定的时代里，就会具有很大的意义。看增量不看存量，这可能是下一代产品经理（的特质）。

杨植麟：对，很多时候没有太多历史包袱，是一件好事。

张鹏：我看刚才已经有人在弹幕问，「还招人吗？」「还招实习生吗？」你们现在还招人吗？

杨植麟：对，全职、实习现在都有。

张鹏：你们对人的挑选有什么标准？对人的要求也要按照新范式来？

杨植麟：我们相对来说比较 open，不同背景（都可以）。我觉得 AGI 是个很综合的事情，今年市场上有一些热度，会吸引各种背景的人，这个很重要，因为如果只有单一背景，很难做好，市场上各种人才流动是很重要的。

举个例子，现在所谓的 AGI 技术，背后其实有 NLP（自然语言处理）的部分，有 Computer Vision（计算机视觉），有 RL（强化学习），有做对齐的，还要有很好的基础设施，要有写 Kernel 的，这是一个非常全栈的东西，光是技术就要很综合，更别说技术以外，产品、运营、商业化这些，每个职能都需要，理想中应该是很多元的背景。但你的 Vision 应该是一样的，我们很欢迎对 AGI、对 Super-App、对全球市场有激情的小伙伴。

06

AGI 时代重要的指标是场景摩尔定律

张鹏：Sam Altman 曾经提过，智能摩尔定律，他也写过文章阐述万物摩尔定律，智能的成本和能力之间，存在摩尔定律的关系，我不知道你是否认同他的这个观点？

杨植麟：是的，我觉得摩尔定律也会出现一个范式上的变化。以前摩尔定律，每 n 个月晶体管数量翻倍，到后来模型的参数和算力符合摩尔定律，每 n 个月 flops 翻倍。

现在对我们来说，智能摩尔定律本质上反映的是每 n 个月，可用的 use case 的数量，会翻一倍。它本质上是 scaling law 的延伸，更多反应在模型预训练阶段，它到底符合怎样的规律。加入更多的算力、更多的数据、更多的参数后，模型的 training loss 会发生怎样的变化，这是之前标准的 scaling law 的说法。

但最终，其实最重要的指标，是场景的摩尔定律，有多少场景达到可用？它必须是指数上升的过程，不能是线性的，每 n 个月翻一倍，就是指数上升的，不能再用传统 AI 的方式，每次加一个场景，每次加一个数据，让它在这个场景上 work，那样的话就永远无法用指数的办法上升。

我觉得这个点是很关键的，衡量有多少场景被解锁，有了这个，PMF 的寻找过程就会极大地被加速，同时可以去试很多东西。就是，不应该种一棵树，而是划一片地，有一片地之后，有场景摩尔定律，就能一下全部试一遍，这是一个超级快速的种树机器，很快，不用种树它就知道，在这里种树是死是活。

当你（场景）多到一定程度的时候，就会成为一个超级入口。

张鹏：智能摩尔定律和场景摩尔定律应该是一个双螺旋，互相有关系，成本不断下降，解锁的能力越来越高，能力越高，成本越下降，场景理论上就会越多。

下个时代最伟大的公司会是两种文化的结合

张鹏：你对硅谷应该很熟悉，在 Meta 和 Google 都工作过，你怎么看硅谷的文化，硅谷工程师的能力的特点？对比中国这边未来创新者的文化和能力，你怎么理解两者之间的差异，各自擅长什么？

杨植麟：硅谷工程师的文化非常典型。比如 Noam（Shazeer，Character.AI 创始人），他们今天做了这样一个有一定程度 PMF 的产品，但是他们早期其实没有太多专门的产品经理的角色，这是很重要的工程师文化的体现，这些工程师他们可能有很多自己的想法，把技术和这些想法结合到一起，尝试向前走一步。

我觉得这其实是我们很多时候可以借鉴的一个东西，特别是在 AI Native 的新范式下，很多时候需要全民和技术「双向奔赴」，需要这种工程师文化，去把技术和需求连接在一起。比如刚刚说那两个数据集怎么构建，如果没有这种工程师文化，不往前去走一步的话，很多事情就很难做了。

所以在这个点上，如何更加 AI Native，本质上是这种工程师文化提供的很大的驱动力。

然后我觉得，底层我们还希望借鉴另一个东西，东西方文化里优质的地方都可以去吸收。

比如 OpenAI 有很强的技术理想主义，「我想做 AGI」，但也没想好商业模式怎样，一开始就有很多投资。这种技术理想主义的驱动下，包括最近很火的有效加速主义（Effective accelerationism，e/acc），我觉得这都是技术理想主义的体现。包括硅谷以前的公司，Google、微软，都是在一定的技术理想主义的驱动下产生的。

东方文化里有很多东西，强调「有用」，有商业模式的前提下考虑事情。

我觉得接下来 10 年，在 AGI 的时代里，最伟大的公司可能要结合这两种文化。一方面是能够从功利角度去找到非常好的商业模式，它是你持续燃烧的燃料；另一方面也需要技术理想主义的驱动，不光是为了赚钱或者有用，而是本身就有很强的理想，去看看，月球背面到底什么样。

张鹏：感谢植麟，我非常喜欢你思考问题的角度，尤其你说的最后一点，我们以前善于目标导向，通向有用，但未来把一件事变得有用、普惠的过程中，可能需要一点 moonshot 的精神，你在瞄向一个高位置的东西，不管打不打得中，至少要往宇宙深处走，往星河深处走，我觉得这是让人兴奋的，往往可能是一些兴奋的目标，聚集起了真正优秀的人。

我也很期待月之暗面用这种对组织、对创新的新范式的思考，接下来能够有所成就。

Kimi Chat 公布“大海捞针”长文本压测结果，也搞清楚了这项测试的精髓

https://mp.weixin.qq.com/s/IC5-FGLVHzHHYqH6x-aNng

时光荏苒，岁月如梭。在过去的两个月里，“长文本”成为基础大模型的一个关键突破方向：

10 月上旬，Moonshot AI 带着首个支持 20 万汉字输入的智能助手产品 Kimi Chat 问世；

11 月上旬，OpenAI 发布支持 128K 上下文窗口的 GPT-4 Turbo 模型；

11 月下旬，Anthropic 发布支持 200K 上下文窗口的 Claude 2.1 模型。

从绝对数字上看，全球主流基础大模型的上下文窗口长度都得到了大幅升级。

但考虑到大模型过去曾经存在的“迷失在中间”现象——就是在回答问题时会忽略一段较长文本的中间部分细节内容，人们迫切地想知道如今大模型在支持更长的文本之后，性能究竟如何，“老毛病”还会不会出现。
一位 AI 大模型领域的开发者 Greg Kamradt 也对此充满了困惑。于是，他设计了一个名为“大海捞针”的大模型长文本性能测试方法：
在文本语料中藏入一个与文本语料不相关的句子（可以想象是在整本《西游记》里放入一句只会在《红楼梦》里出现的话），然后看大模型能不能通过自然语言提问的方式（Prompt）把这句话准确地提取出来。
Greg Kamradt 的“大海捞针”实验简述：
“大海”：Paul Graham 的文章合集作为语料
“针”：“The best thing to do in San Francisco is eat a sandwich and sit in Dolores Park on a sunny day.”
提问："What is the most fun thing to do in San Francisco based on my context? Don't give information outside the document"
期待模型输出的正确答案：
The best thing to do in San Francisco is eat a sandwich and sit in Dolores Park on a sunny day.
Greg Kamradt 把藏起来的那句话（也就是大海捞针的“针”）分别放到了文本语料（也就是大海捞针的“大海”）从前到后的 15 处不同位置，然后针对从 1K 到 128K（200K） 等量分布的 15 种不同长度的语料进行了 225 次（15×15）实验。为了保证实验的严谨，他还重复进行了实验，来取平均值，测试时调用 API 的费用总计花了几百美元。
Greg Kamradt 在 Twitter（现名 X）上公布了他在 GPT-4 Turbo 上测得的结果：
GPT-4 Turbo（128K）在语料长度超过 72K 且句子（“针”）藏在文本头部的时候，准确率不佳。
Image

然后又用完全一样的实验内容，测试了 Claude 2.1（200K）模型的结果：
有些惨不忍睹， Claude 2.1 似乎在语料长度超过 20K 之后就开始准确率不佳，而且句子（“针”）藏在语料靠前的位置时，准确率尤其差。
Image
来源: https://twitter.com/GregKamradt/status/1727018183608193393
第一次实验

作为大模型长文本技术的重要玩家，Greg Kamradt 的实验成功引起了我们的注意。由于 Greg Kamradt 将他的实验所有语料和代码都放在了 Github 上，Moonshot AI 的工程师很容易就可以在 Kimi Chat 上重复做这个实验。
在条件完全一致的情况下，Kimi Chat 在“大海捞针”实验中的测试结果是这样的：
Image
全绿！
这个结果让 Moonshot AI 的工程师感到有些意外，虽然想到了结果会不错，但没想到能这么好。
好奇的工程师又试了 Kimi Chat 的一个更新的测试版本（不同版本使用了不同的对齐训练数据，但基础能力大致相同），结果反而没有这么好。
鉴于基础能力几乎相同两个版本的大模型，在“大海捞针”实验中表现却有明显的区别……
于是，工程师们想到了，不同的结果很可能是 Prompt 不够适配造成的。
第二次实验

因此，Moonshot AI 的工程师决定展开第二轮实验。将原始实验中的 Prompt 从
“What is the most fun thing to do in San Francisco based on my context? Don't give information outside the document”
调整为
“What is the most fun thing to do in San Francisco based on above document? Don't give information outside the document”。
根据经验，在基于文档的问答场景下，新的 Prompt 表意更加明确。
在第二次实验中，Kimi Chat 虽然没有得到“全绿”那么惊艳的结果，但修改 Prompt 之后错误情况的波动也在我们的预期之内。
Image
我们同样测试了在新的 Prompt 下，GPT-4 Turbo 表现：

Image
以及 Claude 2.1 的表现：
Image
跟原始实验结果相比，GPT-4 Turbo 的表现更好了，出错更少，而且出错的位置也发生了变化：之前是集中在长文本的前半部分，现在分散在“针”位于语料头部的时候，在较短的语料和较长的语料中都有分布；
Claude 2.1 比之前的“惨不忍睹”也稍微更好了一些，出现了很多接近正确（正确率 70%）的色块——没有按标准答案输出，但也提及了“针”的内容。
经过两次实验，Moonshot AI 的工程师初步判断：
“大海捞针”的长文本性能测试很有启发性，但是不同 Prompt 对实验结果也有较大影响，而且模型的错误没有表现出较强的一致性，需要进一步测试以及仔细分析 bad case 才能充分了解这项测试的参考价值和局限性。
于是 Moonshot AI 的工程师决定设计一个中文版的“大海捞针”实验，做进一步的测试和研究。
第三次实验

我们将 Greg Kamradt 的“大海捞针”英文实验语料换成了 AI 科技媒体《量子位》的文章集合，原实验中藏进去的句子换成了：
“在北京最好的事情就是秋天里坐在五道营胡同里的铁手咖啡馆，喝一杯热美式”。
向大模型提问的句子换成了：
“北京最值得体验的活动是什么？仅基于上述文档，不要给出上述文档以外的信息。”
然后分别跑了 Kimi Chat、GPT-4 Turbo 和 Claude 2.1 的结果。
Kimi Chat：
Image
GPT-4 Turbo：
Image
Claude 2.1：
Image
Kimi Chat 出现三处错误，其他地方全绿；GPT-4 Turbo 则“红”了一大片；Claude 2.1 继续出现了很多 70% 正确率的情况，其实也算把“针”找出来了，只是没有按照预期的标准答案形式输出。
结果有些出人意料。为了进一步了解“大海捞针”实验的局限性，Moonshot AI 的工程师去研究了一下大模型没答对的所有情况（Bad Case）。经过一番仔细分析，我们发现“错误”通常是由两方面的原因造成的：
第一，由于提问的 Prompt（“北京最值得体验的活动是什么？仅基于上述文档，不要给出上述文档以外的信息”）对大模型而言有一定的歧义，大模型有时候会“咬文嚼字”，坚决认为喝咖啡不是一种“活动”。
第二，由于大模型的传统“技能”——幻觉 hallucination 造成的，模型直接编造了一个不相关的答案。
第四次实验

于是，Moonshot AI 的工程师决定再次改进中文的“大海捞针”实验，将实验中藏进去的句子换成了：
“月之暗面科技有限公司北京地址是海淀区知春路量子芯座”。向大模型提问的句子换成了：
“月之暗面科技有限公司北京地址是哪里？仅基于上述文档，不要给出上述文档以外的信息。”
这下应该没有什么二义性了。
第四次实验，也是第二次中文实验，跑出来的结果如下。
Kimi Chat：
Image
GPT-4 Turbo：
Image
Claude 2.1：
Image
Kimi Chat 表现优异，只出现了一个错误；GPT-4 Turbo 的错误，主要表现在当“针”藏在语料前半部分的时候；Claude 2.1 则有些惨不忍睹，工程师分析了 bad case，发现最主要的原因是 Claude

2.1 对中文“地址”的认知不太准确，不认为“针”的内容是在讲一家公司的地址，所以没有提取出来……

最后的话

经过四次“大海捞针”实验，Moonshot AI 的工程师基本上摸清楚了这项实验的关键点和局限性，除了大模型本身的长文本记忆能力和指令遵循能力，其实还有两个关键点对结果起了明显作用：

第一，藏在大海中的“针”是否完全没有歧义；

第二，向大模型提问的 Prompt 写的是否足够明确。

感谢 Greg Kamradt 设计了这个大模型长文本性能压测实验，让我们可以直观地了解几个基础大模型的长文本能力特点。不过，就像所有的考试一样，成绩只是一种反馈，并不能全面定义我们对技术的理解和追求。

当前对整个行业而言，大模型性能的客观、准确、全面评估，仍然是一件很有挑战的事情。如果你想到了一种新的测试方法，不妨联系 Moonshot AI 交流探讨。

当然，我们始终把用户的体验放在最重要的位置。如果你还没有体验过支持 20 万汉字上下文的 Kimi Chat（现已全面开放），欢迎扫码试试！

招聘信息

MoonshotAI（月之暗面）的招聘联系邮箱：hr@moonshot.ai
如果关于 Kimi Chat 产品本身有疑问，可以联系：support@msh.team