

今天北京风特别大 感谢文风今天还跑过来录着祭博客 然后正如北京的大风

一个非常尴尬的转场 今天AI最热闹的大风天就是A针头了

所以文风正好做了满长时间A针头 跟大家介绍一下自己

大家好 我是王文芬

是一个做A针头两年的一个程序人出人的串业者

同时也是缺老师的粉丝

然后我们再做了一个产品是DataEgent

现在大家都在讲各种各样的A针头

那A针头到底怎么定义

我觉得最好的定义是ASROPE的一个定义

ASROPE的定义的话是说A针的是模型及环境反馈去使用兔的一个程序

那这里面的话其实我觉得有三个点需要去拆开说下

第一点是模型

这个大家其实是最容易理解的

第二个的话我觉得是环境反馈 这很关键

第三个的话就是兔

就说A针头我们需要去Tekaxon

其实通过调用兔的方式去实现

所以你怎么看最近这波A针头热

我觉得这波A针头跟过去非常不一样的一个点

是这一次A针头真正的是能够去实际的解决问题了

像上一轮A针头大家讨论的时候

其实是在23年的4月份当时代表新的产品是RGBT

但是当时它更多的可能是一个板剧

它并没有什么实际的解决的一些问题

但是经过两个发展A针头真的一种落地了

就在实际的工作生活场里面给大家带有真正的价值

这两年主要的变化就是什么

我看到的有两点

第一点的话确实是底层的模型有了非常大的一个进步

尤其是OETRA之后给A针头带了常词未能力

主要就是因为RR0真正决习在决作上的使用吧

对 我觉得这个是以部分吧

还有另外一部分可能是大家没有关注到的一个点

就是说其实A针头工程这个事情上面

也有这样大的突破

就是说是怎么去给A针头更好去购件一个合适的Kontax

然后去解决问题

这个我觉得是从工程这一次或者说产品这一次的一个进展

所以你刚说第一点是模型的进展

对 第一点是模型第二点是A针头的Kontax

Kontax这个怎么理解

Kontax其实是定义了所有大模型需要去利用的信息的一个总合

比如说呢 比如说以代码为例的话

那它包括了整个弹码库的结构

弹码库的原文件弹码

还有整个弹码库里面

我有哪些一片可以使用

包括人类的输入

其实这些都算Kontax

比如说呢 被几个距离的产品的力量

其实每家的产品的Kontax不一样

我以我们产品为例子的话

其实我们产品是一个帮助大家去收集数据分析数据的一个被认识

对于我们二爷

我们Kontax其实就包括了网页

还有说我们帮用户设计过来的数据表格

我还要包括说用户它想要一个什么样的数据

就是内具 断Prompt

还有包括我们在一起数据的时候

神称的一些SQL

其实都是Kontax

其实Kontax是比较索税的

大家加它一起一个整体

把整体然后去一起数给大模型

然后大模型来判断出来说

我应该给用户一个什么样的一个结构

我明白

但我才想这些过程当中的信息

被留下来在使用

或者不管是加到Prompt还是Rag里面

这个不是之前本来的

就都应该是约定俗成的一套做法了

其实这个核心是你的这些Kontax

从那来

比如说我刚才觉得立则网页

是吧

网页里面它其实网页一般

有很多无关紧要的信息

首先第一点我怎么把信用关注的信息

我出去出来

这是一部分

另外一部分的话

比如说我要伸进SQL

那SQL是怎么准确的

那我需要去教验

我教验的话

我需要利用到出去库

那我一下SQL能够进行成功

其实这些信息是

需要以某种自动化的形式去提链出来

那以前用Rag也好

还是说其他什么的

也好其实这些信息主要

输入来源是人

虽然能跟强调的是

我这些信息来源是

我通过某种自动化的方式

会去到

是

我觉得我们先把一些基础概念

快速的给大家介绍一下

因为大家经常听到很多概念

比如说什么方向口

包括MCP

包括Google前两天出的A2A

然后也有这种所谓的

可能就是Rose

不知道在Rose等到各种各样的说法吧

就在你看来

你觉得这些是怎么分类

会有没有什么优点啊

其实刚才

全靠的是你提到的

像方向靠也好

像MCP

然后说A2也好

或者不知道用的这些东西

要参与其实都是在解决一个事情

给弹模型提供更好的

一种使用工具的方法

让弹模型更好的就Take action

那这里面的区别是什么呢

其实你可以理解方向靠

是最早OPI提出来的

让弹模型使用工具的一套方法

可以理解它是个0.1的一个版本

然后MCP在这个机子之上

它又更进了一步

我们可以理解它是弹模型使用工具的

0.2的版本

但是MCP非常大的众在于它其实

类似于说统一杜良痕的一个事情

因为以前的方向靠这个阶段呢

大家使用工具的标准是不一样的

像每个国家它可能有自己的

终于一套杜良痕的一个方法

我到另外一个国家中

我就得存一点

但是MCP把这件事情做成标准化了

那这时候相当于有了一个说

放置塞都接种的一个方法

大家都可以去调用任何人的工具

其实相当于这个事情的门砍一下子降低了

然后A2A

其实我觉得A2A没什么意思

更像是一个KPI工具产

因为A2A它自己定位

自己跟MCP的区别是说

MCP它只能跟其他的2

或者说是API去交互

但是A2A它说的是

我可以A认识A认识这件的交互

其实从我们KPI者的角度来讲

这个事情是没有区别的

因为A认识A认识的

其实你也可以定存一个图

就是它可能有个掉落的那个韩束入口

那个东西也用MCP的包装一下

它也可以被其他的A认识去交亡



也可以当MCP的放置内

所以从这角度来讲

A推我觉得是属于强行造了一个自己的概念

然后找了一堆合作伙伴上

过来去推一个东西

我觉得这本账是一种

标准快乐权的一个政策

但我觉得从工程师的角度来讲

我们是跟其他MCP的

A推没有提供性的东西

明白

然后某一样的幼稚

它其实强调的是说

我把刘兰气这个事情当作一个工具

我让大盟新去调用

可能说全世界有成家开玩意的工具

那刘兰气可能是最重要的工具的一种

明白

我现在听起来

我感觉其实整体是分成两派

一派是直接代码来解决问题

我理解不管是MCP还是A2A等等东西

它是用代码去解决问题

然后另外一派是模拟人去解决问题

不管是根本的幼稚还不要的幼稚

其实它是通过讯决士笔加上CAPA的方案去解决

其实不矛盾

你也可以用于MCP的方式去进行

所以的不知道的幼稚

怎么做的

因为实际上我们是怎么去用幼稚的

一种方式其实就是

纯粹的通过普向GY的方式

去给我推算开末

又做标准备

然后可能书标

移到标准备上面

我点击一下

或者说我输入了什么东西

那实际上这个方式远不成熟

就过来有一些adapt的公司

其实是在23年11年非常火的一家公司

但实际上公司已经死掉了

就是因为这个事太难了

所以说现在大家去用博尔的的时候

其实也是通过刘兰器的API

把它包装成一个类似说MCP的兔

然后我在通过代码的形式

我去雕用这个

明白明白

就是其实它前面类似野类上细给人看嘛

背后还是用的那些

MCP跟博尔机有其实是两个

叫做镇胶的两个东西

就是后不影响

但现在是不是有问题

就是说因为毕竟很多公司

没有监用MCP的这些东西

然后甚至于说往后走

我可能有些公司为了保护自己的任何数据

它也不会监用

所以就变成有些产品是必须要使用

真的不知道在幼稚

真的是模拟人操作的那去完成

我觉得首先是这个样子的

这尔巴仅去使用MCP的标准姿势是什么

假如我是开发者

我现在发现

我想要用MCP去用某一个SARS软件

那这时候我其实不需要SARS软件

本身去支持MCP

说我不需要它已经给我提供了一个MCP的一个接口了

因为MCP是个标准化的东西

我自己可以安装SARS软件

它开放出来的就是Open API

来Open API这样包一层

我就可以把它当作一个MCP去用了

所以说前提在于说

我这个SARS是不是它已经有了一个Open API

因为Open API在国外的软件上来讲

基本上是一个标准的情况

所有的SARS都会有

不但国内它都是没有的

所以这个位置就是

国内卡外是非常不一样的

那如果说这个SARS产品

或者另外一个软件产品

它确实没有开放出来一个

所以Open API或者一个什么样的SDK的话

那这时候确实是没有办法以代码的形式去用

但是你也可以这个时候尝试着去

一用视觉的方式就是我们前面提到了

我用刘案期

相当于把刘案期的截图

我传给弹幕

现在弹幕上判断上面的一些交互元素

把作标算出来

然后我再通过水标减盘的方式

就两种办法

就是API也可以

然后机外也可以

就取得说在什么样子你才没更高效

对

所以简单给得到一条结论

就是说如果未来MCP或者后端各种接口之势的话

其实它直接雕用就OK

那如果它不支持的话

那就只能通过视觉和模拟人

去使用电脑的方式来去解决这个问题

对

而且我觉得这边还有一个点的话

是每个的Agentine

它可能想起觉得问题会不太好

所以我觉得具体说

哪个部分站的比例更高一点

可能是需要Agentine卡

根据自己的需求实际的配比

另外一点

我觉得视觉的方式

虽然现在包括稳定线

准确度没有结果那么高

但是它有个好处是它的成问题

因为我们自己的产品

其实这两种方法都用了

实际体验下来你给它结个图

然后让它去判断

其实速度要更快的

而且通效量也会很少

至少会少一个数量

比得这么一个情况

问题是机外的真正的结果

有的时候会不准确

比如说我不电上有一个提交表

单单的一个按钮

很多时候可能会去把那个

作表位置传错

明白

所以你看我前几周

在美国的时候就有一个

也是很专业的做Agent ine

犯法相关的问了我一个问题

他说他非常不理解

为什么manice要用流暖器来做这件事情

就是他的理解

是其实后端的代码

什么都可以接通就直接试下了

如果是你的话

你会怎么回答这个问题

因为我看你们的那个产品

里面其实也用了类似manice

那套逻辑吗

我觉得这边其实会有两个视角

可以大家去回答一下这个问题

一方面就有从技术的视角

一方面是从产品去的

技术的视角我简单说一下

就是用流暖器之外方式

在成本上来讲会更低一些

对 另外一点的话

其实让Agentine结果



还会很重要的一个点是

在说这个结果

我用故是不是可信的

那我们再去让用故使用

Agentine郭当中

怎么去给

用故引造这么一种可信的分为感

我觉得这个非常关键

请问一个非常重要的一个手段

比如说我让用户可以全程看到

我是带怎么去

帮你完成一个工作

让它能看到其中的每个细节

而且这个细节是以它能懂的方式

去盖子到

那对外面气质这种

天然而讲对于人友好的这种

视觉化的呈现方式

是要比代码这种

看起来黑胡胡的一个闯口来讲

要更称动 更直观的

Agentine算是每一种方案

Agentine其实就是从扣定的方案

Agentine其实都有

它也算有什么意思

对

然后Manus是典型的

也是扣定和

不知道这又是结合的

对

然后最近那个Jansbar个新出的

我不知道你看没看

我看了一些

对 你觉得它是什么方案的

跟Manus是很类似的吗

稍微会有点不一样

Jansbar我自己跑了些任务

其实我没有看到的特别多的

像Diving 或者说是Mass

这样就是有一个肉眼

人能看到的转气

或者说是

感觉它是在后端去跑

它应该后端也会用了些网约的一片

但没有把这个事情包了零户

所以我觉得它可能还不是一个

我心目当中的Agentine

但从用户是小来讲

你反正最后帮我解决问题就好了

我为什么要care

说你到底有没有在哪儿跑一个什么东西

或者它多Fancy的在

像人一样在使用电脑

或者使用有暖心

这是个非常好的一个问题

从我的事小来讲

其实我想举个例子比如说

我们俩是同事

那我们两个之间

应该怎么去建立新人关系呢

那我觉得更多的一个办法

就是假如你给我分配一个任务

你能看到我是怎么做这个事情的

以及说你后面知道

我大概是个什么样的思路

就相当于你足够的聊聊之后

你才会对我产生信任

我觉得这个点是Mexamser

就是包括像Mex的这种

我觉得其实一个核心的点

在说大家觉得Agentine这个事

还是Gradity不靠谱

所以我需要再给你看到过程

当中的东西

然后我需要说人不断去引爆武进来

对吧

你要回答它的问题

就是让人时刻感受到

你是在掌控一切的

因为那人一样

就是大家其实都会有种不安全的

这种感觉

那怎么去给人建立

使用安全感的这种边界

其实把一切做透明是很关键的

对

但这个在我自己用Manager的时候

也会觉得它也有个缺点

就是有时候它跑着跑着

它提的一个问题

我可能没看到

我就也没关

对

我本来等了可能半小时的时候

我觉得它要跑完了

对

回来一看可能还在第二步

问了个问题

其实这个就是一个Ui上

还是可以终于优化了一个点

另外一个这个事情的话

其实我觉得也是

可能是他们现在集团性的一个点

就是后面比如说它跟你的邮件

或者跟你的短信

或者说跟你平时工作用的 i-m

打通受

其实你会有一个更

容易去看到的方式去收到这个通知

所以我觉得这不是那么大问题

所以总结下

就是其实应该所有的这些

不管是什么U字也好

还是什么Coding也好

都是当下AZ的可以用的工具

对

然后根据不同的场景

好像目前来看

大多是一个组合的方案

来实现一些东西

是的

然后最终你觉得

比如说

AlCoding

核心地大家讲的AZ的

最终是不是一件事情

因为至少过去半年的一元

就是市场上最热的

拿到最多钱的两条菜道

就是AZ的和AlCoding

然后本来这两条菜道

我觉得是分别的两件事情

但愈看这两条菜道

好像愈未来有可能走到一起

因为现在反正很多AZ

也在用AlCoding的绝方案

然后AlCoding里面以后再讲说

其实Coding这个东西是

整个互联网

整个

一切的技术是实际的

然后这两条我忘了看到

哪个新闻说

Coding可能也是未来什么

EJI的一个技术

理论呆说也确实市场

就是比如说

我举个可能

不能夹上极端的力品

说Browder又死

我可以let AICoding自己做个Browder

自己做的

理论上是

只是说复用了问题

就是经济性的问题

还有是时间沉门的问题

AICoding我认为

其实只能说是大模型的一个图

这里面其实

我觉得有两个点



很关键

第一个是写作

第二个是复用

现在大家说呀

Coding讲的是说

我现在有一个问题

然后把它拆解

拆解的时候

我得每一个字问题

可能我都去写个程序

把它跑下来

像在我这个时候

是从头到尾去做的

但实际上

这是一个非常低效器

消耗成本的一个事情

那其实我们去看

在现代化的人

看过当中

我们非常强调

水

我怎么去复用

本账其实是为了让这个事情更高效

所以对于

AZNR也最优的一个选择是

我现在解了一个问题

说我首先去看

我手边也没有一个

我直接能用的工具

那当假如说

AZNR

找了一圈之后

哎

我现在没有一个

我可以直接拿用的工具

这个时候

其实他可以再退化

他说

我用AICoding的方案

我就现场

造了一个我自己独特的工具出来

所以我觉得它们两个之间

是一个这样的关系

所以你觉得AICoding

可能是对AZN的一个不充

对对对就是一个很强务力的一个工具

对

然后另外一个问题

就是我自己是觉得

当下这个市场

大家对于AZNR的讨论和理解

跟两年前

大家对于打模型什么的是有眼相

当时大家在问说

会不会有一个同用的AZN模型

还是会有垂直的模型

还是说很多传统公司

要做自己的小模型等等的

现在其实大家也在

开始讨论说

为了还是通用的AZNR

还是垂直的AZNR

我不知道你怎么看这个问题

我觉得我们其实现在处于

也加长期处于一个所谓的垂直

这么一个时代

对我最近其实非常喜欢去举一个例子

比如说做饭这个事

就很多人都会做饭

但大家的区别是说

假如说需要是咱俩做饭

可能就是我们先拿出手机拿出来

把材料转件打开

对着材料我们做的饭

但可能更好的一个人才是说

它就像是五星级决定一个大厨

我们能说

我们跟五星级的大厨是一样吗

不能核心区别是因为人家做的那个饭

可能从美观程度

从每位程度来讲

都超出我们很多倍

所以说人家是专业厨场

我们可能就是一个会做饭的一个普通人

讲回来就是今天

我感觉至少从资本时上来讲

当你说你要做个AZNR

大家就会尤其想问

说你有没有好的算法的人

你后面要怎么做ARO

怎么做差异化

从你的视角来讲

你觉得ARO这件事

跟创意公司的关系是怎么样的

或者说ARO跟AZNR之间的关系

它最后到底怎么应用呢

ARO这个事情呢

我觉得有几个点比较关键

首先是AZNR这个概念

就是从前华学习里面出来了

所以不如说ARO对AZNR多重要

而说你脱离了前华学习这么领域

你AZNR这个概念还是不成立了

所以说我们做产品的人

一定要追根寿园的看

最早AZNR他们的定义

在前华学里面是这么定义的

那在前华学里面的AZNR定义

其实只要有三个关键的东西

第一个就是状态

第二个就是行动

然后第三个就是基地信号

怎么去理解呢

实际上状态

其实就是我们前面一直在强调的Context

就说我在经过了ROMNR步骤以后

我当前这个AZNR是在一个什么样一个情况

这个地方去包括了

可能是说它的记忆

它现在看到的東西

然后AZNR其实就是Tuneos

那我现在有了一个判状以后

我怎么把这个AZNR实际的去执行

它其实就是通过韩束电话的方式去执行

因为韩束接口的东西就是

代码于数字世界交互的什么一个每一件

所以这个是AZNR

另外一个就是最后那个基地信号

怎么来理解呢

其实就是说

我现在AZNR通过韩束电话的方式

执行完了

那这是我需要做一件事情来判断是说

这一步我做完之后

我的整个状态

理我想要打弯目标

是更接近了还更远的

那这个其实就是一个一跟负移的

一个区别

就是我需要去通过这种

激励信号来判断

我应该是往左边左边左边走

那对于串野公司来讲

非常关键的一个点是

你如何去让你的产品变成一个环境

因为你只有有了环境

你才能够去描述说

强化学义里面的那个状态是什么

以及说你可选的这个行动空间有多大

那这时候就说你串训面

我浮落这个事情就说

你有多大的这个节点

它其实是由你的这个行动空间去决定的

以及说你怎么去定义你的结果

那为什么一定要把结果定义好的

因为只有把结果定义好了

它才能收脸

只有我能收脸

你才能够让大模型去判断是说

我一个行动做完之后



我理你的这个目标是更近了

还是更远

那这时候你才能够去设计

这么一个激烈信号

不断的去让你一步两步

一轮两轮三轮的去

让AZN的自我迭代

直到十千了

那个目标

所以说具体的建议呢

我是建议所有的AZN的开发者

或者说产品设计者

他们去看一下强化学义支付三层线的那本书

你就看完这么书之后

或者说有了这样的一个mine算了之后

你才能在设计产品的时候

不断去死格

不断去调整

去定义

你是一个什么样的影哀们

在影哀们身边你的程序之间的状态

是什么样子的

然后你的行动空间有多大

通这种方式

我觉得才能够去

定义出来一个好的能够自我迭代

然后能够去基于动态的情况

然后去不断地修正自己的这个路径

最后能够实现用户

目标的这么一个程序出来

所以说

我觉得就是你不理解强化学习

你就很难理解

AZN的到底是什么

你很难理解AZN的到底是什么

你就很难去设计

你的产品上什么样子

对

然后我们留一下你刚说的那几个点

第一是状态

第二是

其实类似于行动吧

特有的

对

第三个是讲你韩述

对

我觉得第二和第三个

其实是相对好理解的

因为你打怕

不是一阵它领域里面

包括所有的模型领域里面

大家在讲说

其实最重要的是

Evaluation 体系的搭建

就是评估体系

你这个东西的标准

等等

这个东西肯定是特别重要的一部分

然后左右子其实

我们刚才已经讲了很多了

那不外乎就是在现有的环境之下

我不管使用 Workflow

还使用AZN

还在AZN的去使用 Workflow

还使用各种

不知道在Use

还使用Coding

反正各种方式都上

怎么样能把它解决好吧

然后第一个

它的状态跟环境

或者说就是Contact 这件事情

那从这个角度来讲

ID肯定是一个特别好的环境吧

但Manus那种

其实

流产器

你很难讲

它是一个多好的环境吧

相对不是通用的有工具吗

对于Manus来讲

刘兰汐他的2

不是他的环境

他的环境是

他的流贩图那个系统

下面的那个目标

我觉得是这样的时间

我们来讨论一下

怎么去定一个环境是好

当然是坏的

其实刚才我们在讲

相对性里面AZN

干扬的时候

我们讲讲立信号

环境的核心

作用是提供讲立信号的反规矩制

所以我们要去看

一个环境好不好

我们得要去看

对环境能不能基于

我行动的结果

对这个结果提供一个讲立信号

那ID为什么是好的

是因为它的

选择名字叫做

集成开发环境

就是我AZN的现在申请到代码

我就代码立码

我能在这个环境里面

我去运行一下

那这时候

代码它如果出错

跑不起来

那这时候这个ID

它其实就会升成一个错误信息

那这个错误信息

天然就是一个反规矩号

那又班图

它提供不了成反规矩制

它其实是更多是个容器

你为了构建起来这种反规矩号

你可能还需要为了

无班图然后去

自己在上面打一套

容器出来

OK

然后你自己是之前做了一年多的AZN对吧

对

所以你在做的一年多里面

你觉得跟今年

比如说新进来做的人

对这件事的理解

因为我张经历也比较过

很多做一件事情的人

你觉得大家认知上

有哪些区别

或者有哪些是你踩过了坑

什么的

可以跟男人分享

有一个非常重要的一个点的话

就是一个好的AZN

应该是尽可能的

去不让用肚脑子的一个产品

那这怎么理解呢

就是说我们可以来看上个时代

产品怎么样子

上个时代的话

我们会看到一个产品

功能越多

它就能够解决的问题越复杂

但用户的学习理解

使用软件的成本会越来越大

就是用户的整个任志负担是会

不断被加重的

但是AZN头就不一样

AZN头我觉得应该是一个让用户

越简单

不用动脑子

然后就能够用起来一个产品

然后它越强大



它应该更懂用户的偏好

所以我觉得整个AZN的这个产品设计

应该是得往这个方向去走的

就像是我能接待更多的Contacts

我有更多的用户的数据

用户的一图视频

我还知道怎么样在适合的时间

问出对的问题

是

所以说

从这个角度出发的话

我觉得其中我自己的一个

跟大家可能现在有些不一样的观点

是大家现在可能会想说

恰恰到底是不是一个对于

一个男人一样好的一个交流形态

我认为其实这个聊天框

就是最最最重要的一个叫不入口

因为我觉得

对于一个AZN的产品来讲

用户交互的自由度

是远远要比用户交互的准确度

让来的关键

就是用户相同讲都可以

如果你一旦限制用户交互自由度

对于用来讲

就说

它的要习惯你要不是你习惯用户

那就是说用户的任职负担就加重了

所以我认为AZN的产品

用户交互的自由度是第一种要的东西

从目前看到心态里面来看

什么样的这个形态是最有助于用户交互自由度的提高的

其实就是这么一个聊天框

它说什么都行

难道用户交互的准确度不重要吗

其实也很重要

现在大家就讲说

我需要写了一个很好的prong

这个其实就是说你准确度的问题

但是我觉得你作为Garder

我是产品顺利者而已

其实有很多的方法

能够扶住提高这个准确度的

比如说Garder的自由度

包括说你可能记录了一些用户的偏好

其实现在代表也好manice

它能够记得一些知识下来

这是可能用户的偏好

其实你可以在产品设计里面

不断地去做些像用户提问的问题

比如说它应该提供一个比较模糊的东西

然后你去跟它不断地去成亲

然后具体然后detail

到最后这个东西

主要就变成去了

所以从这个角度出发的话

由于你有很多的方法

可以提高用户交互的准确度

那其实这应该是你作为Garder去考虑的事

你不能把这种交给用户

我们要做的是怎么让你这产品更是难

有让用户就像一个非常幸福的小朋友一样

能够用你的产品

这个就有点所谓的Coding的那种感觉

就是它就户在外部里面心流的状态

你可能感受不到时间而流失

你不需要动脑子

你又感觉自己坐在那儿就是一个非常放松的状态

最后事情就做完了

对 让事情就做完

我觉得AZN的就应该做到这点

所有实现不了这个承诺或者在设计上

没有这种意识的AZN的产品

我觉得都是为AZN的

所以说聊天况就是最好的交互形态

你不需要加个外交接口

你不需要再加其他的组件

你只需要做的事情是

你把合适的一个组件

在适当的时机跳到用户面前

然后让它去交互

比如说可能你在后台设计的底层代码时间

里面看到200个组件

这200个组件可能会用回来讲

它可能一直整靠是

身杨一般九十个

它用不到

用不到就意味着

不会影响这个用户的理解

这个我是同意的

我觉得如果纯是比较天况

不一定是最高效的方式

但是我觉得它在事实的时候

推出一个合适的产品线面

然后用户去选

应该是一个很合选的线

对

所以总结下

我觉得首先一突是别是特别重要的

首先得知道用户到底要干吗

是的

但这个我觉得Context是两个互相印证的

就是我如果Context足够多

可能能猜到用户要干吗

是的

或者用户要告诉我要干吗以后

我还是要收集更多的Context

帮助用户完成

以及说我都要知道中间有哪些

问题是要提出来的

所以这里面很重要一点

就是你自己的模型得有能力的判断

当起来Context是不是足够了

如果不够

你应该在环境里面

通过环境给你提供的一篇

你自动地去盖子

无论是Ragged

或者是没有方式一样

这个我觉得跟模型的智能式相关

另外一可能也是跟垂直里面一样

都好事相关

其实这里面有点点点是System Prompt

那System Prompt其实就是

你穿了AZNK大人

你需要就负责的东西

那System Prompt基本上现在

像我们看Cursor

你会发现他们System Prompt非常长

几枪好

对 就是预制的

Prompt

而且我觉得这个确实

在持续领域里面才成立的东西

因为我只有知道

他上来要干嘛

我才能

写好这个System Prompt

对 而且我必须要了解这个领域

我可能才能写得更好

比如说就举研究的例子

我就知道说他肯定需要举搜网页

肯定要去搜集一些数据跟文章

然后他肯定要把其中的一些东西

也要摘出来

把这东西再放到

不管一下还是一个PPT

一个什么就这样报告里面

我知道他一定会做这几步

所以我可以针对每一步去做优化

去提前预制好我的Prompt的等等这些东西

是的

但如果这个东西

它不是一个比如说研究类的

它是一个统员针的话

这跟上来

它有可能是想做研究

有可能想做个动画篇

也可能想生识了什么东西



这个准确率一下子就下枪

因为它可能提完每一步

就90%成功

但是若干步者以后

它是一个陈述的一个关系

而不像是认识的

首先第一个

它可以把每一步的成功率提到100%

第二个

它哪怕出现问题

它也不会说上一步的

会影响下一步的结果

所以我记得之前好像是苹果

还是谁

它做过一个事情

就是说

就你打开这个网页的上一步

你是在看什么网页

也是一种context

对

其实就是

我们讲看待在里面

包括历史的对话信息

那其实历史对话信息

只是说人看到的历史对话信息

那其实对于机想

它在你没看到的后台

可能访问了很多个网页

像当时说

我当前的这个状态

其实有过去做的所有的事情

所决定的

那我过去做的这些事情

包括我做了什么

我看什么

我发了什么

整体而言

构成了

现在的这么一个状态

那这个状态的整个集合

即就是所谓的上台

或者一个context

对

然后包括

OP&I究竟刚出的那个

它的记忆系统

其实也是一种context

对

最后的东西

我觉得就是

如果起手的时候

能有更多的context

肯定是最好的

你看我前几周跟张鱼光吃的饭

我觉得它提的一个点就特别好

它就说

当你点开

把我一个APP的时候

就打开了一下

其实就已经提供了海量的context

是的

我点美团

我就告诉他

我要点外卖了

然后点滴滴

我就告诉他

我要打车了

然后这个产品里面的所有的东西

是基于这个context在设计的

对

因为这个样子的话

相当于把大家去拉到了

一个共同的范围下面

对

就把一度十别收集到

然后再在

跟他交互合作的工程当中

不断地去积累context

是的

这看看有可能是有用户的信息

有各种

你在直星流程当中的信息

然后有各种网站

各种环境的信息

等等

所有的这一切都集合起来

然后继续去识别意图

然后去判断中间

哪一步可能会遇到什么问题

然后再问用户准确的问题

对

就是我们说

你想更好的聊天一个人

你去看他的过去

你更好的想了解用户的意图

你去看他从哪里来

他中间的路级是怎么样

你把路级都需要保存一下

所以Google很早就开始保存吗

对

所以这个就是Google在A and A 90

最大的经验就是

它是有一大堆的

用户的点击的数据

然后可以去帮忙去分析一图的

对

包括其实大家讲说

到底数据对A and A这么重要

数据这东西还是很重要

但我觉得有一个限制条件是说

它得是高级量的数据

那什么是高级量的数据

高级量的数据是说

我不仅要知道数据的

输入是什么

我还得要知道结果是什么

我同时更要知道

从输入到输出

中间发生的事情

中间这种各个变化

的数据是什么

只有我这些数据中和它一起

就是它其实是一个数据区域

这个是很关键

比如说夏为奇

我只知道这一步骑

怎么下

不关键

重要的是

我得知道前面它一白手骑

是怎么下的

最后才可能说

哎

下这一步骑

从来我能够推到后面几步骑

它是怎样的

这个时候整个过程的数据

这个骑盘面怎么来的

才是被关键

那我们再回到刚才那个地方

就是有没有技术过产品程的点

我觉得

大家毛都费得很重要

就是你得要去能够去理解到

现在模型的编辑上现在呢

对吧

哪怕是现在最先进的

OE

还有就是3.7也好

你得要知道

在极限性我想

你想做那个事情

它能给你做到

这个顺便可以聊下

你们现在带做的产品

你之前给我

受过一个你们的Demo

对

我觉得这个点上可能就是

你刚才讲那个



相关的一个延伸

就是你们选的一个方向

是基本是已经实现了一个东西吗

对

其他的好多做AZN的

我觉得还是在偏讲故事的阶段

所以可以给大家介绍一下

你们目前的一些产品吗

我们产品其实是一个电台AZN

然后我们的目标是

把整个的数据收集

然后到数据处理

到最后几率数据的行动

整个链路都避缓了

然后我们现在呢

其实是在数据的收集处理质上面

其实已经做得比较好

大概的它使用的场景

会是怎么样的

能够举一两个例子

首先第一个场景的话

就比如说是

你需要去找一些吸收线索

比如说我们的一个用户

现在是它们有很多的这个看用用户

然后他们希望知道

这些看用户是哪个公司的

因为他们很有可能想把

他们自己的善化的版本

去卖到这个公司里面去

那以前这个事情是没有办法做的

就是你只能可能忍弱的一个

一个去分析

但现在的话

郭芯是能够做这个事情

具体怎么做的

其实就是我们会进入到

用户的给他App的主业上面去

在主业上面去分析

它有没有一些可能跟它

是那些公司的一些线索

比如说代码鞋鸟进入

或者它自己的一些吃椒没体上号

如果有事要没这样

我们会进一把事要没这样号

再去看它有没有公司信息

其实跟人去手动翻盘是一样的

我们是模拟的人的中文一个过程

OK

还有别的例子吗

另外一个例子就比如说是

找到WSE

他们最近几个败迟的公司列表

然后再找到公司列表的创始人

然后再找创始人的Twitter

去关注一下

让他们发个私信

然后由于我们

用一些

来扣顶了一些技术

所以我们能够保证的这个流程

是100%准确的

实际我们也跟

TPS测试的产品

还有Miles的产品对比一下

他们其实并不能保证

这些公司一个不拉的抓去下来

这是一个点

第二点的话

其实我拿到顺以后

我是需要有进一步的动作的

就是我要可能要去跟这些公司的创始人

去建立一个联系

跟那些稍后退下

但像TPS测试这样的产品

它有报告

甚至万就是一个报告

在美国后面

Miles只是他们期待

也不能做后面那一步吗

他们可能可以

Miles的话

现在就是没有办法保证成功率

因为它是在中间

动态随机生成的代码

它需要不断的去做第八个跟调整

对

所以你们从进入上去怎么解决这个问题

其实我们现在

自己内部做了很多的这种小的兔

然后让它去掉进小的兔的时候

因为这个兔我们自己内部人

是一样过策果的

所以它能够去保证

就是你给它提供了一些更好的工具

是的

而且这个工具

其实因为我要成券来出身

所以我非常喜欢用一个辞教服用

如果有一个工具

我下次用的时候

我不是把它从你写编代码

二手把已经测试过的工具用起来之后

其实效率跟高就这么更低了

但像Miles的话

现在它没有这个机制

它每次都是打开那个ID

然后从零开始写代码

明白

所以进进去还是encoding的能力

还不足够

百分百的准确地完成这件事情

所以是人先去做了很多工作

然后再一样AI去调用这些工具

会是一个更好的结果

对

所以这就是你要选择

通用性还是准确性的这么一个区别

你越通用

你以为你的方法要更范化

更范化的意味着

它的实际性更高

我觉得这是隧道是吧

对就是因为如果很通用的话

它就自己要写超多量的一些工具

对

对

所以我觉得核心还是你怎么去更好的

去利用这些工具

比如说它油件或者这个

你可能这种场景比较简单

但是如果你写一个

比如说我们讲

你出去哭

你又可能

从头给它写一样

不可能

你更好的想

我怎么去更好的方式

然后一个mcp

或者怎么样起来方式也好

我去在我的人里面

跟出去哭的叫明白

然后我就仅仅想到另外一个问题

就现在也开始有人在讲

说Workflow是不是未来要被agent

颠覆掉

但如果这么讲的话

可能至少相当长时间内

Workflow还是非常有价值的

因为我觉得可能对人性有关系

大家再去聊一个事情的时候

非常希望去把一个什么东西踩一下

创造一种对立的视角

或者是从流量角度或者怎么样

获得更多的眼球吧

那实际上

我觉得agent跟Workflow长期会

处于一个共存的一个状态

那Workflow跟任的核心的区别



其实很简单就是

Workflow是任取懂的

Agent是AI取懂的

那任取懂的跑速是什么的

好 所以它一定是稳定可靠靠谱的

但是它的缺点是在于说

它不够范化

那Agent的特点是反怪了

就AR取懂它足够的范化

它能够解决

我之前没有想到的问题

来问题在于说

实自里面它可能会有五次把这事情搞砸

所以我就可能

AI会负责20%更开放的问题

但剩下80%可能是更常见的问题

换言之心

AR主要负责常委

明白

所以你最终

你们现在带做的产品

跟其他的那些Agent的区别

你觉得是什么

我判断Agent不同的标准是

我去看他们的底里面的real t

而不是说

大家好像看了一个事情都能干

对

那从底里面real t的角度去看的话

你会发现真实麦克也好

你不是真实也好

麦克思要

其实最后给大家呈现的主要的形式的内容

其实就是一个报告

当然

麦克思要做得更深入一点

他们有可能还给你呈现投票

或者说是简单了一个网页

但我觉得这个只是报告的另外一种表现形式

但本账也他是给你提供了一个报告

所以我觉得

从这个角度出发

我觉得大家都是

调案Agent

那这个是依赖

因为至少我们目前没看到这个

麦克斯可以帮我去美团下个单

或者京东买个什么东西

另外一类的Agent就是所谓的扣顶Agent

那扣顶Agent的底里面的结果

那就是代码等生吗

我觉得你代案定义挺好的

就是

可能像麦克斯在展示发了什么

它其实属于调案类Agent

对

然后大家可能是扣顶Agent

对

那你把自己定义成什么类

这是个表格Agent

那我们之前的核心区别呢

是说大家在去分析问题的时候

一般来讲会有两种摆赛的

第一种就是可能只是说

电信的去分析一下某一类问题

那电信分析一般来讲可能就是

用Deeper 这是这类的报告

然后我去看一下

大概是个什么样的情况

建立一种感觉

那另外一种分析场景呢

其实是定量的分析

我需要知道一个

非常准确的一个数字

那这个数字

怎么能保证这个准确的

其实我得要这个分析的数据员

是准确的

那这个数据员一般来讲

就是一个比较完整干净的一个表格

那我们做的事情呢

其实就是把各种各样的数据员

首先变成一个完整的表格

而不是说我就是一个

什么搜索的一个借口

拿到各种各样的网页

去做总结对

我们不是的

我们现在把各种各样的数据

变成一个结果和一个表格

然后再拿这个表格

然后去做下一步的分析

所以我们之间是一个

定义分析的一个差异

对

所以你的核心是中间会有一个

AI 声声的表格

对

但是这个表格

你的游戏里面数据

什么的会遇到

AI 我们经常讲的那些

它的冠决什么问题吗

然后我们在工程上

把这个问题已经解决了

OK

然后我想到了一件事

就是

是不是AI coding

它相当于是大模型的一个

翻译跟一个住手的感觉

就是是不是

所有的任务都中间

可以加一个AI coding来解决

正确定的问题

它解决所有的换决的那种问题

我觉得是的

用现在还比较十毛大化来讲

我觉得AI coding

就是大模型的一个零巧手

对

因为我们现在都在讲

说如果让大模型

不断地去做一个任务规划的话

一旦它每个任务的

成功率是90%

那可能是不以后

它就是一个零年久的十次方

它成功率就很低了

这样的话

我们怎么去让上一步的失败

不影响

下一步

那可能我们就需要一中间

一旦满了心去运行它

因为一旦满

我只要能保证

90%成功率

我可能十次里面9次成功

我只要把9次成功

那个带满了留下

然后做一个正确的case

然后再进入了下一步

那就是说

其实就能保证每一步的

100%中确的

这个很合理

但这个

因为我不知道

你们从实际的操作

和经验来讲

是每一个大模型的操作

都应该先翻译成Coding

翻译成一段程序吗

我觉得这段还是想拿

MCP来说例子吧

因为MCP

背后

店铺的那些

所有的免



本质上全是代码

所以从这角度来回答

却要是你的问题来讲

我的答案是是的

我想再回到我刚才

问过的一个问题

是说你自己做了一年多一阵灯

然后你有什么能跟别人分享的吗

对 这种东西没讲的

刚才其实提到了

普产模特费劲

第二点我觉得很重要的

就是你得想明白

你给用户

到底底里味的result

是什么

那你觉得把result的想明白了之后

你才能够

不断地去

通过设立其反馈的一个

激励机制

然后不断地去优化这个结果

因为如果你这个事情

想不明白

它就不收脸

不收脸就意味着你可能

底里外那个结果

就是一个比较差的一个质量

比如说像我们做的

对它睡在这个场景

我们其实试过

operator

试过真是buck

试过gorg

你会发现它这是演的帮你

比如说

WSE刚才提到的例子

它确实演的是

WSE公司的表格

但是你会发现那个表格

永远缺一些公司的

而且你让它去进一步的

你想去知道

公司的创始人

是谁

创始人是那个学校的

工作多少年

它的推特账号是谁

它是完全解决不了

这种解决

我已经有了一些基本的事情

我在这个数据上

我进一步去分复我的数据

它做不到

你就要把结果想明白了

你才能够去不断地

在基于低步的结果的

进一步上去优化

第二步结果

就是一旦你把结果想明白

你就会发现你这个东西

就算不好

变成一个所谓的看起来

有点想通的一个证则

我觉得另外一点还有很重要的就是

我觉得做 agent的人

要解决两个新人的问题

第1个新人的问题

是你作为 agent的casher

你要足够的去相信大模型

这个点会缠什么也影响

如果你不相信大模型

你会做一个事情

你说你不断在代码里面

加些什么限制条件

比如说就写pop

你就是个谁

你只能干某某事情

不能干别的事情

那这种事儿的影响

再来说

你会发现大模型的整个的

就是范化能力

其实被你认为的影响

变低了

那用于现在我自己的一个材料来讲

就是你会导致

你们的 agent 对大模型的

智能利用率下降

对

我们刚才聊言

经常会发现去受这种历史

就是本来在大模型上

你是想去做更好

然后封装好的东西

做个反而

做个半天

你可能不如我把这个去求直接

放大模型的理解

是的

是的

这个就是优于你自己对大模型

不过的信任

你作为开发者

你会产生一种内心的不安

这种不安稳

然后促使你

用一些传统的

所以如被死的方式

几十个年

时间最后你会发现

这是在开道车

所以说

agent的开发者

需要去解决自己内心对

大模型的讯论

另外一个点就是

你得一定要想好

你的这个产品

该怎么通过设计

去解决

用户对你结果的信任

很好

一个例子的话

我觉得还是

DPCR1

是在DPCR1

就前期

我用一些像

DPCR这样的产品

它是那个报告

我在看到的

第一眼

我其实会以一种

更批判的角度去看出

这东西到底是对待的错了

为什么呢

因为我不理解

它是这东西怎么来的

但是

R1穿以后

它把中间这个

Rinnie的过程

告诉我

在我心里上

让我更舒适的感觉说

我看到它怎么想的

所以我更

愿意去信任这个结果

对

我前几天跟朋友聊

我就讲了一个事

我说

其实Manus

很大的一个价

就是它给

所有的AZ

充电车打的样

就也许

过一段时间以后



大家回头看

就是Manus这个产品

就好像当年的

Kart. ai

就是让大家知道

说OK上一代的

大模型

最适合的就是

CIA这样的产品形态

对吧

对话题

大概就是这么做

然后大家在这个

技术时上去

不断地怎么样改

然后现在的AZ

可能很多

都是在Manus那个

技术上去

不断地去修改

我们说你怎么看

这个问题

是一个比较

通用的一个产品形态的

我觉得是的

我觉得Manus

是在

单秘的

技术上

在UI事情上

跟进一步了

其实核心

就是它把中间

所有的细节

都暴露传

人看这个东西

以后它心里面

就会存在一种

满足感觉

爱选感

这你对于

为了AZ

真的发展

会有什么预测吗

我觉得

以现在

AZ发展速度去

预测几年

这么一个时间快乐

我就太难了

预测半年到一年了

就分享一个框架

就是跟大家一起去思考这个事情

首先

想要去预测

一定要先去抓关键编辑

那AZ的一个

事情里面的关键编辑

是什么

其实前面

也讲到了

区分AZ的方法

是去看它的

结果的好坏

通过结果的好坏

来判断

它到底

是不是一个

更专业的AZ

结果呢

其实就是

上下玩

Connection

加M

所以说

我觉得核心编辑

就是这两个

你想AZ的突破

那你觉得

Connection或者说

M至少有一个是突破了的

首先来看M

M的下个突破呢

我觉得基本上也就是看

GPT5什么是出来

那我觉得从时间线来讲

我觉得跟GPT5是在

今年年历左右

会出来

可能年历的时候

或者明年

元旦之后

我们可能发现

AZ能力会更进一步

这个是一个界点

另外一个界点的话

就是说

GP5出来了以后

什么时候成本

会

变成一出来的

5分之一

甚至说是十分之一

这个过程可能

也得需要一年的时间

比如说

DeepSecretary

Maybe

我现在不知道

它会有多大的突破好

因为

本质上

我觉得这个事情还是得需要

访内是摩托去做突破的

那也就是说可能

比如说是什么时候

DeepSecret微似出来

然后去微似的可能

我在生成一个

什么样的有推理模型

所以我觉得这个是

一个就是说

你从模型的角度

现在第一个

各线硬的方式是摩托什么时候出来

以及说

访内是摩托的成本

什么时候能降下

因为房间是摩托出来

这是我们就能做Demo了

Demo有了之后

大家就有V7了

但是只有得到成本降低

之后最后最后

才能够化使用

所以我就接下来

真正很好

用的客内人的以外的

也可能出来

我觉得可能还是得到

26年的下半年

就是弹幕的

大家真正能用上

这个是一块

另外一块的话

我觉得就是从Connection的角度来出来

其实也可以来看个例子

就是Cursor

它为了把整个

Coding

这个事情的Connection的做好

每绕VSCode

其实做了大量的

二次开发

他们是从二二年开始

然后到去年下半年

其中间是花了一年半的时间

这里面呢

我觉得工程量是非常大的

我觉得这也是



大家现在普遍的

我认为的一个物件

就是会觉得

Agent这个套口

其实我觉得不是Agent

怎么样的工程复杂度

我觉得是愿意超出大家想象

你想让Agent

在Connection这个层面

去得多

你是需要大量的工程上的

尝试跟积累的

所以说我们

从现在这个时间点去看的话

就是假如说

DPC出来之后

大家第二天立马

开始干这个Agent

然后优秀的同学

干的快点

我觉得也得要6个月

就卖一年

就可能就一年

所以说

可能到今年的

Q3吧

我觉得可能会有些

更好的Agent的一个产品出来

那我再不许过

因为你都在讲到

说判断Agent

就是它的RESOT

怎么样吗

但RESOT的好的

就一定是

说这个Agent的工资好吗

因为有可能有一个工资

就是通过各种手段

可能它就是某一个

Prompt的写得对比好

然后它的RESOT在阶段线就更好

所以我想说

如果假设你是投损的话

现在你面前

有好多做Agent的工资

你会问哪几个问题

来判断这家工资的好欢

因为投针线的遇到

各种工资对吧

你来的就是一个

挺好Prova的Founder

然后可能有一个算法

或者什么技术的

挺好背景的一个入

然后想说

我要做某种领域的Agent

然后其他的讲的东西

其实大差不差

也都差不多

那其实我为其为就一个问题

我就会看他们团队面

有没有人去看我前面

提到那个就是

SATA那么输

因为我觉得

你只要看过这个输

你又一定会有漫赛的

有这种漫赛的人

它一定不会去

能够长期容忍

我要用一些

非常符号主义的那套方法

去达到一个结果

那你这个要求可能有点高的

有很多人可能透巧

就没看过那边输吧

那我其实就会可能

可能问一个事情

就是你的产品里面

对

环境反馈的基地信号是怎么设计的

就是到底在你的产品里面

什么样子

一个行为是一个好的行为

什么样的行为

是一个差的行为

我觉得这个问题很关键

因为我这个信号之后

你才能够让大模型去点带

就说把上一位的结果

跟环境提供的这个

好与坏的评价

对为大模型

让大模型再去点带一个

从这个角度来讲

确实通用了

好像就不如垂直

通用你很难定义这个东西了

对

就是这里面补充一个点

就是说

大家现在觉得有大模型之后

结果化数据这个事情不重要了

不

它其实还很重要

因为大模型的输入

我们可以是非结果化的

就是我把什么乱七八糟的文档

闻笨PDF

我一共脑子进去

但是大模型的输出

一定得是结果化

因为你只有结果化了

你才能够去用

带马或者说是归子的那一套东西去教验

让它是没有办法去教验

所以说

你的记归信号

其实最后在

带马上体向来讲的话

就是你大模型里面

输出结果里面

坑爹子段的指导的是什么

对

就是你这套标准是什么

所以我觉得让我问的话

我会有这样的一个问题

Mindless的记归信号可能是什么

如果你负责

Mindless这个产品的话

我想不到

因为现在

他们广场的那些例子

我看下来其实

本质上的判断

还是

再让大模型本身

或者我写了另外一个prompt

然后去判断说这个东西是

好的一个机会

坏了一个机会

但实际上

如果大家去用

XGB4G的产品

你让它比如说

我最近在写PET

写了一段看上板

然后让它不断一划看上板

我给它说

不满意

你敢

你只是说提供这样的

非得办的话

你会发现它最后

是现有一个奇怪的一个

死云皇里面

它就出来的

所以你得要给它提供

大模型本身

所不具备的那些



反归型号

我觉得财亚才能够

把这个结果从这个死云皇里面

跑出来

所以如果我问你这个问题

你们最底产品的那个反归型号

是什么

其实我可以讲一点

就是我们取出这个事情

不存在Helps Nation

就是因为我们其实

让大模型负责的是整个

业面结果的分析

然后业面与业面之间关系的分析

通过这些分析的结果

我会身上一段墙本

然后我这耳八景的

其实往下来

搜集数据

深层数据的这个过程

其实是用代码实现的

那代码其实就是

一个点形的能够给我提供

反归型号的这么一个

这是一个点

另外一个点的话

表格这个东西

其实很直观

一个地方是空的

我一眼就能看见

那其实这个也是一个反归型号

明白

很合理我觉得

那报告不好不一样

报告那个东西就是

你很难去精确的去描述到

说哪一行哪一个字

怎么样

但是表格的话

由于是个结构化

所以我是可以去描述它

嗯

好

我觉得今天我们其实

讨论了很多

AZN的相关的问题

我相信是把那些

大家核心关注的问题

都比较快速直接的

去开拓到了

然后包括文风也提了一些

他自己做这么长时间

AZN的一些经验吧

我最后一个问题

给大家三个理由

我说

图像场景之下

为什么用你们

而不用

MADNESS或者

ZANCE8或者其他的

工具产品

其实我觉得都不用三个

就是你只要有对于

高级量是你抓据的需求

你会发现

现在市面上的AZN产品

只有我们的产品可以做到

就是百分百准确定

对百分百准确定

然后百分百稳定

百分百不给你丢数据

所以什么时候

大家可以用到

在这周的时候

我们就是开放

线上的WeTinList

看我做大家注册

一个DinList的以后

可能在两到四周的

左右时间

我们就会能够

这样八点的使用我们的产品

大家可以去

访问我们的官方很暂

实际体验一下

以防大家不太只要

走平台随之

意义体能点抗

就是表格的那个单词

OK

好

那谢谢问方

好 谢谢