AiDD AI+研发数字峰会
AI+ Development Digital summit
第5届

AI辅助测试脚本失败分析和修复

胡应广 | 华为

科技生态圈峰会 + 深度研习

——1000 + 技术团队的共同选择

KEYLINK ing

K+峰会

K+峰会 · 敦煌站
K+ 思考周®研习社
时间：2025.08.29-30

K+峰会 · 上海站
K+ 金融专场
时间：2025.10.17-18

K+峰会 · 香港站
K+ 思考周®研习社
时间：2025.11.25-26

K+峰会详情

AiDD峰会

AiDD峰会 · 上海站
AI+研发数字峰会
时间：2025.05.17-18

AiDD峰会 · 北京站
AI+研发数字峰会
时间：2025.08.08-09

AiDD峰会 · 深圳站
AI+研发数字峰会
时间：2025.11.28-29

AiDD峰会详情

演讲嘉宾

胡应广

高级测试开发工程师

华为ICT软件测试工程与自动化专家，多年云原生软件测试开发相关经验，负责GTS产品测试系统构架设计与开发，主导产品线大模型应用测评系统能力构建和AI辅助研发质效提升项目的探索与实践。

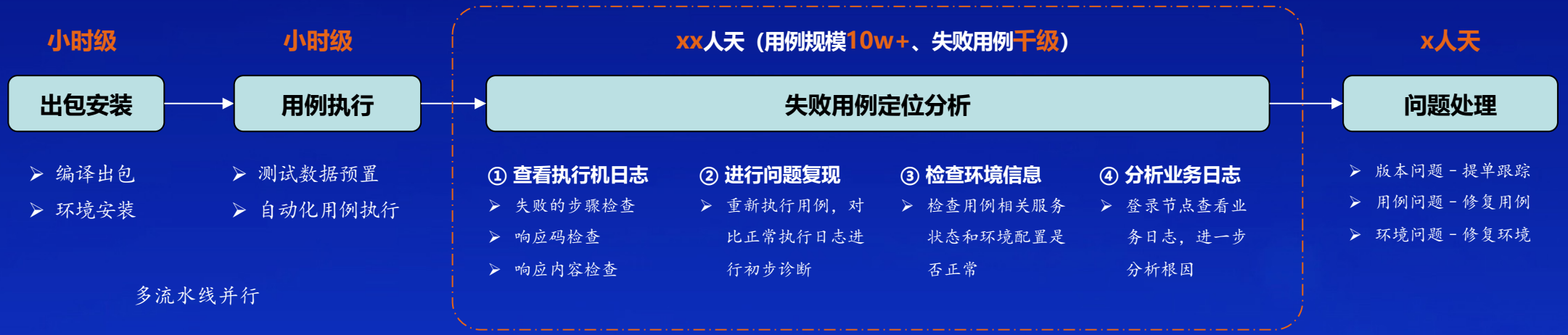
目录
CONTENTS

PART 01

背景介绍

自动化测试过程分析与业务痛点

- **自动化测试流程与耗时统计**



- **业务现状与痛点**

敏捷开发模式下需要通过执行大量自动化用例快速反馈版本质量和存在的缺陷，产品自动化测试用例越来越庞大，全量执行一次失败用例多，全部依赖人工分析投入**效率低**、**工作量大**，**问题反馈周期长**，无法满足业务诉求；即便把自动化用例通过率提升到99%以上，每天仍然有上千个失败用例要人工分析，按照现有效率计算仍需要投入数人天的工作量。

PART 02
问题与挑战

测试脚本失败分析主要瓶颈和挑战

日志获取困难

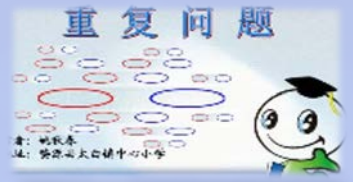
- 业务环境集群化部署组网复杂，日志分散在不同的节点容器上，人工获取用例执行日志困难
- 自动化测试脚本并行执行，无法准确区分失败用例对应的服务日志

人工分析效率低

- 产品框架复杂，测试脚本失败影响因素多，人工分析效率低、周期长
- 历史经验无法很好的继承和复用

相同问题重复分析

- 同一批次相同/相似失败测试脚本重复分析
- 相同版本跨测试环境相同/相似失败测试脚本重复分析
- 测试脚本历史相同/相似问题重复分析



PART 03

解决思路与整体方案

关键问题解决思路

问题1：相同问题重复分析

解决思路：通过日志特征语义聚类和相似度对比去重

1、通过聚类算法进行问题去重，避免同一批次相同/相似失败测试脚本重复分析

2、构建已知问题根因失败用例向量知识库，通过语义相似检索召回历史失败根因，避免历史相同问题重复分析

问题2：人工分析效率低

解决思路：大语言模型在NLP任务上能力强大甚至超越人类，引入LLM替代人工进行测试脚本失败根因诊断

1、通过大模型外挂领域向量知识库，增强LLM进行测试脚本日志解析能力

2、构建典型失败特征-问题根因因果关系知识图谱（KGs），通过RAG技术增强LLM对失败日志根因推理性能

整体方案

4 应用层

- 测试作业平台
- 对接自动化测试流水线
- 失败分析Copilot
- 模型评估

3 服务层

- 失败日志特征聚类
- 失败日志解析
- 历史相似问题检索
- 失败问题根因联合诊断

2 资产层

Prompt Engineering

- 问题根因推理Prompt模板 (few-shot)
- 失败日志解析Prompt模板

日志处理引擎

- 日志采集
- 日志模板
- 规则管理
- 失败日志预处理（降噪）
  - 失败重试日志
  - 等待时间日志
  - 日期、时间戳
  - 超长的HTML Body体
  - 无用的请求Header
  - 超Token长度裁剪 ...

知识库 (RAG)

- 专家规则库
- 典型问题案例库
- 存量问题已知根因知识库
- 知识图谱（KGs）

1 模型层

- 已知问题根因/问题分类 - SFT
- 华为盘古大模型
- **LLM** 华为盘古-S大模型
- **Embedding Model**
- **聚类模型**

PART 04

具体实现与技术实践

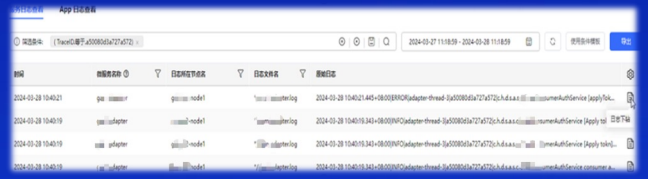
测试脚本日志采集和清洗

日志采集 → 日志解析 → 特征提取 → 日志清洗

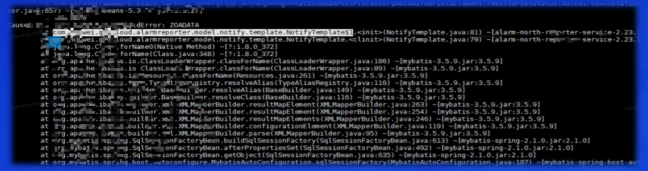
失败脚本日志采集

1、测试脚本执行日志

2、API接口调用链日志



3、服务端业务失败日志



日志加载和预处理

- 日志加载
- 选择模板
- 解析日志内容
- 日志保存

提取失败要素

- 测试步骤
- 请求指令
- 接口标识
- 预期结果
- 实际结果
- 响应码
- 响应内容
- 错误信息
- 调用链日志
- …… ……

对日志降噪和过滤

- 失败重试日志
- 等待时间日志
- 日期、时间戳
- IP/Port mask替换
- 超长的HTML Body体
- 无用的请求Header
- 无用的solution日志
- 超长Token内容裁剪
- …… ……

原始日志

特征日志1
特征日志2
特征日志3
…… ……

清洗后的日志

通过日志清洗，保留用例失败原因相关的关键特性要素，保证向量知识库相似问题检索召回准确性和大模型推理输入必要信息的完备性。

测试脚本日志内容特征聚类去重

针对同一批次失败测试脚本日志特征提取和清洗，利用聚类算法，并通过失败特征进行二次分类和离群点移除，实现不同失败用例相同/相似异常模式精确聚类，避免相同问题失败的自动化用例重复分析。

常用聚类算法：

K-Means

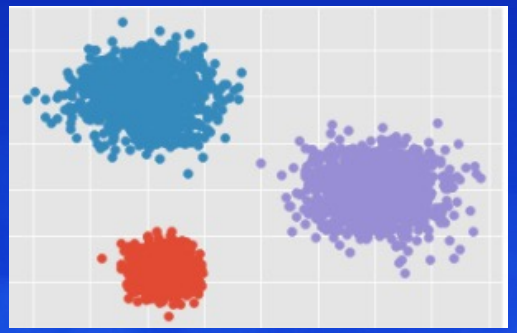
K-Means是一种**基于距离**的聚类算法，它将数据点划分为K个簇，通过最小化数据点与所属簇的质心之间的平方距离来确定聚类结果。

优点：

1. 原理简单、实现容易
2. 收敛速度快
3. 对正态分布的数据效果好

缺点：

1. 需要预先指定聚类的数量K
2. 对噪声和孤立点敏感
3. 只能发现凸形簇聚类



DBSCAN

DBSCAN是一种**基于密度**的聚类算法，通过确定数据点的密度来划分聚类。DBSCAN将高密度区域视为聚类，并能够识别出噪声点和孤立点。

优点：

1. 能够对任意形状的数据聚类
2. 能够识别出噪声点和孤立点
3. 不需要预先指定聚类数量

缺点：

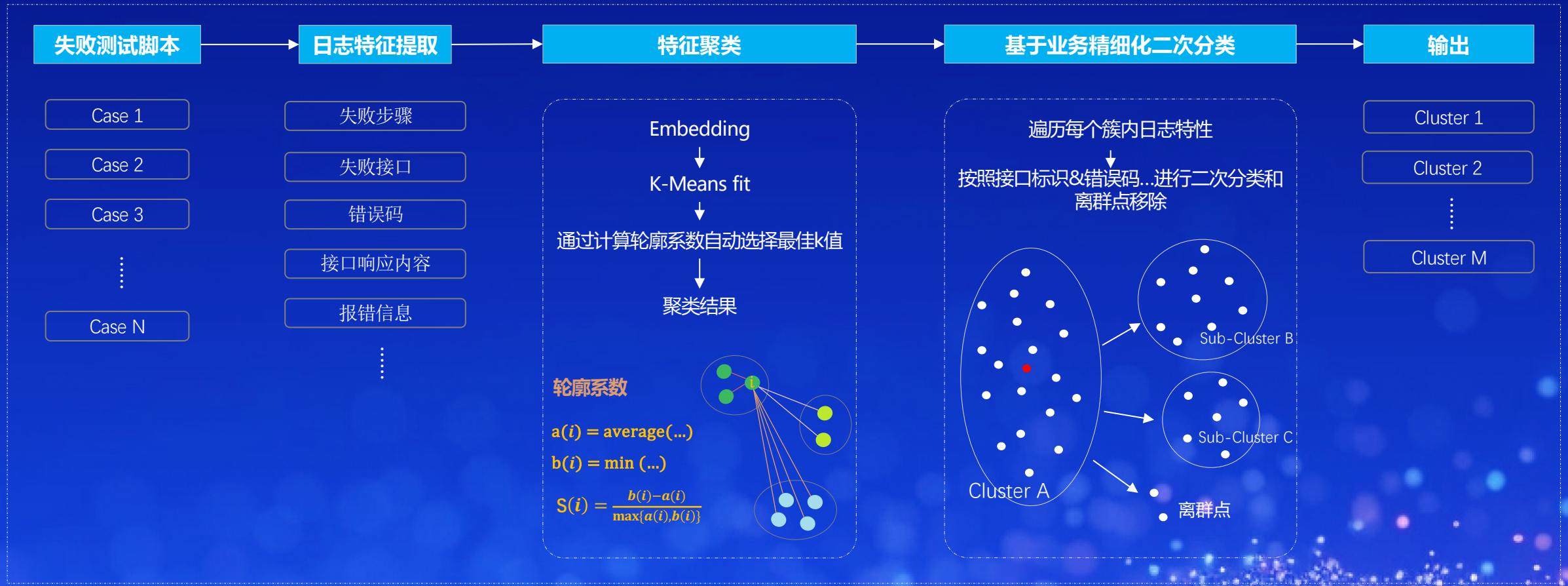
1. 需要指定领域半径和最小样本数
2. 参数调优困难
3. 对高维数据处理困难

根据日志特征选择合适的聚类算法作为基础进行算法优化或者进行二次分类，实现簇内样本的同质性。

测试脚本日志内容特征聚类去重

针对同一批次失败测试脚本日志特征提取和清洗，利用聚类算法，并通过失败特征进行二次分类和离群点移除，实现不同测试脚本相同/相似异常模式精确聚类，避免相同问题测试脚本重复分析。

实施方案：



相似的失败日志其根因也类似，通过聚类将日志特征相同/相似的失败测试脚本进行分类，避免同类问题重复分析。

聚类效果

➢42个失败用例聚类为7类，收敛比为83.3%

	问题分类	平均轮廓系数	用例数量	占比
>	ERROR - exception: java.l ...	0.76	4	9.52%
>	ERROR - exception: java.l ...	0.60	2	4.76%
>	ERROR - exception: java.l ...	0.57	17	40.47%
>	ERROR - exception: java.l ...	0.48	10	23.80%
>	ERROR - exception: java.l ...	0.28	6	14.28%
>	ERROR - exception: java.l ...	0.07	2	4.76%
>	其他	-1.00	1	2.38%

➢161个失败用例，聚类后为9类问题，收敛比95.7%

	问题分类	平均轮廓系数	用例数量	占比
>	ERROR - exception: java.lang nError: $body.result.t l expect [==][1] but found [0]... expected [true] but found [false]	0.85	2	1.24%
>	ERROR - exception: java.lan tionError: $code expect [==][200] but found [500]... expected [true] but found [false]	0.81	2	1.24%
>	ERROR - exception: java.lang.NullPointerException	0.81	92	57.14%
>	ERROR - exception: java.lang. ionError: $code expect [==][200] but found [400]... expected [true] but found [false]	0.71	2	1.24%
>	ERROR - exception: java.lan tionError: $body.resu status expect [equals][0] but found [1]... expected [true] but found	0.61	28	17.39%
>	ERROR - exception: java.lan onError: $body.data.node-fai witch expect [equals][NA] but found [forced_on]... expe found [...	0.52	2	1.24%
>	ERROR - exception: java.la ionError: $code expect [==][400] but found [200]... expected [true] but found [false]	0.47	9	5.59%
>	ERROR - exception: java.lang tionError: $body.result[0].regi ype expect [contains][thirdType] but found [internal]... expe foun...	0.04	2	1.24%
>	其他	-1.00	1	0.62%

向量知识库的构建和相似问题检索

为了更好的复用历史经验，避免历史相似问题重复分析，我们将人工标注的历史问题日志特征转化为嵌入存储在向量数据库中，然后遇到新的问题时优先进行相似性检索，置信度高的直接复用历史结论，置信度低的用于RAG增强LLM生成。

- **知识库知识定义**

专家规则库

定义：根据历史经验总结出来的规则，特点是规则与问题根因一一对应。

用途：可以根据报错日志的内容直接进行规则匹配，匹配成功的即命中规则。

历史问题库

定义：经人工标注的单个测试脚本历史执行失败日志特征向量库。

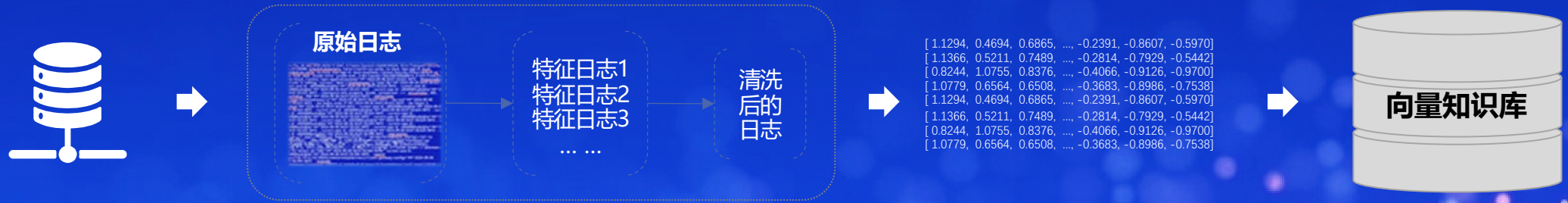
用途：测试脚本失败优先检索该脚本历史执行失败的日志并召回相似度高的Top历史记录。

典型问题库

定义：经人工标注的典型测试脚本失败日志特征向量库。

用途：典型问题库存储的是具有共性失败特征的日志向量，用于所有测试脚本失败相似性检索。

- **向量知识库入库流程**



存量失败脚本日志
人工标注结果

日志特征要素提取和日志清洗

Embedding

Vector DB

向量知识库的构建和相似问题检索

向量索引和距离：

- 索引构建与检索算法

  FLAT、HNSW、IVF系列

- 向量间距离度量方法

  a. 内积（IP）

  b. 欧式距离（L2）

  c. 余弦相似度（COSINE）

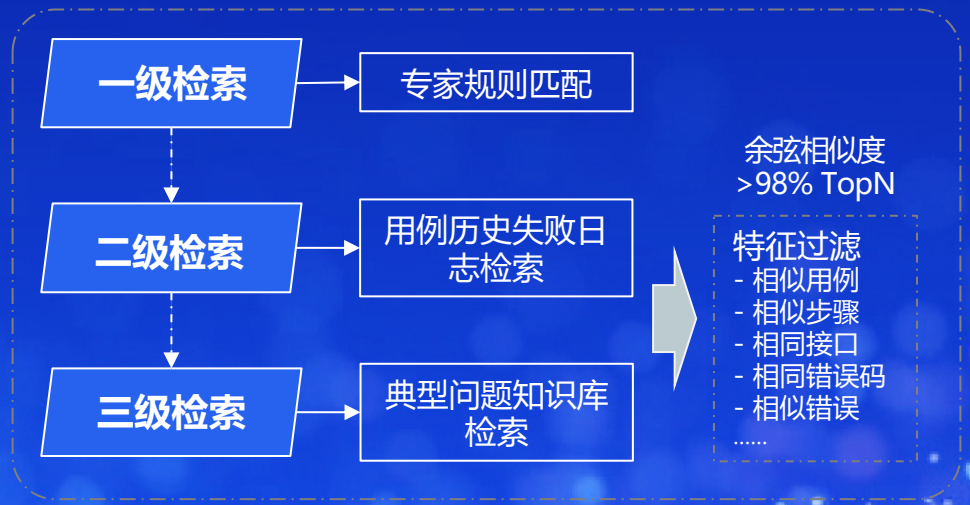
常用索引算法对比：

索引类型	算法说明	向量规模	召回率	检索速度	备注
FLAT	暴力检索，召回率100%。	10万以内	100%	慢	适用于小型数据集或查询时间无关的情况
HNSW	一种基于图的近似最近邻搜索算法，具有超快的搜索速度和出色的召回率。	10万-1亿	95%	非常快	质量高，速度快，但内存使用量大
IVF系列	基于倒排索引和向量量化的高效近似最近邻搜索算法。	1亿以上	95%	快	良好的可扩展选项。高质量，合理的速度和内存使用

相似问题检索流程

测试脚本执行日志 -> **特征要素提取和日志清洗**

相似检索、过滤与倒排



检索结果

相似脚本名称/编号

相似脚本失败日志

相似失败脚本问题分类

相似失败脚本根因描述

相似失败脚本问题解决措施

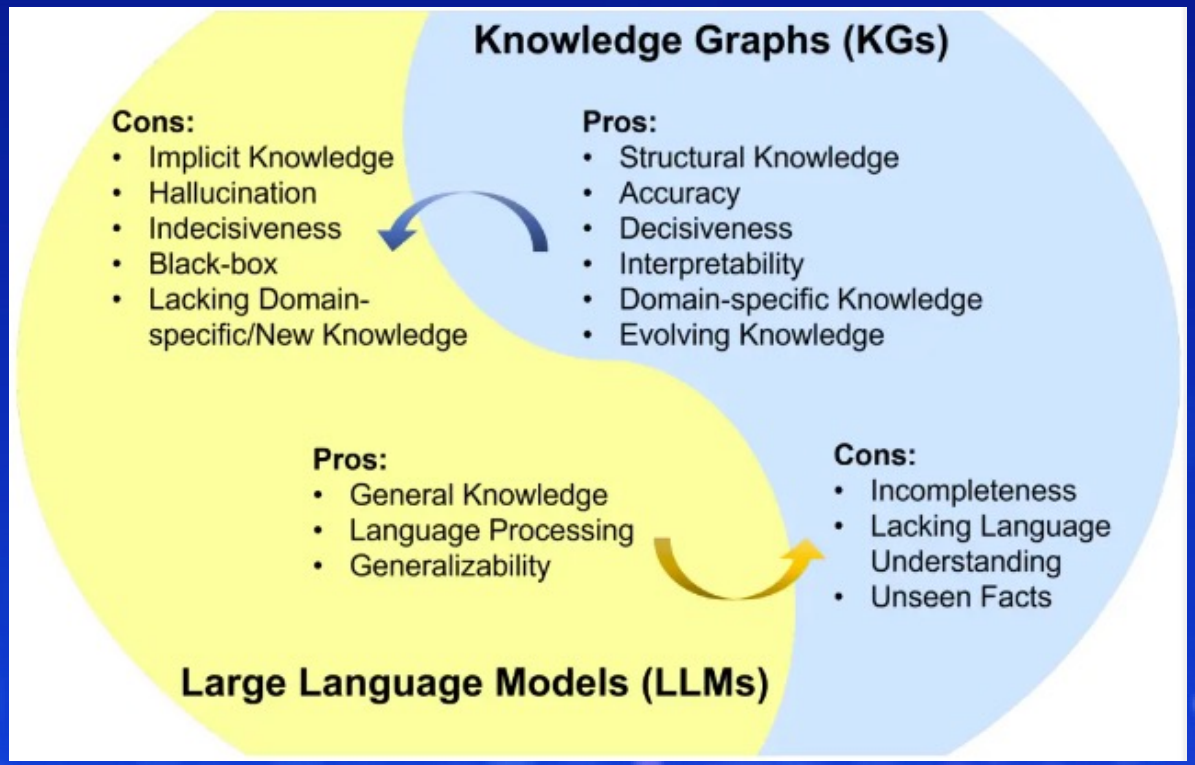
通过向量知识库的应用，可以帮助我们在测试脚本失败分析时直接复用或参考已有的历史经验知识，避免历史相同/相似问题重复分析。

基于知识图谱增强 LLM 推理

当前问题：大模型对日志内容及上下文的理解能力比较擅长，但缺乏产品领域业务日志特征和问题根因关联知识，导致LLM日志分析定位定界存在幻觉和误判。

解决思路：将知识图谱信息融入到Prompt中，为LLM提供图上下文信息，从而增强大模型日志特征和问题根因因果关系推理结果的准确性和可解析性。

知识图谱（Knowledge Graph），是一种表示实体与实体之间复杂关系的结构化语义网络，随着AI应用的不断发展，知识图谱已广泛应用于智能搜索，智能问答、文本分析、异常监测、数据挖掘、个性化推荐等领域。



知识图谱增强LLM方案分析：

1、测试脚本失败根因影响因素多样化（环境问题、数据依赖、配置依赖、脚本问题、执行端问题、服务端缺陷等），单纯将测试日志输出给大模型分析很难准确地定位出问题的根本原因；

2、测试脚本失败日志和业务日志特征（比如错误码、错误信息、异常堆栈等）与问题根因之间存在一定的因果关系；

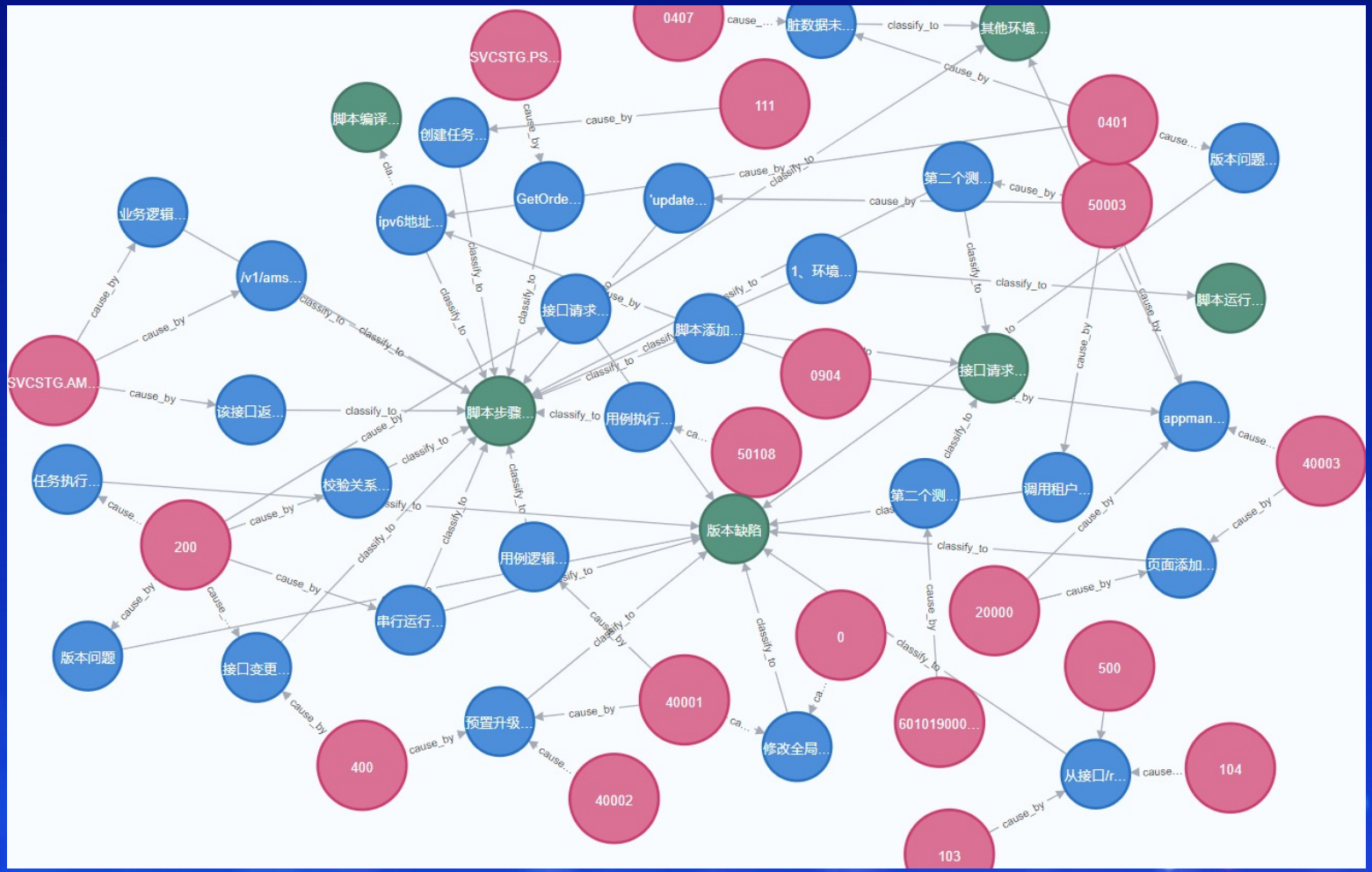
3、通过微调和上下文学习可以使能大模型更清晰的理解失败特征和问题根因之间的联系，可以避免或缓解大模型的幻觉，从而更准确地推理。

单纯的RAG技术无法针对关联性比较强的失败日志进行准确的分析，知识图谱增强将会成为解决这一类问题的关键。

基于知识图谱增强 LLM 推理

领域知识图谱：

- 测试脚本失败影响因素复杂多样化：错误码和错误描述与实际问题根因呈多对多的关系；
- 失败根因与问题分类对应关系：出于对失败脚本后置处理的需要进行问题分类，大模型由于缺少相应的先验知识容易分类错误。
- 问题的因果关系不断积累和演进，逐渐形成图结构化的领域知识网络。



基于知识图谱增强 LLM 推理

Part1、知识图谱生成：

- 从存量问题根因数据中提取日志特征相关实体（Entity）；
- 建立实体和问题根因之间的关系（Relation）生成知识图谱；

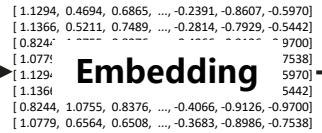
Failure Log Fragment :

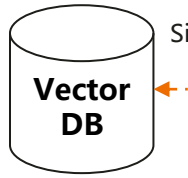
ResponseBody:
{"result":false,"errorCode":"6030***0114","desc":"The version (1.0) of the tenant privacy statement is incorrect","errorParams":{"version":"1.0"}}
[ERROR]-[https-jss*-nio*-fc16:0:0:0:0:0:0:68-18443-exec-2]-[be793a322b656639]-[]-[com.hu*wei.***.***.*.third.*.Third*20C*entService()] Refresh tokn is empty
Failure Cause: *{{Failure Cause of Manual Labeled}}*
Classification: *{{Problem Classification}}*

Entity

Part2、日志三元组链接：

- 日志特征实体抽取；
- 实体链接和子图检索；





Similar Entity Retriever

New Failure Log Content

Subgraph Retriever

Part3、知识图谱作为上下文提示增强LLM推理：

- 子图结构生成自然语言描述信息；
- 将子图信息融入到Prompt中，为LLM提供图上下文信息。

Subgraph Structures

Subgraph Structure as Prompt

Graph Context

Prompt: 我希望你扮演一名专业的软件工程师，我将给你一个测试脚本执行日志，您的任务是 ... *Task Description And Log Content ...* **[Graph Context]** *More In-Content Examples...*

Prompt

LLMs

基于知识图谱增强 LLM 推理

Standard Prompting

Prompt

Question： 我希望你扮演一名专业的软件工程师，我将给你一个测试脚本执行日志，您的任务是 ... *Task Description ...*

###

Request Url https://{HOST}/rest/v2/32b4a../a**s/metric/upload
ResponseBody
"errorCode":"SVCS*T*G.**MS.2000001","errorMessage":"Import fails because another file is being imported. Try again later."...

More Log Content ...

###

Few In-Context Examples...

Model Output

Failure Cause：根据日志分析，该测试脚本执行过程中upload接口执行失败，预期的errorMessage与实际返回的错误信息不符。预期错误信息提到'指标层级Zh字段不合法'，但实际返回的是'导入失败，有文件正在导入中，请稍后重试'。这表明可能是接口处理逻辑错误，或者业务规则对于'metricName_zh'字段的验证不正确，或者在上传数据时有并发处理导致的暂时性冲突。

Classification：版本BUG-服务端处理错误 ✗

Knowledge Augmented LLM Prompting

Prompt

Question： 我希望你扮演一名专业的软件工程师，我将给你一个测试脚本执行日志，您的任务是 ... *Task Description ...*

###

Request Url https://{HOST}/rest/v2/32b4a../a**s/metric/upload
ResponseBody
"errorCode":"SVCS*T*G.**MS.2000001","errorMessage":"Import fails because another file is being imported. Try again later."...

More Log Content ...

###

以下是与日志特征相关的问题根因和问题分类信息，请参考来完成你的任务：

[Graph Context] ◄

Few In-Context Examples...

Model Output

Failure Cause：根据所提供的错误日志分析，导入文件upload接口执行失败，问题的根本原因是接口在处理数据上传时遇到了并发问题或者处理逻辑错误。由于实际返回的错误信息提示“导入失败，有文件正在导入中，请稍后重试”，这表明数据上传操作可能因为其他并发操作而被阻塞，或者服务端的处理逻辑对于并发请求的管理存在问题。

Classification：用例运行时问题-用例并行冲突 ✓

知识图谱增强LLM失败脚本根因分析

对于新的失败日志，采用大语言模型结合领域知道对日志进行分析，提取日志特征实体检索相关知识图谱子图结构，并采用Few-shot多示例上下文学习增强的方式进行大模型失败根因推理和问题分类推荐。

【方案思路】对于全新失败脚本，采用LLM+RAG+KGs & Few-shot Prompting上下文学习的方式进行大模型失败根因推理。

1、首先采用大语言模型+RAG对失败用例日志进行分析和总结；

2、然后，从测试日志中提取失败特征相关实体Entity；

3、根据实体Entity，从图数据库中查询子图结构体并转化为自然语言描述，作为LLMs Prompt提示图上下文信息；

4、使用Few-shot多示例上下文学习的方式拼接最终的Prompt请求，输入给大模型进行失败根因推理和问题分类推荐;

【总结】通过此方式，避免了微调模型大量的人工语料投入和训练成本，以及微调后模型性能不确定性问题。并且通过知识图谱上下文，增强大模型对问题根因因果关系的理解，增强大模型推理结果的准确性和可解析性。

参考论文

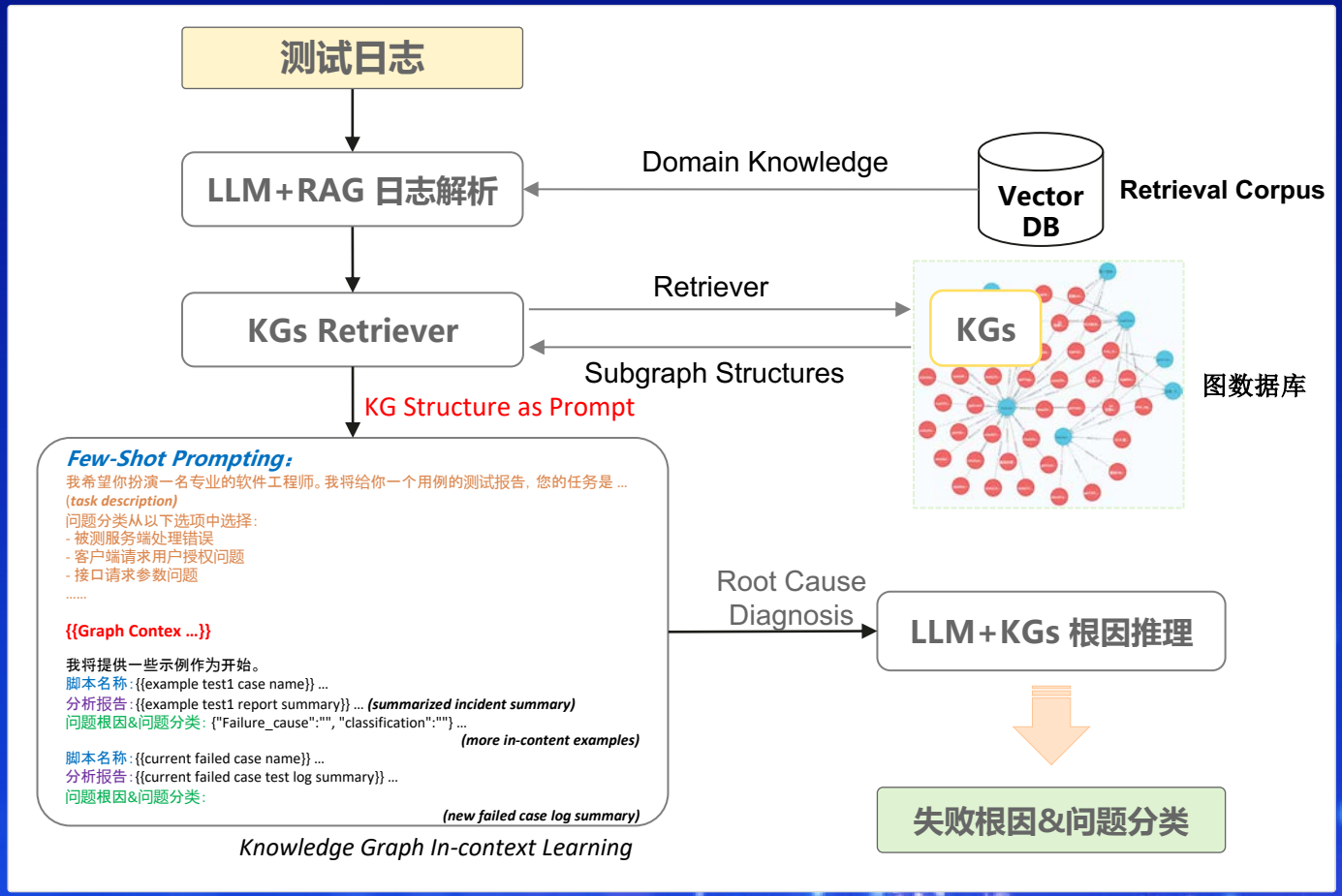
1.《Automated Root Causing of Cloud Incidents using incontext learning via gpt4》

2.《Knowledge Graph Structure as Prompt: Improving Small Language Models Capabilities for Knowledge-based Causal Discovery 》

3.《Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering》

4.《Unifying Large Language Models and Knowledge Graphs: A Roadmap》

5.《 Knowledgeable Preference Alignment for LLMs in Domain-specific Question Answering》



Knowledge Graph In-context Learning

提示工程，与模型沟通的艺术

- **拟人原则**

  将大模型想象成是一个人，与人的沟通技巧同样适用于大模型。

- **沟通原则**

  a. 假设性提问

  b. 问题聚焦

  c. 表述清晰

  d. 背景完善

- **知识边界**

  如果大模型无法理解或回答垂直领域的知识或问题，需要补充Prompt上下文信息，比如接口的定义、日志中专业术语的解释等

LLM+RAG 日志解析 Prompting

作为一位软件自动化测试领域的专家，您的任务是根据以下测试脚本日志内容进行解析并总结，重点关注日志中的以下几个方面：

1、导致用例失败的步骤信息，如接口URI；

2、失败步骤接口请求所依赖的参数以及上下文信息；

3、导致用例失败的特征，例如错误描述、检查点、接口返回内容中的细节等，预期是什么，实际发生了什么；

4、每行日志以INFO - TraceLog开头的是最近一次接口请求的结果日志，如果有错误，错误信息会在这个日志中显示。

###执行日志开始###

{{Test_Script_Log_Content}}

###执行日志结束###

参考以下信息回答：

###

Few In-Context About Domain Info…

###

您的摘要最多应该有5-6句话且不分行，并且应该是第三人称，总长度不超过2000字。您必须以<|endoftext|>结束摘要。

LLM+KGs 根因推理 Prompting

作为一位软件自动化测试领域的专家，正在对自动化执行失败的脚本进行问题根因分析。请根据[执行日志摘要]和[可能与回答这个问题相关的事实]对问题根因进行归类，并总结出最可能的问题根因。

可选问题分类如下：

{{classification_list}}

可能与回答这个问题相关的事实：

{{Graph_Contex}}

###执行日志摘要开始###

{{Test_Log_Summary}}

###执行日志摘要结束###

请结合[执行日志摘要]和[可能与回答这个问题相关的事实]，帮我对失败的用例进行分析给出最可能的问题根因和问题分类。

按照如下的JSON格式输出：

{“failure_cause":"这里只写问题根因(4-5句话且不换行，2000字以内)","classification":"这里写问题分类(只选择一个)"}

PART 05

总结与展望

总结

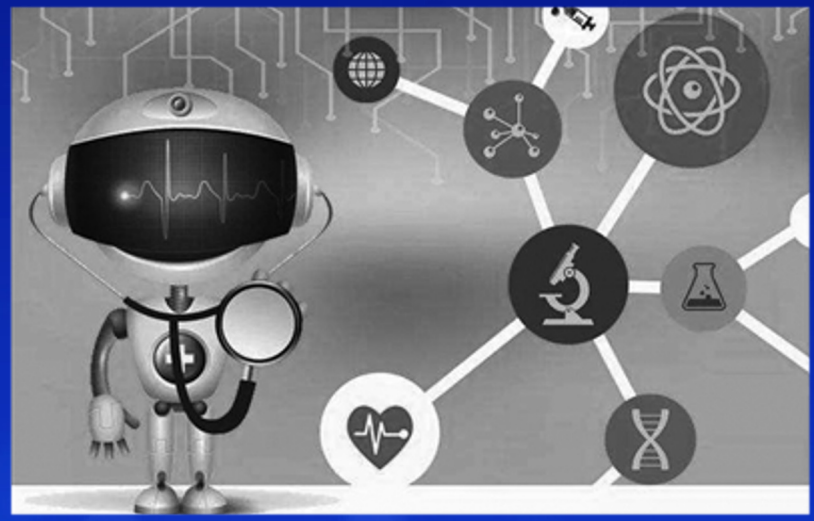
内容回顾和总结：

1. 特征聚类：通过日志特征聚类去重可以大幅降低失败测试脚本分析数量
2. 语义相似度检索：通过文本向量化和语义相似度检索技术可以有效的复用历史经验
3. KGs + LLM：知识图谱在增强大模型日志根因诊断方面有比较好的应用效果

展望

大模型与知识图谱协同或是未来AI应用的发展趋势?

1. LLM辅助知识图谱生成和治理
2. 知识图谱用于检测/评估大模型的幻觉
3. 知识图谱用于AI Agent的Planning的生成
4. 知识图谱用于大模型增量训练与微调提升领域推理能力



科技生态圈峰会 + 深度研习

——1000 + 技术团队的共同选择

KEYLINK ing

K+峰会

K+峰会 · 敦煌站
K+ 思考周®研习社
时间：2025.08.29-30

K+峰会 · 上海站
K+ 金融专场
时间：2025.10.17-18

K+峰会 · 香港站
K+ 思考周®研习社
时间：2025.11.25-26

K+峰会详情

AiDD峰会

AiDD峰会 · 上海站
AI+研发数字峰会
时间：2025.05.17-18

AiDD峰会 · 北京站
AI+研发数字峰会
时间：2025.08.08-09

AiDD峰会 · 深圳站
AI+研发数字峰会
时间：2025.11.28-29

AiDD峰会详情

AiDD AI+研发数字峰会 第5届
AI+ Development Digital summit

利用AI技术深化计算机对现实世界的理解

推动研发进入智能化时代