



2023 CCF国际AIOps挑战赛决赛  
暨“大模型时代的AIOps”研讨会

# 日志大模型LogFM: 大语言模型在ICT 运维领域的研究及应用

陶仕敏-华为-2012文本机器翻译实验室技术专家

主办单位: 中国计算机学会 (CCF)、清华大学、中国建设银行股份有限公司、南开大学

承办单位: 中国计算机学会互联网专委会、清华大学计算机科学与技术系、中国建设银行股份有限公司运营数据中心、南开大学软件学院、北京必示科技有限公司

赞助单位: 华为技术有限公司、国网宁夏电力有限公司电力科学研究院、软通动力信息技术(集团)股份有限公司

# 团队介绍：华为文本机器翻译实验室

## 四项第一

华为再获佳绩

IWSLT 2022  
国际口语机器翻译比赛

IWSLT 2022

International Conference on Spoken Language Translation

华为翻译（HW-TSC）在本次比赛中获得：

- 01 语音到语音翻译英德方向**第一**
- 02 离线语音翻译英日方向**第一**
- 03 等长口语翻译英法方向**第一**
- 04 等长口语翻译英西方向**第一**

### 华为翻译中心3篇论文入选自然语言处理领域顶级会议ACL 2023

HWTSC HWTSC 2023-05-09 18:29



2023年5月2日，自然语言处理领域顶级会议ACL 2023录用结果公布，华为翻译中心共3篇论文入选，其中论文“Text Style Transfer Back-Translation”被主会议录用，论文“Lexical Translation Inconsistency-Aware Document-Level Translation Repair”被子刊Findings录用，论文“Collective Human Opinions in Semantic Textual Similarity”被期刊TACL录用并邀请至主会分享。

## 办公IT场景

 HUAWEI Translate	翻译平台
 WeLink	即时通讯翻译
 IMS	视频字幕翻译
 iSales+	销售美文档翻译
 服务工具市场 Service Tool Market	GTS服务工具翻译
 Linstar Innovation Platform	公司创新平台内容翻译
 IntelliJ IDEA	Java - IDE代码注释翻译

## 2C/2D场景

	浏览器翻译，领域：新闻
	应用市场翻译，领域：应用/游戏
	华为视频翻译，领域：字幕
	华为快译，领域：通用
	HiAI翻译，领域：出行、新闻...



# 目录

# CONTENTS

第一章节 软件日志运维观点

第二章节 AIOps痛点与挑战

第三章节 大模型时代的AIOps应用探索

第四章节 未来畅想



2023 CCF国际AIOps挑战赛决赛  
暨“大模型时代的AIOps”研讨会

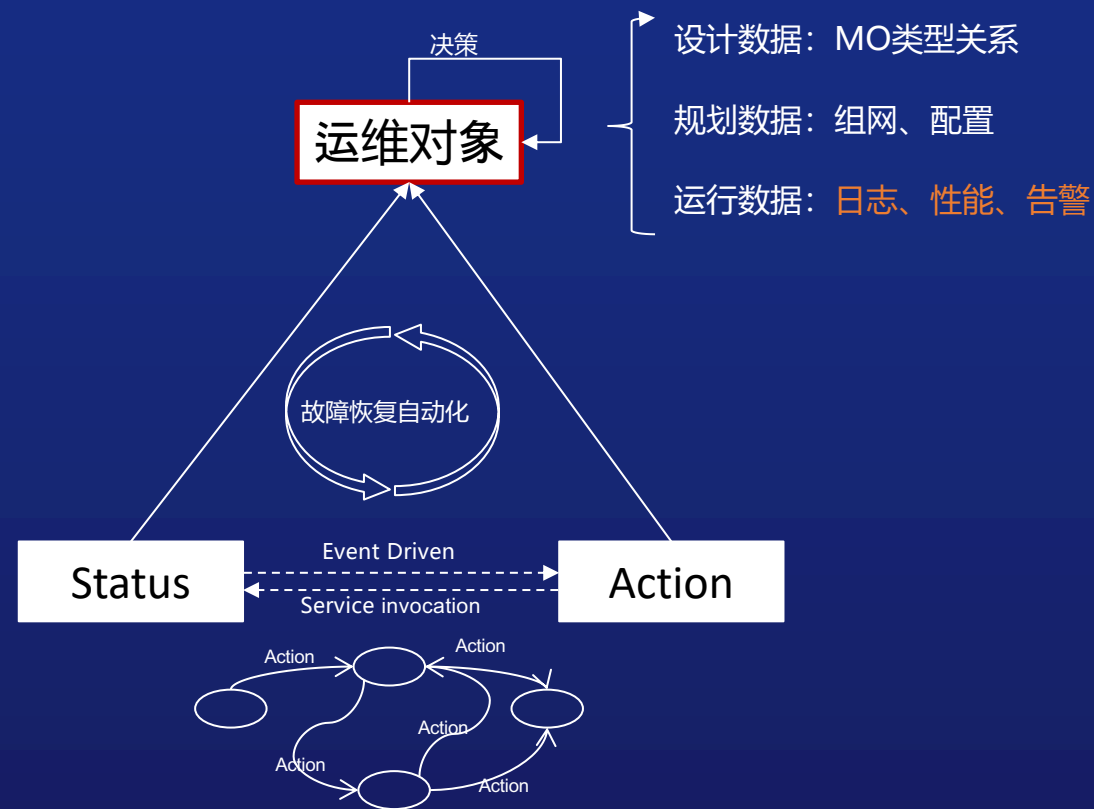
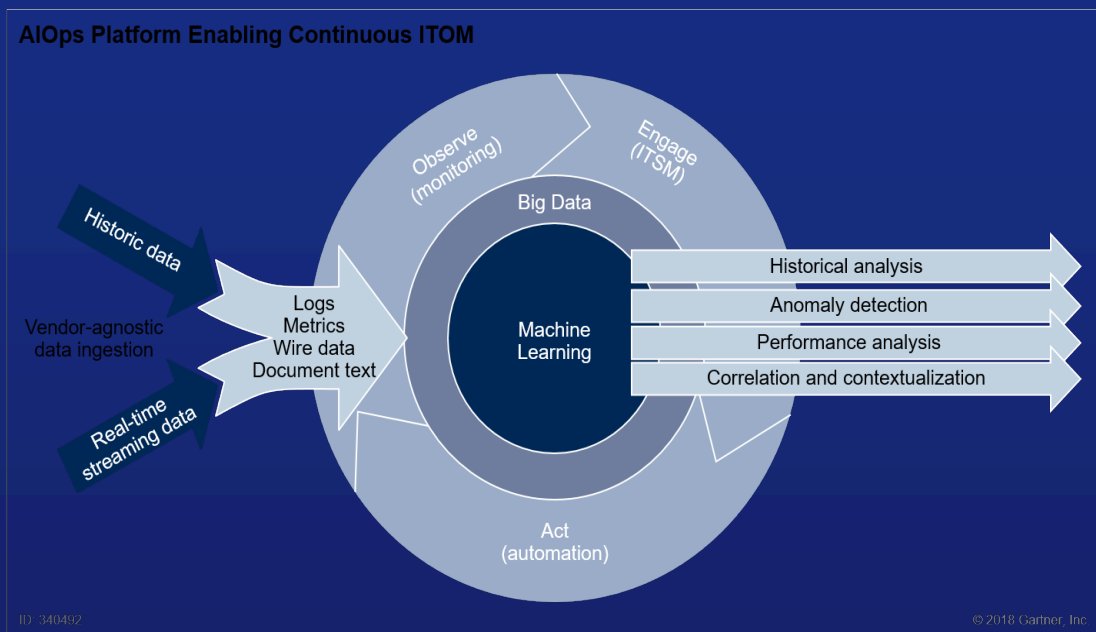
## 第一章节

# 软件日志运维观点

# ICT运维场景及目标：基于数据、算法和场景驱动的智能运维



- 场景及目标：围绕运维对象，发现其与预期不一致的状态，通过事先定义的Action，驱动运维对象状态的恢复。
- 数据：日志、KPI、告警等。
  1. KPI是统计数据，对于服务的本身状态是一种成本开销上的折中。
  2. 告警本质是把机器状态数据通过可接受的成本转化成自然语言，方便人理解。
  3. 日志某种程度上可以看做是机器语言，更加全面。在大模型时代更适合作为模型的语料。



# 大语言模型为什么可以应用到ICT运维领域

## 问题场景

模板提取

异常检测

问题定位

服务恢复

- 日志是一种类自然语言的文本，似乎可以通过LLM来加强对日志文本的理解

数据来源	时间戳	详细信息	类自然语言半结构文本
交换机	Jun 12 19:03:27	Interface te-1/1/59, changed state to down	
路由器	Jun 13 20:22:03	GigabitEthernet1/0/18 link status is DOWN	
分布式文件系统	Jun 12 19:03:27	PacketResponder 1 for block blk_1608 terminating	
高性能计算集群	Jun 12 19:03:27	Component is in the unavailable state(HWID=4432)	

表：一些网络基础设施中的日志消息，日志中的详细信息和自然语言有一定的相似性

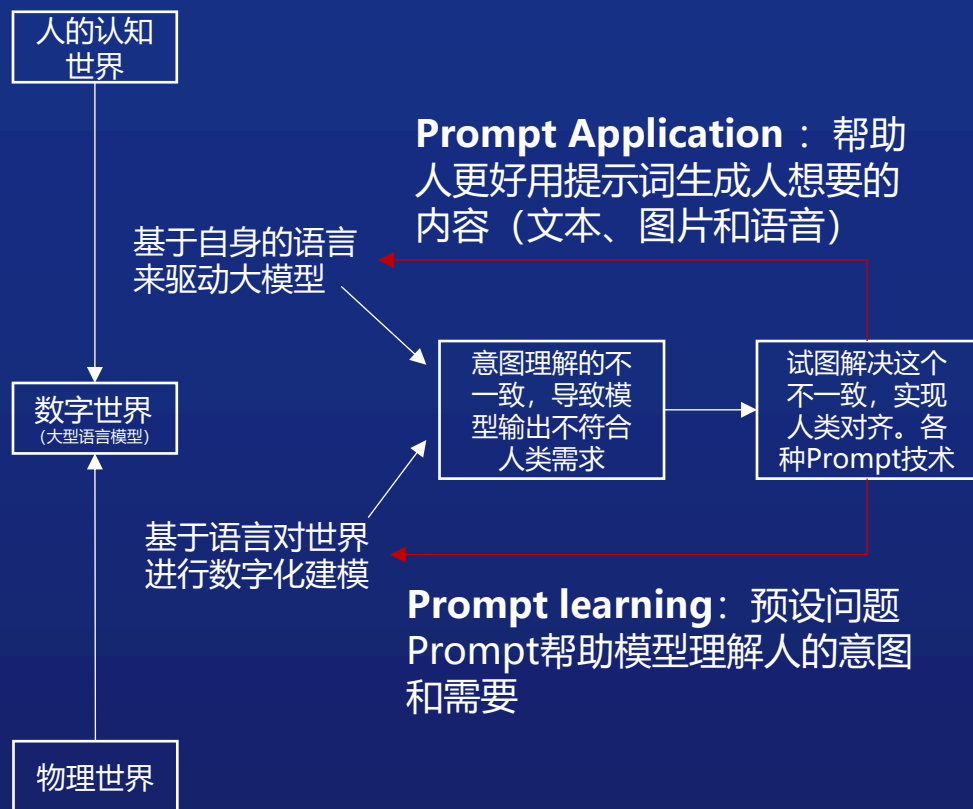
- 通过预训练或者指令学习，可以适应多任务场景，从而可能建立处理框架



图：基于日志的自动化分析任务总结  
(<https://github.com/logpai/awesome-log-analysis>)

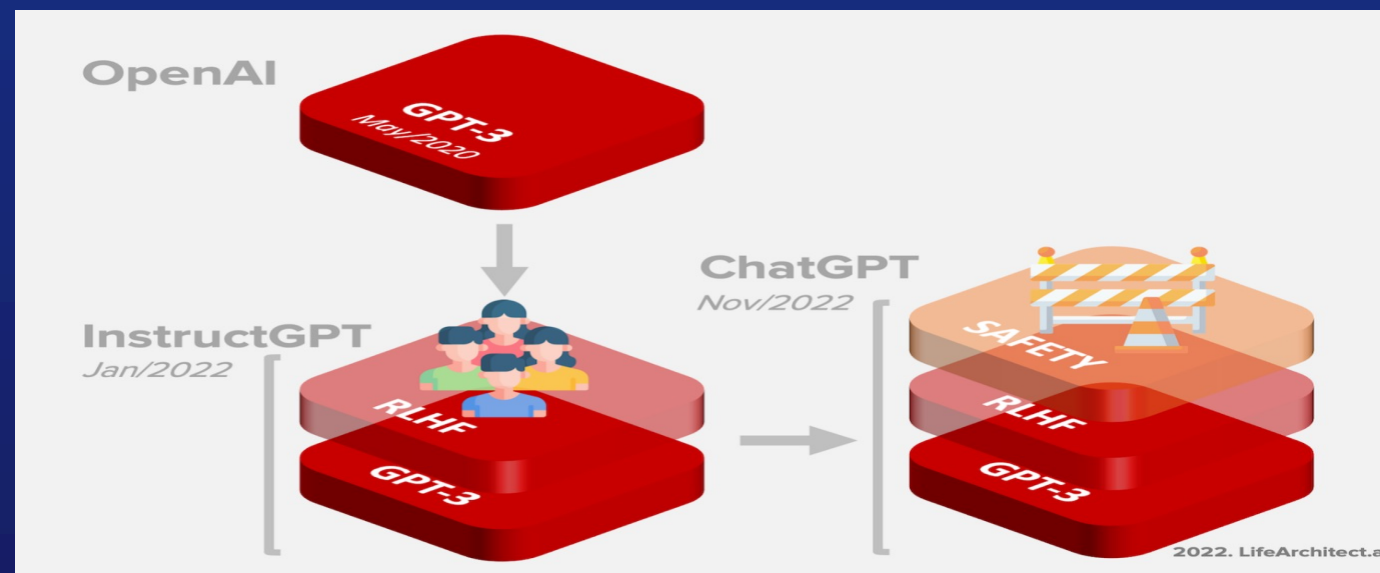
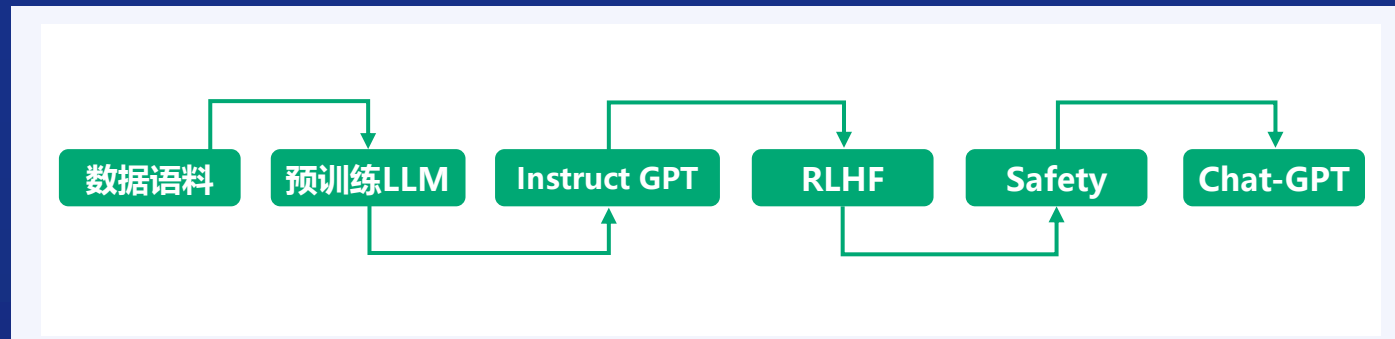
# 观点：人类语言是模型理解世界的桥梁，与人类对齐成就更好的模型效果和体验

## 大语言模型 + Prompt 的架构模型



注：仅代表一种理解和可能

## 5个对齐成就更好的LLM产品



from: <https://lifearchitect.ai/chatgpt/>

# 观点：智能运维演进趋势：从任务数据驱动到自适应运维智慧体

代际	输入	方法	目标	研究成果	类别
第一代	离散特征和KPI	特征识别及统计算法	拟合异常结果	Ft-tree LogParse	任务数据驱动
第二代	日志文本生成token	深度学习	拟合异常结果	LogAnomaly LogStamp	
第三代	段落日志和跨域日志	预训练语言模型	日志语言理解	BigLog Da-Parser	
第四代	原始日志和自然语言文本	大语言模型	可解释性运维	LogPrompt	指令驱动
第五代	自适应运维智慧体：目标自适应、领域自适应、强交互性、可执行性。。。				

表：LogAIBox研究项目代际演进思路

[1]LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs (IJCAI 2019)

[2]LogParse: Making Log Parsing Adaptive through Word Classification. (ICCCN 2020)

[3]LogStamp: Automatic Online Log Parsing Based on Sequence Labelling. (WAIN Performance 2021)

[4]BigLog:Unsupervised Large-scale Pre-training for a Unified Log Representation(IWQoS 2023)

[5]DA-Parser: A Pre-trained Domain-Aware Parsing Framework for Heterogeneous Log Analysis(COMPSAC 2023)

[6]LogPrompt Under Review.

团队repo地址：<https://github.com/LogAIBox>



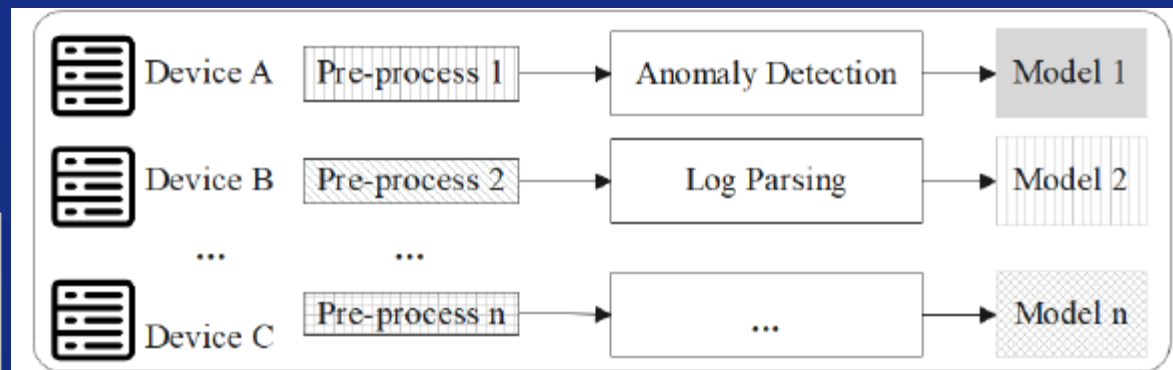
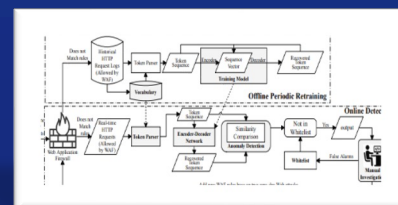
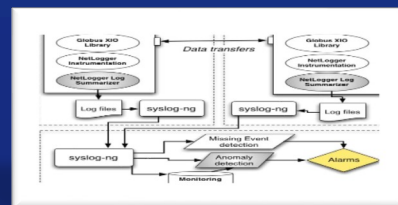
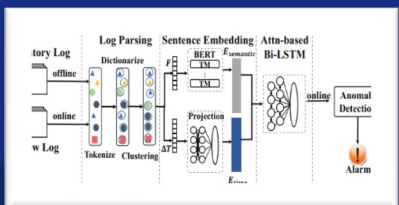
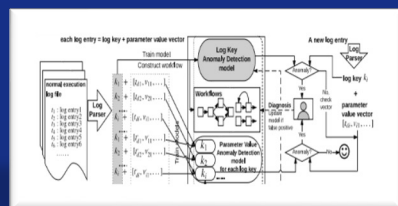
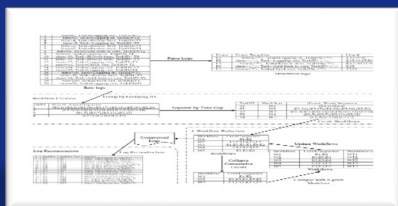
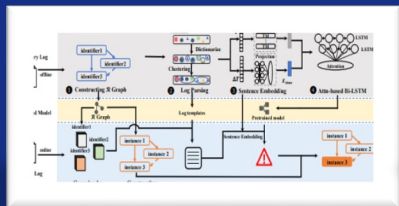


2023 CCF国际AIOps挑战赛决赛  
暨“大模型时代的AIOps”研讨会

## 第二章节

# AIOps痛点与挑战

# 痛点1：传统运维系统中定制化严重、各自为战，缺乏统一框架

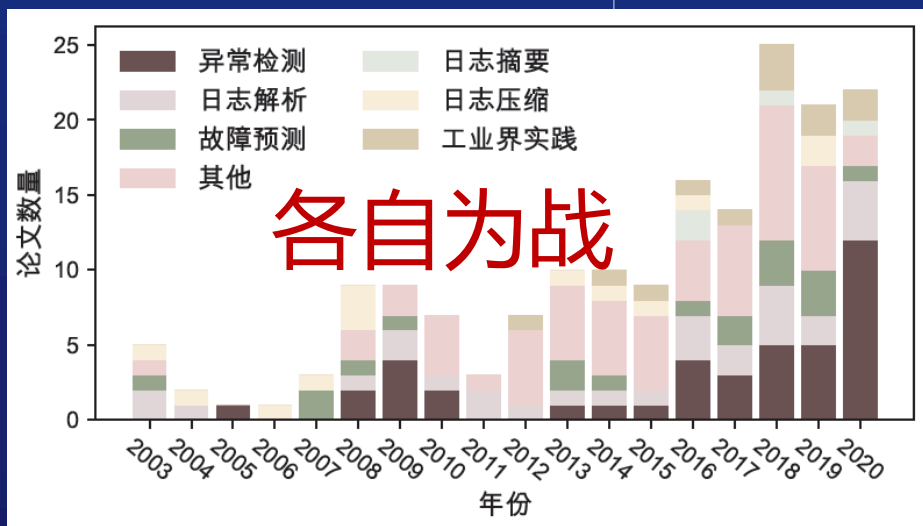


基于日志的异常检测

日志摘要

日志分类

日志分析系统现状

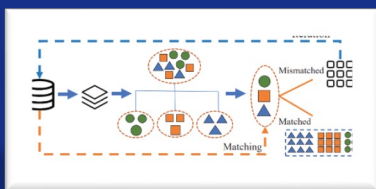


## 目前相关技术的缺点：

- 1、缺少统一的模型：现阶段针对不同任务需要不同的算法，维护成本高，需要统一的大模型涵盖所有的日志分析需求，实现多业务场景融合
- 2、处理流程定制化：不同的算法对日志的预处理、后处理不同，开发成本高。
- 3、日志特征提取定制化：不同方法提取日志特征向量依赖于手工特征或者定制模板，泛化能力不够，需要统一的日志特征提取方式。

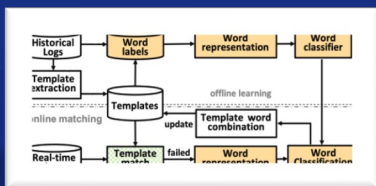
# 痛点2：传统日志分析方法中缺少对日志语义特征的深度挖掘

## 传统日志表征方法



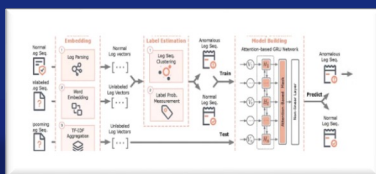
PLELog、LogRobust

基于通用词向量的日志特征提取：无法理解日志内的术语/特殊词



LogAnomaly

基于私有域数据训练日志特征：对于新领域的迁移性较差



PreFix

基于专家规则的日志特征：人力消耗太高，不具有可操作性

```
[Jun 09 19:23:31 2005] [error] [client 81.199.21.119] File does not exist: /var/www/html/sumthin

Step 1
Timestamp:Jun 09 19:23:31 2005
Content:File does not exist: /var/www/html/sumthin

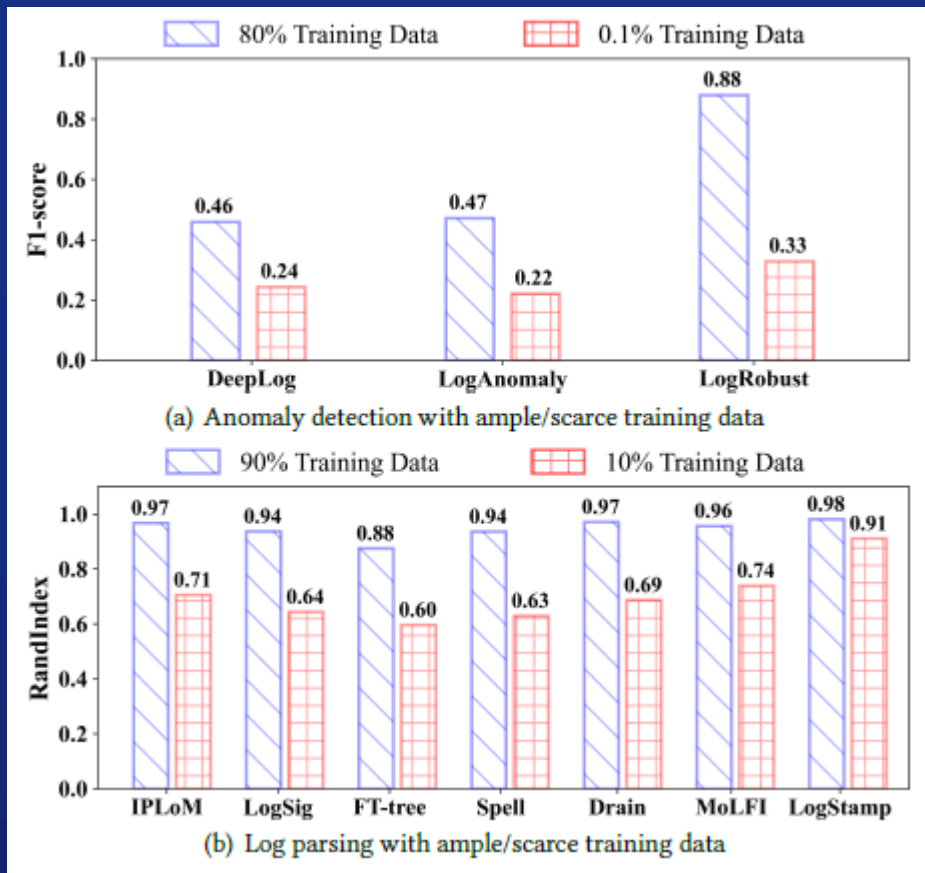
Step 2
File does not exist: [FILE] 非语义符号

Step 3
File does not exist: [FILE] 句间依赖
Directory index forbidden by rule: [FILE][FILE]

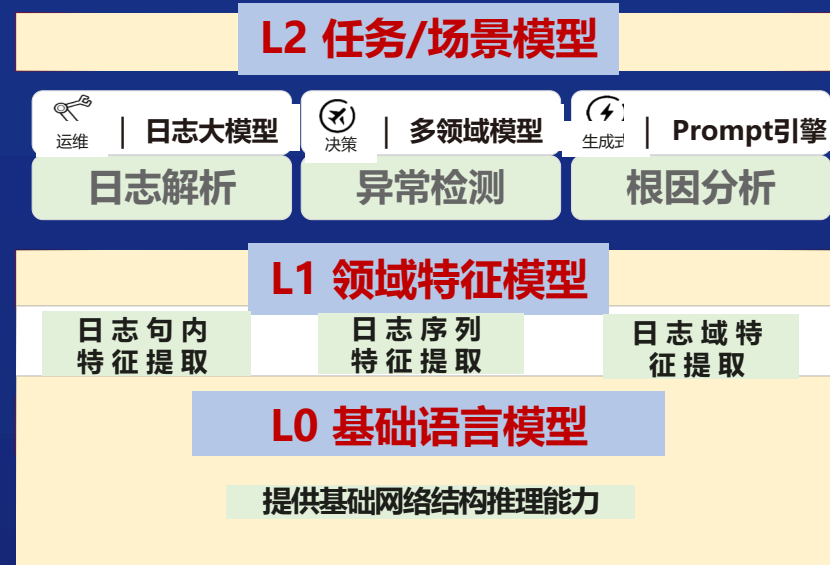
Step 4
File does not exist: [FILE] 领域术语
Directory index forbidden by rule: [FILE]
```

日志作为一种机器语言，有其特殊的语义特征，类比于NLP领域的词向量，也需要强泛化性的日志表征。

# 痛点3：传统智能运维算法依赖于任务数据，专家标注耗时耗力



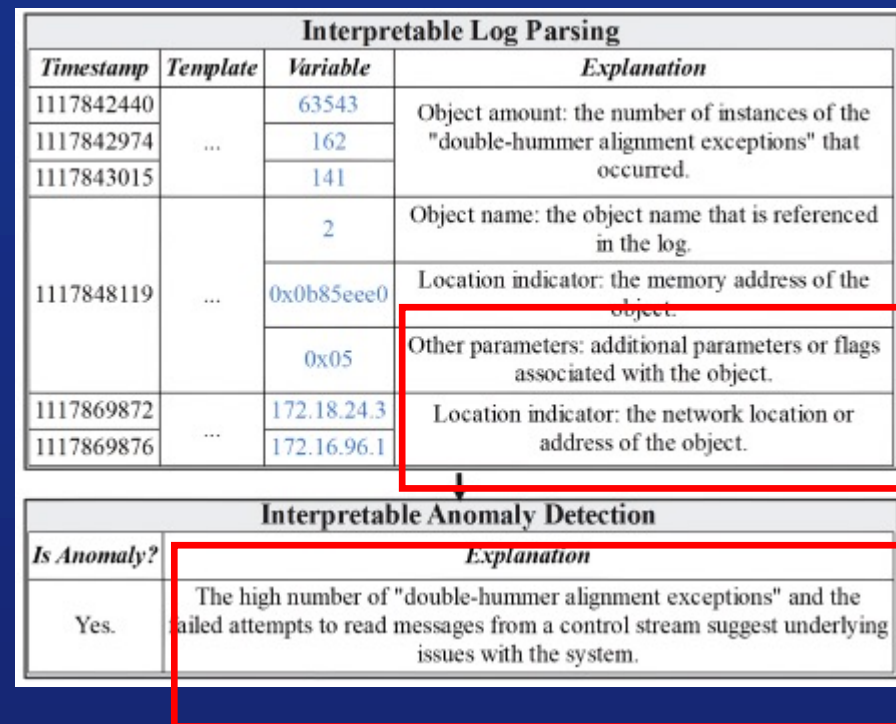
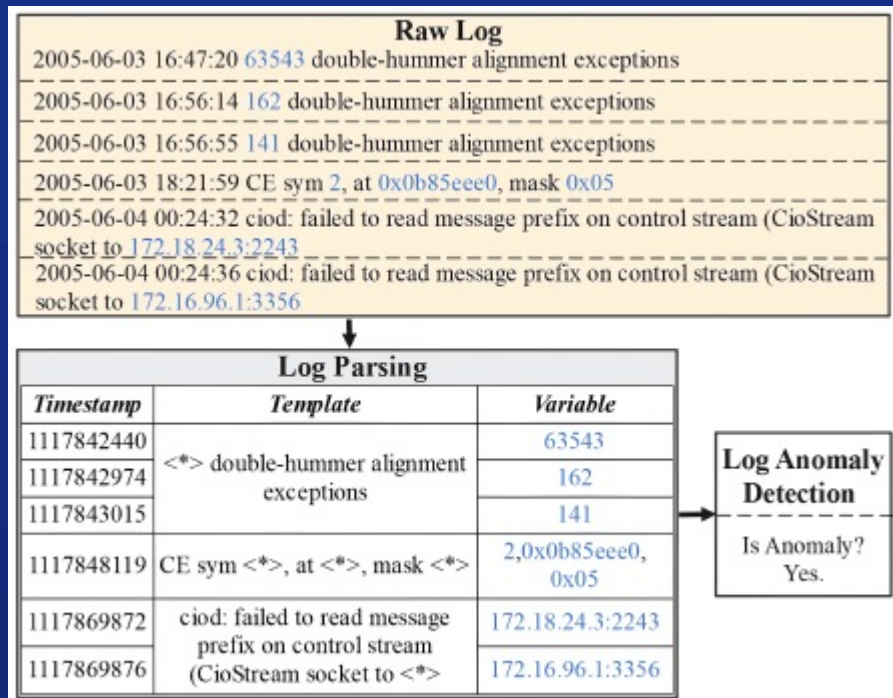
当任务训练数据减少时，传统方法普遍出现了预测精度下降。因此，要将其应用到私有系统中，必然需要大量标注数据。



底座模型对日志语义特征挖掘不足、忽略日志域特征，导致上层模型需要大量标注数据建模异常特征。



# 痛点4：传统运维系统可解释性差、可交互性弱



对异常日志生成了解释，可以快速判断虚报、漏报。

- (1) 传统日志分析算法只输出“告警/正常”，对于异常日志无反馈，需要专家阅读相关日志模板，人力整理生成分析报告，费时费力。
- (2) 只给出预测结果，对于报假警、漏报等情况不能很快地排除，需要结合原始日志进行分析排查。

根据本轮分析结果由大语言模型自动生成了分析报告，推荐解决方案。



2023 CCF国际AIOps挑战赛决赛  
暨“大模型时代的AIOps”研讨会

## 第三章节

# 大模型时代的AIOps应用探索

# 大模型 + 预训练 + Prompt驱动解决日志运维4大痛点

痛点1: 传统运维系统中定制化严重、各自为战, 缺乏统一框架

痛点2: 传统日志分析方法中缺少对日志语义特征的深度挖掘

痛点3: 传统智能运维算法依赖于任务数据, 专家标注耗时耗力

痛点4: 传统运维系统可解释性差、可交互性弱

**Biglog: 基于大规模日志预训练理解日志机器语言、构建日志多任务统一框架**

**LogPrompt: 基于日志Prompt策略驱动强语言模型构建零样本日志分析框架**

# Biglog: 基于大规模日志预训练理解日志机器语言、构建日志多任务统一框架



## 日志解析任务

7个公开数据集上取得19项最佳表现

细粒度场景下超过当前最佳模型40.74%

## 日志异常检测

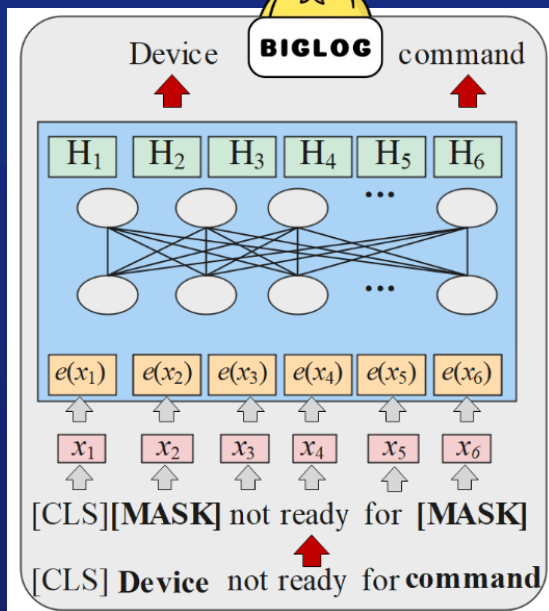
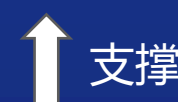
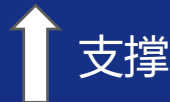
2个公开数据集上取得12项最佳表现

小样本场景下超过当前最佳模型40%以上

## 日志故障预测

3个公开数据集上取得6项最佳表现

迁移学习场景下超过当前最佳模型32.30%



## Biglog技术难点

### 通用语言模型BERT对日志理解能力不足

- 对来自新设备、新领域的日志理解能力不足
- 无法捕捉日志序列间的上下文依赖关系

### 原始日志内容存在大量噪声

- 大量非自然语言字符串 (127.0.0.1、/etc/cron.hourly)
- 不规范词、复合词 (SCHEDULE\_RECORDREMAINING)

### 通信领域日志中存在大量术语

- ospf、ftp、mac、ip

## 解决方案

### 预训练日志领域的语言模型

- 引入总共78G、来自16个域的日志作训练集
- 设计时间窗口式的日志序列拼接方法，让模型关注时间片关联

### 设计新的日志预处理流程

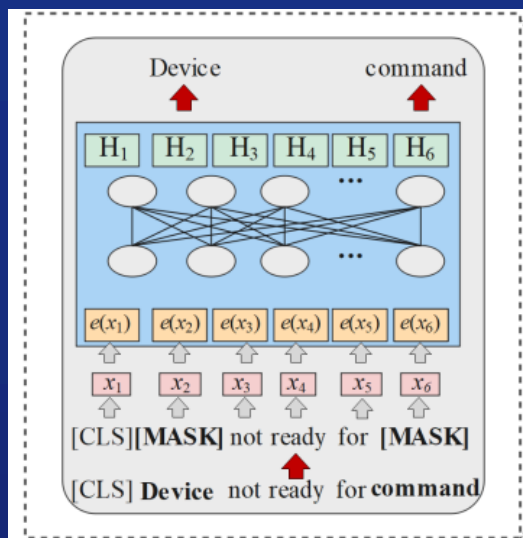
- 设计多种方法过滤不规范文本
- 利用[IP]、[FILE]、[CODE]等特殊符号替换原始串

### 利用数通术语增强模型预训练(V4)

- 挖掘原始日志语料中出现的高频术语
- 利用prompt、术语mask等方式增强预训练



# Biglog: 通过大规模预训练来理解日志语义特征, 生成通用日志向量表征



+

Dataset	# of messages
HDFS	82,293,702
Hadoop	394,308
Zookeeper	74,380
BGL	4,713,493
HPC	433,489
Linux	25,567
Proxifier	21,329
Spark	33,236,604
OpenStack	207,820
Thunderbird	211,212,192
Windows	114,608,388
Mac	117,283
Android	30,348,042
HealthApp	253,395
OpenSSH	655,146
Apache	56,481
Switch <sup>b</sup>	29,174,680

=

MLM Pre-training Design

78G logs from 16 domains

EXAMPLE OF LOG RECONSTRUCTED BY DIFFERENT LANGUAGE MODELS

Source	Log Text
Original <sup>a</sup>	( $T_1$ ):Connected to <u>watchd</u> service sent first pulse 16. ( $T_2$ ):Ibsm connect to <u>watchdog</u> manager. ( $T_3$ ):Connection was refused by host <u>10.155.101.233</u> on port 1 check that Lustre is running on that node 1.
Corrupted	Connected to [MASK] service sent first pulse 16. Ibsm connect to watchdog manager. Connection was [MASK] by host 10.155.101.233 on port 1 check that Lustre is [MASK] on that node 1.
Reconstructed (BERT)	Connected to <b>the</b> service sent first pulse 16. Ibsm connect to watchdog manager. Connection was <b>made</b> by host 10.155.101.233 on port 1 check that Lustre is <b>present</b> on that node 1.
Reconstructed (Biglog)	Connected to <b>watch</b> service sent first pulse 16. Ibsm connect to watchdog manager. Connection was <b>refused</b> by host 10.155.101.233 on port 1 check that Lustre is <b>running</b> on that node 1.

<sup>a</sup> watchd and watchdog is a pair of cross-sequence dependency. The original sequence consists of logs from three time bins ( $T_1$  to  $T_3$ ).

## ZERO-SHOT PERPLEXITY OF LANGUAGE MODELS IN UNSEEN DOMAINS

Language Model	Spirit	Switch
BERT [16]	240.28	2201.38
Biglog	<b>44.07</b>	<b>1292.88</b>

Outstanding Log Understanding Ability

# Biglog: 统一日志框架+ Biglog日志表征 在日志解析任务上的效果

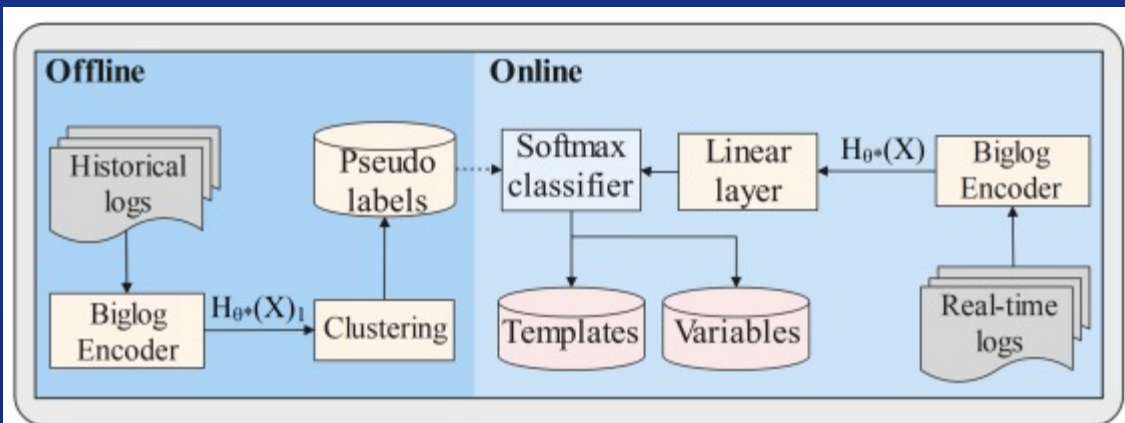


Fig. 4. Architecture of Biglog's module for log parsing. The linear layer is used to project log embeddings into the label domain and the Softmax layer converts scores into probabilities.

$$\begin{aligned} & \max_{\theta^*, W_1, b_1} \sum_{t=1}^T \log P(k(x_t) | X; \theta^*, W_1, b_1) \\ &= \sum_{t=1}^T \sum_{i=1}^2 \delta(k(x_t) - i) \log \frac{\exp(f_{W_1, b_1}(H_{\theta^*}(X)_t)_i)}{\sum_{j=1}^2 \exp(f_{W_1, b_1}(H_{\theta^*}(X)_t)_j)} \end{aligned}$$

FINE-LEVEL COMPARISON OF LOG PARSING USING F1-SCORE AS METRIC

Methods	HDFS	Hadoop	Zookeeper	BGL	HPC	Linux	Proxifier	Avg.
IPLoM	0.389	0.068	0.225	0.391	0.002	0.225	0.500	0.257
LKE	0.424	0.198	0.225	0.379	0.381	0.388	0.309	0.329
LogSig	0.344	0.050	0.225	0.333	0.002	0.146	0.339	0.205
FT-tree	0.385	0.046	0.186	0.497	0.002	0.211	0.420	0.249
Spell	0.000	0.058	0.045	0.536	0.000	0.091	0.000	0.104
Drain	0.389	0.068	0.225	0.397	0.002	0.225	0.500	0.258
MoLFI	0.000	0.095	0.000	0.333	0.000	0.026	0.000	0.065
LogParse	0.632	0.502	0.348	0.665	0.330	0.588	0.334	0.486
LogStamp	0.523	0.594	0.275	0.818	0.434	0.658	0.438	0.534
<b>Biglog</b>	<b>1.000</b>	<b>0.967</b>	<b>0.891</b>	<b>0.969</b>	<b>0.948</b>	<b>0.842</b>	<b>0.974</b>	<b>0.942</b>

RANDINDEX SCORES OF LOG PARSING IN THE ONLINE SCENARIO

Methods	HDFS	Hadoop	Zookeeper	BGL	HPC	Linux	Proxifier	Avg.
IPLoM	0.914	0.636	0.787	0.858	0.228	0.695	0.822	0.706
LKE	0.861	0.150	0.787	0.848	0.119	0.825	0.379	0.567
LogSig	0.872	0.651	0.787	0.806	0.119	0.715	0.559	0.644
FT-tree	0.908	0.668	0.773	0.275	0.119	0.709	0.722	0.596
Spell	0.871	0.721	0.102	0.503	0.882	0.706	0.621	0.630
Drain	0.914	0.647	0.787	0.822	0.119	0.695	0.822	0.687
MoLFI	0.871	0.699	0.899	0.792	0.881	0.410	0.621	0.739
LogParse	0.907	0.349	0.982	0.992	0.194	0.825	0.490	0.677
LogStamp	0.954	0.927	0.992	0.984	0.949	0.760	0.811	0.911
<b>Biglog</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>0.993</b>	<b>0.965</b>	<b>0.883</b>	<b>0.828</b>	<b>0.953</b>

# Biglog: 异常检测模块 + Biglog日志表征 在异常检测任务上的效果

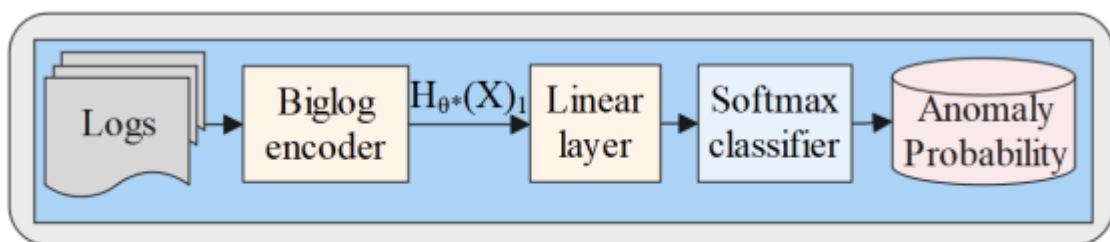
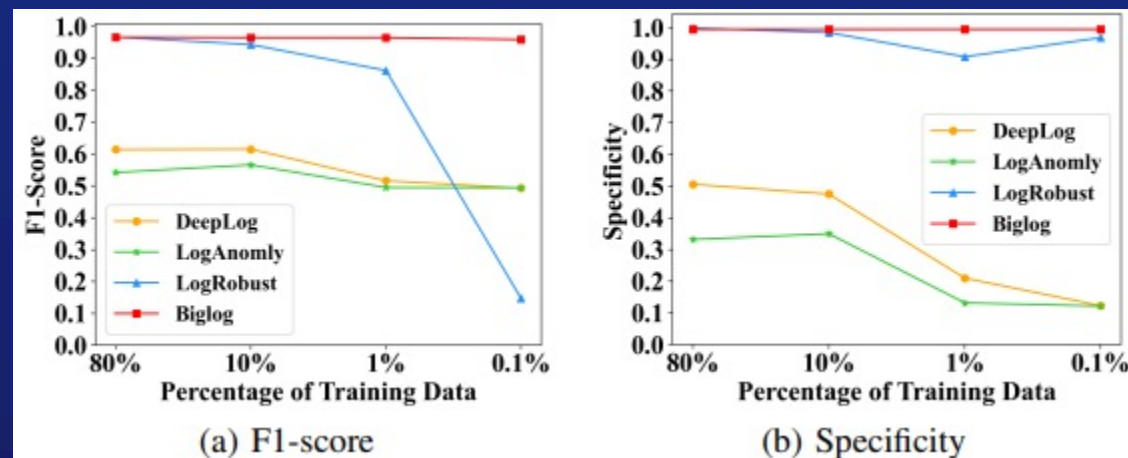


Fig. 5. Architecture of Biglog's module for anomaly detection. The Softmax layer outputs anomalous probability based on projected log embeddings.

Methods <sup>a</sup>		BGL			Spirit			Avg.
		Drain	Spell	IPLoM	Drain	Spell	IPLoM	
DeepLog	S	0.437	0.440	0.443	0.099	0.535	0.545	0.416
	F1	0.426	0.426	0.427	0.609	0.751	0.755	0.565
LogAnomaly	S	0.551	0.599	0.831	0.099	0.554	0.545	0.529
	F1	0.483	0.504	0.700	0.609	0.759	0.755	0.634
PLELog	S	0.899	0.938	0.952	0.690	0.901	0.535	0.819
	F1	0.744	0.476	0.521	0.841	0.859	0.570	0.668
LogRobust	S	<b>0.999</b>	0.963	0.922	0.990	0.960	0.980	0.969
	F1	<b>0.967</b>	0.914	0.833	0.949	0.973	0.986	0.936
CNN	S	0.970	0.999	0.986	0.990	0.970	0.990	0.985
	F1	0.908	0.979	0.948	0.993	0.979	0.993	0.984
Biglog	S	0.995	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>0.999</b>
	F1	0.966	<b>0.992</b>	<b>0.996</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>0.992</b>

$$\begin{aligned} & \max_{\theta^*, W_2, b_2} \log P(q_i | \tilde{X}_i; \theta^*, W_2, b_2) \\ &= \sum_{j=1}^2 \delta(q_i - j) \log \frac{\exp(f_{W_2, b_2}(H_{\theta^*}(\tilde{X}_i)_1)_j)}{\sum_{k=1}^2 \exp(f_{W_2, b_2}(H_{\theta^*}(\tilde{X}_i)_1)_k)} \end{aligned}$$

## Offline



## Online

# Biglog: 故障预测 workflow + Biglog 日志表征 在长序列故障预测任务上的效果

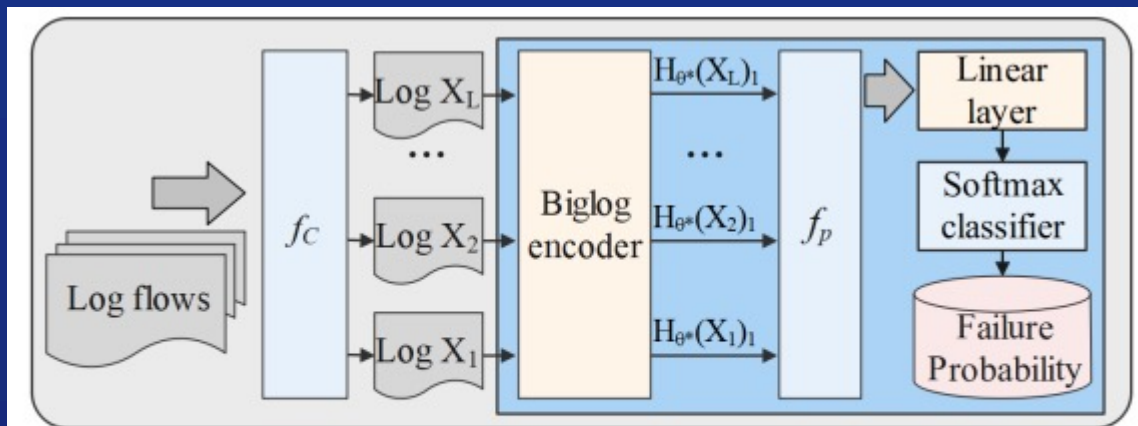


Fig. 6. Architecture of Biglog's module for failure prediction.  $f_C$  denotes the sequence construction function and  $f_P$  is the pooling layer. The future failure is predicted using the pooled embeddings of all logs in a sequence.

EXPERIMENTAL RESULTS OF LOG-BASED FAILURE PREDICTION

Methods	Switch-M1			Switch-M2			Switch-M3		
	P <sup>a</sup>	R	F1	P	R	F1	P	R	F1
HSMM	0.323	0.953	0.482	0.003	0.606	0.006	0.263	0.111	0.156
SKSVM	0.083	0.761	0.149	0.045	0.087	0.059	0.008	0.919	0.016
PreFix	<b>0.874</b>	0.744	0.803	0.598	0.586	0.592	0.840	0.525	0.646
UniLog	0.843	0.796	0.819	<b>0.657</b>	0.632	0.645	<b>0.879</b>	0.646	0.745
<b>Biglog<sup>b</sup></b>	0.743	<b>0.955</b>	<b>0.835</b>	0.587	<b>0.737</b>	<b>0.654</b>	0.852	<b>0.952</b>	<b>0.900</b>

PERFORMANCES OF BIGLOG WITH DIFFERENT STRATEGIES<sup>a</sup>

$f_C$	$f_P$	Precision	Recall	F1
Sampling	LSTM	0.844	<b>0.952</b>	0.895
Sampling	Max pooling	0.852	<b>0.952</b>	<b>0.900</b>
Window	LSTM	0.867	0.857	0.862
Window	Max pooling	<b>0.871</b>	0.857	0.864

<sup>a</sup> The experiment is done on Switch-M3.



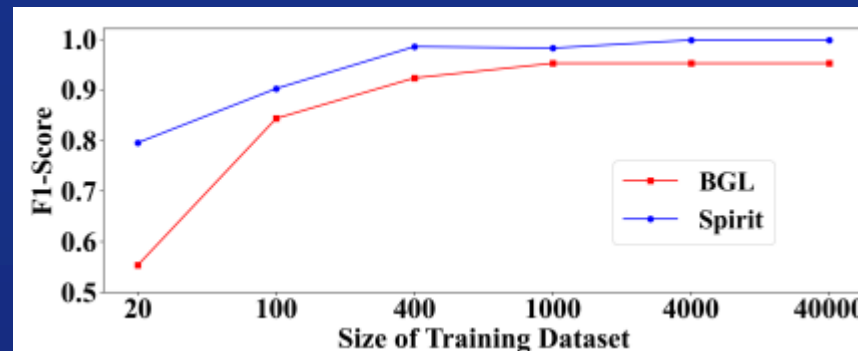
# Biglog: 预训练提升了模型的泛化能力, 减少任务数据依赖

COMPARISON WITH LOGTRANSFER IN DOMAIN ADAPTATION SCENARIO

Methods	Precision	Recall	Specificity	F1
LogTransfer [15]	0.793	<b>1.000</b>	0.026	0.885
Biglog	<b>0.998</b>	0.999	<b>1.000</b>	<b>0.998</b>

## Domain-adaptiveness of Biglog:

1. Superior to specialized domain adaptation methods, such as LogTransfer
2. Achieves good performance with limited target-domain samples
3. Accurately discriminates normal and abnormal logs in the target domain
4. Outperforms LogTransfer even with a small amount of target-domain data
5. Generalized log representations from pre-training phase enhance domain-adaptiveness



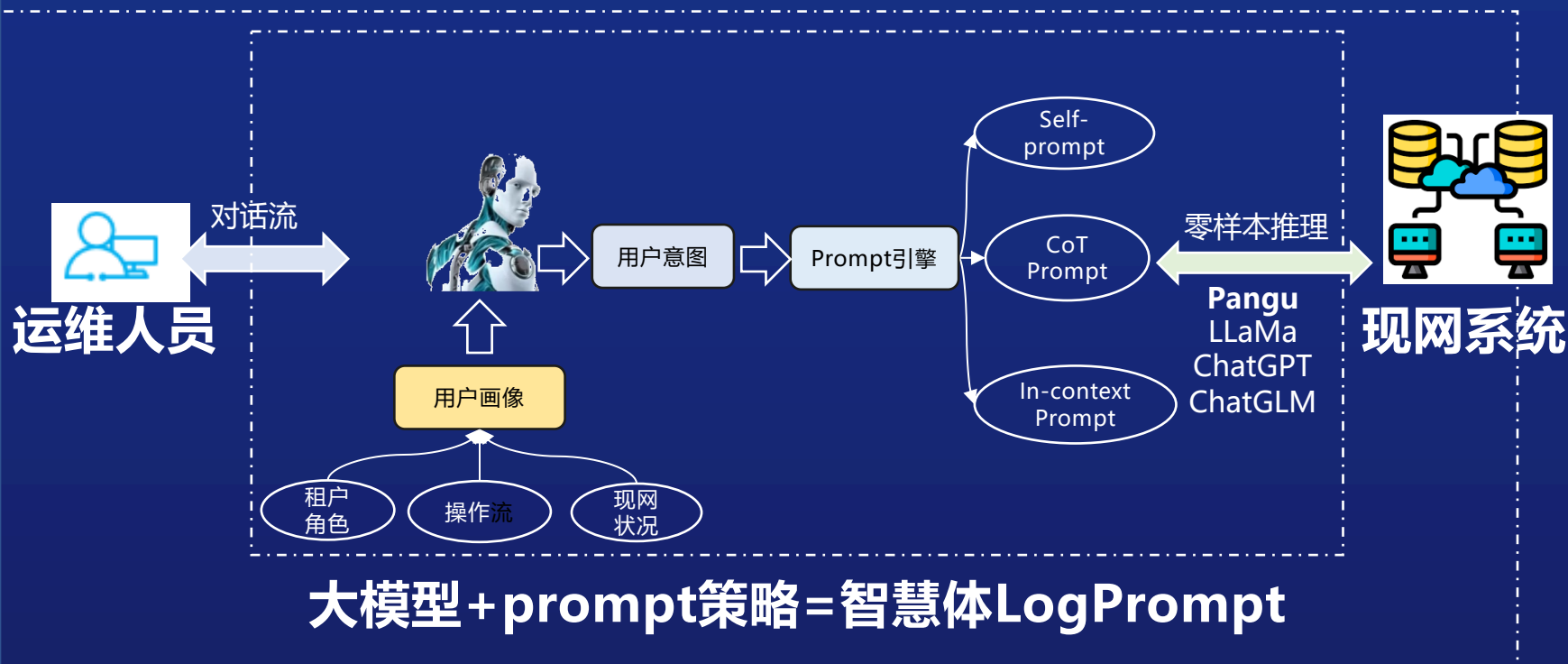
## Few-shot Learning Ability of Biglog:

1. Exceptional performance with extremely limited training data
2. Outperforms DeepLog and LogAnomaly with as few as 20 samples
3. Surpasses multiple baselines when training samples increase to 100
4. Demonstrates remarkable F1-score on Spirit dataset without prior exposure
5. Promising results in new domains with few labeled historical logs

# LogPrompt: 日志运维向智慧体演进, 增强可解释性, 摆脱任务数据依赖

Biglog 实现了日志理解, 基于少量任务数据可以自动映射故障现象, 但依然没有完成智能运维的最后一步: 根因分析和故障自恢复。Biglog 的交互设计缺乏反馈与互动, 离“智慧体”距离遥远。

运维智慧体LogPrompt: 并非数据驱动, 而是Prompt驱动, 激发大模型的日志分析和推理能力, 充当设备系统与工程师之间交流沟通的桥梁。



可交互性运维分析	
用户	日志中出现如下命令该怎么办: net use \\192.168.xx.xx /del /y
智慧体	如果您在日志中遇到了此命令, 意味着有人试图删除与IP地址为192.168.xx.xx的计算机的网络连接。
用户	有什么建议?
智慧体	请执行以下操作: 打开命令提示符或PowerShell窗口。可以通过按下Win + R组合键, 键入"cmd"或"PowerShell", 然后按下回车键来打开它们。 在命令提示符或PowerShell窗口中, 输入以下命令并按下回车键: net use \\192.168.xx.xx /del /y
	系统将尝试删除与该IP地址的网络连接。如果成功删除, 您将收到一条确认消息。

# LogPrompt: Prompt策略+大模型=强大的零样本故障推断与可解释性日志分析能力



现网设备日志

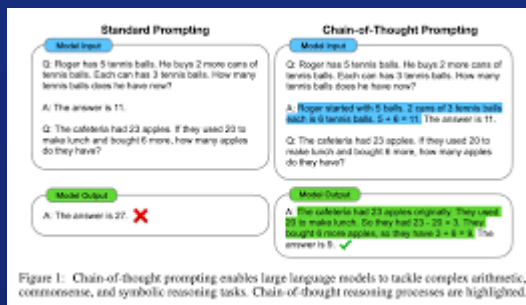


Prompt引擎

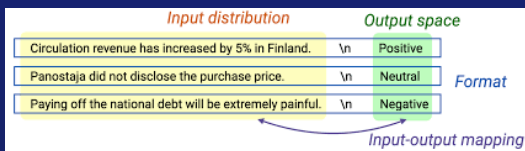


Self-prompt

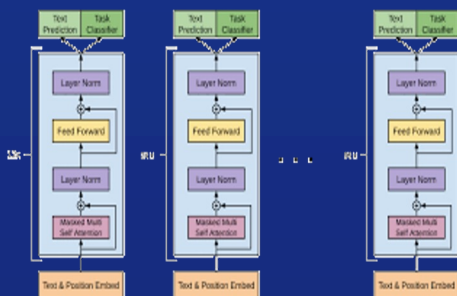
CoT-prompt



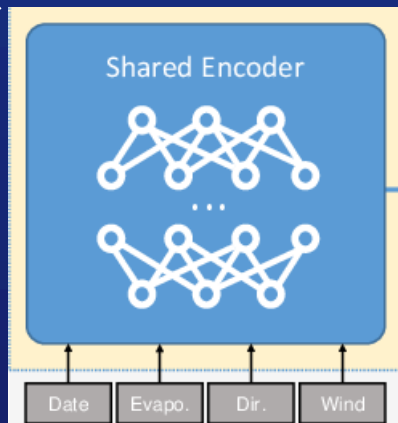
In-context prompt



任意大模型



大模型盘古



开源大模型

Methods	T.R. <sup>a</sup>	HDFS		Hadoop		Zookeeper		BGL		HPC		Linux		Proxifier		Android		Avg.	
		RI <sup>b</sup>	F1	RI	F1	RI	F1	RI	F1	RI	F1	RI	F1	RI	F1	RI	F1	RI	F1
IPLoM [19]		0.914	0.389	0.636	0.068	0.787	0.225	0.858	0.391	0.228	0.002	0.695	0.225	<b>0.822</b>	0.500	0.918	0.419	0.733	0.277
LKE [20]		0.861	0.424	0.150	0.198	0.787	0.225	0.848	0.379	0.119	0.381	<b>0.825</b>	0.388	0.379	0.309	0.045	0.000	0.502	0.288
LogSig [21]		0.872	0.344	0.651	0.050	0.787	0.225	0.806	0.333	0.119	0.002	0.715	0.146	0.559	0.339	0.732	0.116	0.655	0.194
FT-tree [22]		0.908	0.385	0.668	0.046	0.773	0.186	0.275	0.497	0.119	0.002	0.709	0.211	0.722	0.420	0.918	0.581	0.636	0.291
Spell [23]	10%	0.871	0.000	0.721	0.058	0.102	0.045	0.503	0.536	0.882	0.000	0.706	0.091	0.621	0.000	0.822	0.245	0.654	0.122
Drain [24]		0.914	0.389	0.647	0.068	0.787	0.225	0.822	0.397	0.119	0.002	0.695	0.225	<b>0.822</b>	0.500	0.916	0.413	0.716	0.277
MoLFI [25]		0.871	0.000	0.699	0.095	0.899	0.000	0.792	0.333	0.881	0.000	0.410	0.026	0.621	0.000	0.173	0.208	0.668	0.083
LogParse [26]		0.907	0.632	0.349	0.502	0.982	0.348	<b>0.992</b>	0.665	0.194	0.330	<b>0.825</b>	0.588	0.490	0.334	0.288	0.233	0.628	0.454
LogStamp [2]		<b>0.954</b>	0.523	<b>0.927</b>	0.594	<b>0.992</b>	0.275	0.984	0.818	<b>0.949</b>	0.434	0.760	0.658	0.811	0.438	0.974	<b>0.899</b>	<b>0.919</b>	0.580
LogI	0%	0.890	0.927	0.879	<b>0.862</b>	0.948	<b>0.934</b>	0.964	<b>0.943</b>	0.934	<b>0.796</b>	0.758	<b>0.860</b>	0.567	<b>0.998</b>	<b>0.978</b>	0.725	0.865	<b>0.881</b>

Methods	BGL <sup>c</sup>				Spirit			
	T.N. <sup>a</sup>	P	R	F	T.N.	P	R	F
DeepLog [31]		0.156	<b>0.941</b>	0.268	<b>0.302</b>	0.821	0.441	
LogAnomaly [32]	4713	0.250	0.001	0.001	7958	0.297	0.855	0.441
LogRobust [33]		0.131	0.627	0.217	0.285	0.942	0.438	
Log	<b>0</b>	<b>0.249</b>	0.834	<b>0.384</b>	<b>0</b>	0.290	<b>0.999</b>	<b>0.450</b>

LogPrompt (零训练数据) 在异常检测、日志解析任务相较于现有方法 (大量训练数据) 取得**12项**最佳表现, 超过现有方法**51.9%**。

Tasks		Correctness	Usefulness	Readability
Log Parsing	Reviewer 1	73.98%	90.24%	93.50%
	Reviewer 2	82.93%	91.06%	94.31%
	Reviewer 3	78.86%	84.55%	94.31%
	Avg.	78.59%	88.61%	94.04%
Anomaly Detection	Reviewer 1	-	100.00%	99.00%
	Reviewer 2	-	92.00%	98.00%
	Reviewer 3	-	92.00%	98.00%
	Avg.	-	94.67%	98.34%

LogPrompt的根因生成与变量解释能力在业界专家评测中取得了**90+**以上的高分。

# LogPrompt: Prompt引擎实际应用效果

1. 你是用户的一个运维助手，你将基于给定的用户查询给出可能的API及其参数选择

2. 你需要给出一个需要调用确定的API

3. 你需要给出调用API的参数

4. 只允许通过给定的已知内容生成回答，不允许编造未知的API及参数

导入带有API信息的本地知识库



知识库：查询哪些ap最繁忙，第一步需要调用的API：`/xxx/inter/query/api`，第二步调用API的传参：`id`、`startTime`、`endTime`、`xxx`、`xxx`、`xxx`。





2023 CCF国际AIOps挑战赛决赛  
暨“大模型时代的AIOps”研讨会

## 第四章

# 未来畅想



## 通用大模型目标AGI



期望

1. 纯自然语言的交互
2. 被动式运维
3. 智能Agent自运维

可控

1. 违背人的意图
2. Safety保障人生命
3. ...

- 语言驱动基础模型(NLP)
- 面向多模态统一(NLP+视觉)



2023 CCF国际AIOps挑战赛决赛暨“大模型时代的AIOps”研讨会

# THANKS