



(12) 发明专利申请

(10) 申请公布号 CN 118626330 A

(43) 申请公布日 2024. 09. 10

(21) 申请号 202410592483.2

(22) 申请日 2024.05.14

(71) 申请人 复旦大学

地址 200433 上海市杨浦区邯郸路220号

(72) 发明人 周水庚 张路 霍思捷

(74) 专利代理机构 上海正旦专利代理有限公司

31200

专利代理师 陆飞 陆尤

(51) Int. Cl.

G06F 11/30 (2006.01)

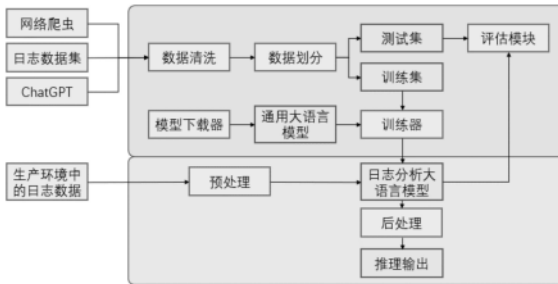
权利要求书2页 说明书7页 附图1页

(54) 发明名称

一种基于大语言模型的日志分析方法

(57) 摘要

本发明属于自然语言处理技术领域,具体为一种基于大语言模型的日志分析方法。本发明包括:使用网络爬虫技术从多个数据源自动收集日志问题相关的问答数据;使用ChatGPT扩充数据量;去除数据集中冗余的数据;将数据集分割为训练集和测试集;利用训练集对本地通用大语言模型进行微调;利用测试集对本地通用大语言模型进行评估;针对不同细分任务给出各自相对应的指标;利用待分析的新的日志数据进行评估,实现对新的日志数据的分析,得到分析结果与评估指标。本发明并简化了通用大语言模型应用到日志处理数据上的微调和部署过程,方便易用、准确性高,特别适用于需要处理大量日志数据的任务。



1. 一种基于大语言模型的日志分析方法,其特征在于,具体步骤如下:

(1) 使用网络爬虫技术,从多个数据源自动收集日志问题相关的问答数据;

(2) 使用ChatGPT,利用收集到的问答数据以及日志语句的公开数据集,基于自聊天技术批量生成更多日志语句,用于扩充数据量;

(3) 去除数据集中冗余的数据:将过长的日志语句分解为多条较短的日志语句,并使用分词器将日志语句拆分为token序列;

(4) 将日志数据的公开数据集和由ChatGPT生成的数据集合并为一个统一的数据集,再分割为训练集和测试集;

(5) 输入拟用于微调的通用大语言模型名称,根据所输入的大语言模型名称自动从Huggingface和modelscope开源大模型平台上搜寻对应的大语言模型,若存在匹配的大语言模型,将这一模型下载到本地;

(6) 利用缓存路径下的训练集对本地的通用大语言模型进行微调;

(7) 利用缓存路径下的测试集对本地的通用大语言模型进行评估,针对不同细分任务给出各自相对应的指标,以便于进行比较;

(8) 利用待分析的新的日志数据进行评估,即实现对新的日志数据的分析,得到在这一批数据上的分析结果与评估指标。

2. 根据权利要求1所述的基于大语言模型的日志分析方法,其特征在于,步骤(1)的具体流程为:使用爬虫技术在stackoverflow上爬取有关日志问题的问答,以及最受欢迎的问答;对于每个问题,爬取获赞数最多的回答作为标准答案,并忽略其他回答。

3. 根据权利要求1所述的基于大语言模型的日志分析方法,其特征在于,步骤(2)中所述批量生成更多日志语句的流程为:

加入以Loghub和LogPub的混合日志集合作为基础,使用ChatGPT生成的日志分析对话数据;从日志集合中随机抽取k个连续的日志数据,其中k的值是通过均匀分布采样设置,基于这些k个连续的日志x,随机抽取任务集T中的一个任务 T_i ;为了确保每个任务不局限于特定的提示指令,每个任务都有使用ChatGPT 4生成的相应的指令集,表示为 P_i ;随机抽取 P_i 中的一个指令p作为当前日志x的指令,将抽取的日志数据x和任务指令p结合在一起,发送给ChatGPT以生成答案;使用自聊天让ChatGPT基于抽取的k个连续日志x生成相应的问题和答案。

4. 根据权利要求3所述的基于大语言模型的日志分析方法,其特征在于,步骤(3)所述去除数据集中冗余的数据,包括:

对于日志故障预测,只保留每个日志片段中的前几个日志进行异常检测,并删除最后的N个日志;

对原始日志数据进行分割,将长篇的日志记录分解为更小、更易于处理的片段,然后使用一个分词器将文本消息转换为分词后的ID序列。

5. 根据权利要求4所述的基于大语言模型的日志分析方法,其特征在于,步骤(6)中所述对本地的通用大语言模型进行微调,具体包括:

采用低秩自适应(LoRA)的有监督微调方法,其支持使用python的accelerate库,缩短每一轮次微调所消耗的时间;具体地,有监督微调方法中弹出一个黑框,以文本形式给出一系列可用参数的提示,允许使用这些参数对模型进行多样化的调整;通过指定CUDA_

VISIBLE_DEVICES,可以确定微调任务在哪些GPU设备上运行,从而充分利用多个GPU进行并行计算;通过设置stage和finetuning_type参数,可以指定微调的阶段和类型,以满足不同的微调需求;template参数用于指定模型模板,而flash_attn和shift_attn则控制是否使用特定的注意力机制加速模型训练;cutoff_len控制输入序列的最大长度,max_samples限制训练使用的最大样本数;训练参数learning_rate、num_train_epochs、per_device_train_batch_size,用于调整训练的学习率、轮数和批次大小;lr_scheduler_type用于指定学习率调度器的类型,max_grad_norm控制梯度裁剪的最大值,warmup_steps用于指定预热步数;NEFT训练中的alpha参数、LoRA微调中的秩参数、dropout率和目标参数通过命令来指定;fp16参数控制是否使用混合精度训练提高性能;微调过程完成后,通过文本匹配,去除格式化的、重复的或无关紧要的内容,同时,保留关键的错误信息、警告和系统状态变更。

6. 根据权利要求5所述的基于大语言模型的日志分析方法,其特征在于,步骤(7)中所述对本地的通用大语言模型进行评估,具体包括:

对于异常检测和故障预测任务,选择准确率、召回率、F1分数指标;

对于日志解析任务,选择GA、PA、编辑距离指标;

对于根因分析任务,选择ROUGE-1、ROUGE-2、ROUGE-L指标;

根据这些指标对模型的性能进行量化分析,帮助识别当前模型的性能瓶颈,还为未来的优化方向提供数据支持。

一种基于大语言模型的日志分析方法

技术领域

[0001] 本发明属于自然语言处理技术领域,具体涉及基于大语言模型的日志分析方法。

背景技术

[0002] 在目前的软件系统中,日志信息作为记录系统运行时事件和状态的关键信息,对于系统的异常检测、故障诊断和根本原因推断等任务至关重要。日志分析任务分为上游任务(如日志解析)和下游任务(如异常检测、根本原因分析和故障预测),它们共同推动了运维的智能化和云计算行业的发展。

[0003] 随着大语言模型的快速发展,它们在多个行业中引发了颠覆性变革,尤其是在理解和生成类似人类文本方面的能力,为自动化和增强以前依赖于手动和劳动密集型过程的任务提供了新的可能性。然而,尽管大语言模型在日志解析任务中已有成功应用,对于更复杂的下游日志分析任务,如异常检测和故障预测,大语言模型的应用场景还有待开拓。

[0004] 为了在真实系统中部署以解决各种日志分析问题,开发一个可接受规模的、面向日志的、高性能的大语言模型日志分析方法是必要的。日志数据与自然语言之间存在如下几个差异:日志数据包含大量重复片段;需要特定领域的专业知识;常常包含公司系统的敏感信息;以及数据量较大(尤其是在云计算系统中)。这些特点使得直接应用现有的开源通用大语言模型进行日志分析的效果不佳。在实际部署环境中,出于对成本和隐私问题的考虑,直接调用在线大语言模型的API也不切实际。

[0005] 目前,日志领域对于大语言模型的运用主要集中于上游任务,即日志解析上,而对于日志分析下游任务中的运用尚未出现,亟待设计一种高效、准确的适用于日志分析下游任务的基于大语言模型的日志分析方法。

发明内容

[0006] 本发明的目的在于提供一种基于大语言模型的日志分析方法,旨在通过自动化的日志数据收集与处理,简化大语言模型在日志分析任务上的微调和部署过程,并提供对不同细分任务的评估支持。

[0007] 本发明中,所述日志数据,是由系统生成的、本质上是半结构化或非结构化的文本,用于记录操作事件。日志解析的挑战在于从这些文本中提取结构化信息,以便进行后续的分析任务,如异常检测、系统诊断和根本原因推断。

[0008] 本发明中,所述日志解析,是将日志解析为日志模板和日志变量,两者之间有占位符<*>连接;日志变量表示在实际日志中可以变化的部分;一个日志解析的例子如下:HDFS系统(Hadoop分布式文件系统)中的一条日志信息是“PacketResponder 2for block blk6653557694518759805terminating”。该日志可以被解析为日志模板和日志变量,其中日志模板是“PacketResponder<*>for block<*>terminating”,而日志变量是“2”和“blk6653557694518759805”。“<*>”是占位符,用于表示在实际日志中可以变化的部分。在给定的例子中,“2”和“blk 6653557694518759805”是实际的值,它们替换了模板中的占位

符。经过这样的解析后,便于后续进行其他的下游分析任务。

[0009] 本发明提供的基于大语言模型的日志分析方法,给出了日志数据的自动化数据收集与处理过程,简化了大语言模型在日志处理数据上的微调和部署过程,并支持对日志分析的不同细分任务进行评估,具体步骤如下:

[0010] (1) 使用网络爬虫技术,从多个数据源自动收集日志问题相关的问答数据;

[0011] (2) 使用ChatGPT (Schulman J,Zoph B,Kim C,et al.OpenAI.ChatGPT: Optimizing Language Models for Dialogue[EB/OL].[2022-11-30].<https://openai.com/index/chatgpt/>),利用收集到的问答数据以及日志语句的公开数据集,基于自聊天技术批量生成更多日志语句,用于扩充数据量;

[0012] (3) 去除数据集中冗余的数据:将过长的日志语句分解为多条较短的日志语句,并使用分词器将日志语句拆分为token序列;

[0013] (4) 将日志数据的公开数据集和由ChatGPT生成的数据集合并为一个统一的数据集,再分割为训练集和测试集;

[0014] (5) 输入拟用于微调的通用大语言模型名称,根据所输入的大语言模型名称自动从Huggingface和modelscope等开源大模型平台上搜寻对应的大语言模型,若存在匹配的大语言模型,将这一模型下载到本地;

[0015] (6) 利用缓存路径下的训练集对本地的通用大语言模型进行微调;

[0016] (7) 利用缓存路径下的测试集对本地的通用大语言模型进行评估,针对不同细分任务给出各自相对应的指标,以便于进行比较;

[0017] (8) 利用待分析的新的日志数据进行评估,即可实现对新的日志数据的分析,得到在这一批数据上的分析结果与评估指标。

[0018] 进一步地,步骤(1)的具体流程为:

[0019] 使用网络爬虫技术,从不同的数据源自动收集数据,为了增强大型模型的日志分析和根因分析能力;具体地址,使用爬虫技术在stackoverflow上爬取有关日志问题的问答,以及最受欢迎的问答。本发明实施方式中,总共爬取了stackoverflow上有关日志的20,000个问答,以及7,300个最受欢迎的问答。对于每个问题,爬取获赞数最多的回答作为标准答案,并忽略其他回答。由于stackoverflow本身是以问答形式存在,它可以自然地用作聊天数据来微调大型模型。

[0020] 进一步地,步骤(2)中所述批量生成更多日志语句的流程为:

[0021] 加入以Loghub和LogPub的混合日志集合作为基础,使用ChatGPT生成的日志分析对话数据。从日志集合中随机抽取k个连续的日志数据,其中k的值是通过均匀分布采样设置的,即 $k \sim U(a, b)$ 。抽取的日志表示为x;在实验中,设置 $a=1, b=50$;基于这些k个连续的日志x,随机抽取任务集T中的一个任务 T_i ;为了确保每个任务不局限于特定的提示指令,每个任务都有使用ChatGPT 4生成的相应的指令集,表示为 P_i ;随机抽取 P_i 中的一个指令p作为当前日志x的指令,将抽取的日志数据x和任务指令p结合在一起,发送给ChatGPT以生成答案。使用自聊天让ChatGPT基于抽取的k个连续日志x生成相应的问题和答案。所生成的这部分数据不会被划分到测试集中。本步骤可以提升步骤(6)中微调的效果;

[0022] 进一步地,步骤(3)所述去除数据集中冗余的数据,包括:

[0023] 对于日志故障预测,由于故障预测的本质是预测未来可能出现的异常,只保留每

个日志片段中的前几个日志进行异常检测,并删除最后的N个日志。在本发明的实验中,N设置为20。为了确保数据集能够在步骤(6)、(7)中以适合模型训练和评估的格式被使用,对原始日志数据进行分割,将长篇的日志记录分解为更小、更易于处理的片段,然后使用一个分词器(tokenizer)将文本消息转换为分词后的ID序列。这一步骤在确保每个片段完整性的同时,有效减少信息的冗余。

[0024] 步骤(6)中所述对本地的通用大语言模型进行微调,具体包括:

[0025] 为了让通用大语言模型加深对日志数据的理解,使用有监督微调方法对步骤(8)中所指定的大语言模型在步骤(7)中得到的训练集上进行微调,本发明采用低秩自适应(LoRA,摘自“Hu J E, Shen Y L, Wallis P, et al. LoRA: Low-Rank Adaptation of Large Language Models[C]. Proceedings of the International Conference on Learning Representations (ICLR), 2021”)这一典型的有监督微调方法。本方法支持使用python的accelerate库,缩短每一轮次微调所消耗的时间,这不仅加快了处理速度,还有助于模型推理准确率的提高。具体而言,本方法会弹出一个黑框,以文本形式给出一系列可用参数的提示,允许使用这些参数对模型进行多样化的调整。通过指定CUDA_VISIBLE_DEVICES,可以确定微调任务在哪些GPU设备上运行,从而充分利用多个GPU进行并行计算;通过设置stage和finetuning_type参数,可以指定微调的阶段和类型,以满足不同的微调需求;template参数用于指定模型模板,而flash_attn和shift_attn则控制是否使用特定的注意力机制加速模型训练;cutoff_len控制输入序列的最大长度,max_samples限制训练使用的最大样本数(为确保成功部署运行,这两个参数的设定应以不超过机器显存为限度);训练参数learning_rate、num_train_epochs、per_device_train_batch_size等,可用于调整训练的学习率、轮数和批次大小等;lr_scheduler_type用于指定学习率调度器的类型,max_grad_norm控制梯度裁剪的最大值,warmup_steps用于指定预热步数;NEFT训练中的alpha参数、LoRA微调中的秩参数、dropout率和目标参数等也可以通过命令来指定;fp16参数控制是否使用混合精度训练提高性能。微调过程完成后,通过文本匹配,去除格式化的、重复的或无关紧要的内容,如建议性的起始句子。同时,保留关键的错误信息、警告和系统状态变更。

[0026] 步骤(7)中所述对本地的通用大语言模型进行评估,具体包括:

[0027] 对于异常检测和故障预测任务,选择准确率、召回率、F1分数等指标;对于日志解析任务,选择GA、PA、编辑距离等常规指标;对于根因分析任务,选择ROUGE-1、ROUGE-2、ROUGE-L等指标。根据这些指标对模型的性能进行量化分析。这些评估指标不仅帮助识别当前模型的性能瓶颈,还为未来的优化方向提供了数据支持,如更换其他模型或更改训练参数。具体而言,先删除冗余词句,利用最长公共子序列计算诸如编辑距离等指标,这样计算保证了指标结果的精确性,便于进行后续比较与参数调整。接着依据模型微调的输出统计准确率、召回率,并计算F1分数。

[0028] 本发明至少包括以下有益效果:

[0029] (1) 本发明通过自动化的数据收集与处理流程,不仅大幅提高了日志分析的效率,还减少了人工介入,从而降低了出错的可能性。此外,本发明利用大语言模型对日志数据进行深入分析,能够更准确地识别和解析日志中的信息,包括对异常的检测和故障的预测,这极大增强了日志分析的准确性。

[0030] (2) 本发明简化大语言模型在日志分析任务上的微调和部署过程,减少了对外部

API的依赖,这不仅降低了财务成本,也减少了隐私风险,实现了成本的降低和数据安全性的提升。

[0031] (3) 本发明自动化和优化的日志分析流程,使得系统运维团队能够更快地响应系统问题,提高系统的稳定性和可维护性,同时也增强了系统的智能化水平。

[0032] (4) 本发明通过集成多种大语言模型,日志分析不再局限于简单的文本处理,而是成为一个具有一定智能水平的分析过程。

[0033] (5) 本发明通过本地化处理日志数据减少了对外部服务的依赖,进一步增强了数据的安全性和隐私保护。本发明提供了实时的微调和评估进度显示,以及交互式的用户界面,这使得即使是非专业用户也能方便地使用和理解日志分析结果。本发明所构建的ChatLog数据集和LogLLM框架支持不同大小的多个开源大语言模型,这为未来的研究和开发工作提供了灵活性和可扩展性,允许在这个基础上进一步扩展和改进。

[0034] 本发明的其他优点、目标和特征将部分通过下面的说明体现,部分还将通过对发明的研究和实践而为本领域的技术人员所理解。

附图说明

[0035] 图1为本发明的框架图。

具体实施方式

[0036] 下面通过实施例进一步介绍本发明。

[0037] 本实施例提供的基于大语言模型的日志分析方法,其流程如图1所示。

[0038] 首先,使用网络爬虫技术,从多个数据源自动收集数据。本骤是整个流程的基础,数据的质量和完整性将直接影响后续处理的效果。具体地,从stackoverflow的问答平台上爬取有关日志问题的问答,以及最受欢迎的问答,从而获取自然语言处理的高质量数据。在爬取数据时,需要特别注意获取获赞数最多的回答作为标准答案,以确保数据的准确性和代表性。具体来说,总共爬取了stackoverflow上关于日志的20,000个问答,以及7,300个最受欢迎的问答。对于每个问题,爬取了获赞数最多的回答作为标准答案,并忽略了其他回答。由于stackoverflow本身是以问答形式存在,可以自然地用作聊天数据来微调大型模型。

[0039] 接着,对收集到的数据集进行解析,以确定数据集的加载来源,并执行相应的数据集加载逻辑。这一步骤涉及到对数据集属性的深入理解,以确保数据能够被正确地加载和使用。例如,可以从开源专业数据集Loghub和LogPub的子集中生成训练数据集和测试数据集,同时调整日志分析各个子任务的数据集中样本数量,为每个聊天生成指令,包括任务提示和一个特定的日志进行分析。对于日志解析,我们从Loghub (除Android和CTS子集外) 和LogPub的所有子集中生成用于的微调的数据。使用Loghub的Android和CTS子集生成用于的评估的数据。过滤掉其中长度小于100个字符的回答和非英语回答,以及未能成功响应的数据。同时,由于不同子集中的数据量差异很大,我们将每个子集中的样本数量限制在20,000以内。最后,为了防止其他任务的数据因日志解析数据过多而被淹没,我们将所有子集的样本混合,并随机选择100,000个样本。

[0040] 为了确保数据的完整性和一致性,需要对加载的数据集进行校验和对齐处理。这

一步骤至关重要,因为它能够确保数据文件的准确性,为后续的数据处理打下坚实的基础。在校验过程中,需要特别注意过滤掉长度小于100个字符的回答、非英语回答以及未能成功响应的数据,以提高数据集的质量。

[0041] 将多个数据集合并为一个统一的数据集后,加载预定义的模板,并根据模板特性对分词器进行适配。这一步骤涉及到对分词器的调整,以确保其能够适应不同模板的特性,从而提高数据处理的效率和准确性。

[0042] 使用分词器对数据集进行预处理后,需要将预处理后的数据集分割为训练集和测试集,并保存到缓存路径。这一步骤是机器学习流程中的关键环节,因为它涉及到如何将数据集有效地分配给模型进行训练和评估。

[0043] 在输入拟用于微调的通用大语言模型名称后,系统将自动从Huggingface和modelscope等多个开源平台上进行搜寻,以寻找匹配的模型。若存在匹配的模型,将这一模型下载到本地,为后续的微调工作做好准备。

[0044] 利用缓存路径下的训练集对本地的通用大语言模型进行微调,是提升模型性能的重要步骤。在微调过程中,本方法使用有监督微调方法,包括但不限于低秩自适应(LoRA),以提高训练的效率和模型的推理准确率。同时,通过文本匹配去除无关内容,保留关键信息,进一步提升模型的分析能力。

[0045] 最后,利用缓存路径下的测试集对本地的通用大语言模型进行评估。算法根据不同的日志分析任务,计算出相应的评估指标的值,对于异常检测和故障预测任务,选择准确率、召回率、F1分数等指标;对于日志解析任务,选择GA、PA、编辑距离等常规指标;对于根因分析任务,选择ROUGE-1、ROUGE-2、ROUGE-L等指标。根据这些指标,可以对模型的性能进行量化分析。这些评估指标不仅帮助识别当前模型的性能瓶颈,还为未来的优化方向提供了数据支持,如更换其他模型或更改训练参数。

[0046] 对实验所涉及的日志分析的四个下游任务的定义如下:日志解析涉及从系统生成的半结构化或非结构化文本日志中提取关键信息,将其转换为结构化数据,以便于进行更深入的分析和处理;根本原因分析(RCA)是一种诊断过程,旨在从大量日志数据中识别异常模式或错误信息,追溯并确定系统异常或故障的深层原因;日志异常检测:通过实时分析日志数据,使用机器学习和深度学习技术来识别与正常操作模式不符的异常事件,以便在问题变得严重前进行干预;日志故障预测:基于当前和历史日志数据中的征兆,预测未来可能发生的系统故障或异常,是一种预防性维护手段,具有高度的前瞻性和实用价值。

[0047] 为验证本方法的优越性,针对上文所述的四个日志分析下游任务,本实施例使用以下五种现有的通用大模型进行微调以进行对比:Baichuan2-7B-Chat(摘自“Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models[J/OL]. arXiv:2307.09288. 2023.”), ChatGLM3-6B-Chat(摘自“Du Z, Qian Y, Liu X, Ding M, Qiu J, Yang Z, Tang J. Glm: General language model pretraining with autoregressive blank infilling[C]. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2022: 320-335.”), LLaMA2-7B-Chat(摘自“Yang A, Xiao B, Wang B, et al. Baichuan 2: Open large-scale language models[J/OL]. arXiv:2309.10305. 2023.”), Gemma-2B-Chat(摘自“Mesnard T, Hardin C, Dadashi R, et al. Gemma: Open Models Based on Gemini

Research and Technology[J/OL].arXiv:2403.08295.2023.”),Mistral-7B-Instruct(摘自“Jiang A Q,Sablayrolles A,Mensch A,et al.Mistral 7b[J/OL].arXiv:2310.06825.2023.”)。

[0048] 本实施例中,对于异常检测和故障预测任务,选择准确率、召回率、F1分数等指标;对于日志解析任务,选择GA、PA、编辑距离等常规指标;对于根因分析任务,选择ROUGE-1、ROUGE-2、ROUGE-L等指标,以衡量各算法的性能。

[0049] 表1至4给出了五个大语言模型在使用本发明微调前后在四个任务对应的数据集上的预测精度。对于日志解析任务,未经过本发明微调的LLM在组准确率方面表现较差,最高仅为4.10%,而经过微调后,最高组准确率达到16.31%。所有LLM在解析准确率方面均有显著改善,且微调后的效果更为明显,表明本发明对于提高日志解析任务的性能有显著效果。对于日志异常检测任务,未经过微调的LLM在准确率、F1分数、精确率和召回率方面表现极低,例如,Baichuan2-7B-Chat模型的召回率仅为24.99%。这反映出对于日志解析任务而言,本发明对大语言模型的各项指标均有所提升,尤其是精确率和召回率。对于日志故障预测任务,经过微调后,大语言模型的故障预测性能得到了显著提升,例如,Baichuan2-7B-Chat模型的准确率从0.19%提升至4.66%。此外,微调后召回率也得到了显著提升。对于根因分析任务,因为该任务挑战性较大,微调后大语言模型的性能虽然有所改善,但仍然面临挑战。例如,Baichuan2-7B-Chat模型在BLEU-4指标上仅从9.36提升至10.61。这表明即便对于相对困难的根因分析任务,使用本发明所提出的框架也可以进一步提升性能。综上所述,本发明对于改善LLM在日志分析各项任务中的性能至关重要,但不同任务的性能提升程度有所不同。

[0050] 表1:五个大语言模型在日志解析任务中微调前后的预测精度对比

微调前/后	模型名称	GA	PA	编辑距离
[0051] 微调前	Baichuan2-7B-Chat	0.71	0.0	61.63
	ChatGLM3-6B-Chat	2.51	0.0	54.57
	LLaMA2-7B-Chat	3.49	0.0	53.66
	Gemma-2B-Chat	10.91	0.0	46.03
	Mistral-7B-Instruct	6.32	0.0	49.48
微调后	Baichuan2-7B-Chat	9.92	0.0	49.32
	ChatGLM3-6B-Chat	35.25	0.0	34.91
	LLaMA2-7B-Chat	55.48	0.0	12.93
	Gemma-2B-Chat	62.60	0.0	10.46
	Mistral-7B-Instruct	5.37	0.0	12.18

[0052] 表2:五个大语言模型在异常检测任务中微调前后的预测精度对比

[0053]	微调前/后	模型名称	正确率	F1 分数	准确率	召回率
	微调前	Baichuan2-7B-Chat	0.19	0.46	0.23	24.28
		ChatGLM3-6B-Chat	0.38	0.48	0.24	31.59
		LLaMA2-7B-Chat	0.19	0.65	0.34	25.97
		Gemma-2B-Chat	0.33	0.62	0.27	35.54
		Mistral-7B-Instruct	0.56	0.27	0.15	35.82
	微调后	Baichuan2-7B-Chat	5.32	0.41	0.2	34.51
		ChatGLM3-6B-Chat	3.42	0.47	0.24	34.41
		LLaMA2-7B-Chat	5.51	0.71	0.52	38.4
		Gemma-2B-Chat	11.6	0.67	0.34	47.41
		Mistral-7B-Instruct	16.16	0.33	0.17	42.28

[0054] 表3:五个大语言模型在故障预测任务中微调前后的预测精度对比

[0055]	微调前/后	模型名称	正确率	F1 分数	准确率	召回率
	微调前	Baichuan2-7B-Chat	0.19	0.47	0.24	24.99
		ChatGLM3-6B-Chat	3.3	0.51	0.26	33.39
		LLaMA2-7B-Chat	3.69	0.67	0.34	29.12
		Gemma-2B-Chat	4.1	0.58	0.33	28.77
		Mistral-7B-Instruct	5.63	0.3	0.23	30.32
	微调后	Baichuan2-7B-Chat	4.66	0.39	0.2	33.5
		ChatGLM3-6B-Chat	6.41	3.61	3.4	36.09
		LLaMA2-7B-Chat	5.83	2.02	1.84	38
		Gemma-2B-Chat	16.31	0.67	0.35	48.97
		Mistral-7B-Instruct	16.12	0.4	0.2	42.52

[0056] 表4:五个大语言模型在根因分析任务中微调前后的预测精度对比

[0057]	微调前/后	模型名称	BLEU-4	Rouge-1	Rouge-2	Rouge-l
	微调前	Baichuan2-7B-Chat	9.36	5.41	1.48	3.49
		ChatGLM3-6B-Chat	27.71	31.26	7.8	16.28
		LLaMA2-7B-Chat	25.85	28.13	6.34	14.29
		Gemma-2B-Chat	21.32	31.9	8.21	14.03
		Mistral-7B-Instruct	28.48	35.79	8.05	17.72
	微调后	Baichuan2-7B-Chat	10.61	7.04	1.16	3.46
		ChatGLM3-6B-Chat	12.11	22.62	3.42	10.62
		LLaMA2-7B-Chat	11.02	21.99	3.32	10.3
		Gemma-2B-Chat	22.75	29.19	7.24	14.79
		Mistral-7B-Instruct	26.87	32.04	9.4	17.08

。

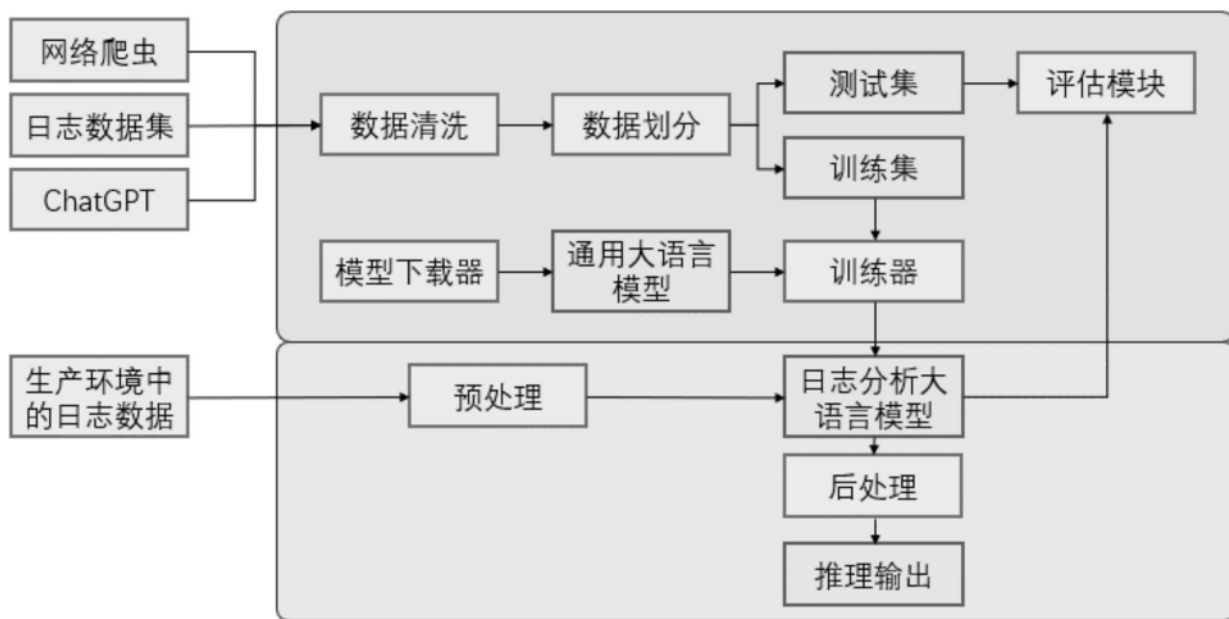


图1