文章编号:1000-2367(2023)06-0047-10 DOI:10.16366/j.cnki.1000-2367.2023.06.006

基于数据中台的日志解析技术

金铭,崔硕,温阳,卞琳,郭学良,冯函宇

(国家电网有限公司 大数据中心,北京 100032)

摘 要:数据中台是一种利用数据技术为客户提供高效服务的模式.日志是数据中台记录系统运行状态的一种方式,它可以为故障诊断、性能优化、系统安全等任务提供支持,分析日志中的信息对中台日常运维具有重要意义.日志解析是日志挖掘的重要步骤,它将非结构化的日志文本转换为结构化的数据.综述了日志解析算法和评估方法,分析了工业界和学术界的解决方案,总结了日志解析算法的主要类别和特点,比较了不同算法在不同数据集上的性能和效果.发现日志解析算法缺乏统一的标准和数据集,导致结果难以对比和验证.针对这种情况,对未来的研究方向提出建议,应关注建立统一的评估指标和日志数据集,促进工业界和学术界的交流,以提高日志解析算法的适用性和可靠性,对日志解析领域的研究具有参考价值.

关键词:数据中台;日志解析;日志挖掘;算法评估

中图分类号:TF407;TP302 **文献标志码**:A

数据中台是指通过数据技术对海量数据进行采集、计算、存储和处理,同时统一标准和口径,形成全域级、可复用的数据资产中心和数据存储能力中心,形成大数据资产层,进而为客户提供高效的服务.数据中台的建立,通过数据采集、数据治理等手段可逐渐消除信息化领域典型的“数据孤岛”问题,打通企业内部的数据流通,减少数据开发成本;通过数据分析,能够充分挖掘数据的价值,为上层应用提供丰富的数据服务.数据中台在日常运行中通过多种方式记录系统的运行状态,而日志的产生是因为系统会在运行时信息记录,一般以静态文本和自由文本组合的形式存储.日志包含丰富的数据,允许开发人员和数据中台管理人员了解系统运行状态,同时可以通过日志数据对系统进行管理与诊断[1],比如通过统计信息进行分析[2-3]、保证应用程序的安全性[4-6]、对性能异常进行识别[7-8],或者在系统出现错误与崩溃后进行诊断[9-12].通过以上可以看出,日志内含有巨大的价值,因此现在最大的挑战就是如何挖掘日志与分析日志的内容[13].在物联网和云计算发展迅速的今天,系统每天产生的日志条目日益增多.以国家电网大数据中心为例,大数据中心每天传输几百 TB 的数据并生成上万条日志消息记录.此外,大数据中心对于数据的应用不同导致日志格式存在差异,使得分析日志更为复杂.面对海量的数据和日志格式的差异,手动处理已无法满足日志处理的需求.

典型的日志挖掘由 3 个步骤组成:日志收集、日志解析和日志分析[14].日志收集包含描述系统状态和运行时信息的原始日志数据.日志数据中每个日志条目包括一条消息,消息中包含描述某个事件的自由形式自然语言文本.在日志收集后,日志解析会预处理原始日志并提取出结构化数据.日志分析将结构化数据作为该步骤的输入数据,通过将数据编码为数字特征向量,实现异常检测[15-18]、模型推理[19-20]、故障定位[21]、故障预测[22]、信息安全[23-24]、应用互动[25-26]等的日志分析工作.

日志解析作为日志挖掘的重要步骤,工业界和学术界都对此进行了广泛的研究和探索,形成了大量的解析方法和工业解决方案.工业界提供了工业解决方案,如 Splunk、ELK、Logentries,现如今,文本搜索功能与通过机器学习进行的分析能力已经实现[27].学术界开发了日志解析算法,算法能够实现数据预处理,将原始日志数据转换为自动分析技术所需的结构化事件.值得注意的是,在工业界和学术界提供的解决方案背后,尚

收稿日期:2022-12-03;**修回日期**:2023-06-09.

基金项目:国家电网有限公司大数据中心项目(SGSJ0000HGJS2200037).

作者简介(通信作者):金铭(1992—),女,甘肃定西人,国家电网有限公司大数据中心高级技师,研究方向为数据监测与数据管理,E-mail:420519149@qq.com.

未有标准化的评估手段对这些方案进行评估,实现方案之间的对比.

本文将重点对日志解析技术、日志解析技术的评估进行详细介绍.

1 日志解析技术

原始日志文件中的每条日志都表示一个特定事件.日志通常由日志头和与该日志头相关联的事件组成.代码定义了输出日志的框架,框架一般包含结构化内容和自由文本两部分.结构化内容是日志头的字段,该字段包括时间戳、严重性级别和软件组件等数据,很容易提取和解析[28-29].自由文本内容是与日志头相关联的事件,通常由不同的字符串和格式字符串串联组成,一般没有"结构化"的格式,因此难以提取和解析,这也是日志解析技术的研究重点.自由文本内容一般由静态字段和动态字段组成,动态字段是在运行时分配的变量.静态字段是文本消息,它们不会随着事件的发生而变化,用来表示日志消息的事件类型.日志字段可以用任何分隔符分隔,例如空格、括号、逗号、分号等.

日志解析技术致力于提取并减少日志文件中的日志条目,提取日志中自由文本内容包含的信息.通过将静态字段和动态字段分离,用特定符号取代动态字段(通常用 *),并将每个原始日志消息转换为唯一的事件类型,该事件类型包含所有的相同事件[30].图 1 对日志的组成及解析后的事件进行了事件描述.日志第一行是日志头,包含了时间戳、详细级别和组件三部分内容;第二行是自由文本.自由文本经过解析后形成事件模板.

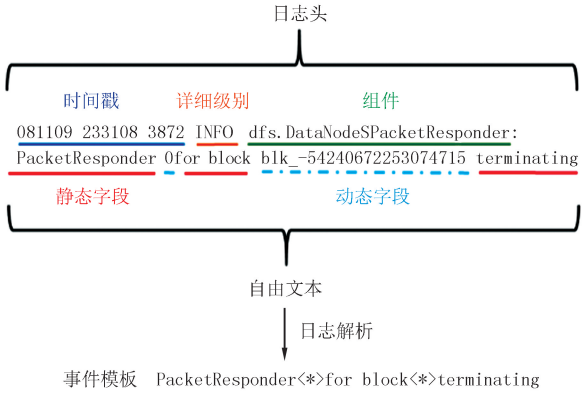

图1 日志组成及解析示例

Fig.1 Log composition and parsing example

下面将对工业界和学术界的日志解析技术进行详细阐述.

1.1 工业界日志解析技术

工业生产中会产生大量的日志,但是缺乏有效的日志分析工具将日志转换为有价值的数据.FU 等[31]通过源解析的方法对微软两个大型工业系统日志进行了分析,帮助开发人员进行决策.PECCHIA 等[32]提出了对关键工业领域中事件记录实践的测量研究,帮助开发人员重新设计任务的优先级排序.CHEN 等[29]通过研究发现包含日志消息的错误报告比不包含日志消息的错误报告需要更长的时间来解决,日志更新的较高部分用于提高日志的质量,而不是与功能实现的共同更改.除此以外,工业生产的特殊情况要求日志解析工具能够自动解析,这也是工业界目前较为流行产品的特点.最近自动日志解析如今已经作为一个关键组件,成了部分新产品的一个用于吸引用户的特点.但是目前自动日志解析只能针对常见日志类型.

1.2 学术界日志解析技术

日志解析算法存在一些具有实际意义的关键特性.日志解析算法可以按照不同的特性进行分类,如按照解析的场景可分为在线式、离线式,按照解析依据可分为源代码解析和日志解析,按照解析技术可分为频繁模式挖掘、聚类、迭代划分、最长公共子序列、解析树、进化算法、自然语言分析和其他启发式方法.从解析依据两个方面简要总结了这些日志解析算法使用的技术.图 2 是目前工业界和学术界在日志解析算法方面解析方案的对比.

1.2.1 基于源代码的解析

源代码是日志的"模式".系统信息会被控制台以自由文本的格式进行记录,不同控制台日志的内容、形式差别很大,事实上,日志消息的生成是由于系统源码中比较小的一组日志打印语句生成的,因此它们是非常结构化的.这意味着通过源代码分析更容易明确日志输出的格式、内容,对于日志中没有记录的日志消息类型,源代码分析也可以将其展现.基于这种思路,学者们提出一种基于源码分析来恢复日志的继承结构.

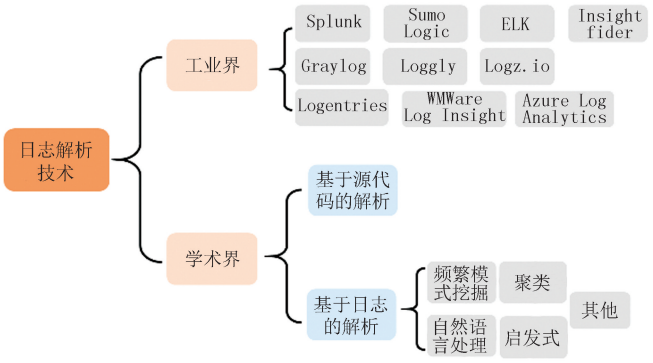

图2　日志解析方案

Fig.2　Log parsing schemes

XU 等[33]最早给出了源代码解析的解决方案,方案主要包括以下 4 个步骤:(1)日志解析.日志解析过程主要由两个步骤组成,一是静态源代码解析,二是运行时日志解析.静态源代码分析的输入是程序源和记录器类的名称.最重要的一步是生成程序代码的抽象概念句法知识树;第二步,在所有的类中枚举调用 toString,然后在调用中继续检查 string 类型文件格式的语法结构,然后就得以在消息自定义模板中接收到函数的参数类别后进行推断,用一些自定义的模板替换与这个类别有所关联的信息,最后递归执行此操作.等到所有的自定义模板都只插入原始类别后停止.(2)建立特性.根据所抽取的数据,选取适当的变量及分组的相关性,构建出相应的特征矢量.XU 等[33]构造了一个状态比矢量和消息技术矢量特性.(3)异常的探测.利用了一种特殊的识别技术,对各特征矢量进行了识别.利用主成分分析(Principal Component Analysis,PCA)中的异常探测技术对异常进行了检测.PCA 是一种不需要任何监督的学习方法,它可以在不需要预先手动操作的情况下,自行选取和调节.(4)视觉效果.为便于系统整合和操作人员更好地了解 PCA 的异常,将分析的结果可视化到决策树中,并以与系统集成者和操作者所熟知的事件处理法则相似的方式,对问题进行更细致的分析.值得注意的是,由于源代码的私密性和安全性问题,导致源代码难以获取,相比之下,系统日志具有相对容易获取、数据量大、信息多等特点.因此,研究人员通过各种技术手段开展了基于日志文件的日志解析研究[34-35].

1.2.2　基于日志的解析

基于日志的解析按照解析技术可归为以下 5 类:频繁模式挖掘、聚类、启发式、自然语言处理、其他方式.事实上,部分日志解析算法可以互相归类,如 HE 等[36]可以归为聚类,也可以归为启发式(本文将其归为启发式).在将算法归类为启发式时参考的依据是该算法中是否应用了启发性假设,部分文献在划分聚类时参考的依据是是否使用相似度或文本距离公式来分辨该算法是基于聚类还是基于启发式,在这种分类下,启发式严格意义上只有 IPLoM 和 POP 两种算法.在此,主要遵循 ZHU 等[27]给出的分类方法对算法分类.表 1 列出了 5 种解析技术及其包含的解析算法、算法提出的时间、算法解析模式以及算法的解析思路.这些日志解析算法都旨在实现自动日志解析.

频繁模式挖掘.数据频繁出现的一组项,称之为频繁模式.该方法对日志的解析较为简单,给定阈值,超过阈值视为常量,低于阈值的视为参数.日志解析后形成的事件模板的确定是根据一组经常出现在日志中的常量令牌.频繁模式挖掘的主要方法包括 SLCT[37]、LFA[38]、LogCluster[39]和 FT-tree[40],这些日志解析器均为离线的方法.这一类算法都有着相似的解析过程:(1)首先进行多次日志数据遍历;(2)其次在每次遍历的同时进行频繁项集(例如令牌和令牌位置对)构建;(3)然后日志消息会被分配至若干个集群内;(4)最后从每个集群中提取出各事件的模板.最早的日志解析方法为 SLCT 方法[37],同时 SLCT 也是首个应用频繁模式挖掘来进行日志解析的算法.对于 LogCluster,事实上 LogCluster 是 SCLT 的扩展,对令牌位置的移动具有鲁棒性,对可能产生的过拟合采取了两种方法来处理[39].FT-tree 采用了独创的 FT-tree 的数据结构进行解析,将单词按照出现频率高低进行排序并形成列表,形成列表后插入到 FT-tree 中,频率高的单词更靠近根节点,频率低的单词更远离根节点,“剪枝”后剩下的就是常量[40].频繁模式挖掘的日志解析方法效率很高,但是存在一个比较普通的问题:判断频繁词的阈值难以确定.在实际日志中,某些代码打印的次数非常少,某些代码打印的次数非常多,这会导致该算法将打印次数少的代码生成日志中的常量认为是参数.LFA 在该问题上表现良好,能够解析发生次数两次以上的日志,但是对于只发生一次的无法检测.

聚类.单个日志消息无法分辨日志的事件,而真正可以复原事件“原貌”的为一组日志消息形成的日志模板.因此,日志解析问题可以以日志消息的聚类问题为模型来解决.现如今有 3 种离线方法(即 LogSig[41]、

LogMine[42]和 LKE[43])和 2 种在线方法(即 LenMa[44]和 SHISO[45])可以将聚类算法应用于日志解析.基于聚类的日志模式解析一般分为两步:(1)通过文本相似度或文本距离,将日志进行分组;(2)一条新日志需要解析时,计算该日志与所有聚类簇的聚类中心之间的相似度,找到相似度最大的聚类簇.若相似度满足阈值要求,则将待解析日志并入该聚类簇中,并更新聚类中心,若没有满足阈值要求的聚类簇,则将待解析日志作为聚类中心,创建新的聚类簇.LogMine 和 LKE 为分层聚类. LogMine 先生成事件模板,再将日志消息自下而上分组为集群,LKE 基于两两日志消息之间进行加权编辑距离,LogSig 是一种基于消息签名的算法,通常运用在日志消息聚类到预定义数量的簇中.SHISO 和 LenMa 都是在线方法,它们以类似的流方式解析日志.对于每个新出现的日志消息,解析器首先计算其与现有日志集群的代表性事件模板的相似度.如果匹配成功,日志消息将被添加到现有的集群中,否则将创建一个新的日志集群.然后相应地更新相应的事件模板.大部分聚类算法中,阈值是经过试验选择较好的解析结果设定的,解析的日志与样本日志不同,阈值也需要改变.目前也有自动确定阈值的方法,如 LKE 提出通过 k-means 聚类获得相似度阈值,但是自动确定阈值的方法仍需进一步验证.从聚类方法可以看到,聚类具有很大的多样性,这是因为聚类的依据是相似度,可以采用多种方式来定义相似度.

表 1 日志解析技术总结

Tab. 1 Summary of log parsing technology

解析技术	算法名称	发表年份	解析模式	解析思路
频繁模式挖掘	SLCT	2003	离线	频繁词聚类
	LFA	2010	离线	频繁项集
	LogCluster	2015	离线	频繁词聚类
	FT-tree	2017	离线	FT-tree 数据结构
	LogHound	2008	离线	频繁项集
启发式	AEL	2008	离线	启发式策略, 相似度
	Drain	2017	在线	启发式策略,解析树
	IPLoM	2012	离线	启发式策略
	POP	2018	离线	启发式策略
聚类	LKE	2009	离线	文本距离
	LogSig	2011	离线	文本相似度
	LogMine	2016	离线	解析树,层次聚类
	SHISO	2013	在线	解析树,层次聚类
	LenMa	2016	在线	特征相似度
自然语言处理	Logram	2020	在线	n-gram 频率
	Spell	2016	在线	最长公共子序列
其他	MoLFI	2018	离线	进化算法

启发式.启发式是一种基于直观或经验的方法,它往往只给出一些指导信息,而非解决问题的直接方法(如相似度等),换句话说,启发式是有依据的猜测、实际经验估计或是常识.日志文本与一般的文本数据不同,往往有一些独有的特征,比如说同一组日志的长度相同、同一组日志可能拥有某一个固定位置的特殊单词等等.因此,一些工作(即 Drain[46]、POP[47]、IPLoM[48])提出了一种根据启发式的日志解析方法.AEL 中是根据对比日志的长度及 key-value 对的对数这样的启发性策略给日志进行分组,然后再在组别下计算相似度进行聚类[46].IPLoM 中用到的启发式策略有 3 个:根据日志长度分组、根据单词位置分组、根据双射关系分组,通过迭代分区的方式来对日志进行归类解析[47].Drain 应用固定深度树的方法来表示日志结构,根据对比日志的长度以及日志的前几个单词这样的启发式策略来进行分组.当有一条日志需要解析时,会根据上述分组策略向下搜索,直到存储着该组别中聚类簇的叶子节点,然后再根据相似度计算结果更新聚类中心或者创建新的聚类簇[36].POP 中所用的启发式策略有两个:根据日志长度分组、根据单词位置分组[48].用一种

启发式策略进行日志解析难以覆盖日志中所有的情况,因此采用多种启发式算法能够利用日志的各种特性,从而在许多情况下表现良好.

自然语言处理.系统开发人员设置的日志消息通常类似于特定领域中使用的一种自然语言.KOBAYASHI 等[49]认为,处理日志数据可以被视为一种特殊类型的自然语言处理问题,因此许多自然语言处理方法可以用于分析数据和提取模板库.LI 等[50]基于系统的"最小先验知识"原则,选择了几种经典的自然语言模型对日志解析进行了实现.该方法主要分为三步:(1)通过快速 n-gram 语言模型进行琐碎类型的消息的预过滤;(2)使用潜在狄利克雷分配(Latent Dirichlet Allocation,LDA)将日志消息里所有还没有被标记的单词表示的大型存储库中构建语义空间;(3)运用标准前馈多层感知器执行三向分类,同时保证在最少人工监督的情况下.实验结果表明,该方法在速度和精度方面取得了良好的性能.此外,DAI 等[51]提出了 Logram 日志解析算法.

其他.有些方法难以归类到以上 4 种分类当中,例如,DU 等[52]利用最长公共子序列算法(Longest Common Subsequence,LCS)以流方式解析日志.读取新日志消息时,通过 LCS 检测其是否与现有日志类型模板"匹配",若匹配,则将其归于该现有日志类型模板,若不匹配,则创建新的日志类型模板.MESSAOUDI 等[53]提出了 MoLFI.MoLFI 方法是把日志解析变成一个多目标的优化问题,然后对此优化问题进行进化算法求解.表 1 对当前的日志解析进行了简单的汇总.

2　日志解析算法评估

随着日志解析的不断发展,解析思路也从最初的频繁模式挖掘不断开拓,但是使用者仍未意识到不同日志解析算法的优势,也不了解其对后续日志挖掘任务的影响.使用者经常重新编写甚至重新设计新的日志解析算法,这将耗时且多余.主要原因有两个方面:缺乏统一的衡量标准和统一的算法评估日志集.因此,对日志解析算法进行统一评估能够有效解决这个问题,方便使用者根据评估结果在处理不同格式的日志时选择合适的算法.该部分将从算法评估、统一评估两个方面详细阐述日志解析算法的评估.算法评估阐述算法提出时与其他算法的对比结果,统一评估描述算法在评估标准、数据集方面的进展.

2.1　算法评估

SLCT 作为最早的日志解析算法,首次将频繁模式挖掘应用于日志解析中[37],后续提出的诸多日志解析算法(AEL、LogHound、LFA、IPLoM、SHISO、LogCluster、POP)均与 SLCT 进行了比较,结果表明后续算法在识别日志模板、检测少量事件方面拥有优势,AEL 在召回率和检测精度方面优于 SLCT.LogSig[42]算法构造了目标函数 F 描述所有组中公共对的总数,如果一个组有更多的公共对,它更有可能有更长的公共子序列,该组的评估分数会更高.在 ThunderBird 日志上的平均 F 测量值略低于 IPLOM 算法.在其他 4 种日志上 Drain 与 LKE、IPLoM 两种离线日志解析方法和 SHISO、Spell 两种在线日志解析方法在 6 种数据集上运行并对运行结果进行比较,数据表明 Drain 的精度明显高于 LKE 和 SHISO,其效率高于其他 4 种被比较的方法 50%～80%;Drain 在 HDFS 数据集上获得了几乎最优的异常检测性能.MOLFI 与 Drain 和 IPLoM 进行了对比,结果表明 MOLFI 精度高,设置参数简单(参数会影响性能),日志格式识别转为多目标优化,有效性不受模板数量的影响.在面对更大规模的种群,MOLFI 执行时间急剧增加,可以通过预处理中去重降低执行时间.Logram 选择了解析效果最好的几种算法(Drain、Spell、AEL、IPLoM、Lenma)进行了对比,Logram在 17 个数据集上平均解析精确度为 0.825,比解析精确度最高的 Drain(0.748)高,8 个数据集中的解析精度高于 0.9,随机抽取 5 种日志不同大小的数据块(300 kB、1 MB、10 MB、100 MB、500 MB 和 1 GB)进行了解析时间对比,Logram 快 1.8～5.1 倍,可以从较小的日志中提取词典且有较好的一致性.LKE、Lenma、LogMine 这 3 种算法未与其余算法进行对比.

不同算法的对比结果反映了两个问题,算法在与其他算法进行评估时,对比条件不统一,存在计算机的硬件配置不一致、解析的日志集不同等问题.此外,算法在对比时缺乏统一的指标,检验模板数量、召回率、检测效率(运行效率)、准确率、目标函数等是对比时提到的指标,然而这些指标部分或全部出现在上述算法中,这导致结果缺乏可比性和全面性.两方面的结果表明,算法的评估需要建立统一的指标和数据集.

2.2 统一评估

在日志解析算法中,虽然没给出统一的标准,但是在评估算法时已经对部分指标达成了共识,如:解析效率、准确率.部分算法对日志解析效果评估的标准给出了自己的定义.TANG 等[42]在 LogSig 算法中提出目标函数 F 作为日志解析算法的标准,HAMOONI 等[43]在 LogMine 中定义了算法应该具有的 4 个理想属性:无监督、异构性、效率和可扩展性,HE 等[36]在 Drain 中提出了调整算法参数的挑战,以及考虑参数调整工作的重要性.

此外,更多学者在建立日志解析算法的统一标准上提出了自己的见解.JIANG 等人[54]定义了关于日志解析算法评估的 4 个方面:可解释性、系统知识、努力和覆盖.DEISSENBOECK 等[55]指出,在评估质量模型中,质量方面的评估是定性或定量的,对不可直接测量的方面进行了定性描述.MIZUTANI[44]指出了立即提取日志消息对于解决问题的重要性.MAKANJU 等[56]传达了高覆盖率和发现罕见事件的重要性.DIANA 等[30]第一次广泛研究了日志解析算法和基于质量模型的日志解析算法推荐程序,比较 17 种算法后的结果表明没有一个算法可以满足所有指标要求.

数据集由于机密问题,真实世界的日志数据集很难在公共场合收集到,这阻碍了新日志分析技术的研究、开发和评估.在建立统一的日志数据集上,HE 等人[13]使用准确性和效率作为质量评估的标准,对 4 种算法在 5 个数据集上进行解析以此评估算法的性能,5 个数据集具有超过一千万原始日志消息.ZHU 等人[27]在 loghub 日志数据集上测量了 13 个日志解析算法的性能,loghub 日志数据集内总共有 16 个不同系统的日志,这 16 个系统的日志包括操作系统、分布式系统、服务器应用程序、移动系统和超级计算机,包含总计 4.4 亿条日志消息,大小达 77 GB,是目前最大的日志数据集.16 个日志集(如 Hadoop 、HDFS、BGL)中的一些是来自以前的研究版本的生产日志,而其他的(如 Health App、Zookeeper、Android、Spark)则是从他们实验室中的真实系统中收集的.2019 年,来自行业(35%)和学术界(65%)的 150 多个组织下载了超过 1 000 次 loghub 数据集.ZHU 等[27]在 16 个系统生成的相同日志数据集上评估了各种日志解析算法的精度,并表明一些日志解析算法在某些文件上的精度较高,但在其他文件上的精度较低.

表 2 对日志解析算法评估结果进行了汇总,汇总结果包括算法在提出时对比的算法、评估的数据集、统一评估时的解析效率和解析精确度.

综上所述,算法评估在不同算法之间进行了对比,但是缺乏统一的评估标准,使得结果难以对比,统一评估的结果一定程度上可以填补工业界和学术界之间的差距.工业界无需研究最先进的日志解析算法,可以专注于根据日志解析算法在公开数据集和评估指标的表现选择合适的算法进行测试,并决定哪个是最适合的算法.最大的公开数据集 loghub 虽然涵盖了 16 个不同系统的日志,但是相比于广大的日志格式还是太少,从业者最好在自己的日志数据集上进行评估[30,57].

3 未来研究方向与挑战

本文在对日志解析算法和算法评估的文献进行分析和综述的基础上,列出了一些主要的研究方向和未来研究的困难.

(1)统一的评估指标.基于各种模型,学者提出了各种评估指标,但是目前指标缺乏普遍的认可.未来需要对评估指标进行检验和认证,力求学术界和工业界对评估指标达成共识.

(2)统一的日志数据集.可用的日志数据集种类多,而且所有类型的日志之间存在非常大的差异,loghub 提供了 16 种不同的数据集,这为建立统一的日志数据集提供了很好的演示.但是该数据集存在部分日志数据量较小、日志结构不够丰富的问题,因此建立一个“通用”log 数据集,在该数据集上对所有日志解析算法进行评估,这将简化它们的比较.

(3)算法的资源利用率.算法在评估中,提到了算法的运行效率,资源利用是评估效率的一个重要方面,目前算法仍然缺乏这方面的评估和报告.

(4)日志解析算法的开发.自然语言算法为日志解析带来了新的思路,也让日志解析的效率得到了提升,未来如何让其他领域的知识与日志解析算法得到结合,仍需要进一步探索.

(5)中文日志的解析和提取.日志解析算法的默认前提是解析英文日志,随着中国云计算等能力的加强,日志数量也迅速增多,基层工作人员需要中文日志以减少处理日志的成本,目前企业已开始在日志文件中加入中文,而目前的日志解析算法处理含有中文字符的日志效果不佳,这需要新的日志解析算法来解决这个问题.

(6)工业界与学术界的沟通.工业界和学术界存在交流的鸿沟,软件工程师不了解学术界开发的所有日志解析技术的特点,也难以负担搜索、实验算法所花费的大量时间;算法的解析思路不同,导致软件工程师在学习和应用时需要花费大量的时间;算法最初提出的目的是改善准确率、解析时间等问题,这可能与实际生产中的期望差距较大.因此,学术界通过统一的评估标准和日志数据集可以减少交流障碍,可以将算法更多地应用到工业界,工业界可以提供大型的工业日志文件来帮助培训和改进算法.

表 2　日志解析算法评估结果总结

Tab. 2　Summary of evaluation results of log parsing algorithms

算法名称	对比算法	对比日志	效率	解析精确度
SLCT	—	—	高	低
AEL	SLCT	某大中型企业应用程序,LoadSim,Blue Gene/L logs	高	高
MoLFI	SLCT,LogHound,Teiresias	HPC,SysLog,Windows,Access,Error,System,Rewrite	高	高
LKE	AEL	Hadoop,SILK	低	高
LFA	SLCT	Virtual Computing Lab	高	中
LogSig	VectorMod	FileZilla,ThunderBird,PVFS2,Apache Error,Hadoop	中	中
SHISO	—	HDFS,Hadoop	高	中
LogCluster	SCLT	来自欧盟国家的国家关键信息基础设施的大型机构的 6 条日志	高	中
LenMa	SHISO	公安日志共享站点,WIDE 项目操作的虚拟机监控程序集群,自己实验室的服务器集群	中	高
LogMine	HLAer	—	中	中
Spell	IPLoM,CLP	Los Alamos HPC log,BlueGene/L log	高	高
Drain	LKE,IPLoM,SHISO,Spell	BGL,HPC,HDFS,Zookeeper,Proxifier	高	高
MoLFI	Drain,IPLoM	HDFS,BGL,Zookeeper,HPC,Proxifier,some proprietary dataset	低	高
Logram	Drain,Spell,AEL,IPLoM,Lenma	Loghub 数据集	高	高
LogHound	—	SysLog,Windows,Access,Error,System,Rewrite,HPC	高(除在 HPC 上)	—
POP	SLCT,IPLoM,LKE,LogSig	BGL,HPC,HDFS,Zookeeper,Proxifier	高	高
FT-Tree	Signature Tree,STE,LogSimilarity	tier-1 云服务器日志	高	—

4　总　结

数据中台的目的是给客户提供效率更高的服务.为了实现此目标,数据中台会通过用数据技术采集海量数据,并对其计算后进行存储,最后处理一系列过程,同时会对标准与口径进行统一化处理,从而可以逐渐形成全区域级别、可重复使用的数据资产中心和数据存储能力中心,进而可以形成大数据的资产层,以达到数据中台的作用.数据中台在日常运行中通过多种方式记录系统的运行状态,日志是数据中台运行时信息记录的一种方式,一般以静态文本和自由文本组合的形式存储.日志包含丰富的数据,允许开发人员和数据中台管理人员了解系统运行状态,支持系统的管理和诊断任务日志作为记录系统运行状态的重要信息资源,可为故障诊断、性能诊断、系统安全、预测和分析提供支持.日志挖掘中的日志解析,是实现日志管理的重要步骤.日志解析可以结构化提取原始日志中的信息.在日志解析方法中,有工业界和学术界两派解决方案.工业界解决方案能够基本满足工业生产中的需要,学术界可分为基于源代码解析、基于日志解析两大类.基于日志解析可大致归为聚类、频繁模式挖掘、启发式、自然语言处理、其他 5 类.目前,日志解析算法在评估时的条件

不一致,导致评估效果难以衡量,统一的评估标准和日志数据集正在逐步建立,有利于缩小工业界和学术界的鸿沟.

随着云计算等技术的发展,日志文件将越来越大,合适的解析算法将为后续的日志分析提供便利.在后续的研究中,不仅要关注工业界、学术界日志解析技术的发展,还需要促进工业界、学术界的交流,使得算法能够针对工业生产的痛点提出解决措施并提升生产效率,保证系统正常运行.

参 考 文 献

[1] ZHU J M,HE P J,FU Q A,et al.Learning to log:helping developers make informed logging decisions[C]//2015 IEEE/ACM 37th IEEE International Conference on Software Engineering.[s.l.]:IEEE,2015.

[2] OLINER A,GANAPATHI A,XU W.Advances and challenges in log analysis[J].Communications of the ACM,2012,55(2):55-61.

[3] BUSANY N,MAOZ S.Behavioral log analysis with statistical guarantees[C]//Proceedings of the 38th International Conference on Software Engineering.New York:ACM,2016:877-887.

[4] MIRANSKYY A,HAMOU-LHADJ A,CIALINI E,et al.Operational-log analysis for big data systems:challenges and solutions[J].IEEE Software,2016,33(2):52-59.

[5] LANDAUER M,SKOPIK F,WURZENBERGER M,et al.System log clustering approaches for cyber security applications:a survey[J].Computers & Security,2020,92:101739.

[6] OPREA A,LI Z,YEN T F,et al.Detection of early-stage enterprise infection by mining large-scale log data[C]//2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks.[s.l.]:IEEE,2015.

[7] HE S L,ZHU J M,HE P J,et al.Experience report:system log analysis for anomaly detection[C]//2016 IEEE 27th International Symposium on Software Reliability Engineering(ISSRE).[s.l.]:IEEE,2016:207-218.

[8] NAGARAJ K,KILLIAN C,NEVILLE J.Structured comparative analysis of systems logs to diagnose performance problems[C]//Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation.New York:ACM,2012:26.

[9] KITCHENHAM B,PEARL BRERETON O,BUDGEN D,et al.Systematic literature reviews in software engineering-A systematic literature review[J].Information and Software Technology,2009,51(1):7-15.

[10] YUAN D,MAI H H,XIONG W W,et al.SherLog:error diagnosis by connecting clues from Run-time logs[C]//Proceedings of the fifteenth International Conference on Architectural support for programming languages and operating systems.New York:ACM,2010:143-154.

[11] OLINER A,STEARLEY J.What supercomputers say:a study of five system logs[C]//37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks(DSN'07).[s.l.]:IEEE,2007.

[12] HUANG L,KE X D,WONG K,et al.Symptom-based problem determination using log data abstraction[C]//Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research.New York:ACM,2010:313-326.

[13] HE P J,ZHU J M,HE S L,et al.An evaluation study on log parsing and its use in log mining[C]//2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks(DSN).[s.l.]:IEEE,2016:654-661.

[14] SUNHARE P,CHOWDHARY R R,CHATTOPADHYAY M K.Internet of Things and data mining:an application oriented survey[J].Journal of King Saud University-Computer and Information Sciences,2022,34(6):3569-3590.

[15] CHEN R,ZHANG S L,LI D W,et al.LogTransfer:cross-system log anomaly detection for software systems with transfer learning[C]//2020 IEEE 31st International Symposium on Software Reliability Engineering(ISSRE).[s.l.]:IEEE,2020.

[16] LE V H,ZHANG H Y.Log-based anomaly detection without log parsing[C]//2021 36th IEEE/ACM International Conference on Automated Software Engineering(ASE).[s.l.]:IEEE,2021.

[17] CAVALLARO C,RONCHIERI E.Identifying anomaly detection patterns from log files:a dynamic approach[M]//Computational Science and Its Applications-ICCSA 2021.Cham:Springer International Publishing,2021:517-532.

[18] DU M,LI F F,ZHENG G N,et al.DeepLog:anomaly detection and diagnosis from system logs through deep learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.New York:ACM,2017:1285-1298.

[19] AMAR H,BAO L F,BUSANY N,et al.Using finite-state models for log differencing[C]//Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.New York:ACM,2018:49-59.

[20] BESCHASTNIKH I,BRUN Y,ERNST M D,et al.Inferring models of concurrent systems from logs of their behavior with CSight[C]//Proceedings of the 36th International Conference on Software Engineering.New York:ACM,2014:468-479.

[21] CHEN A R.An empirical study on leveraging logs for debugging production failures[C]//2019 IEEE/ACM 41st International Conference on Software Engineering:Companion Proceedings(ICSE-Companion).[s.l.]:IEEE,2019:126-128.

[22] AMAR A,RIGBY P C.Mining historical test logs to predict bugs and localize faults in the test logs[C]//2019 IEEE/ACM 41st Interna-

tional Conference on Software Engineering(ICSE).[s.l.]:IEEE,2019:140-151.

[23] COPSTEIN R,SCHWARTZENTRUBER J,ZINCIR-HEYWOOD N,et al.Log abstraction for information security:heuristics and reproducibility[C]//Proceedings of the 16th International Conference on Availability,Reliability and Security.New York:ACM,2021:1-10.

[24] AIT EL HADJ M,KHOUMSI A,BENKAOUZ Y,et al.Efficient security policy management using suspicious rules through access log analysis[M]//Networked Systems.Cham:Springer International Publishing,2019:250-266.

[25] LOCKE S,LI H,CHEN T H P,et al.LogAssist:assisting log analysis through log summarization[J].IEEE Transactions on Software Engineering,2022,48(9):3227-3241.

[26] KORZENIOWSKI Ł,GOCZYŁA K.Discovering interactions between applications with log analysis[C]//Annals of Computer Science and Information Systems","Proceedings of the 17th Conference on Computer Science and Intelligence Systems.[s.l.]:IEEE,2022:861-869.

[27] ZHU J M,HE S L,LIU J Y,et al.Tools and benchmarks for automated log parsing[C]//2019 IEEE/ACM 41st International Conference on Software Engineering:Software Engineering in Practice(ICSE-SEIP).[s.l.]:IEEE,2019:121-130.

[28] YUAN D,PARK S,ZHOU Y Y.Characterizing logging practices in open-source software[C]//2012 34th International Conference on Software Engineering(ICSE).[s.l.]:IEEE,2012:102-112.

[29] CHEN B Y,MING Z.Characterizing logging practices in Java-based open source software projects-a replication study in Apache Software Foundation[J].Empirical Software Engineering,2017,22(1):330-374.

[30] DIANA D,PETRILLO F,GUÉHÉNEUC Y G,et al.A systematic literature review on automated log abstraction techniques[J].Information and Software Technology,2020,122:106276.

[31] FU Q,ZHU J M,HU W L,et al.Where do developers log? an empirical study on logging practices in industry[C]//Proceedings of the 36th International Conference on Software Engineering.New York:ACM,2014:24-33.

[32] PECCHIA A,CINQUE M,CARROZZA G,et al.Industry practices and event logging:assessment of a critical software development process[C]//2015 IEEE/ACM 37th IEEE International Conference on Software Engineering.[s.l.]:IEEE,2015:169-178.

[33] XU W,HUANG L,FOX A,et al.Detecting large-scale system problems by mining console logs[C]//Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles.New York:ACM,2009:117-132.

[34] KORZENIOWSKI Ł,GOCZYŁA K.Landscape of automated log analysis:a systematic literature review and mapping study[J].IEEE Access,2022,10:21892-21913.

[35] HE S L,HE P J,CHEN Z B,et al.A survey on automated log analysis for reliability engineering[J].ACM Computing Surveys,2021,54(6):130.

[36] HE P J,ZHU J M,ZHENG Z B,et al.Drain:an online log parsing approach with fixed depth tree[C]//2017 IEEE International Conference on Web Services(ICWS).[s.l.]:IEEE,2017:33-40.

[37] VAARANDI R.A data clustering algorithm for mining patterns from event logs[C]//Proceedings of the 3rd IEEE Workshop on IP Operations & Management(IPOM 2003).[s.l.]:IEEE,2003:119-126.

[38] NAGAPPAN M,VOUK M A.Abstracting log lines to log event types for mining software system logs[C]//2010 7th IEEE Working Conference on Mining Software Repositories(MSR 2010).[s.l.]:IEEE,2010:114-117.

[39] VAARANDI R,PIHELGAS M.LogCluster-A data clustering and pattern mining algorithm for event logs[C]//2015 11th International Conference on Network and Service Management(CNSM).[s.l.]:IEEE,2016:1-7.

[40] ZHANG S L,MENG W B,BU J H,et al.Syslog processing for switch failure diagnosis and prediction in datacenter networks[C]//2017 IEEE/ACM 25th International Symposium on Quality of Service(IWQoS).[s.l.]:IEEE,2017:1-10.

[41] FU Q,LOU J G,WANG Y,et al.Execution anomaly detection in distributed systems through unstructured log analysis[C]//2009 Ninth IEEE International Conference on Data Mining.[s.l.]:IEEE,2009:149-158.

[42] TANG L,LI T,PERNG C S.LogSig:generating system events from raw textual logs[C]//Proceedings of the 20th ACM international conference on Information and knowledge management.New York:ACM,2011:785-794.

[43] HAMOONI H,DEBNATH B,XU J W,et al.LogMine:fast pattern recognition for log analytics[C]//Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.New York:ACM,2016:1573-1582.

[44] MIZUTANI M.Incremental mining of system log format[C]//2013 IEEE International Conference on Services Computing.[s.l.]:IEEE,2013:595-602.

[45] SHIMA K.Length matters:clustering system log messages using length of words[EB/OL].[2022-11-16].https://arxiv.org/abs/1611.03213

[46] JIANG Z M,HASSAN A E,FLORA P,et al.Abstracting execution logs to execution events for enterprise applications(short paper)[C]//2008 The Eighth International Conference on Quality Software.[s.l.]:IEEE,2008.

[47] MAKANJU A A O,ZINCIR-HEYWOOD A N,MILIOS E E.Clustering event logs using iterative partitioning[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.New York:ACM,2009:1255-1264.

[48] HE P J,ZHU J M,HE S L,et al.Towards automated log parsing for large-scale log data analysis[J].IEEE Transactions on Dependable

and Secure Computing,2018,15(6):931-944.

[49] KOBAYASHI S,FUKUDA K,ESAKI H.Towards an NLP-based log template generation algorithm for system log analysis[C]//Proceedings of The Ninth International Conference on Future Internet Technologies.New York:ACM,2014:1-4.

[50] LI G F,ZHU P J,CAO N,et al.Improving the system log analysis with language model and semi-supervised classifier[J].Multimedia Tools and Applications,2019,78(15):21521-21535.

[51] DAI H T,LI H,CHEN C S,et al.Logram:efficient log parsing using n-gram dictionaries[EB/OL].[2022-11-13].https://doi.org/10.48550/arXiv.2001.03038.

[52] DU M,LI F F.Spell:streaming parsing of system event logs[C]//2016 IEEE 16th International Conference on Data Mining(ICDM).[s.l.]:IEEE,2016.

[53] MESSAOUDI S,PANICHELLA A,BIANCULLI D,et al.A search-based approach for accurate identification of log message formats[C]//Proceedings of the 26th Conference on Program Comprehension.Gothenburg Sweden.New York:ACM,2018.

[54] JIANG Z M,HASSAN A E,HAMANN G,et al.An automated approach for abstracting execution logs to execution events[J].Journal of Software Maintenance and Evolution:Research and Practice,2008,20(4):249-267.

[55] DEISSENBOECK F,JUERGENS E,LOCHMANN K,et al.Software quality models:purposes,usage scenarios and requirements[C]//2009 ICSE Workshop on Software Quality.[s.l.]:IEEE,2009:9-14.

[56] MAKANJU A,ZINCIR-HEYWOOD A N,MILIOS E E.A lightweight algorithm for message type extraction in system application logs[J].IEEE Transactions on Knowledge and Data Engineering,2012,24(11):1921-1936.

[57] XIE X S,WANG Z,XIAO X H,et al.A confidence-guided evaluation for log parsers inner quality[J].Mobile Networks and Applications,2021,26(4):1638-1649.

Log parsing technology based on data platform

Jin Ming, Cui Shuo, Wen Yang, Bian Lin, Guo Xueliang, Feng Hanyu

(Big Data Center, State Grid Corporation of China, Beijing 100032, China)

Abstract: Data platforms are a mode of providing efficient services to customers by using data technologies. Logs are a way of recording system status in data platforms. They can support tasks such as fault diagnosis, performance optimization, system security, etc. Analyzing the log information is significant to daily operation and maintenance of platform. Log parsing is an important step in log mining. It transforms unstructured log text into structured data. This paper reviews the log parsing algorithms and evaluation methods, analyzes the solutions from industry and academia, summarizes the main categories and features of log parsing algorithms, compares the performance and effectiveness of different algorithms on different datasets. This paper finds that log parsing algorithms lack a unified standard and dataset, making it difficult to compare and verify the results. To address this issue, this paper suggests that future research should focus on establishing unified evaluation indicators and log datasets, promoting communication between industry and academia, and improving the applicability and reliability of log parsing algorithms. This paper has important reference value for the research in the field of log parsing.

Keywords: data platform; log parsing; log mining; algorithm evaluation

[责任编校 陈留院 赵晓华]