计算机科学
COMPUTER SCIENCE

基于BERT和弱行为轮廓的可解释性事件日志修复方法

李炳辉, 方欢, 梅振辉

引用本文

李炳辉, 方欢, 梅振辉.基于BERT和弱行为轮廓的可解释性事件日志修复方法[J]. 计算机科学, 2023, 50(5): 38-51.

LI Binghui, FANG Huan, MEI Zhenhui. Interpretable Repair Method for Event Logs Based on BERT and Weak Behavioral Profiles [J]. Computer Science, 2023, 50(5): 38-51.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220900030

基于 BERT 和弱行为轮廓的可解释性事件日志修复方法

李炳辉　方　欢　梅振辉
安徽理工大学数学与大数据学院　安徽 淮南 232001
安徽省煤矿安全大数据分析与预警技术工程实验室　安徽 淮南 232001
(kawaayii.1024@gmails.com)

摘　要　由异常值和缺失值导致的低质量事件日志在实际的业务流程中通常不可避免,低质量的事件日志会降低过程挖掘相关算法的性能,从而干扰决策的正确实施。在系统参考模型未知的条件下,现有方法在进行日志异常检测与修复工作中,存在需要人为设定阈值、不知预测模型学习何种行为约束以及修复结果可解释性较差的问题。采用遮掩策略的预训练语言模型 BERT 可以通过上下文信息自监督地学习文本中的通用语义,受此启发,提出了模型 BERT4Log 和弱行为轮廓理论,并结合多层多头注意力机制进行低质量事件日志的可解释修复。所提修复方法不需要预先设定阈值,仅需要进行一次自监督训练,同时该方法利用弱行为轮廓理论量化行为上的日志修复程度,并结合多层多头注意力机制实现对具体预测结果的详细解释。最后,在一组公开数据集上对方法性能进行评估,并与目前性能最优的研究进行对比分析,实验结果表明 BERT4Log 的修复性能整体优于对比方法,可以学习弱行为轮廓并实现修复结果的详细解释。

关键词:事件日志修复;弱行为轮廓;BERT;可解释模型;注意力机制
中图法分类号　TP391

Interpretable Repair Method for Event Logs Based on BERT and Weak Behavioral Profiles

LI Binghui,FANG Huan and MEI Zhenhui
School of Mathematics and Big Data,Anhui University of Science and Technology,Huainan,Anhui 232001,China
Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety,Huainan,Anhui 232001,China

Abstract　In practical business processes,low-quality event logs due to outliers and missing values are often unavoidable. Low-quality event logs can degrade the performance of associated algorithms for process mining,which in turn interferes with the correct implementation of decisions. Under the condition that the system reference model is unknown,when performing log anomaly detection and repair work,the existing methods have the problems of needing to manually set thresholds,do not understand what behavior constraints the prediction model learns,and poor interpretability of repair results. Inspired by the fact that the pre-trained language model BERT using the masking strategy can self-supervise learning of general semantics in text through context information,combined with attention mechanism with multi-layer and multi-head,this paper proposes the model BERT4Log and weak behavioral profiles theory to perform an interpretable repair process for low-quality event logs. The proposed repair method does not need to set a threshold in advance,and only needs to perform self-supervised training once. At the same time,the method uses the weak behavioral profiles theory to quantify the degree of behavioral repair of logs. And combined with the multi-layer multi-head attention mechanism to realize the detailed interpretation process about the specific prediction results. Finally,the performance of the proposed method is evaluated on a set of public datasets,and compared with the current research with the best performance. Experimental results show that the repair performance of BERT4Log is better than the comparative research,and at the same time,the model can learn weak behavioral profiles and achieve detailed interpretation of repair results.

Keywords　Event log repair,Weak behavioral profiles,BERT,Interpretable model,Attention mechanism

目前,许多组织都使用过程感知信息系统(Process Awareness Information System,PAIS)来管理其流程,同时也会使用 PAIS 来收集数据。通常这些收集的数据描述了用户在相关系统的执行过程中发生的所有事件,即产生系统的事件日志。事件日志对业务过程挖掘起到举足轻重的作用,其不仅可以用于模型发现和业务增强,还被大量应用于预测型

到稿日期:2022-09-05　返修日期:2023-01-26
基金项目:国家自然科学基金(61902002)
This work was supported by the National Natural Science Foundation of China(61902002).
通信作者:方欢(fanghuan0307@163.com)

流程监控(Predictive Process Monitoring，PPM)[1-2]。基于事件日志的过程挖掘方法，特别是其在 PPM 中的应用研究是目前备受关注的研究热点。

在数据挖掘领域，大多数分析与挖掘算法的性能和准确性等指标在很大程度上取决于基础数据的质量[3]，这同样适用于基于事件日志的过程挖掘算法[4]。然而，在实际的行业应用中，由于存在不完全可靠的记录方式或是出于节省存储空间的目的，事件日志中不可避免地会出现一定程度的异常，因而产生低质量的事件日志。这些日志会使得一些过程挖掘算法的性能下降甚至无法使用[5]。例如，从低质量事件日志中挖掘出的过程模型并不能真正地反映业务流程的真实情况，亦或者在 PPM 研究中，需要事先进行异常检测来丢弃包含异常或缺失的样本，这对于需要大量数据去训练 PPM 的预测模型来说非常不利。另外，大部分 PPM 研究重点关注预测模型的性能，或者仅尝试在非行为的视角下预测结果。因此，低质量事件日志的修复及行为视角的可解释过程是过程挖掘研究中的一项重要内容。

最近，预训练语言模型(Pre-training Language Model，PLM)在自然语言处理(Natural Language Processing，NLP)中取得了令人瞩目的成就，如 ELMo[6]，GPT[7]，BERT[8]等。这些 PLM 首先利用大量的语料库自监督地学习文本中的通用语义，之后将训练好的语言模型应用到不同的下游任务中。文献[2]利用 GPT-2 进行下一活动预测并取得了不错的准确度，但没有对预测结果进行解释。不同于 GPT 在一个方向上的预测，BERT 是一种掩码语言模型(Mask Language Models，MLM)，其在预训练过程中可以同时利用双向信息预测位于遮掩候选集中的活动，期间能够学习到文本中的语义并具有一定的纠错能力[8]。而低质量事件日志的修复任务同样需要纠错模型学习日志中的行为约束并利用该模型进行修复，BERT 本身的纠错能力也使其与日志修复任务天然地契合，并且注意力机制可以增强修复结果的可解释性。但据笔者所知，目前还没有相关研究将 BERT 或其他 PLM 用于低质量事件日志的修复工作中，并从行为的视角进行修复结果的可解释过程。

为此，本文提出了模型 BERT4Log，结合弱行为轮廓理论进行事件日志的可解释修复，整体的修复流程如图 1 所示。BERT4Log 采用独特的遮掩策略，实现在预训练期间学习并判断迹中被遮掩活动是否正确；并在微调阶段对含有不正确和缺失这两种噪音的日志进行修复。BERT4Log 是端到端的，不需要使用阈值进行单独的判错，并且其结合注意力机制和弱行为轮廓(Week Behavioral Profiles，WBP)，可以对修复结果进行解释。最后，在仿真实验中，本文与该场景下目前最优的研究[4]展开对比分析，在公开数据集上利用常用的机器学习指标对方法修复性能进行评估，还利用不同含噪率下事件日志修复前后的弱行为轮廓量化模型在行为上的修复程度，进而度量无过程模型下基于事件日志的 BERT4Log 对低质量事件日志修复的综合提升程度。

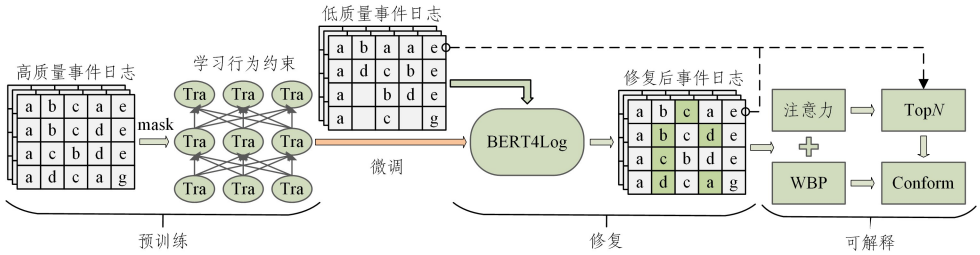

图 1 BERT4Log 的可解释事件日志修复流程

Fig. 1 Interpretable repair process for event log by BERT4Log

本文的主要贡献如下：

(1)基于 BERT 和注意力机制，结合遮掩策略，提出 BERT4Log 的日志修复模型。

(2)展开基于一组公开数据集的仿真实验，与现有研究工作进行对比分析，验证了本文方法在修复两种主要噪音以及行为关系时的有效性和优越性。

(3)利用注意力机制和弱行为轮廓构建解释算法，针对低质量事件日志中两类主要噪音的修复结果进行具体的可解释分析。

本文第 1 节介绍低质量事件日志修复的相关研究；第 2 节介绍本文使用的一些基本概念和公式，并形式化弱行为轮廓的概念；第 3 节详细介绍 BERT4Log 模型的构建方法和网络细节；第 4 节进行模型的仿真实验，对实验结果进行对比分析，以及在行为视角上对具体实例进行修复结果的解释；最后总结全文并展望未来。

1 相关工作

传统的事件日志修复往往需要结合过程模型来找出其中可能存在的异常行为或数据并将它们视为噪音。文献[9]使用 α 算法挖掘工作流网，并利用特定模式来识别可能为异常的流程，从而实现欺诈检测。文献[10]频繁地执行模式发现来对事件日志中的迹进行聚类，该方法不仅可以找出不符合过程模型的迹，也可以找出不符合行为模式的迹。基于特定上下文中活动发生的频率，文献[11]不仅检测给定事件数据中的异常行为，还可以通过允许删除不常见的行为来改进模型发现结果。但上述研究都只在较为单一的视角下进行异常检测或噪音处理，然而在真实 PAIS 的日志记录过程中会产生多方面的问题。为了更加细致地辅助事件日志质量的提升，文献[5]分析了日志中可能存在 4 类流程特征问题的 27 类事件日志质量问题，并将不准确(被记录但是不符合事件

日志行为约束的数据)和缺失(未被记录的相关数据)作为导致事件日志质量下降的主要原因。其中,对于缺失噪音的修复,文献[12]基于随机 Petri 网,使用对齐算法以及贝叶斯网络来修复缺失的活动和时间戳,但由于在计算概率时耦合了活动和时间属性,因此计算复杂性较高。因果网同样可以描述流程发生规则,文献[13]将其与分支框架结合,从而提供具有 top-k 召回率的缺失活动修复推荐,并且提出的索引和修剪技术可以提高修复的时间效率。对于不准确噪音,文献[14]提出了一种基于 Petri 网的图修复方法来修复不一致的事件名称,并且该方法有可能检测到不健全的结构(不同于 Petri 网中描述的行为约束),但不能进行自动修复。上述研究都必须具备过程模型,其可以将事件日志中的行为约束进行显式地表达,并且可以明确地解释日志的异常检测或修复的过程,但由于给定或者挖掘出的过程模型过于简单,这些方法可能并不适用于真实的业务流程。

现实中往往不具备可参考的过程模型,或者由于真实事件日志过于庞大和复杂,使其挖掘出的过程模型的使用效率过低甚至无法使用[15](如"意大利面"类型的过程模型)。为此,出现了许多不依靠过程模型、仅基于事件日志进行日志修复的研究。例如,文献[16]使用后继关系来生成迹的表示向量,对完整事件日志中的迹进行聚类后找出与异常迹向量最相似的簇,通过对比进行缺失活动的修复。虽然该方法在一个真实事件日志上进行验证,但缺失活动占总活动的比例较小,并且没有使用表达行为约束较全面的行为轮廓。文献[17]将事件日志质量问题描述为模式的组件,并从多个事件日志中提炼出 11 种缺陷模式,然后使用存储模式的知识库进行活动的修复。但每个 PAIS 实现的流程可能不同,并不能提取所有流程的模式。对于特定的事件日志,采用通用的神经网络自动学习日志中包含的行为约束是有前景的。例如,文献[18]利用高质量事件日志中的活动和用户资源自监督地训练自编码器,从而进行异常检测,其大致的检测过程是将迹输入到训练好的自编码器中得到该迹在每个位置上活动的预测概率,进而分析低概率位置是否存在异常,但该方法没有进行修复。文献[19]利用自编码器中的高预测概率来重建事件日志中缺失的属性值,但无法处理不正确噪音。进一步地,文献[4]将日志修复分为清理和重构两个阶段。第一阶段类似于文献[18]中的方法,即需要学习并预测不正确噪音的位置;第二阶段则类似于文献[19]中的方法,即删除这些位置上可能为不正确噪音的属性,并选取该位置上预测概率最大的属性作为修复结果。虽然此方法可以同时处理低质量事件日志的两类主要噪音,但需要考虑判定噪音的阈值;并且由于自动编码器是黑盒模型,该研究并没有对修复结果进行单独的可解释分析。

近年来,NLP 中的预训练语言模型得到了快速发展,包括 ELMo 和 GPT,以及备受瞩目的 BERT。其中 ELMo 基于双向的长短期记忆模型(Long Short-Term Memory,LSTM),而 GPT 和 BERT 都基于使用注意力机制的 Transformer[20],但前者是单向模型,后者是双向模型。这些 PLM 的共同点是都需要利用大量的语料库数据去自监督地学习文本中的通用语义,然后再通过基于特征(提取特征向量)或者微调(调整 PLM 架构或参数再进行特定任务的训练)的方式应用到下游任务中,如词性标注和情感分析等,这些都表现出 PLM 强大的特征提取能力以及较高的泛用性。相关的 PLM 技术已经被应用在一些过程挖掘任务中,例如,文献[2]使用 GPT-2 进行下一活动预测并得到了高于基准的性能,但该方法并没有进行预测结果的解释。文献[1]提出了基于 LSTM 的两种注意力机制,可以找出对下一活动预测结果影响较大的事件以及事件属性,并且对两种架构进行了详细的对比,但并没有分析该预测模型是否学习到日志中的某种具体的行为约束。BERT 结合注意力和遮掩策略可以使其同时利用序列中的双向信息预测遮掩候选集中的 token。文献[21]利用 BERT 处理事件日志中的文本数据,从而增强异对常行为的检测能力,但并未通过预训练来学习日志中的行为约束。虽然上述研究在异常检测或 PPM 中使用过相关 PLM,但并没有将具有一定纠错能力的 BERT 或其他 PLM 用在低质量事件日志的修复工作中。

虽然各种深度学习模型在实验中取得了令人满意的效果,但许多实际应用场景需要对模型预测结果进行解释,以此提高用户对该结果的信任程度。在 PPM 中,文献[22]提出了一种基于事后解释算法的可视化方法来分析模型预测出错的原因,文献[23]利用博弈论中的 Shapley 值来获得对预测的稳健解释。两者都进行了基于事后和数据的预测结果解释,但由于行为是过程挖掘领域中一种非常重要的视角,而以往的工作并没有尝试使用行为轮廓,比如没有利用行为轮廓具体地分析预测模型学习了事件日志中的何种行为约束,也没有将基于行为的可解释方法应用在日志修复工作中。因此,在无过程模型的条件下,从日志出发研究具有高性能日志修复能力以及行为相关的可解释修复过程是至关重要的。基于此,本文提出了 BERT4Log 和 WBP,并结合多层多头注意力机制来进行低质量事件日志修复的可解释过程。

2 基础知识

本文在前期过程挖掘的相关研究的基础上[24-27],对事件、日志、活动投影函数和日志行为轮廓等基本定义进行形式化描述,并对本文提出的弱行为轮廓以及衡量在行为上的日志修复程度的行为符合度和行为相似度进行详细定义。

定义 1(事件) 事件日志中任意事件 e 是其对应的活动在某时刻的执行步骤,e 包含多个属性,可以记为多元组 $e=\{caseid, act, timestamp, attr_{1,2,\cdots,n}\}$,其中,$caseid$ 为 e 所属的流程实例,act 为 e 执行的活动,$timestamp$ 为 e 执行的时间戳。以上为一个事件所必须的属性,其余非必须属性由 $attr_{1,2,\cdots,n}$ 表示,如资源等属性。

定义 2(迹和事件日志) 迹 σ 为若干事件组成的有序序列,记为 $\langle e_1, e_2, \cdots, e_l\rangle$,其中,$e_i \in E, i \in [1, l]$,$E$ 为事件的域,l 为 σ 的长度。事件日志 log 是 σ 的集合,记为 $\{\sigma_1, \sigma_2, \cdots, \sigma_L\}$,其中 L 为 log 的大小。

定义 3(投影函数) 给定一条迹 σ,设 π_A 为 σ 到对应活动的集合 $\{act_1, act_2, \cdots, act_m\}$ 的映射,记为 $\pi_A(\sigma) \subseteq A$。其中,$act_i \in A, i \in [1, m]$,$A$ 为活动的域,m 为 σ 中不同活动的个数。设 σ 中活动 act 到其所属事件的集合 $\{e_1, e_2, \cdots, e_n\}$ 的

映射为 π_E，记为$\pi_E(act)\subseteq E$。其中，$act\in\pi_A(\sigma)$，$e_j\in E$，$j\in[1,n]$，E 为事件的域，n 为 act 对应 σ 中的 e 个数。

定义 4(弱序关系[24]) 给定一条迹 σ，其对应的活动集合为$\pi_A(\sigma)$，设 $x,y\in\pi_A(\sigma)\subseteq A$，$x\neq y$。$x$ 在 σ 中对应的事件集合为$\pi_E(x)$，y 在 σ 中对应的事件集合为$\pi_E(y)$。设$e_x\in\pi_E(x)$，$e_y\in\pi_E(y)$。在 σ 中，若 $\exists e_x$，$\exists e_y$，且e_x在e_y之前发生，则记 $x\succ y$，即 σ 中活动 x 与 y 满足弱序关系。反之，若 $\forall e_x$，$\forall e_y$，且e_x不在e_y之前发生，则记 $x\not\succ y$，即 σ 中活动 x 与 y 不满足弱序关系。

定义 5(迹行为轮廓[25]) 给定迹 σ，$(x,y)\in(\pi_A(\sigma)\times\pi_A(\sigma))\subseteq(A\times A)$，$x\neq y$。将迹级别上的 x 与 y 行为轮廓记为 $BP^{\sigma}(x,y)$，其中 x 与 y 至多存在下面 3 种行为关系的 1 种。

(1)严格序：若 $\forall x,y\Rightarrow x\succ y\wedge y\not\succ x$，则 $x\rightarrow y$，对应 $BP^{\sigma}(x,y)=\rightarrow$；

(2)反严格序：若 $\forall x,y\Rightarrow y\succ x\wedge x\not\succ y$，则 $x\leftarrow y$，对应 $BP^{\sigma}(x,y)=\leftarrow$；

(3)交叉序：若 $\forall x,y\Rightarrow y\succ x\wedge x\succ y$，则 $x\parallel y$，对应 $BP^{\sigma}(x,y)=\parallel$。

定义 6(活动共现[26]) 给定一个包含 L 条迹的事件日志 log，对于 $x,y\in A$，若两者同时出现在一条迹中，则称 x 与 y 共现。将满足 x 与 y 共现的迹的集合记为 $C(x,y)$，其中，$|C(x,y)|\leqslant L$。

得到迹级别上的行为轮廓$BP^{\sigma}(x,y)$后，可以验证活动对(x,y)在 $C(x,y)$ 中的关系，进而得到日志级别上的行为轮廓。

定义 7(日志行为轮廓[27]) 给定一个事件日志 log，对于$(x,y)\in(A\times A)$，$x\neq y$，$\sigma\in C(x,y)$，将日志级别上的 x 与 y 的行为轮廓记为$BP^{L}(x,y)$，其中 x 与 y 至多存在下面 3 种行为关系的 1 种。

(1)若 $\forall\sigma\in C(x,y)\Rightarrow BP^{\sigma}(x,y)=\rightarrow$，则$BP^{L}(x,y)=\rightarrow$；

(2)若 $\forall\sigma\in C(x,y)\Rightarrow BP^{\sigma}(x,y)=\leftarrow$，则$BP^{L}(x,y)=\leftarrow$；

(3)若 $\exists\sigma\in C(x,y)\Rightarrow BP^{\sigma}(x,y)=\parallel$，则$BP^{L}(x,y)=\parallel$。

所有的$BP^{L}(x,y)$构成日志行为轮廓，记为BP^{L}。

由于定义 5 对活动对(x,y)行为轮廓判定严格，且判定公式并不能反映出日志中两个活动之间的主要关系和次要关系的比例。例如，对于两个不同活动 $x,y\in A$，log 中满足 $x\rightarrow y$ 的迹的条数占 $C(x,y)$ 的 99%，然而，若在剩下的迹中出现 $x\leftarrow y$ 或者 $x\parallel y$，定义 7 就会将 x 与 y 的日志行为轮廓判定为 $x\|y$。因此，为了在细粒度上体现日志的行为轮廓，本文提出了弱行为轮廓。它将活动关系的判定范围缩小到迹的级别上，而整个日志的弱行为轮廓则由每个迹中的活动关系统计构成。

定义 8(日志弱行为轮廓) 给定一个事件日志 log，设$(x,y)\in(A\times A)$，$x\neq y$，$C(x,y)\neq\varnothing$，$\sigma_i\in C(x,y)$，$i\in[1,|C(x,y)|]$。将 x 与 y 在迹级别上的弱行为轮廓记为$\boldsymbol{WBP}^{\sigma}(x,y)$，将日志级别上的弱行为轮廓记为$\boldsymbol{WBP}^{L}(x,y)$，它们都为 3 维向量，具体计算公式如下：

$$\boldsymbol{WBP}^{\sigma}(x,y)=\begin{cases}(1,0,0), & \text{if } BP^{\sigma}(x,y)=\rightarrow\\ (0,1,0), & \text{if } BP^{\sigma}(x,y)=\leftarrow\\ (0,0,1), & \text{if } BP^{\sigma}(x,y)=\parallel\end{cases}$$

$$\boldsymbol{WBP}^{L}(x,y)=(p_{\rightarrow},p_{\leftarrow},p_{\parallel})=\frac{\sum_{i}^{|C(x,y)|}\boldsymbol{WBP}_i^{\sigma}(x,y)}{|C(x,y)|}$$

其中，$p_{\rightarrow},p_{\leftarrow},p_{\parallel}\in[0,1]$，分别表示在 log 中出现 $x\rightarrow y$，$x\leftarrow y$，$x\parallel y$ 的比率，并且满足三者之和为 1。所有活动对的$\boldsymbol{WBP}^{L}(x,y)$构成日志弱行为轮廓，记为$\boldsymbol{WBP}^{L}$。

例如，给定事件日志 $log=\{\langle a,b,c,a,d\rangle,\langle a,c,b,d\rangle,\langle a,c,d,b,d\rangle,\langle c,d,b\rangle\}$，其中的 4 条迹分别记为$\sigma_1$，$\sigma_2$，$\sigma_3$和$\sigma_4$，活动域 $A=\{a,b,c,d\}$。对于活动 a 和 b，$C(a,b)=\{\sigma_1,\sigma_2,\sigma_3\}$。由定义 5 可以得到$BP_1^{\sigma}(a,b)=\parallel$，$BP_2^{\sigma}(a,b)=\rightarrow$，$BP_3^{\sigma}(a,b)=\rightarrow$。依定义 7 则得到的日志行为轮廓中$BP^{L}(x,y)=\parallel$。若按照定义 8，先计算$\boldsymbol{WBP}_1^{\sigma}(a,b)=(0,1,1)$，$\boldsymbol{WBP}_2^{\sigma}(a,b)=(1,0,0)$，$\boldsymbol{WBP}_3^{\sigma}(a,b)=(1,0,0)$，$|C(a,b)|=3$，因此可得日志弱行为轮廓中$\boldsymbol{WBP}^{L}(a,b)=(2/3,0,1/3)$。显而易见，通过日志弱行为轮廓可以看出日志 log 中活动 a 和 b 的关系主要为严格序，而不是根据定义 5 将其判定为出现较少的交叉序。同理，还可以计算 A 中其余的活动之间的$\boldsymbol{WBP}^{L}$，结果如表 1 所列。

表 1 一个关于$\boldsymbol{WBP}^{L}$的简单例子

Table 1 A Simple Example about $\boldsymbol{WBP}^{L}$

	a	b	c	d
a		$\left(\frac{2}{3},0,\frac{1}{3}\right)$	$\left(\frac{2}{3},0,\frac{1}{3}\right)$	$(1,0,0)$
b	$\left(0,\frac{2}{3},\frac{1}{3}\right)$		$\left(\frac{1}{4},\frac{3}{4},0\right)$	$\left(\frac{1}{2},\frac{1}{4},\frac{1}{4}\right)$
c	$\left(0,\frac{2}{3},\frac{1}{3}\right)$	$\left(\frac{1}{4},\frac{3}{4},0\right)$		$(1,0,0)$
d	$(0,1,0)$	$\left(\frac{1}{4},\frac{1}{2},\frac{1}{4}\right)$	$(0,1,0)$	

为了在行为上量化一条迹中某位置上活动的修复程度以及对于整体日志的修复程度，本文分别提出了行为符合度和行为相似度。

定义 9(行为符合度) 给定迹 σ 中 $x,y\in A$，x 与 y 的行为符合 log 中对应行为的程度为：

$$conform=\begin{cases}1-\frac{1}{2}\left\|\boldsymbol{WBP}^{\sigma}(x,y)-\boldsymbol{WBP}^{L}(x,y)\right\|_1, & \\ \qquad\text{if } (x,y)\in\boldsymbol{WBP} & \\ 0, \qquad\text{otherwise} & \end{cases}$$

其中，$conform\in[0,1]$。若$(x,y)\in\boldsymbol{WBP}$，由于$\boldsymbol{WBP}^{\sigma}(x,y)$与$\boldsymbol{WBP}^{L}(x,y)$都是 3 维向量，分别记为$(p_{\rightarrow}^{\sigma},p_{\leftarrow}^{\sigma},p_{\parallel}^{\sigma})$和$(p_{\rightarrow}^{L},p_{\leftarrow}^{L},p_{\parallel}^{L})$，则 $\left\|\boldsymbol{WBP}^{\sigma}(x,y)-\boldsymbol{WBP}^{L}(x,y)\right\|_1=|p_{\rightarrow}^{\sigma}-p_{\rightarrow}^{L}|+|p_{\leftarrow}^{\sigma}-p_{\leftarrow}^{L}|+|p_{\parallel}^{\sigma}-p_{\parallel}^{L}|\geqslant 0$。由于$|p_{\rightarrow}^{\sigma}-p_{\rightarrow}^{L}|\leqslant|p_{\rightarrow}^{\sigma}|+|p_{\rightarrow}^{L}|$，根据定义 8 知向量内所有元素的值在 0 到 1 之间，并且$p_{\rightarrow}^{\sigma}+p_{\leftarrow}^{\sigma}+p_{\parallel}^{\sigma}=1$，$p_{\rightarrow}^{L}+p_{\leftarrow}^{L}+p_{\parallel}^{L}=1$，由此可以得到 $\left\|\boldsymbol{WBP}^{\sigma}(x,y)-\boldsymbol{WBP}^{L}(x,y)\right\|\leqslant|p_{\rightarrow}^{\sigma}|+|p_{\leftarrow}^{\sigma}|+|p_{\parallel}^{\sigma}|+|p_{\rightarrow}^{L}|+|p_{\leftarrow}^{L}|+|p_{\parallel}^{L}|=2$，从而推出 $0\leqslant conform\leqslant 1$。例如，可以接着使用表 1 中的例子，即$\boldsymbol{WBP}_1^{\sigma}(a,b)$与$\boldsymbol{WBP}^{L}$的行为符合度为：

$$conform=1-\frac{1}{2}\left\|(0,0,1)-\left(\frac{2}{3},0,\frac{1}{3}\right)\right\|_1=\frac{1}{3}$$

定义 10(行为相似度) 给定两个具有相同活动域的日志，记为 log_1 和 log_2，对应的弱行为轮廓分别为$\boldsymbol{WBP}_1^{L}$和$\boldsymbol{WBP}_2^{L}$，若 $x,y\in A$，$x\neq y$，将 log_1 与 log_2 的行为相似度记为

BS，则其计算式为：

$$getSingleBS(x,y)=\begin{cases}1-\frac{1}{2}\parallel \boldsymbol{WBP}_1^L(x,y)-\boldsymbol{WBP}_2^L(x,y)\parallel_1, \\ \quad \text{if } (x,y)\in \boldsymbol{WBP}_1^L \wedge (x,y)\in \boldsymbol{WBP}_2^L \\ 0, \quad \text{otherwise}\end{cases}$$

$$BS=\frac{1}{(|A|^2-|A|)}\sum_{x}^{A}\sum_{y}^{A-x} getSingleBS(x,y)$$

其中，$|A|^2-|A|$ 为弱行为轮廓矩阵中满足 $x\neq y$ 的元素个数。同理，由 $0\leqslant \parallel \boldsymbol{WBP}_1^L(x,y)-\boldsymbol{WBP}_2^L(x,y)\parallel_1\leqslant 2$，可推出 $BS\in[0,1]$。

3 BERT4Log 模型

本节将从 3 个方面展开描述。其中 3.1 节介绍本文所使用的 Mask 语言模型，并与传统的语言模型作对比；3.2 节分别对 BERT4Log 的基本网络架构、激活函数，以及其他网络细节进行阐述；3.3 节描述 BERT4Log 的微调、针对修复结果解释的两个主要问题以及结合注意力分数和弱行为轮廓的修复结果解释算法。

3.1 Mask 语言模型

NLP 中的语言模型可以判断一个序列是一个句子的概率，其中该句子符合相关自然语言的规则。给定一条具有 N 个 token 的序列，记为 $seq=\langle t_1,t_2,\cdots,t_N\rangle$。标准条件语言模型(Standard Conditional Language Models，SCLM)在计算第 i 个位置上 token 的概率时，依靠的是该位置之前的信息，即 $(t_1,t_2,\cdots,t_{i-1})$。因此，SCLM 预测该序列为一个句子的概率为：

$$p(seq)=\prod_{i=1}^{N}p(t_i\mid t_1,t_2,\cdots,t_{i-1})$$

其模型训练的目标是最大化每个序列的预测概率之和。

ELMo 使用 LSTM 进行多层双向的自监督训练，其在预测第 i 个位置上的 token 时用上了该位置后面的信息[6]，即 $\langle t_{i+1},t_{i+2},\cdots,t_N\rangle$，因此该模型预测一个句子的概率为：

$$p(seq)=\prod_{i=1}^{N}p(t_i\mid t_1,t_2,\cdots,t_{i-1})+\prod_{i=1}^{N}p(t_i\mid t_{i+1},t_{i+2},\cdots,t_N)$$

但由于 LSTM 具有顺序性，ELMo 在预测目标 token 时只能考虑一个方向的信息，即向前预测时只能使用从前向后的信息，反向预测时只能使用从后向前的信息，并不能同时看到双向的信息[8]。

GPT 使用自注意力机制来代替传统的循环神经网络(Recurrent Neural Network，RNN)，即它预测目标 token 时，会同时计算序列中所有 token 对该位置的注意力分数，但不能直接使用所有的注意力分数。因为该模型同 SCLM 一样，依然遵守从左到右(或右到左)的方式进行训练；并且在利用当前信息预测下一个 token 时，并不能看到该 token 之后的信息。例如，在预测 t_i 时，GPT 只能利用 t_i 之前的信息，即 $(t_1,t_2,\cdots,t_{i-1})$，若使用了 t_{i+1} 的信息去预测 t_i，在预测 t_{i+1} 时就会使模型提前知道预测目标的信息。由于自注意力可以利用整个句子的信息，因此在预测 t_i，需要将后面的 token 遮盖掉，即将后面的 token 换成[MASK]标志。

为了能够双向地利用自注意力机制，BERT 使用了 MLM 的方式。不同于利用上述标准条件语言模型去预测下一个 token，该模型是利用序列中不属于遮掩候选集中的 token 来预测属于的 token。给定一个序列 seq，按概率 P_c 选出需要遮掩的 token 并构成遮掩候选集，记为 C_{mask}，其包含的 token 有概率被替换成[MASK]标志。然后让模型利用上下文信息去预测 seq 中每个[MASK]对应的原始 token，这样在使用注意力进行预测时，就能同时利用 seq 未被遮掩的 token，实现双向的学习。C_{mask} 中的 token 会按照概率 P_{mask} 被替换为[MASK]，按照概率 P_{replace} 被替换为随机的一个 token，按照概率 P_{constant} 保持不变，三者之和为 1。使用这种掩盖策略的原因是，在预训练 MLM 时使用的数据中包含[MASK]，但在后续使用该模型解决特定任务时，所使用的数据是不包含此标志的，这样就会在预训练和微调之间产生不匹配的问题[8]。然而，按照上述遮掩策略，MLM 并不知道[MASK]替换的是哪一个 token，因为任何一个 token 都可以被视为被替换的。这会迫使模型在预测当前时刻的 token 时不能过于依赖其本身，而是需要考虑它的上下文，甚至利用上下文来进行纠错。在本文中，BERT4Log 使用的 3 种概率与原 BERT 相同，即 P_{mask}，P_{replace} 和 P_{constant} 的值分别为 0.8，0.1 和 0.1。例如，给定一条迹 σ 对应的活动序列为 $\langle a,b,d,e,g,i\rangle$，若其遮掩候选集为 $C_{\text{mask}}=\{d\}$，则会出现下面 3 种情况：

(1)d 有 80% 的概率被遮掩，即替换成[MASK]，如 $\sigma\rightarrow\langle a,b,[MASK],e,g,i\rangle$；

(2)d 有 10% 的概率被替换成随机的一个活动，如 $\sigma\rightarrow\langle a,b,a,e,g,i\rangle$；

(3)d 有 10% 的概率保持不变，即保持原活动，如 $\sigma\rightarrow\langle a,b,d,e,g,i\rangle$。

综上所述，给定迹 $\sigma=\langle e_1,e_2,\cdots,e_l\rangle$，BERT4Log 预测一个 C_{mask} 中的事件的概率为：

$$p(e_{\text{mask}})=p(e_{\text{mask}}\mid e_1,e_2,\cdots,e_m)$$

其中，$e_{\text{mask}}\in C_{\text{mask}}$，$e_1,e_2,\cdots,e_m\notin C_{mask}$，$m<l$。

目前低质量事件日志修复的场景中还没有使用相关 PLM，但本文不选取 ELMo 或 GPT 而是选择类 BERT 架构的原因是：ELMo 使用双向 LSTM 来捕获语句中的语义，但当语句过长时，LSTM 无法捕获语句中的长依赖关系。例如，位于一段话开始的主语与结尾的代词之间的语义关系。并且对比自然语言文本与事件日志可以发现，后者中案例的长度往往大于前者中的一段文本的长度，因此本文不使用传统的 RNN，而是使用注意力机制来代替它。GPT 使用了基于注意力机制的 Transformer 来提取语义关系，但由于它是一种单向的语言模型，在预测目标 token 时没有同时考虑两个方向上的信息。虽然 ELMo 是采用双向 LSTM，但由于 SCLM 的约束，它在单方向上预测某个位置的 token 时并不能同时利用另一个方向上的信息，否则将泄露信息给后续时间步的预测。由定义 5 知，迹中活动与其前后活动都具有行为关系。作为一种 MLM，BERT 结合多层多头注意力机制实现的双向 Transformer 架构在预测 token 时可以同时考虑语句中所有位置上的信息，直观上，这种架构有利于学习被遮掩活动与迹中所有未遮掩活动之间的行为关系。综上所述，本文使用类似

BERT 的架构来修复低质量事件日志，并且注意力机制有助于增强预测结果的可解释性。

3.2　BERT4Log 的模型构建

给定一条长度为 l 的迹 $\sigma=\langle e_1,e_2,\cdots,e_l\rangle$，其中，$e\in E$。首先，BERT4Log 需要将事件编码成数值形式的向量，这样才能输入到语言模型中进行训练。对于事件的编码，可以考虑分别编码事件中的各个属性，然后采取串联或者相加的方式得到事件的特征表示。为了尽可能地学习到事件日志中的行为关系，本文仅选用事件中的活动属性。活动是每个事件日志中最基本的属性，这也符合 BERT4Log 的泛用性。对于事件中的其他属性，可以将其看作 BERT4Log 的下游任务来处理（如从 BERT4Log 中提取包含日志行为约束的活动嵌入或迹嵌入，串联其他属性编码用于该属性修复或 PPM 的预测任务）。因此，本文将迹 σ 直接由其对应的活动序列 $\langle act_1,act_2,\cdots,act_l\rangle$ 表示，其中 $act\in\pi_A(\sigma)$。

迹中每个活动的编码由图 2 中的嵌入层实现。本文将事件属性看成最小单元数据，因此 BERT4Log 的嵌入层直接使用可学习的编码方式，而不是 BERT 中使用的字符嵌入。嵌入层是一个线性层，其参数矩阵可以被当作嵌入矩阵 $\boldsymbol{M}$。对于活动，嵌入层输入向量的维度为 $|A|$，而输出向量维度由人为设置。活动嵌入矩阵 $\boldsymbol{M}_a$ 可以使不同活动映射为不同的活动嵌入，活动序列经过 $\boldsymbol{M}_a$ 可以得到每个活动对应的嵌入向量，即 $\langle \boldsymbol{x}_1^a,\boldsymbol{x}_2^a,\cdots,\boldsymbol{x}_l^a\rangle$，其中 $\boldsymbol{x}_i^a=\boldsymbol{M}_a(act_i)$，$i\in[1,l]$。由于 BERT4Log 使用自注意力机制来代替传统的 RNN，这会导致序列中每个活动对应位置的信息缺失，因此需要对迹中活动的位置进行单独编码，以使活动的嵌入向量包含位置信息。BERT 使用的是可以包含相对位置的编码方式，为了简单起起见，本文使用与活动属性相同的可学习的编码方式。位置嵌入同样可以得到位置嵌入矩阵 $\boldsymbol{M}_p$，迹中每个位置对应的嵌入向量为 $\langle \boldsymbol{x}_1^p,\boldsymbol{x}_2^p,\cdots,\boldsymbol{x}_l^p\rangle$，其中 $\boldsymbol{x}_i^p=\boldsymbol{M}_p(i)$，$i\in[1,l]$。最后，将活动与对应位置的嵌入向量逐元素相加后，即可得到每个活动最终输入到模型的嵌入向量，记为 $\boldsymbol{X}=\langle \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_l\rangle$，其中 $\boldsymbol{x}_i=\boldsymbol{x}_i^a+\boldsymbol{x}_i^p$，$i\in[1,l]$。由于活动嵌入向量和位置嵌入向量需要相加，因此要求两者的维度相同，即 $\boldsymbol{x}_i^a,\boldsymbol{x}_i^p,\boldsymbol{x}_i\in\mathcal{R}^{1\times d_{\text{model}}}$，$\boldsymbol{X}\in R^{l\times d_{\text{model}}}$。其中，$d_{\text{model}}$ 为嵌入层的输出维度。

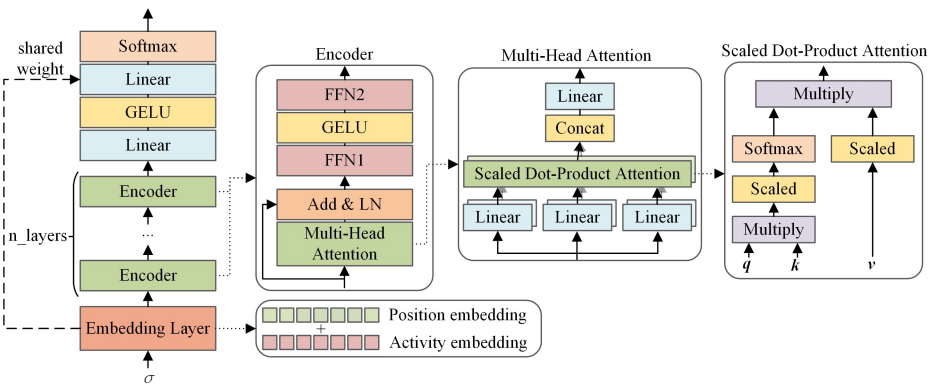

图 2　BERT4Log 的网络架构

Fig. 2　Network architecture of BERT4Log

得到迹 σ 对应的嵌入向量 $\boldsymbol{X}$ 后，将 $\boldsymbol{X}$ 中每个活动嵌入向量 $\boldsymbol{x}_i$ 分别经过 3 个不同的线性层，分别记为 L_Q，L_K 和 L_V。然后输出 3 个不同的向量，分别记为活动查询 $\boldsymbol{q}_i$、活动键 $\boldsymbol{k}_i$、活动值 $\boldsymbol{v}_i$，维度分别记为 d_q，d_k 和 d_v，并且 $d_q=d_k$。由于使用自注意力机制，三者都由 x_i 得到。具体地：

$$\boldsymbol{q}_i=L_Q(\boldsymbol{x}_i)=\boldsymbol{x}_i\cdot\boldsymbol{W}_{\boldsymbol{Q}}+\boldsymbol{b}_{\boldsymbol{Q}}$$

$$\boldsymbol{k}_i=L_K(\boldsymbol{x}_i)=\boldsymbol{x}_i\cdot\boldsymbol{W}_{\boldsymbol{K}}+\boldsymbol{b}_{\boldsymbol{K}}$$

$$\boldsymbol{v}_i=L_V(\boldsymbol{x}_i)=\boldsymbol{x}_i\cdot\boldsymbol{W}_{\boldsymbol{V}}+\boldsymbol{b}_{\boldsymbol{V}}$$

其中，$i\in[1,l]$，l 是迹的长度，$\boldsymbol{W}_{\boldsymbol{Q}}\in\mathcal{R}^{d_{\text{model}}\times d_q}$，$\boldsymbol{W}_{\boldsymbol{K}}\in\mathcal{R}^{d_{\text{model}}\times d_k}$ 和 $\boldsymbol{W}_{\boldsymbol{V}}\in\mathcal{R}^{d_{\text{model}}\times d_v}$ 分别为 3 个线性层的权重矩阵，$\boldsymbol{b}_{\boldsymbol{Q}}$，$\boldsymbol{b}_{\boldsymbol{K}}$ 和 $\boldsymbol{b}_{\boldsymbol{V}}$ 则为偏差项。

接下来介绍图 2 中的缩放点积注意力块（Scaled Dot-Product Attention），该部分利用 $\boldsymbol{q}$ 和 $\boldsymbol{k}$ 来计算活动之间的注意力分数 $\boldsymbol{s}$。具体表示为：

$$s_{ij}=\begin{cases}\text{softmax}\left(\dfrac{\boldsymbol{q}_i\cdot\boldsymbol{k}_j}{\sqrt{d_k}}\right), & \text{if } a_j\notin C_{\text{mask}}\\ \varepsilon, & \text{if } a_j\in C_{\text{mask}}\end{cases}$$

其中，s_{ij} 代表活动 a_i 关于活动 a_j 的注意力分数，$i\in[1,l]$，$j\in[1,l]$。若 $a_j\in C_{\text{mask}}$，则将 s_{ij} 置为 ε，这样做的原因是，训练目标是预测 C_{mask} 中的活动，因此模型不应该知道 C_{mask} 中的信息。但是由于训练时需要反向求导，因此不能将注意力分数置为零，而是设为一个非常小的值。除以 $\sqrt{d_k}$ 可以使求导更稳定[20]。紧接着计算活动 a_i 与迹中所有不属于 C_{mask} 的活动的隐藏表示 $\boldsymbol{z}_i$，即迹中所有活动键 $\boldsymbol{v}_i$ 与注意力分数 s_{ij} 的乘积之和。具体表示为：

$$\boldsymbol{z}_i=\sum_j^l s_{ij}\cdot\boldsymbol{v}_j$$

上述过程描述的是对单个活动的处理，接下来需要将整条迹进行处理，表示为：

$$\boldsymbol{Z}=Attention(\boldsymbol{q},\boldsymbol{k},\boldsymbol{v})=\sum_i^l\sum_j^l s_{ij}\cdot\boldsymbol{v}_j$$

其中，$\boldsymbol{Z}\in\mathcal{R}^{l\times d_v}$。这个过程体现了注意力与循环神经网络的区别，即后者是顺序地进行每个活动的预测，很难并行处理；而前者可以同时处理整条迹中的所有活动。

文献[20]发现将 $\boldsymbol{q}$，$\boldsymbol{k}$ 和 $\boldsymbol{v}$ 分别多次进行不同的线性投影是有益的，因此其使用多头注意力块，它允许模型关注来自不同位置上不同表示子空间的信息。对于 PM 中的事件日志，则有可能学习到不同的行为约束。多头注意力块（Multi-head Attention）中包含多个并行的缩放点积注意力块，然后将它们

的输出串联起来。因为之后需要进行残差连接，所以要求此块的输出和输入向量的维度相同。为此，可以将串联的向量经过一个线性层来使输入输出维度相同。具体地：

$$\boldsymbol{H}=MultiHead(\boldsymbol{Z})=Concat(\boldsymbol{Z}_1,\boldsymbol{Z}_2,\cdots,\boldsymbol{Z}_{n\_heads})\boldsymbol{W}^C+\boldsymbol{b}_C$$

其中，n_heads 为注意力的头数；$\boldsymbol{W}^C\in\mathcal{R}^{hd_v\times d_{\text{model}}}$ 为线性层的参数矩阵；$\boldsymbol{b}_C$ 为偏差项；$\boldsymbol{H}\in\mathcal{R}^{l\times d_{\text{model}}}$ 为多层注意力的输出，并与此块的输入 X 的维度相同。

至此得到了多头注意力块的输出 $\boldsymbol{H}$，本文接着将其进行残差连接，然后进行层标准化(Layer Normalization，LN)。不同于标准的 BERT，本文依次经过全连接层 FFN1，GELU 和 FFN2。具体表示为：

$$\boldsymbol{H}_1=LN(\boldsymbol{H}+\boldsymbol{X})$$

$$\boldsymbol{H}_2=FFN2(\text{GELU}(FFN1(\boldsymbol{H}_1)))$$

其中，$\boldsymbol{H}_1,\boldsymbol{H}_2\in\mathcal{R}^{l\times d_{\text{model}}}$。至此，即构成了一个基本的 Encoder 块。为了堆叠 n_layers 个 Encoder 块，此块输入和输出的维度需相同。例如，第一个 Encoder 块的输入 $\boldsymbol{X}$ 与输出 $\boldsymbol{H}_2$ 的维度都为 $l\times d_{\text{model}}$。

最后，将经历多层 Encoder 块的输出输入到一个使用 GELU 激活函数的线性层。紧接着输入到最后一个线性层，并且这个线性层的权重共享嵌入层的权重。最终经过 Softmax 层后得到迹中每个位置上的预测所有活动的概率。

不同于 BERT，本文去除了下一关联句的预测任务，由此可以得到 BERT4Log 的目标函数为：

$$\max\sum_{\sigma}^{Log}\sum_{act_{\text{mask}}}^{C^{\sigma}_{\text{mask}}}\log p(act_{\text{mask}}\mid act_1,act_2,\cdots,act_m;\boldsymbol{\theta})$$

其中，$act_{\text{mask}}\in C^{\sigma}_{\text{mask}}$，$act_1,act_2,\cdots,act_m\notin C^{\sigma}_{\text{mask}}$，$C^{\sigma}_{\text{mask}}$ 为 σ 中的遮掩候选集，$\sigma\in Log$，$\boldsymbol{\theta}$ 为 BERT4Log 中的可学习参数。

3.3 微调与可解释修复

预训练期间，语言模型主要进行自监督训练。训练完成后，一般有两种主要的方式将 PLM 应用到实际任务中[8]。一种是基于特征的方式，即从 PLM 中取出包含 token 依赖关系的表示向量，然后将其输入到针对特定任务的预测模型中。另一种是基于微调的方式，即对特定任务在 PLM 的基础上再进行一次有监督的训练，期间可能会在 PLM 中加入额外的网络层、冻结部分参数或者其他微调技巧。本文将采取基于微调的方法，并冻结训练好的 BERT4Log 的全部参数。由于真实事件日志中没有[MASK]标志，因此对于微调阶段的预测，在计算注意力分数时改为考虑迹中所有的活动。并且，本文直接输入包含噪音的事件日志，并得到修复后的日志，整个过程只有预训练时的一次自监督训练，修复阶段并不需要额外地进行有监督的训练。

对于 BERT4Log 修复结果的可解释性，本文尝试解释下面两个主要问题：

Q1：为什么 BERT4Log 将迹中某个位置上的活动判错？

Q2：为什么 BERT4Log 将它认为错误的活动预测为其他活动？

本文提出结合多头注意力和弱行为轮廓的算法来解决上述问题。在详细地阐述解释算法之前，需要对算法中使用的相关函数进行定义和说明(相关函数和算法以类 Python 的伪代码形式展示)。

首先，需要描述 BERT4Log 的输入和输出。模型的输入是包含噪音的事件日志中的迹，记为 σ_{in}。输出则包含两个部分：一个是修复后的迹，记为 σ_{out}，并且 σ_{in} 与 σ_{out} 的长度都为 l；另一个是 σ_{in} 中活动之间的注意力分数，记为 ***score***。由于 BERT4Log 使用多层多头的自注意力机制，因此在某一层某一头上的每条迹中，任意两个位置上的活动之间都具有一个注意力分数，即 s。将 σ_{in} 在每一层、每一头上的注意力分数矩阵记为 $\boldsymbol{score}_{mn}$，其每个元素记为 s^{mn}_{ij}。其中，$i,j\in[1,l]$，$n\in[1,n\_layers]$，$m\in[1,n\_heads]$，n_layers 为 3.2 节介绍的 Encoder 堆叠的层数，n_heads 为每个 Encoder 块中注意力的头数。最后，可以得到关于 σ_{in} 在所有层中的注意力分数，即 $\boldsymbol{score}\in\mathcal{R}^{n\_layers\times n\_heads\times l\times l}$。上述过程可表示为：

$$BERT4\text{Log}(\sigma_{\text{in}})=\sigma_{\text{out}},\boldsymbol{score}$$

其次，通过对比 σ_{in} 和 σ_{out} 来得到 BERT4Log 将 σ_{in} 中的哪些活动判定为错，并且纠错为哪些活动。为此，本文定义函数 $getTraceDiff$。它的输入是 σ_{in} 和 σ_{out}，输出为 $diff$，它的每一个元素都为一个三元组 $(err\_pos,err\_act,repair\_act)$。其中，$err\_pos$ 为 BERT4Log 判断 σ_{in} 中活动出错的位置，err_act 为出错的活动，$repair\_act$ 为纠错后的活动，具体过程如算法 1 所示。

算法 1 function $getTraceDiff$

```
Input: σ_in, σ_out
Output: diff
Def getTraceDiff(σ_in, σ_out):
diff=[]/* 初始化为一个列表 */
for i in[1,l]:
    act_in=σ_in[i]
    act_out=σ_out[i]
    if act_in≠act_out:/* 如果σ_in与σ_out相同位置上活动不同，则存在纠错的情况 */
        diff←(i,act_in,act_out)/* 记录判错的位置以及活动的转变 */
    else:
        continue
    end
return diff
```

以一个简单的例子来说明函数 $getTraceDiff$ 的作用。给定一个包含噪音的迹 $\sigma_{\text{in}}=\langle a,c,c,a,b\rangle$，$A=\{a,b,c,d\}$，将其输入到 BERT4Log 中得到 $\sigma_{\text{out}}=\langle a,b,c,a,d\rangle$ 以及注意力分数 ***score***。通过 $getTraceDiff(\sigma_{\text{in}},\sigma_{\text{out}})$ 可以得到 $diff=[(2,c,b),(5,b,d)]$，其元素分别表示模型将 σ_{in} 中第 2 个位置上和第 5 个位置上的活动 a 和 b 判错，并纠错为 σ_{out} 中相同位置的 b 和 d。

由于 BERT4Log 使用多层多头的注意力，因此迹中每个位置上的活动都具有 $n\_layers\times n\_heads\times l$ 个个注意力分数。但是，对于 σ_{in} 中某个目标活动的预测而言，所有活动并不都是重要的。为了找出对模型产生判错和纠错结果有重要影响的活动，本文定义函数 $getMaxScorePos$。它的输入是 ***score*** 和 pos，其中后者为目标活动在 σ_{in} 中的位置。输出是每一层的每一头上具有最大注意力分数的活动对应的位置，共有 $n\_layers\times n\_heads$ 个。具体过程如算法 2 所示。

算法 2 function *getMaxScorePos*(***score***, *pos*)

Input: **score**, pos
Output: MaxScorePos
1. Def getMaxScorePos(**score**, pos):
2. max_score_pos=[]
3. for n in[1, n_layers]: /* 循环每一层 */
4. for m in[1, n_heads]: /* 循环每一头 */
5. max_score_pos←getMax($score_{nm}$[pos]) /* 记录位置 i 的最大 s_{ij}^{nm} 对应的位置 j */
6. end
7. end
8. return MaxScorePos

例如，输入上述得到的 ***score*** 以及 *pos*，记 ***score*** 的第一层第一头的注意力矩阵为 $\boldsymbol{score}_{11}$。由于 $l=5$，则 $\boldsymbol{score}_{11}$ 是一个 5×5 的矩阵。若 *pos*=2，并假设 $\boldsymbol{score}_{11}$ 的第 2 行的注意力分数为(0.1, 0.3, 0.1, 0.1, 0.9)，则通过 *getMax*($\boldsymbol{score}_{11}$[2]) 得到的分数最高为 0.9，对应位置 5，代表 σ_{in} 中在第一层第一头得到与位置 2 最相关的是位置 5 上的活动，最后将位置 5 记录到 *MaxScorePos* 列表中。同理，循环得到不同层上所有头的最大注意力分数对应的位置。

基于上述 3 个函数，可以构造 BERT4Log 修复的解释算法 Explan。该算法的输入为 σ_{in} 和 $\boldsymbol{WBP}^{L}$，前者为含有噪音的迹，后者为训练 BERT4Log 所使用的事件日志对应的弱行为轮廓。输出为 *topN_act* 和 *con*，分别记录对整条迹修复影响较大的活动以及修复前后活动对行为符合度的变化。具体过程如算法 3 所示。

算法 3 algorithm *Explan*(σ_{in}, $\boldsymbol{WBP}^{L}$)

input: σ_{in}, $\mathbf{WBP}^{L}$
output: top N_act, con
1. Def Explan(σ_{in}, $\mathbf{WBP}^{L}$):
2. σ_{out}, score=BERT4Log(σ_{in})
3. diff=getTraceDiff(σ_{in}, σ_{out})
4. top N_act=[]
5. con=[]
6. for d in diff:
7. ep=d[0] /* err_pos */
8. ea=d[1] /* err_act */
9. ra=d[2] /* repair_act */
10. con_{in}=[] /* 用于记录修复前活动对与 $\mathbf{WBP}^{L}$ 的行为符合度 */
11. con_{out}=[] /* 用于记录修复后活动对与 $\mathbf{WBP}^{L}$ 的行为符合度 */
12. msa=[] /* 用于记录每一层每一头关于 ep 的最大注意力位置 */
13. msp=getMaxScorePos(score, ep)
14. for pos in msp:
15. if σ_{in}[pos]==σ_{out}[pos]:
16. key_act=σ_{in}[pos]
17. msa←key_act
18. $\mathbf{WBP}_{in}^{\sigma}$=$\mathbf{WBP}^{\sigma}$(ea, key_act)
19. $\mathbf{WBP}_{out}^{\sigma}$=$\mathbf{WBP}^{\sigma}$(ra, key_act)
20. $\mathbf{WBP}_{in}^{L}$=$\mathbf{WBP}^{L}$(ea, key_act)
21. $\mathbf{WBP}_{out}^{L}$=$\mathbf{WBP}^{L}$(ra, key_act)
22. con_{in}←conform($\mathbf{WBP}_{in}^{\sigma}$, $\mathbf{WBP}_{in}^{L}$)
23. con_{out}←conform($\mathbf{WBP}_{out}^{\sigma}$, $\mathbf{WBP}_{out}^{L}$)
24. else:
25. continue
26. end
27. top N_act←getTopN(msa, N) /* 获取 msa 中出现频率前 N 的活动 */
28. con←getMean(con_{in}−con_{out}) /* 获取此修复过程在行为上的变化度 */
29. end
30. return topN_act, con

首先，算法计算得到 σ_{out}，***score*** 和 *diff*(第 1－3 行)并初始化列表 *topN_act* 和 *con*(第 4－5 行)，紧接着循环 *diff* 中每个三元组(第 6 行)，对每个判错位置上的修复情况进行单独的分析，并将三元组分别赋给 *ep*, *ea* 和 *ra*(第 7－9 行)，意为 σ_{in} 在 *ep* 位置上的活动 *ea* 经 BERT4Log 修复后变为 σ_{out} 在 *ep* 位置上的活动 *ra*。然后初始化 con_{in}, con_{out} 和 *msa*(第 10－12 行)。使用 *getMaxScorePos* 函数得到 *ep* 位置在每一层每一头上的具有最大注意力分数的位置列表 *msp*(第 13 行)，其中 |*msp*|=*n_layers*×*n_heads*。接着逐个分析 *msp* 中每一个位置 *pos*(第 14 行)，并且需要满足条件 σ_{in}[*pos*] 等于 σ_{out}[*pos*]，即 σ_{in} 和 σ_{out} 在位置 *pos* 上的活动一致，此举是为了控制变量。用 *key_act* 表示 σ_{in}[*pos*] 并将其加入到 *msa* 列表中(第 15－17 行)，此列表用于解释第一个问题 Q1，把 *msa* 中出现频率较高的活动看作 BERT4Log 将 *ea* 判错的最关键的因素(第 27 行)。第二个问题的解释利用了活动对之间的 $\boldsymbol{WBP}^{\sigma}$ 和 $\boldsymbol{WBP}^{L}$(第18－23行)。计算 σ_{in} 中活动对(*ea*, *key_act*)之间的 $\boldsymbol{WBP}_{in}^{\sigma}$，以及 σ_{out} 中活动对(*ra*, *key_act*)之间的 $\boldsymbol{WBP}_{out}^{\sigma}$(第 18－19 行)，接着找出 $\boldsymbol{WBP}^{L}$ 中这两个活动对的弱行为轮廓(第 20－21 行)，最后分别计算 $\boldsymbol{WBP}_{in}^{\sigma}$, $\boldsymbol{WBP}_{out}^{\sigma}$ 和 $\boldsymbol{WBP}^{L}$ 的行为符合度(第 22－23 行)，并对比修复后的结果是否更符合原事件日志(第 28 行)。若提升，则说明 BERT4Log 将 σ_{in} 中的 *ea* 改为 σ_{out} 中的 *ra*，可能的原因是该模型判断 *ep* 位置上出现活动 *ra* 是较为符合其在预训练期间学习到的行为约束 $\boldsymbol{WBP}^{L}$，从而对 Q2 进行解释。至此，对 σ_{in} 中一个位置上的修复过程进行了完整的解释(第 28 行)，同理循环 *diff* 中其他元组来得到对整条迹中修复过程的解释。

4 案例分析

本节首先介绍原始事件日志、预训练和微调阶段对数据集的处理、两种衡量日志修复性能的评价指标以及预训练 BERT4Log 的具体参数；然后展示 BERT4Log 对于不正确和缺失问题的修复性能并与自编码器[4]进行对比分析；最后对事件日志中的具体流程实例进行修复结果的解释。

4.1 实验设计

本文使用的公开数据来源于 4TU[1)]。其中 Bpic2012 是记录关于荷兰大型金融机构贷款申请流程的事件日志；

1) https://data.4tu.nl/search?q:=keyword:%20real%20life%20event%20logs

Bpic2013 记录了关于沃尔沃 IT 问题管理系统的流程实例；Bpic2017 与 Bpic2012 相似，但其记录的每个申请可以包含多个报价并进行跟踪；Bpic2020_PL 是与差旅费用索赔相关业务的子流程，记录了与国际申报相关的流程；Hospital Billing 来自于某地区医院系统的财务模块，其中包含与医疗服务计费相关的流程；Nasa 是记录关于 NASA 船员探索车软件的事件日志，其包含方法级别的事件，并描述了详尽的单元测试套件的单次运行流程。上述事件日志的具体统计信息如表 2 所列，其中，Cases 和 Events 分别为事件日志中迹和事件的个数，Trace Variants 和 Activities 分别为不同活动序列和不同活动种类的个数，Max. case length 和 Avg. case length 则分别指迹的最大长度和平均长度。

表 2 事件日志的统计描述

Table 2 Statistical description of event logs

Datasets	Cases	Trace Variants	Events	Activities	Max. case length	Avg. case length
Bpic2012	13087	4366	262200	36	175	41.79
Bpic2013	7554	2278	65533	13	123	8.68
Bpic2017	31509	16692	1202267	66	180	38.16
Bpic2020_PL	7065	1478	86581	51	90	12.25
Hospital Billing	100000	1020	451359	18	217	4.51
Nasa	2566	2513	73638	94	50	28.70

为了分析本文方法的修复性能，本文选取目前得到该场景最优结果的文献[4]作为对比方法。该研究将低质量事件日志的修复过程分为两个阶段，分别为清理和重构，前者根据阈值找出可能出错的活动，后者根据前者的结果进行重新预测并作为修复结果，两个阶段的预测模型为自编码或其相关变体。同对比方法[4]一样，本文假设这些公开的原始事件日志是没有噪音的，由于它们没有对应的包含噪音标签的公开数据集，因此还需进行手动加噪。不同的是，本文仅考虑单属性，即充分表达行为的活动序列，而不考虑时间等其他属性。这样可以使 BERT4Log 在预训练期间专注于学习事件之间的行为约束，若输入多属性信息，则可能会偏向学习事件属性之间的关联。并且活动作为事件最基本的属性可使实验结果具有更加宽松、广泛的对比条件和空间，从而保证模型的泛用性。由于文献[4]的方法需要输入时间属性，为了公平比较，对于对比方法，本文将时间属性设为固定值，以此减少其带来的影响。

在预训练阶段，由于事件日志中的样本量远小于自然语言处理中的语料库，为了充分利用原始日志，即每条迹中的所有活动都有可能被选入到遮掩候选集C_{mask}中，这需要一条迹产生多个样本。给定一个无噪音事件日志，记为$log_{\text{no_noise}}$，对于$\forall\sigma\in log_{\text{no_noise}}$，$\alpha\in[0,1]$，随机选取 $\max(1,|\sigma|\times\alpha)$个不同位置上的活动构成$C_{\text{mask}}$，结合 3.1 节中介绍的遮掩策略，重复执行直到产生 10 个样本。在微调阶段，为了尽可能全面地模拟现实事件日志中的噪音，同样由一条迹产生多个样本。具体地，对于$\forall\sigma\in log_{\text{no_noise}}$，$\beta\in[0,1]$，随机选取 σ 中 $\max(1,|\sigma|\times\beta)$个不同位置上的活动构成噪音位置的候选集，记为$C_{\text{noise}}$，其中 β 为含噪率。接着执行替换和去除操作来模拟导致事件日志质量低下的两种主要噪音，即不正确和缺失。其中，替换是将C_{noise}中的活动换成不同于本身的其他活动，而去除是将C_{noise}中的活动直接换成[MASK]标志。最后，两个操作分别执行多次直到产生$\langle 1/\beta\rangle+1$个样本。在本文中，选取$\alpha=\{0.15,0.55\}$，$\beta=\{10\%,20\%,30\%,40\%,50\%\}$，并将 $\alpha=0.15$ 对应的 BERT4Log 模型记为 LowMask，$\alpha=0.55$ 对应的 BERT4Log 模型记为 HighMask，而作为对比方法的文献[4]中综合性高的自编码模型记为 AE。接着，本文分别在 6 个事件日志和5种含噪率共30种情况下对比3种模型的修复性能。

不同于一般的多分类预测任务，不正确噪音的修复方法应尽可能地将正确的活动预测为自身以及将错误的活动预测为正确的活动。为此，本文分别使用 TPR(True Positive Rate)和 FPR(False Positive Rate)作为验证指标，两者皆由混淆矩阵计算得到。混淆矩阵分为 4 个部分，即 TP，TN，FN 和 FT，分别表示正例被预测为正、负例被预测为负、正例被预测为负，以及负例被预测为正的情况对应的样本数。在本文中，正例代表事件日志中正确的活动，负例代表不正确的活动，由此可以得到以下公式：

$$TPR=\frac{TP}{TP+FN}$$

$$FPR=\frac{FP}{FP+TN}$$

$$AVG=\frac{TPR+FPR}{2}$$

其中，TPR 用于衡量正确的活动有多少保持不变，FPR 用于衡量不正确的活动有多少被修复为正确的活动，AVG 用于衡量修复方法的综合性能。此外，本文还使用了定义 10 中的行为相似度 BS 来衡量日志在行为上的修复程度，即计算修复前后日志的弱行为轮廓与$Log_{\text{no_noise}}$的弱行为轮廓之间的 BS，验证经过修复后 BS 是否会有所提升。由于缺失位置上没有原生的活动，可以将其看成简单的二分类任务，因此本文仅使用准确率来衡量此类噪音的修复性能。

对于 BERT4Log 中的网络参数，本文选择 $d\_model=256$，$d_q=d_k=d_v=64$，$n\_heads=6$ 和 $n\_layers=3$，而最大长度根据表 2 中的 Max. case length 项决定。经过多次实验，预训练的参数选择为 $batch\_size=\{16,32,64\}$，$epoch=100$，使用自动下降的学习率，起始为 0.1，训练 70 个 $epoch$ 后下降为 0.01，损失函数选用交叉熵，并使用 Adadelta 算法进行优化。微调阶段冻结 BERT4Log 的参数，然后直接输入预处理后的含噪事件日志即可完成日志修复。

4.2 模型性能

表 3 列出了不同噪音率 β 下 3 种模型对缺失活动修复的准确率。从表中可以看到，不同 β 下，LowMask 性能优于 AE

的日志比例数分别为(5/1,5/1,3/3,3/3,1/5)，由此可以看出 LowMask 在低 β 下的修复性能优于 AE，但随着 β 升高，LowMask 在大部分日志上的性能不如 AE。原因可能是 LowMask 的预训练样本中[MASK]标记最多不超过样本迹长的 15%，并没有高缺失率的样本供其学习。因此，将参数 α 提高的 HighMask 的修复性能全面优于 AE，特别是在 β 为 50% 时，其性能仍然优于前者。此外，HighMask 的性能在多数情况下优于 LowMask，但当 β 为 10%时，两者的修复性能相差无几。从事件日志视角出发，对于自循环活动较少的事件日志，例如 Nasa，3 种模型都具有较好的修复性能；但当自循环活动较多时，例如 Bpic2013，3 种模型的修复性都有一定程度的下降。原因可能是能够自循环的活动具有较强的随机性，使模型并不能很好地学习其身上的行为约束，并且其本身的出现也影响了其他缺失噪音的修复。

表 3 缺失噪音修复的准确率

Table 3 Accuracy of missing noise repair

dataset	β=10%			β=20%			β=30%			β=40%			β=50%		
	AE	Low Mask	High Mask	AE	Low Mask	High Mask	AE	Low Mask	High Mask	AE	Low Mask	High Mask	AE	Low Mask	High Mask
Bpic2012	0.795	0.975	**0.992**	0.786	0.947	**0.984**	0.765	0.910	**0.975**	0.726	0.859	**0.964**	0.666	0.788	**0.945**
Bpic2013	0.717	0.677	**0.807**	0.700	0.632	**0.803**	0.665	0.583	**0.789**	0.633	0.539	**0.761**	0.575	0.497	**0.728**
Bpic2017	0.736	0.938	**0.988**	0.733	0.905	**0.982**	0.728	0.851	**0.975**	0.719	0.768	**0.964**	0.706	0.656	**0.945**
Bpic2020_PL	0.750	0.920	**0.976**	0.746	0.821	**0.961**	0.729	0.728	**0.942**	0.700	0.616	**0.907**	0.664	0.512	**0.861**
Hospital Billing	0.838	0.905	**0.916**	0.840	0.869	**0.932**	0.824	0.805	**0.923**	0.779	0.730	**0.891**	0.733	0.643	**0.860**
Nasa	0.901	0.995	**0.999**	0.899	0.978	**0.998**	0.894	0.949	**0.994**	0.885	0.908	**0.988**	0.861	0.842	**0.969**

表 4 列出了不同含噪率下 HighMask 与 AE 对于不正确噪音修复性能的对比。可以看到 HighMask 的综合性能在多数情况下优于 AE，即前者的 *AVG* 高于 AE 的情况占比为(27/30)。对于低 β 的事件日志，HighMask 具有更明显的优势，但是随着 β 的升高，HighMask 的 *AVG* 下降速度快于 AE，并且 β 为 50%时仅在一半的日志修复上优于 AE。具体分析 TPR 和 FPR 可以发现，HighMask 在所有情况下的 *TPR* 都优于 AE，这说明前者不易将正确的活动预测为错误的活动。对于错误活动的修复，HighMask 在低 β 下的 *FPR* 优于 AE，但在高 β 下，例如 β 为 50%时，其仅在两个事件日志上优于 AE。这说明当一条迹中不正确活动较多时，HighMask 不能很好地对不正确噪音进行修复。原因是在微调阶段，HighMask 模型预测每个位置上的活动时利用了所有活动的信息，而较多的不正确活动可能会给模型预测带来负面影响。但这也从侧面反映出 HighMask 在预测缺失活动时考虑了与迹中其他的活动之间的行为关系。

表 4 不正确噪音修复的性能对比

Table 4 Performance comparison of incorrect noise repair

dataset	Class	β=10%		β=20%		β=30%		β=40%		β=50%	
		AE	High Mask	AE	High Mask	AE	High Mask	AE	High Mask	AE	High Mask
Bpic2012	*TPR*	0.814	**0.998**	0.811	**0.996**	0.803	**0.990**	0.785	**0.980**	0.746	**0.962**
	FPR	0.794	**0.945**	0.785	**0.913**	0.764	**0.868**	0.720	**0.805**	0.645	**0.719**
	AVG	0.804	**0.972**	0.798	**0.955**	0.784	**0.929**	0.752	**0.893**	0.696	**0.841**
Bpic2013	*TPR*	0.899	**0.974**	0.897	**0.969**	0.892	**0.961**	0.889	**0.954**	0.881	**0.939**
	FPR	0.621	**0.647**	0.595	**0.613**	0.578	**0.582**	**0.549**	0.545	**0.518**	0.500
	AVG	0.760	**0.810**	0.746	**0.791**	0.735	**0.771**	0.719	**0.750**	0.700	**0.720**
Bpic2017	*TPR*	0.760	**0.998**	0.759	**0.996**	0.757	**0.991**	0.754	**0.982**	0.748	**0.965**
	FPR	0.733	**0.953**	0.729	**0.924**	0.720	**0.880**	0.709	**0.817**	0.690	**0.725**
	AVG	0.746	**0.976**	0.744	**0.960**	0.738	**0.935**	0.732	**0.899**	0.719	0.845
Bpic2020_PL	*TPR*	0.827	**0.995**	0.824	**0.990**	0.821	**0.982**	0.818	**0.967**	0.809	**0.943**
	FPR	0.733	**0.808**	0.731	**0.750**	**0.705**	0.669	**0.676**	**0.587**	**0.634**	0.499
	AVG	0.780	**0.901**	0.777	**0.870**	0.763	**0.826**	0.747	**0.777**	**0.722**	0.721
Hospital Billing	*TPR*	0.938	**0.987**	0.935	**0.982**	0.931	**0.971**	0.926	**0.955**	0.918	**0.932**
	FPR	0.753	**0.796**	0.748	**0.775**	**0.711**	0.710	**0.666**	0.637	**0.611**	0.558
	AVG	0.845	**0.891**	0.842	**0.879**	0.821	**0.841**	**0.796**	0.796	**0.764**	0.745
Nasa	*TPR*	0.907	**0.999**	0.905	**0.997**	0.902	**0.993**	0.898	**0.983**	0.892	**0.963**
	FPR	0.897	**0.982**	0.893	**0.962**	0.888	**0.927**	0.878	**0.878**	**0.859**	**0.805**
	AVG	0.902	**0.990**	0.899	**0.980**	0.895	**0.960**	0.888	**0.930**	0.875	**0.884**

为了进一步验证 BERT4Log 是否学习到弱行为轮廓关系以及对其修复的程度，本文按照定义 10 计算不同 β 下不正确活动的事件日志与 log_{no_noise} 之间的行为相似度 *BS*，并将其视作基准，记为 Baseline。同理分别计算出经 HighMask 与 AE 修复后的日志与 log_{no_noise} 之间的 *BS*，比较结果如图 3 所示。可以看出，不同 β 下，HighMask 的 *BS* 全部高于 Baseline，特别是在低 β 下，前者的 *BS* 接近于 1，这说明 HighMask 在预训练期间学习到了 log_{no_noise} 中的弱行为轮廓。其次，随着

β的增加，Baseline 和 HighMask 的 BS 逐步下降，可以看出 HighMask 对日志行为关系的修复依赖于日志中正确活动的占比。对于 AE，其 BS 在不同的 β 下趋于稳定，并没有呈现出随 β 升高而下降的趋势，因此 AE 没有学习到弱行为轮廓关系，而可能是学到了其他的统计关系。对比 HighMask 与 AE 的 BS，前者在大部分情况下优于 AE，仅在图 3(e)中 β 为 50%时低于 AE，但是这种现象是正常的，因为 AE 的 BS 对 β 的依赖程度较小。综上所述，HighMask 学习到了弱行为轮廓关系，以及实现了在行为上对日志进行修复，并且其性能整体优于 AE。

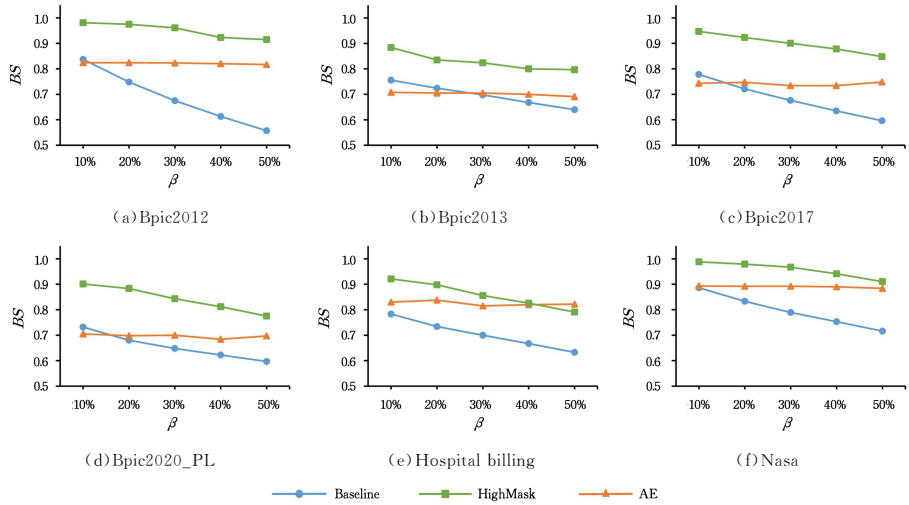

图 3 不正确噪音修复前后日志的行为相似度

Fig. 3 Behavior similarity of logs before and after incorrect noise repair

4.3 修复结果解释

为了对修复结果进行具体分析，本文选择一条关于 Nasa 的长度为 14 的含噪音的迹，并将其记为 Noise。图 4 给出了 Noise 在 HighMask 中不同 Encoder 层上得到的注意力分数矩阵，其中每个子图从左到右和从上到下的方向代表位置 1 到位置 14，每个小方块表示迹中两个位置对应活动之间的注意力分数 s。将每一行的 s 标准化以便观察。从不同的注意力层上可以看到，Layer1 上所有 Head 中高 s 的分布是较为单一的，即每一行的最大 s 与其余多数 s 的差距较大。然而，在 Layer2 和 Layer3 中最大 s 与其余部分 s 的差距较小，这可能是由于高层的输入包含了位于低层中所有位置上的信息，使得高层注意力块更容易得到每个位置上的信息。接着从不同的注意力头中可以看到，底层 Head 学习的行为约束较为简单，例如 Layer1 上 Header2 偏向于关注后一个位置，而高层 Head 可能学习到更加隐晦的特征，比如同时在多个位置具有较高的 s，以此来关注多活动之间的潜在行为约束。但不同于 NLP 中具有明确语义的文本，事件日志中较高级别的语义是难以理解的，因此，由注意力机制得到的对修复结果影响较大的关键活动仅可以作为 Q1 的一种回答，但不能回答 Q2，因此需要 Explan 算法做进一步的分析。

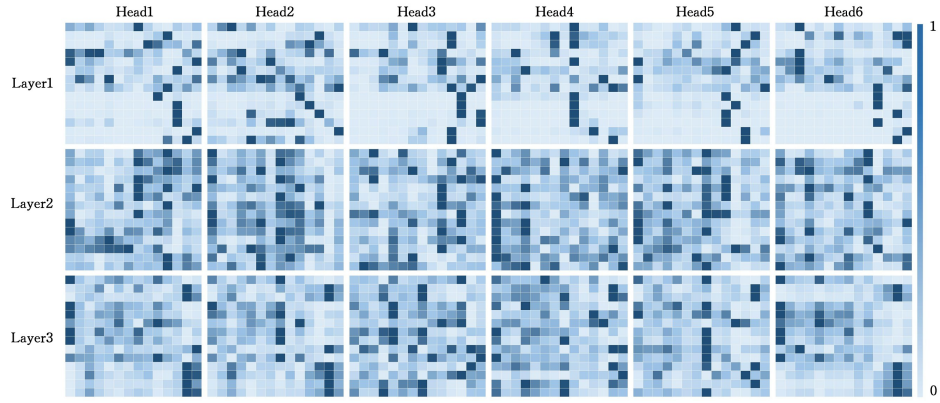

图 4 一个 Nasa 实例的注意力分数

Fig. 4 Attention score of a Nasa instance

图 5 给出了 Noise 具体的细节和修复过程，由于日志 Nasa 的活动种类较多，所以本文使用数字去代替活动。为了与位置 Pos 中的数字区分开，将第 i 个位置表示为 Pos_i。Label 指无噪音的迹，用于展示修复结果是否真的正确，但 Explan 算法并不需要其作为输入。Noise 中的噪音使用灰色方块表示，其包含了 2 个缺失(由于缺失噪音没有原生活动，为了同不正确噪音做相似处理，本文用标记[MASK]对应单词表的下标来充当原生活动，并记为数字 3)以及 3 个不正确活动。Repair1 表示 Noise 经过一层 Encoder 后跨过剩余的 Encoder 直接输入到后续网络

中得到的结果，同理 Repair2 表示经过两层 Encoder 后的输出结果，这些中间层的输出可以方便观察活动在各层的变化。Output 则是 Noise 完整的通过 3 层 Encoder 后得到的预测结果，其中绿色方块代表该位置上噪音被正确修复。按照 Explan 的步骤，首先对比 Noise 和最终输出 Output 可以得到 HighMask 判错的位置为 $\{Pos_2, Pos_3, Pos_7, Pos_9, Pos_{14}\}$，接下来对各个位置上的修复结果做单独的分析。

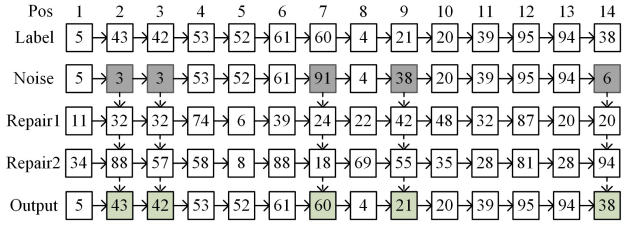

图 5 Nasa 中包含缺失和不正确活动的修复流程实例（电子版为彩图）

Fig. 5 Repair process instance with missing and incorrect activity in Nasa

图 6 给出了 Pos_2 上缺失活动被修复为活动 43 的过程，其中虚线箭头上方的 MaxScorePos 是 Explan 算法选取每个 Head 上具有最大注意力分数的关键位置，该变量可由图 4 获得。虚线箭头下方则描述修改活动对应行为符合度 *conform* 的变化，其中绿色矩阵表示 *conform* 上升，白色矩阵表示不变，最后一行的 Average 表示经过该层所有头后 *conform* 的平均增加程度。具体地，MaxScorePos1 中的第一个元素为 Layer1 中 Head1 上计算得到的 14 个位置上具有最大注意力分数的位置，即 Pos_{11}。接着从 Noise 中找到对应关键位置上的活动为 39，由于缺失标记 3 被修改为活动 32，因此根据 Explan 算法，分别计算活动对(3,39)与活动对(32,39)的弱行为轮廓。然后对比 Nasa 对应的 $\boldsymbol{WBP}^L$，按照定义 9 计算出两个活动对的 *conform*。由于缺失标记 3 未在 $\boldsymbol{WBP}^L$ 中出现，因此(3,39)对应的 *conform* 为 0，而修改后的活动对(32,39)则在 $\boldsymbol{WBP}^L$ 中出现，计算出其对应的 *conform* 为 1。而 Head4 中的两个活动对均未在 $\boldsymbol{WBP}^L$ 中出现，因此其符合度变化为 0 到 0。由于包含缺失标记的活动对的 *conform* 必为 0，因此对于缺失修复，第一层 *conform* 增加的程度较大，平均达到 0.833。紧接着 Repair1 输入到第二层多头注意力块，除了该层需要在 Repair1 中找 MaxScorePos2 对应的活动，活动 32 被修改成活动 88，其他过程与第一层类似。例如 Head2 中 Pos_8 对应的关键活动 22，构建活动对(32,22)和(18,22)并求得符合度上升 0.167，最后一层同理。从 3 个 MaxScorePos 中可以得到对 Pos_2 缺失修复较为关键的位置为 Pos_{13} 和 Pos_8。对于 Q1，缺失标记对应预训练样本中不存在原活动表中的[MASK]，这使得 HighMask 能直接知道哪些位置存在缺失噪音，而 Q2 的原因则为，修复后的活动和对其具有最高注意力的另一个活动，构成的活动对与事件日志的行为符合度更高，即 HighMask 认为修改后的活动更加符合其预训练期间学习的弱行为轮廓。

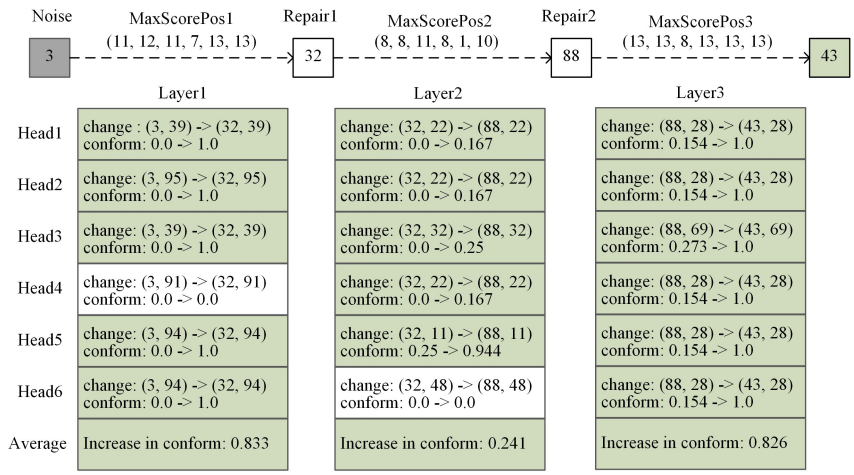

图 6 Nasa 实例中 Pos_2 上缺失活动被修复为活动 43 的过程（电子版为彩图）

Fig. 6 Process of repairing missing activity on Pos_2 to activity 43 in Nasa instance

图 7 给出了不正确活动 38 到活动 21 的修复过程，可以看到出现了较多的橙色（*conform* 下降）和白色矩阵，表明修复不正确活动比修复缺失活动难度更大。第二层仅有一个修复变化的 *conform* 是正增长的，可能的原因是不正确活动 38 在第一层已经得到了较大程度的修复，这一点可以从符合度的平均增长值为 0.833 得出，因此第二层的修复可能会在符合度上呈现一定程度的倒退。这里第一层的 Average 不同于图 6 中第一层的 Average，虽然它们的数值相同，但包含不正确噪音的活动对是有可能出现在 $\boldsymbol{WBP}^L$ 中的，只是这里修复前的情况恰好都不属于 $\boldsymbol{WBP}^L$，这也说明了经过第一层的多头注意力块后，Repair1 中的关键活动对更加符合 Nasa 的弱行为轮廓。经过第二层后 *conform* 平均下降 0.104，但在第三层的修复中开始回升，即使上升程度不高，但 Head1 到 Head4 的 *conform* 都保持了最大值，即 1；并且 Head6 上符合度下降的程度也非常小，在总体上维持高 *conform* 的同时平均增加了 0.161。同图 6 描述的修复流程一样，Q1 可以由各个 MaxScorePos 中对应的关键活动来回答。而图 7 中呈现的整体上升并且维持高值的行为符合度也可以回答问题 Q2。

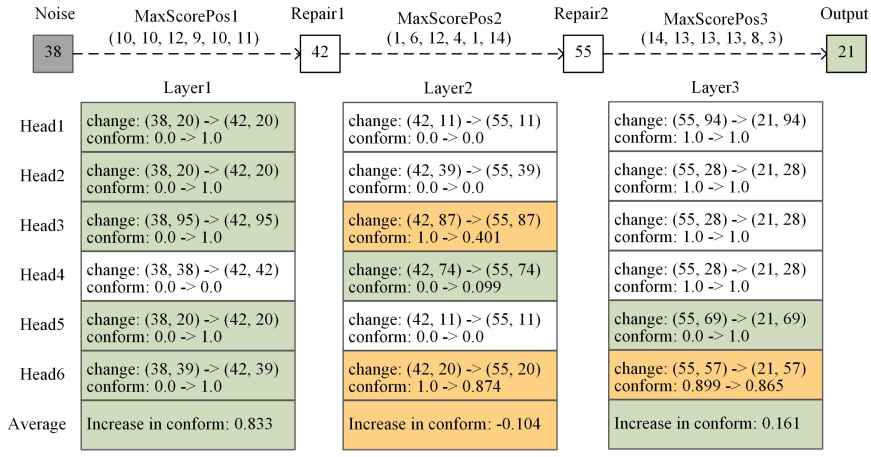

图 7 Nasa 实例中Pos_9上不正确活动 38 被修复为活动 21 的过程(电子版为彩图)

Fig. 7 Process of repairing incorrect activity 38 on Pos_9 to activity 21 in Nasa instance

图 6 和图 7 的一个共同点是修复实例在中间层的预测活动与最终输出有较大的差别,原因可能是原本的 HighMask 具有 3 层 Encoder,跳过一些块而直接输入到后续网络层中可能会产生一定的偏差,但这并不影响最终修复结果的正确性。有趣的是,将中间层的输出重新输入到 HighMask 中得到的修复结果仍然是正确的,这表明中间层的输出与 Output 的特征在一定程度上相似,但这些相似的特征并没有表现在中间层修复前后的迹的形式上,而是表现在弱行为轮廓或其他潜在的行为约束上。综上所述,利用注意力分数和弱行为轮廓的解释算法 Explan 可以为关于日志修复结果的两个主要问题提供一种解释方式。另外值得关注的是,对于 BERT4Log 超参数的选取问题,遮掩策略中 α 的取值对 BERT4Log 的修复性能影响较大。从表 3 中可以明显看到,相比采用 BERT 原始数值 $\alpha=0.15$ 时 LowMask 的修复性能,采用 $\alpha=0.55$ 的 HighMask 的性能在低含噪率和高含噪率的情况下均更优,这也是本文在后续实验中用 HighMask 对比 AE 的原因;并且从表 4 和图 3 中可以看出,HighMask 确实得到了整体优于对比方法 AE 的性能。因此,本文建议在预训练 BERT4Log 时选用较高的 α 值。

结束语 在无可参考过程模型的条件下,现有的方法没有利用 BERT 进行事件日志修复以及行为视角上修复结果的可解释过程。基于此,本文提出了 BERT4Log 模型来修复低质量事件日志中存在的缺失噪音以及不正确噪音,并结合弱行为轮廓理论和多层多头注意力机制对修复结果进行解释。本文在 6 个事件日志和 5 种含噪率下进行对比实验,验证了本文方法修复低质量事件日志的有效性和优越性。不同于文献[1,16]中仅利用注意力分数将影响 PPM 预测结果的关键因素作为解释,或者文献[22-23]利用深度学习社区中相关解释算法对 PPM 预测结果进行解释,本文更加专注于过程挖掘领域中十分重要的行为视角,将弱行为轮廓这种行为约束作为一种预训练修复模型可以学习的语义,并且利用行为相似度证明了 BERT4Log 确实学习到了弱行为轮廓以及量化事件日志在行为上的修复程度。所提出的 Explan 算法可以解释与修复结果相关的两个主要问题,即为什么判错以及为什么修改成某个其他的活动,并在具体实例中结合行为符合度进行描述。

未来工作可以尝试利用行为轮廓针对事件日志的特性,在相关学习模型和网络结构上进行一些改进或适应某种优化,以期获得更好的性能表现。例如,添加单独的判错器结合行为轮廓使其将不正确噪音修复引入到性能较好的缺失噪音修复上,或是类比 NLP 中存在不同种类的语义,考虑不同的行为约束,验证是否能被该模型学习并对修复结果进行更加细致和深入的解释。

参 考 文 献

[1] WICKRAMANAYAKE B, HE Z, OUYANG C, et al. Building interpretable models for business process prediction using shared and specialised attention mechanisms [J]. Knowledge-Based Systems, 2022, 248: 108773.

[2] MOON J, PARK G, JEONG J. Pop-on: Prediction of process using one-way language model based on nlp approach [J]. Applied Sciences, 2021, 11(2): 864.

[3] BATINI C, CAPPIELLO C, FRANCALANCI C, et al. Methodologies for data quality assessment and improvement [J]. ACM Computing Surveys (CSUR), 2009, 41(3): 1-52.

[4] NGUYEN H T C, LEE S, KIM J, et al. Autoencoders for improving quality of process event logs [J]. Expert Systems with Applications, 2019, 131: 132-147.

[5] BOSE R P J C, MANS R S, VAN DER AALST W M P. Wanna improve process mining results? [C]// 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). IEEE, 2013: 127-134.

[6] SARZYNSKA-WAWER J, WAWER A, PAWLAK A, et al. Detecting formal thought disorder by deep contextualized word representations [J]. Psychiatry Research, 2021, 304: 114135.

[7] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training [EB/OL]. (2018-12-30) [2022-07-15]. https://www.cs.ubc.ca/~amuham01/LING530/papers/radfod2018improving.pdf.

[8] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [J].

arXiv:1810.04805,2018.

[9] VAN DER AALST W M P,DE MEDEIROS A K A. Process mining and security: Detecting anomalous process executions and checking process conformance [J]. Electronic Notes in Theoretical Computer Science,2005,121:3-21.

[10] GHIONNA L,GRECO G,GUZZO A,et al. Outlier detection techniques for process mining applications [C]// International Symposium on Methodologies for Intelligent Systems. Springer,2008:150-159.

[11] FANISANIM,ZELST S J,VAN DER AALST W M P. Repairing outlier behaviour in event logs [C]// International Conference on Business Information Systems. Springer,2018:115-131.

[12] ROGGE-SOLTIA,MANS R S,VAN DER AALST W M P,et al. Improving documentation by repairing event logs [C]// IFIP Working Conference on the Practice of Enterprise Modeling. Heidelberg,Springer,2013:129-144.

[13] WANG J,SONG S,ZHU X,et al. Efficient recovery of missing events [J]. Proceedings of the VLDB Endowment,2013,6(10):841-852.

[14] WANG J,SONG S,LIN X,et al. Cleaning structured event logs:A graph repair approach [C]// 2015 IEEE 31st International Conference on Data Engineering. IEEE,2015:30-41.

[15] CHINCES D,SALOMIE I. Optimizing spaghetti process models [C]// 2015 20th International Conference on Control Systems and Computer Science. IEEE,2015:506-511.

[16] LIU J,XU J,ZHANG R,et al. A repairing missing activities approach with succession relation for event logs [J]. Knowledge and Information Systems,2021,63(2):477-495.

[17] SURIADI S,ANDREWS R,TER HOFSTEDE A H M , et al. Event log imperfection patterns for process mining:Towards a systematic approach to cleaning event logs [J]. Information Systems,2017,64:132-150.

[18] NOLLE T,LUETTGEN S,SEELIGER A,et al . Analyzing business process anomalies using autoencoders [J]. Machine Learning,2018,107(11):1875-1893.

[19] NGUYEN H T C,COMUZZI M. Event log reconstruction using autoencoders [C]// International Conference on Service-Oriented Computing. Springer,2018:335-350.

[20] VASWANI A,SHAZEER N,PARMAR N,et al. Attention is all you need [C]// Advances in Neural Information Processing Systems. 2017:5998-6008.

[21] VAN DER A A H,REBMANN A,LEOPOLD H. Natural language-based detection of semantic execution anomalies in event logs [J]. Information Systems,2021,102:101824.

[22] RIZZI W,DI FRANCESCOMARINO C,MAGGI F M. Explainability in predictive process monitoring: when understanding helps improving [C]// International Conference on Business Process Management. Springer,2020:141-158.

[23] GALANTI R,COMA-PUIG B,DE LEONI M,et al. Explainable predictive process monitoring [C]// 2020 2nd International Conference on Process Mining(ICPM). IEEE,2020:1-8.

[24] WEIDLICH M,MENDLING J,WESKE M. Efficient consistency measurement based on behavioral profiles of process models [J]. IEEE Transactions on Software Engineering,2010,37(3):410-429.

[25] FANG H,JIN P P,FANG X W,et al. Process variants cluster mining method based on causal behavioral profiles [J]. Computer Integrated Manufacturing System,2020,26(6):1538-1547.

[26] FANG H,FANG X W,WANG L L. Review of Reliability Analysis Based on Petri Nets [J]. Computer Science,2014,41(7):40-44.

[27] FANG H,SUN S Y,FANG X W. Behavior change mining methods based on incomplete logs conjoint occurrence relation[J]. Computer Integrated Manufacturing System,2020,26(7):1887-1895.

LI Binghui, born in 1998, postgraduate, is a member of China Computer Federation. His main research interests include process mining and deep learning.

FANG Huan, born in 1982, postgraduate supervisor, professor, Ph. D. Her main research interests include Petri nets theory and application, behavioral profiles, change mining and process mining.

(责任编辑:何杨)