Software Engineering and Applications 软件工程与应用, 2025, 14(5), 1013-1025
Published Online October 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.145090

Hans汉斯

基于结构知识增强的网络设备日志理解方法

曹祥龙，张明西*，殷菘泽，李雨辰，王凌璇

上海理工大学出版学院，上海

收稿日期：2025年8月28日；录用日期：2025年9月18日；发布日期：2025年9月26日

摘　要

随着网络系统规模和复杂性的不断增长，系统日志已成为故障诊断和运维管理的重要数据源。然而，现有的日志理解方法普遍忽视了日志文本的结构化特征以及系统组件间的关联关系，导致在复杂故障场景下的理解能力有限。为解决这一问题，本文提出了一种基于结构知识增强的网络设备日志理解方法。首先，我们构建了包含设施－错误－严重性三层语义关系的日志知识图谱，显式建模系统组件间的故障传播路径和依赖关系。在此基础上，设计了结构化掩码预测任务，通过对日志中的关键结构化字段采用更高的掩码概率，引导模型重点学习系统架构和错误类型的语义表示。同时，提出了图神经网络增强的文本对齐机制，并通过自注意力机制动态融合多实体图嵌入，实现知识图谱结构信息与文本语义的有效对齐。实验结果表明，所提的方法在多个任务指标上显著优于主流基线模型，验证了其各任务下的有效性与泛化能力。

关键词

日志理解，知识图谱，图神经网络，预训练模型

Structural Knowledge-Enhanced Approach for Network Device Log Understanding

Xianglong Cao, Mingxi Zhang*, Songze Yin, Yuchen Li, Lingxuan Wang

College of Publishing, University of Shanghai for Science and Technology, Shanghai

Received: August 28, 2025; accepted: September 18, 2025; published: September 26, 2025

Abstract

With the continuous growth in scale and complexity of network systems, system logs have become an important data source for fault diagnosis and operations management. However, existing log

*通讯作者。

文章引用: 曹祥龙, 张明西, 殷菘泽, 李雨辰, 王凌璇. 基于结构知识增强的网络设备日志理解方法[J]. 软件工程与应用, 2025, 14(5): 1013-1025. DOI: 10.12677/sea.2025.145090

understanding methods generally neglect the structured characteristics of log texts and the associative relationships among system components, resulting in limited understanding capabilities in complex fault scenarios. To address this issue, this paper proposes a structural knowledge-enhanced approach for network device log understanding. First, we construct a log knowledge graph encompassing three-layer semantic relationships of facility-error-severity, explicitly modeling fault propagation paths and dependency relationships among system components. Building upon this foundation, we design a structured masking prediction task that employs higher masking probabilities for key structured fields in logs, guiding the model to focus on learning semantic representations of system architecture and error types. Meanwhile, we propose a graph neural network-enhanced text alignment mechanism that dynamically fuses multi-entity graph embeddings through self-attention mechanisms, achieving effective alignment between knowledge graph structural information and textual semantics. Experimental results demonstrate that the proposed method significantly outperforms mainstream baseline models across multiple task metrics, validating its effectiveness and generalization capability across various tasks.

Keywords

Log Understanding, Knowledge Graph, Graph Neural Network, Pre-Trained Model

Copyright © 2025 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

网络设备(如路由器、交换机和防火墙)生成的日志数据是监控系统健康状况、诊断故障和检测异常的关键资源。这些日志由结构化的模板与非结构化的动态参数部分组成[1] [2]，这种混合特性使得日志既不完全是结构化数据(如数据库表)，也不完全是非结构化数据(如纯文本文档)，而是介于两者之间的半结构化形式。因此日志分析既需要结构化处理能力，又需要文本语义理解能力。在现代复杂的网络系统中，单日生成的日志数据量常常超过 1 TB [3]，面对如此海量的非结构化数据，即便是经验丰富的运维专家也难以依靠传统的人工方式实现高效、全面的分析与故障定位。因而，发展自动化的日志理解与分析技术已成为保障系统可用性与可靠性的关键路径。

在现有的日志分析任务中深度学习方法凭借其端到端的建模能力与优越的特征表达性能，成为当前研究的主流方向。现有的深度学习方法主要可分为两类：其一是基于传统神经网络架构的方法，如卷积神经网络(CNN)和循环神经网络(RNN) [4] [5]。这类方法通常结合词嵌入(word embedding)或事件编码技术，将原始日志序列转化为向量表示，并通过神经网络建模其语义与时序特征，从而实现异常检测、分类识别等任务。这类方法具有结构相对简单、可解释性较强的优势，适用于中小规模日志数据场景。其二是基于预训练语言模型的方法，如 BERT [6]、RoBERTa [7]等。这类方法在大规模自然语言语料上进行预训练，以学习通用的语言表示能力，随后通过微调(fine-tuning)适配日志领域的具体任务。由于其具备强大的上下文建模与语义理解能力，在多种日志分析任务中表现出显著优势，尤其在异常检测、事件预测与语义对齐等方面取得了突破性进展。近年来，越来越多的研究开始探索如何将预训练模型与日志的外部知识相结合，进一步提升模型对日志的理解能力。

近年来，越来越多的研究开始探索如何将预训练模型与日志的领域知识相结合，以进一步提升模型的理解能力[8] [9]。然而，现有方法在处理结构化日志时仍存在两个关键不足：首先，现有方法将日志视

为普通自然语言文本，忽视其固有的结构化特性。尽管预训练语言模型在日志分析中展现出强大的语义理解能力，但现有方法普遍采用通用文本处理范式，直接将结构化日志作为无差别的字符序列输入模型，日志不同于普通文本，其遵循严格的“Facility/Severity/Mnemonic: Message”格式规范，每个字段都具有明确的语义角色和不同的信息密度。设备类型(Facility)标识了故障来源，错误代码(Mnemonic)编码了具体的异常模式，而这些结构化字段的诊断价值远超一般描述性文本。然而，当前方法采用的均匀随机掩码策略将所有 token 一视同仁，无法区分关键结构组件与普通词汇，导致模型在有限的训练资源下难以充分学习高价值信息。其次，现有方法难以识别日志结构化字段之间的关系。日志字段并非孤立存在，其诊断意义往往依赖上下文关系。例如，错误代码“NBRCHG”在不同语境下含义截然不同：当在“OSPF”模块下，其严重性级别为 2，它表示一个关键的连接问题。相反，在“LDP”模块下，严重性级别为 5 的同一错误代码可能只是反映了一个常规的邻居状态更新。然而，现有方法通常仅在词元层面进行建模，缺乏对字段间交互与全局依赖的显式表示，因而难以形成对日志的整体化理解。

为了解决上述问题，本文提出了一种基于结构知识增强的网络设备日志理解方法。在本文方法的设计过程中，主要面临以下两个关键挑战。第一，在日志中不同字段的信息密度和诊断价值差异显著，难以突出学习高价值信息。为此，我们设计了结构化掩码预测任务(SMP)通过引入字段感知的非均匀采样机制，动态提升 Facility 与 Mnemonic 等高价值字段的掩码概率，从而引导模型重点学习关键结构组件的表示。第二，领域专业知识与文本语义表示的有效整合面临挑战。网络设备日志的深层语义理解需要同时整合文本语义信息和结构化领域知识，但文本编码器学习的连续向量表示与图神经网络学习的离散结构表示存在显著的模态差异和语义对齐困难，导致结构化知识无法有效指导文本理解过程。为此，本文设计了知识图谱增强的图文对齐机制，通过引入可学习的对齐变换矩阵将图结构表示映射到文本语义空间，并采用余弦相似度损失函数约束同一实体在不同模态下的表示一致性，同时设计自适应权重机制根据知识图谱匹配程度动态调整对齐强度，从而实现文本语义与图结构知识的深度融合和协同学习。总的来说，本研究的主要贡献如下：

- 设计了一种基于结构化实体感知的掩码预测任务，通过字段感知的非均匀采样策略，提升关键字段的掩码概率，引导模型重点学习高价值结构信息，缓解传统均匀掩码策略的信息稀释问题。
- 提出了一种知识图谱增强的图文对齐机制，通过图神经网络建模日志知识图谱，并设计可学习投影与自适应对齐约束，实现文本语义与结构化知识的协同优化，提升模型对日志字段全局依赖与诊断语义的理解能力。
- 在不同的日志理解任务上进行了系统实验与消融分析，实验结果显示，本文方法在多个日志理解任务上均显著优于主流基线方法，充分验证了所提方法的有效性、鲁棒性与适应性。

2. 相关工作

日志分析作为网络运维的关键环节，已经从传统的基于规则和模式匹配的方法发展到采用深度学习技术。Zhang 等人[10]提出 LogRobust，结合注意力机制和 CNN 改进了日志表示学习，显著提升了模型对日志模板变化的适应性。Meng 等人[11]设计了 LogAnomaly，同时考虑日志语义和数值参数，实现了更全面的异常检测。Guo 等人[12]设计了 LogBERT，通过结合 BERT 的上下文建模能力与自监督训练任务，解决了传统日志异常检测方法在长距离依赖和模式泛化上的不足，提升了系统日志分析的鲁棒性。这些方法展示了深度学习在自动提取日志特征方面的优势，但它们对日志的理解仍停留在表面层次，缺乏对日志结构和专业术语的深入理解。

为克服纯数据驱动方法的局限性，研究者开始探索结合专业知识的日志理解技术。Ma 等人[13]提出了基于知识增强的预训练语言模型 KnowLog，首次尝试将文档中的领域知识注入到预训练阶段，显著提

升了模型的日志理解能力。但该方法高度依赖人工创建和维护的文档，在实际应用中受到限制。后续又提出了一种新的知识增强框架 LUK [14]，该框架从 LLM 中获取专家知识，从而在更小的 PLM 上实现日志理解，但是相比文档中的领域知识，利用 LLM 生成的专业知识质量有所降低，所以其在相关下游任务的表现略有降低。值得注意的是，现有的知识增强方法都忽略了日志本身所带有的专业的结构化知识，如设备类型、错误分类和严重程度等关键信息。

知识图谱作为结构化知识的重要表现形式，已被用于增强文本理解。Peters 等人[15]开发了 KnowBERT，通过实体链接增强 BERT 的知识意识。Sui 等人[16]设计了 Logkg，通过知识图谱融合日志的多字段信息提升故障根因分析的准确性和效率。Liao 等人[17]提出了 LogBASA，通过构建知识图谱整合日志信息，利用图卷积和 Transformer 模型实现多维度特征融合，显著提升了异常检测的效果。这些方法表明，知识图谱可以有效补充文本表示的语义信息。知识图谱作为一种结构化的知识表示形式，不仅包含丰富的实体和关系信息，还能提供清晰的知识推理路径。将知识图谱与预训练模型相结合，有望从根本上解决现有方法的局限性，特别是对于网络日志这类同时具有结构化特征和专业语义的数据。通过构建专门针对网络日志领域的知识图谱，我们可以捕获日志中的结构化信息，并将其与文本表示统一起来，从而提升模型对日志的理解能力。

3. 日志知识图谱的构建

为了捕获日志数据中的结构化知识和故障传播模式，我们设计了一个三层异构图模型。该模型不仅编码了日志的静态属性，还建模了动态的故障传播关系。

日志知识图谱定义为$\mathcal{G}=(\nu,\varepsilon,\mathcal{A})$，其中$\mathcal{V}=\mathcal{V}_f\cup\mathcal{V}_m\cup\mathcal{V}_s$表示节点集合，包含设施(Facility)、助记符(Mnemonic)和严重级别(Severity)三种类型；$\mathcal{E}=\mathcal{E}_{fm}\cup\mathcal{E}_{ms}\cup\mathcal{E}_{ff}$表示边集合；$\mathcal{A}$表示节点和边的属性集合。

图谱中的每个节点类型携带特定的语义信息和统计属性。设施节点记录了日志总数、错误类型集合、严重级别分布等信息；助记符节点包含出现次数、影响的设施集合、主要严重级别等属性；严重级别节点则统计了关联的设施和错误类型分布。我们定义三种边类型来捕获不同层次的关系。设施–助记符边表示设施产生特定错误类型的关系，边权重定义为条件概率$w_{fm}=P(m\mid f)=\dfrac{count(f,m)}{count(f)}$。助记符–严重级别边表示错误类型与严重程度的关联，权重为$w_{ms}=P(s\mid m)=\dfrac{count(m,s)}{count(m)}$。

故障传播路径定义为图中的有向路径$p=v_f\xrightarrow{w_{fm}}v_m\xrightarrow{w_{ms}}v_s$，路径的传播概率计算为：

$$P_{path}=w_{fm}\times w_{ms}=P(v_m\mid v_f)\times P(v_s\mid v_m) \tag{2}$$

这种概率化的路径表示使我们能够量化故障传播的可能性。例如，路径“IPRT → NULL_RDB → Severity 3”的概率为$P(\text{NULL_RDB}\mid\text{IPRT})\times P(3\mid\text{NULL_RDB})=0.25\times0.92=0.23$，表明 IPRT 设施有 23%的概率通过 NULL_RDB 错误导致严重级别 3 的故障。

4. 方法

4.1. 整体框架

本研究提出的模型架构如图 1 所示，将文本编码器与图神经网络深度融合。该框架包含：日志文本编码器、知识图谱编码器、以及两个具体的预训练任务，通过预训练最终总体优化目标为：

$$\mathcal{L}=\alpha\mathcal{L}_{SMP}+\beta\mathcal{L}_{GEA} \tag{3}$$

其中α，β是平衡不同任务的权重系数。

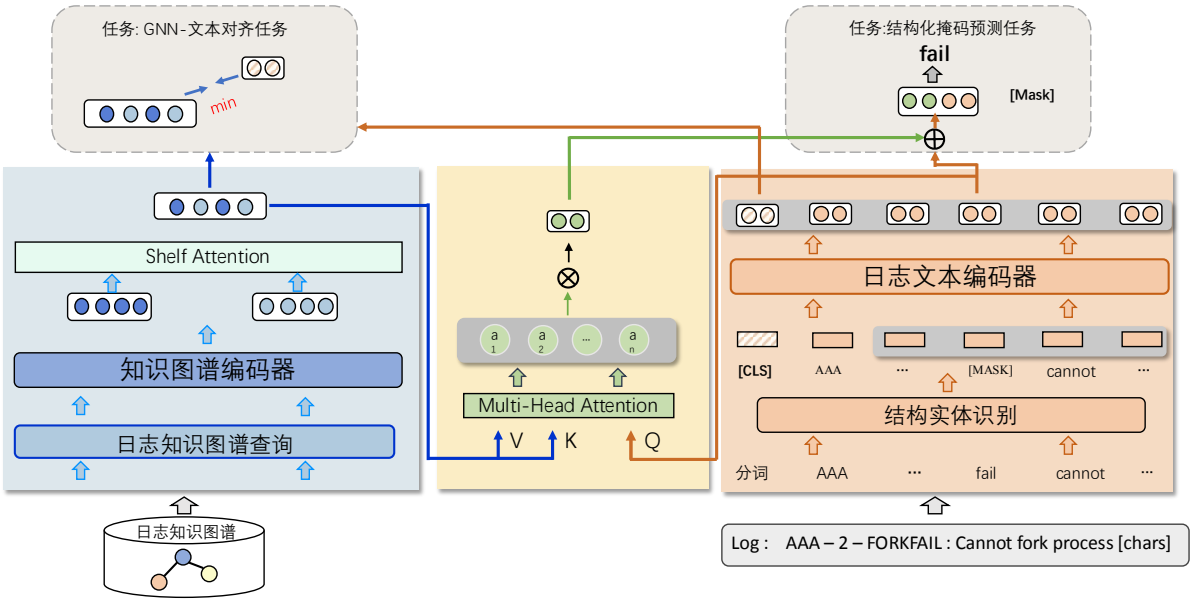

Figure 1. Framework of pre-training
图 1. 预训练框架图

4.1.1. 日志文本编码器

日志作为文本字符串，必须经过分词、编码等预处理操作后才能作为模型的输入。日志中的不仅包含部分自然语言中的单词，还包含专业领域的缩写(例如 AAA、OSPF)如果我们直接利用预训练语言模型的分词器进行分词，就会因为词汇表外(OOV)问题破坏领域缩写词的完整性(如“OSPF”可能被拆分为[“OS”, “##PF”])为了解决这个问题，我们将缩写词显式加入到分词器的词汇表中[13]。

对于给定输入日志文本 L，使用分词器将其转换为一系列 token，为日志序列 l 即 $\{[\mathrm{CLS}]l_1,l_2,\cdots,l_n[\mathrm{SEP}]\}$ 作为编码器的输入。其中特殊标记[CLS]被添加在整个序列的开头，其最终的隐藏表示被视为整个序列语义表示的整合。编码器首先通过词嵌入和位置嵌入层获得初始表示，然后通过 l 层 Transformer 层进行深层编码：

$$h^{(l)}=\mathrm{Transformer}\left(h^{(l-1)}\right),\quad l=1,\cdots,L \tag{4}$$

最终得到上下文感知的 token 表示 $H=\left\{h_1^{(L)},\cdots,h_n^{(L)}\right\}$。

4.1.2. 知识图谱编码器

基于 3.1 节构建的日志知识图谱，我们设计了一个层次化的图神经网络编码器，以学习融合了拓扑结构和传播模式的节点表示。该编码器通过异构消息传递机制和路径感知聚合策略，将离散的图结构知识转化为连续的向量表示。

对于图中的每个节点 $v\in\mathcal{V}$，我们构造初始特征向量 $X_v\in\mathbb{R}^{d_{node}}$，由节点类型编码和统计特征两部分组成：

$$x_v=\left[e_{type}(v);f_{stat}(v)\right] \tag{5}$$

其中 $e_{type}(v)\in\{0,1\}^3$ 是节点类型的 one-hot 编码(3 维，对应设施、助记符、严重级别三种类型)，$f_{stat}(v)\in\mathbb{R}^3$ 是节点的统计特征向量。

为了捕获不同类型边的语义差异，我们设计了一个路径感知的图卷积层。对于边$\left(v_i, v_j\right) \in \mathcal{E}$，消息传递过程定义为：

$$\mathbf{m}_{ij}^{(l)} = \sigma\left(\mathbf{W}_{e_{ij}}^{(l)} \mathbf{h}_j^{(l-1)} \cdot w_{ij} \cdot \alpha_{ij}^{(l)}\right) \tag{6}$$

其中$\boldsymbol{h}_j^{(l-1)} \in \mathbb{R}^d$是节点$v_j$在第$l-1$层的隐藏表示，$\boldsymbol{W}_{e_{ij}}^{(l)} \in \mathbb{R}^{d \times d}$是边类型$e_{ij}$特定的权重矩阵，$w_{ij}$是从知识图谱中获取的边权重(概率或相似度)，$\alpha_{ij}^{(l)}$是注意力权重，计算方式为：

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\text{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}\left[\boldsymbol{h}_i^{(l-1)} \,\|\, \boldsymbol{h}_j^{(l-1)} \,\|\, \boldsymbol{e}_{ij}\right]\right)\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\text{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}\left[\boldsymbol{h}_i^{(l-1)} \,\|\, \boldsymbol{h}_k^{(l-1)} \,\|\, \boldsymbol{e}_{ik}\right]\right)\right)} \tag{7}$$

其中，$\boldsymbol{a} \in \mathbb{R}^{3d}$是可学习的注意力参数，$e_{ij}$是边类型嵌入，$\mathcal{N}(i)$表示节点$v_i$的邻居集合。

图神经网络编码器由L层包含路径信息的图卷积层组成。每层的节点表示更新规则为：

$$\boldsymbol{h}_i^{(l)} = \text{LayerNorm}\left(\boldsymbol{h}_i^{(l-1)} + \text{ReLU}\left(\boldsymbol{W}_{self}^{(l)} \boldsymbol{h}_i^{(l-1)} + \sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ij}^{(l)}\right)\right) \tag{8}$$

其中$\boldsymbol{W}_{self}^{(l)}$是自连接权重矩阵。

为了更好地捕获故障传播模式，我们在消息传递过程中保留了路径的方向性。设施节点主要接收来自其产生的错误类型的反馈信息，助记符节点同时聚合上游设施和下游严重级别的信息，而严重级别节点则汇总所有相关错误类型的统计信息。这种设计确保了图编码器能够有效捕获日志系统中的因果传播关系。经过L层图卷积后，每个节点都聚合了其L跳邻域内的结构信息，最终的节点表示$\boldsymbol{h}_v^{(L)}$既包含了局部的直接关联，也编码了全局的传播模式。

4.2. 预训练任务

4.2.1. 结构化掩码预测任务

日志文本中通常包含丰富的结构化实体信息，如设备标识(Facility)和操作码(Mnemonic)等，这些实体是日志语义的关键承载。传统掩码语言模型(Masked Language Model, MLM)多采用均匀随机掩码策略，忽略了实体在日志理解中的重要性，导致模型难以充分捕捉关键结构化信息。

为增强模型对日志实体的感知能力，我们设计了结构化掩码预测任务(Structured Masked Prediction Task, SMP)。该任务在预训练过程中，优先对日志中的 Facility 和 Mnemonic 实体所在的词元进行掩码，赋予其更高的掩码概率，从而促使模型重点学习这些关键实体的语义表达。

设输入日志文本的 token 序列为$X = \left[x_1, x_2, \cdots, x_n\right]$，对应词元掩码指示变量为$m = \left(m_1, m_2, \cdots, m_n\right)$，其中$m_i = 1$表示词元$l_i$被掩码。掩码概率根据词元所属的实体类别分配为

$$p\left(t_i\right) = \begin{cases} p_{\text{facility}}, & x_i \in \mathcal{E}_{\text{facility}} \\ p_{\text{mnemonic}}, & x_i \in \mathcal{E}_{\text{mnemonic}} \\ p_{\text{other}}, & \text{otherwise} \end{cases} \tag{9}$$

其中，满足$p_{\text{facility}} > p_{\text{mnemonic}} > p_{\text{other}}$，对实体词元的优先掩码,得到掩码后的序列$\tilde{X} = \left[x_1, \cdots, [\text{MASK}], \cdots\right]$。

任务的训练目标为最小化实体感知掩码语言模型的交叉熵损失：

$$\mathcal{L}_{\text{SMP}} = -\sum_{i=1}^{n} m_i \log P_\theta\left(x_i \mid \tilde{x}\right) \tag{10}$$

其中 P_θ 是模型参数 θ 对被掩码词元的预测概率，$\tilde{x}$ 是掩码后的输入序列。该损失函数促使模型更准确地恢复被掩码的关键实体词元，提高对日志结构化信息的理解能力。

4.2.2. GNN-文本对齐任务

知识图谱编码器和文本编码器分别从结构化和非结构化两个视角学习日志表示。然而，文本编码器和图神经网络分别工作在不同的表示空间中，导致结构化知识难以有效融入文本理解过程。

为了解决这一问题，我们提出了图神经网络增强对齐任务(GNN Enhancement Alignment, GEA)，通过显式对齐文本表示和图结构表示，实现跨模态知识融合。该任务利用多层图神经网络聚合实体的多跳邻居信息，包括直接关联的错误类型、严重性分布以及相似设施等结构化知识。

具体而言，对于包含设施实体 f 和错误类型实体 m 的日志样本，我们首先通过知识图谱编码器获取相应的图嵌入表示 $\boldsymbol{h}_f^{gnn}$ 和 $\boldsymbol{h}_m^{gnn}$，形成实体嵌入集合 $\mathcal{E}=\left\{\boldsymbol{h}_f^{gnn},\boldsymbol{h}_m^{gnn}\right\}$ 然后我们采用自注意力机制动态学习各实体的权重：

$$\alpha_i=\mathrm{softmax}\left(W_{att}^{\mathrm{T}}\tanh\left(W_e\mathcal{E}_i+b_e\right)\right) \tag{11}$$

其中 $W_{att}\in\mathbb{R}^d$、$W_e\in\mathbb{R}^{d\times d}$ 为可学习参数，b_e 为偏置向量，tanh 函数用于引入非线性，softmax 确保所有实体权重之和为 1。

基于注意力权重对实体嵌入进行加权求和，得到融合多实体信息的整体图嵌入：

$$h^{gnn}=\sum_{i=1}^{N}\alpha_i\cdot\mathcal{E}_i \tag{12}$$

其中 N 表示参与融合的实体数量。

为了将图嵌入对齐到文本表示空间，我们引入可学习的对齐变换矩阵：

$$\boldsymbol{h}^{align}=\mathrm{ReLU}\left(\boldsymbol{W}_{align}\boldsymbol{h}^{gnn}+\boldsymbol{b}_{align}\right) \tag{13}$$

其中 $\boldsymbol{W}_{align}$ 是对齐权重矩阵，$\boldsymbol{b}_{align}$ 是偏置向量。

最终，GNN-文本对齐任务的损失函数定义为：

$$\mathcal{L}_{GEA}=\sum_{i=1}^{N}1_{i\in\mathcal{K}}\mathrm{P}_{path}^{(i)}\cdot\left(1-\cos\left(\boldsymbol{h}_i^{text},\boldsymbol{h}^{gnn}\right)\right) \tag{14}$$

其中 $1_{i\in\mathcal{K}}$ 是指示函数，表示样本 i 是否在知识图谱 $\mathcal{G}$ 中有对应的实体匹配，$\mathrm{P}_{path}^{(i)}$ 是日志 l_i 对应的故障传播路径概率。

通过这种对齐机制，模型能够有效地将知识图谱中的结构化信息整合到文本表示中，使得最终的文本嵌入不仅包含语言学特征，还融合了故障传播模式、实体关联关系等图结构知识，从而提升了模型对复杂日志场景的理解能力。

4.3. 下游任务进行微调

在预训练完成后，我们针对不同的日志相关下游任务对预训练模型进行微调。与预训练阶段类似，使用[CLS]标记的表示向量表示整个输入。在下游任务中，我们实验了两类输入任务：

单日志任务(如故障现象识别)，输入为单一日志序列；

日志对任务(如日志与可能原因排序)，输入为成对日志数据。

对于单日志任务，我们将对应编码后的向量直接输入分类器进行预测。对于日志对任务，参照 Sentence-BERT [18]的方法，分别独立编码两个输入，我们独立编码输入以生成表示向量 l，d 随后将向量 l，d 及其按元素差异 $\{l-d\}$ 进行拼接，并将拼接结果输入 $\left[l;d;\{l-d\}\right]$ 分类器。

针对每项任务，我们只需将输入传递至模型中，随后端到端微调所有参数，最终微调后的模型即可用于特定任务的推断。具体实现细节将在下一节详述。

5. 实验

5.1. 实验设置

在预训练中我们使用了 110M 参数的 bert-base-uncased 的模型。采用 Adam 优化器[21]对模型参数进行优化，其学习率为 5e−5，权值衰减为 0.01。批大小设置为 8，Epoch 设置为 50。在微调中，我们在下游任务中采用交叉熵损失作为损失函数，并在单日志任务和日志对任务上分别将 Epoch 设置为 20 和 10。

所有实验在一台配备 Intel Xeon Silver 4310 CPU @ 2.10GHz、NVIDIA RTX 3090 GPU 及 256GB 内存的服务器上完成，操作系统为 Windows Server 2022 Standard。实验环境基于 PyTorch 1.10.0 深度学习框架，CUDA 版本为 11.1。

5.1.1. 数据集

在本文中，日志分析方案研究面向网络设备。在预训练阶段，我们基于文献[13]，从 Cisco 和 Huawei 两家供应商的公开文档中提取了 18,481 条网络设备日志模板，涵盖交换机、路由器和 WLAN 三类设备。

在下游任务进行微调阶段通过四种不同的日志理解下游任务。根据输入类型，这些下游任务可以分为两类：单日志任务(输入为单个日志)和日志对任务(输入为日志对或日志－自然语言对)。对于每个数据集，我们按照 6:2:2 的比例划分训练集、验证集和测试集。在表 1 中，我们提供了其数据集不同任务的统计数据。

Table 1. Statistics on upstream and downstream task datasets for network devices (training/validation/testing scale)
表 1. 网络设备上下游任务数据集的统计(训练/验证/测试规模)

TASK		Switches	Routers	Security*
MC	Cisco	13,495/4498/4498	7265/2422/2421	--
	Huawei	3439/1146/1146	2539/846/845	--
	H3C*	1241/413/413	1336/445/444	--
FPI	Huawei	362/120/120	--	--
LDSM	Cisco	49,954/16,651/16,651	26,975/8992/8991	1894/631/631
	Huawei	7702/2567/2567	5977/1992/1991	4485/1495/1494
LPCR	H3C*	2606/868/868	2837/946/945	2223/741/740

*表示未参与预训练的日志。

5.1.2. 下游任务

我们使用训练集微调模型，以获得验证集上的最佳结果，最后在测试集上评估并报告结果。

1) 模块分类(MC)

MC 是一种单日志类型的多类分类任务，旨在识别日志所属的模块。该任务的输入是一个模块名称被屏蔽的日志，输出是相应的模块名称。

我们使用收集的日志模板作为原始数据，日志中的模块名称作为真实标签，然后将模块名称替换为[MASK]以避免标签泄露，如输入：[MASK]-3-DUPLICATE_IFINDEX:%s has %d duplicate ifIndices.，希望可以对掩盖部分识别并分类为 SNMP。

2) 故障现象识别(FPI)

FPI 是一个日志单任务，用于识别日志所属的故障类别。该任务基于真实世界的数据。是一个多标签分类任务，因为一个日志可能出现在多个故障类别。输入是日志，输出是一个或多个故障现象。

3) 日志和描述语义匹配(LDSM)

日志和描述语义匹配是一个日志对任务，旨在确定给定日志的语义是否与相应的自然语言描述对齐，输入为日志和描述对，输出为真或假。

4) 日志和可能原因排名(LPCR)

日志和可能原因排序是一个日志对排序任务，旨在从给定日志的可能原因列表中找到最可能的答案，输入为日志作为查询和答案候选集，输出为排序结果。

5.1.3. 基线

我们将日志理解的基准分为两类：传统深度学习方法和预训练语言模型。

CNN [19]. 采用 Word2vec 模型[20]来学习日志的表示向量。然后将这些向量输入卷积神经网络(CNN)以支持下游任务。

BiLSTM [10]. BiLSTM 是日志分析中的一种神经网络架构，它将每个日志消息通过单词嵌入模型转换为向量，然后将向量输入到基于注意力的 BiLSTM 模型。

BERT [6]. BERT 模型作为一种预训练的网络日志模型，具有良好的语义表示能力，能够很好地表示日志。

Knowlog [13]. Knowlog 模型是一种利用从官方文档中收集的日志描述，对缩写进行特殊处理以进行日志理解的预训练模型。

LUK [14]. LUK 模型是一种日志理解的预训练模型，利用 LLM 中获取专家知识，弥补了领域内专业知识不足的问题，能够很好的利用外部知识对日志进行理解。

5.1.4. 评价指标

模块分类(MC)任务作为一个不平衡的多类分类任务，并考虑到不同类别的重要性，我们使用准确率和加权 F1 作为评估指标。

故障现象识别(FPI)任务涵盖了 43 个故障类别，这些日志都是由专家标注的[13]。与多类分类任务不同，我们使用所有样本的平均准确度[22]作为评估指标，每个样本的准确度是正确预测的标签的数量占全部标签的比例。

日志和描述语义匹配(LDSM)作为语义匹配的二元分类任务，肯定和否定的情况都需要注意，我们使用准确率和加权 F1 得分作为评价指标。

日志和可能原因排名(LPCR)任务中我们使用 Precision@k 和平均倒数秩(MRR)作为评价指标。Precision@1 表示第一位的准确率。

5.2. 实验结果

我们在上述四个任务上进行了实验。表 2~4 展示了 MC、LDSM 和 LPCR 的实验结果，FPI 的结果如图 2 所示。

本文提出方法在所有设置下的模块分类任务中均取得了优异的性能。如表 2 所示，对于思科设备，在交换机和路由器上的准确率/加权 F1 得分分别为 64.70/64.48 和 65.18/63.82，超越了此前最佳模型 Knowlog (64.07/63.75 和 64.85/64.08)。在华为设备上，在交换机和路由器上的准确率分别为 86.18/85.22 和 85.56/84.41，准确率比 Knowlog 最高提升了 1.29%。这表明结构化掩码策略通过重点关注结构化字段的关键 token，显著提升了模型对系统组件的识别能力。

Table 2. Result on MC
表 2. 模块分类结果

Methods	MC (Accuracy/Weighted F1)					
	Cisco		Huawei		H3C*	
	Switches	Routers	Switches	Routers	Switches	Routers
CNN	56.89/56.85	57.46/54.92	74.52/73.95	72.78/72.23	69.49/67.55	70.72/69.71
BiLSTM	55.74/55.63	57.17/56.76	76.52/75.49	73.96/73.30	70.21/68.45	71.40/69.93
BERT	62.67/61.38	62.72/62.60	82.37/81.20	81.18/79.20	81.11/79.78	77.93/76.05
KnowLog	64.07/63.75	64.85/64.08	85.43/84.78	84.38/83.51	81.36/79.31	78.38/76.77
LUK	63.78/63.57	64.44/63.42	84.21/83.53	83.20/82.09	82.08/80.25	79.05/77.36
Ours	**64.70/64.48**	**65.18/63.82**	**86.18/85.22**	**85.67/84.41**	**83.05/81.63**	**79.95/78.15**

在表 3 中的日志和描述语义匹配的结果显示，本文提出的方法在此任务上取得了更佳的性能，在 Cisco 数据集的交换机和路由器上分别达到 95.42%/95.42%和 93.29%/93.29%的准确率/加权 F1 值。此外在未见数据集 H3C 上表现的更为优异，相比此前最好的 LUK 模型准确率最高提升了 1.59%，展现出所提出方法具备更好的日志与其对应的自然语言描述匹配能力，GNN-文本对齐机制成功将结构化的系统知识与非结构化的文本描述进行语义对齐，使模型能够更好地理解日志的实际含义。

Table 3. Result on LDSM
表 3. 日志和描述语义匹配的结果

Methods	LDSM (Accuracy/Weighted F1)					
	Cisco		Huawei		H3C*	
	Switches	Routers	Switches	Routers	Switches	Routers
CNN	84.04/84.04	80.99/80.99	86.05/86.05	82.37/82.30	83.29/83.19	83.60/83.59
BiLSTM	89.45/89.44	85.42/85.41	87.85/87.85	84.43/84.40	80.88/80.83	83.81/83.80
BERT	93.06/93.06	91.46/91.46	93.18/93.18	90.06/90.05	87.44/87.41	88.25/88.25
KnowLog	94.65/94.65	91.80/91.80	95.13/95.13	93.07/93.06	89.63/89.62	90.48/90.48
LUK	95.23/95.23	92.71/92.71	96.18/96.18	95.43/95.43	92.51/92.51	93.54/93.54
Ours	**95.42/95.42**	**93.29/93.29**	**96.69/96.69**	**96.23/96.23**	**94.12/94.12**	**94.60/94.60**

表 4 的 LPCR 任务结果展现了所提出方法的卓越的排序能力。在 huawei 的各个网络设备上都达到了最优，尤其是在华为交换机上，Precision@1/Precision@3/MRR 分别达到了 89.43/97.03/93.53，相比 LUK 分别提升了 3.59%、2.56%和 2.43%。排序性能的显著提升，特别是在 Precision@1 上，表明本文方法能够更准确地识别给定日志条目中最相关的原因。此功能对于在网络维护中高效地进行根本原因分析至关重要。

此外在表 2~4 的 H3C*与 Security*这些预训练未见数据集中，本文所提出方法也具有更好的表现。在 MC 任务中，我们的方法在 H3C 交换机和路由器上分别达到了 83.05/81.63 和 79.95/78.15，优于所有基线方法。在表 3 的日志和描述语义匹配任务准确率相比基线中最优的 LUK 模型分别提高了 1.61%与 1.06%。在 LPCR 任务上也取得了优异的表现(89. 94/96.11/92.96)。通过这些实验证明了该模型对未知网络设备供应商的强大泛化能力，日志实体及其关系的结构化表示提供了可迁移的知识，从而弥合了供应商特定术语之间的差距。这对于现实世界的部署场景至关重要。

Table 4. Result on LPCR
表 4. 日志和可能原因排名结果

Methods	LPCR (Precision@1/3/MRR)		
	Huawei		
	Switches	Routers	Security*
CNN	54.30/77.26/67.99	53.45/75.77/67.35	56.05/79.07/69.95
BiLSTM	59.27/78.04/71.22	51.45/69.56/63.76	55.65/79.21/69.80
BERT	76.18/91.54/84.70	72.57/91.59/82.61	67.89/89.73/79.55
KnowLog	80.18/91.84/86.83	78.87/93.08/86.45	82.78/ 93.21/88.59
LUK	86.29/95.18/91.18	85.39/95.36/90.99	85.30/94.30/90.10
Ours	**89.88/97.74/93.61**	**88.25/97.24/92.80**	**88.94/96.11/92.96**

在真实场景的故障现象识别(FPI)任务中，如图 2 所示我们的方法在 FPI 任务优于所有基线，达到了 96.65%，相比基础 BERT 模型提升了 4.15 个百分点。这一显著提升验证了知识图谱增强策略在复杂故障现象识别任务中的有效性。

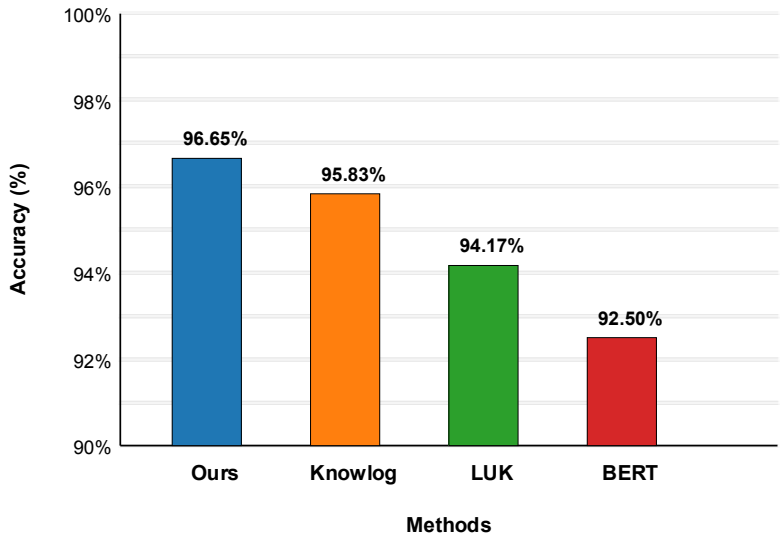

Figure 2. Results on FPI task
图 2. 故障现象识别任务结果

5.3. 消融实验

为了验证所提出方法的有效性，我们对多类分类任务和语义匹配任务中的两个典型任务 MC 和 LDSM 进行了消融实验。结果如表 5 所示，我们注意到：(1) 总体而言，所提出方法在完整模块的情况下实现了最佳性能。任何模块的缺失都会导致性能下降，这证明所设计的预训练任务都有积极的贡献；(2) 在没有结构化掩码预测任务(SMP)或 GNN-文本对齐任务(GEA)的情况下，模型的性能下降，这表明利用日志与日志本身的结构化知识进行预训练可以提高日志理解能力。具体来说，在没有 SMP 任务的情况下，模型在 MC 任务上下降得更明显，这意味着结构化掩码预测任务在预训练阶段帮助模型建立了日志文本的结构感知能力对识别被掩盖的模块名称至关重要，因为不同模块通常具有特定的结构模式和组件分布特征。结构预测过程强化了模型识别和区分日志中的不同实体类型(设备、错误码、错误级别和

Message)，使其能够更准确地推断出缺失的模块信息。在缺少 GEA 任务的情况下，模型在 LDSM 任务上的下降更明显，说明模型能够有效地将知识图谱中的结构化信息整合到文本表示中。LDSM 任务本质上需要模型理解日志的结构化信息与其语义描述之间的对应关系，而 GEA 任务正是通过多维度的对齐机制来建立这种对应关系，能更精确进行理解。

Table 5. Result on ablation studies
表 5. 消融实验结果数据

Methods	MC		LDSM	
	Huawei		Huawei	
	Switches	Routers	Switches	Routers
BERT	82.37/81.20	81.18/79.20	93.18/93.18	90.06/90.05
Ours	**86.18/85.22**	**85.67/84.41**	**96.69/96.69**	**96.23/96.23**
--w/o SMP	84.77/83.94	83.54/82.37	95.23/95.23	94.88/94.88
--w/o GEA	85.36/84.68	84.02/83.14	94.63/94.63	93.42/93.42

6. 结论

在本文中，本文提出了一种结构知识增强的网路设备日志理解的预训练语言模型，它提高了日志理解任务的最新性能。我们提出了在不使用日志描述知识的基础上，有效地利用知识图谱以及日志本身的结构化知识来增强模型，使日志的表示更加通用。经过预训练后，本文方法在四个不同的下游任务上进行了微调。与其他预训练模型相比，本文方法实现了最先进的性能，这证明了日志本身的机构化内容对于提高日志理解的有效性。消融分析证明了这些预培训任务对于利用知识理解日志的有效性。未来，我们将探索将日志本身所包含的机构化知识与外部知识高效融合，进一步全面提升对日志的理解能力。

参考文献

[1] Jiang, Z.X., Li, T., Zhang, Z.G., Ge, J.G., You, J.L. and Li, L.X. (2021) A Survey on Log Research of Aiops: Methods and Trends. *Mobile Networks and Applications*, **26**, 2353-2364. https://doi.org/10.1007/s11036-021-01832-3

[2] Zhang, X., Xu, Y., Qin, S., He, S., Qiao, B., Li, Z., *et al.* (2021) Onion: Identifying Incident-Indicating Logs for Cloud Systems. *Proceedings of the* 29*th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Athens, 23-28 August 2021, 1253-1263. https://doi.org/10.1145/3468264.3473919

[3] Du, M., Li, F., Zheng, G. and Srikumar, V. (2017) DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. *Proceedings of the* 2017 *ACM SIGSAC Conference on Computer and Communications Security*, Dallas, 30 October-3 November 2017, 1285-1298. https://doi.org/10.1145/3133956.3134015

[4] Wit, E. and McClure, J. (2004) Statistics for Microarrays: Design, Analysis, and Inference. 5th Edition, Wiley. https://doi.org/10.1002/0470011084

[5] Zhang, C., Peng, X., Sha, C., *et al*. (2022) Deeptralog: Trace-Log Combined Microservice Anomaly Detection through Graph-Based Deep Learning. *ICSE*'22: *Proceedings of the* 44*th International Conference on Software Engineering*, 623-634. https://doi.org/10.1145/3510003.3510180

[6] Devlin, J., Chang, M.W., Lee, K., *et al.* (2019) Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the* 2019 *Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, *Volume* 1 (*Long and Short Papers*), Minneapolis, 2-7 June 2019, 4171-4186.

[7] Li, X., Chen, P., Jing, L., He, Z. and Yu, G. (2020) SwissLog: Robust and Unified Deep Learning Based Log Anomaly Detection for Diverse Faults. 2020 *IEEE* 31*st International Symposium on Software Reliability Engineering* (*ISSRE*), Coimbra, 12-15 October 2020, 92-103. https://doi.org/10.1109/issre5003.2020.00018

[8] Gholamian, S. and Ward, P.A.S. (2021) On the Naturalness and Localness of Software Logs. 2021 *IEEE*/*ACM* 18*th International Conference on Mining Software Repositories* (*MSR*), Madrid, 17-19 May 2021, 155-166.

https://doi.org/10.1109/msr52588.2021.00028

[9] Han, X. and Yuan, S. (2021) Unsupervised Cross-System Log Anomaly Detection via Domain Adaptation. *Proceedings of the* 30*th ACM International Conference on Information & Knowledge Management*, 1-5 November 2021, 3068-3072. https://doi.org/10.1145/3459637.3482209

[10] Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y., *et al.* (2019) Robust Log-Based Anomaly Detection on Unstable Log Data. *Proceedings of the* 2019 27*th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Tallinn, 26-30 August 2019, 807-817. https://doi.org/10.1145/3338906.3338931

[11] Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., *et al.* (2019) LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, 10-16 August 2019, 4739-4745. https://doi.org/10.24963/ijcai.2019/658

[12] Guo, H., Yuan, S. and Wu, X. (2021) LogBERT: Log Anomaly Detection via Bert. 2021 *International Joint Conference on Neural Networks* (*IJCNN*), Shenzhen, 18-22 July 2021, 1-8. https://doi.org/10.1109/ijcnn52387.2021.9534113

[13] Ma, L., Yang, W., Xu, B., Jiang, S., Fei, B., Liang, J., *et al.* (2024) KnowLog: Knowledge Enhanced Pre-Trained Language Model for Log Understanding. *Proceedings of the IEEE/ACM* 46*th International Conference on Software Engineering*, Lisbon, 14-20 April 2024, 1-13. https://doi.org/10.1145/3597503.3623304

[14] Ma, L., Yang, W., Jiang, S., Fei, B., Zhou, M., Li, S., *et al.* (2025) LUK: Empowering Log Understanding with Expert Knowledge from Large Language Models. *IEEE Transactions on Software Engineering*. https://doi.org/10.1109/tse.2025.3594046

[15] Peters, M.E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., *et al.* (2019) Knowledge Enhanced Contextual Word Representations. *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9*th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, 3-7 November 2019, 43-54. https://doi.org/10.18653/v1/d19-1005

[16] Sui, Y., Zhang, Y., Sun, J., Xu, T., Zhang, S., Li, Z., *et al.* (2023) LogKG: Log Failure Diagnosis through Knowledge Graph. *IEEE Transactions on Services Computing*, **16**, 3493-3507. https://doi.org/10.1109/tsc.2023.3293890

[17] Liao, L., Zhu, K., Luo, J. and Cai, J. (2023) LogBASA: Log Anomaly Detection Based on System Behavior Analysis and Global Semantic Awareness. *International Journal of Intelligent Systems*, **2023**, Article ID: 3777826. https://doi.org/10.1155/2023/3777826

[18] Reimers, N. and Gurevych, I. (2019) Sentence-BERT: Sentence Embeddings Using Siamese Bert-Networks. *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9*th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, 3-7 November 2019, 3982-3992. https://doi.org/10.18653/v1/d19-1410

[19] Lu, S., Wei, X., Li, Y. and Wang, L. (2018) Detecting Anomaly in Big Data System Logs Using Convolutional Neural Network. 2018 *IEEE* 16*th Intl Conf on Dependable*, *Autonomic and Secure Computing*, 16*th Intl Conf on Pervasive Intelligence and Computing*, 4*th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress* (*DASC/PiCom/DataCom/CyberSciTech*), Athens, 12-15 August 2018, 151-158. https://doi.org/10.1109/dasc/picom/datacom/cyberscitec.2018.00037

[20] Pennington, J., Socher, R. and Manning, C. (2014) Glove: Global Vectors for Word Representation. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, 25-29 October 2014, 1532-1543. https://doi.org/10.3115/v1/d14-1162

[21] Kingma, D.P. (2014) Adam: A Method for Stochastic Optimization. arXiv: 1412.6980.

[22] Sorower, M.S. (2010) A Literature Survey on Algorithms for Multi-Label Learning. Oregon State University.