

基于BERT模型对操作系统软件包分层分类的研究

张欢欢, 王耀华

(统信软件技术有限公司 产研平台, 北京 100020)

摘要: 随着操作系统的日趋成熟和软件包种类的日益丰富, 操作系统软件包的管理和系统构建要求也越来越高。传统的软件包管理和构建方式通常依赖于人工主观判断, 存在效率低下、难成体系等问题, 同时对相关人员的技术水平要求较高。因此, 研究一种高效、准确的操作系统软件包分层分类方法具有重要意义。提出一种基于BERT模型的对操作系统软件包进行分层分类的方法。BERT模型具有双向Transformer编码器, 更适用于语言理解, 能够对软件包进行自动、准确的分层分类。该方法为操作系统软件包的管理和构建提供支撑, 使其更加高效和便捷。

关键词: 神经网络; 操作系统; 软件包; 软件包依赖关系

DOI: 10.11907/rjdk.241614

中图分类号: TP311.52

文献标识码: A

开放科学(资源服务)标识码(OSID):

文章编号: 1672-7800(2025)009-0070-05



Research on Hierarchical Classification of Operating System Software Packages Based on BERT

ZHANG Huanhuan, WANG Yaohua

(Department of Industry-Research Platform, Uniontech Software Technology Co., Ltd., Beijing 100020, China)

Abstract: As operating systems mature and the variety of software packages increases, higher requirements are imposed on the management and system construction of software packages for operating systems. Traditional methods of managing and building software packages often rely on manual subjective judgments, which are inefficient, difficult to systematize, and require a high level of technical expertise from relevant personnel. Therefore, it is particularly important to explore an efficient and accurate method for hierarchical classification of operating system software packages. This paper proposes a method for hierarchical classification of operating system software packages based on the BERT model, which has a bidirectional Transformer encoder that is more suitable for language understanding. This method enables automatic and accurate hierarchical classification of software packages, providing support for the management and construction of operating system software packages to make them more efficient and convenient.

Key Words: neural network; operating system; software package; software package dependencies

0 引言

随着计算机技术飞速进步和软件应用全面普及, 操作系统软件包数量急剧增多, 种类持续扩容, 同时, 软件包之间复杂的依赖关系使得Linux操作系统发行版的构建与维护工作面临巨大挑战。在此背景下, 神经网络技术的快速

发展为解决上述问题提供了创新性的思路。本文提出一种针对深度操作系统发行版的方法: 首先, 采集软件源中的相关信息; 其次, 利用智能神经网络模型对软件包进行精准分类, 并将其合理分层至特定体系结构中; 最后, 将处理结果存储至数据库。该方法旨在为操作系统构建提供理论支持, 辅助工程师全面理解软件包在系统架构中的定位, 从而更高效、便捷地选择和集成所需的软件包。

收稿日期: 2024-07-19

扫描二维码阅读全文:

基金项目: 北京市科技计划项目(Z231100005923009)

作者简介: 张欢欢(1996-), 女, 统信软件技术有限公司高级系统研发工程师, 研究方向为计算机操作系统、神经网络; 王耀华(1991-), 男, 统信软件技术有限公司产研平台副总经理, 研究方向为计算机操作系统。本文通讯作者: 张欢欢。



1 相关工作

软件包管理系统是操作系统中的重要组成部分,随着开源软件运动的开展,开源软件包的数量正在以指数级的速度增长,如何正确而有效地对软件包进行管理已经成为当前操作系统研究的一个重要课题^[1]。以往工作对于软件包管理进行了诸多探索。例如,文献[2]提到分层操作系统可以减少模块之间过于紧密的连接,并确保流程调用的有序性。但研究表明,目前针对特定操作系统层次结构的专门性研究工作仍然较为有限。文献[3]将软件包建模为节点,将它们之间的依赖关系建模为边。针对 Gentoo Linux 操作系统的包依赖网络提出了一种新的网络演化模型,但没有对包进行任何分类行为。文献[4]首次基于软件包间的依赖关系,对软件包的出入度、优先级和分类等特征进行了统计分析,进而从依赖关系的角度总结出操作系统版本构建的主要原则。该研究为提升版本自动化定制以及软件发行演化提供了重要指导和参考。

上述研究虽在不同程度上利用依赖网络来辅助操作系统中的包维护,但仍存在明显局限,大多只基于图论相关指标进行整体分析,无法针对每个包进行客观指导。

针对上述局限性,本文提出以下改进方案:①对系统中的每个包进行明确的层次推理,即为每个包提供客观的参考信息;②构建完整的层次化架构,该结构不仅通过纵向依赖关系实现类别划分,还通过横向包应用类型明确各类别间的关联关系。

神经网络是一种模拟人脑神经元连接和工作方式的计算模型,由多个节点(神经元)和连接这些节点的权重组成。每个节点接收输入数据,进行计算处理后输出结果。神经网络通过训练来学习模式和规律,以便在新的输入数据上进行预测和决策。神经网络的基本原理是通过模拟神经元之间的连接和信息传递,来实现复杂的计算任务。与常用于处理具有复杂关系结构的数据(如社交网络分析、推荐系统、生物信息学等领域)的图形分类不同,本文使用到的是文本分类方法。在神经网络领域,针对文本进行训练预测的模型架构,包括但不限于:CNN、RNN^[5]、长短期记忆网络(LSTM)^[6]、Seq2seq 模型^[7]以及 Transformer^[8]等,均具有独特的技术特点。通过借鉴研究构建 Linux 发行版的软件包进行分类^[9],本文采用 BERT^[10]模型开展软件包分层分类研究探索。

1.1 分层介绍

分层是指根据软件包对操作系统分布的重要性,将其垂直划分为不同层级的方法。本文依据 Linux 软件包的内容属性及其依赖/调用关系,将其划分为 4 个层级(L1-L4),具体包括:内核层(L1)、核心层(L2)、系统层(L3)和应用层(L4)。

1.2 分类介绍

根据库中 Linux 包的内容属性,将其分为工具类、库类、服务类和其他类。该分类方式是依据软件包的主要文件在系统中的存在方式而定义的,因此将这种分类方式称为形态分类。

2 应用实验

2.1 材料

本文从镜像站点中获取了 Linux 操作系统发行版深度操作系统的相关数据库。基于该数据库中的 rpm 源包,提取了包括包的名称、依赖以及描述等相关信息。从数据库中共提取了 6 759 个二进制软件包,根据行业内专家的经验,将操作系统软件包进行手动注释。软件包的分类情况如表 1 所示,软件的分层情况如表 2 所示。

Table 1 Classification of software packages

表 1 软件包分类情况

类别	rpm 软件包的数量
库	3 343
服务	694
工具	2 044
其他	383

Table 2 Hierarchical structure of software packages

表 2 软件包分层情况

层级	rpm 软件包的数量
L1	83
L2	360
L3	1 961
L4	4 355

2.2 方法与结果

实验环境配置如下:CPU 为 Intel Xeon 4core; GPU 为 Nvidia-RTX 3060Ti-8G; Intel Corporation Device 9b43; Python 版本为 V3.7; Tensorflow 版本为 V2.9.1; Keras 版本为 V. 2.9.0; nltk 版本为 V3.7; transformer 版本为 V4.21.1。软件包 E-R(Entity-Relationship Diagram)图信息如图 1 所示。

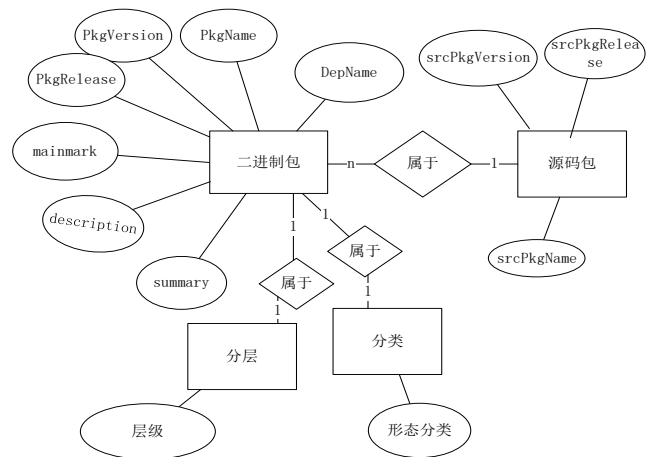


Fig. 1 E-R diagram for software package

图 1 软件包的 E-R 图

本文探讨了一种通过使用BERT模型来实现操作系统软件包分层分类的方法。具体实现路径如图2所示。

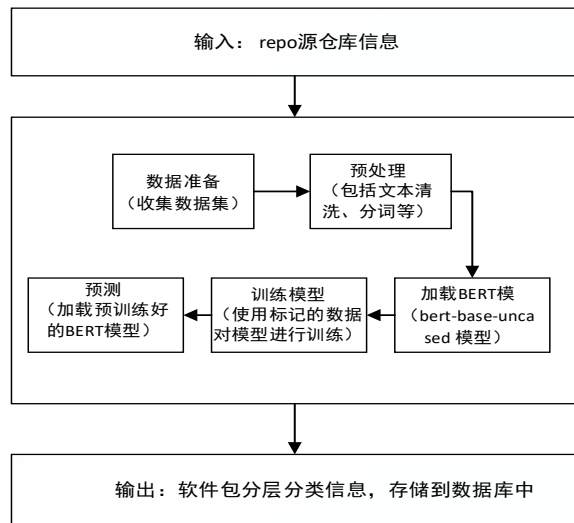


Fig. 2 Flowchart of algorithm implementation

图2 算法实现的流程图

2.2.1 基于BERT模型实现软件包分类

基于BERT模型实现软件包分类的实现路径如下：

(1)文本预处理。本文的repo信息来源于uos1050源码库,数据库存储格式为SQLite数据库。该数据库包含2个主要部分:主数据库和文件列表数据库。本文的代码库信息展示如图3所示。软件包的描述和概要信息可通过主数据库packages表中的description和summary字段获取,具体细节如图4所示。

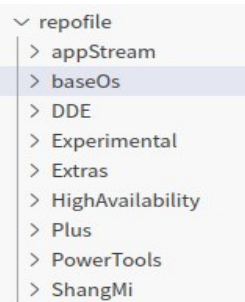


Fig. 3 Repo source information

图3 repo源信息

	name	summary	description
	Search column...	Search column...	Search column...
1	389-ds-base	389 Directory Serve...	389 Directory Serve...
2	389-ds-base	389 Directory Serve...	389 Directory Serve...
3	389-ds-base-devel	Development librar...	Development Librar...
4	389-ds-base-devel	Development librar...	Development Librar...
5	389-ds-base-legacy...	Legacy utilities for ...	Legacy (and deprec...
6	389-ds-base-legacy...	Legacy utilities for ...	Legacy (and deprec...
7	389-ds-base-libs	Core libraries for 38...	Core libraries for th...
8	389-ds-base-libs	Core libraries for 38...	Core libraries for th...
9	389-ds-base-snmp	SNMP Agent for 389...	SNMP Agent for the ...
10	389-ds-base-snmp	SNMP Agent for 389...	SNMP Agent for the ...

Fig. 4 The summary and description fields in the packages table

图4 packages表summary字段和description字段

在进行分类使用时,将软件包的概要信息和描述信息合并为一个文本字符串,设定字符串的最大长度为192个字符。如果文本长度不足192个字符,则使用[PAD]标签填充至192个字符;如果文本长度超过192个字符,则将超出部分截断。随后,将这个文本字符串转换为BERT模型能够处理的格式。具体转换步骤如下:

1.FOR 每个软件包DO

2.使用nltk模块清洗软件包的描述信息跟概述信息(去除标点符号、分词token及停用词)

3.为句子添加[cls]标签和[sep]标签

4.转换为token ids

5.创建attention mask

6.创建token type ids

7.截断或填充以匹配max_seq_length(192)

8.END

(2)微调模型。本文聚焦于文本分析领域,通过集成先进的深度学习技术与自然语言处理技术,实现对操作系统软件包的精准分类。为提升分类任务的性能,本文在BERT模型的基础上进行了扩展,在BERT模型的顶层添加了softmax全连接层。这一设计将BERT的丰富语义表示转化为具体的分层分类预测输出,并针对操作系统软件包的复杂特性进行了精准的建模。分类涉及的超参数信息如表3所示。

Table 3 Hyperparameter information for classification

表3 分类的超参数信息

参数名	数值
learning_rate	1e-5
epochs	20
maxlen	192
batch_size	8
drouout	0.8

(3)模型训练与评估。首先,对提取的6759个二进制软件包数据进行精心的划分,采用十则交叉验证法将其分配为训练集、验证集和测试集,三者比例设定为7:2:1;其次,使用model.fit对分类模型进行深入训练,让模型充分学习并掌握数据中的特征与模式,以期在后续的任务中展现出良好的分类能力;再次,为全面量化模型性能,本文选用精确率和召回率这两个指标对其进行评估。

(4)结果显示。分类的具体训练情况如图5所示。随着训练轮次的递增,模型的准确率呈现稳步上升趋势,最终达到95.2%的最高值。这一显著的性能提升验证了训练过程的有效性,同时证明了模型学习能力的增强。BERT模型的实验结果如表4所示,其中最后一列显示了测试中的支持。数据分析表明,采用BERT模型作为分类器的核心时,其分类性能达到了极高的水平。具体而言,分类任务的整体精确率高达0.91,这证明了BERT模型在分类任务中的卓越表现。且进一步细分到各个类别,如服务类、工具类、库类和其他类,模型均展现出了出色的预测

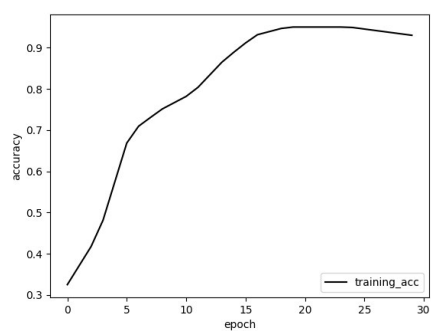


Fig. 5 Training situation of classification
图 5 分类的训练情况

Table 4 Classification prediction results

表 4 分类预测结果

衡量指标	BERT			support
	precious	recall	F1-score	
其他类	0.95	0.97	0.97	91
工具类	0.86	0.9	0.81	1 432
库类	0.93	0.93	0.9	1 288
服务类	0.83	0.7	0.80	385
accuracy			0.91	3 196
macro avg	0.89	0.90	0.90	3 196
weighted avg	0.90	0.91	0.91	3 196

能力,预测准确率分别达到了 83%、86%、93% 和 95%。该结果表明,模型不仅能够准确捕捉各类别的特征,也反映了其广泛的适用性和强大的泛化能力。

2.2.2 基于 BERT 模型实现软件包分层

本文提出一种基于 BERT 模型的操作系统软件包分层的方法。该方法在实现流程上与传统分类任务基本一致,但在文本预处理阶段需要特殊处理。由于文本处理长度限制,需要额外添加软件包入度、出度的描述信息与概述信息。软件包入度与出度处理后的结果如图 6 所示;BERT 模型在分层应用的超参数信息如表 5 所示。

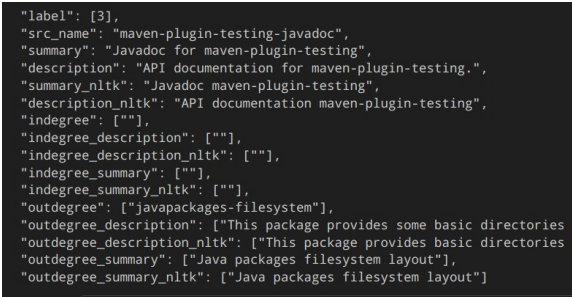


Fig. 6 Hierarchical input information
图 6 分层的输入信息

Table 5 Hierarchical hyperparameter information

表 5 分层的超参数信息

参数名	数值
learning_rate	1e-5
epochs	30
maxlen	192
batch_size	8
drouput	0.8

表 6 展示了 BERT 模型的实验结果,其中最后一列显示了测试中的软件包。数据分析表明,当 BERT 模型采用全连通网络作为分层处理机制时,其在不同层级(L1、L2、L3)及整体层预测上都有较好的表现。具体而言,随着层级从 L1 到 L3 的提升,模型精确率呈现显著增长趋势,分别达到 68.7%、75.3% 和 86.6%。其中,L3 层的精确率与整体层预测的精确率相近,均为 85.2%,表明随着层级的深入,模型对数据的分类能力逐渐增强。同时,召回率同样呈现显著上升趋势,从 L1 层的 69.3% 提升至 L3 层的 85.9%,并在整体层预测中达到 86.7%,充分证明了模型在识别相关类别上的高效性。实验结果不仅验证了该模型架构在复杂数据处理中的优势,而且为需要精细分层处理的应用场景提供了可靠的技术支持。

Table 6 Hierarchical structure of software packages

表 6 软件包分层情况

	L1	L2	L3	L4	ALL
软件包数量	83	360	1 961	4 355	6 759
参与分层的包数量	83	360	1 961	4 355	6 759
分层正确数量	57	271	1 699	3 710	5 737
分层错误数量	26	89	262	645	1 022
PPV	0.687	0.753	0.866	0.852	-
TPR	0.693	0.768	0.859	0.867	-
ACC	-	-	-	-	0.849

2.2.3 结果分析与应用

软件包经过分层分类的预测结果如表 7 所示。通过软件包的分层分类预测,能够精确描绘出 repo 源中所有软件包的架构与布局,从而清晰地揭示软件包之间的关联。这种结构化的关系不仅优化了软件包的组织方式,还显著加速了系统的构建流程和维护工作。在实际测试中,相较于传统构建过程中随意挑选软件包的方式,遵循预测所得的软件包层级顺序进行构建,显著降低了构建过程中的错误率,并实现了高达 80% 的软件包编译通过率,充分验证了其高效性与可靠性。

Table 7 Software package prediction results

表 7 软件包预测结果

软件包名	层级	类别
hunspell-ia	2	工具
elfutils-default-yama-scope	2	库
texlive-textpos	2	工具
google-noto-sans-coptic-fonts	2	其它
python3-networkx	3	库
freeradius-mysql	2	服务
java-1.8.0-openjdk-devel-fastdebug	2	库

3 结语

本文系统地探究了神经网络在操作系统软件包分层分类中的应用效能,重点评估了 BERT 模型在软件包分层和分类两个维度的性能表现。实验结果表明,基于 BERT 模型对操作系统软件包进行分层分类的方法取得了显著

成效,预测准确率均超过 80%。这一性能表现验证了通过应用神经网络技术,可以初步实现对操作系统软件包的自动分类和分层,从而提高软件管理的效率和准确性。未来,随着技术的不断发展和应用需求的日益增长,神经网络在操作系统中的应用将更加广泛和深入。目前,本文仅在单一特定的发行版上进行了验证,未来将尝试探索更多的发行版,以开发具普适性的包层次类别结构方法。

参考文献:

- [1] MA Y. Research on package management for secure operating systems [D]. Beijing: Institute of Software Chinese Academy of Sciences, 2007.
马勇. 安全操作系统软件包管理研究[D]. 北京: 中国科学院软件研究所, 2007.
- [2] LU S N. Operating system tutorials [M]. Beijing: Publishing House of Electronics Industry, 2014.
陆松年. 操作系统教程[M]. 北京: 电子工业出版社, 2014.
- [3] ZHENG X L, ZENG D, LI H Q, et al. Analyzing open-source software systems as complex networks[J]. Physica A: Statistical Mechanics and its Applications, 2008, 387(24): 6190–6200.
- [4] MA J, ZHOU K, REN Y, et al. Package dependency analysis for operating system distribution building[J]. Computer Engineering and Science, 2021, 43(11): 1926–1933.
马俊, 周凯, 任怡, 等. 面向操作系统版本构建的软件包依赖关系分析[J]. 计算机工程与科学, 2021, 43(11): 1926–1933.
- [5] LI Z Z. Research on text classification method based on CNN and RNN [D]. Jingzhou: Yangtze University, 2021.
栗征征. 基于 CNN 和 RNN 的文本分类方法研究[D]. 荆州: 长江大学, 2021.
- [6] REN M, GAN G. Sentiment analysis of text based on bi-directional long short-term memory model [J]. Computer Engineering and Design, 2018, 39(7): 2064–2068.
任勉, 甘刚. 基于双向 LSTM 模型的文本情感分类[J]. 计算机工程与设计, 2018, 39(7): 2064–2068.
- [7] CHEN Q, HAN L, WANG S G, et al. Multi-label classification of options based on seq2seq model of hybrid attention[J]. Computer Engineering and Applications, 2023, 59(4): 104–111.
陈千, 韩林, 王素格, 等. 基于混合注意力 Seq2seq 模型的选项多标签分类[J]. 计算机工程与应用, 2023, 59(4): 104–111.
- [8] EKVALL M, TRUONG P, GABRIEL W, et al. Prosit transformer: a transformer for prediction of MS2 spectrum intensities[J]. Journal of Proteome Research, 2022, 21(5): 1359–1364.
- [9] QI J, LIAN Y, LEI Z, et al. Classifying packages for building Linux distributions [C]//Torino: IEEE 47th Annual Computers Software and Applications Conference (COMPSAC), 2023.
- [10] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [DB/OL]. <https://arxiv.org/abs/1810.04805>.
- [11] BEEKMANS G, DUBBS B, MOFFAT K. Linux from scratch [M]. Bloomington: Iuniverse Inc, 2024.
- [12] QIU X P. Neural networks and deep learning [M]. Beijing: China Machine Press, 2020.
邱锡鹏. 神经网络与深度学习[M]. 北京: 机械工业出版社出版, 2020.
- [13] CHEN R. Research on text classification algorithm based on neural network and domain adaptation [D]. Zibo: Shandong University of Technology, 2024.
陈榕. 基于神经网络和领域自适应的文本分类算法研究[D]. 淄博: 山东理工大学, 2024.
- [14] TREISMAN A, MGELAD G. A feature-integration theory of attention [J]. Cognitive Psychology, 1980, 12(1): 97–136.
- [15] FENG X J, ZENG Y Z. Joint deep modeling of rating matrix and reviews for recommendation [J]. Chinese Journal of Computers, 2020, 43(5): 884–900.
冯兴杰, 曾云泽. 基于评分矩阵与评论文本的深度推荐模型[J]. 计算机学报, 2020, 43(5): 884–900.
- [16] CHOI E, BAHADORI M T, SCHUETZ A, et al. RETAIN: an interpretable predictive model for healthcare using reverse time attention mechanism [DB/OL]. <https://arxiv.org/abs/1608.05745>.
- [17] XU D M. Principles of operating systems: Linux edition [M]. Beijing: National Defense Industry Press, 2004.
徐德民. 操作系统原理 Linux 篇[M]. 北京: 国防工业出版社, 2004.
- [18] ZHANG J. Theory and application of Linux operating system [J]. Manufacturing Automation, 2012, 34(4): 154–156.
张君. Linux 操作系统构建原理与应用[J]. 制造业自动化, 2012, 34(4): 154–156.
- [19] ZHANG Y X, SONG H, ZHANG G. Tutorial on computer operating systems (4th edition) [M]. Beijing: Tsinghua University Publishing House, 2013.
张尧学, 宋虹, 张高. 计算机操作系统教程(第4版)[M]. 北京: 清华大学出版社, 2013.
- [20] ZHOU D H. Research and implementation on optimization technology of packet classification based on Netfilter [D]. Changsha: National University of Defense Technology, 2024.
周东浩. 基于 Netfilter 的包分类优化技术的研究与实现[D]. 长沙: 国防科学技术大学, 2024.
- [21] GUPTA P, MCKEOWN N. Algorithms for packet classification [J]. IEEE Network, 2001, 15(4): 24–32.
- [22] TAN S B. Research on high-performance text categorization [D]. Beijing: Graduate University of Chinese Academy of Sciences (Computer Architecture), 2024.
谭松波. 高性能文本分类算法研究[D]. 北京: 中国科学院研究生院(计算技术研究所), 2024.
- [23] REN H, WANG X G. Review of attention mechanism [J]. Journal of Computer Applications, 2021, 41(S1): 1–6.
任欢, 王旭光. 注意力机制综述[J]. 计算机应用, 2021, 41(S1): 1–6.
- [24] GAO G, ZHAO Y, YU S J, et al. Research on text classification algorithm based on GNN [J]. Computer Technology and Development, 2023, 33(5): 138–144.
高贵, 赵阳, 于舒娟, 等. 基于 GNN 的文本分类算法研究[J]. 计算机技术与发展, 2023, 33(5): 138–144.
- [25] XIAO C, XU L L. Loosely coupled graph convolutional neural network for text classification [J]. Journal of Chinese Computer Systems, 2021, 42(3): 449–453.
肖驰, 徐林莉. 松耦合图卷积文本分类网络[J]. 小型微型计算机系统, 2021, 42(3): 449–453.
- [26] ZHENG C, NI X H, ZHANG S H, et al. Information fusion combined with GNN is used for inductive text classification [J]. Journal of Chinese Computer Systems, 2023, 44(6): 1170–1176.
郑诚, 倪显虎, 张苏航, 等. 结合 GNN 的信息融合用于归纳式文本分类[J]. 小型微型计算机系统, 2023, 44(6): 1170–1176.
- [27] HU J N, XU W R, GUO J, et al. Study on feature selection methods in Chinese text categorization [J]. Study on Optical Communications, 2005(3): 44–46.
胡佳妮, 徐蔚然, 郭军, 等. 中文文本分类中的特征选择算法研究[J]. 光通信研究, 2005(3): 44–46.
- [28] XU X X. Multi-label text classification based on neural network [D]. Chengdu: University of Electronic Science and Technology of China, 2024.
徐香香. 基于神经网络的多标签文本分类[D]. 成都: 电子科技大学, 2024.

(责任编辑: 陈维捷)