

融合 BERT 与图卷积网络的软件需求自动分类^①



关 慧^{1,2}, 高 琦¹, 韩志远¹

¹(沈阳化工大学 计算机科学与技术学院, 沈阳 110142)

²(辽宁省化工过程工业智能化技术重点实验室, 沈阳 110142)

通信作者: 关 慧, E-mail: h.guan@syuct.edu.cn

摘 要: 考虑到软件需求文本区别于其他普通文本的独特领域信息外, 还包含一些重要的上下文关系以及固有的二义性问题, 本文提出了一个图卷积与 BERT 融合的软件需求自动分类模型——BERT-FGCN (BERT-FusionGCN), 将图卷积网络 (GCN) 用于软件需求分类领域, 利用 GCN 对邻居节点信息进行信息传播和特征聚合的优势, 捕捉需求语句中单词或句子之间的上下文关系, 以进一步提高需求分类的结果. 首先构建需求文本的文本共现图和依存句法图, 将两种图进行融合来捕获句子的结构信息, 利用 GCN 对建模后的需求语句的图结构进行卷积得到图向量, 最后将图向量与 BERT 特征提取后得到的向量进行融合, 以此来对软件需求文本自动分类. 在 PROMISE 数据集上进行实验, BERT-FGCN 在二分类上的 $F1$ 分数达到 95%, 多分类任务的 $F1$ 分数提高 2%.

关键词: 软件需求分类; BERT; 图卷积网络 (GCN); 非功能需求; PROMISE

引用格式: 关慧,高琦,韩志远.融合 BERT 与图卷积网络的软件需求自动分类.计算机系统应用,2025,34(1):145-152. <http://www.c-s-a.org.cn/1003-3254/9754.html>

Integration of BERT and GCN for Automatic Software Requirement Classification

GUAN Hui^{1,2}, GAO Qi¹, HAN Zhi-Yuan¹

¹(College of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang 110142, China)

²(Key Laboratory of Industrial Intelligence Technology on Chemical Process of Liaoning Province, Shenyang 110142, China)

Abstract: Considering the unique domain-specific information inherent in software requirement texts, as well as the important contextual relationships and inherent ambiguities they contain, this study proposes a model that integrates graph convolutional network (GCN) with BERT for automatic software requirements classification, named BERT-FGCN (BERT-FusionGCN). This model leverages the advantages of GCN in propagating information and aggregating features from neighboring nodes to capture the contextual relationships between words or sentences in requirement statements, thereby improving the classification results. Initially, a text co-occurrence graph and a dependency syntax graph of requirement texts are constructed. These graphs are then fused to capture the structural information of the sentences. The GCN is then employed to perform convolution on the graph structure of the modeled requirement statements to obtain graph vectors. Finally, these graph vectors are fused with the vectors obtained from BERT feature extraction to achieve automatic classification of software requirement texts. Experiments conducted on the PROMISE dataset demonstrate that BERT-FGCN achieves an $F1$ -score of 95% in binary classification, and increases the $F1$ -score by 2% in multi-class classification tasks.

Key words: software requirement classification; BERT; graph convolutional network (GCN); non-functional requirement (NFR); PROMISE

① 基金项目: 辽宁省 2021 年度教育科学研究经费项目 (LJKZ0434)

收稿时间: 2024-06-05; 修改时间: 2024-07-10; 采用时间: 2024-08-20; csa 在线出版时间: 2024-11-28

CNKI 网络首发时间: 2024-11-29

软件需求分类是对软件需求进行系统化的分类以更好地理解、管理和实现这些需求。通过将需求分成不同的类别,开发团队和利益相关者可以更清晰地识别各类需求的性质、优先级和相互关系,从而提高软件开发的效率和质量。软件需求可以分为两种类型:功能需求(functional requirement, FR)和非功能需求(non-functional requirement, NFR)。前者主要是对软件系统的服务、函数行为的描述,而后者往往涉及可靠性、可用性、安全性、隐私性或软件权限等非功能问题。对软件需求的不正确处理会增加软件维护成本并影响软件质量^[1],因此,软件需求的识别和分类已成为需求工程中的关注领域。

软件需求之间的关系复杂且紧密,如电商系统的购物车功能(功能性需求)可能会影响系统响应时间(非功能性需求)。某些需求的定义和说明可能分布在文档的不同部分,需要结合多个段落甚至多个文档来理解。例如,业务需求在文档开头定义,具体的功能性需求在文档中间详细说明。然而,目前的技术所采用的特征选择技术无法保持需求语句的句子和句法结构且泛化能力有限。本文提出一种融合BERT和图卷积的需求分类模型,通过不同的构图方法捕获需求语句的句子结构信息,引入BERT获取句子中的上下文信息并提高泛化能力。将两种特征信息进行融合丰富句子的语义信息,从而提高需求语句的分类效果。

1 相关工作

在过去的几年中,研究人员已经对需求工程中的功能需求和非功能需求进行了分类^[2]。Amasaki等人^[3]比较了TF-IDF、Word2Vec、Doc2Vec和稀疏复合向量等5种词向量模型的性能,证明利用Doc2Vec和稀疏复合向量比传统的词嵌入取得更好的分类效果。Tiun等人^[4]提出了FastText,改进了字符级分类的单词嵌入。此外,Devlin等人^[5]提出了BERT模型,在NLP任务中胜过现有的词表示技术。Hey等人^[6]设计了基于迁移学习的NoRBERT方法用于需求分类,在需求分类方面表现了出色的性能。

Kurtanovic等人^[7]使用SVM,将需求自动分类为FR和NFR以及非功能需求的子类。Abad等人^[8]提出了监督和无监督机器学习技术来提高决策树的性能^[9]。作者设计了NFR预处理方法,使用4种机器学习算法比较了使用预处理和未使用预处理的分类器性能,F1

分数分别为45%和90%。

深度学习的快速发展产生了许多用于NLP的语言模型。CNN^[10]和RNN^[11]越来越多地应用于需求分类任务。Navarro-Almanza等人^[12]使用CNN对PROMISE数据集的12个NFR子类进行分类,F1分数为77%。Dekhtyar等人^[13]利用CNN进行NFR分类,并引入Word2Vec嵌入^[14]来代表需求中的每个单词,在分类准确性方面提供了准确和可衡量的改进。Baker等人^[15]提出了一种CNN模型,可以有效地将NFR分类为5个不同的子类,同时实现F1分数范围在82%–92%之间。Rahimi等人^[16]使用4种深度学习模型LSTM、BiLSTM、GRU和CNN的组合来构建NFR分类器并开发两阶段分类系统。Kici等人^[17]通过实验对比,提出了蒸馏BERT,在需求分类任务中优于LSTM和BiLSTM。

图结构数据可以编码句子中复杂的语义关系,学习更多的特征信息。许多研究人员^[18,19]已经提出了复杂的神经网络模型,通过引入规则的网格结构来处理任意结构的图形。Kipf等人^[20]提出图卷积网络GCN,可以直接在图上运行神经网络,并在多个任务中实现分类效果。一些作者^[21,22]探索了不同的组合方法结合图神经网络进行文本分类,通过句子和文档之间的依赖关系构建图表或句子中单词之间的关系。Yao等人^[23]提出了一种基于GCN的文本分类算法,该算法通过词共现和文档-词关系为语料库构建独立的文本图,该方法在没有外部词嵌入的情况下,在几个文本分类任务上优于当时最先进的算法。

2 研究方法

本节介绍了用于软件需求分类模型的整体结构。BERT-FGCN模型结构如图1所示。首先对每个需求语句进行文本共现统计和依存句法分析,得到两种不同类型的图信息。将两种图信息进行融合以获得更丰富的关系表示。利用GCN对文本关系图进行卷积操作,通过两层GCN进行信息传递和聚合,得到需求语句的图向量表示。使用BERT进行特征提取,将得到的特征向量与GCN输出的图向量进行融合,最后使用分类器对需求语句进行分类。通过并行处理,可以让GCN专注于捕捉句子的结构信息和依存关系,BERT捕捉词语的上下文依赖关系,充分发挥各自优势,不必在早期阶段进行信息融合,避免信息损失,同时提高处理效率,简化模型设计,最终提高文本分类的效果。

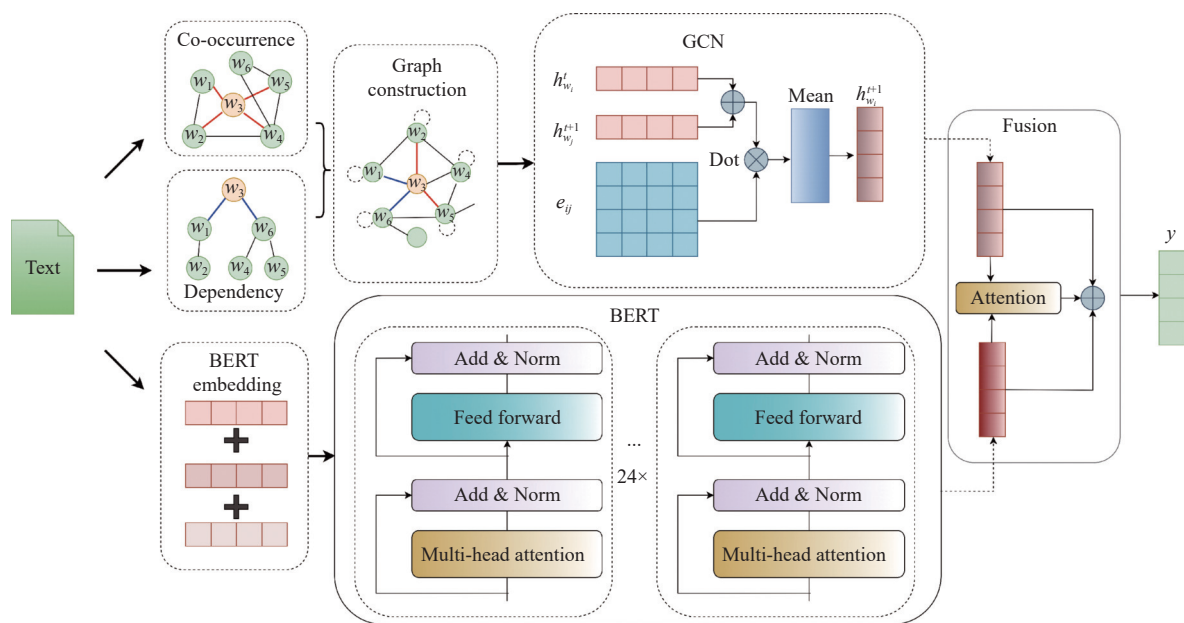


图1 BERT-FGCN 结构图

2.1 文本关系图构建

将一条软件需求语句转换为图结构的数据, 才能利用 GCN 对其进行卷积操作从而获得句子的全局信息, 因此首先要做的是对需求语句进行图建模. 对于整个语料库 $D = \{d_1, d_2, d_3, d_4, \dots, d_N\}$, N 是需求语句的数量, 需求语句 d_i 经过分词、去除停用词、词干或词性还原后得到单词序列 $W = \{w_1, w_2, w_3, w_4, \dots, w_N\}$, 其中 w_i 是句子中的第 i 个单词, n 是句子中单词的数量. W 对应的文本图 G 定义为 $G=(V, E, Weight)$, 其中 V 为节点集合, E 是边的集合. $v_i \in V$ 表示节点, 边 $(v_i, v_j) \in E$. 创建一个 $Dic \times Dic$ 大小的全局关系矩阵 M , 其中 Dic 是词典单词的数量, M_{ij} 表示单词 w_i 和 w_j 的共现次数, 初始化为 0.

(1) 文本共现图

在文本共现图中, 节点代表文本中的单词, 边代表这些单词之间的共现关系. 假设滑动窗口大小为 m , 窗口内出现的词语认定为具有共现关系. 句子 W 进行共现分析的步骤如下.

- 1) 将句子 W 中的所有单词作为节点.
- 2) 遍历每个节点, 若单词 w_j 出现在以节点 w_i 为中心的窗口内, 将 w_i 与 w_j 作为一条边添加到 E .
- 3) 全局关系矩阵中 w_i 和 w_j 对应的元素 M_{ij} 增加 1.

以“The system shall interface with the student master server”为例, 将需求语句转换为文本共现图如

图2所示. 在滑动窗口为3的情况下, 单词的实线表示两个单词节点之间通过滑动窗口而连接的词共现关系, 单词节点“interface”与其他节点之间有4条边. 单词“interface”除了与直接相邻节点“shall”和“with”连接的边③和边④, 还有其他长距离邻居节点的边①和边②.

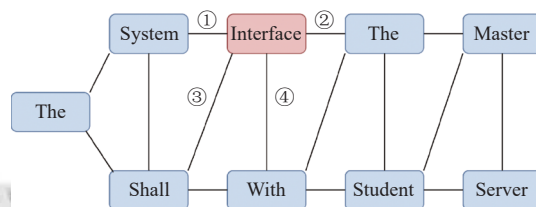


图2 滑动窗口为3的词共现关系图

(2) 依存句法图

依存句法图是一种用于表示句子结构的图结构, 句子中的每个单词表示为图中的一个节点, 节点包含单词本身以及其在句子中的位置信息. 而边则表示单词之间的依存关系. 依存关系是指一个单词对另一个单词的依赖或修饰关系, 是通过单词之间的语法关系来定义的. 对于每个单词的依存关系, 将其表示为图中的一条边. 边的起始节点是依存父节点, 结束节点是当前单词. 文本 W 进行依存分析的步骤如下.

- 1) 对单词列表 S 进行依存句法分析, 得到依存关系集合: $R = \{(w_1, w_{head_1}, r_1) \dots (w_n, w_{head_n}, r_n)\}$, 其中 w_{head_i} 代表与单词 w_i 具有依存关系的单词 w_j , 依存关系类型表示为 r .

2) 对每个单词 w_i , 将 w_i 和对应依存关系的单词 w_j 添加为一条无向边 $e_{ij}=(w_i, w_j)$ 到 E .

3) 全局关系矩阵中 w_i 和 w_j 对应的元素 M_{ij} 增加 1.

如图 3 所示, 句法分析的一条关系线从单词“allow”

指向单词“save”, 而位于两个单词中间的“xcomp”线表明这两个单词的句法信息是补语修饰关系, 表示允许用户进行的动作. 如果一个词语依赖于另一个词语, 则在它们之间添加一条有向边. 通过依存句法分析, 句子中丰富的句法信息可以用来捕捉需求句子的更多特征.

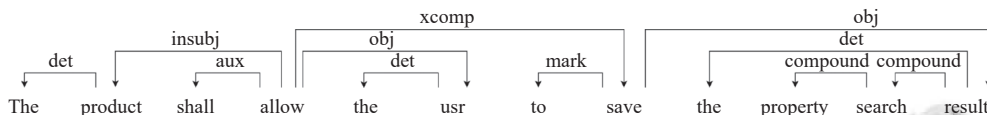


图 3 对需求语句进行语法解析

(3) 图融合

文本共现图主要考虑了单词在邻域内的共现关系, 而句法依存图相对复杂, 它是通过分析文本的语法特性来构建的. 在基于 GCN 的方法中, 每个图的节点只能通过单条边连接到其他节点. 因此, 将构建出的两个图的边融合成一个边, 减少图的节点规模, 更能代表文本关系, 丰富节点之间的信息, 同时也是模型 BERT-FGCN 中 F 的含义. 因此, 在应用于 GCN 之前, 可将不同的上下文关系集成到单个图中, 而不是将 GCN 单独应用于不同的文本结构. 由于保留了重复出现的单词, 通过更新单词节点的表示, 可以保持完整的序列和语义信息.

通过构建文本共现图和句法依存图得到两个边的集合, 将两个集合进行相加以融合边的信息. 在构建图时创建全局关系矩阵 M , 矩阵 M 不仅统计了文本共现信息, 在进行句法依存分析时同样将单词之间的依存信息作为共现信息添加到全局关系矩阵 M 中, 因此可以利用全局关系矩阵 M 来为单词之间边的设置权重信息.

PMI (pointwise mutual information) 用于衡量两个词在语料库中出现的概率与它们各自单独出现的概率的比值. PMI 常用于构建文本共现图中, 用于分析文本数据中词语之间的关联程度. 通过全局关系矩阵 M 计算 PMI 来对边进行加权赋值. PMI 的计算过程如下:

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i) \cdot P(w_j)} \quad (1)$$

其中, $P(w_i, w_j)$ 是词 w_i 和 w_j 共现的概率, $P(w_i)$ 和 $P(w_j)$ 分别为词 w_i 和 w_j 的边缘概率. PMI 中的会出现负值, 将负值替换为 0, 从而强调了正相关性, 减轻了负值对分析结果的影响:

$$weight_{ij} = PPMI(w_i, w_j) = \max(PMI(w_i, w_j), 0) \quad (2)$$

文本关系图 G 的构建算法如算法 1.

算法 1. 文本关系图构建算法

```

1) preprocess( $W$ ); //进行预处理
2)  $Edge = list()$ ,  $M = []$ ,  $pmi$ ,  $pmi\_list$ ,  $G$  //初始化边  $E$ , 关系矩阵  $M$ , 权重列表  $pmi\_list$ ,  $pmi$  和图  $G$ 
3)  $window\_size = m$  //设置滑动窗口大小
4) for  $i$  in range(len( $S$ )): //共现关系分析
    for  $j$  in range(window_size):
         $M[w_i][w_j] += 1$ 
         $E.append((w_i, w_j))$ 
5)  $dep = nlp(S)$  //句法依存分析
    for  $w_i$  in  $dep$ :
         $M[w_i][w_i.child] += 1$ 
         $E.append((w_i, w_i.child))$ 
6) for  $w_i, w_j$  in  $E.items()$ : //计算  $PMI$ 
     $pmi\_list.append(pmi(w_i, w_j))$ 
7)  $G.add\_edge(E)$ 
8)  $G.add\_vertex(W)$ 
9)  $G.add\_weight(pmi\_list)$ 

```

由于文本图中包含的单词之间的边比单个关系文本图更多, 因此较浅的 GCN 模型会学习到更好的表示, 避免过度参数化.

2.2 图嵌入

GCN 中的核心操作是图卷积操作, 类似于传统卷积神经网络中的卷积操作, 但是适用于图结构数据, 一般有两层卷积构成. 在 GCN 中, 每个节点都有一个特征向量, 表示节点的属性信息, 而边则表示节点之间的关系. 对于每个节点, 首先收集其邻居节点的特征表示, 并将它们进行聚合, 将聚合后的邻居特征与当前节点的特征进行结合, 生成新的节点表示, 最后将上述特征更新过程应用于整个图的所有节点, 从而得到图的新表示. 在 GCN 中, 第 l 层的节点特征 $H^{(l)}$ 由前一层的节点特征 $H^{(l-1)}$ 通过以下公式计算得到:

$$H^{(l)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l-1)} W^{(l)}\right) \quad (3)$$

$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij} \quad (4)$$

其中, A 是邻接矩阵, $\tilde{A}=A+I$ 是邻接矩阵加上自连接的结果, I 是单位矩阵, $\tilde{D}$ 是 D 的度矩阵, $W^{(l)}$ 是第 l 层的权重矩阵, 用于学习节点特征的线性变换, σ 是 ReLU 激活函数. 每个节点的新特征表示由其邻居节点的特征通过权重 $W^{(l)}$ 进行线性变换得到, 然后通过激活函数进行非线性变换, 得到最终的特征表示.

2.3 特征融合

BERT 的输入包含单词向量, 段向量和位置向量 3 个部分, 单词向量即对应词的嵌入表示, 段向量表示的是该词所在句子的嵌入表示, 位置向量即该词在句子中对应的位置嵌入表示. 软件需求语句经过 BERT 模型后, 每一句话的会添加上[CLS]特殊标记, 结尾是[SEP]标记, 需求语句经过 BERT 分词后表示为 $W_{\text{BERT}}=\{\text{[CLS]}, w_1, w_2, w_3, \dots, w_n, \text{[SEP]}\}$. 在文本分类任务中一般只输入一个句子, [CLS]标记通常被用作整个句子的语义表示. 在微调的训练过程中提取出文本对应的[CLS]向量, 表示为 R_{CLS} .

经过图卷积, 文本图中的每一个节点都获取到了新的表示, 这些表示蕴含着文本图中的信息, 然后将文本图中所有的节点表示求和, 得到了整个文本图的表示 H :

$$H = \sum_{i \in N} H_i \quad (5)$$

其中, H_i 表示的是经过图卷积后文本中第 i 个节点的表示, N 表示的是文本中的节点的数量, H 是整个文本的图表示.

将文本的图表示 H 和 BERT 特征提取得到 R_{CLS} 进行特征向量的融合, 因为两种特征包含不同的信息, 所占的权重也不同, 权重越大, 特征越重要. 注意力机制通过计算输入序列中每个位置与其他位置的相关性来确定注意力权重, 因此能够自适应地学习不同位置之间的关联程度. 使用自注意力机制来动态调整每个特征向量的权重. 使用缩放的点积注意力, 根据式 (6) 计算注意力分数:

$$A = \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (6)$$

其中, 分母是用于控制注意力分数比例的比例因子,

d 是 Q 和 K 向量的维度. 将注意力分数 A 用于两个特征的融合, 最后将整个文本的表示放入全连接层和激活函数进行训练获得文本的分类标签, 训练过程中通过交叉熵来实现最小化损失.

$$R = \alpha H + (1-\alpha)R_{\text{CLS}} \quad (7)$$

$$y = \text{Softmax}(\sigma(WR + b)) \quad (8)$$

$$\text{loss} = -y_{\text{true}} \log(y) \quad (9)$$

其中, W 是权重矩阵, b 是偏置. σ 是 ReLU 激活函数, y 文本的预测标签, y_{true} 是文本的真实标签.

3 实验分析

3.1 数据集

为确保实验具有可比性和说服力, 使用 PROMISE 数据集, 该数据集通常用于软件需求分类领域. PROMISE 数据集由 625 条需求描述文本组成, 其中包括 255 个功能需求和 370 个非功能需求. NFR 被标记为 11 个类型, 分别是: 可用性 (A), 合法性 (L), 外观 (LF), 可维护性 (MN), 操作性 (O), 性能 (PE), 可扩展性 (SC), 安全性 (SE), 可用性 (US), 容错性 (FT), 可移植性 (PO). 数据集由 15 个不同的项目组成, 为测试泛化能力提供了基础.

3.2 基线模型

为了评估 BERT-FGCN 模型的性能, 选取软件需求分类领域的深度学习模型作为基线方法进行比较:

CNN: 由嵌入层、卷积层、最大池化层组成^[24].

ANN: 由 Word2Vec 和全连接层组成^[15].

Att-BiLSTM: 该模型通过 Glove 获得单词表示, 对单词向量使用 BiLSTM, 注意力机制从 BiLSTM 层提取重要单词, 最后全连接层分类需求^[25].

NoRBERT: 将迁移学习方法和 BERT 模型应用于需求分类任务, 并考虑了需求的功能和质量方面^[6].

DBGAT: 该模型结合了图注意力机制 (GAT) 和 BERT 模型用于软件需求的分类^[26].

BertGCN: 将 BERT 模型的输出作为 GCN 的输入进行文本分类^[27].

3.3 实验设置及评价指标

模型使用深度学习框架 PyTorch 实现, 构图方法使用 DGL 库. 使用斯坦福大学的 NLP 工具包来处理图构建中的需求语句, 并将需求语句最大句子长度设

置为 30. 采用“BERT-base-uncased”模型初始化 BERT 的权重. 关于单词嵌入, 使用预先训练的 GloVe 向量, 其中 $k=200$ 来初始化单词嵌入, 词汇表外 (OOV) 单词被替换为 UNK, 并从均匀分布 $[-0.01, 0.01]$ 中随机抽样. 默认情况下, 堆叠两层 GCN 模块, GCN 隐藏层大小设置为 200. *dropout* 大小设置为 0.5, 学习率为 $1E-5$, 使用 Adam 优化器. 对于基线模型, 使用默认的参数设置, 与原始论文或实现保持一致, 并使用其论文中给出的结果.

对于两个分类任务使用 $F1$ 分数 ($F1$) 作为评价指标, 验证模型有效性时使用精确度和召回率来评估模型.

3.4 实验结果分析

通过建立 3 个任务来评估模型在 PROMISE 公共数据集上的性能.

(1) FR/NFR 分类

任务 1 是分类功能需求 (FR) 和非功能需求 (NFR) 两类. 使用 PROMISE 数据集并折叠所有 NFR 子类来表示 NFR, 此时 NFR 有 370 条语句. 模型在 PROMISE 数据集上进行训练和测试, 以预测 FR 和 NFR 经过模型分类后得到分类精确度、召回率和 $F1$ 分数, 并与其他需求分类模型比较, 详细结果如表 1 所示.

表 1 PROMISE 数据集二分类 $F1$ 分数

Approach	FR	NFR
ANN	0.40	0.96
CNN	0.74	0.82
Att-BiLSTM	0.81	0.87
NoRBERT	0.90	0.93
BertGCN	0.92	0.94
DBGAT	0.92	0.96
BERT-FGCN	0.94	0.95

ANN 在 FR 上的 $F1$ 分数仅为 0.40, 而在 NFR 上的 $F1$ 分数为 0.96, 表明 ANN 模型在处理这些特定任务时, 欠缺足够的复杂性来捕捉数据的特征. CNN 在两个类别上的表现较为均衡, 相对于 ANN 有显著的性能提升, 因为 CNN 能够更好地提取局部特征. Att-BiLSTM 在两个类别上的表现也较为均衡, $F1$ 分数分别为 0.81 和 0.87. Attention 机制帮助模型更好地捕捉到长距离依赖信息, 提升了性能. NoRBERT 在 FR 上的 $F1$ 分数达到了 0.90, 表明预训练语言模型的强大语言理解能力使其在处理多样化和复杂数据时表现优异. BertGCN 相比于 NoRBERT 的 $F1$ 获得提升, 表明 BERT 与 GCN 的互补优势. DBGAT 的表现进一步提升, 在

NFR 上 $F1$ 分数高达 0.96, 显示了图结构和注意力机制的有效结合. BERT-FGCN 在所有模型中表现最佳, 两个类别上的 $F1$ 分数分别为 0.94 和 0.95. 这表明结合 BERT 的强大语言理解能力和 GCN 的特征提取能力是解决软件需求分类任务的一种有效方法.

(2) 4 个 NFR 子类别分类

PROMISE 中可用性有 67 条样本, 安全性有 66 条样本, 可操作性有 62 条样本, 性能有 54 条样本. 此 4 类 NFR 数量分布均衡, 可以避免数据类别不平衡带来的干扰. 因此, 在任务 2 中使用这 4 个 NFR 子类作为相对平衡的样本测试分类模型. 经过模型训练, 得到 4 个 NFR 类的 $F1$ 分数分别为 0.86、0.95、0.96、0.89, 并将分类结果和其他模型进行比较, 具体结果如图 4 所示.

通过对比不同分类模型在 4 个分类指标上的 $F1$ 分数, BERT-FGCN 模型在各个指标上均表现出显著的优势. 相比于 ANN, BERT-FGCN 在所有类别上的得分均有所提升, 尤其在 PE 和 SE 指标上, BERT-FGCN 分别提升了 0.31 和 0.35, 显示出其在捕捉复杂模式和关系上的明显优势. 相比 CNN, BERT-FGCN 在 PE 和 SE 类别上分别提升了 0.16 和 0.37, 展现了其在处理序列数据和长距离依赖关系方面的卓越能力. 与 Att-BiLSTM 相比, BERT-FGCN 在 PE 和 US 指标上分别提升了 0.22 和 0.21, 表明其综合性能更为优越. 尽管 NoRBERT 和 DBGAT 在某些指标上表现接近, 但 BERT-FGCN 在 SE 和 PE 指标上仍分别提升了 0.05 和 0.04. BERT-FGCN 比 BertGCN 总体性能也有提升, 表明 BERT 与 GCN 进行并行融合对于软件需求分类的有效性. 总体而言, BERT-FGCN 结合了 BERT 和图卷积网络的优势, 在处理软件需求方面表现出色, 具有较强的泛化能力和卓越的分类性能.

3.5 消融分析

由于所提出的模型由几个部分以及一系列的模型变体组成, 在本节中进一步验证模型 BERT-FGCN 每部分的有效性. 所涉及的模型变体如下.

w/o BERT: BERT-FGCN 没有结合 BERT;

w/o Co-occurrence: BERT-FGCN 没有构建关系共现图, 使用了句法依存分析;

w/o Dependency: BERT-FGCN 没有构建句法依存图, 仅考虑词共现关系;

w/o Attention: BERT-FGCN 没有使用注意力机制.

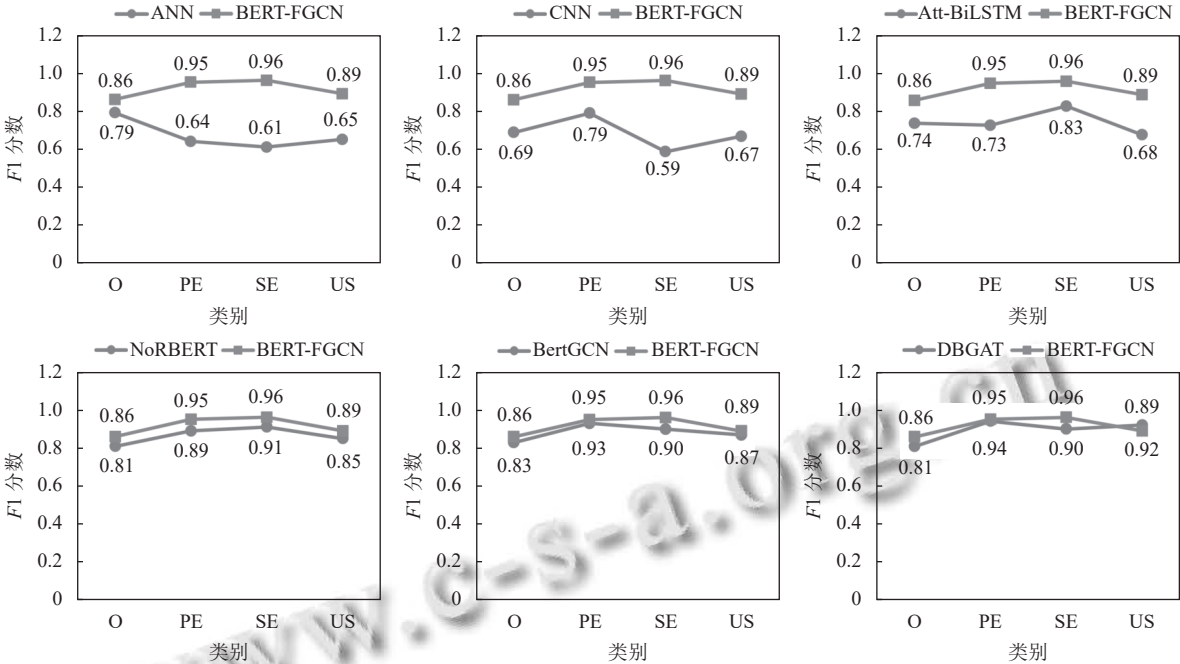


图4 PROMISE 数据集 4 个 NFR 分类 F1 分数

使用 PROMISE 数据集训练和测试所有模型变体,并将其与提出的 BERT-FGCN 模型进行了比较。表 2 显示了模型中各部分的有效性。与 BERT-FGCN 模型相比,每个变体模型性能都有所下降,说明 BERT-FGCN 模型中各个部分在处理软件需求文本的过程中能够有效提升模型的整体性能。

表 2 模型变体的性能数据

Variants	精确度	召回率	F1
w/o BERT	0.88	0.86	0.87
w/o Co-occurrence	0.88	0.88	0.88
w/o Dependency	0.87	0.92	0.89
w/o Attention	0.90	0.94	0.93
BERT-FGCN	0.92	0.96	0.94

4 结论与展望

本文通过融合多种构图方法从而获取句子更丰富的语义结构信息,通过图卷积神经网络将句子信息进行传递更新,得到文本的图向量表示,同时通过微调 BERT 模型获取句子的上下文信息,最后将两者信息进行融合再进行软件需求分类。通过在软件需求领域的公开数据 PROMISE 数据集上进行实验,并与其他方法进行对比,本文提出的方法 BERT-FGCN 在 FR/NFR 分类和 4 个 NFR 分类任务中取得很高的性能,总体上超过了其他方法。软件需求领域公开数据集较少,很多需求类别存在不平衡的现象,这对提升模型性能带来

了一定的挑战。未来的工作将研究数据不平衡的问题,从而提高模型的性能,更好地适应实际的软件项目需求。

参考文献

- 1 卢蒙蒙. 用户评论中非功能需求的自动分类 [硕士学位论文]. 武汉: 武汉大学, 2018.
- 2 Dias Canedo E, Cordeiro Mendes B. Software requirements classification using machine learning algorithms. Entropy, 2020, 22(9): 1057. [doi: 10.3390/e22091057]
- 3 Amasaki S, Leelaprute P. The effects of vectorization methods on non-functional requirements classification. Proceedings of the 44th Euromicro Conference on Software Engineering and Advanced Applications. Prague: IEEE, 2018. 175–182.
- 4 Tiun S, Mokhtar UA, Bakar SH, et al. Classification of functional and non-functional requirement in software requirement using Word2Vec and fast Text. Journal of Physics: Conference Series, 2020, 1529(4): 042077. [doi: 10.1088/1742-6596/1529/4/042077]
- 5 Devlin J, Chang MW, Lee K, et al. 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. Minneapolis: ACL, 2019. 4171–4186.
- 6 Hey T, Keim J, Koziolk A, et al. NoRBERT: Transfer learning for requirements classification. Proceedings of the

- 28th IEEE International Requirements Engineering Conference. Zurich: IEEE, 2020. 169–179.
- 7 Kurtanović Z, Maalej W. Automatically classifying functional and non-functional requirements using supervised machine learning. Proceedings of the 25th IEEE International Requirements Engineering Conference. Lisbon: IEEE, 2017. 490–495. [doi: [10.1109/RE.2017.82](https://doi.org/10.1109/RE.2017.82)]
- 8 Abad ZSH, Karras O, Ghazi P, *et al.* What works better? A study of classifying requirements. Proceedings of the 25th IEEE International Requirements Engineering Conference. Lisbon: IEEE, 2017. 496–501. [doi: [10.1109/RE.2017.36](https://doi.org/10.1109/RE.2017.36)]
- 9 Hussain I, Kosseim L, Ormandjieva O. Using linguistic knowledge to classify non-functional requirements in SRS documents. Proceedings of the 13th International Conference on Applications of Natural Language to Information Systems. London: Springer, 2008. 287–298.
- 10 Winkler J, Vogelsang A. Automatic classification of requirements based on convolutional neural networks. Proceedings of the 24th IEEE International Requirements Engineering Conference Workshops. Beijing: IEEE, 2016. 39–45. [doi: [10.1109/REW.2016.021](https://doi.org/10.1109/REW.2016.021)]
- 11 Mikolov T, Karafiát M, Burget L, *et al.* Recurrent neural network based language model. Proceedings of the 11th Annual Conference of the International Speech Communication Association. Makuhari: ISCA, 2010. 1045–1048.
- 12 Navarro-Almanza R, Juarez-Ramirez R, Licea G. Towards supporting software engineering using deep learning: A case of software requirements classification. Proceedings of the 5th International Conference in Software Engineering Research and Innovation. Merida: IEEE, 2017. 116–120. [doi: [10.1109/CONISOFT.2017.00021](https://doi.org/10.1109/CONISOFT.2017.00021)]
- 13 Dekhtyar A, Fong V. RE data challenge: Requirements identification with Word2Vec and TensorFlow. Proceedings of the 25th IEEE International Requirements Engineering Conference. Lisbon: IEEE, 2017. 484–489. [doi: [10.1109/RE.2017.26](https://doi.org/10.1109/RE.2017.26)]
- 14 Mikolov T, Sutskever I, Chen K, *et al.* Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2013. 3111–3119.
- 15 Baker C, Deng L, Chakraborty S, *et al.* Automatic multi-class non-functional software requirements classification using neural networks. Proceedings of 43rd IEEE Annual Computer Software and Applications Conference. Milwaukee: IEEE, 2019. 610–615.
- 16 Rahimi N, Eassa F, Elrefaei L. One-and two-phase software requirement classification using ensemble deep learning. Entropy, 2021, 23(10): 1264. [doi: [10.3390/e23101264](https://doi.org/10.3390/e23101264)]
- 17 Kici D, Malik G, Cevik M, *et al.* A BERT-based transfer learning approach to text classification on software requirements specifications. Proceedings of the 34th Canadian Conference on Artificial Intelligence. Canadian Artificial Intelligence Association, 2021.
- 18 Tang H, Mi Y, Xue F, *et al.* An integration model based on graph convolutional network for text classification. IEEE Access, 2020, 8: 148865–148876.
- 19 Henaff M, Bruna J, LeCun Y. Deep convolutional networks on graph-structured data. arXiv:1506.05163, 2015.
- 20 Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. Proceedings of the 5th International Conference on Learning Representations. Toulon: OpenReview.net, 2017.
- 21 Peng H, Li JX, He Y, *et al.* Large-scale hierarchical text classification with recursively regularized deep graph-CNN. Proceedings of the 2018 World Wide Web Conference. Lyon: ACM, 2018. 1063–1072. [doi: [10.1145/3178876.3186005](https://doi.org/10.1145/3178876.3186005)]
- 22 Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016. 3844–3852.
- 23 Yao L, Mao CS, Luo Y. Graph convolutional networks for text classification. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu: AAAI Press, 2019. 7370–7377.
- 24 Fong VL. Software requirements classification using word embeddings and convolutional neural networks [MS. Thesis]. San Luis Obispo City: California Polytechnic State University, 2018.
- 25 Kaur K, Kaur P. SABDM: A self-attention based bidirectional -RNN deep model for requirements classification. Journal of Software: Evolution and Process, 2024, 36(2): e2430. [doi: [10.1002/smr.2430](https://doi.org/10.1002/smr.2430)]
- 26 Li G, Zheng CP, Li M, *et al.* Automatic requirements classification based on graph attention network. IEEE Access, 2022, 10: 30080–30090. [doi: [10.1109/ACCESS.2022.3159238](https://doi.org/10.1109/ACCESS.2022.3159238)]
- 27 Lin YX, Meng YX, Sun XF, *et al.* BertGCN: Transductive text classification by combining GNN and BERT. arXiv:2105.05727, 2021.

(校对责编: 张重毅)