

融合 Bert 和图卷积的深度集成学习软件需求分类

康雁, 吴志伟, 寇勇奇, 张兰, 谢思宇, 李浩

引用本文

康雁, 吴志伟, 寇勇奇, 张兰, 谢思宇, 李浩. [融合 Bert 和图卷积的深度集成学习软件需求分类](#)[J]. 计算机科学, 2022, 49(6A): 150-158.

KANG Yan, WU Zhi-wei, KOU Yong-qi, ZHANG Lan, XIE Si-yu, LI Hao. [Deep Integrated Learning Software Requirement Classification Fusing Bert and Graph Convolution](#)[J]. Computer Science, 2022, 49(6A): 150-158.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[一种新的中文电子病历文本检索模型](#)

New Text Retrieval Model of Chinese Electronic Medical Records

计算机科学, 2022, 49(6A): 32-38. <https://doi.org/10.11896/jsjcx.210400198>

[一种可快速迁移的领域知识图谱构建方法](#)

Fast and Transmissible Domain Knowledge Graph Construction Method

计算机科学, 2022, 49(6A): 100-108. <https://doi.org/10.11896/jsjcx.210900018>

[基于不平衡数据与集成学习的属性级情感分类](#)

Aspect-level Sentiment Classification Based on Imbalanced Data and Ensemble Learning

计算机科学, 2022, 49(6A): 144-149. <https://doi.org/10.11896/jsjcx.210500205>

[一种融合多层次情感和主题信息的 TS-AC-EWM 在线商品排序方法](#)

TS-AC-EWM Online Product Ranking Method Based on Multi-level Emotion and Topic Information

计算机科学, 2022, 49(6A): 165-171. <https://doi.org/10.11896/jsjcx.210400238>

[TI-FastText 自动商品分类算法](#)

TI-FastText Automatic Goods Classification Algorithm

计算机科学, 2022, 49(6A): 206-210. <https://doi.org/10.11896/jsjcx.210500089>

融合 Bert 和图卷积的深度集成学习软件需求分类

康 雁 吴志伟 寇勇奇 张 兰 谢思宇 李 浩

云南大学软件学院 昆明 650091

(562530855@qq.com)

摘 要 随着软件数量和种类的快速增长,有效地挖掘软件需求的文本特征,并对软件功能性需求的文本特征进行分类,成为软件工程领域的一大挑战。软件功能性需求分类为整个软件开发过程提供了可靠的保障,并减小了需求分析阶段潜在的风险和负面影响。但是,软件需求文本的高分散性、高噪声、数据稀疏等特点限制了软件需求分析的有效性。提出双层词汇图卷积网络模型,创新性地对软件需求文本进行图建模,建立软件需求的图神经网络,有效捕获单词的知识边以及单词与文本之间的关系;并提出深度集成学习模型,集成多个深度学习分类模型,对软件需求文本进行分类。在数据集 Windows_a 和数据集 Windows_b 的实验中,融合 Bert 和图卷积的深度集成学习模型的准确率分别达到 96.73%和 95.60%,其明显优于其他文本分类模型,充分证明融合 Bert 和图卷积的深度集成学习模型能有效判别软件需求文本的功能特性,提高软件需求文本分类的准确性。

关键词: 软件需求;文本分类;BERT;GCN;文本特征;集成学习

中图法分类号 TP181

Deep Integrated Learning Software Requirement Classification Fusing Bert and Graph Convolution

KANG Yan, WU Zhi-wei, KOU Yong-qi, ZHANG Lan, XIE Si-yu and LI Hao

College of Software, Yunnan University, Kunming 650091, China

Abstract With the rapid growth of software quantity and types, effectively mine the textual features of software requirements and classify the textual features of software functional requirements becomes a major challenge in the field of software engineering. The classification of software functional requirements provides a reliable guarantee for the whole software development process and reduces the potential risks and negative effects in the requirements analysis stage. However, the validity of software requirement analysis is limited by the high dispersion, high noise and sparse data of software requirement text. In this paper, a two-layer lexical graph convolutional network model(TVGCCN) is proposed to model the graph of software requirement text innovatively, build the graph neural network of software requirement, and effectively capture the knowledge edge of words and the relationship between words and text. A deep integrated learning model is proposed, which integrates several deep learning classification models to classify software requirement text. In experiments of data set Wiodows_A and data Wiodows_B, the accuracy of deep ensemble learning model integrating Bert and graph convolution reaches 96.73% and 95.60% respectively, which is obviously better than that of other text classification models. It is fully proved that the deep ensemble learning model integrating Bert and graph convolution can effectively distinguish the functional characteristics of software requirement text and improve the accuracy of software requirement text classification.

Keywords Software requirements, Text classification, BERT, GCN, Text features, Ensemble learning

1 引言

软件需求是软件开发中不可或缺的一部分,是导致软件成败的根源性因素。随着软件行业的快速发展,软件需求变得愈加复杂,规模愈加庞大,对应的需求变更愈加频繁。需求是一个确定和理解不同用户类的需要和限制的过程。而软件需求源于实际,更具有客观性和易变性。随着时间推移,软件需求会不断改变和拓展,用户在此期间难以给出非常准确和

完美的需求描述,而且对需求的描述中通常包含一些模糊和不全面的信息,不能清晰描述系统边界,开发人员无法清晰了解用户的需求。此外,软件需求数据存在维度高、噪声干扰大、数据稀疏等缺点,国内外对需求文本的分类进行分析研究的成果比较少。在软件开发过程中,挖掘出软件需求文本特征并进行分类对软件的质量和开发进度有重要作用。

国内外对需求文本的分类进行的研究相对较少。现有研究比较了使用手动和自动的方法对需求进行分类^[1-2]的效果。

基金项目:国家自然科学基金(61762092);云南省科技厅重大专项(2019ZE001-1,202002AB080001-6);云南省软件工程重点实验室开放基金(2020SE303)

This work was supported by the National Natural Science Foundation of China(61762092), Major Project of Yunnan Provincial Science and Technology Department(2019ZE001-1,202002AB080001-6) and Open Foundation of Yunnan Key Laboratory of Software Engineering(2020SE303).

通信作者:李浩(lihao707@ynu.edu.cn)

Rahimi 等提出了一套机器学习和数据挖掘方法,用于从需求、特征请求和在线论坛中自动提取质量关注点。然后,他们根据需求规范生成一个基本的目标模型,每个关注点都被建模为一个软目标。Naive 贝叶斯分类器在多个研究^[3-4]中被用于需求的自动分类。Rahimi^[5]引入一种新的集成机器学习技术来分类 FR 语句,以提高其准确性和可用性。该技术结合了不同的机器学习模型,并在加权集成投票方法中使用提高的准确性作为权重,但其所花的时间不是最短的。这 5 个组合模型分别是 Naive Bayes、支持向量机(Support Vector Machine,SVM)、决策树、Logistic 回归和支持向量机分类(Support Vector Machine,SVC)。

传统机器学习对需求文本的表示是高维度、高稀疏的,特征表达能力较弱,此外需要人工进行特征工程,成本很高。而深度学习最初在图像和语音领域取得的巨大成功,也相应推动了其在 NLP 上的发展,使得深度学习的模型在软件需求文本分类上也取得了不错的效果。

本文提出了一种融合 Bert 和图卷积的深度集成学习软件需求分类模型,创新地对软件需求文本进行图建模,利用 Bert 抽取更深层次的文本特征,利用图卷积网络对软件需求文本进行分类,有效提高了软件需求文本分类的准确性。本文的主要贡献有以下几方面:

(1)通过抓取大量的国内外软件需求文本数据,并对国内外软件需求文本数据进行归并化处理,创新性地对需求文本数据进行图建模,有效地捕捉了需求文本中单词的知识边以及单词与文本之间的关系。

(2)提出一种新的图卷积神经网络 TVGCN,TVGCN 能更好地捕捉单词与文本之间的关系,建立软件需求的图神经网络。将 TVGCN 与 Bert 融合,可以捕获更深层次的图结构信息,从而提高软件需求文本分类的准确性。

(3)提出 BERT 和图卷积的不同组合模型,利用集成模型有效地组合不同的深度集成学习软件需求分类模型,对多种深度学习的分类模型进行集成学习,来捕获数据中的线性以及非线性的复杂关系,有效提高模型的分类性能。

2 相关研究

在软件开发过程中,软件需求文本是软件开发的关键,对软件需求文本进行分类能帮助软件开发人员高效地进行软件设计。但从复杂的软件需求文本中提取有价值的信息非常困难,并且软件需求的类型复杂多样,对相似性需求进行分类是软件工程存在的很大挑战。

2.1 需求文本分类

软件是计算机技术开发的核心和关键,软件的开发和应用更需要适应时代和社会需求的不断发展和创新。而对软件需求分析及其变更的探讨是软件开发项目的基础和前提条件,也是软件开发能否取得成功的关键因素之一。在软件使用规模不断扩大、应用领域不断拓宽的新阶段,软件需求的分析对软件开发过程至关重要。

需求文本分类不仅有助于需求分析人员进行需求审查,同时也有利于软件开发人员进行软件编程和维护人员对软件系统的修改和变更^[6]。Martin 等^[7]的研究成果表明软件产品的故障主要来源于需求阶段,根据需求文本对功能相似的

语句进行分类,有利于软件开发人员快速了解需求,从而提高软件开发的效率。Zahra 等^[8]利用二值化朴素贝叶斯对软件的非功能需求进行分类,有效提高了分类的准确率,但对需求文本全局单词之间的关系缺乏考虑。文献^[9]利用 Fasttext 对软件需求文本进行分类,Fasttext 模型网络结构简单,并且训练速度比复杂的深度学习分类器更快,能够有效提取文本特征,但无法捕获文本的深层次语义信息。

2.2 文本分类方法

传统的文本分类方法一般先对文本做预处理,利用连续的词袋模型、向量空间模型或基于主题特征的向量表示方法等多种方式进行文本表示并提取特征,然后用分类器对分类结果进行预测。传统的文本分类方法提取特征的能力差,并且需要大量的人工进行特征工程。基于深度学习的文本分类方法能端到端地解决问题,有较强的特征自抽取能力。Kim^[10]提出了文本卷积神经网络(Text Convolutional Neural Networks,TextCNN)模型,该模型并行能力强,训练速度快,能够有效地提取特征。但是 TextCNN 模型中全局的池化操作(pooling)中丢失了文本的结构信息。递归神经网络(Text Recurrent Neural Network,TextRNN)模型,能够提取远距离的上下文信息,但无法从全局考虑单词与单词之间的关系。Yao 等^[11]提出 TextGCN,通过对整个语料库构造图,并用图神经网络联合学习词和文档嵌入,有效捕获文档和词的关系以及全局单词的共现信息。

BERT 模型由 Devlin 等^[12]提出,能提取文本深层次的语义特征,被广泛应用于自然语言处理的多个任务中并获得显著成效。然而,Bert 与大多数基于注意力的深度神经网络一样,主要关注局部连续单词序列,提供局部上下文信息,缺乏对文本全局信息的考虑。Kipf 等^[13]创造性地提出了基于谱图理论的图卷积网络(Graph Convolution Networks,GCN)。GCN 能根据给定的关系图构建一个对称邻接矩阵,并且在卷积操作中根据图中的邻居关系和对应关系融合每个节点的代表。Yao 等^[11]提出了一种新的文本分类的图神经网络方法 Text GCN,首次将整个语料库建模为异构图(Heterogeneous Graph),有效捕获文档和词的关系以及全局单词的共现信息。GCN 及其变体擅长于将图中的全局信息卷积成句子,但无法考虑已经训练完成的本地信息,因此,将 GCN 与捕获本地信息的 Bert 模型结合起来,更利于文本分类。

本文提出一种融合 Bert 和图卷积的深度集成学习软件需求分类模型,创新地对软件需求文本进行图建模,有效地捕捉需求文本中单词的知识边以及单词与文本之间的关系;并利用 Bert 抽取更深层次的文本特征,将图卷积网络与 Bert 相融合对软件需求文本进行分类,有效提高软件需求文本分类的准确性。

3 TVGCN-BERT 模型设计

图 1 为 TVGCN-Bert 模型图,第一阶段对软件需求文本数据集进行预处理,将输入句子的嵌入(第二阶段)与词汇图(第三阶段)进行结合,生成一个双重嵌入图,并将嵌入图连接到输入的句子嵌入后面(第四阶段)。第五阶段,在连接的表示中采用了多层多头自注意叠加的变压器模块,将单词嵌入与图嵌入之间进行交互。最后一层的嵌入被输入到一个完全连接的层(第六阶段)进行分类。

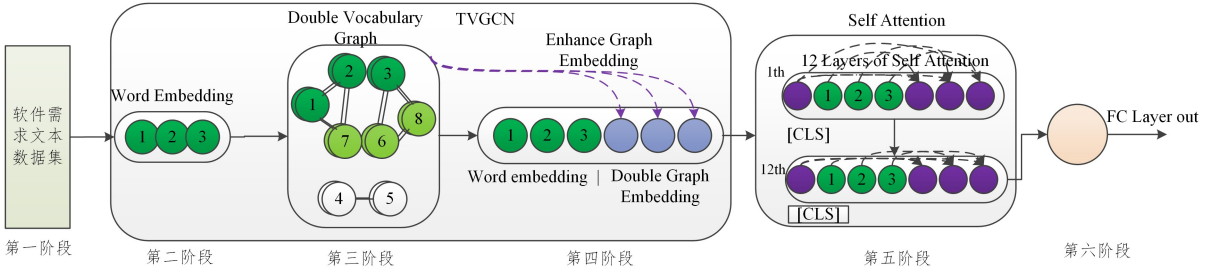


图1 TVGCN-Bert 模型图

Fig. 1 TVGCN-Bert model diagram

在一个语料库中,我们将软件需求文本数据构建成为一个包含单词节点和文档节点的大型异构文本图。本文特征矩阵 $\mathbf{X}=\mathbf{I}$ 设置为单位矩阵,这意味着每个单词或文档都被表示为一个向量,作为文本 GCN 的输入。我们基于文档中的单词出现(文档-单词边缘)和整个语料库中的单词共现(单词-单词边缘)来构建节点之间的边。文档节点和单词节点之间的边缘的权值是单词在文档中的词频-文档频率(TF-IDF)的反比。TF-IDF 计算公式如式(1)所示:

$$TF-IDF = \text{词频}(TF) \times \text{逆文档频率}(IDF) \quad (1)$$

其中,词频 TF 是该单词在文档中出现的次数,逆文档频率 IDF 是包含该单词的文档数量的对数比例的反比。

为了充分利用词汇的共现信息,我们对语料库中的所有文本使用固定大小的滑动窗口进行共现统计。这里的词汇表使用标准化的点相互信息(NPMI)构造,如式(2)所示。

$$NPMI(i, j) = -\frac{1}{\log p(i, j)} \log \frac{p(i, j)}{p(i)p(j)} \quad (2)$$

其中, i 和 j 是单词, $p(i, j) = \frac{\#W(i, j)}{\#W}$, $p(i) = \frac{\#W(i)}{\#W}$, $\#W(*)$ 是包含一个或一对单词的滑动窗口的数量, $\#W$ 是滑动窗口的总数。为了获得远程依赖,我们设置了窗口。对整个句子, NPMI 的取值范围为 $[-1, 1]$ 。NPMI 值为正,表示词语之间语义相关性高; NPMI 值为负,表示语义相关性低或不相关。如果两个单词的 NPMI 大于阈值,就在它们之间创建一个边界。GCN 是一个多层图神经网络,它直接在一个图上进行卷积,并根据其邻域的属性引出节点的嵌入向量。形式上,考虑一个图 $G=(V, E)$, 其中 $V(|V|=n)$ 和 E 分别是节点和边的集合。GCN 只需要一层图卷积就能捕获到相邻节点的信息。当多个 GCN 层堆叠时,则较大范围的信息被整合。对于 GCN 的单一卷积层,计算表达式如下: $\mathbf{H}=\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}$ 。其中, $\mathbf{X} \in R^{n \times m}$ 为特征的 n 个节点, m 个维度的输入矩阵; $\mathbf{W} \in R^{m \times h}$ 是一个权值矩阵; $\mathbf{A}=\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ 是标准化的对称邻接矩阵,其中 $D_{ii}=\sum_j A_{ij}^2$ 。这里对邻接矩阵 $\mathbf{A}$ 进行平方处理,可以获取到更多的相邻节点的信息。同时,对 $\mathbf{A}$ 的归一化操作,是为了避免数值不稳定性和在深度神经网络模型中的梯度消失或者梯度爆炸问题。

在本例中,我们的目标是对相关单词进行卷积,而不是对语料库中的文档进行分类。因此,我们提出的 GCN 图是在词汇表上构造的。假设文档是由词汇表中的单词组成的行向量 $\mathbf{X}$, 则定义一个卷积层,如式(3)所示:

$$\mathbf{h}=(\tilde{\mathbf{A}}\mathbf{x}^T)^T\mathbf{W}=\mathbf{x}\tilde{\mathbf{A}}\mathbf{W} \quad (3)$$

其中, $\tilde{\mathbf{A}}^T=\tilde{\mathbf{A}}$ 表示该词汇表。 $\tilde{\mathbf{A}}\mathbf{x}$ 提取与输入句子 a 相关的词汇图部分, $\mathbf{W}$ 持有单个文档的隐藏状态向量的权值,维数为

$|V| \times h$, 然后对一个小批的 k 个文档进行分类。式(3)中的单层图卷积为: $\mathbf{H}=\mathbf{X}\tilde{\mathbf{A}}\mathbf{W}$ 。而对应的带有 ReLU 函数的二层词汇 GCN 为:

$$TVGCN = \text{ReLU}(\mathbf{X}_{kv}\tilde{\mathbf{A}}_{uv}\mathbf{W}_{vh})\mathbf{W}_{hc} \quad (4)$$

其中, k 为小批量大小, v 为词汇量大小, h 为隐藏层大小, c 为类大小或句子嵌入大小。 $\mathbf{X}_{kv}$ 的每一行都是一个包含文档特征的向量,可以是一个词袋向量,也可以是 BERT 的多次嵌入。式(4)的目的是对图进行一层卷积,通过 $\mathbf{X}_{kv}\tilde{\mathbf{A}}_{uv}$ 捕获图中与输入相关的部分,然后执行 2 层的卷积,将输入句子中的单词与词汇图中与之相关的单词进行组合。通过对 GCN 的不同层数进行实验对比得出, GCN 在采用两层的情况下,效果是最好的。

本文借鉴最近的一些研究将 GCN 和 BERT 进行结合,利用 BERT 在海量语料的基础上运行自监督学习方法,为单词学习一个好的特征表示,在没有人工标注的数据上进行监督学习。在特定的 NLP 任务中,可以直接使用 BERT 的特征表示作为该任务的词嵌入特征。如图 1 所示,将 Bert 学习的一个词嵌入特征表示作为第二阶段的词嵌入向量,用 GCN 获取到相邻节点的信息并生成邻接矩阵,并对邻接矩阵进行平方(第三阶段),再将生成的嵌入图连接到输入的句子嵌入后面(第四阶段)。

Jong 等^[14]提出了另一种使用论文引文图的引文推荐任务组合。该模型只是将下游预测任务的 GCN 输出和 BERT 输出连接起来。当 BERT 应用于文本分类时,一个典型的解决方案包含 3 个部分。第一部分是包含单词位置信息的单词嵌入模块;第二部分是采用多层多头自注意叠加的变压器模块;第三部分是利用输出句嵌入进行分类的全连接层。下面通过交互层,可以将 GCN 中捕获的信息应用于输入文本,并将输入文本的表示传播到 GCN 上并运用在分类上。Self-attention 使用查询 $\mathbf{Q}$ 、操作键 $\mathbf{K}$ 和值 $\mathbf{V}$ 对进行计算。注意分数的计算方法如下:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \quad (5)$$

其中,分母是一个用于控制注意力评分尺度的比例因子, d_k 是查询和关键向量的维数。利用这些注意力分数,每个单词都可以得到一个编码上下文信息的加权向量表示。

我们不再只使用 BERT 中输入句子的单词嵌入,而是将由式(5)得到的词汇图嵌入和单词嵌入顺序输入到 BERT transformer 中。这样不仅保留了单词在句子中的顺序,而且利用了 TVGCN 获得的背景信息。通过式(6)计算出注意力分值,12 层 12 头自注意编码器通过层层交互将局部嵌入和全局嵌入充分融合。

$$G_{embedding} = ReLU(\mathbf{X}_{kv} \tilde{\mathbf{A}}_v \mathbf{W}_{vh}) \mathbf{W}_{hg} \tag{6}$$

其中,用于分类的 $\mathbf{W}_{vh}$ 作为图嵌入大小 g 的输出,其维数与每个字嵌入相同; k 为小批量尺寸; e 为单词嵌入的维数, v 为词汇量。

4 集成学习框架

4.1 集成模型设计

集成学习框架图如图 2 所示。集成模型将抓取的软件需求文本数据集和公开的 SST, COLA 数据集进行文本处理,并按 6:2:2 分成训练集、测试集、验证集 3 个数据集;再把训练集分成 5 个子训练集,用其中 3 个数据集创新性地对软件需求文本进行图建模,捕捉语言中单词的知识边以及单词之间

的全局关系,并进行图卷积神经网络分类。将输入句子的文本嵌入与词汇图结合,生成一个嵌入图,并将嵌入图连接到输入句子。注意,从词汇图中只提取与输入相关的部分并嵌入。在连接的表示中应用了多层自注意力机制,允许单词嵌入和图嵌入层次之间交互。最后一层的嵌入被馈入一个完全连接的层进行分类,同时与 Bert 融合。可在本例中,我们的目标是对相关单词进行卷积,而不是对语料库中的文档进行分类。因此,我们提出的 GCN 图是在词汇表上构造的,而不是在文档上构造的。对于单个文档,假设文档是由词汇表中的单词组成的行向量 $\mathbf{X}$,则定义一个卷积层,将输入句子中的单词与词汇图中与之相关的单词进行组合,以更好地捕捉图结构的信息,最后用全连接层进行分类。

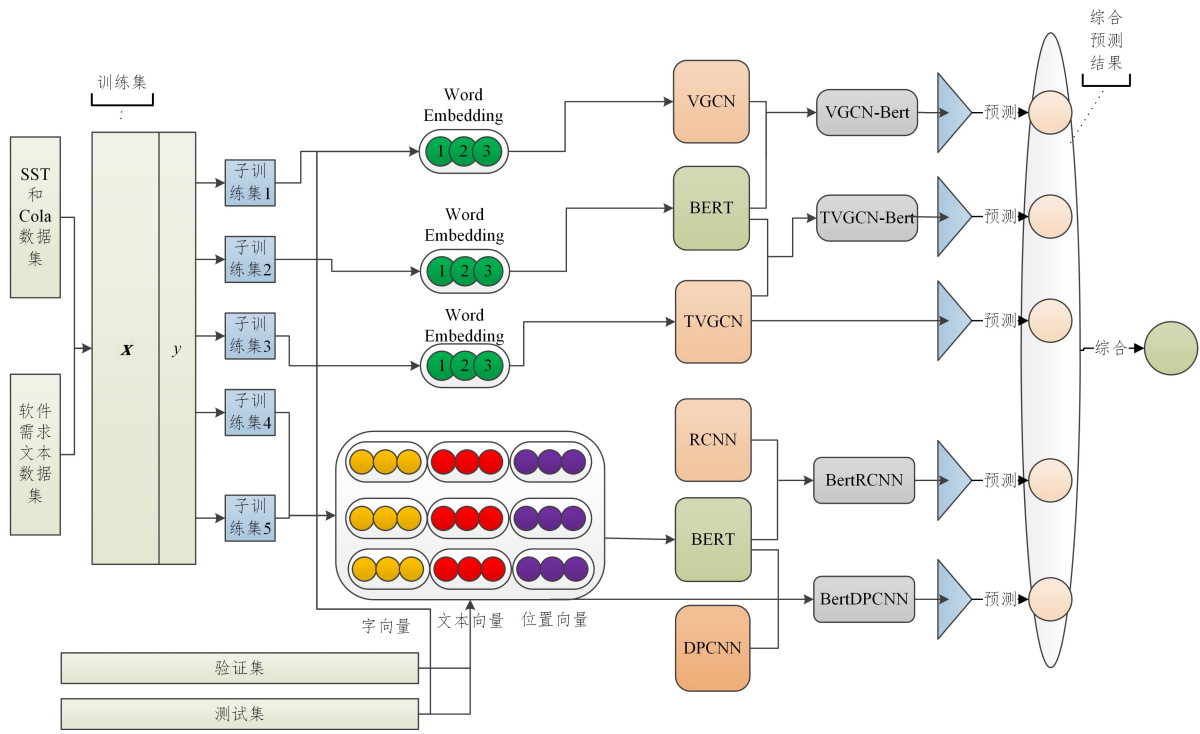


图 2 集成学习框架图

Fig. 2 Integrated learning framework diagram

最后,提出一种集成多种深度学习的软件需求文本分类方法,得出更优的分类效果。结果表明,模型融合对软件需求文本分类的准确率得到明显的提升,可以有效地判别软件需求文本的功能特性。

4.2 模型融合分类器

(1) 权重集合投票

考虑与分类器 C_j 相关的权重 W_j ,预测类别标签:

$$\hat{y} = \arg \max_i \sum_{j=1}^m w_j XA(C_j(x) = i) \tag{8}$$

其中, XA 为特征函数, A 为一组类标签, 计算公式如下:

$$[C_j(x) = i \in A] \tag{9}$$

(2) 权重集合投票的准确性

该方法的工作原理与加权精度相同,即用每个基分类器的精度代替权重。这里将 W_j 替换为精度,得:

$$Accuracy = \frac{\sum \text{True Positive} + \sum \text{True Negative}}{\sum \text{Total Population}} \tag{10}$$

本文采用的权重集合投票分类器,权重反映了每个分类

器的准确性,从而决定每种输入的最佳分类。具体来说,所提出的集成方法的意义在于所有基分类器中使用每个类别或标签的准确性。由于将总体准确性作为性能的绝对度量具有误导性,因此在提出的方法分类决策中没有使用该指标。此外,由于基分类器之间决策机制的差异,很难找到整个标注过程中最优的分类器。因此,该方法利用了每个基分类器的强度,更好地对每个软件需求文本数据进行分类;并利用堆叠泛化策略的交叉机制,在兼顾计算时间少和准确率高的前提下,将训练集分为 5 个等份进行 5 次交叉验证。这样做的优点是每次迭代过程中每个样本点只有 1 次被划入训练集或测试集的机会,易于找到模型泛化能力性能最优的参数值。在第 1 层中,将 TVGCN, VGCN-BERT, TVGCN-Bert, BertDPCNN, BertRCNN 等模型进行交叉验证,将数据集随机分为 10 份后,8 份作为训练,1 份作为验证,1 份作为测试,步骤重复 5 次,预测值刚好与训练数据长度吻合,得到的结果输入第 2 层,作为第 2 层模型的训练来源,由逻辑回归进行融合得到最终的结果。

4.3 模型融合策略设计

Bert 是一种 Transformer 的双向编码器,旨在通过在左右上下文中共有的条件计算来预先训练来自无标号文本的深度双向表示。因此,经过预先训练的 Bert 模型只需一个额外的输出层便可以进行微调,从而为各种自然语言处理任务生成最新模型。

4.3.1 BertRCNN

RCNN^[16]是一种结合了 RNN 和 CNN 的用于文本分类的模型。为了更准确地表达单词的意思,作者使用了单词本身和其上下文来共同表示这个词。在论文中,使用双向循环结构来实现。使用 $c_l(w_i)$ 来定义词 w_i 左边的文本, $c_r(w_i)$ 来定义词右边的文本。这里, $c_l(w_i)$ 和 $c_r(w_i)$ 是长度为 $|c|$ 的稠密向量。从框架结构图 2 中左边一块的箭头指向可以发现 $c_l(w_i)$ 和 $c_r(w_i)$ 的计算公式如下:

$$c_l(w_i) = f(W^{(l)} c_l(w_{i-1}) + W^{(sl)} e(w_{i-1})) \tag{11}$$

$$c_r(w_i) = f(W^{(r)} c_r(w_{i+1}) + W^{(sr)} e(w_{i+1})) \tag{12}$$

与 TextCNN 比较类似,两者都是把文本表示为一个嵌入矩阵,再进行卷积操作。不同的是:TextCNN 中的文本嵌入矩阵每一行只是文本中一个词的向量表示;而在 RCNN 中,文本嵌入矩阵的每一行是当前词的词向量以及上下文嵌入表示的拼接。CNN 使用固定词窗口(捕获上下文信息),实验结果受窗口大小的影响;RCNN 使用循环结构捕获广泛的上下文信息。在 RCNN 论文使用的实验数据集上,RCNN 比 CNN 好。

Bert 模型采用的是双向的 Transformer 结构进行编码,其中“双向”意味着模型在处理一个词时,其可以根据上下文的语义关系,表征字在上下文中的具体语义。预训练,是 Bert 模型的一个重要阶段,通过对海量语料进行训练,使得单词学习到很好的特征表示。通过 Bert 模型训练得到文本的向量表示。BertRCNN 将 Bert 模型与 RCNN 进行结合,实验效果好于前面几个模型的结合。

4.3.2 BertDPCNN

DPCNN^[17]是一种基于 word-level 级别的网络。TextCNN 不能通过卷积获得文本的长距离依赖关系;而 DPCNN 通过不断加深网络,可以抽取长距离的文本依赖关系。实验证明在不增加太多计算成本的情况下,增加网络深度就可以获得最佳的准确率。DPCNN 主要的特色是引入了残差结构,增加了多尺度信息,并且增加了用于文本分类 CNN 的网络深度,以提取文本中的远程关系特征,同时没有带来较高的复杂度。实验表明,其效果比普通 CNN 结构要好。BertDPCNN 是 Bert 模型和 DPCNN 模型的融合,将 Bert 生成的文本数据作为 DPCNN 模型的输入,并利用 DPCNN 模型进行文本分类。本文通过实验对比可发现 BertDPCNN 的模型效果比 BertCNN 模型效果好。

5 实验

5.1 数据集

本文主要使用的数据是软件的功能性需求文本和公开的两个数据集,软件需求文本数据来自一个英文网站:Softpedia。Softpedia 平台会为用户提供各式各样的软件工具,包括 Windows 软件、Windows 设备驱动程序、电子游戏和工具、Mac 应用程序、Android 应用程序、Linux 软件等几乎所有的

平台。Windows 软件类别有防毒软件、创作工具、CD/DVD/蓝光工具等几十类,同时该网站也会提供最新的新闻与评论。截至北京时间 2021 年 4 月 26 号,该网站总共收录应用程序 1 303 765 个,应用程序被下载总次数为 3 404 890 200 次。

Cola 数据集:语言可接受性语料库是一个二元单句分类任务。Cola 被手工标注为可接受性(语法性)^[18]。本文使用公共版本,包含 8551 个训练数据和 1043 个调查数据,总共有 6744 个阳性病例和 2850 个阴性病例。Cola 数据集的平均长度是 7.72 字。由于没有测试集的标签,我们将 5% 的训练集分割为验证集,并使用原始的验证集作为测试集。

SST-2 数据集:斯坦福情感树库是一个二元单句分类任务,由电影评论中提取的句子组成,且句子的情感由人工标注^[19]。本文使用公开版本 2,其中训练集包含 6920 个例子,验证集包含 872 个例子,测试集包含 1821 个例子,总共有 4963 个正面评论和 4650 个负面评论。评论的平均长度是 19.31 个单词。

Windows_a 数据集:Windows 软件类下的软件需求数据^[20],一共有 11 类,共 15568 条软件数据。具体的软件类型和数量如表 1 所列。其中,训练集包含 9337 个例子,验证集包含 3118 个例子,测试集包含 3113 个例子。软件需求文本的平均长度是 17.81 个单词。

表 1 Windows_a 数据集
Table 1 Windows_a dataset

软件类别	数量	软件类别	数量
Antivirus	625	Gaming-Related	114
Authoring-Tools	676	iPod-Tools	174
CD-DVD-Blu-ray-Tools	1545	Maps&-GPS	25
Compression-tools	466	Mobile-Phone-Tools	433
Desktop-Enhancements	7612	Network-Tools	3382
File-managers	546		

Windows_b 数据集:挑选 Windows 类下的软件需求数据进行爬取,一共爬取了 22 类软件数据,共 15568 条。爬取到的软件类型和数量如表 2 所列。其中,训练集包含 44126 个例子,验证集包含 14722 个例子,测试集包含 14712 个例子。软件需求文本的平均长度是 4.32 个单词。

表 2 Windows_b 数据集
Table 2 Windows_b dataset

软件类别	数量	软件类别	数量
Antivirus	589	Network Tools	3377
Authoring Tools	660	Office tools	8983
CD / DVD / Blu-ray Tools	1444	Others	10777
Compression tools	471	Portable Software	1118
Desktop Enhancements	7206	Programming	6890
File managers	572	Science / CAD	4283
Gaming Related	271	Security	4509
Internet	7983	System	9934
Maps / GPS	36	Tweak	2001
Mobile Phone Tools	452	Window Widgets	1800
Multimedia	28	iPod Tools	176

Linux 数据集:挑选 Linux 类下的软件需求数据进行爬取,一共爬取了 25 类,共 9744 条软件数据。爬取到的软件类型和数量如表 3 所列。其中,训练集包含 5839 个例子,验证集包含 1957 个例子,测试集包含 1948 个例子。软件需求文本的平均长度是 3.94 个单词。

表 3 Linux 数据集
Table3 Linux dataset

软件类别	数量	软件类别	数量
Adaptive Technologies	11	Multimedia	680
Adobe AIR Apps	30	Office	224
Artistic Software	17	Printing	65
Communications	312	Programming	444
Database	100	Religion	7
Desktop Environment	1435	Science	62
Documentation	50	Science and Engineering	268
Education	146	Security	199
Games	713	System	1320
Home Automation	17	Terminals	45
Information Management	114	Text Editing Processing	273
Internet	632	Utilities	617
Linux Distributions	1963		

Games 数据集:挑选 Games 类下的软件需求数据进行爬取,一共爬取了 11 类,共 25 337 条软件数据。爬取到的软件类型和数量如表 4 所列。其中,训练集包含 15 198 个例子,验证集包含 5 073 个例子,测试集包含 5 066 个例子。软件需求文本的平均长度是 5.42 个单词。

表 4 Games 数据集
Table 4 Games dataset

软件类别	数量	软件类别	数量
Cheat / Solutions	3622	Online Games Clients	361
Emu	76	Patch	1736
Free to Play	104	Server	59
Freeware Games	10745	Shareware Games	942
Games Demo	5511	Tools	889
Mods/ Addons	1292		

Mac 数据集:挑选 Mac 类下的软件需求数据进行爬取,一共爬取了 31 类,共 24 195 条软件数据。爬取到的软件类型和数量如表 5 所列。其中,训练集包含 14 504 个例子,验证集包含 4 852 个例子,测试集包含 4 839 个例子。软件需求文本的平均长度是 4.12 个单词。

表 5 Mac 数据集
Table 5 Mac dataset

软件类别	数量	软件类别	数量
Antivirus	40	Graphics	1879
Audio	1556	HTML Tools	56
Automator Actions & Workflows	80	Internet Utilities	2145
Business	529	Math/Scientific	2026
Communications	81	Multimedia	731
Compression Tools	54	Network/ Admin	389
DTP/Prepress	135	Screensavers	339
Dashboard Widgets	378	Security	361
Desktop	147	Spotlight Plugins & Utilities	13
Developer Tools	1578	System Utilities	876
Development	517	Utilities	5303
Drivers	1195	Video	651
Educational	547	Word Processing	363
Finance	173	iPhone— Applications	4
Font Tools	70	iPod Tools	37
Games	1942		

5.2 基线模型

我们对多个模型进行评估,并将该模型在 7 个数据集上

的基线模型和融合模型进行对比,以验证集成模型的效果。

VGCN:这个模型的输出维度为类的大小。Bert 的预训练词嵌入被用作输入。VGCN 输出中继到一个具有 Softmax 功能的全连接层,以产生分类评分。该模型仅使用词汇表图中的全局信息。

TVGCN:该模型对应于式(4),输入为邻接矩阵的平方,获取更多的图信息,并采用两层图卷积进行融合,输出中继到一个具有 Softmax 功能的全连接层,以产生分类评分。

Text-GCN:原始的 Text-GCN 模型使用了与 MLP 模型相同的输入特征,使用了与文献[21]中相同的训练参数。

BERT:使用小版本(Bert-base-uncase)预训练 BERT^[12]。

VGCN-Bert:该组合由 ReLU 和全连通层进行分类^[22]。这个模型的不同之处在于,不允许输入文本和图形之间进行交互。

TVGCN-Bert:该组合由改进版的 TVGCN 与 Bert 进行结合,再由 ReLU 和全连接层进行分类。

由于 Bert,CNN,LSTM,RNN,TextCNN 等基线模型在软件需求文本数据集上的提升效果不够大,我们创新性地将 Bert 与其他的深度模型融合后作为基线模型进行实验,以验证我们模型的有效性。下面对 3 个融合模型进行介绍。

(1)BertCNN

卷积神经网络最早被应用在计算机视觉中,也早已应用于自然语言处理的各种任务。在文本中,一句话所构成的词向量作为输入。每一行代表一个词的词向量,卷积核覆盖上下几行的词,卷积核的宽度与输入的宽度相同,通过这样的方式,我们就能够捕捉到多个连续词之间的特征,并且能够在同一类特征计算时中共享权重。接着对整个输出进行池化,得到一个相同维度的输出结果,用作最后的分类。而 TextCNN^[10]则是利用卷积神经网络对文本进行分类的算法,一个简单的 CNN,通过少量的超参数调整和静态向量,并引入已经训练好的词向量,在多个基准上取得了很好的结果。Bert 通过微调学习特定任务向量进一步提高性能。BertCNN 由 Bert 模型与 TextCNN 模型进行融合,利用 Bert 进行文本数据向量化,再把数据输入到 TextCNN 模型进行文本分类,可以很好地融合两个模型的优点,对本文的软件需求文本数据进行分类。

(2)BertLSTM

LSTM^[23]是具有记忆长短期信息能力的神经网络。LSTM 提出的动机是为了解决上面提到的长期依赖问题。传统的 RNN 节点输出仅由权值、偏置以及激活函数决定。LSTM 之所以能够解决 RNN 的长期依赖问题,是因为其引入了门(gate)机制用于控制特征的流通和损失。本文将 Bert 与 LSTM 进行结合,将 Bert 生成的向量表示输入到 LSTM 中,用于文本分类。通过实验对比可以发现,该模型的实验效果并没有得到较大的提升。

(3)BertRNN

尽管 TextCNN 在很多任务中有不错的表现,但 CNN 固定了 filter_size 的视野,这一方面无法建模更长的序列信息,另一方面 filter_size 的超参调节也很繁琐。CNN 本质是做文本的特征表达工作,而自然语言处理中更常用的递归神经网络,能够更好地表达上下文信息。具体在文本分类任务中,Bi-directional RNN(实际使用的是双向 LSTM)从某种意义上可以理解为可以捕获变长且双向的“n-gram”信息。BertRNN

由模型 TextRNN 与模型 Bert 结合。

5.3 预处理和设置

对不同的基线模型进行参数设置,由于 Bert 训练数据花费的时间较长,本文以 Bert-Base-Case 作为预训练模型。我们删除了文本内容只有数字或者空的行数,并将所有的标签转化为对应的数字进行表示,并使用 NLTK 的 Tweet Tokenizer 对文本进行小写和标记。使用拆分文本的 Bert Tokenizer 函数,使用的 GCN 的词汇表是预先训练好的 Bert 词汇表的子集,在数据集上计算 NPMI 时,使用整个句子作为文本窗口来构建词汇图。

将所有数据集的 NPMI 阈值设置为 0.2,以过滤出单词之间无意义的关系。在 TVGCN-Bert 模型中,将图嵌入输出大小设为 16,图嵌入的隐藏维数设为 128。我们使用 Bert-base-uncased 版本,并将最大序列长度设置为 200。该模型在 9 个 epoch 中进行训练,丢失率为 0.2。以下是针对不同数据集的其他参数设置。图 3 展示了不同 epoch 训练的效果,这里不同的模型设置的参数学习率不一致。

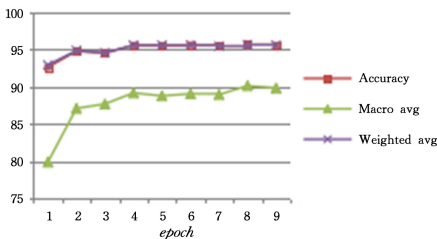


图 3 Windows_a 数据集在 TVGCN-Bert 不同 epoch 的效果
Fig. 3 Effects of Windows_a on TVGCN-Bert with different epoches

- (1)SST-2: mini-batch=16, 学习率=1×10⁻⁵, l2 减重衰减=0.01。
- (2)CoLA 和 Mac: mini-batch=16, 学习率=8×10⁻⁶, l2 损耗衰减=0.01。
- (3)Windows_a 和 Windows_b: mini-batch=16, 学习率=

- 1×10⁻⁵, l2 损耗衰减=1×10⁻³。
- (4)Linux 和 Games: mini-batch=12, 学习率=4×10⁻⁶, l2 损耗衰减=2×10⁻⁴。

这些参数是根据实验测试设置的。本文还使用缺省的微调学习率和文献[24]中的 l2 减重衰减。其他基线模型的参数设置与原始论文相同。

5.4 损失函数

使用平均平方误差作为损失函数:

W_classes = (#dataset) / (#classes × #everyclass) (13)

其中, #dataset 是数据集的总大小, #classes 是类的数量, #everyclass 是每个类的计数。

5.5 评价指标

我们采用两个最广泛使用的指标来评价分类器的性能——加权平均的 Weighted avg F1 和宏平均的 Macro F2[25]。

Weighted avg F1 = ∑(F1Ci × WCi) / ∑Ci (14)

Macro F2 = (1/C) × ∑F2Ci (15)

5.6 实验结果

表 6 列出 Bert 与多种模型融合的实验结果,由加权平均分和括号中的宏平均分两个指标进行评价。在 Windows_a 数据集上,Bert 的多种融合模型效果都不错,模型 BertRNN 的加权平均分最高,达到了 95.18%,宏平均分达到 90.59%;在 SST-2 数据集上,模型 BertCNN 的加权平均分达到了 91.65%,模型 BertRCNN 的加权平均分达到 91.49%;在 CoLa 数据集上,模型 BertRCNN 的加权平均分达到 80.87%;在 Windows_b 数据集上,模型 BertDPCNN 的加权平均分为 80.43%;在数据集 Linux 上,模型 BertRCNN 的加权平均分为 47.06%;在 Mac 数据集上,模型 BertRCNN 的加权平均分最高,为 43.02%;在 Games 数据集上,模型 BertATT 的加权平均分为 85.38%。

表 6 Bert 与多种模型融合实验效果
Table 6 Experimental results of Bert fusion with various models (单位: %)

Model	SST-2	Cola	Windows_a	Windows_b	Linux	Mac	Games
Bert	81.32	62.88(55.25)	95.13(90.00)	59.31(49.80)	45.08(20.15)	42.65(29.40)	84.44(61.57)
BertDPCNN	90.88	80.12(75.76)	94.87(69.43)	80.43(51.22)	31.05(7.9)	35.45(16.56)	83.92(51.92)
BertCNN	91.65	80.00(75.37)	95.09(90.50)	60.13(51.24)	46.58(24.53)	40.78(28.32)	84.67(62.42)
BertLSTM	91.27	56.51(40.87)	94.45(83.90)	59.04(48.99)	30.26(7.9)	41.52(28.18)	84.59(53.93)
BertATT	90.67	79.28(74.80)	94.72(89.59)	59.96(50.42)	45.68(22.33)	42.80(27.07)	85.38(62.99)
BertRCNN	91.49	80.87(76.70)	94.78(89.98)	59.56(49.70)	47.06(24.48)	43.02(31.15)	85.09(63.08)
BertRNN	90.28	80.12(75.76)	95.18(90.59)	59.50(49.24)	46.05(23.30)	42.48(34.13)	85.23(63.88)

表 7 列出 3 个不同的图卷积神经网络的实验结果。可以看出,模型 TVGCN 在其中 6 个数据集的加权平均分都比其他 2 个模型高,尤其在数据集 Windows_b 上,TVGCN 的提升效果最明显,加权平均分高于模型 VGCN 16.21%;在数据集 Windows_a 上,模型 Text-GCN 的效果最好,达到了 92.51%。

表 7 不同图卷积进行文本分类的实验结果
Table 7 Experimental results of text classification with different graph convolution (单位: %)

Model	SST-2	Cola	Windows_a	Windows_b	Linux	Mac	Games
Text-GCN	80.45	56.18(52.30)	92.51(82.06)	45.90(30.86)	31.46(11.53)	32.11(16.85)	69.24(41.94)
VGCN	81.64	63.59(54.82)	80.30(47.25)	60.84(49.44)	48.09(15.16)	51.93(13.48)	76.04(49.12)
TVGCN	81.78	63.69(55.20)	84.88(57.24)	77.05(55.08)	49.37(16.00)	53.26(18.77)	78.86(54.40)

表 8 列出图卷积与 Bert 融合的实验结果。模型 TVGCN-Bert 在其中 5 个数据集的加权平均分都比模型 VGCN-Bert

的结果好,在数据集 Games 上加权平均分提升了 1.82%,宏平均分提高了 4.11%。

表 8 图卷积与 Bert 融合的实验结果

Table 8 Experimental results of Bert and convolutional neural network fusion

(单位:%)

Model	SST-2	Cola	Windows_a	Windows_b	Linux	Mac	Games
VGCN-Bert	91.93	83.68(80.46)	94.94(70.58)	95.00(70.91)	76.81(44.03)	78.14(57.84)	84.46(68.44)
TVGCN-Bert	92.04	83.25(81.31)	96.33(91.20)	94.98(69.98)	77.21(44.36)	79.02(56.68)	86.28(72.55)

表 9 列出集成深度学习分类器的最佳方法的实验结果。集成模型 Integrated-Model 在 7 个数据集的加权平均分都好于每一个基模型,在数据集 Windows_a 上的加权平均分达

到了 96.73%,好于改进的单一模型 TVGCN-Bert。集成能够避免大部分的基分类器所产生的错误,其正确率高于其中任何一个基分类器。

表 9 集成模型的实验结果

Table 9 Experiment results of integrated models

(单位:%)

Model	SST-2	Cola	Windows_a	Windows_b	Linux	Mac	Games
TVGCN	81.78	63.69(55.20)	84.88(57.24)	77.05(55.08)	49.37(16.00)	53.26(18.77)	78.86(54.40)
VGCN-Bert	91.93	83.68(80.46)	94.94(70.58)	95.00(70.91)	76.81(44.03)	78.14(57.84)	84.46(68.44)
TVGCN-Bert	92.04	83.25(81.31)	96.33(91.20)	94.98(69.98)	77.21(44.36)	79.02(56.68)	86.28(72.55)
BertRCNN	91.49	80.87(76.70)	94.78(89.98)	59.56(49.70)	47.06(24.48)	43.02(31.15)	85.09(63.08)
BertDPCNN	90.28	80.12(75.76)	95.18(90.59)	59.50(49.24)	46.05(23.30)	42.48(34.13)	85.23(63.88)
Integrated-Model	92.34	84.38(80.96)	96.73(91.70)	95.60(71.91)	78.11(45.36)	79.82(57.78)	87.48(73.55)

5.7 实验结果可视化分析

5.7.1 VGCN 与 TVGCN

由图 4 可知,模型 TVGCN 在 SST-2 和 Cola 数据集上的性能提升不明显,在 Windows_a 和 Windows_b 数据集上性能提升明显。在 Windows_b 数据集上加权平均分提升了 16.21%,在 Windows_a 数据集上加权平均分提升了 4.58%,在其他软件需求数据集加权平均分也有相应的提高。

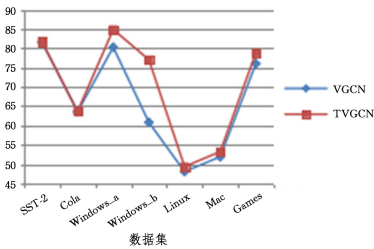


图 4 模型 VGCN 与 TVGCN 在不同数据集的加权平均分
Fig. 4 Weighted average fraction renderings of model VGCN and TVGCN in different datasets

5.7.2 VGCN-Bert 与 TVGCN-Bert

由图 5 可知,模型 TVGCN-Bert 在 SST-2 和 Cola 数据集上性能提升较少,在软件需求文本数据集上性能都有相应的提升,在 Windows_a 数据集上加权平均分提升了 1.39%,在 Games 数据集上加权平均分提升了 1.82%,在 Mac 数据集上加权平均分提升了 0.88%。

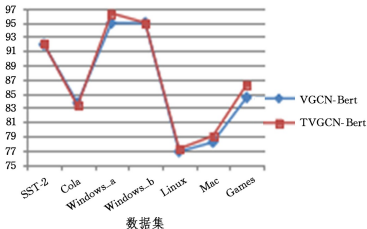


图 5 模型 VGCN-Bert 与 TVGCN-Bert 在不同数据集的加权平均分
Fig. 5 Weighted average fraction renderings of model VGCN-Bert and TVGCN-Bert in different datasets

5.7.3 融合 Bert 的不同模型

由图 6 可知,模型 BertDPCNN 在 Windows_b 数据集的效果好于其他模型 20% 以上,模型 BertRCNN 在 Linux 和 Mac 数据集上的效果好于其他模型。

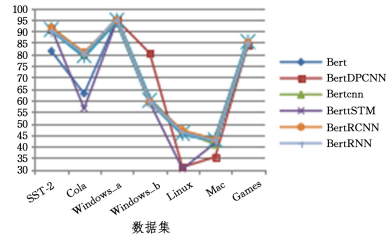


图 6 融合 Bert 的不同模型在不同数据集的加权平均分
Fig. 6 Weighted average score renderings of different models integrated Bert in different datasets

5.7.4 集成模型

由图 7 可知,集成模型在不同数据集的加权平均分都高于其他的基模型。采用集成模型,可以避免大部分的基分类器所产生的错误,其正确率高于其中任何一个基分类器。

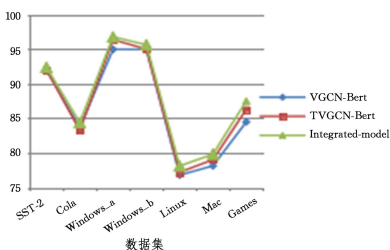


图 7 集成模型在不同数据集上的加权平均分
Fig. 7 Weighted average score of integred models in different datasets

结束语 需求分析阶段是整个软件生命周期的重要阶段,前期的需求分析直接决定后期工作的成败。本研究创新性地对软件需求文本进行图建模,建立软件需求的图神经网络,有效捕获单词的知识边以及单词和文本之间的关系,并引入了一个集成模型,利用每个基集成的每个类的精度来确定

5个基模型的最佳分类,提高加权投票集成的精度和可用性。该模型使用TVGCN,VGCN-BERT,TVGCN-Bert,BertDPC-NN,BertRCNN 5个分类器得到的准确率与集成模型相同,能有效判别软件需求文本的功能特性,提高软件需求文本分类的准确性。同时研究发现,所提模型对一些短文本数据的分类效果没有长文本的好,如Linux数据集的分类效果明显低于其他几个数据集,因此,如何对短文本数据进行更准确的分类将是未来工作的重点。

参 考 文 献

[1] ERNST N A,MYLOPOULOS J. On the perception of software quality requirements during the project lifecycle[C]//16th International Working Conference(REFSQ 2010). Springer Berlin Heidelberg,2010:143-157.

[2] NIU NEASTERBROOK S. Extracting and modeling product line functional requirements[C]//16th IEEE International Requirements Engineering Conference. 2008:155-164.

[3] KNAUSS E,DAMIAN D,POO-CAAMANO G,et al. Detecting and classifying patterns of requirements clarifications[J]. IEEE Computer Society,2012:251-260.

[4] KO Y,PARK S,SEO J,et al. Using classification techniques for informal requirements in the requirements analysis-supporting system[J]. Information & Software Technology,2007,49(11/12):1128-1140.

[5] RAHIMI N,EASSA F,ELREFAEI L. An Ensemble Machine Learning Technique for Functional Requirement Classification [J]. Symmetry,2020,12(10):1601.

[6] HU W S,YANG J F,ZHAO M. Demand analysis based on grey clustering algorithm [J]. Computer Science,2016,43(S1):471-475.

[7] MARTIN J,KLEINROCK L. Excerpts from:An Information Systems Manifesto [J]. Communications of the ACM,1985,28(3):252-255.

[8] ABAD Z,KARRAS O,GHAZI P,et al. What Works Better? A Study of Classifying Requirements[C]//2017 IEEE 25th International Requirements Engineering Conference. IEEE, 2017: 496-501.

[9] TIUN S,MOKHTAR U A,BAKAR S H,et al. Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text[J]. Journal of Physics:Conference Series,2020,1529(4):042077.

[10] KIM Y. Convolutional Neural Networks for Sentence Classification[J]. arXiv:1408.5882,2014.

[11] YAO L,MAO C,LUO Y. Graph Convolutional Networks for Text Classification[J]. Proceedings of the AAAI Conference on Artificial Intelligence,2019,33(1):7370-7377.

[12] DEVLIN J,CHANG M W,LEE K,et al. BERT:Pre-training of Deep Bidirectional Transformers for Language Understanding [J]. arXiv:1810.04805,2018.

[13] KIP F N,WELLING M. Semi-Supervised Classification with

Graph Convolutional Networks[J]. arXiv:1609.02907,2016.

[14] JEONG C,JANG S,SHIN H,et al. A Context-Aware Citation Recommendation Model with BERT and Graph Convolutional Networks[J]. arXiv:1903.06464,2019.

[15] RASCHKA S. Ensemble Vote Classifier-mlxtend [EB/OL]. http://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/.

[16] GIRSHICK R,DONAHUE J,DARRELL T,et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J]. arXiv:1311.2524,2013.

[17] JOHNSON R,TONG Z. Deep Pyramid Convolutional Neural Networks for Text Categorization[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017.

[18] WARSTADT A,SINGH A,BOWMAN S R. Neural network acceptability judgments[J]. arXiv:1805.12471,2018.

[19] SOCHER R,PERELYGIN A,WU J,et al. Recursive deep models for semantic compositionality over a sentiment treebank [C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing(EMNLP). 2013:1631-1642.

[20] KANG Y,CUI G R,LI H,et al. Software Requirements Clustering Algorithm Based on Self-attention Mechanism and Multi-channel Pyramid Convolution [J]. Computer Science, 2020, 47(3):48-53.

[21] YAO L,MAO C,LUO Y. Graph convolutional networks for text classification[J]. Proceedings of the AAAI Conference on Artificial Intelligence,2019,33(1):7370-7377.

[22] LU Z,DU P,NIE J Y. VGCN-BERT: augmenting BERT with graph embedding for text classification[C]//European Conference on Information Retrieval. Cham:Springer,2020:369-382.

[23] HOCHREITER,SEPP,SCHMIDHUBER,et al. Long short-term memory[J]. Neural Computation,1997,9(8):1735-1780.

[24] DEVLIN J,CHANG M W,LEE K,et al. :Bert Pre-training of deep bidirectional transformers for language understanding[J]. arXiv:1810.04805,2018.

[25] LEVER J,KRZYWINSKI M,ALTMAN N. Classification evaluation[J]. Nature Methods,2016,13(8):603-604.



KANG Yan, born in 1972, Ph.D, associate professor. Her main research interests include transfer learning, deep learning and integrated learning.



LI Hao, born in 1970, Ph.D, professor. His main research interests include distributed computing, grid and cloud computing.