doi:10.11835/j.issn.1000-582X.2021.213

强化文本关联语义和多特征提取的重复缺陷报告检测模型

周文杰,谢　琪,崔梦天
(西南民族大学 计算机系统国家民委重点实验室,成都 610041)

摘要:针对重复缺陷报告检测研究中存在语义长距离依赖以及缺陷报告特征的单一性问题,提出一种强化文本关联语义和多特征提取的重复缺陷报告检测模型。引入自注意力机制捕获缺陷报告文本序列内部的语义关联性,从而动态计算上下文语义向量进行语义分析,解决长距离依赖问题;利用隐含狄利克雷分布算法捕获缺陷报告文本的主题特征,同时针对缺陷报告的类别信息,构建一种特征提取网络计算类别差异特征;最后基于3类特征向量进行综合检测。实验结果表明,该模型实现了更优的检测性能。

关键词:重复缺陷报告检测;长距离依赖;自注意力机制;语义分析;多特征提取
中图分类号:TP311.5　　**文献标志码:**A　　**文章编号:**1000-582X(2023)07-053-10

A duplicate bug report detection model with enhanced text relevance semantics and multi-feature extraction

ZHOU Wenjie, *XIE Qi*, *CUI Mengtian*
(The Key Laboratory for Computer Systems of State Ethnic Affairs Commission, Southwest Minzu University, Chengdu 610041, P. R. China)

Abstract: A duplicate bug report detection model with enhanced text relevance semantics and multi-feature extraction was proposed to address the issues of semantic long-distance dependence and the singleness of bug report features in the current research on duplicate bug report detection. The model introduced the self-attention mechanism to capture the semantic relevance within the bug report text sequence. This mechanism calculates the contextual semantic vector dynamically for semantic analysis and resolves the problem of long-distance dependence. Additionally, the model employed the latent Dirichlet allocation algorithm to capture the topic characteristics of the bug report text. Furthermore, a feature extraction network was constructed to calculate category difference features, providing category information for the bug report simultaneously. Finally, comprehensive detection was performed based on three types of feature vectors. The experimental results

收稿日期:2021-05-31　　**网络出版日期:**2021-10-18
基金项目:国家自然科学基金资助项目(61502401,12050410248);四川省科技计划项目(2021YFH0120);西南民族大学中央高校基本科研业务费专项资金(2020YYXS59)。
Supported by National Natural Science Foundation of China(61502401, 12050410248), Sichuan Science and Technology Program(2021YFH0120), and Fundamental Research Funds for the Central Universities, Southwest Minzu University (2020YYXS59).
作者简介:周文杰(1997—),男,硕士研究生,主要从事自然语言处理研究,(E-mail)940554665@qq.com。
通信作者:谢琪,女,博士,副教授,(E-mail) qi.xie.swun@gmail.com。

demonstrate that the model achieves improved detection performance.

Keywords: duplicate bug report detection; long distance dependence; self-attention mechanism; semantic analysis; multiple features extraction

软件缺陷报告作为缺陷跟踪系统的实体之一,是描述软件缺陷、失效或不符合用户预期的软件表现的文档。软件缺陷报告由文本信息和类别信息组成,其中文本信息是以自然语言对软件缺陷进行描述,类别信息通过枚举值确定缺陷报告所属的类别。现实中由于不同人员之间信息不共享,往往针对同一个软件缺陷各自提交缺陷报告,从而导致了缺陷报告重复问题。据统计,Mozilla Core、Firefox和Eclipse Platform开源项目中含有的重复缺陷报告占比分别到达了21.8%、30.9%和16.9%[1]。分配重复的缺陷报告给不同的开发人员会导致人力资源的浪费,而自动化地检测重复缺陷报告将有效降低软件的维护成本。

目前对于重复缺陷报告检测已有大量研究。早期的研究主要是结合自然语言处理与文本检索技术的方面展开,这类研究的缺点在于难以解决同义不同词的匹配检索问题,导致检测性能较差[2]。为了解决这一问题,近期的研究转向利用神经网络捕捉文本语义,取得了良好的效果,但仍存在一些不足。这类研究普遍使用2种神经网络结构:卷积神经网络(convolutional neural network, CNN)和循环神经网络(recurrent neural network, RNN)。一方面,CNN网络擅长捕捉局部特征,但对全局上下文信息的归纳能力较差。RNN网络虽然能够对整个文本进行语义建模,但是随着文中长度的增加,存在语义长距离依赖的问题[3]。另一方面,不少研究往往局限于文本语义,缺乏挖掘缺陷报告文本所隐藏的主题模式,忽略或未充分利用缺陷报告的类别信息在检测中所起到的作用,因此存在特征单一性的问题。

为此,文中提出一种新的重复缺陷报告检测方法,通过构建一种基于自注意力机制[4]的神经网络模型,捕捉缺陷报告文本序列内部的相关性实现强化语义向量,克服语义长距离依赖问题。由于使用单一的特征向量不能全面反映缺陷报告之间的差异性,因此利用隐含狄利克雷分布(latent dirichlet allocation, LDA)算法挖掘缺陷报告的主题特征,同时构建一种差异特征提取网络,从而生成类别差异特征。由此,文中提出的模型不仅能更准确地度量语义相似性,还具有综合利用主题特征、类别差异特征进行多特征检测的能力。

1 相关研究

1.1 传统机器学习方法

早期对重复缺陷报告检测的主要手段是基于传统的机器学习技术。Runeson[5]最早提出应用自然语言处理技术将缺陷报告中包含的单词转换为词袋向量,然后在向量空间模型(vector space model)中计算余弦距离来对相似性进行度量。Sureke等[6]认为词袋模型完全忽视了自然语言文本的序列信息,提出一种基于n元文法的相似度量模型。Sun等[7]进一步指出余弦距离不能准确、真实地反映相似距离,提出引入SVM(support vector machine)作为分类器以提升检测性能。Yang等[8]认为应当差异化地对待有不同重要性的单词,提出通过BM25算法加权词袋向量。这些方法的确能够检测相当一部分的重复缺陷报告,但是它们的主要不足之处在于不能很好地检测以不同的术语书写但实质是描述相同的技术问题的重复缺陷报告报告,无法正确地处理同义词。

1.2 基于深度学习的方法

为了解决传统机器学习方法的不足,不少学者相继提出了一些基于深度学习方法的模型,这些模型直接研究缺陷报告文本语义信息。Kukkar等[9]使用预训练的词嵌入查找表得到文本的词向量,通过基于孪生结构的卷积神经网络捕捉2类语义特征,从而得到不同抽象层次的语义信息。由于该模型对每份缺陷报告进行独立编码,He等[10]提出一种多通道的卷积神经网络同时编码一组缺陷报告,以获得缺陷报告之间的语义关联,实现语义交互。针对卷积神经网络只能获得局部语义特征,Deshmukh等[11]进一步提出针对文本长度较长的文本域利用循环神经网络捕获基于全局上下文的语义表示,而卷积神经网络只用于捕捉短文本的语义,实验结果表明该模型取得了更好的效果。Prifti等[12]发现重复缺陷报告之间的提交时间间隔可以用来限制潜

在重复匹配的搜索空间。Poddar等[13]通过分步训练的方式，在进行主题聚类的同时利用Bi-GRU(bi-gated recurrent unit)网络进行语义编码，并引入注意力机制利用主题聚类的结果对语义特征向量加权，再输入到多层感知机里进行分类，实验结果表明经过加权后的语义向量具有更准确的表达能力。Rocha等[14]认为在进行相似性检测时缺少对缺陷报告语料库的综合分析，提出一种基于语料库主题特征的语义学习方法。

2　文中方法

重复缺陷报告检测的目标是针对当前缺陷报告，准确地从缺陷报告库返回与之重复的缺陷报告，实现篇章级检测。它的核心策略是迭代地计算缺陷报告之间的相似性程度，返回相似度最高的缺陷报告。

针对上述目标，文中提出的重复缺陷报告检测模型如图1所示，主要包括5个模块。其中数据预处理模块主要对缺陷报告的文本内容进行适当的转换和去除噪声数据，具体预处理步骤有分词、去除特殊字符、统一大小写和词形还原；语义特征提取模块主要是利用自注意力机制捕捉缺陷报告文本在各编码时间步上的语义内部相关性，实现语义聚合，得到上下文语义特征；主题特征提取模块对文本主题进行分析，得到主题特征向量；类别差异特征提取模块度量类别属性值之间的相似距离，得到类别差异特征向量；分类器根据这3类特征进行相似度计算，输出缺陷报告q和缺陷报告库中的任意缺陷报告p之间的相似程度，最后按相似程度降序排序返回相似度最高的缺陷报告。

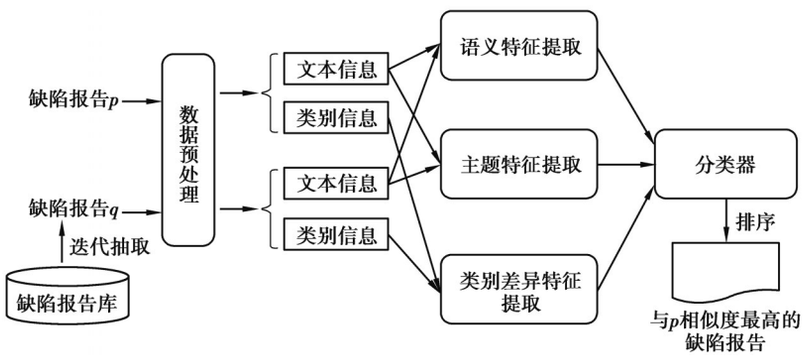

图1　总体框架图

Fig. 1　Overall framework

2.1　语义特征提取

2.1.1　输入层

该层的目标是将缺陷报告中的自然语言文本转换为神经网络能够处理的向量，具体而言是将缺陷报告中的title和description字段域中的文本所包含的每一个单词转换为一个固定维度的向量。基于Pennington等[15]的工作，该层使用Glove模型生成的预训练词嵌入查找表，对任意缺陷报告$\boldsymbol{q}$得到对应的词向量表示为

$$\boldsymbol{q}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\},\tag{1}$$

式中，N为缺陷报告$\boldsymbol{q}$中title和description文本字段域的文本的总长度。

对于词向量$\boldsymbol{x}_n\in\boldsymbol{q}$，它的形式为$[\boldsymbol{x}_{\text{glove}},x_{\text{field}}]\in\boldsymbol{R}^{1\times d}$，$d$是词向量维度。其中$\boldsymbol{x}_{\text{glove}}$是利用Glove模型得到的词嵌入向量，由于涉及到需要对来自2个不同字段域的信息进行处理，使用x_{field}作为是一个数值标识位，以表示该单词所归属的字段域，其值取0或1。$[\boldsymbol{x}_{\text{glove}},x_{\text{field}}]$符号表示将$\boldsymbol{x}_{\text{glove}}$向量扩充一个维度，由$x_{\text{field}}$的值进行填充。

2.1.2　初步编码层

理论上CNN和RNN神经网络都可用于编码缺陷报告的语义，但是CNN通过滑动窗口仅能捕捉局部特征，同时经典的RNN网络通常更适应短文本分析，而缺陷报告的title和description字段域中的文本总长度往往较长，容易导致在反向传播过程中的出现梯度消失的问题。为此，该层使用RNN的一种变种网络GRU模型对缺陷报告进行初步编码。GRU模型引入可学习的重置门控单元和更新门控单元控制输入信息的流动，从而捕获时间步距离较大的依赖关系，如图2所示。

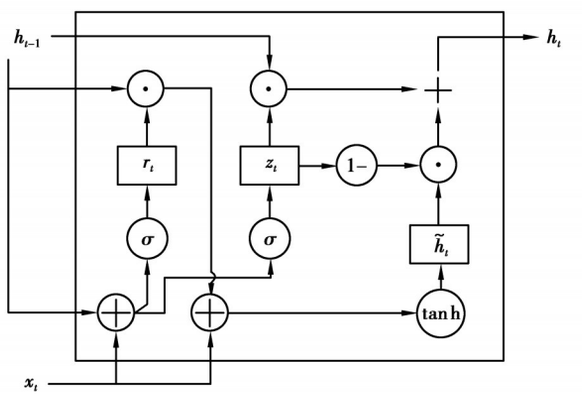

图2 门控循环单元(GRU)

Fig. 2 Gated recurrent unit(GRU)

在当前时间步t,GRU单元接收2个输入:前一时刻的隐状态$\boldsymbol{h}_{t-1}\in\boldsymbol{R}^{1\times d}$和当前时刻的词向量$\boldsymbol{x}_t$,然后分别计算重置门$\boldsymbol{r}_t$和更新门$\boldsymbol{z}_t$向量,分别决定上文环境信息哪些需要遗忘或继续传递,为

$$\boldsymbol{r}_t=\sigma((\boldsymbol{x}_t\oplus\boldsymbol{h}_{t-1})\times\boldsymbol{W}_r+\boldsymbol{b}_r)\in\boldsymbol{R}^{1\times d},\tag{2}$$

$$\boldsymbol{z}_t=\sigma((\boldsymbol{x}_t\oplus\boldsymbol{h}_{t-1})\times\boldsymbol{W}_z+\boldsymbol{b}_z)\in\boldsymbol{R}^{1\times d},\tag{3}$$

式中:⊕符号表示向量水平拼接;$\boldsymbol{W}_r\in\boldsymbol{R}^{2d\times d}$和$\boldsymbol{W}_z\in\boldsymbol{R}^{2d\times d}$均是可训练权重参数;$\boldsymbol{b}_r\in\boldsymbol{R}^{1\times d}$和$\boldsymbol{b}_z\in\boldsymbol{R}^{1\times d}$是偏置量;符号$\sigma$是sigmoid激活函数。

重置门向量$\boldsymbol{r}_t$被用来丢弃与当前输入无关的历史信息,然后和更新门向量$\boldsymbol{z}_t$共同计算得出当前时刻的隐状态$\boldsymbol{h}_t$,为

$$\tilde{\boldsymbol{h}}_t=\tanh(((\boldsymbol{r}_t\odot\boldsymbol{h}_{t-1})\oplus\boldsymbol{x}_t)\times\boldsymbol{W}_h+\boldsymbol{b}_h)\in\boldsymbol{R}^{1\times d}\tag{4}$$

$$\boldsymbol{h}_t=\boldsymbol{z}_t\odot\boldsymbol{h}_{t-1}+(1-\boldsymbol{z}_t)\odot\tilde{\boldsymbol{h}}_t\in\boldsymbol{R}^{1\times d}\tag{5}$$

式中:⊙符号表示逐元素相乘;$\boldsymbol{W}_h\in\boldsymbol{R}^{2d\times d}$和$\boldsymbol{b}_h\in\boldsymbol{R}^{1\times d}$同样分别是可训练权重参数和偏置量;tanh为激活函数。

将缺陷报告q的词向量序列$\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$前向依次传入到GRU中,即可生成相应的隐状态序列$\{\vec{\boldsymbol{h}}_1,\vec{\boldsymbol{h}}_2,\cdots,\vec{\boldsymbol{h}}_N\}$。由于前向迭代只能捕捉上文环境信息,为了同时能够捕捉下文环境信息,再将词向量序列后向依次传入到GRU中得到后向隐状态序列$\{\overleftarrow{\boldsymbol{h}}_1,\overleftarrow{\boldsymbol{h}}_2,\cdots,\overleftarrow{\boldsymbol{h}}_N\}$。最终通过向量水平拼接生成每一个时间步的综合隐状态,其形式为

$$H=\{\overleftrightarrow{\boldsymbol{h}}_1,\overleftrightarrow{\boldsymbol{h}}_2,\cdots,\overleftrightarrow{\boldsymbol{h}}_N\}。\tag{6}$$

对任意$\overleftrightarrow{\boldsymbol{h}}_n\in H$,它的计算方法为

$$\overleftrightarrow{\boldsymbol{h}}_n=\vec{\boldsymbol{h}}_n\oplus\overleftarrow{\boldsymbol{h}}_n\in\boldsymbol{R}^{1\times 2d}。\tag{7}$$

2.1.3 自注意力加权

GRU模型引入2类门控单元优化语义分析过程,输出的隐状态序列H中的每一个成员向量理论上融合了当前时间步的全局上下文信息,因此理论上H的最后一个序列成员$\overleftrightarrow{\boldsymbol{h}}_N$应当表征了整个缺陷报告的综合语义。然而当文本的长度过长时,较远时间步所生成的隐状态由于迭代次数的增多,对当前时间步的隐状态的影响越来越小。因此对于$\overleftrightarrow{\boldsymbol{h}}_N$来说,实际上几乎只表征了临近几个单词所构成的上下文语义,即存在长距离依赖的问题。为了克服这一问题同时提升模型的语义表征能力,引入自注意力机制,分析其他时间步的隐藏状态和当前时间步的隐状态的交互信息,如图3所示。

自注意力机制可以自动地发现隐状态序列内部的关联关系,得到相应的注意力权重,能够聚合向量,从而达到强化上下文语义向量的目的,为

$$C_j=\boldsymbol{v}_j^T\ \mathrm{ReLu}((\overleftrightarrow{\boldsymbol{h}}_N\oplus\overleftrightarrow{\boldsymbol{h}}_j)\times\boldsymbol{W}_j+\boldsymbol{b}_j),\tag{8}$$

$$c_j=\exp(C_j)/\sum_{i=1}^{N}\exp(C_i),\tag{9}$$

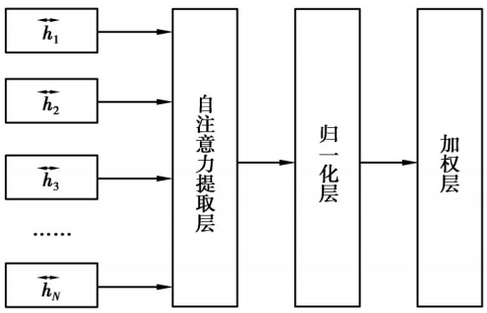

图 3　自注意力模块

Fig. 3　Self-attention module

$$\boldsymbol{v}_{\text{text}}=\sum_{j=1}^{N} c_j \overleftrightarrow{\boldsymbol{h}}_j \in \boldsymbol{R}^{1\times 2d}, \tag{10}$$

式中：$\overleftrightarrow{\boldsymbol{h}}_j \in \boldsymbol{R}^{1\times 2d}$表示在时间步$j$生成的隐状态向量；$\boldsymbol{W}_j \in \boldsymbol{R}^{4d\times e}$和$\boldsymbol{v}_j \in \boldsymbol{R}^{1\times e}$是可训练参数；$\boldsymbol{b}_j \in \boldsymbol{R}^{1\times e}$是偏置量；$C_j$表示第$j$个隐状态$\overleftrightarrow{\boldsymbol{h}}_j$与$\overleftrightarrow{\boldsymbol{h}}_N$进行动态关联的注意力得分；$c_j$是进行归一化后的分数；$\boldsymbol{v}_{\text{text}}$是经过对各个隐状态加权后最终得到的综合语义特征向量；ReLu是激活函数；exp是绝对值函数。

自注意力机制通过式(8)定义自注意力提取层，该层挖掘任意时间步j的上下文语义与$\overleftrightarrow{\boldsymbol{h}}_N$之间的交互关联关系，经式(9)进行归一化运算后，加权每一时间步的上下文语义，得到综合语义特征向量$\boldsymbol{v}_{\text{text}}$。相较于$\overleftrightarrow{\boldsymbol{h}}_N$特征向量，$\boldsymbol{v}_{\text{text}}$由于融合了其他较远时间步的上下文语义，因此能够表征更加丰富的全局语义信息，克服语义长距离依赖问题。值得注意的是，当$c_1,c_2,\cdots,c_{N-1}=0$时，最终的语义向量退化为$\overleftrightarrow{\boldsymbol{h}}_N$。经过实验发现并不存在这种情况，这恰恰说明引入自注意力机制的必要性。利用上述语义特征提取方法，分别得到缺陷报告q和p的上下文语义特征向量为$\boldsymbol{v}_{\text{text}}^{p}$与$\boldsymbol{v}_{\text{text}}^{q}$。

2.2　主题特征提取

语义特征提取将缺陷报告所包含的信息通过深度学习映射到一个语义空间中，这一过程可以理解为一种语义归纳。缺陷报告的内容也可以被看作是反映一个或多个技术问题的表述，如软件适配错误、网络异常等。因此，有必要引入LDA模型对缺陷报告的主题分布进行推断。LDA模型是一种无监督的生成模型，如图4所示。

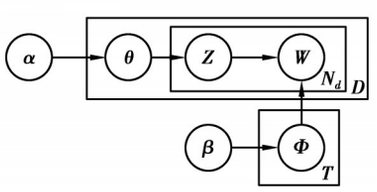

图 4　LDA的图过程

Fig. 4　The graph process of the LDA

图4中，“文档-主题”概率分布θ和“主题-词”概率分布Φ是LDA主题模型的核心，分别由可学习的狄利克雷参数α和β所决定；矩形表示重复抽样，N_d、D和T表示抽样次数，其值分别为缺陷报告q中含有的单词数量、缺陷报告库中含有的文档数量和主题个数。整个过程为：当从“文档-主题”概率分布θ中确定一个主题Z，再结合该主题所对应的“主题-词”概率分布Φ，即可确定一个单词W。由于在实际中除了缺陷报告中的单词可被观测和主题个数T可被指定，其他参数均未知，因此需要通过联合概率函数进行推断，联合概率函数为

$$P(\theta,Z,W|\alpha,\beta)=(\theta|\alpha)\prod_{n=1}^{N}P(Z_n|\theta)\,P(W_n|Z_n,\beta)。 \tag{11}$$

通过Gibbs采样算法使得联合概率函数收敛，即可得到“文档-主题”概率分布 和“主题-词”概率分布，有

关具体实现细节可查文献[16]。通过对缺陷报告 q 和 p 的文本主题进行推断，得到主题特征向量 $\boldsymbol{v}^{p}_{\text{topic}} \in \boldsymbol{R}^{1\times T}$ 和 $\boldsymbol{v}^{q}_{\text{topic}} \in \boldsymbol{R}^{1\times T}$，其中 T 为超参，表示主题个数。

2.3 类别差异特征提取

与一般的文本相似性度量任务不同，缺陷报告不仅包括由自然语言构成的文本域信息，还包含一些其他枚举类型数据，例如 bug severity、priority 和 component 等类别信息。这些分类信息无法提取出语义或主题，但是它们对重复缺陷报告检测的作用不可忽视。为此，文中提出一种类别差异特征提取方法，如图 5 所示。

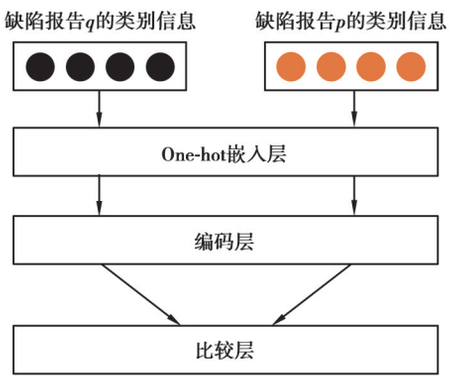

图 5 类别差异特征提取模块

Fig. 5 Category difference feature extraction module

采用 One-hot 方法分别对类别信息的每一种枚举值进行映射。由于 bug severity、priority 和 component 类别属性分别有 5、4 和 7 种枚举值，因此对于任意一个缺陷报告都将被转换为一个 16 维度的向量，其中 3 个维度为 1，剩余维度为 0。依次通过编码层和比较层得到类别差异特征，为

$$\boldsymbol{a}_q = \text{ReLu}(\boldsymbol{e}_q\boldsymbol{W}_a + \boldsymbol{b}_a) \in \boldsymbol{R}^{1\times u}, \tag{12}$$

$$\boldsymbol{a}_p = \text{ReLu}(\boldsymbol{e}_p\boldsymbol{W}_a + \boldsymbol{b}_a) \in \boldsymbol{R}^{1\times u}, \tag{13}$$

$$\boldsymbol{v}^{q,p}_{\text{cat}} = \text{ReLu}((\boldsymbol{a}_q - \boldsymbol{a}_p)^2\boldsymbol{W}_{c1} + (\boldsymbol{a}_q \odot \boldsymbol{a}_p)\boldsymbol{W}_{c2} + \boldsymbol{b}_c) \in \boldsymbol{R}^{1\times u}, \tag{14}$$

式中：$\odot$ 仍然表示向量逐元素相乘；$\boldsymbol{e}_q \in \boldsymbol{R}^{1\times 16}$ 和 $\boldsymbol{e}_p \in \boldsymbol{R}^{1\times 16}$ 分别是待检测的缺陷报告 q 和 p 的 One-hot 向量；$\boldsymbol{W}_a \in \boldsymbol{R}^{16\times u}$、$\boldsymbol{W}_{c1} \in \mathbf{R}^{u\times u}$ 和 $\boldsymbol{W}_{c2} \in \boldsymbol{R}^{u\times u}$ 是可训练参数；$\boldsymbol{b}_a \in \boldsymbol{R}^{1\times u}$ 和 $\boldsymbol{b}_c \in \boldsymbol{R}^{1\times u}$ 是偏置量；u 是超参；控制编码向量和类别差异特征向量 $\boldsymbol{v}^{q,p}_{\text{cat}}$ 的维度。

值得注意的是，理论上 $\boldsymbol{e}_q$ 和 $\boldsymbol{e}_p$ 可以直接传入比较层，但这种方式无法有效地刻画枚举值之间的差异。以 bug severity 类别属性为例，它的枚举值有 low、medium、high、critical 和 catastrophic 五种。假设缺陷报告 q 的 bug severity 属性为 low，对应的 One-hot 向量为[1,0,0,0,0]，不论缺陷报告 p 的 One-hot 向量是[0,1,0,0,0]还是[0,0,0,0,1]，如果没有编码层，那么它们在比较层的输出结果是一致的，均为零向量。但事实上，当缺陷报告 p 的 bug severity 类别属性值为 medium 比 catastrophic 应当更接近缺陷报告 q。因此加入编码层的意义就在于学习同一类别属性值之间的差异关系。

2.4 分类器

分类器具体由一个全连接层和分类层组成，全连接层的输入包括 3 类：上下文语义特征、主题特征和类别差异特征，分类器则是一个标准的逻辑回归，预测缺陷报告 q 和 p 之间的相似性。分类器定义为

$$\boldsymbol{v}^{q,p}_{\text{all}} = \text{ReLu}((\boldsymbol{v}^{q}_{\text{text}} \oplus \boldsymbol{v}^{p}_{\text{text}})\boldsymbol{W}_{t1} + (\boldsymbol{v}^{q}_{\text{topic}} \oplus \boldsymbol{v}^{p}_{\text{topic}})\boldsymbol{W}_{t2} + \boldsymbol{v}^{q,p}_{\text{cat}}\boldsymbol{W}_{t3} + \boldsymbol{b}_t) \in \boldsymbol{R}^{1\times m}, \tag{15}$$

$$P(y|q,p) = \text{sigmoid}(\boldsymbol{v}^{q,p}_{\text{all}}\boldsymbol{W}_s + \boldsymbol{b}_s), \tag{16}$$

式中：$\boldsymbol{W}_{t1} \in \boldsymbol{R}^{4d\times m}$、$\boldsymbol{W}_{t2} \in \boldsymbol{R}^{2k\times m}$、$\boldsymbol{W}_{t3} \in \boldsymbol{R}^{u\times m}$ 和 $\boldsymbol{W}_s \in \boldsymbol{R}^{m\times 1}$ 是可训练参数；$\boldsymbol{b}_t \in \boldsymbol{R}^{1\times m}$ 和 $\boldsymbol{b}_s \in \boldsymbol{R}$ 是偏置量。

3 实验与结果分析

3.1 实验设计

3.1.1 数据集

为了验证文中所提方法的有效性，采用 Eclipse 公开的缺陷报告数据集[17]进行实验。该数据集中共含有

25 423个软件缺陷报告文档，其中被标记为重复的缺陷报告共有15 423对。每个缺陷报告由14个字段域组成，文中所提出的模型仅需使用到其中7个字段域即可，分别是bug_id、title、description、dup_id、bug severity、priority和component。字段域bug_id可唯一地标识该缺陷报告；字段域title和description分别表示缺陷的总体描述和详细描述；字段域dup_id表示当前缺陷报告是否含有与之重复的其他缺陷报告，如果有则字段域的值为对应的bug_id，否则就缺省；字段域bug severity表示缺陷的严重程度；字段域priority表示缺陷的处理优先级，字段域component表示缺陷所属的功能组件。

3.1.2　损失函数

文中所提出的模型每次同时处理任意一组缺陷报告q和p，并输出重复概率$P(y|q,p)$，同时记缺陷报告q和p的标签为y_{label}，定义交叉熵损失函数为

$$J_{\text{loss}}=-\log(y_{\text{label}}\log(P(y|q,p))+(1-y_{\text{label}})\log(P(y|q,p))+\frac{\lambda}{2}\|W\|_2^2, \tag{17}$$

式中：λ和W分别是正则化参数和模型权重；表示对损失函数进行L_2正则化以缓解过拟合问题。

3.1.3　评价指标

由于重复缺陷报告检测的实质是从缺陷报告库已有的缺陷报告中选出与当前缺陷报告相似度最高的一份或几份，并按照相似度生成一个有序列表，实际上是一种排序任务。文中采用平均精度均值（mean average precision，MAP）和k召回率（recall rate@k，RR@k）作为评估模型性能的指标，分别记为E_{MAP}和$E_{\text{RR@k}}$。E_{MAP}的计算方法为

$$E_{\text{MAP}}=\frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{p_i}, \tag{18}$$

式中：Q是测试集中的样本个数；p_i是第i个样本所记录的重复缺陷报告在生成的有序列表中的位置。$E_{\text{RR@k}}$的计算方法为

$$E_{\text{RR@k}}=\frac{n_k}{Q}, \tag{19}$$

式中，n_k表示有序列表的前k项包含重复缺陷报告的样本的个数。

3.1.4　参数设置

在实验中，是使用Natural Language Tool Kit工具对缺陷报告文本进行数据预处理。模型训练过程中，参数设置如表1所示。

表1　参数设置

Table 1　Parameter settings

超参数	值	超参数	值
文本最大长度	200	批处理大小	32
词向量维度	100	学习率	0.000 1
GRU单元堆叠层数	1	正则化参数	8
GRU隐状态维度	100	失活率	0.2
主题数	10	优化器算法	Adam
类别差异特征向量维度	10		

3.2　实验结果分析

本节将从以下3个方面加以阐述：1）与其他模型相比，文中所提出的模型是否表现出更优的性能；2）各个模块对模型性能的影响；3）模型在不同参数下的性能。

3.2.1　不同模型的对比实验分析

为验证文中所提出的重复缺陷报告检测模型的有效性，在Eclipse公开缺陷报告数据集上进行实验，并与REP、DWEN和BSO模型进行对比。相关模型在数据集上的实验结果如表2所示。

从表2中可以看出，DWEN[18]模型的性能最差，原因在于DWEN模型仅仅计算缺陷报告的所有词向量在各个维度上的均值，组成语义特征向量，同时也没有将类别信息作为影响相似性的因素。该语义特征构造过

程无法有效区分单词的重要程度和对文本语义的贡献,且后期单词输入序列所表达的上下文语义覆盖了前期上下文语义,存在语义长距离依赖问题。而文中所提出的模型由于引入自注意力机制,通过注意力权重计算,动态决定每一个时间步的上下文语义对全局上下文语义的贡献,实现了语义聚合。

表2 实验结果(1)

Table 2 Experimental results(1) %

模型	$E_{RR@1}$	$E_{RR@2}$	$E_{RR@3}$	$E_{RR@4}$	$E_{RR@5}$	E_{MAP}
REP	35.12	44.25	48.92	52.32	54.29	42.61
DWEN	25.71	30.58	35.84	37.86	40.68	38.95
BSO	36.15	46.52	52.16	53.89	56.21	45.48
文中模型	37.29	47.95	53.84	55.61	57.53	47.25

REP[7]模型针对缺陷报告具有结构化的特征,利用文本域权重加权域内单词的重要度,实现了粗粒度地区分不同单词的重要性,同时对比7种类别属性值组成类别特征,再结合文本特征后进行综合检测,因此该模型取得了明显优于DWEN模型的检测性能,性能提升的原因主要在于REP模型充分利用了缺陷报告的类别信息,整体地反映了缺陷报告之间的相似性。但是REP模型由于缺少语义分析,因此难以解决同义不同词的问题,检测性能依然弱于文中所提出的模型。

BSO[19]模型在REP的基础上提出再对缺陷报告文本对应的词向量矩阵进行奇异值分解,得出主题特征。与文中所提出的模型相比,BSO模型同样实现了从3个方面对缺陷报告文档进行建模,但其使用词袋模型得到的词向量表示非常稀疏,同时在语义归纳过程中忽略了句法结构信息,语义提取不够准确。而文中所提出的模型既具有对缺陷报告进行多特征抽取的能力,又通过自注意力机制强化全局上下文语义表示,使得语义提取更为准确,因此检测性能最好,在评价指标$E_{RR@1-5}$上比第二优的BSO模型分别高出3.15%、3.07%、3.22%、3.19%、2.34%,E_{MAP}值高出3.89%。

3.2.2 各个模块性能实验分析

为了验证文中所提出的模型中不同模块对模型性能的影响,本组实验设计一个基准模型,该模型以不具备自注意力加权的语义特征作为分类器的唯一输入向量,分别针对自注意力加权模块、主题特征提取模块和类别差异特征提取模块对实验结果的影响进行验证。实验结果见表3所示。

表3 实验结果(2)

Table 3 Experimental results(2) %

模型	$E_{RR@1}$	$E_{RR@2}$	$E_{RR@3}$	$E_{RR@4}$	$E_{RR@5}$	E_{MAP}
Baseline	27.58	31.49	37.67	40.26	42.31	40.03
+自注意力机制	32.49	41.62	46.28	50.24	52.03	44.29
+主题特征	28.84	33.16	38.49	42.55	44.03	41.24
+类别差异特征	29.85	35.28	40.29	44.62	46.09	42.05
文中模型	37.29	47.95	53.84	55.61	57.53	47.25

从表3中可以看出,3个模块对实验的性能均有提升的作用。以$E_{RR@1}$为例,引入自注意力机制、提取主题特征和类别差异特征分别较基线模型提升17.80%、4.57%和8.23%。从各模块对模型性能的提升幅度来看,引入自注意力机制对语义向量进行加权对模型性能影响最大,在所有评价指标上提升最多。

值得注意的是,在与表2数据对比之后可以发现,引入自注意力机制尽管使得模型性能大幅上升,但检测性能依然弱于REP和BSO这2类方法。原因在于基于基线模型上引入自注意力机制,依然局限于语义分析这一个方面,但是与同为仅有语义分析的DWEN模型相比,相关数据清晰地表明引入自注意力机制后模型实现了更为准确的语义分析。

相较于主题特征,引入类别差异特征对模型能力的提升能力更强,原因在于类别信息由枚举值进行清晰

刻画,相较于对文本信息进行主题分析,后者更容易受噪声信息的影响。纵向地对比各模块对模型性能提升作用可以发现,尽管引入自注意力机制的绝对提升幅度最大,但主题特征和类别差异特征能从其他侧面刻画语义分析难以挖掘的其他特征模式,从而使得模型检测性能进一步提升。因此通过3种模块进行基于多特征检测的策略达到了最佳的检测效果,在各个评价指标上依次提高了35.21%、52.27%、42.92%、38.12%、35.97%和18.04%。

3.2.3　不同参数下的实验性能分析

本节对模型中涉及的一些参数设置对实验性能的影响进行分析。

1)向量维度对模型性能的影响。

在语义特征提取中有2种重要的向量:作为输入的词向量和决定输出的隐状态向量。在训练中,它们同时具有相同的维度。为了分析这2类向量维度对实验结果的影响,本组实验设计向量维度分别为25维、50维、100维和200维的4组实验。实验结果见表4所示。

表4　实验结果(3)

Table 4　Experimental results(3)　%

向量维度	$E_{RR@1}$	$E_{RR@2}$	$E_{RR@3}$	$E_{RR@4}$	$E_{RR@5}$	E_{MAP}
25	36.54	46.14	51.73	52.11	54.19	45.35
50	37.08	47.12	52.42	53.33	55.27	46.19
100	37.29	47.95	53.84	55.61	57.53	47.25
200	36.89	47.03	52.97	54.20	55.79	46.25

由表4可以看出,随着向量维度从25维增加到50维再到100维,模型的性能不断提升,当增长到200维时,模型性能出现下降。实验结果说明,在一定范围内随着维度的增加,向量越能准确反映词的特征和提升隐状态的上下文信息描述能力,而当维度过大时,反而弱化了词之间的差异,隐状态也包含了冗余的上下文信息导致模型性能下降。

2)失活率对模型性能的影响。

使用失活率是一种有效减缓模型过拟合的方法,本组实验为了分析不同失活率下模型的性能,设计失活率由0.05增长至0.25的多次实验。实验结果见表5所示。

表5　实验结果(4)

Table 5　Experimental results(4)　%

失活率	$E_{RR@1}$	$E_{RR@2}$	$E_{RR@3}$	$E_{RR@4}$	$E_{RR@5}$	E_{MAP}
0.05	36.82	46.15	53.25	54.95	56.26	45.07
0.10	37.29	47.95	53.84	55.61	57.53	47.25
0.15	35.10	45.05	52.49	53.13	55.39	45.05
0.20	32.91	41.86	49.30	50.68	52.50	42.19
0.25	29.84	37.62	45.62	46.29	49.90	40.25

从表5中可以看出,当失活率为0.1时模型在各个指标上均达到最优。实验结果说明失活率的确起到了防止过拟合的作用,但是随着失活率的逐渐增大,本质上等同于逐步减少模型的有效神经元数量,因此损害了模型的性能。

4　结束语

提出了一种强化文本关联语义和多特征提取的重复缺陷报告检测模型,该模型包括基于双向门控循环单元的语义特征提取、基于隐含狄利克雷分布的文本主题特征挖掘以及基于全连接神经网络的类别差异信

息度量。在语义特征提取中,还通过引入自注意力机制,评估跨时间步之间的上下语义之间的相关性,实现语义聚合以克服语义长距离依赖问题。实验结果表明,该模型在平均精度均值和召回率上优于近期提出的其他模型。在下一步的工作中,将深入地研究获得更准确的缺陷报告语义向量表征,此外,将对缺陷报告之间的语义关联关系进行探索。

参考文献

[1] Xie Q, Wen Z Y, Zhu J M, et al. Detecting duplicate bug reports with convolutional neural networks[C]//2018 25th Asia-Pacific Software Engineering Conference(APSEC). IEEE, 2018:416-425.

[2] Zou W Q, Lo D, Chen Z Y, et al. How practitioners perceive automated bug report management techniques[J]. IEEE Transactions on Software Engineering, 2020, 46(8):836-862.

[3] Altınel B, Ganiz M C. Semantic text classification: a survey of past and recent advances[J]. Information Processing & Management, 2018, 54(6):1129-1153.

[4] Lin Z H, Feng M W, dos Santos C N, et al. A structured self-attentive sentence embedding[EB/OL]. 2017: arXiv: 1703.03130. https://arxiv.org/abs/1703.03130.

[5] Runeson P, Alexandersson M, Nyholm O. Detection of duplicate defect reports using natural language processing[C]//29th International Conference on Software Engineering. IEEE, 2007: 499-510.

[6] Sureka A, Jalote P. Detecting duplicate bug report using character N-gram-based features[C]//2010 Asia Pacific Software Engineering Conference. IEEE, 2011: 366-374.

[7] Sun C N, Lo D, Khoo S C, et al. Towards more accurate retrieval of duplicate bug reports[C]//26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011). IEEE, 2011:253-262.

[8] Yang C Z, Du H H, Wu S S, et al. Duplication detection for software bug reports based on BM25 term weighting[C]//2012 Conference on Technologies and Applications of Artificial Intelligence. IEEE, 2013: 33-38.

[9] Kukkar A, Mohana R, Kumar Y, et al. Duplicate bug report detection and classification system based on deep learning technique[J]. IEEE Access, 2020,8: 200749-200763.

[10] He J J, Xu L, Yan M, et al. Duplicate bug report detection using dual-channel convolutional neural networks[C]//Proceedings of the 28th International Conference on Program Comprehension. New York: ACM, 2020: 117-127.

[11] Deshmukh J, Annervaz K M, Podder S, et al. Towards accurate duplicate bug retrieval using deep learning techniques[C]// 2017 IEEE International Conference on Software Maintenance and Evolution(ICSME). IEEE, 2017:115-124.

[12] Prifti T, Banerjee S, Cukic B. Detecting bug duplicate reports through local references[C]//Proceedings of the 7th International Conference on Predictive Models in Software Engineering. IEEE, 2011:1-9.

[13] Poddar L, Neves L, Brendel W, et al. Train one get one free: partially supervised neural network for bug report duplicate detection and clustering[C]//Proceedings of the 2019 Conference of the North. Minneapolis-Minnesota. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019: 157-165.

[14] Rocha T M, Da Costa Carvalho A L. SiameseQAT: a semantic context-based duplicate bug report detection using replicated cluster information[J]. IEEE Access, 2021, 9: 44610-44630.

[15] Pennington J, Socher R, Manning C. Glove: global vectors for word representation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics, 2014: 1532-15.

[16] Chai H M, Lei J N, Fang M. Estimating Bayesian networks parameters using EM and Gibbs sampling[J]. Procedia Computer Science, 2017, 111:160-166.

[17] Lazar A, Ritchey S, Sharif B. Generating duplicate bug datasets[C]//Proceedings of the 11th Working Conference on Mining Software Repositories. New York: ACM, 2014:392-395.

[18] Budhiraja A, Dutta K, Reddy R, et al. DWEN: deep word embedding network for duplicate bug report detection in software repositories[C]//Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings. New York: ACM, 2018: 193-194.

[19] 范道远, 孙吉红, 王炜, 等. 融合文本与分类信息的重复缺陷报告检测方法[J]. 计算机科学, 2019, 46(12): 192-200.
Fan D Y, Sun J H, Wang W, et al. Detection method of duplicate defect reports fusing text and categorization information[J]. Computer Science, 2019, 46(12): 192-200.(in Chinese)

(编辑 詹燕平)