软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(6):1597−1611 [doi: 10.13328/j.cnki.jos.006239]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

大粒度 Pull Request 描述自动生成*

邝 砾[1], 施如意[1], 赵雷浩[1], 张 欢[1], 高洪皓[2]

[1](中南大学 计算机学院,湖南 长沙 410083)

[2](上海大学 计算机工程与科学学院,上海 200444)

通讯作者: 高洪皓, E-mail: gaohonghao@shu.edu.cn

摘 要: 在GitHub平台中,许多项目贡献者在提交Pull Request(PR)时往往会忽略提交PR描述,这使得提交的PR容易被评审者忽略或者拒绝.因此,自动生成PR描述以帮助项目贡献者提高PR通过率是很有必要的.然而,现有PR描述生成方法的表现会受到PR粒度影响,无法有效为大粒度的PR生成描述.因此,该工作专注于大粒度PR描述的自动生成.首先对PR中的文本信息进行预处理,将文本中的单词作为辅助节点构建词-句异质图,以建立PR语句间的联系;随后对异质图进行特征提取,并将提取后的特征输入至图神经网络进行图表示学习,通过节点间的消息传递,使句子节点学习到更丰富的内容信息;最后,选择带有关键信息的句子组成PR描述.此外,针对PR数据集缺少人工标注的真实标签而无法进行监督学习的问题,使用强化学习指导PR描述的生成,以最小化获得奖励的负期望为目标训练模型,该过程与标签无关,并且直接提升了生成结果的表现.在真实的数据集上进行了实验,实验结果表明,提出的大粒度PR描述生成方法在*F*1值和可读性上优于现有方法.

关键词: Pull Request描述;异质图神经网络;强化学习;非结构性文档;摘要生成

中图法分类号: TP311

中文引用格式: 邝砾,施如意,赵雷浩,张欢,高洪皓.大粒度 Pull Request 描述自动生成.软件学报,2021,32(6):1597−1611. http://www. jos.org.cn/1000-9825/6239.htm

英文引用格式: Kuang L, Shi RY, Zhao LH, Zhang H, Gao HH. Automatic generation of large-granularity pull request description. Ruan Jian Xue Bao/Journal of Software, 2021,32(6):1597−1611 (in Chinese). http://www.jos.org.cn/1000-9825/6239.htm

Automatic Generation of Large-Granularity Pull Request Description

KUANG Li[1], SHI Ru-Yi[1], ZHAO Lei-Hao[1], ZHANG Huan[1], GAO Hong-Hao[2]

[1](School of Computer Science and Engineering, Central South University, Changsha 410083, China)

[2](School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China)

Abstract: In GitHub platform, many project contributors often ignore the descriptions of pull requests (PRs) when submitting PRs, making their PRs easily neglected or rejected by reviewers. Therefore, it is necessary to generate PR descriptions automatically to help increase PR pass rate. The performances of existing PR description generation methods are usually affected by PR granularity, so it is difficult to generate descriptions for large-granularity PRs effectively. For such reasons, this work focuses on generating descriptions for large-granularity PRs. The text information is first preprocessed in PR and word-sentence heterogeneous graphs are constructed where the words are used as secondary nodes, so as to establish the connections between PR sentences. Subsequently, feature extraction is performed on the heterogeneous graphs, and then the features are input into graph neural network for further graph representation learning, from which the sentence nodes can learn more abundant content information through message delivery between nodes. Finally, the sentences

* 基金项目: 国家重点研发计划(2018YFB1003800); 国家自然科学基金(61772560)

Foundation item: National Key R&D Program of China (2018YFB1003800); National Natural Science Foundation of China (61772560)

本文由“形式化方法与应用”专题特约编辑姜宇副教授推荐.

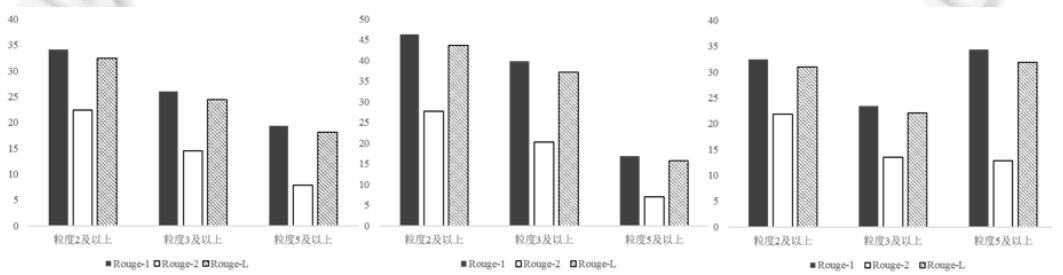
收稿时间: 2020-08-09; 修改时间: 2020-10-26; 采用时间: 2020-12-19; jos 在线出版时间: 2021-02-07

with key information are selected to form a PR description. In addition, the supervised learning method cannot be used for training due to the lack of manually labeled tags in the dataset, therefore, reinforcement learning is used to guide the generation of PR descriptions. The goal of model training is minimizing the negative expectation of rewards, which does not require the ground truth and directly improves the performance of the results. The experiments are conducted on real dataset and the experimental results show that the proposed method is superior to existing methods in F1 and readability.

Key words: Pull Request description; heterogeneous graph neural network; reinforcement learning; unstructured document; summarization generation

GitHub 是主流的面向开源及私有软件项目的托管平台,目前在世界范围内已有超过 5 百万的用户在该平台上托管了 1 千多万个开源项目[1].Pull Request(PR)是该平台上的消息通知机制,通过 Pull Request,项目贡献者将其对项目作出的变更通知给项目持有者.当PR经过评审后,项目持有者可根据评审结果决定是否将贡献者作出的变更合并到项目中.有研究表明,好的 PR 描述能够提高项目变更被合并的概率[2].然而,目前该平台上大量的 PR 缺少描述,或描述过于简单[3],这将不利于 PR 的合并.因此,为 PR 自动生成描述是非常有必要的.

已有研究者将 Pull Request 描述自动生成任务建模成文本摘要生成问题,其将 PR 中的 commit message 和 code comment 等文本信息看作原始文档,PR 描述看作摘要,利用带有注意力机制的编解码模型来生成 PR 描述,并得到了不错的 ROUGE 分数和人工评价分数[3].但是经研究发现,该模型的表现受 PR 的粒度影响非常大.一条 PR 会包含一条或多条 commit,commit 是项目贡献者对项目做出的具体改动,本文将一条 PR 包含的 commit 数目称为 PR 的粒度.在文献[3]使用的 PR 数据集中,有将近一半的 PR 仅包含 2 条 commit.生成小粒度 PR 的描述非常简单,即使将两条 commit 的文本信息简单罗列出来,也可以获得简短且准确的 PR 描述.如图 1 所示,我们移除了数据集中粒度为 2 的 PR,按照该方法重新训练了模型,并进行了测试,结果发现,模型性能有所下降.当利用其方法尝试为粒度为 5 及以上的大粒度 PR 生成描述时,该方法的表现在 $F1$ 和召回率上的下降幅度更大.因此本文认为:该方法不适用于生成大粒度 PR 的描述,需要进一步探索能够为大粒度 PR 生成 PR 描述的方法.



(a) $F1$ (b) 召回率 (c) 准确率

Fig.1 Performance of Liu *et al.*'s method on the PR data sets at different granularities

图 1 Liu 等人的方法在不同粒度下的 PR 数据集上的表现

为了解决上述问题,我们将PR描述生成问题建模为文本摘要生成问题.由于使用抽象式文本摘要生成模型生成的句子不具备良好的可读性,易出现逻辑错误,因此本文建立了抽取式文本摘要生成模型,选取 PR 文档中带有关键信息的句子组成 PR 描述.然而由文献[4]可知:目前主流的抽取式文本摘要模型学习到的大多是句子的位置信息而非内容信息,因此这些模型在新闻等结构性文档[5]数据集上表现良好[6].结构性文档中的句子重要程度往往与句子的位置有很大关系,例如在新闻文档常使用的倒三角结构中,位置越靠前的句子越为重要,越容易出现在摘要中.但是在 PR 文档中,句子的重要程度与句子所在位置并没有很强的关联性,因此需要模型有很强的学习文档内容的能力,能够通过句子内容判断句子的重要程度.Wang 等人[7]在 Zhong 等人[4]的工作基础上假设图神经网络更有利于帮助模型学习到文档内容信息,并使用其来完成摘要生成任务,取得了良好的结果,从而验证了该假设.在上述工作的启发下,我们将 PR 文本信息建模为词-句异质图,利用图神经网络进一步学习

到 PR 文本中句子的内容信息.具体的,将 PR 词-句异质图作为输入,首先分别对异质图中的节点和边进行特征提取,随后将特征向量输入图神经网络,进一步学习图结构信息和节点内容信息.最后将图神经网络学习到的句子节点特征向量经过线性变换后得到的结果作为该句被选为 PR 描述句的概率,并选取概率最大的 k 个句子组成 PR 描述.此外,我们使用 REINFORCE 算法,通过给生成的 PR 描述加以奖励并将训练目标设置为最小化奖励负期望的方式,避免了使用人工标注的真实标签指导 PR 描述生成,并且由于直接对生成 PR 描述的评价指标进行优化,提升了模型表现.

总而言之,本文工作体现在以下 3 个方面.

(1) 专注于解决大粒度的 PR 描述生成问题,将 PR 描述生成任务建模为抽取式文本摘要生成问题,选取 PR 文档中的关键句生成 PR 描述.为了能够有效学习到 PR 文档中的内容信息,构建了 PR 词-句异质图,建立起 PR 语句间的联系,随后利用图神经网络传递节点间的消息,帮助模型挖掘 PR 语句中更丰富的内容特征;

(2) 传统的抽取式文本摘要生成模型需要数据集提供人工标注的真实标签进行监督学习,然而 PR 数据集并不具备这样的条件.因此,我们使用 REINFORCE 算法,为生成的 PR 描述依据评价指标给出奖励,并将训练的损失函数设计为奖励的负期望.因为损失函数的设计与真实标签无关,所以在训练过程中无需使用真实标签,直接优化了评价指标,使生成的 PR 描述更加可靠;

(3) 在真实的数据集上进行了实验,实验结果表明:本文提出的方法真实有效,并且其表现优于现有其他方法.

本文第 1 节介绍相关工作.第 2 节讨论基于图神经网络的 PR 描述生成方法.第 3 节介绍实验设置,第 4 节分析实验结果.第 5 节对本文进行总结,并对未来工作进行展望.

1 相关工作

本节将介绍有关 PR 的研究、自然语言处理中现有的文本摘要生成方法以及自然语言处理技术在软件工程领域中的应用.

1.1 Pull Request

Pull Request(PR)是 GitHub 平台上的一种消息通知机制.项目贡献者在对项目作出修改后,可将该修改通过创建和提交 PR 的形式通知项目持有者;接下来,PR 评审者会对项目贡献者提交的 PR 进行评审;之后,项目持有者会根据评审结果决定是否将该修改合并到项目中去.目前已有很多的工作围绕着 PR 机制展开,例如何种 PR 更容易被合并以及 PR 机制衍生出的新问题.

项目贡献者创建并提交 PR 的最终目的是希望将其所作出的修改合并到项目中去,因此,大量研究者关注哪些因素会影响 PR 的接受度.Soares 等人[8]提出使用关联规则挖掘哪些因素能够帮助变更被合并到项目中.Chen 等人[9]则在前人的基础上使用了更大规模的数据集分析影响项目合并的因素并设计了预测器,该预测器在 $f1$ 指标上达到了 90%的分数.此外,随着近年来对软件歧视关注度逐渐提高,亦有研究者研究了在合并 PR 时是否也存在歧视现象,例如:Terrell 等人[10]就调查了性别因素对贡献接受率的影响;Jiang 等人[11]则研究了 PR 评论对贡献合并的影响,并提出了评论者推荐方法.

同样,亦有研究者对 PR 机制下产生的新问题展开了研究,例如,根据 PR 信息估算项目贡献者的工作量[12]、依据 PR 重要性对 PR 进行优先级排序[13]、为 PR 进行标记[14]等.此外,还有大量学者研究如何推荐 PR 评审[15−18].Liu 等人[3]则提出了一个新的研究方向,即 PR 描述自动生成,其使用加入了注意力机制的编解码模型来解决该问题.然而,该方法表现易受 PR 粒度影响,无法有效为大粒度的 PR 生成良好的描述.

1.2 文本摘要生成

文本摘要生成是自然语言处理领域中的传统问题,按照生成方法,可以分成抽取式摘要生成方法和抽象式摘要生成方法.其中,抽取式方法直接在原文中抽取带有关键信息的语句组成摘要;而抽象式方法则是由模型根

据学习到的文档内容重新组织语句,形成摘要.

TextRank[19]是抽取式摘要生成方法的典型代表,该方法参考 PageRank 算法[20],将句子作为节点,根据句子间的相似度构造无向有权边,得到文章结构图;随后计算图中各个节点的 TextRank 分数,选取 N 个得分最高的节点代表的句子组成摘要.随着神经网络的发展,越来越多的研究者开始使用神经网络进行抽取式摘要生成.Nallapati 等人[21]使用循环神经网络学习句子的表示向量,然后将其输入到分类器中,预测该句是否应当出现在摘要中.之后,学者们也开始探索其他神经网络结构在抽取式摘要生成中的应用.例如,Liu 等人[22]使用 Transformer 结构对文档进行编码.Zhong 等人为探究 Transformer 和 LSTM 这两种神经网络模型在抽取式摘要生成任务中的表现进行了消融实验,实验发现:两种模型大多学习到的是句子的位置信息而非内容信息,而 Transformer 比 LSTM 能学习到更多的内容信息[4].在此基础上,Wang 等人[7]认为,基于 Transformer 的神经网络结构可看作是一种全连接的图结构,因此其提出了基于图结构的模型,能够学习到更多文章内容信息的猜想.为了验证这一猜想,Wang 等人构建了文章结构图,使用图神经网络帮助学习文章内容信息,并取得了不错的效果.总的来说,抽取式摘要生成方法思路简单,易于实现,但是其得到的摘要却只能由原文中的句子组成,且摘要句之间连接较为生硬.

完成文本摘要生成任务的另一大类方法是抽象式生成方法.带有注意力机制的编解码模型[23]是抽象式摘要生成方法的常见模型,在此基础上,See 等人[24]提出了指针生成网络对该模型进行改进,有效降低了在实际场景中未登录词问题对模型表现的影响.当然,也有学者在注意力机制的设计上对原有模型进行改进,例如,Gehrmann 等人[25]设计了自底向上的注意力机制,使模型能够选取更加有效的信息.Liu 等人[26]则提出了一个新颖的解决抽象式文章摘要问题的思路,首先将源文档解析为抽象语义表示(AMR)图,然后将 AMR 图转换为摘要图,并根据摘要图来生成文章摘要.除此之外,Yu 等人[27]还引入对抗生成网络帮助生成摘要.总之,使用抽象式方法生成的摘要语言要更加灵活,但是也存在生成重复语句、结果可读性差等常见问题.

上述方法作为文本摘要生成的常见方法,在结构性文档数据集(例如新闻数据集)上取得了良好的效果.但非结构性文档没有明显的行文结构,句子的重要性与句子所在位置关联不大,与结构性文档存在较大差异,因此上述方法在非结构性文档数据集上的表现并不尽如人意.为此,有研究者提出了适用于非结构性文档数据集的摘要生成方法[5,28−31].例如,Zhao 等人[5]针对会议文档数据集中文档过长、难以捕捉有效信息的问题,设计了分层自适应的网络结构.Liu 等人[30]则致力于生成客服对话的摘要,其认为客服对话摘要应当具备逻辑的完整性与正确性,因此提出了 Leader-Writer 网络用于生成对话摘要.但是,上述方法解决的主要问题都和其使用的数据集自身特征有很强的关联性,无法直接应用于解决大粒度 PR 描述生成问题.

1.3 自然语言处理在软件工程领域的应用

随着神经网络成功应用在翻译任务中[32],自然语言处理(NLP)技术得到了迅猛的发展,并在各个领域都得到了广泛的应用,软件工程领域亦不例外.

受到神经网络翻译模型[32]的启发,在软件工程领域,很多学者开始思考如何将软件领域专业性很强的特定文档转换为易读的自然语言形式,以便开发人员理解[33−36].例如,一些学者利用神经机器翻译的方法自动生成 commit message,并取得了不错的结果[33,34];亦有学者将代码注释生成任务建模为将程序源代码的抽象语法树翻译为代码注释的问题,并利用神经网络翻译模型有效完成了该任务[35,36].

此外,自然语言处理的其他技术亦在软件工程领域中有所应用.例如:Alon 等人[37]受到词嵌入的启发,提出了 Code2Vec 方法用于对代码进行嵌入,得到代码的向量表示,为开展后续研究奠定了基础;Ye 等人[38]将命名实体识别问题引入软件工程领域,提出了识别软件工程领域特定实体的相关方法;Markovtsev 等人[39]则为了进一步获取代码中的语义信息,根据自然语言处理中的分词算法提出了代码分割的方法;Ferrari 等人[40]利用词嵌入等方法对需求中可能出现的歧义进行检测;Chen 等人[41]则建立了贝叶斯分层主题模型,分析用户与系统交互的行为轨迹;Alreshedy 等人[42]应用 NLP 技术对 Stack Overflow 中问题和代码片段中所使用的程序语言进行了预测;Hao 等人[43]则利用 NLP 技术对众包测试中工人提供的测试报告进行处理,然后通过聚类将众包测试工人的测试报告汇聚成一份综合完整的报告以供软件项目管理人员参考.

PR 描述自动生成是近年来由 Liu 等人提出的新问题,可以看作是 NLP 技术在软件工程领域中的一个新应用.Liu 等人发现,在 GitHub 平台中有大量的 PR 缺少描述,这将不利于 PR 在评审中取得好的结果[3].为此,Liu 等人将 PR 描述自动生成建模为抽象式文档摘要生成问题,将 PR 中的文本信息看作源文档,将 PR 描述看作摘要,利用带有注意力机制的编解码模型生成 PR 描述.在此基础上,生成 PR 描述还需要解决两个关键问题,即未登录词问题以及训练目标和评价指标不一致的问题,因此,其又引入了指针生成网络[22]和强化学习来提升模型表现,并在评价指标和人工评审方面都取得了不错的成绩.然而,该方法却没有考虑到 PR 粒度对结果的影响,并不能为大粒度的 PR 生成良好的描述.

2 基于图神经网络的 PR 描述生成方法

与 Liu 等人[3]的思路类似,本文将 PR 描述生成问题建模为摘要生成问题,将 PR 中的 commit message 和 code comment 等文本信息看作是源文档,将 PR 描述看作摘要.然而,与之不同的是,为了避免生成结果出现逻辑错误,我们使用抽取式摘要生成模型,采用序列标注的思路进行 PR 描述生成.为了能够学习到 PR 语句中更为丰富的内容信息,我们构建了词-句异质图,建立起 PR 语句间的联系,利用图神经网络进行节点间的消息传递,进一步挖掘 PR 语句节点的内容特征.

PR 描述生成问题形式化定义如下:对于给定的 PR 描述源文档 S,其包含一系列的句子$(s_1,s_2,\ldots,s_n)$.PR 描述生成器的目标是在文档 S 中抽取 k 个句子,使其组成一条 PR 描述 D.对于源文档中 S 中的每一个句子 s_n 而言,生成器预测其标签 $y_n\in\{0,1\}$,其中,0 代表该句未被选为描述句,1 代表该句被选为描述句.基于图神经网络的 PR 描述生成方法总体结构如图 2 所示.我们首先对 PR 描述文档进行预处理,建立与之对应的词-句异质图;然后将异质图作为 PR 描述生成器的输入,利用生成器生成 PR 描述.此外,由于 PR 数据集缺少人工标注的真实标签,难以进行监督训练,我们使用强化学习方法对模型进行训练.具体的,将 PR 描述生成器看作智能体,PR 源文档看作状态,生成器选择哪几个句子组成 PR 描述看作智能体作出的策略,生成的 PR 描述看作新的状态.在将 PR 源文档输入进生成器后,生成器会作出相应策略生成 PR 描述,模型利用奖励函数对 PR 描述进行评价,并根据结果调整生成器的相关参数,从而使模型得到训练.本节主要对基于图神经网络的 PR 描述生成方法进行详细描述.

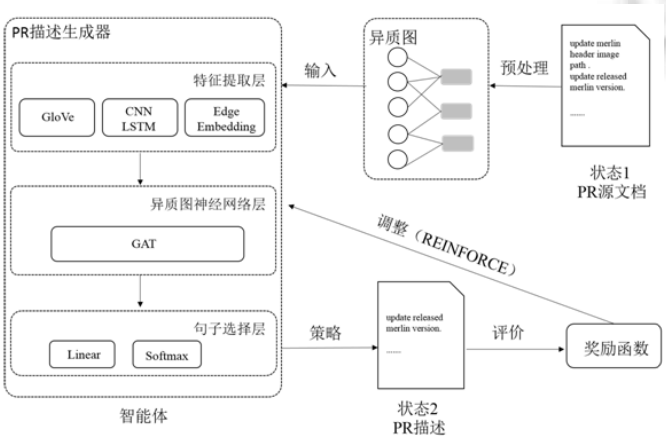


Fig.2 Structure of PR description method

图 2 PR 描述生成方法结构

2.1 图的构成

本文将 PR 源文档建模为以词节点、句子节点及其对应的边组成的带权异质图,其结构如图 3 所示.

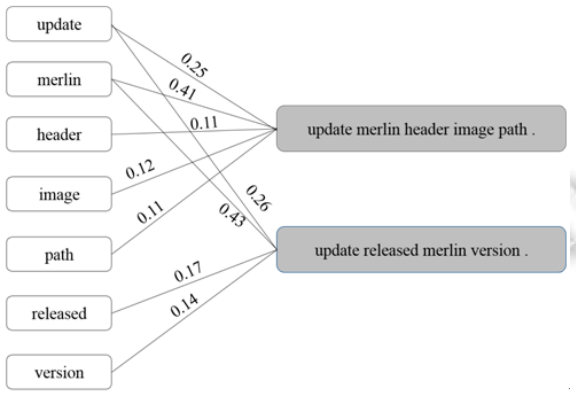


Fig.3 Heterogeneous graph structure

图 3 异质图结构

形式化的,对于给定的图 $G=\{V,E\}$,其中,V 代表节点,E 代表边.进一步地,$V=V_s\cup V_w$,代表句子和词节点的集合:$V_s=\{s_1,s_2,\ldots,s_n\}$表明句子节点集合 V_s 中含有 n 个句子;而 V_w 代表词节点的集合,$V_w=\{w_1,w_2,\ldots,w_m\}$表明集合中有 m 个单词.并且 $E=\{e_{11},\ldots,e_{ij},\ldots,e_{mn}\}$(其中,$i\in\{1,\ldots,m\}$,$j\in\{1,\ldots,n\}$),当 $e_{ij}=0$ 时,表明单词 w_i 与句子 s_j 之间不存在关系;而当 $e_{ij}\neq 0$ 时,表明句子中包含单词.e_{ij} 的值表明单词 w_i 在句子 s_j 中的重要程度,其计算方式如公式(1)、公式(2)所示:

$$t_i=TF_i\times IDF_i \tag{1}$$

$$e_{ij}=\frac{c_i t_i}{\sum_{w_z\in s_j} c_z t_z} \tag{2}$$

其中,t_i 是单词 w_i 在 PR 源文档 S 中的 TFIDF[44]分数.TFIDF 是一种统计学方法,用以评估一个词对于一个文件集或一个语料库中的其中一份文件的重要程度.TF_i 指的是单词 w_i 出现在 PR 源文档 S 中的次数;IDF_i 是指单词 w_i 的逆向文件频率,是一个词普遍重要性的度量,其值越大,则说明该词具有越强的类别区分能力;e_{ij} 是由单词 w_i 出现在句子 s_j 中的次数 c_i 乘以单词 w_i 的 TFIDF 分数 t_i,再除以句子 s_j 中所有单词的 TFIDF 分数分别乘以该单词出现的次数之和得到的.在这里,我们之所以不直接使用单词 w_i 在句子 s_j 中的 TFIDF 分数作为边权重的值,是因为 PR 源文档中的句子大多比较短小,在此情境下使用 TFIDF 分数的计算方法,会使计算结果丧失统计学意义.而利用公式(2)计算出的权重一方面保留了 TFIDF 分数的统计学意义,另一方面也在一定程度上体现了单词 w_i 在 PR 源文档中重要程度.

2.2 基于异质图神经网络的PR描述生成器

本节将介绍基于异质图神经网络的 PR 描述生成器的结构设计以及各部分的功能.PR 描述生成器主要包含 3 层,分别为特征提取层、图神经网络内容学习层、句子选择层.

首先在特征提取层对异质图特征进行提取.根据异质图的构建方式,模型需提取异质图词节点、句子节点以及边这 3 个方面的特征.在这里,使用词嵌入方法对单词特征进行提取.为了能够使模型有更好的表现,我们使用预训练的 GloVe[45]向量作为词节点的特征向量.对于句子节点 s_j,先利用卷积神经网络(CNN)来提取它的局部特征,再利用双向长短期记忆模型(BiLSTM)获得它的全局特征向量.然后,将得到的两个向量进行拼接,作为句子节点的特征向量.此外,为了获取更多的信息,我们也利用一层随机初始化的嵌入层对边的特征进行了提取.

之后,提取到的节点和边的特征向量以及图的结构信息将被输入进图神经网络层,进一步学习 PR 文档的内容信息.在图神经网络层中,节点间会彼此进行消息传递,以丰富某一节点的内容信息.在这里,使用图注意力网络模型(GAT)[46]进行节点间的消息传递. h_i 代表节点 i 的隐藏层状态,图 4 描述了如何得到节点 i 在下一层的隐藏层状态 h_i'.

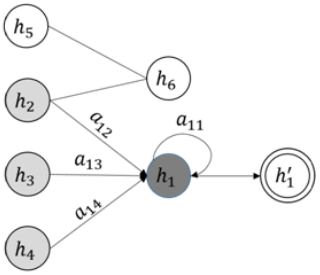


Fig.4 Calculation of graph neural network hidden layer state h_i'

图 4 图神经网络隐藏层状态 h_i' 的计算

以节点 1 为例,假设节点 1 当前的隐藏层状态为 h_1,其邻居节点分别为节点 2~节点 4.节点 5 和节点 6 与节点 1 同在一幅子图内,但与节点 1 并不相邻.我们希望获取节点 1 新的隐藏层状态 h_1'.首先需要计算节点 1 对其自身以及对其邻居节点 2~节点 4 的注意力权重 a_{11}~a_{14};随后对其自身以及邻居节点的隐藏层状态进行加权求和,得到新的隐藏层状态 h_1'.此时,隐藏层状态 h_1' 将包含节点 1 与节点 1 邻居节点的相关信息通过这种方式节点间进行了消息的传递.

在图神经网络层,我们利用节点间的消息传递更新了句子节点的隐藏层状态,得到了带有丰富文章内容信息的句子特征向量.然而,模型最终目的是预测 PR 源文档中的句子 s_j 是否应出现在 PR 描述中,所以将获得的句子向量再通过一层线性变换层,得到该句出现在 PR 描述中的概率.我们选择概率最大的 k 个句子组成 PR 描述.

2.3 融合强化学习的PR描述生成方法

传统的有监督标签预测任务常使用交叉熵损失函数[47]作为训练的损失函数,该函数能够有效衡量预测值与真实值之间的差距.然而在 PR 描述生成任务中,使用交叉熵函数作为损失函数并不合适,其原因有二.

(1) 交叉熵函数是监督学习中常用的损失函数,它需要数据带有真实标签.然而在 PR 数据集中,仅有 PR 原有的文本信息和由项目贡献者们书写的真实描述,并没有源文档中的某个句子是否属于PR描述的标签信息.当然,可以像文献[21]中所做的工作一样,利用贪心策略自动化地生成标签,但是这种标签生成方法是否可靠尚未可知,若该方法存在误差,则会为模型带来误差传递的风险;

(2) 因最终目标是生成 PR 描述,同 Liu 等人[3]一样,我们使用摘要生成任务中常用来衡量生成摘要与真实摘要之间差距的 ROUGE 分数[48]作为评价指标来衡量 PR 描述生成质量的优劣.若使用交叉熵函数作为损失函数,仅仅能够衡量 PR 描述句的预测是否准确,并没有将生成的 PR 描述看作一篇文章对它进行评估,从而无法对 PR 描述进行整体上的质量评价,因此我们认为,交叉熵函数不适宜作为本项工作的损失函数.

为了解决上述问题,我们引入了强化学习提升模型效果.在这里,将 PR 描述生成器看作智能体,PR 源文档看作状态,PR 描述生成器选择哪几个句子组成 PR 描述看作智能体作出的策略,生成的 PR 描述看作新的状态.另外,我们把智能体获得的奖励设置为智能体生成的 PR 描述得到的 ROUGE 分数,将损失函数定义为智能体获得奖励的负期望.通过这些设置,模型的训练目标与真实值无关,因此在训练过程中,能够避免使用真实标签指导 PR 描述的生成,并且对评价指标进行了优化,很好地解决了上述两个问题.

具体地,定义策略 $p(y_i|s_i,S,\theta)$ 表示模型是否将句子 $s_i \in S$ 选为描述句,其中,$y_i \in \{0,1\}$,θ 为训练参数.当智能体阅读完 PR 源文档 S 后,会作出决策,得到 $y_i=1$ 的句子集合 $\hat{y}$.我们将该集合中包含的句子进行拼接,看作智能体生成的 PR 描述.之后,智能体将会得到奖励 $r(\hat{y})$,该奖励能够评价 PR 描述生成器生成 PR 描述的效果.在这里,使用智能体生成的描述所获得的 ROUGE 分数作为奖励.模型的训练目标是最小化智能体得到奖励的负期望,因此,损失函数 $L(\theta)$ 可用公式(3)表示:

$$L(\theta) = -\mathbb{E}_{\hat{y}\sim p(\theta)}[r(\hat{y})] \tag{3}$$

但是,由于奖励函数即 ROUGE 分数计算方式不可导[48],因此,我们利用 REINFORCE 算法[49]近似得到损失

函数的梯度,如公式(4)所示:

$$\nabla L(\theta) = -\mathbb{E}_{\hat{y}\sim p(\theta)}[r(\hat{y})\nabla \log p(\hat{y}\mid S,\theta)] \tag{4}$$

其中,$\nabla L(\theta)$表示损失函数 $L(\theta)$的梯度;$\mathbb{E}$表示期望,$r(\hat{y})$ 表示智能体得到集合 $\hat{y}$ 所获得的奖励; $p(\hat{y}\mid S,\theta)$ 表示智能体在参数为θ时,以文档 S 为输入得到输出集合 $\hat{y}$ 的概率.REINFORCE 算法在进行了统计学上的观察后,认为利用 $p(\hat{y}\mid S,\theta)$ 对数的梯度乘以 $\hat{y}$ 获得的奖励负期望,可近似替代损失函数的梯度.

由于智能体作出的策略存在多样性,将损失函数中的期望全部计算出来的代价是十分高昂的,因此在实践中常常使用单个样本近似代替期望值[32].故损失函数的梯度可用公式(5)计算:

$$\nabla L(\theta) \approx -r(\hat{y})\nabla \log p(\hat{y}\mid S,\theta) \tag{5}$$

其中, $\hat{y}$ 为近似代替期望值的单个样本.依据 REINFORE 算法,梯度$\nabla L(\theta)$可近似看作得到该样本的概率 $p(\hat{y}\mid S,\theta)$ 取对数后的梯度与该样本所获得奖励 $r(\hat{y})$ 乘积的负值.

为了能够使模型取得优良的效果,并提升训练速度,我们希望在采样过程中避免随机化[50],并使用高分数的样本来近似代替期望.因此尝试了两种采样方法.

(A) 使用了 Ranzato 等人[51]提出的采样方法.该方法使用交叉熵损失函数预训练出一个模型用于获得高分的样本.即,使用该方法训练会有两个阶段:在第 1 阶段,使用交叉熵函数预训练模型;第 2 阶段使用 REINFORCE 算法调整模型参数,并得到最终的结果;

(B) 使用了 Shashi 等人使用的采样方法[52],将损失函数中的 $\hat{y}$ 搜索空间限制在预先计算出的高分句集合 Y 中.同其他抽取式摘要生成方法的假设一样,我们认为被选中的句子其 ROUGE 分数应该较高,因此将 Y 近似看作拥有最高 ROUGE 分数的 k 个句子的集合.

3 实验设置

本节将介绍实验的相关设置,包括实验所使用的数据集、实现细节、基线实验以及评价指标.

3.1 数据集

我们使用的是 Liu 等人提供的数据集[3],该数据集是由 Liu 等人从 GitHub 上的 1 000 个 Java 项目中爬取的 333 001 条数据组成的.经过过滤后,该数据集中包含 41 832 有效数据.每条数据均包含了由 commit message 和 code comment 等 PR 文本信息组成的 PR 源文档和由项目贡献者书写的真实的 PR 描述.为分析可能影响 PR 描述生成效果的因素,我们对数据集中不同粒度 PR 的数量进行了统计,统计结果如图 5 所示.

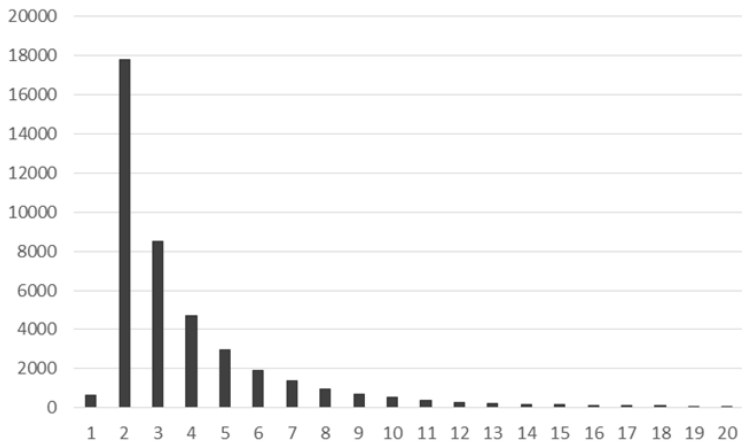


Fig.5 PR granularity statistics

图 5 PR 粒度数目统计

在图 5 中,横坐标代表 PR 的粒度,纵坐标代表 PR 的数目.可以看出:该数据集中 PR 粒度的分布十分不均,

有 17 802 条 PR 只包含 2 条 commit,而粒度为 20 的大粒度 PR 仅有 53 条.由于目标是为大粒度的 PR 生成描述,我们过滤了粒度为 2~4 的 PR,保留了粒度为 5 及以上的 PR.过滤后,最终数据集总共包含 10 144 条数据.

3.2 实现细节

与常规设置相同,实验中使用的词表包含 50 000 个单词.在构图时,为了避免停用词等特殊词汇对实验结果的影响,我们在选择词节点时过滤了低频词、停用词和标点符号.

同 Wang 等人[7]的设置相同,我们初始化单词节点为 300 维的特征向量,句子节点为 128 维的特征向量,初始化边 e_{ij} 为 50 维的向量.在图神经网络层使用 8 头的自注意力机制学习节点信息,并且将隐藏层大小设置为 64.在训练过程中,我们使用了 Adam 优化器,并将学习率设置为 5e–4.在解码过程中选择抽取 3 个句子生成 PR 描述,选择抽取句子的数目是通过调参实验确定的,该实验结果将在第 4.1 节中详细描述.

3.3 基线实验

与 Liu 等人[3]的工作相同,我们选取 LeadCM 和 LexRank 作为基线实验;同时,Liu 等人的工作也被当作基线实验与本文提出的方法进行对比.

- LeadCM:LeadCM 是常用的摘要抽取方法,其实现简单,在摘要生成领域颇受工业界的欢迎.其思想是抽取文档中的前 k 个句子组成摘要,与上述实验的设置相同,本文也利用 LeadCM 获取 3 个句子组成 PR 描述;
- LexRank:LexRank 是一种通过计算句子的重要性对句子进行排序,并选择文章中最重要的句子作为摘要的摘要生成方法.它是 PageRank 算法在抽取式文档摘要生成中的应用.对于给定的 PR 源文档,我们利用 LexRank 方法,首先根据重要性对句子进行排序,然后选择排在前三的句子组成 PR 描述;
- PG+RL:PG+RL 是 Liu 等人生成 PR 描述方法的核心部分,该方法利用带有注意力机制的编解码模型抽象式的生成 PR 描述.为解决 PR 描述生成过程中会出现大量未登录词以及训练目标与评价指标不一致的问题,Liu 等人引入了指针生成网络和强化学习来优化模型.

3.4 评价指标

本文选用摘要生成任务中常使用的 ROUGE[44]分数作为衡量生成的 PR 描述质量高低的评价指标.ROUGE 指标在召回率、准确率和 $F1$ 这 3 个维度衡量生成摘要与真实摘要之间差距,其由一系列评价规则构成,在这里,主要使用最为常见的 ROUGE-N 和 ROUGE-L 评价方法,其中,ROUGE-N 评价方法的计算公式如下:

$$R_{rouge\text{-}n}=\frac{\sum_{gen,ref\in S}\sum_{gram_n\in ref}C_{gen}(gram_n)}{\sum_{gen,ref\in S}\sum_{gram_n\in ref}C_{ref}(gram_n)} \tag{6}$$

$$P_{rouge\text{-}n}=\frac{\sum_{gen,ref\in S}\sum_{gram_n\in ref}C_{gen}(gram_n)}{\sum_{gen,ref\in S}\sum_{gram_n\in gen}C_{gen}(gram_n)} \tag{7}$$

$$F1_{rouge\text{-}n}=\frac{2R_{rouge\text{-}n}P_{rouge\text{-}n}}{R_{rouge\text{-}n}+P_{rouge\text{-}n}} \tag{8}$$

其中,R,P,$F1$ 分别代表召回率、准确率和 $F1$ 这 3 个维度;*rouge-n* 代表 ROUGE-N 评价规则,在这里,n 取 1 或 2,指应用 n 元文法(n-gram);*gen*,*ref*,*S* 则分别代表了生成的 PR 描述、人工书写的参考 PR 描述及测试集;$gram_n$ 代表 n-gram 短语,$C_{gen}(gram_n)$和 $C_{ref}(gram_n)$分别代表 n-gram 短语在生成的 PR 描述和参考描述中出现的次数.

在这里:召回率衡量的是生成描述中出现的关键词(生成描述和参考描述中共同出现的 n-gram 短语)数量占参考摘要的比重,即生成描述要捕捉到了多少参考描述中的信息,更加看重生成描述中所带信息数量;而准确率衡量的则是生成描述中的关键信息占生成描述的比重,其更看重生成描述是否足够精简;$F1$ 维度平衡了召回率与准确率,其既希望生成的描述能带有一定的信息,又希望它是精简的,是一个比较综合的指标.在本问题中,相对于准确率,我们更希望模型在召回率方面能够有所提升,这意味着模型捕捉到 PR 中的关键信息将会更多,生成结果对 PR 的描述会更加全面.但是仅仅关注召回率有可能会使生成结果语言不够精炼,带有大量的无关信

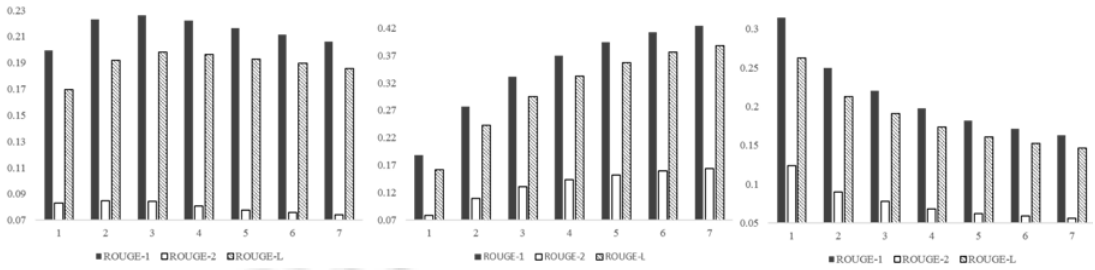
息,故准确率在这里也有一定的参考价值.因此,综合考虑后,我们会更加关心模型在 $F1$ 维度上的表现.

4 实验结果

本节将给出实验的结果,并对实验结果进行分析.为避免实验结果的偶然性,我们进行了十折交叉实验,本节中的数据均为交叉实验的平均结果.

4.1 调参实验

为了验证在解码过程中抽取几个句子最为合适,我们进行了相关的调参实验.图 6 为随着句子数目的变化,利用本文所提方法生成的 PR 描述获得的 ROUGE 分数的变化情况.其中,横坐标代表的是抽取的句子数目,纵坐标代表的是 ROUGE 分数.



(a) $F1$ (b) 召回率 (c) 准确率

Fig.6 ROUGE score changes with the number of sentences extracted

图 6 ROUGE 分数随抽取的句子数目变化情况

由图 6 不难看出:随着抽取的句子数目的增多,召回率分数会不断增加.这是因为抽取的句子数目越多,得到有效信息的概率也会越大.相对的,准确率则会下降,因为抽取的句子越多,得到的无关信息也会越多.因此,为了平衡得到有效信息的数目与生成的 PR 描述凝练程度的关系,在本小节中,我们着重关心生成的 PR 描述的 ROUGE 分数的 $F1$ 维度.我们发现:在 $F1$ 方面,抽取 3 个句子的 PR 描述是得分最高的.因此在最终解码时选择抽取 3 个句子.

4.2 对比实验

我们简称本文提出的融合强化学习的基于异质图神经网络的大粒度 PR 生成方法为 HGPRG-RL,在本节中,我们将 HGPRG-RL 与基线方法进行对比,结果见表 1.

Table 1 Experimental results

表 1 实验结果

	ROUGE-1			ROUGE-2			ROUGE-L		
	f	*r*	*p*	*f*	*r*	*p*	*f*	*r*	*p*
LeadCM	22.74	26.68	26.64	**9.6**	12.11	10.14	18.68	22.39	21.19
LexRank	17.85	23.26	19.50	5.23	7.21	5.36	13.89	18.58	14.99
PG+RL	19.34	16.79	**34.29**	7.86	7.05	**12.84**	18.08	15.76	**31.89**
HGPRG-cross	22.59	**32.88**	22.05	8.38	**12.90**	7.77	19.75	**29.21**	19.10
HGPRG-RL(A)	22.74	26.68	26.64	**9.6**	12.11	10.14	18.68	22.39	21.19
HGPRG-RL(B)	**22.83**	32.51	22.61	8.45	12.67	8.44	**19.98**	28.89	19.62

LeadCM,LaxRank,PG+RL 这 3 行分别为 3 种基线方法得到的结果,HGPRS-cross 这一行是我们提出的基于图神经神经网络的 PR 描述生成器使用交叉熵函数作为训练损失函数得到的结果,HGPRG-RL(A)和 HGPRG-RL(B)这两行则分别是 HGPRG-RL 使用第 2.3 节中提到的两种采样方法(A)和方法(B)得到的结果.

从表 1 中可以看出:除了 ROUGE-2,HGPRG-RL(B)能够在 $F1$ 值上得到最好的效果,并且我们提出的方法在召回率上都取得了不错的效果,意味着本文提出的方法比基线方法能够得到更多有用的信息.

注意到,Liu 等人的方法在准确度上取得了最佳的效果.我们分析,这是因为该方法使用的是抽象式摘要生成模型,生成的 PR 描述长短不受源文档中的句子的长度所限制,可以只提取句子中的关键词.而其他几种方法,包括我们所提出的方法都是抽取式方法,需要抽取原文中的整个句子.虽然我们的方法能够比 Liu 等人的方法捕捉到更多的关键信息,但是我们生成的PR描述的长度受原句的限制,因此模型选取的句子在包含了大量有效信息的同时,也有可能包含一些无关信息,故而我们方法的准确度不及 Liu 等人的方法.但是从总体($F1$)上看,我们的方法还是更优的.

同时,我们还注意到:相比于使用交叉熵函数作为损失函数,使用强化学习的方法以牺牲一定程度的召回率为代价,得到了准确率的提升.但是这种牺牲是值得的,因为从总体($F1$)上看,使用强化学习还是对模型有所优化的.此外,实验发现:使用采样方法(A),会使我们的方法退化成 LeadCM.我们推测,其原因可能是使用交叉熵作为损失函数训练出的模型选择源文档前 3 个句子的概率较大,在使用强化学习模型进一步对模型参数进行调整后,会进一步放大这种倾向性,因此使模型退化成了 LeadCM.

4.3 实例分析

本节将通过实例验证我们提出的方法在大粒度的 PR 描述生成上的优越性.

(1) 实例 1

PR 原文:

Added value ‘unknown’ for ‘repository depth option’. ⟨cm-sep⟩ added a test case for verifying that depth ‘unknown’ works. ⟨cm-sep⟩ [jenkins-0] changed ‘undefined’ to ‘as-it-is’ in Web interface of subversion-plugin. ⟨cm-sep⟩ [jenkins-0] move ‘as-it-is’ option to the end. ⟨cm-sep⟩ [jenkins-0] updated help page so it refers to ‘as-it-is’ instead uf ‘unknown’. ⟨para-sep⟩ enable version mode. Do initial update with infinite depth and check that subdir exists. Simulate job using ‘svn update--set-depth=files’ and check that subdir no longer exists. Trigger new build with depth unknown and check that subdir still does not exist.

真实 PR 描述:

I added value ‘unknown’ for ‘repository depth option’ in subversion-plugin this allows a job to reduce size of working copy by executing ‘svn update--set-depth=...’ and have this reduction preserved when job runs again on the same node.

HGPRG-RL:

Added value ‘unknown’ for ‘repository depth option’.

Enable version mode do initial update with infinite depth and check that subdir exists simulate job using ‘svnupdate--set-depth=files’ and check that subdir no longer exists trigger new build with depth unknown and check that subdir still does not exist.

[jenkins-0] updated help page so it refers to ‘as-it-is’ instead uf ‘unknown’.

PG+RL:

Added value ‘unknown unknown’ for ‘repository depth option’.

在实例 1 中,该 PR 源文档中包含了 5 条 commit message 和一条 code comment.真实 PR 描述是由创建该 PR 的项目贡献者人工书写的.从真实 PR 描述中可以看出,该 PR 作出的主要更改是在 subversion-plugin 中为“repository depth option”添加了“unknown”值.这样做的目的是允许任务通过执行“svn update--set-depth=...”命令减少需要复制的内容,并且任务在相同节点再次执行时,可以同样减少内容复制.如此一来,可以减少内存的消耗,提高运行效率.PG+RL 是使用 Liu 等人的方法生成的 PR 描述,可以看到其方法仅捕捉到了为“repository depth option”添加了“unknown”值这一条有效信息,且还有重复生成词“unknown”的错误出现.而 HGPRG-RL 是利用我们的方法生成的 PR 描述,其不止捕捉到了为“repository depth option”添加了“unknown”值的关键信息,还捕捉到了检查“subdir exists.simulate”作业是否使用“svn update--set-depth=files”执行这一关键信息.

(2) 实例 2

PR 原文:

At a stateless pure javascript/client side component. ⟨cm-sep⟩ wrap models to be able to user resource models. ⟨cm-sep⟩ add an example fo client side tabs. ⟨cm-sep⟩ get rid of extra iteration of tabs. ⟨cm-sep⟩ introduce isactive. ⟨cm-sep⟩ rename model. ⟨cm-sep⟩ make tabs dynamic. ⟨cm-sep⟩ get rid of un-needed space. ⟨cm-sep⟩ improve text. ⟨cm-sep⟩ improve java doc, rename some methods, index start counting at 0 to be compatible with classical tabeedpanel. ⟨cm-sep⟩ add a test unit for client side tabs. ⟨cm-sep⟩ add more tests and add css class specific to each tab. ⟨para-sep⟩ a pure client side tab. it is completely stateless. ⟨para-sep⟩ ⟨*p*⟩ a 'pure' client side stateless tabs component. But instead of generating links that trigger server round trips id does generates just 'client side' links. ⟨/*p*⟩.

真实 PR 描述:

At a stateless pure javascript/client side component.

I just need this for a project and thought it might be useful to other users. It works as in classical but user only needs to use it as 'standard' wicket tabs. Mind that all content panels are created at the same time. If accepted i can add an example to demo application.

HGPRG-RL:

At a stateless pure javascript/client side component.

Wrap models to be able to user resource models.

Improve java doc rename some methods index start counting at 0 to be compatible with classical tabeedpanel.

PG+RL:

Add an example to be able to user to be compatible with classical tabeedpanel.

实例 2 是由 12 条 commit message 和 2 条 code comment 组成的大粒度 PR.由项目贡献者书写的真实 PR 描述可知:该条 PR 所作出的修改是在无状态的纯 javascript/client 端组件中添加了一项针对用户的操作,该操作与传统操作原理相同,用户把它当作标准选项卡使用即可.使用我们的方法生成的 PR 描述能够成功获取到修改的位置信息,亦能得知该功能是针对用户资源模型所作出的修改以及该 PR 作出的一些细节修改.

仔细分析 Liu 等人方法生成的 PR 描述可知:该描述来源于"add an example fo client side tabs", "wrap models to be able to user resource models", "improve java doc, rename some methods, index start counting at 0 to be compatible with classical tabeedpanel"这 3 条 commit message.其分别在 3 条 commit message 中提取了关键词并进行了简单拼接,但是其拼接得到语句的意思却与原文的意思完全不同.项目贡献者添加实例的目的并不是使用户能与原始面板兼容,并且用户与面板兼容也不符合语言逻辑.

综上,可以看出:本文提出的方法生成的 PR 描述具有较好的可读性,能够避免一些逻辑错误,且能够捕捉到能多的有用信息.因此,我们的方法在大粒度的 PR 描述生成问题上更具有优势.

5 结 论

针对先前工作的不足,我们为大粒度的 PR 描述自动生成提出了一种切实可行的方法.将大粒度的 PR 描述自动生成问题建模为抽取式摘要生成问题,为了更好地学习 PR 源文档中的内容信息,我们构建了以词节点为辅助节点的词-句异质图,使 PR 源文档中句子间的联系得以建立,随后提取异质图的节点特征信息,并利用图神经网络进一步学习 PR 异质图的图表示向量,使模型学习到了更丰富的 PR 语句内容信息.同时,我们使用了 REINFORCE 算法,避免了使用人工标记的标签指导 PR 描述的生成,降低了对数据集的要求,且能够使模型在评价指标上取得更好的表现.在真实的数据集上进行了实验,实验结果表明,我们的方法优于现有的 PR 描述生成方法.未来,我们将研究如何减少未登录词对 PR 源文档内容学习的影响,以便更好地生成 PR 描述.

References:

[1] https://github.com

[2] Georgios G, Storey MA, Bacchelli A. Work practices and challenges in pull-based development: the contributor's perspective. In: Kellenberger P, ed. Proc. of the 38th Int'l Conf. on Software Engineering (ICSE). Austin: IEEE, 2016. 285−296. [doi: 10.1145/2884781.2884826]

[3] Liu ZX, Xia X, Treude C, Lo D, Li SP. Automatic generation of pull request descriptions. In: Proc. of the 34th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE). San Diego: IEEE, 2019. 176−188.[doi: 10.1109/ASE.2019.00026]

[4] Zhong M, Liu PF, Wang DQ, Qiu XP, Huang XJ. Searching for effective neural extractive summarization: What works and what's next. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 1049−1058. [doi: 10.18653/v1/P19-1100]

[5] Zhou Z, Pan HJ, Fan CJ, Liu Y, Li LL, Yang M, Cai D. Abstractive meeting summarization via hierarchical adaptive segmental network learning. In: Proc. of the World Wide Web Conf. (WWW 2019). New York: Association for Computing Machinery, 2019. 455−3461. [doi: https://doi.org/10.1145/3308558.3313619]

[6] Kedzie C, Kathleen M, Hal D. Content selection in deep learning models of summarization. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 1818−1828. [doi: 10.18653/v1/D18-1208]

[7] Wang DQ, Liu PF, Zheng YY, Qiu XP, Huang XJ. Heterogeneous graph neural networks for extractive document summarization. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2020. 6209−6219. [doi: 10.18653/v1/2020.acl-main.553]

[8] Soares DM, de Lima Júnior ML, Murta L, Plastino A. Acceptance factors of pull requests in open-source projects. In: Proc. of the 30th Annual ACM Symp. on Applied Computing. New York: Association for Computing Machinery, 2015. 1541−1546. [doi: https://doi.org/10.1145/2695664.2695856]

[9] Chen D, Stolee KT, Menzies T. Replication can improve prior results: A Github study of pull request acceptance. In: Proc. of the 27th Int'l Conf. on Program Comprehension (ICPC). Montreal: IEEE 2019. 179−190.

[10] Terrell J, Kofink A, Middleton J, Rainear C, Murphy-Hill E, Parnin C, Stallings J. Gender differences and bias in open source: Pull request acceptance of women versus men. PeerJ Computer Science, 2017,3:e111. [doi: https://doi.org/10.7717/peerj-cs.111]

[11] Jiang J, Yang Y, He J, Blanc X, Zhang L. Who should comment on this pull request? Analyzing attributes for more accurate commenter recommendation in pull-based development. Information and Software Technology, 2017,84:48−62.

[12] Maddila C, Bansal C, Nagappan N. Predicting pull request completion time: A case study on large scale cloud services. In: Proc. of the 27th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. New York: Association for Computing Machinery, 2019. 874−882.

[13] van der Veen E, Gousios G, Zaidman A. Automatically prioritizing pull requests. In: Proc. of the 12th Working Conf. on Mining Software Repositories. Florence: IEEE, 2015. 357−361. [doi: 10.1109/MSR.2015.40]

[14] Yu S, Xu L, Zhang Y, Wu JS, Liao ZF, Li YB. NBSL: A supervised classification model of pull request in Github. In: Proc. of the IEEE Int'l Conf. on Communications (ICC). Kansas City: IEEE, 2018. 1−6. [doi: 10.1109/ICC.2018.8422103]

[15] Xia X, Lo D, Wang X, Yang XH. Who should review this change? Putting text and file location analyses together for more accurate recommendations. In: Proc. of the Int'l Conf. on Software Maintenance and Evolution (ICSME). Bremen: IEEE, 2015. 261−270. [doi: 10.1109/ICSM.2015.7332472]

[16] Zanjani MB, Kagdi H, Bird C. Automatically recommending peer reviewers in modern code review. IEEE Trans. on Software Engineering, 2016,42(6):530−543. [doi: 10.1109/TSE.2015.2500238]

[17] Lu S, Yang D, Hu J, Zhang X. Code reviewer recommendation based on time and impact factor for pull request in Github. Computer Systems Applications, 2016,25(12):155−161 (in Chinese with English abstract).

[18] Liao ZF, Wu ZX, Wu JS, Zhang Y, Liu JY, Long J. TIRR: A code reviewer recommendation algorithm with topic model and reviewer influence. In: Proc. of the 2019 IEEE Global Communications Conf. (GLOBECOM). Waikoloa: IEEE, 2019. 1−6.

[19] Mihalcea R, Tarau P. Textrank: Bringing order into text. In: Proc. of the 2004 Conf. on Empirical Methods in Natural Language Processing. Barcelona: Association for Computational Linguistics, 2004. 404−411.

[20] Page L, Brin S, Motwani R, Winograd T. The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab, 1999.

[21] Nallapati R, Zhai FF, Zhou BW. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In: Proc. of the 31st AAAI Conf. on Artificial Intelligence (AAAI 2017). San Francisco: AAAI, 2017. 3075−3081.

[22] Liu Y, Lapata M. Text summarization with pretrained encoders. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: Association for Computational Linguistics, 2019. 3730−3740. [doi: 10.18653/v1/D19-1387]

[23] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Proc. of the 3rd Int'l Conf. on Learning Representations (ICLR 2015). 2015.

[24] See A, Liu PJ, Manning CD. Get to the point: Summarization with pointer-generator networks. In: Proc. of the Annual Meeting of the Association for Computational Linguistics (Vol.1: Long Papers). Vancouver: Association for Computational Linguistics, 2017. 1073−1083. [doi: 10.18653/v1/P17-1099]

[25] Gehrmann S, Deng YT, Rush AM. Bottom-Up abstractive summarization. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 4098−4109.

[26] Liu F, Flanigan J, Thomson S, Sadeh N, Smith NA. Toward abstractive summarization using semantic representations. In: Proc. of the 2015 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Denver: Association for Computational Linguistics, 2015. 1077−1086.

[27] Yu LT, Zhang WN, Yu Y. Seqgan: Sequence generative adversarial nets with policy gradient. In Proc. of the 31st AAAI Conf. on Artificial Intelligence (AAAI 2017). San Francisco: AAAI, 2017. 2852−2858.

[28] Fumio N, Nakano YI., Takase Y. Predicting meeting extracts in group discussions using multimodal convolutional neural networks. In: Proc. of the 19th ACM Int'l Conf. on Multimodal Interaction. New York: Association for Computing Machinery, 2017. 421−425. [doi: https://doi.org/10.1145/3136755.3136803]

[29] Pan H, Zhou JP, Zhao Z, Liu Y, Cai D, Yang M. Dial2desc: End-to-end dialogue description generation. arXiv preprint arXiv:1811.00185, 2018.

[30] Liu CY, Wang P, Xu J, Li Z, Ye JP. Automatic dialogue summary generation for customer service. In: Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. New York: Association for Computing Machinery, 2019. 1957−1965. [doi: https://doi.org/10.1145/3292500.3330683]

[31] Tao X, Zhang XX, Guo SL, Zhang LM. Automatic summarization of user-generated content in academic Q&A community based on Word2Vec and MMR. Data Analysis and Knowledge Discovery, 2020,4(4):109−118 (in Chinese with English abstract).

[32] Kyunghyun C, Merriënboer BV, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP). Doha: Association for Computational Linguistics, 2014. 1724−1734.

[33] Jiang SY, Armaly A, McMillan C. Automatically generating commit messages from diffs using neural machine translation. In: Proc. of the 2017 32nd IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE). Urbana: IEEE, 2017. 135−146.

[34] Xu SB, Yao Y, Xu F, Gu TX, Tong HH, Lu J. Commit message generation for source code changes. In: Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence (IJCAI). 2019. 3975−3981.

[35] Hu X, Li G, Xia X, Lo D, Jin Z. Deep code comment generation. In: Proc. of the 2018 IEEE/ACM 26th Int'l Conf. on Program Comprehension (ICPC). Gothenburg: IEEE, 2018. 200−210.

[36] Hu X, Li G, Xia X, Lo D, Jin Z. Deep code comment generation with hybrid lexical and syntactical information. Empirical Software Engineering, 2020,25:2179−2217. [doi: https://doi.org/10.1007/s10664-019-09730-9]

[37] Alon U, Zilberstein M, Levy O, Yahav E. code2vec: Learning distributed representations of code. Proc. of the ACM on Programming Languages, 2019,3:1−29. [doi: https://doi.org/10.1145/3290353]

[38] Ye DH, Xing ZC, Foo CY, Ang ZQ, Li J, Kapre N. Software-Specific named entity recognition in software engineering social content. In: Proc. of the 2016 IEEE 23rd Int'l Conf. on Software Analysis, Evolution, and Reengineering (SANER). Suita: IEEE, 2016. 90−101. [doi: 10.1109/SANER.2016.10]

[39] Markovtsev V, Long W, Bulychev E, Keramitas R, Slavnov K, Markowski G. Splitting source code identifiers using bidirectional LSTM recurrent neural network. arXiv preprint arXiv:1805.11651, 2018.

[40] Ferrari A., Esuli A. An NLP approach for cross-domain ambiguity detection in requirements engineering. Automated Software Engineering, 2019,26:559−598. [doi: https://doi.org/10.1007/s10515-019-00261-7]

[41] Chen H, Damevski K, Shepherd D, Kraft NA. Modeling hierarchical usage context for software exceptions based on interaction data. Automated Software Engineering, 2019,26:733−756. [doi: https://doi.org/10.1007/s10515-019-00265-3]

[42] Alreshedy K, Dharmaretnam D, German DM, Srinivasan V, Gulliver TA. SCC++: Predicting the programming language of questions and snippets of StackOverflow. Journal of Systems and Software, 2020,162:110505. [doi: 10.1016/j.jss.2019.110505]

[43] Hao R, Feng Y, Jones JA, Li YY, Chen ZY. CTRAS: Crowdsourced test report aggregation and summarization. In: Proc. of the 2019 IEEE/ACM 41st Int'l Conf. on Software Engineering (ICSE). Montreal: IEEE, 2019. 900−911.

[44] Shi CY, Xu CJ, Yang XJ. Study of TFIDF algorithm. Journal of Computer Applications, 2009,29(z1):167−170, 180 (in Chinese with English abstract).

[45] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP). Doha: Association for Computational Linguistics, 2014. 1532−1543. [doi: 10.3115/v1/D14-1162]

[46] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. arXiv preprint arXiv:1710.10903, 2017.

[47] de Boer PT, Kroese DP, Mannor S, Rubinstein RY. A tutorial on the cross-entropy method. Annals of Operations Research, 2005, 134(1):19−67.

[48] Lin CY. Rouge: A package for automatic evaluation of summaries. In: Proc. of the Text Summarization Branches Out. Barcelona: Association for Computational Linguistics, 2004. 74−81.

[49] Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn, 1992,8: 229−256. https://doi.org/10.1007/BF00992696

[50] Zhang XX, Lapata M, Wei FR, Zhou M. Neural latent extractive document summarization. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 779−784. [doi: 10.18653/v1/D18-1088]

[51] Ranzato MA, Chopra S, Auli M, Zaremba W. Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732 [cs.LG], 2015.

[52] Narayan S, Cohen SB, Lapata M. Ranking sentences for extractive summarization with reinforcement learning. In: Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol.1 (Long Papers). New Orleans: Association for Computational Linguistics, 2018. 1747−1759.

附中文参考文献:

[17] 卢松,杨达,胡军,张潇.基于时间和影响力因子的 Github Pull Request 评审人推荐.计算机系统应用,2016,25(12):155−161.

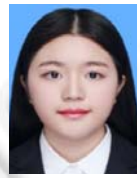
[31] 陶兴,张向先,郭顺利,等.学术问答社区用户生成内容的 W2V-MMR 自动摘要方法研究.数据分析与知识发现,2020,4(4): 109−118.

[44] 施聪莺,徐朝军,杨晓江.TFIDF 算法研究综述.计算机应用,2009,29(S1):167−170, 180.

邝砾(1982－),女,博士,教授,博士生导师,CCF 专业会员,主要研究领域为服务计算,群智软件生态系统,机器学习.

施如意(1996－),女,硕士,主要研究领域为服务计算,群智软件生态系统.

赵雷浩(1998－),男,学士,主要研究领域为服务计算.



张欢(1996－),女,硕士,主要研究领域为机器学习,数据挖掘,群智软件生态系统.

高洪皓(1985－),男,博士,副教授,CCF 高级会员,主要研究领域为软件形式化验证,服务协同计算,无线网络和工业物联网,智能医学影像处理.