计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20190625
57(3):474-486,2020

面向技术论坛的问题解答状态预测

沈明珠　　刘　辉
(北京理工大学计算机学院　北京　100081)
(3120181025@bit.edu.cn)

Status Prediction for Questions Post on Technical Forums

Shen Mingzhu and Liu Hui
(*School of Computer Science and Technology*, *Beijing Institute of Technology*, *Beijing* 100081)

Abstract　When encountered by technical problems, developers often post questions on technical forums such as Stack Overflow, and wait for satisfying answers. QA forums are also an important manifestation of Internet-based group intelligence software development. However, the questions posted in the forums may not get satisfying answers. Therefore, asking problems and passively waiting for solution is not always the best strategy. To this end, we propose a deep neural network based approach to automatically predict whether the questions can obtain satisfying answers. Knowing whether the questions can be effectively answered in advance, developers figure out the best strategy for their technical problems in advance. This approach not only takes full usage of the text information of the problems itself, but also exploits the relevant content of the inquirer of the questions. With the latest deep learning technologies, it fully exploits the intrinsic relationship between the input features and the questions' solving status. Experimental results on the dataset provided by Stack Overflow suggest that the proposed approach can accurately predict the solving status of the questions. The precision of predicting well-answered problems is 58.87%, and the recall is 46.68% (in contrast, random guess results in a precision of 38.77%, and recall of 35.26%), better than *K*NN and FastText.

Key words　group intelligence software; QA forum; status prediction; deep learning; text classification

摘　要　当遭遇技术问题时,开发人员往往会在 Stack Overflow 等技术论坛上发布问题并等待回答.此类 QA 系统也是基于互联网的群智化软件开发的一个重要表现形式.但是论坛上提出的问题并不一定能够获得满意答案.因此,提出问题并被动地等待答案并不总是最佳策略.为此,提出了一种基于深度神经网络的方法以自动预测问题能否获得满意答案.提前预知问题能否及时获得有效答复,开发人员可以提前规划应对策略.该方法不仅充分利用了问题本身的文本信息,也将提问人员相关内容作为预测的主要依据.利用最新的深度学习技术,充分挖掘输入特征与问题解答状态之间的内在关联关系.在 Stack Overflow 提供的数据集上的实验结果表明:所提出的方法能够预测问题的解答情况,结果显示在预测问题是否有满意答案的查准率为 58.87%、查全率为 46.68%(随机猜测的查准率为 38.77%,查全率为 35.26%),并优于机器学习 *K*NN 和浅层神经网络 FastText.

关键词　群智化软件;社区问答;状态预测;深度学习;文本分类

中图法分类号　TP311.56

收稿日期:2019-08-30;**修回日期**:2019-12-13

基金项目:国家自然科学基金重大项目(61690205)

This work was supported by the Major Program of the National Natural Science Foundation of China (61690205).

通信作者:刘辉(liuhui08@bit.edu.cn)

随着软件开发技术的飞速发展,软件开发知识量也越来越大.因此,即便是高级程序员也很难掌握所有的软件开发知识.在碰到技术难题时,程序员常用的策略之一是在诸如 Stack Overflow 等技术社区上请求帮助,从而有效减少时间消耗[1].

技术社区问题的解决主要依赖于互联网上的其他程序员[2].所以,基于互联网的群体智能是解决程序员面临的难题的关键所在.社区问答系统是基于互联网的群智化软件开发的一个重要组织形式.

然而,在技术社区上提出的问题并不一定会获得满意(有效)的答案.因此,提问之后被动等待答案可能并不总是最佳的应对策略.为此,本文提出了一个基于深度学习[3]的问题解答状态预测方法,根据问题的文本信息和用户特征去分析在论坛发布的问题是否能够及时获得满意答案.技术人员在碰到技术难题时,需要根据成本与风险从众多可选的技术资源中选择一个或几个最合适的资源进行求助.这些技术资源包括技术论坛、同事、团队领导、公司技术专家等.其中团队领导和公司技术专家一般日程安排紧张,只有其他途径无法解决的时候才会转而寻求他们的帮助.技术人员在发布问题时可使用本方法预测,如果发现不能获得满意答案,提问者可以及时转向其他更可靠(但通常也更昂贵)的技术来源以寻求帮助,比如向团队领导或者公司技术专家,甚至外部付费的咨询公司等寻求帮助.提问者也可以选择更改问题标题内容、更换问题类型标签等方法,从而改善问题质量,提高问题被解答的可能性.

对社区问答的研究是目前群智化软件开发领域的一个热点.现有研究主要集中于社区问题内容学习分类[4]、问题质量评估[5]、满意答案推荐等.但是目前尚未出现针对问题解答状态的预测方法.

本文的主要贡献有 2 个方面:

1) 提出了一种基于深度神经网络的方法,通过问题标题文本、用户信息等问题特征来预测问题是否能获得满意答案.

2) 实现了提出的方法并基于 Stack Overflow 的真实数据进行了实验验证.实验结果表明该方法可以有效预测技术问题的解答状态,其性能显著高于随机猜测,并高于 KNN 与 FastText.

1 相关工作

1.1 社区问答

社区问答网站内容的分类与分析是目前国内外学者的研究热点[6].文献[7]研究的是通过问题答案的来源来进行质量评估,越是权威网站提供的答案,令人满意程度越高;文献[8]提出现在每天提出的众多问题并未被发送到适合回答它的用户那里,因此导致新问题不能够被及时回答.于是综合用户在技术论坛的历史问答数据中的活跃程度与用户权限和其参与的问题构建用户简档,从而进行专家推荐;文献[9]则是预测问题是否会被关闭.由于论坛问题过多,并且问题本身不能保证质量(比如问题重复、个例化、问题没有建设性、或者不是真正的问题等等),论坛通常会让用户对一个问题是否要被关闭进行表决投票.当支持率达到一定值时该问题就可以被关闭.于是该实验提出可以根据这种特性来预测问题是否会被关闭.许多实验根据非文本特征比如问题长度、用户年龄、问题标签数量等特征来进行分类并预测;文献[10]则是从问题本身质量出发,去预测问题获得的分数.其中使用了 spearman 秩相关系数去测试不同特征与问题分数之间的依赖性,并主要研究了 4 个有高相关性的变量,有问题浏览次数、答案数、满意答案的分数、问题的赞同数.通过 SPSS 进行分析,以了解关系因变量与那些相关系数低的因素之间的关系,总共选择了 16 个变量来说明对问题分数影响的原因,并选出了上述 4 个影响最大的变量,但是这 4 类的得分在统计上有所不同.该文在最后提出可以学习这些因变量之间的共同点,通过运行基于某些规则的分类器来实现它们之间的特征区分.

社区问答中关于自动问答推荐的研究也非常多,如何从多个候选答案中识别推荐出满意答案是现在社区问答发展的方向之一.文献[11]则使用支持向量机、决策树、朴素贝叶斯等 3 种算法来去预测用户满意度,首次提出了关于用户“个性化”需求的概念,但是该预测是在已回答的答案中判断是否能够满足用户提出者;文献[12]同样通过答案的浅层语言文本特征如最长句子长度、平均句子长度、单词的单词长度和用户特征进行比对,从而显著对比出满意答案与一般答案之间的不同点.满意答案文本内容会更长,会将常用词汇转为生僻的单词,所包含的单词也会更长,但是仅凭这些语言特征去预测效果并不是非常高.该文使用了 10 折的交叉验证在众多技术论坛上进行比对,发现不同技术论坛的语言区别较大,因此选择合适的特征去进行预测非常重要.

随着数据量的不断上升,单纯的使用统计工具与线性分析已经不能满足人们的需求,深度学习技

术已经开始运用于社区问答中;文献[13]研究的是图像类问题答案对.通过识别能力现在可以与人类视觉能力相媲美的深度卷积神经网络进行目标识别,其效果在诸如 imagenet large scale visual 等基准测试中较优,可回答比如"这个图像中的主要对象是什么"一类的问题;建立系统给图像和基于文本的问题,是解决识别主导对象或活动之外的图像的一种自然方法[14],最终输出一个基于文本的答案,这被称为开放式视觉问题解答(VQA)问题.它要求将计算机视觉与自然语言处理相结合.实验描述了一个基于贝叶斯框架进行答案类型的预测模型,其中使用了名为 skipthought 的矢量,以某种方式将句子编码成矢量句子并保留显著语句信息,在多个公开可用的 VQA 数据集上进行测试.

综上所述,国外关于使用 Stack Overflow 技术论坛数据进行分析研究较多,例如问题质量评估等领域.现有工作大多研究预测问题内容质量、答案内容质量的评分,或者基于用户的问题内容的解答专家的推荐,但对于问题的解答状态的预测工作尚未展开,因此本文提出一种方法对问题解答状态进行预测.

1.2 文本分类

文本分类问题在自然语言处理(Natural Language Processing, NLP)领域占据着重要地位.它创立在 20 世纪 50 年代,随着专家系统的建立,文本分类有了新的进步.但这种方法不仅费时费力,而且覆盖的范围和准确率都非常有限.随后发展到 90 年代时,出现许多能够解决大规模文本的文本分类方法,下面是 3 种常用的分类算法:

1) Rocchio 算法.Rocchio[14]方法是情报检索领域最经典的算法[15].该算法的基本思路是通过计算同一个类别里的样本文档,得到新向量,该向量是该类别最具代表性的向量表示.将给出的测试文本对其进行判断时,比较新文本与这个中心向量的相似度,判断向量之间距离,进而可以确定新文本属不属于该类别.Rocchio 算法被改良之后不仅判断属于这种类别的文本(正样本),也判断不属于这个类别的文本数据(负样本).这种算法比较简单,但是对错误数据毫无抵抗力,无法包容数据噪声[16].

2) K-近邻算法 KNN.KNN[17]方法是先给定待分类文本,再计算待分类文本与训练样本集中各个样本的文本相似度.根据计算结果找出 N 篇与待分类文本距离最近最相似的文本,根据这 N 篇文本所属的类别判定待分类文本的所属类别,是一种基于实例的分类方法.这种判断方法很好地克服了 Rocchio 算法中无法用一条直线准确划分 2 类数据的缺陷,但是比较过程的代价较大.

3) 支持向量机(support vector machine, SVM)方法.样本数据较大时使用 SVM 方法[18],它是由 Cortes 等人于 20 世纪 90 年代提出的,SVM 利用了统计学习理论的 VC 理论.并且利用结构风险最小化原理,在文本分类方面可以实现降维和分类.SVM 学习的是压缩成的有限数量的信息,为使泛化能力达到最优,就要兼顾模型复杂性与模型的学习能力.

由于文本表达的形式非常麻烦,表达成矩阵形式时维度又高,特征表达能力很弱,同时神经网络不适应于处理稀疏矩阵表达的数据,因此如何解决文本表示变为一大难题.

Google Mikolov 的文献[19]提出一个工具包 Word2Vector,促进词分布式的发展,使用 Word2Vector 工具包能够对语义进行较全的保留效果,极大地推进了文本分析的进程.但分布式表示很早就被提出,能够将每个词表达成实数向量.碍于神经网络数据的需求性,这个实数向量需要有适当的维度,不能稀疏又要连续,也就是词嵌入(word embedding).词向量的提出将文本从神经网络难以处理的方式,变成了连续稠密数据,这种数据形式类似图像、语音,是训练语言模型的附加产物.再利用卷积神经网络、递归循环神经网络等网络结构自动获取特征表达能力.深度学习逐渐在自然语言处理上取得令人瞩目的研究成果,其理论能够很好地应用于文本分类当中,其中最新最令人关注的循环神经网络主要解决了如何处理时间序列的变化.

虽然卷积神经网络、递归循环神经网络用于文本分类时结果非常较好,但是有一个不足的地方就是其表达的结果不能让人直接理解、解释起来也不容易,尤其是在分析坏测试案例时尤其麻烦.注意力(attention)机制[20]是 NLP 领域常用的建模长时间记忆机制,其基本思想就是目标语言端的词往往只与源语言端部分词有关.可以很直观地给出每个词对结果的作用,非常适用于 Seq2Seq 模型.Bahdanau 等人使用双向 RNN(bidirectional RNN),成功使得一个词的隐层状态不仅压缩了其前面的词的信息,还压缩了后面的词.更加关注这一个词语周边的词,使得 RNN 能更好地表达当前的输入.结果证明,引

入 attention 能够解决不同长度的源语言句子都用相同固定维度的压缩向量表示所带来的性能瓶颈,其鲁棒性[21]更好.

有别于使用传统机器学习的文本分类方法,本文将使用最新的深度神经网络技术,进而挖掘更加复杂与潜在的联系.本文在学习问题标题的短文本信息的同时,也利用用户特征进行分析.在训练集的收集方面,本文使用自动化的方法去解析获取社区的数据集,并对获取后的问题特征进行标注.

2 面向技术论坛问题解答状态的预测方法

本节详细介绍面向技术社区的问题解答状态的预测方法.其中,2.1 节给出本文所提出方法的概览介绍,之后的各小节将详细介绍该方法的各个关键步骤.

2.1 方法概述

本文提出的面向技术论坛的基于深度学习的问题解答状态预测方法如图 1 所示.

1) 利用 Stack Overflow 论坛的有关问题、用户的多个存储文件作为语料库,经过分析处理将数据转化为文本型特征与数值型特征.

2) 根据问题是否有满意答案的标志和问题回答的数量对内容进行标注,对文本特征进行还原词根、删除停用词和标点分词等预处理;然后使用 Word2Vector 进行训练生成词向量,将生成的词向量与数值型特征作为神经网络的输入.

3) 抽出有满意答案标志的问题作为正样本集合,其余问题随机抽出作为负样本集合.分类器的预期输出为样本的标注标签也就是本文的预测内容(即问题是否会获得满意答案).分类器经过多次迭代训练后可以获得最终训练好的深度学习分类器.

4) 通过训练好的分类器,输入给定的待预测的技术问题的相关特征,得到对于提出问题的解答状态的预测结果.

实现细节将在 2.2,2.3 节中进行详细介绍.

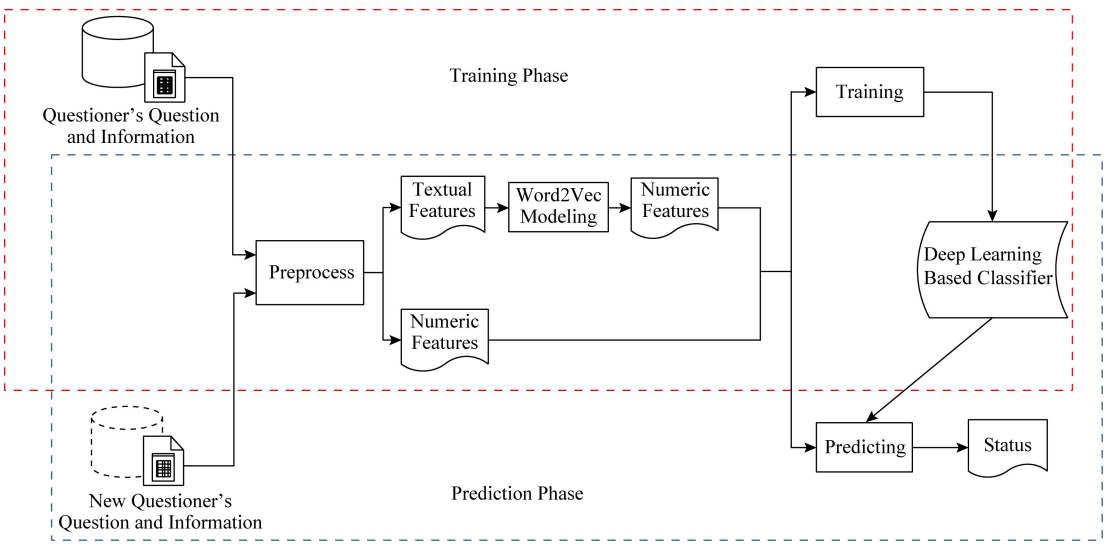

Fig. 1 Overview of prediction methods for answer status of technical forum question

图 1 技术论坛问题解答状态的预测方法概述

2.2 神经网络的输入

由于将有关问题的全部内容和提问者等相关信息直接输入深度神经网络分类器会造成模型学习难度过大,且问题描述内容本身过长,在训练过程中极易造成数据损失.因此我们需要对输入内容进行选择与预处理操作.从语料库中舍弃一部分对分类器训练过程无价值的相关特征,从而降低神经网络模型的构建与训练难度.

同时为防止因特征过多而导致的维度爆炸问题,本文经过多次比较与考量,选取了与技术问题解答状态预测相关的问题特征作神经网络分类器输入的一部分,而文本特征只提取了技术问题的标题.

尽管已经删除了一定的无关的问题特征,但这些提取特征之间的关联关系以及输入特征和输出预测结果之间的潜在映射关系仍然需要进一步分析.因此,我们将提取后的原始特征输入分类器进行映射和学习,从而使最终输出和问题样本标签的分类结果尽量相似.

从第 1 部分中我们可以知道,非文本特征对于问题状态影响很大,因此除问题标题外我们还综合

了6个文本和非文本特征,从不同角度体现技术问题的特性,并且参考提问者的信息,从而能够在问题内容类型、包含代码、用户贡献等方面更全面的表现问题的特征.

表1展示了所选问题特征的详细信息.需要声明的是,这里的tag并不是文本形式,而是长度为50位的全0数组.经过对文本的分析我们选取了出现频率最高的前50位的问题标签(表明了问题所属的类型,如Java,Python),当问题用户标注标签时,数组中对应位置上的值则置为1,未出现则继续设置为默认值0.选取前50位的原因是经过对问题标签的分析,Stack Overflow的库中共存有上万个标签,极大多数标签只出现了一次,因此选取指定样本区间中至少出现1万次以上的标签,过长的问题类型标签同样容易引起维度爆炸,最终定为50.

Table 1 Features of Selected Technical Questions

表1 选取的技术问题特征

Feature Type	Name	Description
Text Feature	Title	The question's title
	Code	The question whether contains code(the value is 0 or 1)
	Tag	The question's tag
Non-text Feature	Creation	Time raised by the question (indicated by the day of the week)
	Reputation	Reputation value of users given by the technical community
	Upvotes	The support of this user's questions and answers
	Downvotes	The unsupported of this user's questions and answers

表1中除了title作为文本型数值输入,其余特征输入形式都为数值型,这些特征共同组成了本文所提出方法的输入:

$$input=\langle title, numeric\_features\rangle, \tag{1}$$

$$numeric\_features=\{code, tag, creation, reputation, upvotes, downvotes\}, \tag{2}$$

其中,$numeric_features$是所有数值型问题特征的集合,共55个值,则是每个问题的标题文本,也是本文方法中唯一字符型的输入.

2.3 问题特征的表示方式

为了能够探索发掘出问题标题文本内容的深层语义关联与含义,本文使用了Mikolov等人提出的著名的词向量化模型Word2Vector,将标题中的词语映射到高纬向量空间[22],以词向量在高维空间中的分布来揭示词与词之间的相似性关系.作为自然语言处理领域的重要工具,Word2Vector构建了一个以给定的文本作为输入输出的神经网络.在进行训练之后,可以利用此模型的隐含层将词语转化为稠密向量,实现以向量相似性来表示语义相似性的目的.

我们利用大量的经过基本文本预处理的问题标题作为语料库,对Word2Vector模型进行训练,构建了一个针对技术社区问题标题文本的向量空间.该过程根据3个步骤对神经网络分类器输入分别进行预处理.

1) 去除标题中的谓词与冠词,对问题标题进行分词,将问题拆分为多个逻辑单字.

2) 将逻辑单字使用nltk语言包进行统一小写和词根还原的操作.

3) 利用已训练好的Word2Vector模型将各逻辑单字分别映射为高维空间中固定长度(200维)的词嵌入向量(word embedding).

由于程序语言中涉及符号标志较多,例如C++、C#等语言中包含的符号,因此在预处理过程中只针对点号、逗号、冒号、感叹号、疑问号共5种用于断句的符号进行删除,其余符号则不予处理.

2.4 基于深度神经网络的分类器

本文所提出的基于深度神经网络的分类器结构如图2所示:

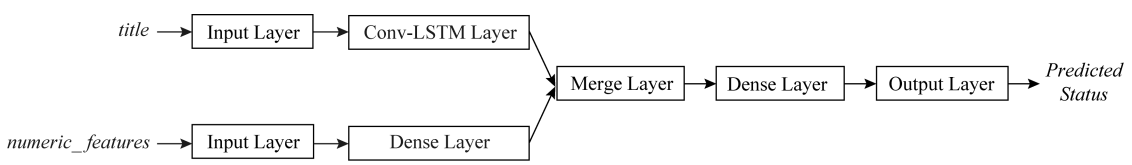

Fig. 2 Classifier based on neural network

图2 神经网络分类器

如 2.2 节所述,本文分类器的输入共有 2 部分,分别为文本输入与数值输入.文本输入的内容是提出问题的标题,标题文本在经过预处理之后(详见 2.3 节),会从文本信息形式转换为数值形式,以词向量矩阵的形式(输入大小为 20×200)经过输入数据屏蔽层之后进入 Conv-LSTM 层.此处长度 20 是考虑到问题标题长度通常不会过长,经过查看 Stack Overflow 对于标题长度的限制与标题长度的统计得出该数值.

Conv-LSTM 层是由卷积神经网络(convolutional neural networks, CNN)的卷积层(convolutional layer)与长短时记忆网络(long short-term memory, LSTM)构成. 将 2 种神经网络结合运用之后,不仅具有 CNN 的刻画局部特征的能力[23],还有 LSTM 的时序建模能力[24],空间和时间上特性都具有[25],使自然语音处理技术得到充分利用.具体结构如图 3 所示:

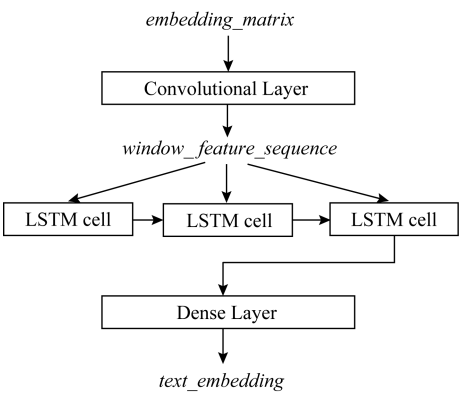

Fig. 3 The structure of Conv-LSTM layer

图 3 Conv-LSTM 层结构

本文中的 CNN 进行一维卷积的对象是词向量,设在句子中第 i 个单词的 200 维词向量为 $\boldsymbol{x}_i$,输入长度为 L 的句子的向量为 $\boldsymbol{X}$,k 为 filter 的长度,m 为一个进行卷积操作的 filter,filter 在句子 j 位置卷积形成的 k 个连续词向量长度的窗口向量为 $\boldsymbol{w}_j$.

卷积后的窗口向量:

$$\boldsymbol{w}_j=[x_j,x_{j+1},\cdots,x_{j+k-1}], \tag{3}$$

其中,逗号表示行向量连接,进行卷积计算操作的 filter 是具有多个指定长度,分别是 3,4,5.同一指定长度卷积一次.这种方法将原始词向量序列卷积之后变得更为抽象,但是对原句子的编码还是使用 LSTM 网络.

Conv-LSTM 层上卷积操作的对象是 200 维的词向量,卷积窗口在由数据生成的词向量矩阵序列上进行滑动卷积操作,获得不同位置的特征,即多个 feature map,设为 c_j.通过映射操作把卷积后内容相同的特征向量依次排列放在同一序列中,设为 W.计算方法:

$$c_j=f(\boldsymbol{w}_j\circ m+b), \tag{4}$$

$$W=[c_1;c_2;\cdots;c_n], \tag{5}$$

其中,$\circ$ 表示的是矩阵元素相乘,映射函数 f 使用 ReLU[26],b 表示偏项,分号表示列向量连接.

卷积后的窗口序列与输入数据的序列相对应,因此本文模型中的卷积操作并不会影响文本内容中的顺序.窗口向量序列按顺序输出向量到下一层的 LSTM.LSTM 的实际输入结果其实是 CNN 中间层的隐含输出层的内容.LSTM 的输入结果不能经过 max-pooling 池化层,因为池化层用于卷积后的特征映射选择出最重要的特征,映射选择功能并不是连续按顺序选择特征,不能保证池化后的语句能保持原始顺序让模型进行语义学习,影响 LSTM 序列学习.因此在卷积运算后就不会进行 pooling 操作.

数据从 LSTM 层学习输出后,进入到全连接层,全连接层激活函数为 tanh 函数.之后,2 部分数据会经由合并层(merge layer)以向量拼接(concatenate,$axis=-1$)的形式进行合并;输出层的激活函数为 sigmoid 函数,最终选取的模型损失函数(loss function)为 binary_crossentropy,优化器(optimizer)选择为 adam 自适应方法,迭代次数 $epoch=12$,批尺寸 $batch\_size=128$.

分类器是在有监督的情况下对输入进行迭代训练,从而训练出模型的最优参数.

3 实验验证

在本节中,我们基于 Stack Overflow 的真实历史数据对本文提出的方法进行实验验证.

3.1 研究问题

在实验验证阶段,我们希望通过分析 4 个问题来对所提出的方法进行评估.

1) RQ1.该方法是否能够准确有效对社区技术问题的解答状态进行预测? 其查全率和查准率是否优于现有方法?

2) RQ2.所提出的神经网络的 2 个特征输入(标题文本特征与数值特征)对最终结果分别有什么影响? 即如果只有其中一个特征输入,分类器的性能会如何变化? 数值特征中的各项特征值又会对分类器的性能有什么影响?

3) RQ3.利用其他网络模型(如卷积神经网络CNN、长短时记忆神经网络LSTM)替代神经网络分类器中所使用的组合神经网络,是否会影响分类器的性能,如查全率、查准率?

4) RQ4.训练集与测试集的数量规模是否会很大程度上的影响分类器的性能?即如果训练集如果数量较多或较少,能否进行准确的预测?

研究问题RQ1关注的是所提出的深度学习方法与传统机器学习方法的预测结果在查准率(*precision*)与查全率(*recall*)等指标上的区别.为了回答这个问题,我们选择了KNN[17]和FastText[27]作为对比方法,KNN是典型的基于传统机器学习的文本分类方法,而FastText是非深度模型的神经网络模型,在训练速度远高于深度学习的同时也保证训练的质量.除此之外,我们还与随机猜测进行对比.随机猜测方法首先统计训练集中正样本的出现概率p,然后以恒定的概率p将测试数据预测为正样本(能获得满意答案).

研究问题RQ2关注神经网络分类器输入特征选取的有效性.我们在保持模型其他部分不变的情况下,分别删除原模型中的问题标题文本及特征值中的不同位置,继续将原模型加以调优并训练.以各分类器在同一测试集上的各项指标作为衡量指标来分析所提出的各个特征分别在整个方法中所起的作用.

研究问题RQ3主要关注在本方法所构造的神经网络分类器中文本特征的处理效果.我们通过将所提出的网络模型中的组合神经网络分别替换为长短时记忆网络和卷积神经网络,并同样以各分类器在同一数据集上的最优平均$F1$值作为指标来帮助考察和分析在已有的方法架构下3种神经网络对于最终结果的影响.

研究问题RQ4则关注训练样本集和测试样本集的内容质量因素,使用不同数量规模的问题样本集进行训练和测试在该分类器模型上是否会有影响.与其余文本分类问题不同,之前不能有满意答案的问题可能随着技术的发展变得可被解答,这种情况会扰乱分类器的训练,因此本问题对样本集其数量范围的影响进行探讨.

3.2 实验过程

由于训练样本是由Stack Overflow提供的数据集获取而来,内容量复杂,问题数量达到上亿个,直接放入到神经网络里训练会使模型训练难度极具增大,并且提出的问题能够获得满意答案的几率并不高.按照本文设定的正负样本集合,也就是按照是否有满意答案标注正负样本,样本比例很可能不均匀,从而影响分类器的训练效果.因此我们以2017年的Stack Overflow社区的问题为样例参考,对问题有满意答案和无满意答案的内容进行统计,如表2所示.其中表2列1是问题的发布时间(月),列2是问题提出的总数,列3和列4分别是有满意答案与无满意答案的数量,列5是有满意答案的百分比.

Table 2 History Data from Stack Overflow in 2017

表2 2017年Stack Overflow历史数据

Question Time	$10^{-3}\times$Total Number of Questions	$10^{-3}\times$Number of Question Resolved	$10^{-3}\times$Number of Open Questions	Solved Probability/%
Jan	101.9	41.8	60.1	41.02
Feb	101.7	41.1	60.6	40.41
Mar	117.7	46.1	71.6	39.17
Apr	105.0	40.7	64.3	38.76
May	109.8	42.2	67.6	38.43
Jun	105.1	39.9	65.2	37.96
Jul	105.3	40.4	64.9	38.37
Aug	103.7	39.9	63.8	38.48
Sep	95.8	35.8	60.0	37.37
Oct	100.8	36.6	64.2	36.31
Nov	111.8	33.6	78.2	30.05
Dec	109.3	34.5	74.8	31.56
Total	1 267.9	472.6	795.3	37.27

我们选取指定年份的问题作为语料库,逐条生成固定格式的正负样本数据集(详见 2.3 节),构建分类器的训练集,每个月以 1 周为时间跨度获取 2 万条,并抽取该月后面的 2 000 条问题作为测试样本.每个训练集对应当月抽取的测试集进行测试.具体选取数量原因可见研究问题 RQ4.所提模型代码基于 Tensorflow 实现,实现评估的评价指标有查准率(*precision*)和查全率(*recall*)以及 $F1$ 值.计算为

$$precision=\frac{TP}{TP+FP} \tag{6}$$

$$recall=\frac{TP}{TP+FN}=1-\frac{FN}{TP+FN} \tag{7}$$

$$F1=2\times\frac{precision\times recall}{precision+recall} \tag{8}$$

TP 表示 True Positive,即做出 Positive 的判定,而且判定是正确的,其数值表示正确的 Positive 判定的个数,剩余同理.请注意,此处评估指标的目标为正样本,即正样本被判断对是 TP.

3.3 RQ1:优于现有方法

为回答研究问题 RQ1,我们总结了本方法与 KNN 以及 FastText 方法在相同测试集上的问题解答状态预测结果,并使用随机猜测的方法证明该方法的有用性.随机猜测是假如训练集中正样本概率为 p,那么就以 p 的概率预测测试样本为正样本,正样本的概率 p 值设定由表 2 可得,设置为 37%.

我们共使用了 3 个月的数据集分别进行测试.对比方法是将特征合并成 1 条总体特征输入到分类器中进行训练与预测.

结果如表 3 所示.其中,列 1 为预测问题的发布时间,列 2 为本方法的测试结果的查准率、查全率和 $F1$ 值.列 3 为使用 KNN 方法的查准率、查全率和 $F1$ 值,列 4 为 FastText 的查准率、查全率和 $F1$ 值.列 5 为随机猜测的查准率、查全率和 $F1$ 值.

Table 3 Results on Status Prediction for Post Questions

表 3 问题解答状态预测的结果 %

Question Time	Proposed Approach			KNN			FastText			Random Guess		
	precision	*recall*	$F1$	*precision*	*recall*	$F1$	*precision*	*recall*	$F1$	*precision*	*recall*	$F1$
Jan	58.34	46.13	51.52	42.86	36.51	39.43	54.62	45.21	49.47	39.13	34.62	36.74
Feb	57.24	46.63	51.39	43.31	31.25	36.30	55.34	41.06	47.14	38.47	35.88	37.13
Mar	61.03	47.28	53.28	42.57	32.48	36.85	59.46	46.83	52.39	38.71	35.29	36.92
Average Value	58.87	46.68	52.06	42.91	33.41	37.53	56.47	44.37	49.67	38.77	35.26	36.93

根据表 3 我们可以看出:

1) 本文所提出方法的平均查准率比 KNN 提高了 15.96%(为 58.87%−42.91%),比 FastText 提高了 2.40%(为 58.87%−56.47%),查全率和 $F1$ 值也都有所提高.

2) 与随机猜测相比,本文查全率和查准率分别提高了 11.42%(为 46.68%−35.26%)和 20.10%(为 58.87%−38.77%).

除此之外,我们还使用单因素方差分析法(One-Way ANOVA)对本方法的性能提升的 $F1$ 值进行显著性分析,设置 $\alpha=0.05$,图 4 显示了本文方法与不同分类方法分类效果的 ANOVA 结果.结果显示 $F=69.585\,7$,P-value 接近于 0 远小于 0.05,表明不同分类方法之间有着显著的差异.

在设置相同的情况下我们也对 Precision 和 Recall 值分别进行了单因素方差分析,F 值分别为 108.426 3 与 30.190 07,其 P-value 分别为 8.091 29E−07 和 0.000 103. ANOVA 分析结果表明,不同方法的处理对查准率和查全率有显著影响.

我们从上述结论可得出,在技术问题的解答状态的预测能力上,本文提出的深度学习方法在总体效果上优于随机猜测,并略优于传统机器学习 KNN 和浅层神经网络 FastText.

3.4 RQ2:输入特征的影响

针对研究问题 2,我们设计了一组对比试验来考察所提出方法中的各个特征分别对于问题解答状态预测结果的影响.通过神经网络的拆分与训练数据集中不同特征数据的删除的方法,由表 1 可得 title,code,tag 都属于文本特征,用户特征 reputation 等属于非文本特征.本实验使用 3 个月的数据,从其平均值中直观地查看 7 个特征的作用.如图 5 所示.图 5 纵坐标代表未输入神经网络训练的特征,Default 则表示输入了全部特征.根据图 5 可以看出:

1) 删除任意输入特征都导致方法性能的降低.

2) 文本型输入的影响效果最大,删除文本型输入 title 导致方法的查准率会大幅降低,其降幅高达 50.83%(为 58.87%−8.04%).

3) 删除 tag 和 code 文本特征时,查准率会分别降低 6.49%(为 58.87%−52.38%)和 4.43%(为 58.87%−54.44%).而删除非文本特征查准率降幅最高为 3.20%(为 58.87%−55.67%).

4) 问题提出者的声誉值及评价会对查全率有较大的影响.删除该输入导致查全率降低了 6.03%(为 46.68%−40.65%).

Groups	Count	Sum	Average	Variance
Proposed Approach	3	1.561 9	0.520 633	0.000 111
KNN	3	1.125 8	0.375 267	0.000 279
FastText	3	1.49	0.496 667	0.0006 92
Random Guess	3	1.107 9	0.369 3	3.81E−06

(a) Summary

Source of Variation	SS	df	MS	*F*	P-value	Fcrit
Between Groups	0.056 703	3	0.018 901	69.585 7	4.5E−06	4.066 181
Within Groups	0.002 173	8	0.000 272			
Total	0.058 876	11				

(b) ANOVA

Fig. 4 *F*1 ANOVA analysis

图 4 方差分析 *F*1

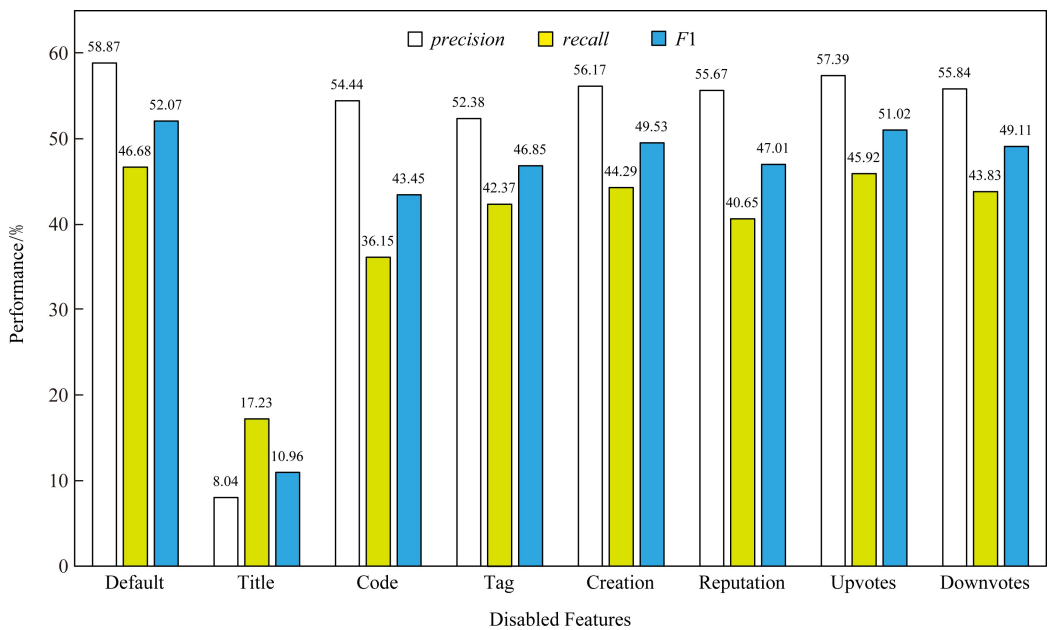

Fig. 5 Influences of different input features

图 5 不同输入特征的影响

从上述结论可得,各个问题特征对于解答状态的预测指标都有着提高作用. 其中文本特征对查准率的影响大于非文本特征对查准率的影响,用户特征对方法查全率影响较大.

3.5 RQ3:神经网络模型的影响

为了回答研究问题 RQ3,我们分别将卷积神经网络、长短时记忆神经网络以及组合网络这 3 种网络模型运用于分类器中的文本特征处理环节,各分类器经过调优后在同一测试集上的具体表现如表 4 所示.需要注意的是,表 4 中 3 种分类器除文本特征提取环节所用模型不同外,网络其余部分都保持一致.其中列 2 使用卷积神经网络和长短时记忆网络依次处理文本特征信息,列 3 只使用 2 层长短期记忆网络(同组合网络中深度相同)处理,列 4 分类器

则采用卷积神经网络进行文本提取.

由表 4 可以看出:

1) 本文方法的平均 $F1$ 值与 LSTM 相比,高出了 1.58%(为 52.06%−50.48%),其平均查准率的涨幅达 1.43%(为 58.87%−57.44%).

2) 本文方法的平均 $F1$ 值与 CNN 相比,高出了 2.52%(为 52.06%−49.54%),其平均查准率的涨幅达 2.33%(为 58.87%−56.54%).

设置 $\alpha=0.05$,对 $F1$ 值进行单因素方差分析,其 F 值为 6.561883,大于临界值 Fcrit 为 5.143253;其 P-value 为 0.030884 小于 0.05,因此不同神经网络之间提取能力差异较明显.

Table 4 Influence of Deep Learning Models

表 4 深度学习模型对预测性能的影响 %

Question Time	Proposed Approach			LSTM			CNN		
	precision	*recall*	*F*1	*precision*	*recall*	*F*1	*precision*	*recall*	*F*1
Jan	58.34	46.13	51.52	56.26	43.73	49.21	55.83	42.20	48.07
Feb	57.24	46.63	51.39	56.39	45.36	50.28	54.85	44.71	49.26
Mar	61.03	47.28	53.28	59.67	45.98	51.94	58.94	45.38	51.28
Average Value	58.87	46.68	52.06	57.44	45.02	50.48	56.54	44.10	49.54

从上述结论可得,组合神经网络对问题文本的提取能力会高于单种神经网络.针对该情况,我们认为选择卷积神经网络与长短时记忆网络的组合(Conv-LSTM)作为整个神经网络中的代码文本特征提取层时,问题特征提取的效果最好.

3.6 RQ4:训练数据的规模对性能的影响

针对研究问题 RQ4,我们对分类器训练集的最佳训练数量进行了评估.从表 2 可以看出,1 个月的问题数量大约为 10 万左右,因此我们以 1 月的数据为范围,分别抽取不同数量的训练集进行训练,抽取数量按照不同的时间跨度为半天、半周、1 周、半个月、1 个月的范围获取并进行测试.从图 6 可以看出:

1) 查准率可根据数量提升而提高,涨幅最大时(2 000 条增长为 2 万条)可达 30.15%(为 58.34%−28.19%).

2) 训练样本过多反而会造成查准率的降低,数量分别为 2 万条和 10 万条时分类器的查准率差值可达 6.61%(为 58.34%−51.73%).

从上述结论可得,训练集的数量对于评估指标影响较大,训练集数量达到一定数值时可提高准确性,训练集范围在 20 000～50 000 条时训练效果最佳.针对该情况,因此我们选择每个月的 20 000 条作

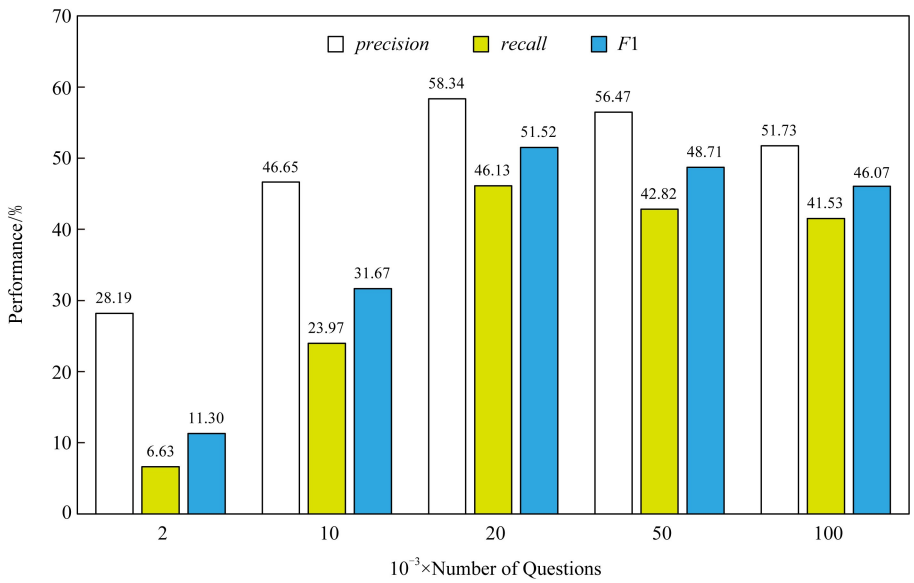

Fig. 6 Influences of training data size

图 6 训练数据的规模对性能的影响

为训练样本,既可以减少数据预处理时间,也能保证查全率、查准率的数值最高.此外,经过人工分析,数量超越一定值时查准率下降的原因是由于训练集之间时间跨度较大,内容差异性较大,致使神经网络学习效果降低,并且文本数据中存在较多数据噪音,如低质量(标题文本过短、内容简单)问题过多,致使神经网络无法从文本内容中学习到有用信息.

3.7 有效性威胁

3.7.1 外部有效性威胁

技术社区问题所在的地域不同、面向群众不同,会造成技术论坛的使用语言也不同,因此技术论坛可能会有英语、中文等.语言的不同不仅会对问题特征提取产生影响,有可能提取到错误的信息,也会对神经网络理解文本信息产生威胁,因此我们选取了使用范围最广的英语作为分类器主要分析的文本语言,使用英语的技术论坛最多,也保证了本文模型可以适用于更多的技术社区的问题内容.同时编程语言或文档多以英文描述为主,自然语言处理技术在英文处理上更为成熟,从而尽量降低地特征提取和神经网络学习文本信息过程中产生的误差,保证分类器的性能.

其次,虽然选择了以英文为描述语言的技术社区,但因为论坛数量很多,本文只选择了使用人群极多的 Stack Overflow 技术论坛进行验证,并没有对所有英文论坛的数据进行测试,这也是对分类器有效性的威胁.因为本文提出的方法除问题标题外,还有问题特征如 Reputation 等,如果其余技术论坛无类似特征,那么会对本文所使用的方法特征提取与最后的问题解答状态预测结果也会导致本文结论出现偏差,使得本文结论不适用于其余技术社区.

3.7.2 内部有效性威胁

对于评估有效性的威胁在于用来验证实验结果的数据集并没有涵盖所有 Stack Overflow 论坛的问题,选取的问题的某些特性可能会使结论产生偏差,从而导致所得结论不适用于其余论坛问题.为了减少这一威胁,我们选取了 2017 年的数据作为测试与验证,可代表论坛一个长周期内的数据,以期减少某些问题的特定关联对于验证结果造成影响.

此外,由于本文的数据集是自行标注,根据问题的回答数量和问题的满意答案标志来生成问题标签,因此如果数据集中有错误,就会造成训练数据的标签出错,进而影响深度神经网络模型的学习效果,使结论产生偏差.此外,如果时间跨度过大,可能会造成相同问题不同解答情况即数据标签会出错的情况.为降低该误差的影响,本文在样本抽取过程中采取随机抽样的方法,并尽可能的增大样本数量从而降低单条样本错误对分类器训练的影响.

4 讨论与总结

4.1 讨 论

我们可通过对样本质量的改善从而提高模型的泛化能力[28],因此对于样本中可能出现的样本噪音,我们所提出的方法对神经网络的文本会进行预处理.同时为增加神经网络的鲁棒性[29],在神经网络设计过程中,会使用一些防过拟合[30]手段来减少样本噪音对模型训练过程的干扰[31].

由于技术论坛社区问题的时间跨度较长,实验使用长度为 1 年的数据进行训练预测时,查准率只有 0.34 左右.经过分析,是由于向量空间中距离相近的句子其标签不一致,原因是旧期不能解答的问题随着技术的发展而能够得以解答,会出现相同问题但解答状态相反的情况,使得相同问题标签不同影响训练效果,产生数据噪音.如果时间跨度过大,文本数量过多,反而会降低分类器的效果.

此外,问题的解答状况与社区中人员,尤其是专业技术人员有很大关联,文献[32]中显示 Stack Overflow 上 10%的“专家”解决了 54%的问题,并提供了 60%的最佳答案,因此这些人员的流动性以及成长性可能也会对技术论坛的这一解答能力造成影响.为降低该影响,实验在选取数据时选取了近期的、连续的、跨度较短的时间段,从而降低专家流动的影响,并且将提问者专业程度的 Reputation 加入到分类内容中.

4.2 总 结

技术社区论坛已经成为程序员解决技术难题的一个重要渠道.提前预知技术问题的解答情况有助于程序员准确制定最佳的应对策略.为此,本文提出了一种基于深度学习的预测方法,能较为准确地预测所提出的问题是否能及时获得满意答案.在 Stack Overflow 的真实数据上进行了实验验证,实验结果表明该方法的平均查全率可达 46.68%,查准率可达 58.87%.而随机猜测的平均查准率为 38.77%,查全率为 35.26%,与本文提出的方法有明显差距.

尽管本文提出的方法在测试集上的性能比其他方法比有明显提高,但总体来说准确率依然有待进一步提高.在未来的研究中,我们将对问题的解答状态的预测方法做进一步改进,并对问题特征中的特

征选取与提取工作进行更深的研究,详细分析各种因素与问题解答状态的关系,以期再提高预测的准确率.

参 考 文 献

[1] Xu Anzhen, Ji Zongcheng, Wang Bin. Quality prediction of community Q&A answers based on user response order [J]. Journal of Chinese Information Processing, 2017, 31(2): 132-138 (in Chinese)
(徐安滢, 吉宗诚, 王斌. 基于用户回答顺序的社区问答答案质量预测研究[J]. 中文信息学报, 2017, 31(2): 132-138)

[2] Han Jiawei, Meng Xiaofeng, Wang Jing, et al. Research on Web mining [J]. Journal of Computer Research and Development, 2001, 38(4): 405-414 (in Chinese)
(韩家炜, 孟小峰, 王静, 等. Web 挖掘研究[J]. 计算机研究与发展, 2001, 38(4): 405-414)

[3] Che Zhengping, Purushotham S, Cho K, et al. Recurrent neural networks for multivariate time series with missing values [J]. Scientific Reports, 2018, 8(1): 6085-6096

[4] Yao Yuan, Tong Hanghang, Xie Tao, et al. Detecting high-quality posts in community question answering sites [J]. Information Sciences, 2015, 302: 70-82

[5] Baltadzhieva A, Chrupała G. Question quality in community question answering forums: A survey [J]. ACM SIGKDD Explorations Newsletter, 2015, 17(1): 8-13

[6] Cai Hong, Li Ziwei, Yan Cuiting, et al. A shallow neural network based short text classifier for medical community question answering system [C] //Proc of the 8th IEEE Annual Int Conf on CYBER Technology in Automation, Control, and Intelligent Systems. Piscataway, NJ: IEEE, 2018: 1537-1541

[7] Jurczyk P, Agichtein E. Discovering authorities in question answer communities by using link analysis [C] //Proc of the 16th ACM Conf on information and knowledge management. New York: ACM, 2007: 919-922

[8] Dong Hualei, Wang Jian, Lin Hongfei, et al. Predicting best answerers for new questions: An approach leveraging distributed representations of words in community question answering [C] //Proc of the 9th Int Conf on Frontier of Computer Science and Technology. Piscataway, NJ: IEEE, 2015: 13-18

[9] Correa D, Sureka A. Fit or unfit: Analysis and prediction of'closed questions' on stack overflow [C] //Proc of the 1st ACM Conf on Online Social Networks. New York: ACM, 2013: 201-212

[10] Alharthi H, Outioua D, Baysal O. Predicting questions' scores on stack overflow [C] //Proc of the 3rd IEEE/ACM Int Workshop on CrowdSourcing in Software Engineering. Piscataway, NJ: IEEE, 2016: 1-7

[11] Agichtein E, Liu Yandong, Bian Jiang. Modeling information-seeker satisfaction in community question answering [J]. ACM Transactions on Knowledge Discovery from Data, 2009, 3(2): 10.1-10.27

[12] Gkotsis G, Stepanyan K, Pedrinaci C, et al. It's all in the content: State of the art best answer prediction based on discretisation of shallow linguistic features [C] //Proc of the 6th ACM Conf on Web Science. New York: ACM, 2014: 202-210

[13] Kafle K, Kanan C. Answer-type prediction for visual question answering [C] //Proc of the 34th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 4976-4984

[14] Yu Jun, Wang Liang, Yu Zhou. Research on visual Q&A technology [J]. Journal of Computer Research and Development, 2018, 55(9): 122-134 (in Chinese)
(俞俊, 汪亮, 余宙. 视觉问答技术研究[J]. 计算机研究与发展, 2018, 55(9): 122-134)

[15] Lewis D D, Schapire R E, Callan J P, et al. Training algorithms for linear text classifiers [C] //Proc of the 19th Int ACM SIGIR Conf on Research. New York: ACM, 1996: 298-306

[16] Gao Jie, Ji Genlin. Research on text classification technology [J]. Journal of Computer Applications, 2004, 21(7): 28-30 (in Chinese)
(高洁, 吉根林. 文本分类技术研究[J]. 计算机应用研究, 2004, 21(7): 28-30)

[17] Zhang Minling, Zhou Zhihua. ML-KNN: A lazy learning approach to multi-label learning [J]. Pattern Recognition, 2007, 40(7): 2038-2048

[18] Kwok J T Y. Automated text categorization using support vector machine [C] //Proc of the 5th Int Conf on Neural Information Processing. Berlin: Springer, 1998: 347-351

[19] Mikolov T, Sutskever I, Chen Kai, et al. Distributed representations of words and phrases and their compositionality [C] //Proc of the 27th Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2013: 3111-3119

[20] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C] //Proc of the 31st Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2017: 5998-6008

[21] Fawzi A, Fawzi O, Frossard P. Analysis of classifiers' robustness to adversarial perturbations [J]. Machine Learning, 2018, 107(3): 481-508

[22] Mikolov T, Chen Kai, Corrado G, et al. Efficient estimation of word representations in vector space [J]. arXiv preprint, arXiv: 1301.3781, 2013

[23] Shi Xingjian, Chen Zhourong, Wang Hao, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting [C] //Proc of the 29th Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2015: 802-810

[24] Song Hongmei, Wang Wenguan, Zhao Sanyuan, et al. Pyramid dilated deeper ConvLSTM for video salient object detection [C] //Proc of the 15th European Conf on Computer Vision. Berlin: Springer, 2018: 715-731

[25] Pang Changqing, Sun Ruibin, Mou Xiaobin, et al. Application of text classification method based on depth learning in email handwriting analysis [C] //Prof of the 4th Int Conf on Intelligent and Interactive Systems and Applications. Berlin: Springer, 2018: 432-439

[26] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines [C] //Proc of the 27th Int Conf on Machine Learning. New York: ACM, 2010: 807-814

[27] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification [C] //Proc of the 15th Conf of the European Chapter of the Association for Computational Linguistics. Stroudsburg: ACL, 2017: 427-431

[28] Takase T, Oyama S, Kurihara M. Effective neural network training with adaptive learning rate based on training loss [J]. Neural Networks, 2018, 101: 68-78

[29] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting [J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958

[30] Le X B D, Thung F, Lo D, et al. Overfitting in semantics-based automated program repair [J]. Empirical Software Engineering, 2018, 23(5): 3007-3033

[31] Yu Zhongxing, Martinez M, Danglot B, et al. Alleviating patch overfitting with automatic test generation: A study of feasibility and effectiveness for the Nopol repair system [J]. Empirical Software Engineering, 2019, 24(1): 33-67

[32] Riahi F, Zolaktaf Z, Shafiei M, et al. Finding expert users in community question answering [C] //Proc of the 21st Int Conf on World Wide Web. New York: ACM, 2012: 791-798

Shen Mingzhu, born in 1996. Master candidate. Received her BSc degree in the Northwest Agriculture & Forestry University. Her main research interests include software refactoring and software testing.

Liu Hui, born in 1978. Professor. Senior member of CCF. Received his BSc degree in control science from Shandong University in 2001, MSc degree in computer science from Shanghai University in 2004, and PhD degree in computer science from the Peking University in 2008. His main interests include software refactoring, AI-based software engineering, software quality and developing practical tools to assist software engineers.