Journal of Computer Applications
计算机应用,2016,36(12):3448-3453,3467
ISSN 1001-9081
CODEN JYIIDU
2016-12-10
http://www.joca.cn

文章编号:1001-9081(2016)12-3448-06
DOI:10.11772/j.issn.1001-9081.2016.12.3448

基于组合分类算法的源代码注释质量评估方法

余　海[1,2,3],李　斌[2,3,4],王培霞[2,3,4],贾　荻[3],王永吉[1,4*]

(1.中国科学院软件研究所 互联网软件技术实验室,北京 100190;　2.中国科学院大学,北京 100190;
3.中国科学院软件研究所 总体部,北京 100190;　4.中国科学院软件研究所 基础软件国家工程研究中心,北京 100190)

(*通信作者电子邮箱 ywang@itechs.iscas.ac.cn)

摘　要:源代码注释是软件的重要组成部分,研究者往往需要利用人工或自动化的方法产生分析注释,注释的质量评估也往往是通过人工来完成,这无疑是低效不客观的。为此,首先从注释的格式、语言形式、内容以及与代码相关度4个方面出发构建注释评估准则;进而,基于这一准则提出了一种基于组合分类算法的注释质量评估方法。该方法将机器学习以及自然语言处理技术引入到注释质量评估中来,利用分类算法将注释分为不合格、合格、良好、优秀四个等级。通过对基本分类算法的组合使用,使得评估效果进一步提高。组合分类算法的准确率和F1值较单独使用某一种分类算法提高20个百分点左右,除宏平均F1值外,各项指标都达到了70%以上。实验结果表明,所提方法能够很好地应用于注释质量评估。

关键词:源码注释;质量评估;文本分类;组合算法;自然语言处理

中图分类号:TP311　　**文献标志码**:A

Source code comments quality assessment method based on aggregation of classification algorithms

YU Hai[1,2,3], LI Bin[2,3,4], WANG Peixia[2,3,4], JIA Di[3], WANG Yongji[1,4*]

(1. *Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing* 100190, *China*;
2. *University of Chinese Academy of Sciences, Beijing* 100190, *China*;
3. *General Department, Institute of Software, Chinese Academy of Sciences, Beijing* 100190, *China*;
4. *National Engineering Research Center of Fundamental Software, Institute of Software, Chinese Academy of Sciences, Beijing* 100190, *China*)

Abstract: Source code comments is an important part of the software, so researchers need to use manual or automated methods to generate comments. In the past, the quality assessment of this kind of comments is done manually, which is inefficient and not objective. In order to solve this problem, an assessment criterion was built in which four aspects of the comments including comment format, language form, content and code-related degree were considered. Then a code comments quality assessment method based on an aggregation of classification algorithms was proposed, in which machine learning and natural language processing technology were introduced into comments quality assessment, by using classification algorithms the comments were classified into four levels, including unqualified, qualified, good and excellent ones. The evaluation results were improved by the aggregation of the basic classification algorithms. The precision and F1 measure of the aggregated classification algorithm were improved about 20 percentage points compared with using a single classification algorithm, and all the indexes have reached more than 70% except the macro average F1 measure. The experimental results show that this method can be applied to assess the quality of comments effectively.

Key words: source code comments; quality assessment; text classification; aggregation algorithm; natural language processing

0　引言

源代码注释是软件的重要组成部分[1],对代码理解和软件维护有着极其重要的作用,然而,多数软件项目并不能提供完整的注释和文档[2-3],因此,研究者试图利用人工以及自动化的方法为代码添加注释。其中,自动化方法主要包括注释复用[4-5]以及摘要抽取[6-8]。注释复用技术先从其他代码或问答系统中匹配将要被注释的代码片段并获取对应的描述性文本,构造"代码-描述"映射;然后,利用自然语言处理技术对描述文本进行处理,并按照一定的格式输出为注释。摘要抽取技术通过将代码中的类名、函数名等标识符切分为单词,然后结合程序的语法结构利用自然语言处理技术将这些单词组合成为陈述语句输出为代码注释。本文将这类由人工或自动化方法产生的注释统称为分析注释,但其质量的好坏往往受到技术限制和研究人员主观因素的影响而参差不齐,因此,客观、高效的质量评估方法,对保证注释的质量有着重要

收稿日期:2016-06-08;**修回日期**:2016-06-20。　**基金项目**:国家科技重大专项(2014ZX01029101-002)。

作者简介:余海(1989—),男,河南信阳人,硕士研究生,主要研究方向:操作系统、机器学习;　李斌(1985—),男,甘肃天水人,工程师,博士研究生,主要研究方向:操作系统、代码分析;　王培霞(1981—),女,山东潍坊人,高级工程师,博士研究生,主要研究方向:信息检索、自然语言处理;　贾荻(1989—),女,北京人,助理工程师,硕士,主要研究方向:操作系统、数据处理;　王永吉(1963—),男,辽宁营口人,研究员,博士,CCF高级会员,主要研究方向:虚拟化技术、隐蔽信道、实时系统、人工智能、数据挖掘、软件工程。

意义。

本文的研究对象正是分析注释。分析注释由研究者而非代码开发者提供,某些时候由于各种因素的限制并不能很好地传递代码所包含的信息,这部分注释将会影响后续研究者对代码的理解。分析注释表达丰富多样,往往以研究者母语表述,同时,一些注释中还会包含分析流图、参考资料链接等信息,这使得注释质量评估更加困难。

在以往的注释质量评估方法中,尤其是针对分析注释的质量评估,通常是提出一些诸如准确性、充分性、一致性、可读性等指标,并召集一些专业人员依照提出的指标进行人工分析并给出相应的判断结果[4-6,8]。这种做法无疑是耗时耗力的,而且由于主观因素的存在往往也不能保证结果的客观公正。

机器学习以及自然语言处理技术的发展,为高效客观地进行注释质量评估提供了思路与方法。本文首先从注释的格式、语言形式、内容以及与代码相关度四个方面出发,构建一个完整、客观的注释评估准则。进而提出了一种基于组合分类算法的源代码注释质量评估方法,将自然语言处理和机器学习中的文本分类相关技术应用到分析注释的质量评估中。同时,通过对已有分类算法的组合使用进一步提高了注释质量评估的效果。其基本思想是,首先,根据已建立的评估准则提取相应的特征项,将原始注释表示成特征向量;然后,利用本文提出的组合分类算法将其分为不合格、合格、良好、优秀四个等级。

本文方法只需对少量注释进行人工评估,通过对人工评估的注释进行学习进而得到评估模型,利用该模型可以高效客观进行注释质量评估。通过实验分析得到了令人满意的结果,准确率和 F1 值较单独使用某一种分类算法提高 20% 左右,除宏平均 F1 值外,各项指标都达到了 70% 以上。

1　相关研究

1.1　注释质量评估相关研究

近年来,研究人员从各种角度展开对注释的研究,大体可分为如下几个方向:注释对软件质量的影响[1,2,9-12]、注释自动生成[4-8]、质量评估[13-15]等。

针对注释质量评估研究的方面取得了一定的成果:Khamis 等[13]在工具原型 JavadocMiner 中从注释语言的使用及代码和注释之间的相关程度两个方面出发,利用一组简单的启发式方法来评估行内注释的质量。但是,在对注释和代码之间的相关程度分析上只是简单地从 javadoc 文档结构上考虑,进而结合错误报告来说明注释质量和代码质量有一定关系。Steidl 等[15]利用决策树对注释分类,针对不同类别注释依据相关度和注释长度来评估注释总体质量。该方法虽然使用了决策树进行分类,但仅仅是利用其对注释类别进行分类,如版权信息、行内注释等,并没有真正地用于对注释质量的评估;同时,在该方法中,对于注释和代码的相关度计算上只是考虑了二者的汉明距离,对注释本身的考察方面也只是针对注释的长度进行了考察。Fluri 等[14]研究了哪些项目中的代码注释能够很好地对代码进行解释,即只对注释和代码之间的相关性进行了研究,而忽略了注释本身的质量。

总体来看,目前对于注释的研究成果不是很多。对于注释质量评估方面的研究主要集中在针对原生注释的研究上,而针对分析注释的研究更是鲜有成果。但现有为数不多的注释研究的相关方法对研究分析注释有一定的借鉴意义。

1.2　作文自动评分相关研究

研究发现,对于注释文本本身的质量评估和作文自动评分[16-22]有共同之处。在该类系统中通过对作文的内容、形式等方法进行评估,最终得到一个分数。基于统计、自然语言处理以及人工智能等技术的作文自动评分研究在国内外已经形成一定的规模,特别值得注意的是,欧美国家已经将相关技术应用于实践,例如 E-rater[23]。由于机器学习技术的发展,一些研究者将诸如文本分类技术引入,并取得了很好的结果。这类作文自动评分系统通常会针对作文的语言流畅程度、语言形式、文本内容本身等方面来考察作文的质量。BETSY(Bayesian Essay Test Scoring sYstem)通过采用多元伯努利模型(Multivariate Bernoulli Model, MBM)和伯努利模型(Bernoulli Model, BM)两个朴素贝叶斯算法将作文自动分到四个集合(不合格、合格、良好、优秀)中,并得到了 80% 的准确率[16]。Xi 等[17]利用 Adaboost、Voting 和 K 近邻(K-Nearest Neighbor, KNN)等分类方法对 CET4 的作文进行分类打分,最终实验表明最优情况下 60% 的结果完全正确(Exact agreement),96% 结果在误差允许范围内(Adjacent agreement)。Li 等[18]将 CET4 的作文表示为 TF-IDF(Term Frequency-Inverse Document Frequency)向量空间模型,然后利用 KNN 模型对其分类,最终得到了 76% 的准确率。值得一提的是,黄志娥等[20]针对汉语水平考试(Hanyu Shuiping Kaoshi, HSK)自动作文评分的特征项提取作了相关研究,最终从作文整体及作文中的字、词、句和篇章各层面上,获取 107 个作文特征,该工作不仅对中文作文自动评分相关研究有重要意义,而且对中文注释的质量评估有着借鉴意义。

通过对作文和注释的对比研究发现,二者有一个本质的共同点:都是自然语言呈现的文本。但是二者之间又有一定区别:

1)注释质量与注释的辞藻是否华丽、句式是否丰富等无关,相比作文,注释只需表达清晰即可,因此,在提取特征项时,和词汇等级等相关的特征将不予考虑。

2)注释中的计算机专业词汇所占比重较大,词汇比较集中,在处理注释文本时需要将计算机专业词汇比例作为一个考察指标。

3)一个好的注释应该有一定的结构,通过考察注释的结构来考察注释的完整性,对注释而言,例如,函数注释需要分别对注释功能、参数、返回值等作相应的解释,在考察注释质量时需要针对这些部分作全面考虑。

4)作文文本语言上保持了一致性,而注释有些情况下并非如此,例如,本文所涉实验数据便是中文中夹杂着英文标识符以及各种符号,因此,需要进行特别处理。

综上所述,作文自动评分的相关研究提供了一种利用文本分类技术来对文本进行评估的方法,这为注释质量评估相关研究提供了借鉴,同时,对于注释和作文的不同点需要进行针对性的处理。

2　数据集及算法效果评价标准

2.1　数据集

本文实验数据来自“核高基”课题成果产出。各个课题参与单位的研究人员通过课题组搭建的源代码协同分析平台提交分析注释,被提交的分析注释以富文本形式存于数据库中。到目前为止该平台收集了大量涉及多个 Linux 内核版本

的分析注释数据,并按照注释对象进行分类,如针对函数的注释、针对变量的注释等。

本文对其中的 Linux-3.5.4 内核版本的 mm 模块中的 1 000条函数分析注释进行人工标注,用于实验。这 1 000 条注释来源于 6 个课题参与单位的 28 位研究人员。图 1 给出了一个例子,该例子中代码来自 Linux 内核 3.5.4 版本/mm/filemap.c 源文件,其中左侧为被注释的源代码,右侧为相应的分析注释。

```
0588 void unlock_page(struct page *page)
0589 {
0590     VM_BUG_ON(!PageLocked(page));
0591     clear_bit_unlock(PG_locked, &page->flags);
0592     smp_mb__after_clear_bit();
0593     wake_up_page(page, PG_locked);
0594 }
```

```
函数功能说明：对锁定的页解除锁定
函数参数说明：page为指向结构体page的指针变量，表示一个指定的页
返回值：空
函数说明：
1.调用clear_bit_unlock函数为指定的锁定的页解除锁定
2.调用smp_mb_after_clear_bit函数将页的标志清除
3.调用wake_up_page函数唤醒处于锁定状态的页
```

图 1 一个分析注释的例子

2.2 算法效果评价标准

在对注释进行质量评估时,其输出为注释的质量等级,其结果有两种情况,输出的注释等级和实际等级一致或不一致。因此,本文将准确率作为一个主要的评价指标。同时,为了全面地评估各算法针对注释质量评估的效果,本文引入了融合准确率和召回率两种指标的 F1 值作为一个评价指标,它是准确率和召回率的调和平均数。

本文将注释的质量分为:不合格、合格、良好、优秀 4 个等级,可表示成一个多类问题,因此需要计算出一个融合每个分类器指标的综合指标。通过研究发现,多数注释的质量等级为合格或良好,少数为不合格和优秀。因此,为了更为准确、全面地评估注释的质量,在评价指标融合时本文同时使用宏平均和微平均两种方式,其中宏平均是在类别之间求平均值,微平均是将每条注释在每个类别(本文指注释质量的 4 个等级)上的判定结果存入一个集合中,然后基于这个集合来计算最终的效果指标,即宏平均平等地对待每个类别,微平均平等地对待每条注释的判定结果。

3 注释特征项提取

本文通过研究注释的特点,试图构建一个完整的注释特征项提取规则。研究表明,从注释与代码相关程度以及注释本身两个方面来考察注释的质量是客观的、完备的[13,15]。在对注释本身的考察方面,本文力图全面地考察影响注释质量相关信息,而非仅仅考察词性、文本长度等为数不多的信息[13,15]。为实现这一目的,本文借助自然语言处理等技术从注释格式、注释语言形式以及注释文本内容三个方面来提取特征项,此三点,由面到点,层层递进,针对注释的整体完整性、语言运用以及文本内容等 3 个层次全面地考察了注释的质量。在注释与代码相关度方面,本文摒弃了以往研究中只是简单考虑代码标识符切分后的单词是否在注释中出现等简单考察方式,引入了基于语言建模的信息检索模型,以便更加准确地表述注释与代码之间的相关程度。

3.1 注释格式相关特征项

注释格式是注释完整性的直观表现。在对代码进行注释时,对于不同类型的注释通常有不同的格式。例如函数注释中,需要对函数功能、参数、返回值等进行分析,那么一个好的函数注释需要包含“函数功能”“参数”“返回值”等固定标签。格式相关的特征项的提取就是通过考察一条注释是否包含该类型注释应该包含的结构信息,进而考察该条注释是否完整。

3.2 注释语言形式相关特征项

注释的语言基本成分使用情况能够反映注释语言运用情况。在考察注释的语言基本成分使用情况时,并不关心注释所表达的何种信息,而仅仅研究注释是如何表达的[20]。以此为出发点,本文将对注释中各种词性的比例、句子数目等进行统计。

值得注意的是,注释是一种特定环境下的文字表述,其主要目的就是表述清楚代码所包含的信息。在研究其语言形式特性时,无需过多考虑辞藻是否华丽,因此对于通常作文自动评分中涉及到诸如词语的等级、庄雅度相关的特征项将不需要进行研究。

3.3 注释文本内容相关特征项

注释文本内容是对文本内容的考查。利用自然语言处理技术将注释文本进行分词,并对停用词进行剔除,进而将注释文本表示为特征空间向量。值得注意的是,这种特征空间向量具有高维性和稀疏性,为了解决这个问题,本文采用卡方(Chi-square)统计来对特征项进行进一步的选取。

3.4 注释与代码相关度

注释与代码的相关度作为注释质量评估研究的重要指标之一,一直以来都是研究者不可避免的难点又往往不易提出更优的解决方法[10,13-15]。本文通过将基于语言建模的信息检索模型应用到注释与代码相关度的计算中,在一定程度上更为准确地表示注释与代码相关程度。其具体表现在既考虑了代码词项在对应注释中的出现频率,同时,通过引用如线性插值平滑等方法,将代码词项在注释集中出现的频率纳入计算,考虑更加全面。语言建模是信息检索中一种通用的形式化方法。在计算注释和代码相关程度时,本文将使用一种最为基本的语言模型——查询似然模型。将代码记作查询 q,将注释记作文档 d。注释和代码的相关度可以表示为:

$$P(d \mid q) \propto P(d) \prod_{t \in q} \left[\lambda \frac{tn_{t,d}}{L_d} + (1-\lambda) \frac{tn_{t,c}}{N} \right]$$

其中:$P(d)$ 对所有注释都一样,可以不予考虑;$tn_{t,d}$ 是词项 t 在注释 d 中出现的次数;L_d 是注释 d 的词项数目;$tn_{t,c}$ 是词项 t 在整个注释文本集合中出现次数;N 是整个注释文本集合中词项数目。

4 注释质量评估

本文结合已有分类算法的成熟度、简易性、效率、效果四个方面选取了朴素贝叶斯(Naive Bayes,NB)、K 近邻(KNN)、决策树(Decision Tree,DT)、支持向量机(Support Vector Machine,SVM)等 4 个分类算法作为基本分类算法。

成熟度 选取的基本分类算法必须已经足够成熟,以确保组合分类算法的稳定性。

简易性　选取的基本分类算法必须足够简易,过于复杂的算法,如神经网络等,本身已经是很复杂的算法了,并不适合进行再组合。

效率　组合分类算法由于使用多个分类算法,效率必然受到影响,因此要求每个基本分类算法的效率足够高。

效果　4 种分类算法的效率要大致相当,否则,组合分类算法由于受效果较差的分类算法影响而不能有较好的表现。

本文选取的 4 种典型的基本分类算法都是在学术界和工业界被广泛应用的分类算法,已经足够成熟,模型的复杂性和效率都可以接受,虽然分类效果上有一定差别,但大致相当。

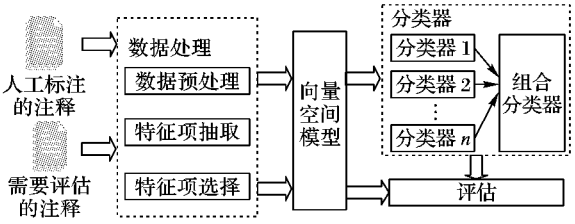

图 2　组合分类算法流程

4.1　基本分类算法

朴素贝叶斯(NB)严格来说它是一类基于贝叶斯公式的监督学习方法。其“朴素”表现在此类方法假设文档中各词项是相互独立的,即特征条件独立假设。首先,对于给定的训练集,基于特征条件独立假设学习输入和输出的联合概率分布,随后利用该模型,对于给定输入 x,利用贝叶斯公式求出后验概率最大的输出 y。NB 算法所需估计的参数较少、实现简单,是一种高效的分类算法。

K 近邻(KNN)分类方法是一种基于向量空间模型的文本分类方法,K 是一个可设定的参数。KNN 分类器就是将距离待测文档最近的 K 篇文档的类别赋给该文档。由于 KNN 分类方法的这种特性,通常它并不需要特别的训练,只需要存储一定数量标记好分类的文档就可以完成对待测文档的分类。

决策树(DT)是从类标记的训练集学习而来的一种树状结构。决策树的每个非叶子节点表示一个特征项,每个叶子节点表示一个类别。首先,利用训练数据,遵循损失函数最小化原则构建决策树模型;然后,对于预测数据,利用构建好的决策树模型进行分类。决策树可以处理高维数据并且有较高的精度,同时,由于模型的树形结构特性,使得决策树分类算法更为直观,可读性强。

支持向量机(SVM)是一种基于向量的分类算法。传统的基于向量分类器只需找到一条直线或一个平面将训练集中的样本分开即可。支持向量机分类器并不满足于此,其目标是在两个类别之间找到一个决策平面,使之到两个类别最远。提到决策平面就决定了支持向量机本质上是一个二类分类器,所以对多类问题的处理需要进一步处理。实际使用中,通常是建立个“一对多”的二分分类器或者 $|c|(|c|-1)/2$ 个“一对一”的二分分类器来实现对多类的分类,其中 $|c|$ 是类别总数。

4.2　组合分类算法

研究发现多分类算法组合往往能够得到更高的分类精度[24-26]。多分类算法组合有两种形式:并联和串联。所谓并联就是分别训练多个基本分类器,然后通过投票系统对基本分类器的分类结果进行组合处理,从而得到最终的分类结果。相对地,串联分类器是将上一级的分类结果作为下一级分类器的输入,这样层层分类,得到最终结果。由于各基本分类器各有侧重点,多分类器组合往往能够相互弥补不足,而得到比单个分类器更好的结果。

本文提出一种并串联结合的组合分类算法。如图 2 所示右侧虚线框部分是组合分类器。同其他监督学习算法类似,组合分类算法分为三个阶段:

数据预处理　对原始数据进行清洗并提取特征项,使用 CHI 统计对提取的特征项进行进一步处理,最终将注释表示成向量空间模型。将处理好的数据分成训练集 X、测试集 Y 以及预测集 Z,并对训练集 X 和测试集 Y 进行人工标记。

学习评估模型　对给定的训练集合 X,$\boldsymbol{x}_i=(x_i^{(1)},x_i^{(2)},\cdots,x_i^{(j)},\cdots,x_i^{(m)},c_i)\in X$,其中 $x_i^{(j)}$ 是特征向量 $\boldsymbol{x}_i$ 的第 j 个特征值,c_i 是对应的类标号。对于输入 x_i 第 j 个基本分类器有输出 $c_i^{(j)}$。n 个基本分类器的输出及类标号可组成特征向量 $(c_i^{(1)},c_i^{(2)},\cdots,c_i^{(j)},\cdots,c_i^{(n)},c'_i)\in C$,$C$ 是融合分类器的训练数据集合。n 个基本分类器和融合分类器训练完毕后,利用测试集 Y 对训练所得模型进行评估、调参,使得误差最小的模型即为最终的评估模型。

注释评估　对于给定预测集 Z,对于输入特征向量 $\boldsymbol{z}_i=(z_i^{(1)},z_i^{(2)},\cdots,z_i^{(j)},\cdots,z_i^{(m)})\in Z$,有输出 c_i',c_i' 即特征向量 $\boldsymbol{z}_i$ 对应注释的质量评估等级。

5　实验结果及分析

本章首先针对 4 种基本分类算法分别进行实验,以此来考察基本分类算法的效果;然后,在完全相同的实验环境中利用本文所提出的组合分类算法进行实验,并将实验结果同第一次实验结果作对比,以此来验证组合分类算法是否优于单独使用基本分类算法;最后,通过对数据集按照不同比例划分训练集和测试集的策略,对 4 种基本分类算法以及由这 4 种基本分类算法组合而成的组合分类算法进行实验,进一步考察组合分类算法的稳定性。

5.1　实验一

通过实验来评估 4 种基本分类算法在本文所提供数据集上的效果。本文使用 scikit-learn[27] 机器学习库提供的分类算法来实现实验。scikit-learn 机器学习库功能完备,底层实现了大量经典实用的机器学习算法;文档整洁详实,便于学习使用;同时,其活跃的开发社区保证了产品的稳定性。通过将已经标注的 1000 条数据按照 7∶3 的比例分为训练集和测试集来进行实验,结果如表 1 所示。通过实验分析,不难发现 4 种基本分类算法在本文提供数据集上的效果并不令人满意。在准确率方面除了 SVM 算法的宏平均准确率超过了 60%,其他算法无论在微平均准确率还是在宏平均准确率方面都在 50% 左右。同时,融合了准确率和召回率的 F1 值也多数在 60% 徘徊。

总体来说,单独使用基本分类器在本文所提供的数据集上的效果仅仅比随机分类有所提高,因此并不能应用于实践中的注释质量评估。为此本文进一步提出了组合分类算法,试图通过组合使用基本分类算法来提高分类效果。

在介绍组合分类算法之前,为方便评估组合分类算法的

效果,需要从 4 种基本分类算法中选取一种算法作为 Baseline。从表 1 得出,除了宏平均 F1 值外,各项评估指标中 SVM 分类算法都要优于其他三个基本分类算法,这也和理论相一致[28]。为了更为直观地比较基本分类算法和组合分类算法的分类效果,本文规定基本分类算法中在本文提供数据集中表现最好的 SVM 分类算法的结果作为 Baseline,来分析组合分类算法的效果是否有所升。

表 1 4 种基本分类算法在本文数据集上的效果 %

基本分类算法	准确率		F1 值	
	微平均	宏平均	微平均	宏平均
NB	44.1	49.6	61.2	59.8
*K*NN	45.7	49.7	62.8	59.3
DT	41.4	45.4	58.6	57.8
SVM	52.6	63.4	69.0	46.7

5.2 实验二

在相同的实验环境下,进行实验二。实验过程中,使用 NB、*K*NN、DT、SVM 中的一个分类算法作为融合分类算法,其输入为余下 3 种分类算法的输出。表 2 所示为将 1 000 条数据按照 7∶3 的比例分为训练集和测试集来进行实验的结果。和表 1 对比,除朴素贝叶斯组合分类算法的宏平均 F1 值外,4 种组合分类算法在各项评估指标上都要明显优于单独使用某一种基本分类算法。相对于实验一种规定的 Baseline,组合分类算法中的支持向量机组合分类算法在微平均准确率上提高了 20.8 个百分点,在宏平均准确率上提高了17.1 个百分点,在微平均 F1 值上提高了 15.6 个百分点,在宏平均 F1 值上提高了 16.7 个百分点。

表 2 4 种组合分类算法在本文数据集上的效果 %

组合分类算法	准确率		F1 值	
	微平均	宏平均	微平均	宏平均
NB	49.0	87.5	65.8	16.6
*K*NN	67.4	55.9	80.6	66.3
DT	71.7	78.4	83.5	61.9
SVM	73.4	80.5	84.6	63.4

组合分类算法在各项评估指标上都有较大幅度的提升,在一定程度上证明了组合分类算法在注释质量评估方面的可行性。但是,实验一和实验二使用的训练集和测试集是固定的,即实验结果可能是偶然的、不稳定的,为此,本文设计了实验三。

5.3 实验三

为了排除实验一和实验二因为训练集和测试集单一,而使得实验结果存在偶然性的可能,在本实验中,将 1 000 条数据平均分成 10 组,每组 100 条,从中随机抽取 $m(1 < m < 9)$ 组数据作为训练集,余下 $10 - m$ 组数据作为测试集,进行多次实验。

图 3 中给出了 Baseline 和四种组合分类算法准确率与训练集大小的关系,从图中可以得出,无论在微平均准确率还是在宏平均准确率上 4 种组合分类算法都要优于 Baseline,其中朴素贝叶斯组合分类算法和 *K* 邻近组合分类算法效果波动较大,但整体上也要优于 Baseline。而决策树组合分类算法和支持向量机组合分类算法不仅效果远优于 Baseline,提升了 20 个百分点~30 个百分点,而且随着训练集数据的增加效果上有小幅上升趋势。

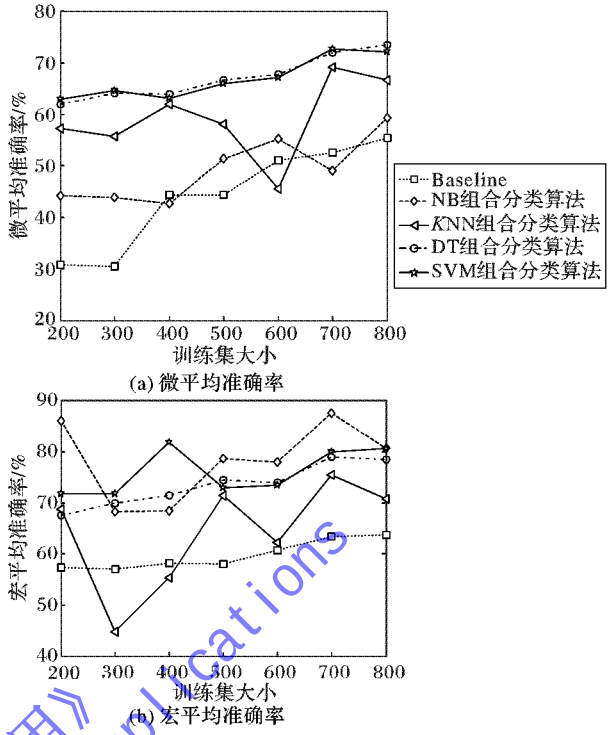

图 3 各算法微平均和宏平均准确率与数据集大小关系

图 4 中给出了 Baseline 和四种组合分类算法 F1 与训练集大小的关系。从图中来看,除了朴素贝叶斯组合分类算法在宏平均 F1 值上要低于 Baseline,其他三种组合分类算法无论在微平均 F1 值还是宏平均 F1 上都要优于 Baseline。通过图 4 可以发现,决策树组合分类算法和支持向量机组合分类算法在 F1 值上的表现也要远优于 Baseline,提升了 10 个百分点~20 个百分点,且依旧表现得很平稳,随训练集数据的增加而小幅增长。

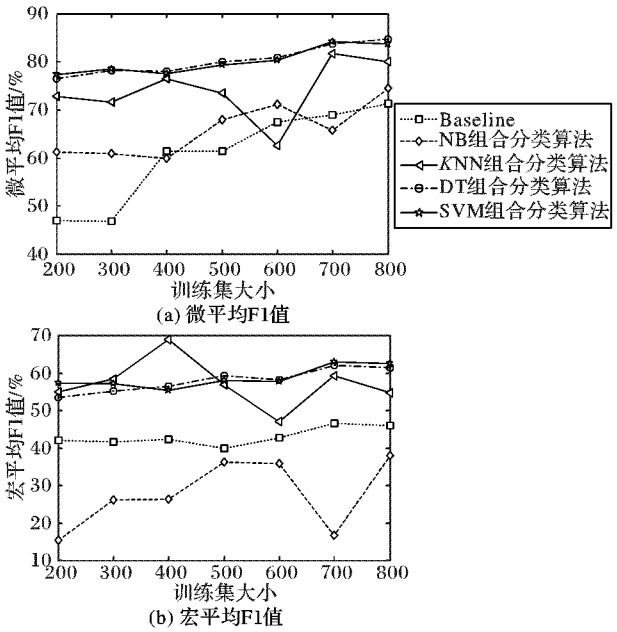

图 4 各算法微平均和宏平均 F1 值与数据集大小关系

综合来看,针对本文所提供的数据,在训练集随机变化的情况下,除了朴素贝叶斯组合分类器外,组合分类算法在准确率和 F1 值两个评估指标上都要优于单独使用基本分类算法,

且有较大的提升,因此本实验能够证明组合分类算法是稳定的。

6　结语

本文提出的基于组合分类算法的源代码分析注释质量评估方法,将自然语言处理以及机器学习相关技术引入到注释质量评估的研究中。通过建立完善的评估准则以及对分类算法的组合应用,注释质量评估效果在准确率和 F1 值两个指标上都有较大提高。同时,通过实验对比,发现决策树组合分类算法和支持向量机组合分类算法在本文提供的数据上表现出令人满意的效果。值得注意的是,虽然组合分类算法能够较好地应用于注释质量评估,但由于注释有着不同于一般文本的特点,给特征项的提取带来了一定的困难,同时,组合分类算法在效率上也提出了更高的要求。下一步,将根据分析注释的特点,进一步优化特征项提取和分类算法,以期得到更好的结果。

参考文献:

[1] TAN S H, MARINOV D, TAN L, et al. @ tComment: testing javadoc comments to detect comment-code inconsistencies [C]// Proceedings of the 5th IEEE International Conference on Software Testing, Verification and Validation. Washington, DC: IEEE Computer Society, 2012: 260 – 269.

[2] DE SOUZA S C B, ANQUETIL N, DE OLIVEIRA K M. A study of the documentation essential to software maintenance [C]// Proceedings of the 23rd Annual International Conference on Design of Communication: Documenting & Designing for Pervasive Information. New York: ACM, 2005: 68 – 75.

[3] KAJKO-MATTSSON M. A survey of documentation practice within corrective maintenance [J]. Empirical Software Engineering, 2005, 10(1): 31 – 55.

[4] WONG E, YANG J, TAN L. AutoComment: mining question and answer sites for automatic comment generation [C]// Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering. Piscataway, NJ: IEEE, 2013: 562 – 567.

[5] WONG E, LIU T, TAN L. CloCom: mining existing source code for automatic comment generation [C]// Proceedings of the 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering. Piscataway, NJ: IEEE, 2015: 380 – 389.

[6] MORENO L, APONTE J, SRIDHARA G, et al. Automatic generation of natural language summaries for Java classes [C]// Proceedings of the 21st IEEE International Conference on Program Comprehension. Piscataway, NJ: IEEE, 2013: 23 – 32.

[7] SRIDHARA G, POLLOCK L, VIJAY-SHANKER K. Generating parameter comments and integrating with method summaries [C]// Proceedings of the 19th IEEE International Conference on Program Comprehension. Piscataway, NJ: IEEE, 2011: 71 – 80.

[8] SRIDHARA G, HILL E, MUPPANENI D, et al. Towards automatically generating summary comments for Java methods [C]// Proceedings of the IEEE/ACM International Conference on Automated Software Engineering. New York: ACM, 2010: 43 – 52.

[9] TAN L, ZHOU Y, PADIOLEAU Y. aComment: mining annotations from comments and code to detect interrupt related concurrency bugs [C]// Proceedings of the 33rd International Conference on Software Engineering. New York: ACM, 2011: 11 – 20.

[10] HIRATA Y, MIZUNO O. Do comments explain codes adequately?: investigation by text filtering [C]// Proceedings of the 8th Working Conference on Mining Software Repositories. New York: ACM, 2011: 242 – 245.

[11] ARAFAT O, RIEHLE D. The commenting practice of open source [C]// Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications. New York: ACM, 2009: 857 – 864.

[12] STOREY M, RYALL J, BULL R I, et al. TODO or to bug: exploring how task annotations play a role in the work practices of software developers [C]// Proceedings of the 30th International Conference on Software Engineering. New York: ACM, 2008: 251 – 260.

[13] KHAMIS N, WITTE R E, RILLING A J. Automatic quality assessment of source code comments: the JavadocMiner [C]// Proceedings of the 2010 Natural Language Processing and Information Systems, and 15th International Conference on Applications of Natural Language to Information Systems. Berlin: Springer-Verlag, 2010: 68 – 79.

[14] FLURI B, WURSCH M, GALL H C. Do code and comments co-evolve? On the relation between source code and comment changes [C]// Proceedings of the 14th Working Conference on Reverse Engineering. Washington, DC: IEEE Computer Society, 2007: 70 – 79.

[15] STEIDL D, HUMMEL B, JUERGENS E. Quality analysis of source code comments [C]// Proceedings of the 21st IEEE International Conference on Program Comprehension. Piscataway, NJ: IEEE, 2013: 83 – 92.

[16] DIKLI S. An overview of automated scoring of essays [J]. The Journal of Technology, Learning, and Assessment, 2006, 5(1): 1 – 36.

[17] XI Y, LIANG W. Automated computer-based CET4 essay scoring system [C]// Proceedings of the 3rd Pacific-Asia Conference on Circuits, Communications and System. Piscataway, NJ: IEEE, 2011: 1 – 4.

[18] LI B, LU J, YAO J M, et al. Automated essay Scoring using the KNN algorithm [C]// Proceedings of the 2008 International Conference on Computer Science and Software Engineering. Piscataway, NJ: IEEE, 2008: 735 – 738.

[19] ATTALI Y, BURSTEIN J. Automated essay scoring with e-rater? V. 2 [J]. The Journal of Technology, Learning, and Assessment, 2006, 4(3): 1 – 31.

[20] 黄志娥,谢佳莉,荀恩东. HSK 自动作文评分的特征选取研究[J]. 计算机工程与应用,2014,50(6):118 – 122. (HUANG Z E, XIE J L, XUN E D. Study of feature selection in HSK automated essay scoring [J]. Computer Engineering and Applications, 2014, 50(6): 118 – 122.)

[21] 彭星源, 柯登峰, 赵知, 等. 基于词汇评分的汉语作文自动评分[J]. 中文信息学报, 2012, 26(2): 102 – 108. (PENG X Y, KE D F, ZHAO Z, et al. Automated Chinese essay scoring based on word scores [J]. Journal of Chinese Information Processing, 2012, 26(2): 102 – 108.)

(下转第 3467 页)

intenance and Reengineering. Washington, DC: IEEE Computer Society, 2009: 219 – 228.

[7] BAKOKA T, FERENC R, GYIMOTHY T. Clone smells in software evolution [C]// ICSM 2007: Proceedings of the 2007 IEEE International Conference on Software Maintenance. Piscataway, NJ: IEEE, 2007: 24 – 33.

[8] THUMMALAPENTA S, CERULO L, AVERSANO L, et al. An empirical study on the maintenance of source code clones [J]. Empirical Software Engineering, 2010, 15(1): 1 – 34.

[9] SAHA R K, ROY C K, SCHNEIDER K A. An automatic framework for extracting and classifying near-miss clone genealogies [C]// Proceedings of the 2011 27th IEEE International Conference on Software Maintenance. Piscataway, NJ: IEEE, 2011: 293 – 302.

[10] CI M, SU X H, WANG T T, et al. A new clone group mapping algorithm for extracting clone genealogy on multi-version software [C]// IMCCC'13: Proceedings of the 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control. Washington, DC: IEEE Computer Society, 2013: 848 – 853.

[11] 张瑞霞, 张丽萍, 王春晖, 等. 基于主题建模技术的克隆群映射方法[J]. 计算机工程与设计, 2015, 36(6): 1524 – 1529. (ZHANG R X, ZHANG L P, WANG C H, et al. Clone group mapping method based on topic modeling [J]. Computer Engineering and Design, 2015, 36(6): 1524 – 1529.)

[12] 涂颖, 张丽萍, 王春晖, 等. 基于软件多版本演化提取克隆谱系[J]. 计算机应用, 2015, 35(4): 1169 – 1173, 1178 (TU Y, ZHANG L P, WANG C H, et al. Clone genealogies extraction based on software evolution over multiple versions [J]. Journal of Computer Applications, 2015, 35(4): 1169 – 1173, 1178.)

[13] KIM M, SAZAWAL V, NOTKIN D, et al. An empirical study of code clone genealogies [C]// ESEC/FSE-13: Proceedings of the 2005 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering. New York: ACM, 2005: 187 – 196.

[14] 陈桌, 张丽萍, 王欢, 等. 基于改进向量空间模型的克隆群映射方法[J]. 计算机应用, 2016, 36(7): 2031 – 2037. (CHEN Z, ZHANG L P, WANG H, et al. Clone group mapping method based on improved vector space model [J]. Journal of Computer Applications, 2016, 36(7): 2031 – 2037.)

[15] 张久杰, 王春晖, 张丽萍, 等. 基于 Token 编辑距离检测克隆代码[J]. 计算机应用, 2015, 35 (12): 3536 – 3543. (ZHANG J J, WANG C H, ZHANG L P, et al. Code clone detection based on Levenshtein distance of token [J]. Journal of Computer Applications, 2015, 35(12): 3536 – 3543.)

[16] 张丽萍, 张瑞霞, 王欢, 等. 基于贝叶斯网络的克隆代码有害性预测[J]. 计算机应用, 2016, 36(1): 260 – 265. (ZHANG L P, ZHANG R X, WANG H, et al. Harmfulness prediction of clone code based on Bayesian network [J]. Journal of Computer Applications, 2016, 36(1): 260 – 265.)

Background

This work is partially supported by the National Natural Science Foundation of China (61462071, 61363017), the National Natural Science Foundation of Inner Mongolia (2014MS0613), the Foundation Project of Inner Mongolia Education Department (NJZY16045).

CHEN Zhuo, born in 1989, M. S. candidate. His research interests include software engineering, software analysis.

ZHANG Liping, born in 1974, M. S., professor. Her research interests include software engineering, software analysis.

WANG Chunhui, born in 1979, M. S., lecturer. Her research interests include software analysis, multimedia, computer aided instruction.

(上接第 3453 页)

[22] 江进林. 近五十年来自动评分研究综述——兼论中国学生英译汉机器评分系统的新探索[J]. 现代教育技术, 2013, 23(6): 62 – 66. (JIANG J L. Rethinking 50 years of studies on automated scoring-explorations of computer scoring system for English-Chinese translations of Chinese learners [J]. Modern Educational Technology, 2013, 23(6): 62 – 66.)

[23] POWERS D E, BURSTEIN J C, CHODOROW M, et al. Stumping e-rater: challenging the validity of automated essay scoring[J]. Computers in Human Behavior, 2002, 18(2): 103 – 134.

[24] ZHANG Y, LO D, XIA X, et al. An empirical study of classifier combination for cross-project defect prediction [C]// Proceedings of the 39th IEEE Annual International Computers, Software & Applications Conference. Piscataway, NJ: IEEE, 2015: 264 – 269.

[25] 王正群, 孙兴华, 杨静宇. 多分类器组合研究[J]. 计算机工程与应用, 2002, 38(20): 84 – 85. (WANG Z Q, SUN X H, YANG J Y. Study on multiple classifiers combination [J]. Computer Engineering and Applications, 2002, 38(20): 84 – 85.)

[26] 付忠良. 分类器线性组合的有效性和最佳组合问题的研究[J]. 计算机研究与发展, 2009, 46(7): 1206 – 1216. (FU Z L. Effective property and best combination of classifier linear combination [J]. Journal of Computer Research and Development, 2009, 46(7): 1206 – 1216.)

[27] PEDREGOSA F, VAROQUAUX G, GRAMFORT A, et al. Scikit-learn: machine learning in Python [J]. The Journal of Machine Learning Research, 2011, 12(10): 2825 – 2830.

[28] 卢苇, 彭雅. 几种常用文本分类算法性能比较与分析[J]. 湖南大学学报(自然科学版), 2007, 34(6): 67 – 69. (LU W, PENG Y. Performance comparison and analysis of several general text classification algorithms [J]. Journal of Hunan University (Natural Sciences), 2007, 34(6): 67 – 69.)

Background

This work is partially supported by the National Science and Technology Major Project (2014ZX01029101-002).

YU Hai, born in 1989, M. S. candidate. His research interests include operating system, machine learning.

LI Bin, born in 1985, Ph. D. candidate, engineer. His research interests include operating system, code analysis.

WANG Peixia, born in 1981, Ph. D. candidate, senior engineer. Her research interests include information retrieval, natural language processing.

JIA Di, born in 1989, M. S., assistant engineer. Her research interests include operating system, data processing.

WANG Yongji, born in 1963, Ph. D., research fellow. His research interests include virtualization technology, covert channel, real-time system, artificial intelligence, data mining, software engineering.