附件1-2：

**开题检查记录表**

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 刘奇 | 学 号 | 20181102867 |
| 专 业 | 计算机科学与技术（师范类） | 开题日期 | 2021.10.23 |
| 指导教师 | 张丽萍 | 职 称 |  |
| 设计（论文）题目 | 面向学生程序代码的相似度检测 | | |
| 指导教师评语：  签字： | | | |
| 检查记录（答辩记录）： | | | |
| 是否通过开题： □通过   □不通过 | | 成绩： | |
| 答辩组长签字： | | 答辩组成员： | |
| 答辩秘书签字： | |  | |

附件2：



**内蒙古师范大学计算机科学技术学院**

**毕业设计（论文）开题报告**

**题 目：面向学生程序代码的相似度检测**

**专 业 计算机科学与技术（师范类）**

**姓 名 刘奇**

**学 号 20181102867**

**指导教师 张丽萍**

**日 期 2021年10月23日**

**计算机科学技术学院制**

**说 明**

**一、开题报告主要内容**

1．课题来源及研究的目的和意义

（正文 宋体小4号字，行距1.25倍，段前0行，段后0行）

2．国内外在该方向的研究现状及分析

3．主要研究内容

4. 研究方案

5．进度安排，预期达到的目标

6．课题已具备和所需的条件、经费

7．研究过程中可能遇到的困难和问题，解决的措施

8．主要参考文献

**二、开题报告要求**

1．开题报告的字数应在2000字以上。

2．参考文献的要求：

（1）理工类论文的参考文献一般为10-15篇，其中学术期刊类文献不少于7篇，外文文献不少于3篇（特殊专业可酌情确定明确要求，并报教务处备案）；文科、管理类论文，参考文献一般为15-20篇，其中学术期刊类文献不少于12篇，外文文献不少于3篇。近五年的文献数不应少于总数的1/3，应有近两年的参考文献。教材、产品说明书、国家标准、未公开发表的研究报告不宜作为参考资料。

（2）参考文献按在开题报告中出现的次序列出。

（3）参考文献书写顺序：序号 作者.文章名.学术刊物名.年，卷（期）：引用起止页。

1．课题来源及研究的目的和意义

随着大数据和人工智能等技术的飞速发展，国内高校也越来越注重学生编程能力的培养，各种语言的编程类课程也随之增加，教师在教学过程中，要注重理论和实践相结合，让学生编写程序代码是不可避免的。而随着学习编程语言的学生的数量不断增加，教师审批学生程序代码作业的数量尤为庞大，无法准确、及时地掌握学生的学习情况。且由于程序代码作业抄袭容易，导致抄袭的现象不同程度的存在于各高校之中，严重影响了学生能力的培养和教师的教学效果。如果依靠教师自己去人为的在大量的学生作业中寻找出抄袭现象，并且对学生作业的整体进行分析和对比，是非常繁琐而且低效的。教师无法对学生的真实水平做出一个准确评估和评价，且无法从作业和考试中获得真实的教学反馈信息，会严重影响教学的效果，并且会使那些独立认真完成作业的同学的积极性受到影响，破坏学风，无法实现高等学校培养学生个人能力的目标。

为此需要以智能且自动化的方式对学生的作业进行相似度检测和分析，针对学生提交的程序作业，为教师呈现多种分析结果，以便教师及时掌握学生的学习情况和学习态度。当前与此相关的代码相似度检测、查重的研究已较为成熟，然而并没有呈现出详细的检测结果给教师。教师无法完全根据相似度检测出来的百分比来发现学生的问题，还是需要看到学生代码的具体情况，因此要给教师呈现出具体的相似的代码的内容和人数，教师才能发现问题并予以针对性的讲解。

1. 国内外在该方向的研究现状及分析

有关程序相似度检测的研究工作在国外开展较早，Baxter给出了利用抽象语法树查找克隆代码的基本算法步骤以及优化子树匹配效率的算法，后续Koschke又提出一种抽象语法树的改进算法。重点提及的是Winnowing算法[1-2]，这是由Schleimer等受Rabin-Karp 算法启发，提出的基于数字指纹的匹配算法。Rabin-Karp 算法是由 Rabin 和 Karp提出的字符串匹配算法。

国内相关的研究有：张文典[3]使用属性度量技术开发了国内第一个抄袭检测系统，朱江[4]将XML引入本领域，王继远[5]采用基于XML的结构度量方法进行相似度计算，赵长海等[6]提出了编译优化和反汇编的程序相似性检测方法，张鹏[7]用本体粗糙集的方法对程序代码进行了相似度度量，熊浩[8]提出了一种基于静态词法树的程序相似性检测方法；王春晖[9]对最长公共子串(GST)算法和其改进算法RKR-GST算法进行了研究，通过实验表明该算法的效果比使用LCS算法和编辑距离来计算相似度的效果要好。

针对高等教育中学生程序代码查重抄袭的研究也备受重视，中国科学技术大学黄振亚[10]、于俊[11]等针对高等学校的教学目标，对面向教育评估的教育辅助平台研究和代码抄袭检测研究，都做出了很成熟的技术成果，其研究过程中也参考了北京邮电大学殷丹平[12]基于CNN对代码检测和代码查重的研究分析。大连民族大学李玲[13]等人也基于一种Simhash算法设计了代码查重系统。潘理虎[14]等人将代码查重系统应用到实践教学中进行了测试，并提出了将检测结果进行可视化分析的想法。韩冰[15]基于FTP教学平台对代码相似度进行了研究。

1. 主要研究内容

3.1 作业文件过滤提取与预处理

对作业压缩包进行解压，找到最终作业文件.然后对压缩包里的文件进行过滤，如对不属于代码的作业文件、重复提交的作业进行过滤等.允许上传的类型就是设置待检测的类型.最后得到过滤后文件。

3.2 相似度计算

对作业代码进行格式化，接着提取格式化后代码的Hash特征，然后使用Winnowing算法进行Hash值的筛选建立文件指纹集，最后计算两份代码文件的指纹重复度得到其相似度。

3.3相似学生分组

对任意两个学生程序代码作业计算相似度后，一个班级的所有学生的相似情况就可以生成一个矩阵，根据重复度挑选相关的学生分组。

3.4相似内容位置定位

使用标记串token记录源代码位置和代码格式化后的位置，能够保留作业原位置的信息和代码格式化后的位置信息，通过指纹匹配对应的原位置和格式化后的位置，定位学生程序代码作业的相似位置。

3.5学生作业相似度计算结果选择性优化

使用CNN进行代码文本特征抽取，利用前馈网络模型替代传统方法中复杂的标记串处理与字符串匹配，进行相似度计算，改进学生程序代码作业相似度计算结果。

3.6计算结果可视化

根据前述计算结果，给出作业重复率矩阵，相似学生分组，少数学生代码对比时相似代码标红。

1. 研究方案

检测流程设计如图1所示。

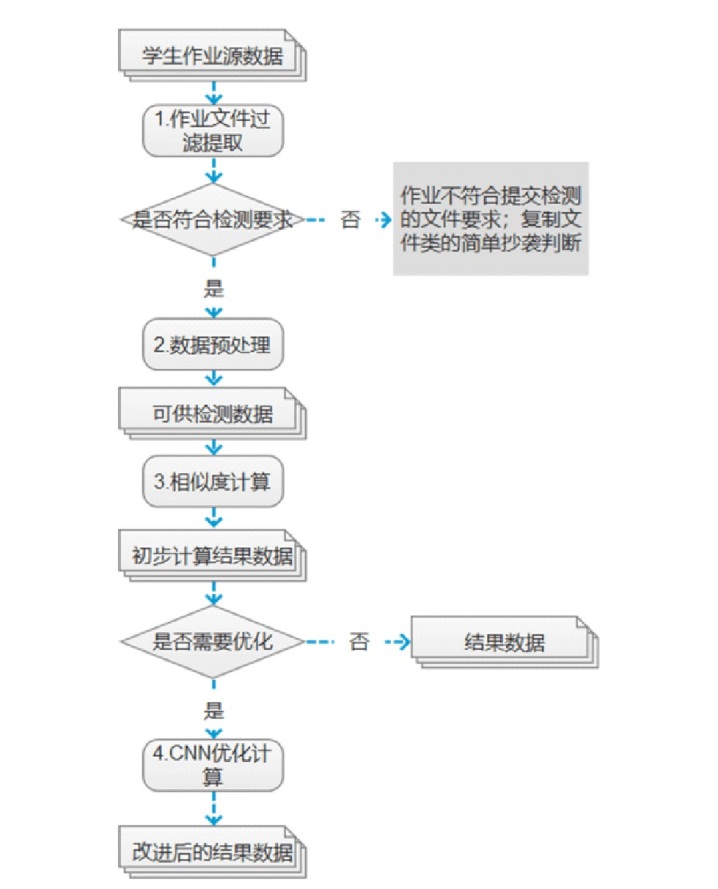


图1 学生程序代码相似度检测计算流程

4.1代码格式化

方法：结构度量。

结构度量是将程序代码格式化为特定标记字符串，主要有基于文本、标记串、语法树等，然后再用字符串匹配的方法计算相似度。

第一步代码转换，可分为基于tree的方法[16]、基于graph的方法、基于token的方法[17]、基于metrics的方法、基于字符串的方法[18]、混合型方法等几大类。

第二步标记串度量，常用方法有：GST[19]、序列匹配、最长公共子串、RKR-GST[20]、点阵图[21]。

我选取的是基于标记串的方法，将空格和注释过滤掉，所有的变量名替换为V，所有字符串替换为S，所有函数名替换为F。

4.2字符串匹配算法

选择随机串匹配（KR）算法计算Hash值。

KR算法是一种随机串匹配算法，该算法把长度为m的模式串按照某个哈希函数计算得到一个对应的散列值，这个函数成为散列函数，把文本串依次切割为n-m+1个长度为ｍ的子串，也按照这个散列函数计算得到相应的散列值．将模式串的散列值与文本串的所有散列值逐一比较，找出所有与模式串具有相同散列值的文本子串，再判断其对应的字符是否相等，决定该模式串是否是文本串的子串．

4.3代码指纹生成算法

根据KR算法生成的Hash值，选取Winnowing算法基于数字指纹来完成代码特征的提取，以精确识别代码重复问题，如图2所示。

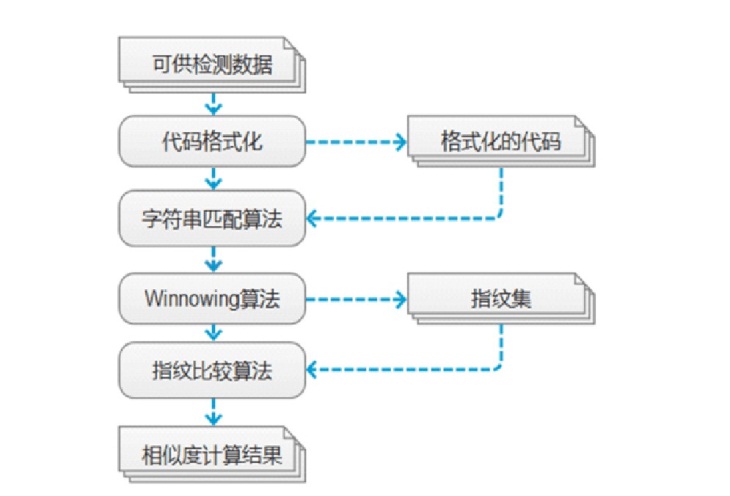


图2 学生代码作业相似度检测计算步骤

关于 winnowing 算法的简单的解释就是：基于 k-gram 的哈希码，以最小规则提取部分gram 的哈希码作为文档特征指纹，并记录gram的位置。

4.4相似内容位置定位

使用k-grams记录哈希值和格式化代码中未知的起点和终点。

如果指纹匹配，先将k-grams在格式化代码中的位置信息和token、格式化代码中的位置信息对应，然后获取token中的原始文件的起始位置和终点位置。

4.5 CNN模型优化计算结果

使用Winnowing算法获得种子样本，设置相似度大于80%时记为正样本1,相似度小于50%记为负样本0;选取完全不相似的1000个样本为种子数据，用常用的抄袭手段对样本1∶15进行抄袭样本构建，产生15000对正样本；种子样本两两匹配，选取30000对为负样本。

使用词法分析器Pygment预处理代码。

将原文T1、正例T2、负例T3的3元输入，把代码变成矩阵，利用矩阵还原学生程序代码作业，使用TF-IDF值进行填充。

使用CNN模型的512个filter模拟k-gram的卷积过程，然后进行池化，得到F1,F2,F3特征向量。

叠加一层DNN模型，得到全部200维度的语义向量，并在此基础上，拼接Winnowing算法得到的指纹向量。

使用目标函数：max(0,simi(F1,F2)-simi(F1,F3)+1)重新计算作业相似度，提升学生作业相似检测的精准度。

1. 进度安排，预期达到的目标

|  |  |  |
| --- | --- | --- |
| 进度安排 | | 预期目标 |
| 2021.09.20 | 完成选题需求分析 | 完成需求分析 |
| 2021.09.27 | 初步准备开题报告，查找资料 | 开题报告初版完成 |
| 2021.10.04 | 修改和完成开题报告，查找学习资源 | 严格按要求的书写格式填写开题报告，内容严谨 |
| 2021.10.11 | 确定开题报告，学习预处理技术 | 开题报告符合导师要求，进一步学习技术 |
| 2021.10.18 | 学生代码收集，进行预处理实验 | 得出可检测的数据 |
| 2021.10.25 |
| 2021.11.01 |
| 2021.11.08 | KR算法学习  根据KR算法初步处理数据 | 得出Hash值 |
| 2021.11.15 |
| 2021.11.22 |
| 2021.11.29 | Winnowing算法学习 | 可初步使用算法检测少量代码相似度 |
| 2021.12.06 |
| 2021.12.13 |
| 2021.12.20 | CNN算法学习 | 可优化代码相似度检测结果 |
| 2021.12.27 |
| 2022.01.03 |
| 2022.01.10 |
| 2022.01.17 | 结合两种算法对大量数据进行处理 | 可得正确的分析结果 |
| 2022.01.24 |
| 2022.01.31 | 设计前端 | 前端大体完成 |
| 2022.02.07 | 检测结果分析处理 | 可具体实现研究内容的功能 |
| 2022.02.14 |
| 2022.02.21 | 前后端连接，数据库建立 | 整体功能实现完成 |
| 2022.02.28 |
| 2022.03.06 | 功能完善 | 完美实现研究内容 |
| 2022.03.15 | 完成论文，准备答辩 | 通过答辩，顺利毕业 |

6．课题已具备和所需的条件、经费

学院提供的上机，资料，学习资源等

1. 研究过程中可能遇到的困难和问题，解决的措施
2. KR算法、Winnowing算法的掌握

（2）针对大量且不同形式的学生程序作业的预处理的合理有效方法

（3）CNN（卷积神经网络）的学习使用

查找相关文献学习；反复试验优选出适用的方法；向指导教师请教。

8．主要参考文献

[1]<https://blog.csdn.net/chichoxian/article/details/53115067>

[2]<https://blog.csdn.net/qq_32023541/article/details/82382808>

[3]张文典,任冬伟.程序抄袭判定系统[J].小型微型计算机系统,1988(10):34-39.

[4]朱江. 基于XML的程序设计自动批改的研究[D].湘潭大学,2005.

[5]王继远. 一种用于软件作业评判系统的程序结构分析算法的设计与实现[D].北京邮电大学,2007.

[6]赵长海,晏海华,金茂忠.基于编译优化和反汇编的程序相似性检测方法[J].北京航空航天大学学报,2008(06):711-715.

[7]张鹏,王国胤,陶春梅,罗海.基于本体粗糙集的程序代码相似度度量方法[J].重庆邮电大学学报(自然科学版),2008,20(06):737-741.

[8]熊浩,晏海华,赫建营,赵长海.一种基于静态词法树的程序相似性检测方法[J].计算机应用研究,2009,26(04):1316-1319+1326.

[9]王春晖. 程序代码抄袭检测中串匹配算法的研究与实现[D].内蒙古师范大学,2008.

[10]黄振亚,苏喻,吴润泽,刘玉苹,刘淇,陈志刚,胡国平.一种面向教育评估的智能教育辅助平台[J].中国科学技术大学学报,2015,45(10):846-854.

[11]于俊,李雅洁,程礼磊,连顺,谭昶,丁德成,刘淇.高教程序代码作业抄袭检测的方法研究与实践[J].中国科学技术大学学报,2020,50(08):1048-1057.

[12]殷丹平. 基于CNN的代码相似度检测研究与代码查重系统[D].北京邮电大学,2018.

[13]李玲,王法胜,李绍民.基于Simhash算法的作业查重系统设计[J].大连民族大学学报,2020,22(01):80-84.

[14]潘理虎,张雷,解丹,陈立潮,赵淑芳.面向实践教学的作业查重系统[J].软件工程,2019,22(05):35-38.

[15]韩冰.基于FTP教学平台的代码相似度检测的研究[J].计算机光盘软件与应用,2012(09):217-218.

[[16] KOSCHKE R,FALKE R,FRENZEL P,et al.Clone Detection Using Abstract Syntax Suffix Trees[C].Working Conference on Reverse Engineering,2006:253-262.](http://trial.discovery.cnki.net/search/index.action?q=Clone%20Detection%20Using%20Abstract%20Syntax%20Suffix%20Trees%5bC%5d" \t "/Users/liuqi/Documents\x/_blank)

[[17] KAMIYA T,KUSUMOTO S,INOUE K,et al.CCFinder:A multilinguistic token-based code clone detection system for large scale source code[J].IEEE Transactions on Software Engineering,2002,28(7):654-670.](http://trial.discovery.cnki.net/search/index.action?q=CCFinder:A%20multilinguistic%20token-based%20code%20clone%20detection%20system%20for%20large%20scale%20source%20code%5bJ%5d" \t "/Users/liuqi/Documents\x/_blank)

[18] BAKER B S.Parameterized duplication in strings:Algorithms and an application to software maintenance[J].SIAM Journal on Computing,1997,26(5):1343-1362.

[19] Wise M J.Running karp-rabin matching and greedy string tiling[J].Software - Practice and Experience,1993.

[20] PRECHELT L,MALPOHL G,PHILIPPSEN M.Finding plagiarisms among a set of programs with JPlag[J].Universal Computer Science,2000,8(11):1016-1038.

[21] GRANVILLE A.Detecting Plagiarism in Java Code[J].Supervisor Yorick Wilks,2002.