

基于大模型试错与概念图的“程序设计” 翻转课堂教学方法探索 *

李志刚 杨吉斌 张睿 王彩玲 吴永芬

陆军工程大学指挥控制工程学院, 南京 210007

摘要 近年来, 翻转课堂为教学效果的提升开辟了新思路, 强调以学生为中心, 注重教学过程的互动与参与。如何科学地设计和组织翻转课堂, 引导学生积极参与其中, 成为决定教学效果的一个关键因素。针对程序设计初学者在实践中遇到的难题, 本研究提出了一种基于 AI 大模型交互的翻转课堂设计, 引入了试错法和概念图等工具。通过引导学生利用 AI 大模型互动, 将编程过程中遇到的常见错误与知识点通过概念图的形式进行关联, 形成阶段性学习成果。这些阶段性成果有助于教师精准掌握学情, 同时还能为翻转环节提供有力依据和素材, 进而在课堂教学的实施环节中, 更有效地引导学生参与讨论, 深化教学互动。

关键字 试错法, 大模型, 程序设计, 翻转课堂

Exploration of Flipped Classroom Teaching Method for "Programming Course" Based on Trial-and-Error Using LLMs and Concept Maps

LI Zhigang YANG Jibin ZHANG Rui WANG Cailing WU Yongfen

Command and Control Engineering College
Army Engineering University
Nanjing 210007, China

Abstract—In recent years, flipped classrooms have opened up new ideas for improving teaching effectiveness, emphasizing student-centered learning and emphasizing interaction and participation in the teaching process. How to scientifically design and organize flipped classrooms, guide students to actively participate, and become a key factor in determining teaching effectiveness. In response to the difficulties encountered by beginners in programming in practice, this study proposes a flipped classroom design based on AI large model interaction, introducing tools such as trial and error and concept maps. By guiding students to use AI big models for interaction, common errors encountered during the programming process are associated with knowledge points through concept maps, forming phased learning outcomes. These phased achievements help teachers to accurately grasp the learning situation, while also providing strong basis and materials for designing flipped links. In the implementation of classroom teaching, they can more effectively guide students to participate in discussions and deepen teaching interaction.

Keywords—Trial-and-Error, LLMs, Programming Course, Flipped Classroom

1 引言

随着教育改革的不断深入, 翻转课堂逐渐在程序设计课程中展现出其独特优势。一般来说, 翻转课堂主要包括课前预习、课堂讨论、课后复习等环节。然而, 当前翻转课堂在实施过程中仍存在一些问题[1, 2]。在课前阶段, 学生主要依赖视频、PPT 等多媒体资源自学, 往往缺乏足够的实践机会, 且难以提出高质量的问题。课堂讨论环节往往因学生提出的问题不聚焦而效果欠佳。程序设计作为一门理论与实践结合紧密

的课程, 实践环节尤为重要, 但是对初学者来说, 编程实践中遇到的问题很有可能阻碍学生的学习进度。因此, 如何更好的指导学生在课前有效预习、课后增加实践机会, 以及在课中积极参与讨论, 成为亟待解决的问题。

基于桑代克的“试误说”, 已有文献提出了“理性创造错误”的教学模式, 鼓励学生通过理性创造编程过程中可能遇到的错误, 在学习过程中提前了解相应知识点可能遇到的问题[3]。在程序设计翻转课堂的课前阶段, 让学生主动试错, 不仅可以避免在后期编程中重复犯错, 还能激发学生的批判性思维, 提出更具价值的问题。生成式 AI 技术自推出以来, 因其强大的学习能力和理解能力, 可以实现“真正像人类一样

*基金资助: 本文得到陆军工程大学教育教学课题课程建设资助。

**通讯作者: 李志刚 lzgnudt@qq.com。

聊天交流”，能够完成撰写诸如邮件、文案、程序代码、研究论文等任务，在辅助课堂教学中也开始发挥积极的作用[4-6]。随着AI大模型的推广，学员在编写程序的过程中，利用大模型技术，通过与大模型的交互，可以比较方便的编写程序、调试程序，找出自己编写的程序的一些错误[7,8]。

然而，孤立地尝试错误还只是零散的学习和记忆知识点，达不到系统化学习的目的。如何引导学生对学习过程遇到的错误进行归纳总结和分析就非常有必要。本研究在试错法的基础上，结合概念图工具，提出在课前阶段或实验阶段引导学生将编程过程中的“错误”以概念图的形式进行整理，形成“错误知识点”概念图[9]。在课中阶段，则利用这些概念图引导学生讨论发言，帮助他们从正反两方面理解知识点及其对应的错误概念之间的关系，逐步完善课程知识体系的认知构建。

2 C 语言程序设计教与学的特点

目前国内多数院校的编程基础课程以C语言开展教学工作，相关自学资源也比较丰富。在程序设计教学的过程中，目前一般的教授方法都是提前布置学生通过观看视频或看书的方式进行自学，然后按照先讲解相关知识点和概念，再讲解相关的语法和原理，再利用应用案例进行实践教学。在讲解和实践的环节，教师会针对具体的问题或错误进行针对性的讲解。对学生而言，在程序设计的学习过程中，对C语言相关概念的理解不是一蹴而就的，而是一个不断迭代的过程。在概念理解的过程中，初学者会产生一些理论认识上的错误。除了理论学习，编写和调试程序也是学生需要掌握一个基本的实践能力。

2.1 初学者调试程序的特点

在初学阶段，部分学生存在不熟悉编译器环境和不熟悉编程语言基本概念的双重问题。学生在调试程序的过程，会经常遇到错误，导致编写的程序不能正常运行。即使编译器给出了提示，因为学生的经验不足和C语言错误的特点，往往不能立刻定位和理解相应的错误。在尝试不同的解决方案的过程中，可能会产生更多的错误。一旦学员在调试不通过的时候，常常出现急躁的情绪，影响学习的积极性和效率。

2.2 “通关式”实验方式的局限性

随着在线测评系统用于程序设计的实验教学，教师通常利用线上平台布置实验任务和题目，再进行分析并给出部分或全部演示代码，由学生上机完成实验。然而，在这一过程中，部分学生往往以实验通过为主要目的，即达到教师提出的任务目标（也称为通关），获得相应的积分奖励。这种以“通关”为目的的实验

方式可能带来的问题是：即使能够比较顺利通关，但是在通关之后，部分学生仍然缺乏对代码的深度思考和对利用代码解决问题的泛化能力。另外在尝试错误和探索方面有所欠缺，这对编程能力的培养显然是不够的。因此，如何引导学生在实验过程中尝试和探索，成为程序设计教学过程的一大难点。

3 C 语言编程错误的特点

一般来说，程序调试过程中的错误大致分为语法错误和逻辑错误。对于语法错误，主要是由于对C语言规范认识不清，编写的程序不符合C语言的语法标准，编译器不能通过的程序。而逻辑错误是指编写的程序虽然通过了编译器的编译，但因为存在逻辑问题，导致程序不能得到正确的结果。部分逻辑错误也是因为语法错误导致的。编译器提示的常见错误约有120种，大多是由语法错误导致的。在调试过程中，C语言的错误有以下几个特点[10]。

3.1 关联性

C语言错误具有关联性，对编程者而言可能是一个错误，但是编译器在提示的时候，会判定出几类关联的错误。比如，在输入程序的过程中，一般而言都是用英文输入。在C语言中一个很基础的知识点“C语言的语句以‘；’结束”。这里的“；”一定是英文的半角的分号。类似这种概念讲解起来，基本上所有的学生都能听懂，不存在理解上的问题。但是一到上机操作，如果学员注意不到，常常造成程序调试出现“严重的”语法问题。如果在一个语句后面没有“；”，则编译器会出现类似以下的错误提示“syntax error : missing ‘；’ before...”。但是如果学生把“；”输入为“；”（中文全角的引号），除了以上提示，则会出现“unknown character ‘0xa3’”和“unknown character ‘0xbb’”错误提示。这个错误的原因是由于编译器不能识别中文字符导致的。在学生编程实践中，这样的错误是有关联性的。而这种关联性，经常导致学生在调试程序的过程中比较迷惑。

3.2 隐蔽性

C语言错误的隐蔽性是指某些错误在代码编写阶段不易被察觉，甚至能通过编译，但在运行时表现出异常行为或产生难以追踪的问题。这类错误往往比简单的语法错误更隐蔽，常常是因为逻辑上出现了错误因其可能长期潜伏，直到特定条件下才暴露，导致调试和差错的难度增大。因为错误的隐蔽性，特别是当学生编写的代码比较长的时候，更不易发现问题。比如有同学经常把“；”放在for(), while()的括号的后面，导致程序虽然能够编译通过，但在运行时出现错误。

3.3 多样性

同一知识点可能因上下文不同,比如因为在编写过程中出现的形式和位置不同,导致的错误“症状”也不一样,引发不同类型的错误。比如同样是“;”符号放置的问题,可能会导致因为放得位置不对,产生语法错误,也看能通过运行,导致逻辑错误。

4 基于大模型试错的案例

大模型技术产生以来,给教学领域也带来了很多机遇与挑战[5,6,7]。上文提到的一些C语言编写过程中的一些容易出错的知识点,学生在编程的过程中很多时候不易察觉和发现这些问题。特别是,当学生人数较多的时候,教师指导的精力和时间跟不上。在大模型技术产生以后,我们在教学实践中,鼓励学生利用大模型技术进行程序的判断。

比如以下代码是学生在学习完“结构体”知识点以后,完成的一段利用结构体的知识来编写求解复数运算的程序代码。根据C语言的语法规规范,在结构体定义结束的“}”后面是需要一个“;”。但是这位同学忽略了这个符号,因此程序会报告一个语法错误。在实验和调试过程中,学生对这个错误不敏感,发现这个错误,可能需要较长的时间。我们鼓励学生把这段代码提交给大模型。

案例: 编写C语言程序,定义一个复数结构体,并计算两个复数的和。

```
#include <stdio.h>
struct complex{
    double real;
    double img;
}
int main()
{
    struct complex a={1, 2}, b={3, 4}, c;
    c.real=a.real+b.real;
    c.img=a.img+b.img;
    printf("c=%lf+%lf i", c.real, c.img);
    return 0;
}
```

大模型反馈的结果:

这段代码是C语言编写的,目的是定义一个复数结构体,并计算两个复数的和。不过,代码中有几个小错误需要修正:

- ① 结构体定义后面缺少分号。
- ② 在main函数中,struct complex后面应该没有大括号{},因为它们是用于初始化结构体变量的,而不是类型定义。

下面是修正后的代码:

```
#include <stdio.h>
struct complex{
    double real;
    double img;
```

```
}; // 在这里添加分号
int main()
{
    struct complex a = {1, 2}, b = {3, 4}, c;
    // 移除了类型定义中的大括号
    c.real = a.real + b.real;
    c.img = a.img + b.img;
    printf("c=%lf+%lf i", c.real, c.img);
    // 正确输出复数c
    return 0;
}
```

这段代码定义了一个名为complex的结构体,用来存储复数的实部和虚部。在main函数中,定义了三个complex类型的变量a、b和c,并初始化了a和b。然后,计算a和b的和,并将结果存储在c中。最后,使用printf函数输出复数c的值。

通过大模型的反馈结果不难发现,大模型技术对程序中的这个细小的错误能够准确的找到并定位,而且能够给出原因和解决方案,另外对代码所反映的逻辑任务也能够分析出来。因此这对学生调试程序来说,比只是简单的通过编译器给出的错误提示要丰富的多。在这个基础上,还可以鼓励学生通过修改代码,采用不断主动注入错误的方式,来制造更多的错误,让大模型来诊断错误,并生成更多的反馈结果。

5 基于“错误”概念图的教学设计案例

当然,在教学的过程中,我们发现学生仅仅利用大模型来辅助程序设计的学习,会带来调试能力退化、知识体系的碎片化等问题。让学生通过主动试错、调试程序这样的方式,学生在知识点的学习过程中,还不够系统,也就是说,很多情况下,学生对程序中的错误还是孤立和碎片化的记忆,没有将程序中的错误进行归类总结,难以和所学知识点进行有效的链接。我们在教学实践中,利用了概念图这个工具来引导学生,建立“错误”概念图,以达到系统学习的目的。

一般来说,概念图常用来描述相关学科的概念之间的联系。利用不同的图形来表示不同类型的知识点,并用线连接不同的知识点,形成树状或者网状的关系图。学习者自己制作和利用概念图可以方便学习者学习,并能够系统化的学习和掌握知识。

本文利用概念图的工具和理念,将程序设计中的知识点,与该知识点相关的错误,以及该错误引起的结果相互连接在一起,形成一种面向程序调试错误的概念图。在设计的过程中,为了突出三者的不同,可以通过不同的图形来表示知识点、错误类型和错误结果。图1给出了利用概念图的形式,将程序设计中的关于分号的用法的部分知识点和错误连接起来,在这个图中分别用不同的图形来表示知识点、错误类型和错误提示(错误导致的结果)。

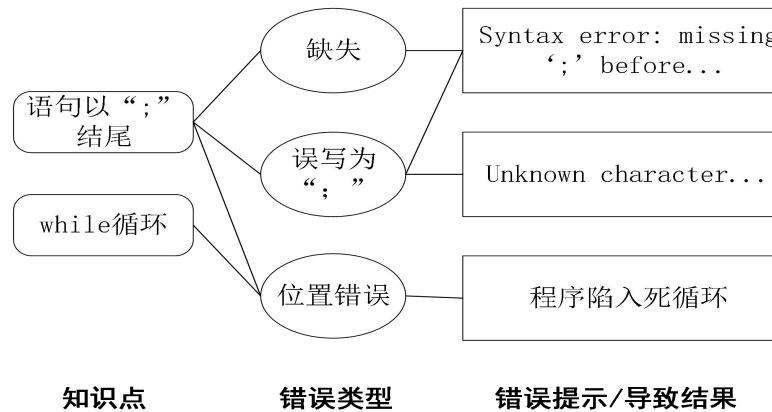


图 1 程序设计中的错误与知识点关联的概念图示例

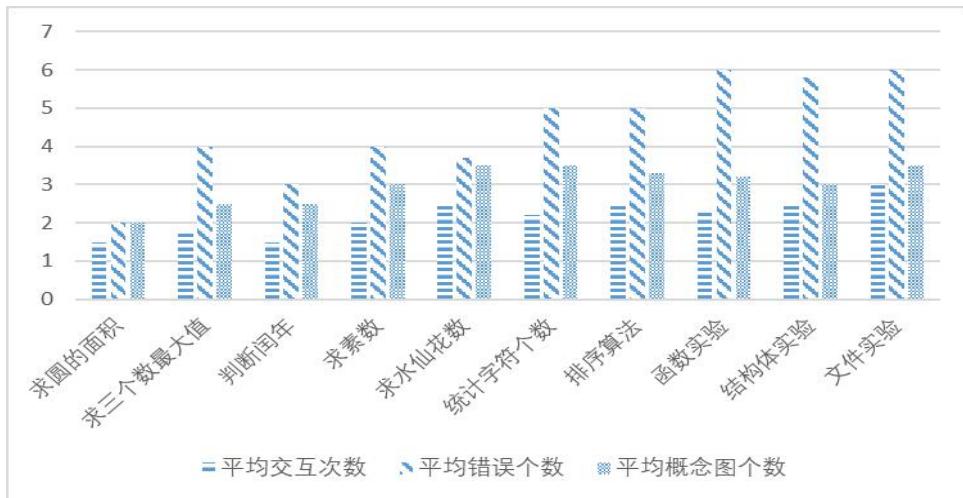


图 2 学生与大模型交互情况

基于试错法和错误概念图，作者在教学过程中总结了如下课堂设计和实践方法。

(1) 课前预习：课前布置预习题目和示例代码，鼓励学生通过AI大模型工具，发现程序中的错误。

(2) 概念图生成：学生将发现的错误，利用概念图工具，总结和生成“错误”概念图。

(3) 课堂讨论：课堂教学中，采用翻转课堂，分组讨论编程过程中的发现的问题，并结合大家的讨论补充完善“错误”概念图，并分享给所有同学便于课下复习。

(4) 知识点讲解：在翻转课堂中，会遇到相关知识点和语法原理大家都不是很明白的地方。在知识点和语法原理讲解之后，不急于进行编程实践，而是提前分析在编程过程中会遇到的一些错误，并将一个初步的“错误”概念图展示给学生。引导学生将初步的

概念图通过实践教学进行迭代和完善。

6 基于大模型的 C 语言教学效果分析

前期我们已经尝试了在程序设计课程中引入大模型辅助教学，并从两个方面采用问卷调查的方式，在课程中对学生利用大模型技术进行 C 语言学习的效果进行了分析。在教学过程中，我们通过采用与大模型问答的方式进行讨论，可以极大的提高学生课堂的参与度，也提高了学生对程序设计的好奇心和学习兴趣。同时方便了教师在翻转课堂实施的过程中，能及时准确的了解学员的状态，挖掘讨论话题，启发学生进一步的思考。60%以上的学员认为大模型技术能提高自己的编程能力[11]。

在 2025 年春季学期的课堂上，学生人数为 45 人，我们在实验课上一共设计了 10 道实验题目，供学生练习。从学生与大模型交互的次数、概念图覆盖的错误

知识点的错误数量,以及人均完善概念图的次数来观察和统计学生使用大模型学习的情况。

从统计来看,对一个题目学生从自己编程,利用大模型辅助试错,每一道题目经过2到3次可以帮助学生完善代码,并从中尝试和发现3-6个错误。在概念图的生成方面,结合发现的错误,在基础概念图的版本上,经过2-3次的完善可以形成一个较完整的版本。

从图2可以发现,随着课程难度的增加,学生与大模型交互的次数并不是显著增加,这说明目前大模型对程序的挑错能力和辅助编程已经比较完善。当然随着知识点的增多,学生利用大模型可以发现的错误也会越来越多。

本次教学过程所设置的题目大多数还是一些基础题目,并没有设置开放性的任务。后期的教学过程中,将尝试设置一些开放性、难度大的任务,让学生来学习和尝试。

7 结束语

本文结合作者在程序设计教学中的经验,在程序设计课堂上尝试利用了AI大模型技术进行翻转课堂的实践。提出了通过AI大模型发现程序中错误,并利用概念图将这些错误系统化的总结呈现出来。这些概念图作为素材,用于在翻转课堂中进行讨论,调动学生讨论和学习的积极性。在后面的教学中,作者将结

合学生的学习表现和考核成绩进一步优化该课程的教学设计。

参 考 文 献

- [1] 李成严,高峻,唐远新,等.翻转课堂教学评价体系研究[J].计算机教育,2015,01(11):100-103.
- [2] 孙静,周涛,李强.翻转课堂在程序设计课程中的实践与反思[J].中国教育信息化,2019(12): 75 - 79.
- [3] 贺广生,结凤克,王勤民,王运成.基于C程序设计的“理性创造错误”实验教学法.计算机工程与科学,202-204.第36卷第A 1期 2014年4月
- [4] 徐锐,黄子文,宋雨轩,皮德常.从AI大模型看高校计算机教育面临的机遇与挑战[J].计算机技术与教育学报,2024,12(03), P99-106.
- [5] 陈丽,张磊,王鹏.基于大模型的智能辅导系统在编程教学中的应用研究[J].远程教育杂志,2024, 42(1): 67 - 74.
- [6] 董帅,庄宇,李锐乔.大模型赋能的人工智能导论实践教学改革[J].计算机技术与教育学报,2024,12(05), P109-114.
- [7] 刘凌,吴永芬,陈卫卫,赵洪华,唐艳琴.大语言模型对高校程序设计类课程的挑战与机遇,计算机教育,2025.02:118-122
- [8] 王聪,万聪.大模型时代计算机程序设计类课程教学模式探索,计算机教育,2025.04:137-141
- [9] 王萍,张华,李明.基于概念图的教学设计在编程课程中的应用研究[J].现代教育技术,2020, 30(4): 88 - 93.
- [10] 郑波,杨帆,刘伟.C语言常见错误分析与教学对策[J].计算机教育,2018(10): 105 - 108.
- [11] 李志刚,杨吉斌,张睿,等.基于ChatGPT的程序设计翻转课堂教学方法实践.计算机技术与教育学报,2023,11(02), P125-129.