

计算科学评论

环球科学

WWW.COMPUTER.ORG

2017年第2期

学习技术进展

聊天机器人：“小会话，大学问” P9

计算机教育改革的推手 P64

ENIAC是如何焕发第二春的 P68

定价：25元

合作机构



IEEE
computer
society



ISSN 1673-5153



云计算

人工智能

工控机

制造业

电子

嵌入计算

传感器

互联网

3D 打印

绿色计算

图形图像

虚拟现实

职场
市场
行业分析
创业
学习 科技新闻
竞赛

找工作
教育

互联网金融

芯片

大数据

软件架构

纳米架构

人机交互

多媒体

普适计算 MEMS



微信名：计算机人 微信号：jisuanren



Copyright

版权

主管单位 Authorities in Charge

中华人民共和国教育部 Ministry of Education of the People's Republic of China

主办单位 Sponsor

中国大学出版社协会 China University Presses Association

出版单位 Publisher

《环球科学》杂志社有限公司 GLOBAL SCIENCE MAGAZINES Co., Ltd

社址 Address: 北京市朝阳区秀水街1号建外交公寓4-1-21 Office 4-1-21, Jianguomen Diplomatic Residence Compound, No. 1, Xiu Shui Street, Chaoyang District, Beijing, China. 邮编 100600

联系电话: 010-85325810 / 85325871

社长 / 总编辑 Editor-in-chief

陈宗周 ChenZongzhou

副校长 / 副总编辑 Deputy Editor-in-chief

刘芳 LiuFang

执行出版人 Publisher

管心宇 Xinyu Guan

资深编辑 Senior Editor

马法达 Falda Ma

刘妍 Yan Liu

特约编辑 Contributing Editor

史彦诚 Yancheng Shi

刘大明 Daming Liu

高天羽 Tianyu Gao

费楠 Yong Fei

王璇 Xuan Wang

运营中心 OPERATING DEPARTMENT

运营机构 Publisher

上海灵宸文化传媒有限公司

发行部 Circulation Department

发行总监 Circulation Director

谢磊 XieLei 010 - 57439192

市场部 Marketing Department

市场总监 Marketing Director

孔祥彬 KongXiangbin 010 - 85325810 - 807

广告部 Advertising Department

销售总监 sales Director

范欢 FanHuan 010-85325871-802 010-85325981

读者服务部 Reader Service

杜珺 Du Jun 010 - 57458982

印刷: 北京博海升彩色印刷有限公司

如发现本刊缺页、装订错误和损坏等质量问题, 请在当月与本刊读者服务部联系调换 (请将坏书寄回)。

国际标准刊号: ISSN 1673-5163

国内统一刊号: CN11-5480/N

广告经营许可证号: 京朝工商广字第8144号

知识产权声明:

IEEE, IEEE Computer, IEEE中文网站的名称和标识, 属于位于美国纽约的电气电子工程师学会有限责任公司所有的商标, 仅通过授权使用。这些材料的一部分由IEEE Computer英文版翻译而来, 版权归IEEE所有, 并经IEEE授权翻译复制。

IEEE Computer杂志的中文版权, 由美国电气电子工程师学会有限责任公司授予上海灵宸文化传媒有限公司, 并由本刊独家使用。

本刊发表的所有文章内容由作者负责, 并不代表上海灵宸文化传媒有限公司、美国电气电子工程师学会有限责任公司的立场。

本刊内容未经书面许可, 不得以任何形式转载或使用。

编辑团队

执行编辑 Carrie Clark

cclark@computer.org

高级编辑 Chris Nelson

编辑 Lee Garber, Meghan O'Dell

特约编辑 Christine Anthony, Rebecca Torres

多媒体编辑 Brian Brannon

设计和产品

Monette Velasco, Lead

Jennie Zhu-Mai, Lead

Mark Bartosik

Larry Bauer

Erica Hardison

封面设计 Andrew Baker

平面设计 Hector Torres

高级商务拓展经理 Sandy Brown

高级广告经理

Marian Anderson Debbie Sir

产品和服务总监 Evan Buttereld

会员总监 Eric Berkowitz

编辑服务高级经理 Robin Baldwin Manager

编辑服务内容开发 Richard Park

主编

Sumi Helal

University of Florida

helal@cise.u.edu

副主编

Ying Dar Lin

National Chiao Tung University

ydlin@cs.nctu.edu.tw

副主编, COMPUTING PRACTICES

Rohit Kapur Synopsis

rohit.kapur@synopsys.com

副主编, PERSPECTIVES

Bob Colwell

bob.colwell@comcast.net

副主编, SPECIAL ISSUES

George K. Thiruvathukal

gkt@cs.luc.edu

副主编, MULTIMEDIA EDITOR

Charles R. Severance

University of Michigan

csev@umich.edu

2016年IEEE计算机协会主席

Roger U. Fujii

Fujii Systems

r.fujii@computer.org

行业编辑

大数据和数据分析

Naren Ramakrishnan

Virginia Tech Ravi Kumar Google

云计算

Schahram Dustdar

TU Wien

计算机架构

David H. Albonesi

Cornell University

Greg Byrd North

Carolina State University

Erik DeBenedictis

Sandia National Laboratories

绿色和可持续计算

Kirk Cameron

Virginia Tech

健康信息学

Upkar Varshney

Georgia State University, Atlanta

高性能计算

Vladimir Getov

University of Westminster

识别科学和生物识别技术

Karl Ricanek

University of North Carolina

Wilmington

物联网

Roy Want

Google

安全和隐私

Rolf Oppliger

eSECURITY Technologies

软件

Renée Bryce

University of North Texas

Jean-Marc Jéz é quel University of Rennes

视觉、可可视化和增强技术

Mike J. Daily

HRL Laboratories

顾问委员会

Doris L. Carver

Louisiana State University (EIC Emeritus)

Carl K. Chang

Iowa State University (EIC Emeritus)

Theresa-Marie Ryne

Consultant

Bill Schilit

Google

Savitha Srinivasan

IBM Almaden Research Center

Ron Vetter

University of North Carolina

Wilmington (EIC Emeritus)

Alf Weaver

University of Virginia

专帮科研人找工作



科研求职，就到科研圈招聘频道

www.keyanquan.net/recruit 

携手数百家科研院校与机构, 第一时间发布职位需求;
高层次人才、博士后、教职、技术人员、企业岗位……多职位全覆盖;
地域、领域、行业等多级筛选, 高效搜索。

计算科学评论



12

导读 学习技术的前沿 进展

新的创新性技术创造了一系列教学场景，可以帮助学生克服许多传统的学习困难，超越诸多既有限制。

2017年6月刊

封面报道

16

26

34

集协作学习和在线
直播编码于一身的
远程教育模式

苏鲁什·格瑞西
卡洛斯·詹森

设计学创新：跨学
科教育

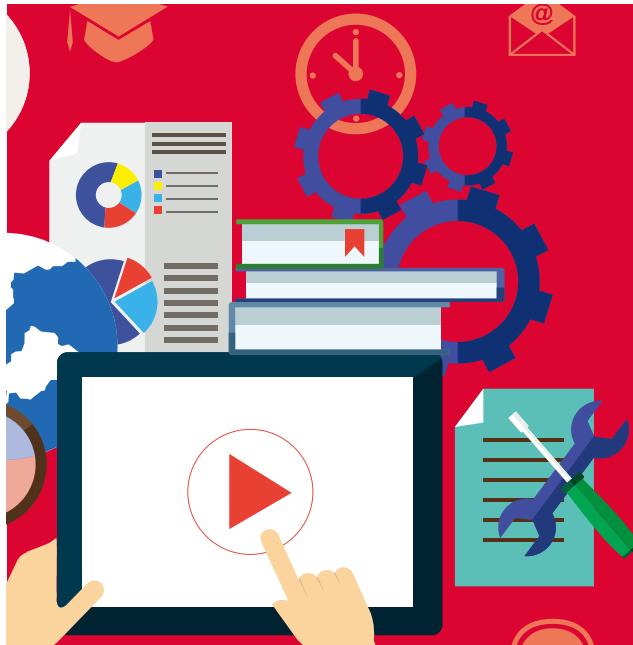
石田亨
榎木哲夫
中小路久美代
十河卓司
京都大学

用以提升教学效果
的情感分析系统

苏耶亚塔·兰尼
帕蒂克·库马尔

2017年6月刊

目录



图片新闻

60 诊断好帮手

62 触觉机器人

42

50

利用“化学式”框架管理网络教育资源

亨利·陈

BlockPy：为引导性程序员准备的开源数据科学环境

奥斯汀·科瑞·巴特
贾维尔·提巴乌
艾利·蒂勒维奇
克利福德·A·沙弗尔
丹尼斯·卡福拉

大师小传

06 艾利希·布洛赫：计算机技术和科学管理领域的先驱

论文分享

09 “小会话，大学问”——如何让聊天机器人读懂对话历史？

吴俣，武威，星辰，李舟军，周明

计算技术

64 计算机教育改革的推手：计算机科学部门和教师

W·理查兹·阿德里安

最近，有关计算机科学部门在“全民计算机科学计划”中的作用进行了小组讨论，尤其讨论了计算机科学的教职员为此做出的贡献，1 并对更广泛的社区兴趣活动提出了一些建议。

计算历史

68 ENIAC 的第二生命

汉斯·诺伊康

世界上第一台通用电子计算机是如何诞生的，又是如何再次焕发活力的？



艾利希·布洛赫： 计算机技术和科学 管理领域的先驱

文 | 乔治·斯特劳 (George Strawn), US National Academies of Sciences, Engineering, and Medicine
坎迪斯·斯特劳 (Candace Strawn)

译 | 刘大明



一篇“大师小传”文章重点介绍了马文·闵斯基 (Marvin Minsky) 和赛摩尔·帕珀特 (Seymour Papert)，他们于2016年离开了人世。在2016年行将结束之时，世界又失去了另一位计算领域的先驱人物：艾利希·布洛赫 (Erich Bloch)，享年91岁。我曾经和他谈了几次关于他在IBM和NSF的工作，他影响了我的整个职业生涯。因此，这是一篇带着个人感情的文章。首先，我将重点讲述他在饱受战争蹂躏的欧洲的早期生活，然后介绍他在IBM取得的成就，接下来是他在担任国家科学基金会 (NSF) 主任时的成就，最后是他在离开NSF之后的职业生涯和获奖情况。

年轻的移民

布洛赫于1925年出生在德国，他的父母都是犹太人，在1939年第二次世界大战爆发前就明智地将他送到了瑞士，当时他年仅14岁。不幸的是，他的父母在那场浩劫中死亡，但是被转移到中

立国的他活了下来。他在瑞士读完了高中，并在瑞士联邦理工学院接受高等教育。1948年，他移民到美国，并于1952年在布法罗大学（现纽约州立大学布法罗分校）获得电气工程学士学位。毕业后，他加入了IBM，当时的IBM是一家机电商业机器公司，还没有专注于计算机领域。

计算机的先锋

在IBM决定专注于计算机领域之后，就开始大规模发力，迅速取得了行业领先地位。（在20世纪40年代，IBM首席执行官老Thomas Watson曾说过，计算机的市场也就能卖出去五台。但他的儿子小Thomas Watson领导IBM进入计算机市场。）作为计算机导向型的IBM的一股“新鲜血液”，布洛赫很快便脱颖而出，被委以重要的开发任务。在20世纪50年代，IBM取得了两项杰出成果，60年代初又取得另一项类似成果。

在20世纪50年代，IBM开发了多台名为“700系列”（701/704/709等）

的真空管计算机。布洛赫开发出了第一个用于商业计算机的铁氧体磁芯存储单元（701使用威廉管存储器，704使用铁氧体磁芯）。到了20世纪50年代中期时，所有的新计算机都将使用晶体管而非真空管。IBM首款晶体管计算机被命名为“7000系列”（包括7040和7090）。1955年，IBM开始与美国原子能委员会实验室（现美国能源部实验室）讨论建造一台基于晶体管的“超级计算机”。该机名为7030 Stretch（见https://en.m.wikipedia.org/wiki/IBM_7030_Stripack），建造目标是比704运算速度快100倍，这台计算机速度将超过4 MIPS（每秒执行的百万指令数）。但遗憾的是，它在1961年终于交付完成时，运算速度只达到了1-2 MIPS。然而，直到1964年CDC 6600交付之前，它仍然是世界上运算速度最快的计算机。² 布洛赫是Stretch项目的工程经理。虽然遭遇了销售上的失败，但IBM从Stretch获悉了高性能计算机的设计思路，通过公司后来推出的计算机

获得了不菲的回报。

接着,布洛赫领导了Solid Logic Technology公司的开发工作(见en.m.wikipedia.org/wiki/IBM_Solid_Light_Technology),该公司为IBM提供的“半微电子”技术为IBM系统及其360计算机提供了支持,该机于1964年发布并于此后不久交付。在20世纪60年代到20世纪70年代末微电子逐步成型阶段,SLT使IBM 360计算机变得更小巧、便宜、运算速度更快。

由于在IBM做出重大技术贡献,布洛赫被提升进入IBM管理层。他的贡献被记录在IBM网站上。(见www-03.ibm.com/ibm/history/exhibit/builders/builders_bloch.html):

他曾(在波基普西实验室)担任多个职务,并于1968年被任命为总监[...]。他于1972年成为企业技术委员会成员,并于1975年被任命为East Fishkill工厂总经理。1976年,他被任命为系统产品部的副总裁,后来又担任数据系统部门副总裁。布洛赫于1980年被任命为技术、数据处理、产品事业部助理执行总裁,1981年8月任公司技术开发人员负责人,并于当月当选为IBM副总裁。

国家科学基金会的创新领导者

1984年,罗纳德·里根总统任命布洛赫担任国家科学基金会(NSF)主任。他是基金会的第一位没有博士学位的主任,也是第一位来自商业界而非学术界的主任。他在任期间,将基础研究与产品创新结合起来,将提高国家的经济竞争力当作第一要务,这一政策引起了一些擅长基础研究的批评者的不满。

科学政策专家迈克尔·克劳(Michael M. Crow)现任亚利桑那州立大学校长,他曾在1994年写道,从纯粹研究到实践研究的转变是“1950年以来科学政策的第一次重大进步”。

布洛赫被《科学》杂志评价为“能干的政治人物、科学研究的强力支持者,在加强基础研究与经济发展的联系上走在了时代之前。”(见www.nationalmedals.org/laureates/erich-bloch#)。布洛赫的数学和物理科学助理主任比尔·哈里斯(Bill Harris)认为,这种方法可以节避免在联邦政府财政紧缩时砍掉NSF的经费。布洛赫说服了里根政府的官员,不仅没有砍掉经费,还要增加预算。

作为他强调竞争力的一部分,他主持落实了Lax报告(因纽约大学的Peter

研究型大学。NSF选择建立的NSFnet以Arpa网络的TCP/IP软件为中心,后来成为互联网之父,就如同Arpanet是NSFnet之父一样。

为了监督这些开发并支持计算领域的基础研究,布洛赫重组了NSF,新的NSF包括计算机领域(名为CISE计算机和信息科学与工程)以及物理、生物、地球和社会科学院(成立于20世纪90年代)和科学教育部门。CISE现在为学术类计算机科学家的研究提供了大量支持。

荣誉等身

在NSF主任的6年任期结束(该职位无人连任)后,布洛赫与他人共同组建了一家咨询公司——华盛顿顾问集团。它是“由国际知名的科学和工程研

布洛赫在担任NSF主任期间,将基础研究与产品创新结合起来,将提高国家的经济竞争力当作第一要务,这引起了一些批评者的不满。

Lax而得名)中的建议⁵,该建议是早1982年提交给NSF和其他机构的。报告指出,大学中缺乏超级计算机设备,并认为,无法获得超级计算机使用权限的科学家正在避开那些需要计算能力的问题,这危及到了美国科学家在某些研究领域的领导地位。随后,NSF着手建立超级计算中心,为那些无法使用政府部门的超级计算机(如能源部实验室)的领域的大学科学家服务。“Lax报告”进一步建议,这些超级计算中心还应通过计算机网络,连接到美国各个

究和高等教育机构的一小群经验丰富的成功领导者于1996年创立的。该公司的目标是为政府、大学、研究基金会和企业提供关于研究战略、管理和领导力上的意见”(见www.aresstrategies.com/wag)。

由于布洛赫的杰出贡献,他获得了许多荣誉和奖项。比如,他因与IBM/360计算机相关的开创性开发对计算机行业的革命性贡献被授予美国国家技术勋章,他曾任美国国家工程院院士、瑞典工程科学院院士、IEEE研究员、日

本工程院外籍院士。1989年，布洛赫获IEEE美国“杰出公共服务活动委员会奖”和IEEE 1990创始人奖章。

前文提过，布洛赫没有博士学位。但他获得了科罗拉多矿业学院、圣母大学荣誉博士学位和伦斯勒理工学院荣誉博士学位。美国马萨诸塞大学阿姆赫斯特分校、乔治华盛顿大学、纽约州立大学布法罗分校、罗切斯特大学、欧伯林学院和华盛顿学院的荣誉理学博士学位，以及俄亥俄州立大学科学与工程荣誉博士学位。

我的职业生涯始于IBM，更确切地说，是始于1964年IBM系统/ 360计算机发布之时。我记得由SLT所产生的兴奋，那是一种新的、更自动化的构建计算机的方式。在我的后来职业生涯中，曾在布洛克创建的NSF CISE担任过总

监，就是为了成为NSFnet项目团队的一员。《大师小传》专栏记述的许多杰出人物都影响了我的职业生涯，但也许没有一个人的影响比艾利希·布洛赫更直接。
■

参考文献

1. G. Strawn and C. Strawn, “Masterminds of Artificial Intelligence: Marvin Minsky and Seymour Papert,” *IT Professional*, vol. 18, no. 6, 2016, pp. 62–64.
2. G. Strawn and C. Strawn, “The Father of Supercomputing: Seymour Cray,” *IT Professional*, vol. 17, no. 2, 2015, pp. 58–60.
3. S. Roberts, “Erich Bloch, Who Helped Develop IBM Mainframe,
4. E. Langer, “Erich Bloch, IBM Pioneer Who Later Led National Science Foundation, Dies at 91,” *Washington Post*, 28 Nov. 2016; wapo.st/2keVzne.
5. P.D. Lax, *Report of the Panel on Large Scale Computing in Science and Engineering*, 26 Dec. 1982; www.pnl.gov/scales/docs/lax_report1982.
6. G. Strawn, “Masterminds of the Arpanet,” *IT Professional*, vol. 16, no. 3, 2014, pp. 66–68.
George Strawn is the director of the Board for Research Data and Information at the US National Academies of Sciences, Engineering, and Medicine. He is the former director of the National Coordination Office for the Networking and Information Technology Research and Development Program (NITRD). Contact him at gostrawn@gmail.com.



听听Diomidis Spinellis说什么

播客：“交易工具”

www.computer.org/toolsofthetrade

Software

IEEE  computer society

乔治·斯特劳恩是美国家科学、工程和医学院研究数据与信息委员会主席。他曾任国家网络与信息技术研究开发项目(NITRD)联络办公室主任。联系方式：gostrawn@gmail.com。

坎迪斯·斯特劳恩曾任高中、社区大学和大学教师，现已退休。联系方式：castrawn@gmail.com。



“小会话，大学问”

——如何让聊天机器人读懂对话历史？

文 | 吴侯，武威，星辰，李舟军，周明，北航 & 微软研究院

来源 | PaperWeekly (微信公众号: PaperWeekly)



人工智能现在的火热程度

大家有目共睹，凯文凯利在《必然》中预测未来时代的人工智能可能会重新定义人类的意义，但时下的人工智能发展水平显然还没有大家想象得那么美好，仍然需要我们不断地探索。目前众多研究人员正深耕其中，作为其中代表的聊天机器人（Chatbot），已然成为科研界研究的热点。

通常来说聊天机器人实现方式有两种：一种是基于生成式，即机器人“自己说话”，一个字一个字创作出回复语句来。另外一种是基于检索式，即机器人“转发”别人的话。从互联网大家的话语中寻找合适的回答予以转发。

现有的生成结果由于技术所限，或多或少都存在语句不流畅的问题，但能够做到“有问必答”偶尔还能蹦出“彩蛋”。检索的模型一般流畅性方面无须担心，不过随着目前网络资源的日益丰富，语料规模也越来越大，如何从众多语句中选择合适的句子作为回复是检

索式聊天机器人的核心问题。

虽然目前市场上的聊天机器人众多，但我们见到的那些貌似都不是那么聪明。最直观的一个体现就是前后不连贯，上下难衔接，因此在进行检索的时候考虑历史信息则显得尤为重要。那么如何让机器人理解对话历史信息从而聪明地进行回复呢？微软亚洲研究院的研究员们提出了一个模型 Sequential Matching Network (SMN)。相关论文的作者吴侯谈到“SMN 模型可以让聊天机器人准确的理解当前和用户的对话历史，并根据历史给出最相关的回复，与用户进行交流，达到人机对话的目的。”

但这在实现的过程中也遇到不少难点，“精准计算聊天历史和候选回复的语义相似度十分困难，主要的挑战有两个方面：(1)由于聊天历史信息繁多，如何将历史中重要的词语、短语以及句子选择出来，并通过这些重要部分刻画聊天历史，是一个亟待解决的问题；(2)如何对聊天历史中的各轮对话进行建模，如何判断对话历史中的跳转，

顺承等关系，也是一个棘手的问题。”

那么论文作者提出的 SMN 模型又是如何解决这两个问题的呢，接下来我们将为您解读。

SMN 模型一共分为三层。第一层为信息匹配层，对之前的历史信息和待回复的句子进行匹配度计算：同时进行词语 (embedding 向量) 和短语级别 (GRU 表示) 的相似度计算。然后再把这两个矩阵分别作为卷积神经网络的一个 channel，利用 CNN 的 max-pooling 进行特征抽象，形成一个匹配向量。第二层为匹配积累层，利用一个 GRU 神经网络，将每一句话和回复所计算出的匹配信息进一步融合。第三层为匹配结果预测层，利用融合的匹配信息计算最终的匹配得分，在预测时他们使用三种策略，分别是只利用 GRU2 最后一个隐藏层 (SMN_last)，静态加权隐藏层 (SMN_static)，和动态加权隐藏层 (SMN_dynamic)。在进行检索的时候，他们将最后一句的信息结合上文历史信息中的 5 个关键词 (历史信息

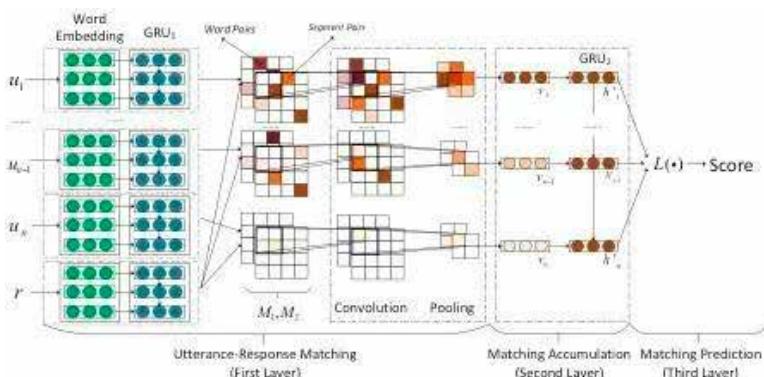


图1. SMN网络架构

中用 tf-idf 进行筛选, 选择 Top 5 关键词)在系统中进行检索, 然后用上述 SMN 网络对候选结果打分从而选出回复句。

作者分别在 Ubuntu 语料(大型公开计算机相关求助与解答语料)和豆瓣语料(作者从豆瓣小组的公开信息中

爬取并在论文中公布)上进行了实验, 分别以 Rn@K 和 MAP、MRR、P@1 为评价指标, 取得的结果均为目前最好。

作者表示“SMN 不同以往的模型, 第一步不进行上下文的建模学习, 而是让每句话和回复进行匹配度计算, 这样可以尽可能多的保留上下文的信

息, 以避免重要信息在学习上下文的向量表示时丢失。而且 SMN 在对上下文句子关系建模时, 考虑了当前回复的影响, 使得回复成为一个监督信号, 这样可以更准确的对上下文历史进行建模。”并且吴保向我们透露“据我们所知, 我们公布的豆瓣语料是第一个人工标注的中文多轮对话语料。”

在谈及目前的方法还有哪些不足之处时, 作者表示“目前我们的检索方式有时找不到正确的候选句, 所以第一步的检索方法还有待改善, 另外虽然我们的模型可以建模当前回复和历史信息的关系, 但还是无法避免一些逻辑上的问题, 不过这也是我们未来工作的重点, 我们将继续提高候选回复在逻辑上的连贯性”, 让我们共同期待他们在未来所取得的成果。

关于中国中文信息学会青工委

中国中文信息学会青年工作委员会是中国中文信息学会的下属学术组织, 专门面向全国中文信息处理领域的青年学者和学生开展工作。



关于PaperWeekly

PaperWeekly 是一个推荐、解读、讨论、报道人工智能前沿论文成果的学术平台。如果你研究或从事 AI 领域, 欢迎在公众号后台点击「交流群」, 小助手将把你带入 PaperWeekly 的交流群里。





中国计算机学会(CCF) 会员专属权益

▽ 项目

▽ 会员

▽ 非会员

▽ 说明

CCF通讯 (CCCF)	免费	480元/年	全年12期，纸质版，每月邮寄；另有PDF版及IPAD版
中国计算机科学技术年度发展报告	免费	96元/册	权威报告，每年一册，电子版
特价加入ACM	100元/年	240元/年	CCF会员特价加入ACM会员，享受ACM会议优惠、ACM电子版通讯、电子刊等
YOCSEF	免费	付费	除北京总部外，CCF已在24个城市建立了分论坛，每年活动逾百次
会员活动中心活动 (CCF城市分部)	免费	付费	CCF已在24个城市建立会员活动中心，每年活动逾百次
计算机职业资格认证 (CSP)	优惠	付费	一年3次，每年近万人参加，认证结果受到知名高校及企业认可
CCF电子刊	免费	无	每月6期
数字图书馆	免费	无	期刊、培训视频等资料
CCF网站信息发布	免费	无	会员登陆会员系统后在CCF官网发布会员成就、推荐会议、求职、招聘等信息
选举权、被选举权、参与学会治理	专有	无	
中国计算机大会 (CNCC)	优惠	全价	每年一次，每次参会人数逾3000人
学科前沿讲习班 (ADL)	优惠	全价	每年10期，每期邀请该领域国内外顶级专家作为讲者
IEEE CS准会员资格	65元/年	+240元/年	享受IEEE CS会议优惠、每月3期电子刊等
专业委员会议	8折	全价	CCF拥有35个专业委员会，每年活动近百次
CCF13种会刊论文版面优惠	8.5折	全价	限第一作者

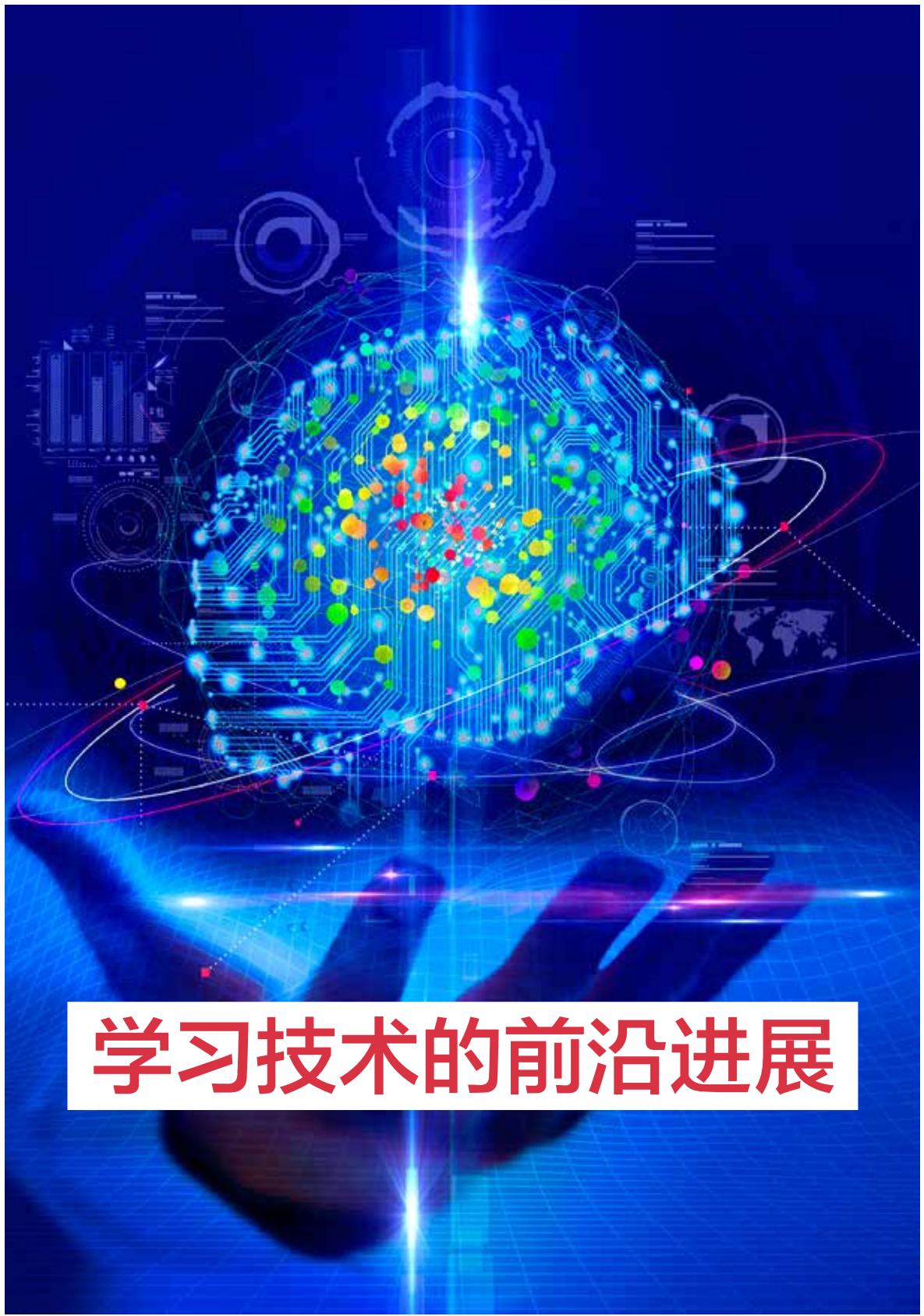
获得以上CCF会员专属服务，只需支付**200**元/年会费！

地址：北京科学院南路6号
电话：(86-10) 62648654
网址：www.ccf.org.cn

通信：北京2704信箱，100190
邮箱：membership@ccf.org.cn



扫码成为CCF会员



学习技术的前沿进展

文 | 张可昭 (Carl K. Chang)，艾奥瓦州立大学 (Iowa State University)
 索雷尔·赖斯曼 (Sorel Reisman)，加利福尼亚州立大学 (California State University)
 埃德蒙多·托瓦尔 (Edmundo Tovar)，马德里理工大学 (Technical University of Madrid)

“学习技术”可以指代一系列教育手段，包括教育与学习可能关注的一些新技术手段、创造出新型教学环境的基础构架技术，以及辅助教学的技术相关性手段。

《计算科学评论》的大部分读者是高等院校或研究机构的教学工作者，都会从事编程教学（这是本科计算机教育的基础）。为了提高学生的编程学习水平、辅助学生的编程学习，教育工作在不断创新。但我们知道，即便如此，很多学生仍不会考虑从事编程工作。编程教学本就存在困难，毕业生的全球就业市场又发生了变化，让这一问题更加复杂。程序员开发的项目的复杂度变高，系统及应用开发的管理开始具有国际性，这让程序员不再只负责某一个软件项目，而是常常同时在多个国际团队中工作。

今天的计算机程序必须能够承担多文化及多平台任务，因此，在设计系统和应用时，工程师需要采用一种全面

的跨学科手段。这种多元性任务为教育者带来了独特的挑战，他们的学生将来进入职场后会与各种跨学科团队合作。如果我们受到学期教学任务的限制，又怎么能帮助学生在高度差异化的就业环境中取得成功？

新的创新性技术创造了一系列教学场景，可以帮助学生克服许多传统的学习困难，超越诸多既有限制。

“课堂”正在日渐从物理空间变得抽象化，教师和学生正在采用一系列先进的技术性教学手段，这将对学生大有益处。

在这样的环境中，最令教师沮丧，同时也许也是限制其职业发展的一个因素，是学校在评估教学质量时对学生意

见调查的制度性依赖。这些调查的评分常常以班级学生的平均打分体现，几乎不关注评分与班级学习成果之间的关系。尽管学校可以分析调查结果与课程总体平均绩点 (GPA) 间的相关性，但大部分学校无法确定调查结果与班级表现的任何因果关系。学校几乎无法从收集到的数据中，解析出个体学生表现与其对任课教师的评估意见间的任何关系。

随着在线教育的日益普及，我们现在可以随时通过大量参与在线学习的学生，采集到多种新型数据，从而进行传统的课堂教学无法想象的新型数据分析。此外，这样的分析能够帮助我们更好的理解教学质量对教学结果的影响。

最后，对于非教学领域的研究型计算科学专家而言，开放教育资源 (open educational resources, OERs) 也值得关注。很多论文在不断研究这一问题，但其研究的视角常常是十分传统的面向对象程序设计 (object-oriented programming,

如果我们受到学期教学任务的限制，又怎么能帮助学生在高度差异化的就业环境中取得成功？

OOP)。然而，OERs 同样是技术元素，是以不同形式呈现的教育资源，可以被反复应用在教学过程中，这使得 OERs 在教育中的重要性日渐增加。

我们一共收到了超过 30 份优秀的投稿，内容覆盖学习技术的全部领域。尽管如此，严格的评审过程将这期杂志聚焦在了那些能够在现实层面呼应以上挑战的文章，这些文章内容与我们的读者的兴趣高度一致。

本期内容

在《BlockPy：为引导性程序员准备的开源数据科学环境》一文中，Austin Cory Bart, Javier Tibau, Eli Tilevich, Clifford A. Shaffer 和 Dennis Kafura 研究了越来越多的学习者在计算机教育中追求正式和非正式学习经验的问题。这些研究往往难以在传统的计算机课程中找到相关性，既要强调抽象概念，要么依赖于非文本化设置。对于传统和非传统的计算经验较少的学习者，教育工作者需要提供与学生能力和动机水平相关的特殊工具，使

得体验变得有趣。作者介绍了一个开源的基于 Web 的平台，用于学习 Python 编程即 BlockPy。与其他类似环境相比，其限制较少，这种多功能平台可用于正式和非正式的场合。因此，不论是在独立学习的过程中，还是在导师主导的小组课程或一对一的辅导中，这个变成环境都是行之有效的。

在《集协作学习和在线直播编码于一身的远程教育模式》一文中，Soroush Ghorashi 和 Carlos Jensen 研究了如何利用一种叫 Jimbo 的工具，通过合作技术，改善线上教育。Jimbo 是一款合作型的设计出集成开发环境 (IDE)，可供 HTML5 开发。作者在一个实验性的数据科学线上课堂上使用了 Jimbo，验证了它的效果。Jimbo 可以让学生在合作中把注意力集中在教学的关键概念上。从学生那里获得的不同的调查验证了之前的研究：双人编程（两个学生在一台计算机上合作解决一个普通问题）是一种有效的合作学习技术。与单独编程相比，双人编程能产生更好的软件代码，学生的表现也要好得多。

Sujata Rani 和 Parteek Kumar 在

《用以提升教学效果的情感分析系统》一文中把情感分析定义为“用户对一篇文本的意见识别和分类的过程（如课程调查或网上论坛评价等）……用以判断用户对特定主体或实体的态度”并解释道“其对教学十分重要”。对比起单一的意见调查，情感分析事实上可以通过分析学生的作业得到更广阔的图景。Rani 和 Kumar 通过以“课程时代”持续一年以上的学生成绩评价和评级以及对一大学两年间不同教师所上的 25 门课程的评级为数据库，验证该情感分析模型有效。此类分析有助于教员根据学生的情感和其学习效果调整课程。

在《设计学创新：跨学科教育》一文中，石田亨、榎木哲夫、中小路久美代以及十河卓司以深入的视角展现了现代计算在维系和提升社会福祉中的作用。2011 年日本发生大海啸后，福岛第一核电站核反应堆的堆芯融毁，这场灾难促使日本各方意识到一种迫切的需求：重新开发教学课程，专注于和重建城市及社区相关的设计议题。作者们将注意力从传统的专注发现问题和解决问题的计算模式上挪开，转而在一个更大的社会环境里思考这些问题。他们描述了创建一个 21 世纪的设计学院的过程。京都大学设计学院的课程处理技术上的问题，其中既包含以人为本的计算，也包括对社会议题的关注（这通

关于作者

常是社会学家的研究领域)。这个学院有一些突出的特点令它与众不同: 将技术解决方案和社会参与结合起来、领导跨学科团队的建设和管理、产业界人士的积极参与并可借此提升商业可行性。

最后, 在《利用“化学式”框架管理网络教育资源》一文中, Henry C.B. Chan 提出了一种新方法来刻画学习对象(LO)的特征。在传统上, 学习对象一般被比作与 OOP 中使用的目标特征类似的软件对象。虽然学习对象有很多不同的“标准”, 但在这方面都是非常相似的。本文作者提出了一个全新的模式和观点, 该观点以分子化学的概念为基础: 即一个单独的学习对象就像一个原子, 通过描述的微观和宏观属性的特质, 可以形成与分子类似的学习对象群。通过这种方式, 按照多样化的标准或属性将学习对象如原子键一样结合在一起。这种替代模式与我们传统上的处理方式有一定距离, 而传统方式已经几乎纳入全部的学习对象标准中。本期封面文章的编辑人员认为, 此文中提出的概念是一个从外部思考的绝佳例子, 可以说是重新定义了人们理所当然地沿用多年的模型。



习技术是一个不断发展的研究领域, 在该领域内, 新兴的全球学习型企业时刻欢迎新的观念、环境和实践。令

张可昭是艾奥瓦州立大学计算机科学及人机交互教授、软件工程实验室主任。他的研究兴趣包括软件工程、服务计算及成功老龄化。张可昭在美国西北大学获得计算机科学博士学位, 曾任《IEEE 软件》主编(1991~1994)及《IEEE 计算机》主编(2007~2010)、IEEE 计算机学会荣誉主席(2004)。他代表 IEEE 计算机学会, 与美国计算机学会的彼得·丹宁(Peter Denning)共同担任联合模式计算课程专门工作组主席, 工作组发布了《CC2001 计算机科学报告》(CC2001 Computer Science Report)。他是 IEEE 会士、美国科学促进会(AAAS)会士、欧洲科学院院士。他的联系邮箱是: chang@iastate.edu。

索雷尔·赖斯曼是加利福尼亚州立大学富勒顿分校信息系统教授、加利福尼亚州立大学校长办公室学习与在线教学多媒体教育资源系统(MERLOT system, www.merlot.org)总经理。他的研究兴趣包括开放教育、数字资料库、计算机教育以及软件工程。莱斯曼在多伦多大学获得文学硕士及计算机应用博士学位, 曾任 IEEE 计算机学会荣誉主席(2011)。他是 IEEE 出版、产品与服务委员会成员, 曾是开放教育联盟(Open Education Consortium)董事会成员; 他还是一位富布赖特(Fulbright)OER 专家。他的联系邮箱是: sreisman@calstate.edu。

埃德蒙多·托瓦尔是马德里理工大学企业信息技术教授。他的研究兴趣包括开放教育、电子化学习、学习技术以及认可机构。托瓦尔在马德里理工大学获得计算机工程博士学位, 曾任马德里理工大学质量与策略规划学院副院长(2007~2012)、开放教育办公室执行主任(2008~2016)。他曾当选开放教育联盟董事会成员(2009~2013)。托瓦尔也是 IEEE 教育学会教育活动与奖励委员会副主席(2013~2018)。他的联系邮箱是: etovar@f.upm.es。

人惊讶的是, 物理上的距离在很大程度上变得无关紧要, 学习的时间框架似乎是瞬间的。我们预计, 新的学习范式将继续重塑“教室”的形式, 按需的、移动平台化的“教室”将成为学习的新常态, 数据驱动型的教学活动和学习成果

评估将推动该领域的发展和进化。大数据分析驱动型教学正迅速成为智能机器导向教学的重要元素, 最终将打造出基于物联网的、真正的适应性学习环境。一个令人无比兴奋的学习世界即将展现在我们眼前。C



集协作学习和 在线直播编码于一身 的远程教育模式

文 | 苏鲁什 · 格瑞西 (Soroush Ghorashi) , 卡洛斯 · 詹森 (Carlos Jensen) , 俄勒冈州立大学
(Oregon State University)

译 | 王欣怡

如何通过线上课程提升计算机专业学生的编程兴趣和技术水平? 为了解答这一问题, 作者打造了一个功能丰富的远程协作学习环境进行试验。试验显示, 远程结对编程、在线直播编码, 以及实时反馈代码运行结果等学习方式能够增强学生的学习效率。

在线课程在高等院校中的热度日益攀升, 特别是对于那些睡着都会抱着电脑的计算机科学专业学生来说。大部分开放式在线课程 (MOOCs) 会无限制地免费对公众开放。当然, 一些学校也会把比较小众或课表安排不下的课程, 通过线上平台开放给校内外学习者们, 通常这类课程会由教员或者助教来主讲。数据显示, 从 2013 年到 2015 年, 参加俄勒冈州立大学线上计算机科学课程的学生人数从 100 人增加到了 2000 多人。协作学习模式非常适用于计算机科学这类学科的线上课

程, 特别是需要大家共同完成某些项目的时候。这就像一个软件工程, 70% 的时间都需要相互协作进行。研究表明, 相比于独自完成学习和编码, 在协作学习模式下进行学习和编码的学生表现更加优异, 他们的编码速度比前者快一半, 而且代码质量更高。更好的学习效果能增加学生的自信心, 从而营造一个良好的学习氛围。

在编程教学中, 结对编程是一个行之有效的协作方法。顾名思义, 结对编程就是两个学生联合起来解决某个问题, 其中一位学生扮演船手的角色, 负责操作键盘和鼠标, 另一

位学生扮演航海家角色，负责在一旁审核编写出来的代码是否有误。研究显示，通过结对编程写出来的代码质量要优于个人独自编程。然而，由于受地域限制，要在线上课程中进行结对编程不容易。无论是从交流方式的局限还是大规模的学员群体看，沟通是在线课程所面对的一大问题。同时，教员需要有一些特殊的模式和方法来实时跟踪每个学员的学习情况，并能在适当的时候纠正学员的学习错误，而不是一味的只顾授课不顾结果。

对于编程相关的开放式在线课程，为保证学员学习质量，我们最好能够通过浏览器这种能够适用于各类移动设备的通用型工具，为学生营造一个畅通无阻的协作编程环境。如此一来，学生和教员就可以把所有的精力都放在课程内容上，而不是在繁琐的应用设置和系统故障中费尽周折。

为了实践这一想法，我们通过研究大量文献，明确了能够更帮助在线课程初学者提高学习效率的功能设计需求，并据此设计出集成开发环境（IDE）Jimbo。使用者可以通过 Jimbo 在 HTML5 页面上进行现场编程，并能够实时得到代码运行结果的反馈。但正如报道《建立协作编程课程机制》所说，这些概念目前也只是零星出现在不同的编程工具中，尚未能被整合到一个单独的 IDE 里。

[结对编程在网上课堂上难以实现，因为相距遥远，彼此交流时在程序语境上经常要换来换去。]

Jimbo 可以支持协作编码和实时预览编码结果。为了验证协作学习模式有助于更快地写出高质量代码，我们在俄勒冈州立大学研究生志愿者的帮助下进行了两个用户研究。第一个研究，通过现场完成编码任务，探索结对编程在编程学习中起到的作用。接着我们通过在线课程进行了试验，以此判断对于编程工具来说，哪些功能具有独立性，哪些功能需要交叉配合使用，而哪些功能对促进学习毫无作用。

研究结果显示，通过使用 Jimbo 进行结对编程的学生能够出色地完成编码任务，无论是从速度还是质量上来说都要优于独自编码。

协作学习实践情况

为了找出能最大限度提升学生学习质量的方法，我们搜集了各类计算机科学会议的会议记录，如 ACM 计算机支持协同工作（CSCW）国际会议、ACM 计算机系统人为因素会议（CHI）以及美国计算机学会计算机科学教育

特殊兴趣组技术研讨会。我们从中选取了超过 50 篇提及传统和远程结对编程、协作和主动学习以及实时编程的文章，掌握了对初级计算机科学学习者有益的关键学习步骤。

支持用户交互

我们发现了协作学习模式中 4 个核心步骤。

同步协作—同步协作功能不仅能够统一不同时区学生和教员的学习进度，还可以将课程与其他编程任务相结合。除此之外，处于同步协作学习模式下的学生们能够实时共享一份教学文件。值得注意的是，系统必须保证共享代码的无误性，因为任何一个开发者都随时可能编译或运行这些代码，这必然就可能会对其他正在编译这段代码的用户产生影响。Collabode 工具通过建立一个自动安全缓冲区来很好地解决了这一问题，通过这样一个缓冲区，系统只会共享修改后正确无误的代码。

其他的一些编码工具则是在程序发生改变时实时通知使用者，提醒

支持协作式编码课程

当前，在线课程的需求量急剧攀升，研究者为了找到有助于提高在线课程学生学习质量的学习工具，进行了大量研究和试验，最后发现了决定学习工具使用效果的4个重要因素：互动机制、结对编程、在线直播编码，以及及时的代码运行情况反馈。

互动机制

大规模在线开放课程（MOOC）的学生无法像在校学生一样能得到教员及时的指导和帮助，协作和教学支持的匮乏极大限制和降低了在线课程的教学质量和学习效率。

结对编程

研究者们研究了在教学过程中，结对编程比独立编程效率更高的原因。他们发现，来自搭档的压力会让学生更加专注和认真，因此结对编程的编码效率要强于独自编程。除此之外，研究者们还发现，结对编程能够为女性计算机专业学生提供更公平的参与机会。结果显示，协作学习能够帮助学生提高领导力、协作力以及冲突管理能力。

在线直播编码

在编码课程中，在线直播编码也是一种备受青睐的教学方式。学生能够实时观摩教员的编码过程，从而更好地理解其编码思想。虽然这种教学方式比照本宣科更有效，但还是会让学生处于被动。有人曾试验过，如果由学生主导进行在线编码演示，教员在一旁观看，能够更加激发学生的学习热情，同时也有助于教员发现学生的问题。研究者称，思维过程比最终结果更值得关注。

及时反馈代码运行结果

代码运行结果的及时反馈，能够有助于学生随时在线编码过程中发现问题，提高学习热情。目前的编程课程，学生们通常只有在程序编码完成后才能知道代码的运行情况。如果出现了Bug，只能又返回到程序中找到相应的地方进行修改编译，然后再次运行查看。如此冗长的修改过程会影响搭档的进度，特别是在修改用户界面展示的程序时，会影响到用户体验。

参考文献

1. L. Williams and R.R. Kessler, "Experimenting with Industry's 'Pair-Programming' Model in the Computer Science Classroom," *J. Software Eng. Education*, vol. 11, no. 1, 2001, pp. 7–20.
2. L.L. Werner, B. Hanks, and C. McDowell, "Pair-Programming Helps Female Computer Science Students," *J. Educational Resources*, vol. 4, no. 1, 2004, article 4; doi: 10.1145/1060071.1060075.
3. A. Gaspar and S. Langevin, "Active Learning in Introductory Programming Courses through Student-Led 'Live Coding' and Test-Driven Pair-Programming," *Proc. Conf. Education and Information Systems, Technologies, and Applications (EISTA 07)*, 2007; cereal.forest.usf.edu/clue/publications/2007-EISTA.pdf.

他们注意避免编码冲突。如编程工具 FASTDash 会向使用者共享开发人员工作区的可视化数据，Palantír 则会通知使用者谁正在修改哪一部分的代码，并提醒相关人员注意编码冲突，而 Syde 编码工具则是通过抽象语法树（AST）分析方法来减少误报率。

集成通信功能—在集成开发环境

（IDE）中融入通信交流功能，能够帮助学习者在编码的同时相互讨论和学习。充分的交流沟通是一个团队成功的关键，而通常编码团队的沟通仅限于坐在一起进行的会议讨论。针对这一限制因素，之前的研究团队尝试通过邮件、发送讯息或利用源代码控制系统的一些功能来达到沟通目的。然

而，无论是 Skype、交互式远程通信（IRC），还是邮件等渠道，都独立于 IDE 之外，而要不断在代码编写系统和交流沟通系统之间来回切换，无疑会降低程序员工作效率。

用户提醒功能—在结对编程中，程序员需要注意和兼顾搭档的编码情况，以达到相互配合的目的。有的人

需要异步编程，而另一些人又需要实时查看代码结果，在了解和配合对方工作的过程中必然会消耗程序员一部分精力，要兼顾编码效率和充分协作并不是件容易的事。

实时预览功能—最近新兴的一些编码系统，像 Superglue 和 Flogo II，都把实时预览作为在线编程的一个部分，用户可以通过实时预览功能反复执行编写好的代码，同时系统会输出结果副本。然而，实时预览不仅仅是查看代码输出结果，更重要的是通过实时预览，编程者可以从中找出代码漏洞并进行修正。当学习者在系统上进行实施编码时，我们需要将实时预览功能整合到结对编程功能中。

支持编码功能

由于在协作学习过程中需要完成编码任务，初学者们需要一个无需进行额外设置的网页式编程环境。为了减少代码错误，提高编码效率，系统应具备代码自动补全和协助编辑功能，包括过渡支架、代码回滚以及自动备份。

Jimbo 学习环境

在线编码课程中，如果编码和应用系统出现问题，是没有专门的 IT 人员来帮大家解决恢复，因此，为了保

为了支持协作编程，学生们需要一种简单、网页式的、无需安装的编程环境。

证课程的顺利进行，教员们不得不选择一些学生们容易上手和理解的编程工具。Jimbo 编程环境只需通过网页就能打开使用，同时在结对编程界面加入了实时预览功能。正如图 1 所展示的，Jimbo 的系统界面含有多个面板和按钮，使用者可以在编码过程中查看教员发布的共享文件。该系统界面不仅能够让学生更容易理解和记忆，还和市面上常用的 IDE 进行了呼应，以减少使用者的适应时间。学生们只需完成在线课程的基础网页设置，就能够轻松使用 Jimbo 开始学习啦。

USER STUDIES 用户研究

为了深入探究 Jimbo 各个功能（包括线上和线下）的实际效果，我们邀请了亥俄州立大学的学生志愿者们参与我们的两个试验。试验对照组的学生们依然用他们平时使用和熟悉的工具独自进行完成编程任务。试验组的学生两两组队通过结对编程的模式完成任务。其中，在线进行任务的学生会使用 Jimbo 编程系统完成结对编程，以此尽量减少合作障碍，让线上的学生们能够体验到和线下结对编程同样

的协作效果。

我们先通过线下编程来验证 Jimbo 功能的必要性，再通过线上编程来探究其实用性。为保证每位志愿者的知识存量和概念体系相同，试验前提设置为志愿者接受的都是相同程度的编程培训。试验组中，线上教员对学生进行简单的 Jimbo 使用培训，然后让他们根据当天所学习的知识完成一个编程任务。试验最后，我们就志愿者的学习体验进行了采访。

线下试验设置

在线下试验中，我们选择了有 16 名学生参与学习高级数据可视化课程。课程目标是让学生能够使用 d3.js 库及其他技术，将各类数据可视化，课程要求学生在短时间内掌握大量的新概念。

试验设置及志愿者分组—16 名学生成志愿者（包括 12 名男性和 4 名女性）均有计算机科学本科背景。我们随机将志愿者对半分为试验对照组和试验组，并将试验组学生又两两分为 4 个结对编程小组。其中 15 位志愿者有结

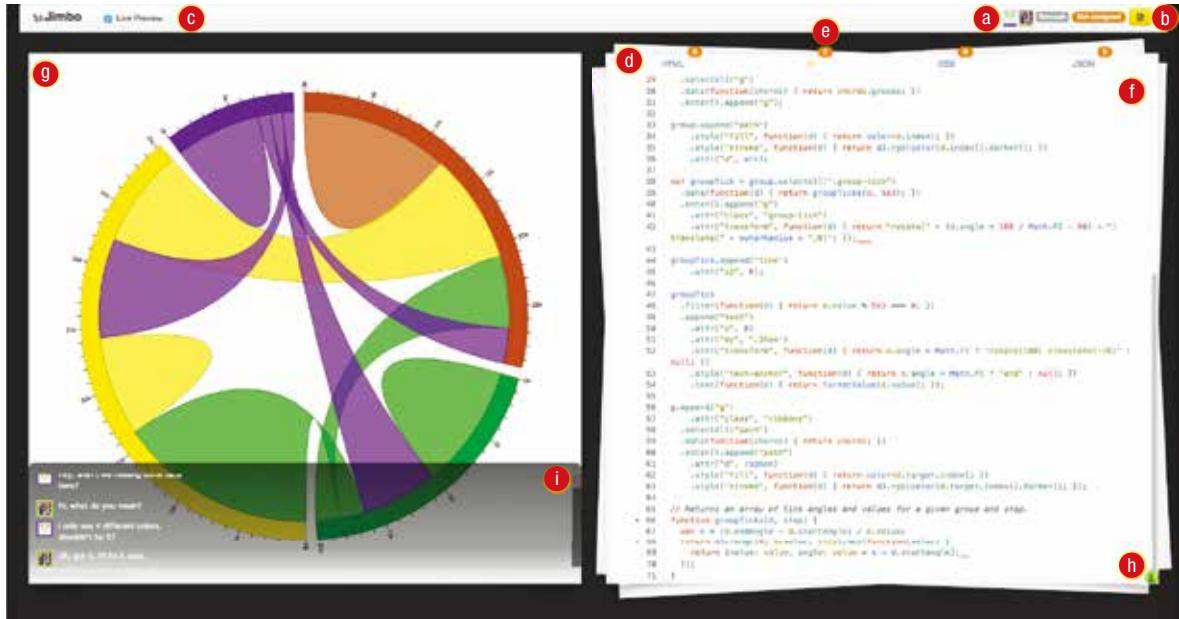


图 1.Jimbo 的结对编程会话界面。(a) 开发者头像及用户名显示; (b) 快速分享按钮; (c) 开启实时预览按钮; (d) 编码区。该区域会显示 HTML5、JS、CSS、JSON 的当前开发者人数。图片中显示有两名开发者正在使用 JS 代码编辑页面进行编码, (f) 区为代码编辑页面视图, (g) 区为代码实时预览区域。不同编辑页面的开发者可以在 (h) 区获取整个编码项目的代码变更提醒。除此之外, 开发者们可以通过 (i) 区进行交流沟通。

对编程经验, 剩下一人仅是使用过实时协作工具。我们没有对志愿者进行结对编程培训, 试验对照组的学生们集中到一个教室里接受教员授课, 然后自由选择编程工具进行编程。

试验组的学生两两挨坐, 分别用自己的电脑在 Jimbo 上进行结对编程。我们邀请了一位志愿者帮助他们远程解决 Jimbo 使用问题。试验组和试验对照组的任务都是在 40 分钟内用 d3.js 库建立一个条形图。

试验结果—所有试验组的学生都在规定时间内完成了任务。然而, 试验对照组的学生无一完成任务, 而且其中

仅 3 名学生做出了较为成型的结果。为了找到这一现象原因, 我们通过试验的观察以及对试验者们的采访进行了分析。所有的参与者都表示, 在学习过程中进行实际操作, 比说教式学习更有效率。一名试验者说: “在大学学习中,

老师只会给我们一些编写好的程序让我们自己去运行, 然后再解释下这些程序的编写思路。我以为自己对这些原理了如指掌, 然而每次轮到自己独立完成作业时, 就两眼一抹黑了。通过这次试验中的实际操作, 我对知识的理解更扎实也更实际了。”

试验组的志愿者们表示, 结对编

程加深了他们对概念和细节的理解。一名试验组志愿者解释道, “当我们作业中遇到困难时自然会去找他人求助。因此, 如果我们都在做同一个项目, 就能够随时互相帮助, 而且还能减轻心里压力”。

试验组的学生还表示, 实时预览能帮助他们更快地完成任务。一位志愿者表示, 相比 jsfiddle .com, 他更喜欢 Jimbo 这样的编程环境, 他说, “每次我在 jsfiddle .com 上编程时, 要不停地点击‘RUN’按钮来进行测试, 而 Jimbo 不需要, 这样可以减少操作引起的干扰。”

通过对试验结果的研究和采访，我们更加明确了传统课堂教学降低交互式教学效果和学习体验的原因，包括课程设置和沟通，以及缺乏学生学习情况的实时反馈。

系统零设置—通过对试验对照组的观察发现，他们花费了大量时间来设置编程系统，即使这些工具是他们再熟悉不过的。因此，这就证明了编程系统的零设置对编程效率的重要性。在本次试验中，耗费大量时间在系统设置是导致该组志愿者全军覆没的主要原因。

同步协作—志愿者们表示，当在编码过程中出现问题，长时间的等待是计算机科学课程的又一主要问题。一名志愿者说，“每次我把求助信息发到教员或助教邮箱后，都要等很久才能得到回复”。另一名志愿者称，“由于时间紧迫，通常我更宁愿自己去想办法解决，或者向身边朋友求助。”

上述情况与我们的观察结果一致。试验中，试验对照组的教员穿梭于一个个同学之间，挨个帮他们解决问题，而还没有排上队的同学除了等待别无他法。

相反，试验组的同学可以直接向教员发起提问，教员也可以随时加入到他们的编码环境中帮他们解决问题。一位志愿者说，“我仅仅是在对话框中提出我的问题，他（教员）就马上

Jimbo 最受欢迎的功能是实时代码编辑，它支持多种协作编程模式。

进到我的编码界面，快速帮我解决了问题。这简直是太棒了”除了用在文字聊天功能中分享链接，代码片段或者寻求帮助，志愿者们在试验过程中也会进行口头交流，因此大家一致认为，编程工具中配备远程语音聊天功能也是必不可少的。

实时预览和用户提醒—试验组的志愿者们一致认为，实时预览功能帮助他们节约了编程时间。每当他们发现程序运行结果发生变化时，都能够立马知道是搭档修改了代码，而这一功能帮助结对编程者们更好地进行合作，同时也避免了代码冲突。一位志愿者说，“我能够快速地在预览窗口中看到被修改的代码。”另一位志愿者称，“我可以快速查看哪些代码正在运行，而且不会影响我的编码速度，真的很实用”

在试验对照组中，志愿者们只能通过编码器和浏览器的来回切换，来实时看到每一次代码修改后的运行结果。

在线设置

对于在线课程，我们选择了网页

设计和开发的网页开发课程。本门课程属于研究生课程范畴，但只要拥有计算机科学以外其他学科本科学历的学生都可以参加。课程来自于一个为期一年的学习项目，通过该项目的学生即可拿到计算机科学学位。参与课程的学生来自不同地方，但绝大多数是美国学生。

课程会对学生进行定期考评，教案包括在线 PPT、编码材料以及教员进行现场直播编码的视频剪辑，教学进度更合理，学生们主要通过在线聊天和邮件与助教进行联系。

我们邀请了 16 名（11 名男性和 5 名女性）拥有不同学科背景的学生参与试验，大部分学生编程经验十分有限，其中只有 3 名学生拥有一些计算机科学背景。这些学生都没有使用过实时协作工具，但意外的是，有十名参与者拥有结对编程经验。

试验及参与者设置—我们随机抽取了一半的学生作为试验组，在编码任务中使用 Jimbo 进行结对编程。剩下的学生作为试验对照组，可以自由选择工具进行编码。此次试验任务为

Jimbo 不会出现各种软件问题， 让使用者能够更专注于写代码， 大大提高编码速度和质量。

调用 NYTimes.com 网页的 RESTful 接口创建一个网络应用，实现通过输入关键字或时间范围搜索文章。在进行任务之前，教员通过在线直播编码指导学生学习 JavaScript 和 XML 的异步调用技术（AJAX），该技术用于在不需要刷新页面的情况下，通过实时与服务器交换数据更新网页内容。我们将试验组的学生随机两两分组，通过谷歌的视频沟通软件指导他们使用 Jimbo 进行远程结对编程。为了便于跟踪试验全过程，试验组学生的所有交流均在 Jimbo 上进行。

任务完成之后，我们对两组学生的程序进行了测试，并采访了他们的感受和看法。

结果—根据教员通常的评分标准，我们将志愿者提交的代码分为两个等级，其中 80% 的评分比重取决于是否能够完成任务参数及交付正确的程序运行结果，20% 取决于代码的可读性和结构。我们采用了两因素绝对相同（absolute agreement）组内相关（ICC）系数来确定评分机制的可信度。试验对照组的 ICC 指数是 0.873，试验组

为 0.965。试验对照组的评分中值为 81.75，试验对照组为 92.5，最终得分取两次评分平均值，两组志愿者代码的评分检验值分布差异显著（曼 - 惠特尼 U 检验值 = 11, n1 = n2 = 8, p < 0.05 双侧）。

和线下试验设置一样，我们分析了试验过程中观察到的现象和学生的观点。

系统零设置—几位试验组的同学表示，Jimbo 和他们通常使用的编码工具非常相似，如 Eclipse, Visual Studio 和 NotePad++，但他们更偏爱 Jimbo，因为 Jimbo 不需要进行任何初始设置。而另一些同学则认为，初始设置也并没有那么费劲，毕竟在第一次设置好之后就不需要进行再次设置了。

异步协作—当我们问及在线课程的缺点时，所有参与者都一致认为，在线课程最大的问题是当他们在进行编程任务时，无法及时得到教员的有效帮助。一位学生解释道，“如果没有助教的指导，我们无法知道如何运行从课程网页上下载下来的代码范例”，其他同学都表示对上述情况深有同感。

一名试验组的同学表示，他非常喜欢能够通过 Jimbo 进行连续性的学习，“当我通过 Jimbo 向教员发出求助信息后，他真的马上就进入到我的编码环境帮我解决问题了，简直是不可思议！”他说道。上述情况与我们对线上教学项目观察到的结论一致。

所有试验组的同学都使用语音聊天功能进行沟通，网站资源或程序代码等信息则通过文字聊天功能发送。他们认为，相比于文字聊天，语音聊天对团队协作和沟通最为直接有效。

所有的参与者都表示，在往常的线上课程中，无论是通过 Git 和 SVN 这样的版本控制软件，还是像 Dropbox 和谷歌硬盘这样的文件共享工具，都可以轻松完成任务协调和代码分享。

Jimbo 的实时代码编辑功能最受试验者欢迎。据我们观察，四个试验组中，只有一组同学严格按照结对编码的经典套路进行编码，即一个人写，一个人看，另外三组学生通过 Jimbo 的实时代码编辑功能都参与了代码编写工作。

实时预览和用户提醒—虽然大部分试验组的学生都对实时预览感兴趣，但他们发现，实时预览需要手动运行代码，这些操作会干扰他们编码。一位试验者解释说，“我当时正在 JavaScript 上修改代码，这时突然弹

出来一个代码运行结果提示。我很肯定自己没有修改那段代码，然后提示在显示一段时间之后才告诉我，我的搭档修改了 HTML 代码”。针对上述问题，一些试验者建议我们增加实时预览的暂停按钮。

评估意见

试验评估结果显示，在通过远程结对编程方法来学习编程新概念时，Jimbo 无疑是一个非常有效的学习工具。我们得到的使用反馈是积极的，基本上所有使用过 Jimbo 进行线上编码课程学习的学生都表示，今后他们还会使用 Jimbo 进行编程学习。更重要的是，试验结果证明，我们的功能设置能够帮助学生显著提升学习效率，特别针对在线课程。

结对编程模式

虽然 Jimbo 能够通过共享编码界面进行传统结对编程，但在试验过程中，教员并没有要求学生必须遵守传统结对编程的合作模式。因此，大部分学生选择较为自由的合作方式，所有学生均参与了代码编写，并通过系统中的文字和语音聊天功能进行沟通协作。

参与者希望能够进行实际编程操作来加深学习印象，因此他们更愿意在结对编程中同时承担船手和航海家

角色。无论如何，Jimbo 既可以支持传统结对编程模式，也可以支持学生自由选择编程模式。

语音聊天 VS 文字聊天

虽然 Jimbo 可以支持多种聊天模式，但试验中，学生们更倾向于使用语音聊天，只有当需要发送代码或链接时才会使用文字聊天功能。

协作方式

试验中学生们对 Jimbo 系统中异步协作和自动解决冲突功能青睐有加，他们表示，现有的资源控制系统功能设置混乱，使用起来十分困难。然而我们发现，虽然是结对编程，但程序员们仍需要有一个独立的编码环境，这样能够增强他们的成就感和自信。

实时预览功能造成干扰

虽然实时预览功能能够即时显示代码运行结果，但不断地进行运行结果提示会干扰用户编码，特别是当很多程序员同时协作完成一个项目时，干扰量会大大增加。解决以上问题很简单，只需要增加提示频率的设置，或增加实时预览暂停按钮。

通过以上试验，我们知道如同步通信功能、用户提醒功能，以及在线编程课程重使用到的变更管理等功能，不仅能够大幅提高学习效率，还有助

于进行远程结对编程和在线直播编码。志愿者们发现，由于 Jimbo 不会出现各种扰人的软件问题，而且能够快速获取教员指导，他们能够腾出更多的精力专注于代码编写，从而大大提高编码速度和质量。长期使用 Jimbo 这样的编程系统进行学习，学生们能够轻松应对各类实际的软件开发问题，同时也能够更好地进行团队协作。■

感谢

在此对所有的评论人、研究参与人员，俄勒冈州立大学人机交互研究小组的同事们，以及我的家人和朋友给予我们的帮助和支持，表示衷心的感谢。

参考文献

1. M. Goldman, “Role-Based Interfaces for Collaborative Software Development,” Proc. 24th ACM Symp. Adjunct User Interface Software and Technology (UIST 11 Adjunct), 2011, pp. 23–26.
2. L. Murphy et al., “Active and Cooperative Learning Techniques for the Computer Science Classroom,” J. Computing in Small Colleges, vol. 18, no. 2, 2002, pp. 92–94.
3. I. Vessey and A.P. Sravanapudi, “Case Tools as Collaborative Support Technologies,” Comm.

ACM, vol. 38, no. 1, 1995, pp. 83–95.

4. J.D. Herbsleb et al., “Distance, Dependencies, and Delay in a Global Collaboration,” Proc. ACM Conf. Computer-Supported Cooperative Work (CSCW 00), 2000, pp. 319–328.
5. L. Williams and R.R. Kessler, “Experimenting with Industry’s ‘Pair-Programming’ Model in the Computer Science Classroom,” J. Software Eng. Education, vol. 11, no. 1, 2001, pp. 7–20.

6. C. McDowell et al., “The Effects of Pair-Programming on Performance in an Introductory Programming Course,” ACM SIGCSE Bull., Feb. 2002; citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.6834&rep=rep1&type=pdf.

7. J.T. Biehl et al., “FASTDash: A Visual Dashboard for Fostering Awareness in Software Teams,” Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI 07), 2007, pp. 1313–1322.

8. A. Sarma, Z. Noroozi, and A. van der Hoek, “Palantír: Raising Awareness among Configuration Management
9. among Configuration Management

关于作者

苏鲁什·格瑞西是俄勒冈州立大学(OSU)电子工程与计算机科学专业博士，研究兴趣包括人机交互、用户体验、软件开发工具及开放资源工具。格瑞西拥有俄勒冈州立大学计算机科学硕士学位，同时是美国人机交互学会(ACM SIGCHI)会员。联系方式: ghorashi@oregonstate.edu.

卡洛斯·詹森是电子工程学院及俄勒冈州立大学(OSU)计算机科学专业副教授，俄勒冈州立大学应用系统及软件中心主任。研究兴趣包括人机交互和软件工程。詹森拥有佐治亚理工学院计算机科学博士学位，同时是美国人机交互学会(ACM SIGCHI)会员。联系方式: cjensen@eecs.oregonstate.edu.

- Workspaces,” Proc. IEEE Int'l Conf. Software Eng. (ICSE 03), 2003, pp. 444–454.
10. L. Hattori and M. Lanza, “Syde: A Tool for Collaborative Software Development,” Proc. IEEE Int'l Conf. Software Eng. (ICSE 10), 2010, pp. 235–238.
11. P. Dourish and V. Bellotti, “Awareness and Coordination in Shared Workspaces,” Proc. ACM Conf. Computer-Supported Cooperative Work (CSCW 92), 1992, pp. 107–114.
12. S. McDermid, “Living It Up with a Live Programming Language,” Proc. 22nd ACM SIGPLAN Conf. Object- Oriented Programming Systems and Applications (OOPSLA 07), 2007, pp. 623–638.
13. C. M. Hancock, Real-Time Programming and the Big Ideas of Computational Literacy, PhD thesis, Dept. of Architecture, Media Arts and Sciences Program, MIT, 2003.
14. S. Ghorashi and C. Jensen, “Jimbo: A Collaborative IDE with Live Preview,” Proc. 9th Int'l Workshop Cooperative and Human Aspects of Software Eng. (CHASE 16), 2016, pp. 104–107.



搜索你的工作机会

IEEE Computer Society 招聘可以帮你轻松找到IT、软件开发、计算机工程、研发、编程、架构、云计算、咨询、数据库很多其他计算机相关领域的的新工作。

新功能：找出那些建议或要求拥有IEEE CS CSDA或CSDP认证的工作！

点击www.computer.org/jobs，
从全世界的雇主那里搜索技术工作岗位和实习机会。

<http://www.computer.org/jobs>

IEEE  computer society | JOBS

IEEE计算机协会是AIP Career Network的合作伙伴。其他合作伙伴包括《今日物理》杂志（Physics Today），美国医学物理协会（American Association of Physicists in Medicine），美国物理教师协会（American Association of Physics Teachers），美国物理学会（American Physical Society），AVS科学和技术学会（AVS Science and Technology），物理学生协会（Society of Physics Students）和Sigma Pi Sigma。



设计学创新： 跨学科教育

文 | 石田亨（Toru Ishida）、榎木哲夫（Tetsuo Sawaragi）、中小路久美代（Kumiyo Nakakoji）、十河卓司（Takushi Sogo），京都大学（Kyoto University）

译 | 郑起

一个设计学教学项目必须将传统上相互独立的学科联合起来，把工程学严格学术边界扩展至社会学领域。这是我们在创建京都大学设计学院时所面临的挑战。我们致力于推动跨学科合作，希望为其他人打造新型设计学提供一个扎实可靠的基础模型。

世

界各地的科研人士都倍受压力，要解决一系列

严峻的挑战，比如全球变暖、减少灾害、能源和食物稀缺等。这些问题超过了任何单一学科的能力所及，也就是说，必须由来自不同领域的专家合作，提供解决问题的各种不同视角。比如，在判断一次撞击事故为何会发生时，一名机械工程师可能会猜测问题出在引擎上，一名计算机科学家会推断控制系统是主要问题，一名建筑师可能会将目光投向糟糕的城市设计，一名心理学家可能会猜想事发时司机心不在焉，一位管理学专家可能会推断司机因加班而过劳。就这样，为防止类似事故再次发生，会有各式各样策略来应对眼前的问题。这种多学科合作也能确保我们一直在解决问题的道路上推进：假如一个难题让某个领域里的专家愁眉不展，其他领域的专家可以继续推动解决方案的演

变。

但跨学科合作并非一个人们熟知的解决问题模式。在上述简单的交通事故案例中，来自各方的贡献很可能会合理地汇聚起来达成某种解决方案，兼顾其中涉及的各类担忧：汽车、司机、城市规划、工作条件，等等。但在挑战整个社会所面临的问题时，专家们对于应当关注什么的看法大相径庭。这些各异的观点和解读成为了跨学科合作的严重阻碍。因此，任何处理重大挑战的项目必须不仅仅诉诸于特定领域的专业知识，还必须审视其他领域看待同一个问题的视角，以拓展对问题的理解。设计学的教育和研究就是一个实现跨学科研究的新兴机制。

在过去十年左右的时间里，大学院校已经创立了许多专门化的设计专业，比如机械设计、系统设计、环境设计和建筑

设计等。京都大学试图将这些专业合并成一个设计学院，把特定领域设计教学和非特定领域设计连接起来。我们基于对设计不断演变的特性的了解，继续完善这个学院，打造设计教学的通用语汇和课程模式。全球各地已经涌现了许多类似的尝试，我们希望设计学教研项目的创办者能公开分享各自的经验。

什么是设计？

在我们把设计学院打造成一个跨学科研究中心之前，必须首先厘清我们所说的“设计”是何意。我们认为下述定义尤其有用：“明确设计的对象、由一个主体来表达、意图达成目标、处于特定环境中、运用一套基本组件、满足一系列需求、受到一些限制”。¹

对定义的诠释

这个定义非常抽象，每个学科都可以轻易代入自己的解读。比如，计算机科学家很可能会把这个定义和优化问题联系起来。假如设计的目标、限制、要求能以特定的形式被明确化，那么优化算法和高效能运算确实可以被用于寻找解决方案。一旦某个问题可被形式化，这种解读就能奏效。但是，“设计”的焦点已经在向远为广泛的、由机构和社区构成的社会体系及结构转移。在这一大背景下的议题不能被处理成简单

的优化问题。比如，建筑师往往会以更社会化的方式来诠释这个定义。城市设计项目可能需要召集利益相关方开会来确定社区的所有要求，这些要求体现了许多不同层面的社会关切。如此，这个项目的边界就变得模糊不清。一旦一个问题突破了形式化的清晰边界，什么是对它的最佳计算理论和方法这一点不再显而易见。

我们以日本在2011年东日本大地震后改造城市和社区设计的努力为例。那场灾害发生后，一个需求凸显出来：要武装社区以保护它们免受未来自然灾害的打击。解决问题的过程（即“如何设计”）不可避免地涉及到明确问题是什么的过程（即“设计什么”）。由于社会可以在各种层面被抽象化，表达问题的模型可能存在许多层面。结果，解决问题的过程可能是在模型的不同层面上下游走。比如，一个改善城市交通网络的项目会带来对现有巴士路线的改变，这就带来了一个额外的挑战——取得居民的同意。这又继而要求他们对于城市的设计抱有相同的展望。从这个意义上说，设计不仅仅关乎在此时此地解决问题，也要回答如何为未来而改造社会的问题。

从制造到培育

日本科学理事会2003年的人工制品设计和生产提案（这里的“制品”是

个抽象概念）指出了这样一种需要：将设计过程从制造产品扩展为包含理解并培育关系及环境的项目。²这种转变在机械工程和建筑学中尤为突出，其研究重点从技术本身转移到了设计用户体验上。与此同时，在计算机科学中，焦点已经不再是执行复杂的逻辑，而是如何把软硬件元素运用到社交功能中。³

设计学的进化

出乎意料的是，自上世纪六十年代起设计就已开始演变。1969年，经济学、决策和人工智能的创新思想家司马贺（Herbert A. Simon）写道：

（人类）行为的复杂性可能大多源自他们所处的环境以及对优秀设计的追求。假如我的阐述已经足够清楚，那么我们可以得出这样的结论：在很大程度上，对人类的研究是一门设计科学，它不仅是技术教育的专业组成，也是所有文科生的核心学科。⁴

仅仅一年后，京都大学的梅棹忠夫在他的“信息工业社会中的设计者”演讲中分享了这样的观点：

多年来，开发产品和材料的方法已经发生天翻地覆的变化。而如今，我们已经能够获取我们所需的所有能量。在这一大背景下，我们最大的挑战是设计问题，或者说，我们应该如何把这些开发方式及能源合并起来。看起来，在这

人类行为的复杂性可能大多源于自身所处的环境以及对优秀设计的追求。

个信息工业时代里的设计者扮演了或者将会扮演所有工业核心元素的连接者。⁵

两人都早早地洞察到信息工业和设计工业将融合——在大学院校刚刚开始创立大型计算中心的年代，这实属不易。随着时间的流逝，作为一个解决社会问题的产业，设计开始因强大的全球性约束力而受阻，这包括技术、文化、经济和政治上的限制。这些限制又生成了相互牵制，从而进一步演变成错综复杂的问题网络。

例如，停用核能可能会增加化石燃料的消耗，而增产生物乙醇很可能导致粮价上涨。鉴于这些联动式的约束力，设计社会体系和结构的任务已不再是一个简单的优化问题，而变成了应对复杂网络的控制问题。在这个新环境里，设计学具有独特的优势来将最新的现代科学技术理论应用于实践。

现有设计院校

设计学院的跨学科性质使得我们

很难将它们划归到一般大学分门别类的组织结构中。这不仅仅因为设计学院的概念仍属新鲜事物，也因为每个学科都已经建立了自己的设计理论和方法。由此，一个设计学院必须是一个跨学科机构，同时又完全扎根于许多不同的学科之中。要实现这种设想有多种方法。但通常，设计学院是从一些专门的学科（计算机科学、建筑学、机械工程学等）中发展起来，而后再朝着一个跨学科教学研究模式成长。

斯坦福大学

斯坦福大学在2004年成立了哈素·普拉特纳设计学院，简称设计学院（d.school）。该学院从机械工程学和计算机科学入手，设有各种各样强调设计实践的研习班和课程。虽然该学院既非研究生院，也非一个科系，没有特定的学生团体，也不提供学位，但仍和斯坦福已有的成熟学科相连接。比如，机械工程学的老师会在设计学院授课，而想要报读设计学院课程的学生从自己的系里注册。设计学院知名的教师也在

IDEO设计公司里工作（尽管它和斯坦福相互独立），这家公司致力于把设计思想转变成商业项目，日益吸引眼球。⁶

哈佛大学

哈佛大学的设计学研究生院建成于1936年，将建筑学、城市规划和园林建筑学结合起来。相比近年创办的设计学院，它的范围更狭窄。哈佛的设计工程协作硕士学位可能更为接近当今的设计理念。这个学位由设计学研究生院和约翰·鲍尔森工程及应用科学学院于2016年联合设立，追求跨学科教研，旨在让设计成为技术、社会和环境的连接者。

代尔夫特理工大学

自1969年创立以来，荷兰代尔夫特理工大学的工业设计工程学院（Faculty of Industrial Design Engineering）一直是一个非常活跃的跨学科教研中心。它由来自心理学、机械工程学和计算机科学的教师引领，致力于在特定领域设计和非特定领域设计之间取得平衡，并建立一个结合社会需求、技术可行性和商业可行性的设计模型。

阿尔托大学

芬兰的阿尔托大学本身就是一个跨学科教研中心，由三所大学（分别侧重工程、艺术和经济学）于2010年合并

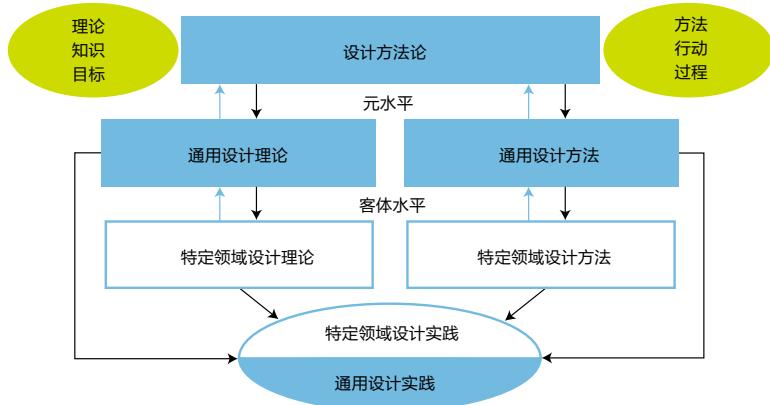


图1.设计学院课程框架。传统的特定领域课程包括专注于制造产品的设计理论、方法及实践（白色部分）。它们最终必须和非特定领域的理论、方法及实践（蓝色部分）合并。后者更关注应用在社会元素中的设计，以及将设计理解为一种概念。

而成。阿尔托的设计项目针对医疗、老龄化、全球变暖等具重大挑战的领域。该校创始人深信这类挑战无法由学校单独攻克，所以非常强调和产业界的合作。

课程范围界定

鉴于其跨学科的性质，一所设计学院不能依赖传统的课程模式。如图一所示，特定领域设计理论、方法和实践必须与非特定领域设计理论、方法和实践相结合。比如，在计算机科学中，特定领域设计理论会包含计算理论；方法课程会包含软件工程；实践会包含编程作业。而非特定领域设计课程将关注人工制品、信息、组织与社区等社会元素的设计，也会关注设计方法论——研究设计的概念、理念和历史的课程。从这个意义上讲，“非特定领域”实际上有两种含义：普遍和精髓。

即使图1这样的架构可被用于将课程系统化，许多研究生仍会缺乏动力在研读本专业的学习设计。他们可能会这么想：我明白社会问题是复杂的，但为何不能将相关学科的专家集结起来，组成项目团队来解决这些问题？我不如学习项目管理，而不是设计，不是吗？

答案是否定的。有两点原因：首先，对社会有重大影响的研究通常都产

生于不同学科的连接和交叉处。计算机科学的创新通常发生在与其他学科的交界。例如，谷歌的广告位拍卖就是在计算机科学和经济学交界上的一项发明。学习设计能让学生获得知识和技能来拓宽其专业研究的视野。

其次，研究本专业以外的领域不仅是学习科研成果，还要求理解获得这些成果所依赖的方法论和过程。假如计算机科学家要和心理学家共事，则双方都需要了解对方使用的研究过程。大学能为这类跨学科学习提供理想的环境。

一种能把不同学科的人们联合起来的通用语。如图2所示，该学位是基于四个研究生院的五个学科：信息学、机械工程学、建筑学、管理学和心理学。信息学、机械工程学和建筑学已被公认为设计学院的核心部分，而我们又加入了管理学和心理学以在技术和社会间建立联系。当然，设计的应用并不只限于这五个领域，而是要对付环境、医疗、灾害管理等领域里涉及面远为广泛的挑战。京都大学设计学院的标志是三个加号，凸显出其目标是要培养出“十字型”人才——有各自的研究领域，但可以通过设计与其他学科连接的人才。

京都大学设计学院

2013年4月，京都大学推出了设计协作研究生课程——更为人熟知的一个称法是京都大学设计学院。这是一个五年制博士学位，目标是培育出能与不同领域的专家协作设计出社会体系和结构的优秀人才。

这个学位旨在将学生培养成有能力改造社会的专家。设计学让他们获得

课堂

整个课程包括上课和做博士研究。学习的前半段包含通用设计课程和主修的特定领域设计课程，后半段包括辅修的特定领域设计课程。

通用设计课程包括：

- »设计方法论，探讨什么是设计；
- »通用设计理论，对跨领域设计的基础展开探讨，包括对人工制品、信息、

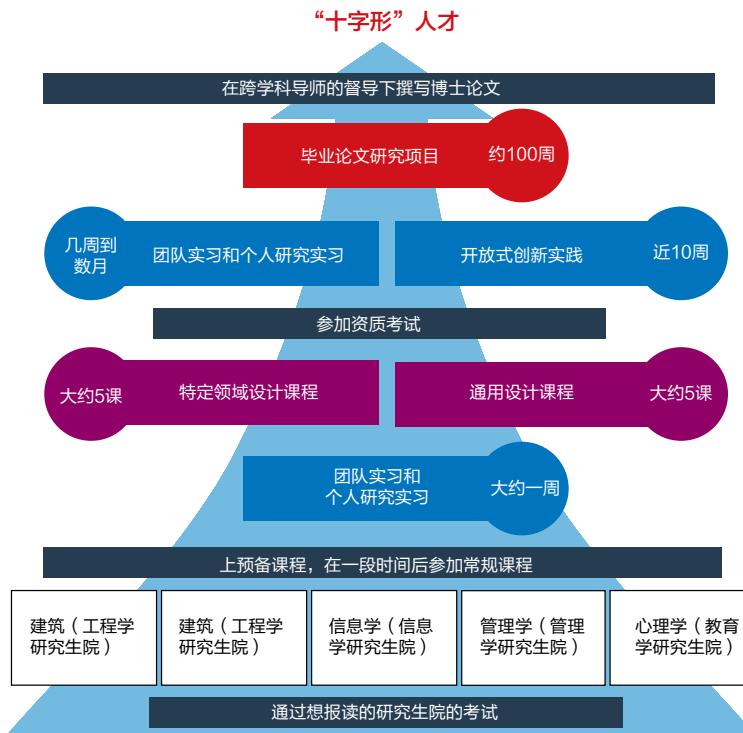


图2. 京都大学的设计学研究生院合作项目。五大学科组成了该项目的核心，但学科的交叉和扩展也是它的一部分。学生们将设计课程（紫色）和设计实践（蓝色）融汇到他们主修的专业中。发现问题式学习（FBL）和解决问题式学习（PBL）增强了课堂学习和做项目的效果。学生们把设计用作一种通用语言来连接不同学科的研究人员，由此达到阶段性的学术里程碑（黑色），一步步向着能够共同工作来改变世界的“十字形”专家靠近。

组织和社区的设计；

→通用设计方法，涵盖区域分析工具，如人种学、数据分析、模拟，以及构成设计和设计过程的方法。

前半程学习还包括两类课程：发现问题式学习（field based learning）和解决问题式学习（problem-based learning）。前者培养识别现实世界问题的技能，后者寻找解决现实问题的方法。大部分课题是从教师挑选的实际问题发展而来，比如如何扩大使用可再生能源、如何让一个收缩中的乡村地区重新焕发生机，等等。这类挑战性议题让学生对真实存在的问题拥有一个全面的认知。

在后半程学习中，学生们会和来自

其他领域的专家一起全面挑战现实议

题。这种开放式的创新实践将处理商业部门存在的真实问题，将学生从准专业

人士发展为能够管理专家团队的协调员。最具挑战的学习体验是实习，跨学

科学生团队会到日本各地或其他国家，

帮助解决当地的现实问题。

随着课程推进，实际运用中遭遇的挑战会逐步增加。被派往印度尼西亚实习的三年级学生很自信能够胜任工作，这是因为他们已经从难度系数不断增加的课程实践中获得了经验。

第五年，学生们完成学业，在跨学科导师的指导下，在他们自己的系里撰写论文。

广泛的设计合作

除常规课程以外，这个博士生专业也提供了在不同环境和文化背景中学习的经历。2011年东日本地震刚刚发生后，京都大学推出夏季设计学院（见图3，当时京都大学设计学院尚未成立）。这个为期三天的项目让学生们有机会和来自其他大学、政府及产业界的参与者合作。2016年，夏季设计学院举办了近30个工作坊，包括学生和教师在内共有350多人参加。

这个夏季项目的主要益处之一，是这些研究活动不同于传统的“教师-学生”两分法。它的教师和学生比接近一比一。这种模式促进了交互学习体验——参与者合作确定要解决的问题。产业界提供了三分之一的研究课题，为学生的学习过程增加了产业-学术交叉的维度。

设计创新集团

大学院校认识到需要和企业及政府机构合作，因此已经为合作研究项目建立起各种架构。当参与合作的所有人都理解和认同一个项目需要什么的时候，这类架构是能够成功运作的。然后，大部分的项目都至少需要一些初步的探索，这就需要参与项目的专家开展研讨。这类研讨有时会发生矛盾，但学校为确保教学的可持续性必须寻求这样的合作项目。它们可能涉及企业、政



图3. 夏季设计学院合作项目。这个为期三天的活动由京都大学设计学院主办，让学生们有机会和来自学校和企业的专家开展设计合作。

府机构以及其他大学和研究机构。对于这样的项目，现有的合作研究框架就不够用了。

参与各方往往带有不同的目标目的，喜欢不同的操作方式，但为了开展有效合作，达成共识是必须的。与其让大家为每一个项目都展开谈判，京都大学决定设立一个机构，让它在一个全面合作的协议下运作。于2014年3月创建的设计创新集团（Design Innovation Consortium）旨在共同寻找发展研究生和企业员工技能的最有效方式（重点放在研究生上）。该机构在2017年3月已拥有近60名企业成员。这个中心分享在大学院校中发展的设计理论和方法，也分享产业界和政府遭遇的现实难题。

成员企业提供专家来和师生共同开展设计项目，建立起长期的合作关系。这些专家都是很有兴趣和动力参与设计活动的人。他们会担任学生们的导师，建议他们如何将大学里的研究成果转化为能够服务社会的形式。学生们得以接触到这些专家广博的知识和经验，这有助于他们更深入地认识到自己面临多样化的职业道路。

设计学院评估

虽然京都大学设计学院第一届学生尚未升至五年级，它已经给参与各方

都留下了深刻的影响。为衡量对本学院的反馈，我们调查了所有三年级学生，并基于该调研发展出一个设计创新者指数。

学生问卷调查

我们向11名学生发放了问卷，它涵盖七个主题：和其他领域人士的合作、推动项目和会议、全球协作、有效沟通、有效展示、设计知识和方法应用、研究重点扩大。

在11名接受问卷的学生中，有7人表示自己发展了与其他领域人士合作的技能，6人认为自己已经获得了推动项目发展的能力，5人认为自己学到了如何开展全球化合作、如何有效沟通，并发展了对设计和设计方法应用的认知。7人认为自己仍然需要学习有效展示和拓宽自己的研究焦点，但他们都有信心将在五年学习结束前获得这两类技能。

在返回的问卷中，这类表达很典型：“我了解了推动合作和协调不同观点有多困难”；“学习以开放开明的态度来逐步扩大讨论，这让我获得信心，能在和不同文化背景、不同专长的人的

合作中取得有意义的成果。”图4的“旅行地图”展示了两名学生为他们的博士研究寻找论题的过程。他们从对相对而言狭窄的研究领域的兴趣出发，自由运用课程中提供的跨学科机会，找到了那些在他们自己的专业中很具吸引力同时也和社会问题相关联的研究课题。

设计创新者指数

设计创新者指数评估京都大学设计学院的教学质量。它从识别优秀的设计创新者拥有的技能组合的经验数据中发展而来。图5展示了这一指数以及技能组合；这些技能覆盖四个领域，共24个目标描述。这四个领域最初是由英国的“研究员发展框架”（Researcher Development Framework）划定的，被视为发展可转化技能所需的核心素质。

我们使用这套指数，在学生们刚参加课程、第三学年参加资质考试，以及博士学业完成时分别展开评估。图5中的比例数字基于对15名学生被录取时的评估和13名学生在之后的学习中的评估。通过这些评估，学生们清楚地理

封面文章：学习技术的发展

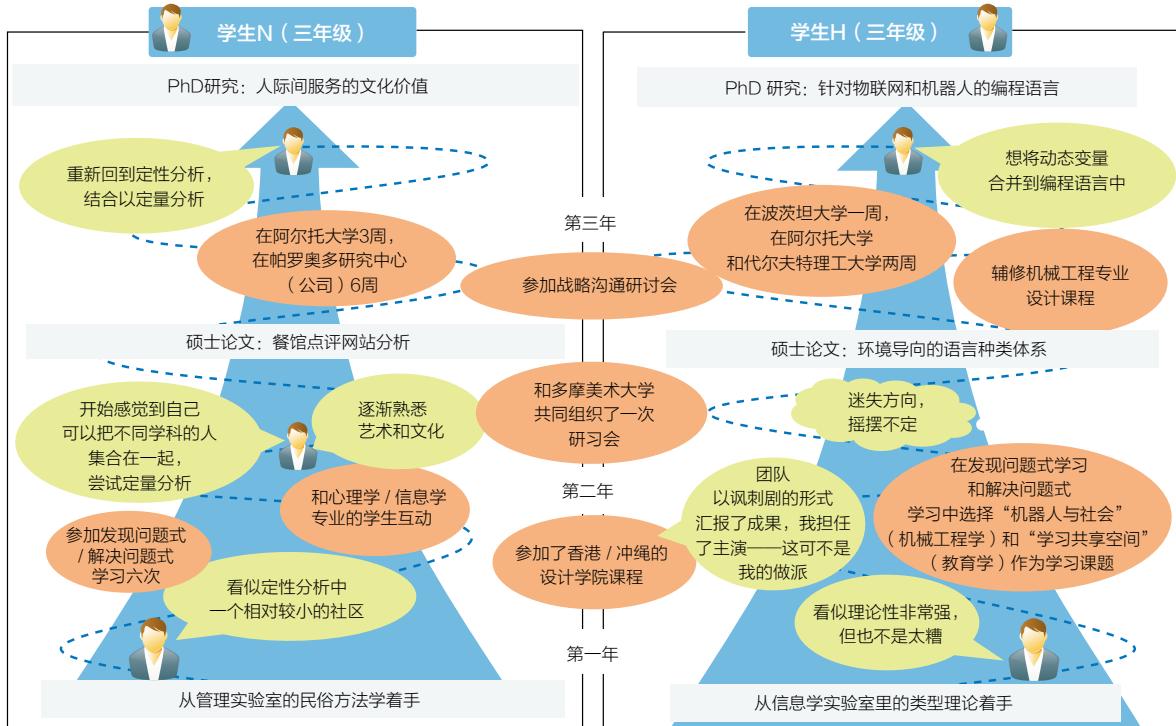


图 4. 京都大学设计学院的两名学生决定博士论文题目的过程。绿色圈内罗列了他们在前三个学年里的评论和想法。橘色圈里列明了他们选择的学习机会，灰色条则代表他们达到的研究和学术里程碑。这两名学生最终选择的论文题目都非常适合自己所在的专业，同时又与社会议题关联。

解了他们将从课程和合作项目中学到的主要知识、所需的行动和品质，以及一大批来自不同行业的潜在雇主的需求。在图5中，圆盘中的紫色线条追踪了企业和政府雇主中强烈或比较期望学生们将在学习过程中获得某种素质的人数在总人数中的占比。

而在学生中，这一比例不断上升，显示他们对于自己的才能越来越有信心，最终超越了雇主的期望。

京都大学设计学院不断演变，但把设计学打造成一个学术领域需要三大关键行动。首先，要将特定领域的设计理论和方法抽象化，并过渡到其他领域。第二，为抽象的设计方法论清晰阐述目标，这样特定领域的设计理论和方法可被实例化。

最后，必须细致研究设计学的基础概念，创造出一套跨学科语汇。单凭一所大学是无法成功开展这些行动的，需 要世界各地的设计学教职员的大量反馈和积极参与。

人们常认为科学和工程学大相径庭，因为在他们的认识里，科学是用来理解现象背后的本质，而工程是为创造能造福社会的技术。然而，工程研究人员所做的不止于此，他们应被视为一个科学与工程连接体系的创造者。同时，设计一直以来都被视为工程的一个元素，它带来了服务于社会的技术的发展。然而在今天这个错综复杂的问题网络里，设计不应该再被仅仅视为工程学的应用阶段。科研人员应当承担起确立设计学范畴的学术挑战，创造出一个科

学、工程与设计联动的体系。■

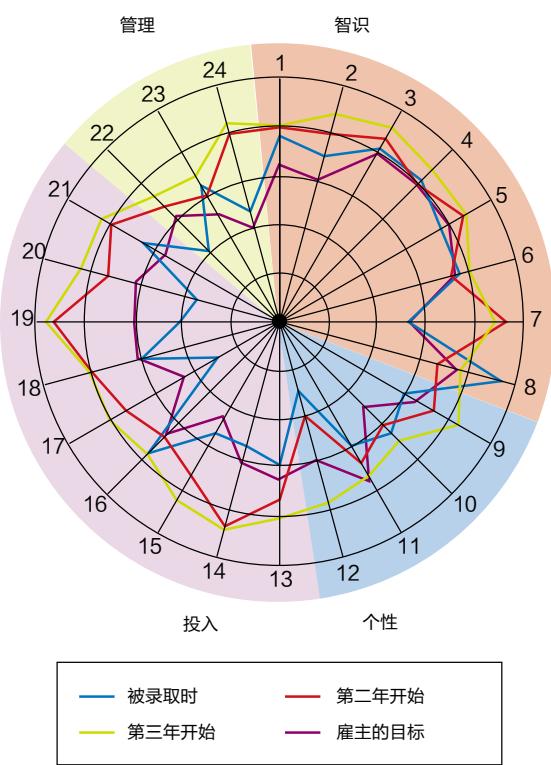
致谢

我们感谢京都大学设计学院的学生和教职员对本文的参与。尤其要感谢荣誉教授门内辉行分享他的知识及提出宝贵建议。本学院获得日本文部科学省顶尖研究生院项目的支持。

参考文献

- P. Ralph and Y. Wand, “A Proposal for a Formal Definition of the Design Concept,” *Design Requirements Engineering: A Ten-Year Perspective*, K. Lyytinen et al., eds., vol. 14, Springer, 2009, pp. 103–136.

才能目标



- 1 在更广泛的社会环境中认识到自己所在专业领域及相关领域的
重要性
- 2 评估并收集恰当来源的观察所得数据
- 3 识别要处理的基本研究问题并确认有效性
- 4 拥有核心知识以及对主要概念的基本理解
- 5 评估自己研究活动（和现实世界事务相关）的结果和影响力
- 6 通过在自己的研究领域里调查和寻找信息来产生创意和方法
- 7 批判性地分析和评估现有产品及社会体系
- 8 对研究将如何影响社会、社会又将如何影响研究的认知
- 9 展现出自律和开展研究的积极性
- 10 认识到自己知识水平的局限以及需要获得支持
- 11 发展国际化的人际网络以接触和理解其他文化
- 12 在为自己的观点辩护时展现专业化的正直与诚实
- 13 建立合作关系，和各类同事分享自己的见解
- 14 积极参与合作和外部关系并贡献自己的力量
- 15 能向各种不同的、非专业人士的受众清楚有效地表达自己的
观点和思想
- 16 能够认识到跨文化环境中的多元化和差异性并在这样的环境里
工作
- 17 能够在团队活动中领导他人时展现出积极性和才能
- 18 理解团队环境中的领导行为，能有效达成互惠目标
- 19 鼓舞团队成员
- 20 引导团队成员，帮助他们澄清自己的角色和职责
- 21 和来自公共或商业部门不同领域的团队成员积极交流知识
- 22 在更广泛的背景下发展对自己的研究的理解，描绘出长期规划
- 23 运用有效的项目管理
- 24 通过撰写研究提案和管理自己的研究资金来掌握融资过程

图 5. 设计创新者指数。该指数被用来衡量学生在四个维度里的发展：智识、个性、投入、管理。这四个领域共列出 24 种目标。学生们在刚被录取时以及第二、第三学年的开始都被要求评估自己对是否已达到目标的自信程度。轮盘图显示了“非常自信”和“比较自信”的学生在学生总人数中的占比。到第三年，大部分学生的自信程度已经超过未来雇主们的预期。

2. A Proposal of Design Vision for Artifact Design and Production in the 21st Century, Science Council of Japan, 15 July 2003.
3. T. Winograd, Bringing Design to Software, ACM Press, 1996.
4. H.A. Simon, Sciences of the Artificial, 3rd ed., MIT Press, 1996.
5. T. Umesao, "Designers in Information Industry Society," Anthology of Works by Tadao Umesao, C. Koronsha, ed., vol. 14, Information and Civilization, 1970.
6. B.M. Katz, Make It New: The History of Silicon Valley Design, MIT Press, 2015.
7. The Vitae Researcher Development Framework, Careers Research and Advisory Centre (CRAC) Ltd., 2011.



用以提升教学效果的情感分析系统

文 | 苏耶亚塔·兰尼 (Sujata Rani) , 帕蒂克·库马尔 (Parteek Kumar) , 塔帕尔大学
译 | 黄美桃

自然语言处理和机器学习可应用于学生反馈，以帮助大学管理人员和教师明确教学过程中的问题。本文所提及系统从课程调查和线上资源两方面对学生评价进行分析，以确定情感极性、情绪表达和满意度及不满度之间的对比。与直接评估所得结果的对比显示该系统可靠。

自然语言处理和机器学习可应用于学生反馈，以帮助大学管理人员和教师明确教学过程中的问题。本文所提及系统从课程调查和线上资源两方面对学生评价进行分析，以确定情感极性、情绪表达和满意度及不满度之间的对比。与直接评估所得结果的对比显示该系统可靠。

情感分析是基于用户对同一文本的不同情感进行识别和分类，比如积极的、消极的、或中立的，又比如是诸如快乐、悲伤或厌恶等的情绪，这决定用户对特定主体的态度。情感分析在许多领域发挥着重要作用，比如在教育领域，学生反馈是对学习技术效果的重要评估途径。

许多大学在课程之中或结束之前通过学生反应系统获得上述反馈以分析教师的表现。¹学生对于教师表现、学习经历以及其他课程特性之反馈，亦可通过社交媒体获得。近年

来，在线学习门户网站，如“课程时代” (www.coursera.org) 通过免费提供经过挑选的院校课程吸引了许多学生，相关的院校数目仍在不断增长。²每年都有成千上万的学生参加这些网上公开课，并在课程论坛分享他们对课程内容和教学质量的意见。学生也会在博客、网上论坛，如“高校机密” (www.collegeconfidential.com) 评价他们的学习体验，而教师则会浏览如“评价我的老师” (www.ratemyprofessors.com)³ 等网站。通过这些反馈，大学管理人员和教师不仅仅能获得有价值的见解，同时这些反馈同样对学生决定去哪所大学或参加哪些课程有重要意义。⁴

情感分析

课程效果可直接或者间接进行评估。直接评估考量的样

品为真实的学生作业，包括：测试、作业、小测验、项目报告。间接评估考量则基于学生对学习的体验和对教学质量的观察。学生反馈的情感分析是间接评估的一种方式，其通过分析学生写下的评语，不管是正式的课程调查或来自于在线平台的非正式评价，以确定学生对课堂的兴趣及确定可以采取相应行动改善的方面。

情感分析提出了许多技术挑战。其一，字词在不同的领域有不同的含义。例如，在教学背景下，“早”在“该课程上太早了！”里有负面的含义，但在消费者的语境下，在句子“快递来得早。”里有正面的含义。其二，在多种语言的情况下使用情感分析系统可能会有困难。比如在印度，人们经常直译印地语来表达他们的意见；因此，他们可能使用

वो क्लास में अच्छा पढ़ते हैं।

这在英语里的意思是“他课上得非常好”。诸如此类的挑战促使科研人员需要开发具有语境敏感度、多语种的情感分析系统。大多数情感分析研究关注用户评价，如产品、电影和酒店评价，这类研究普遍把评价分为积极、消极或是有时还有中立选项。情感分析仍未大规模应用于教育，虽然根据“相关研究”的侧边栏报道，近来这类研究在不断增加。然而，大部分此类研究仍局限于把情感分为上述的两到三类，并无考虑到还有许多情绪因素亦可影响学生

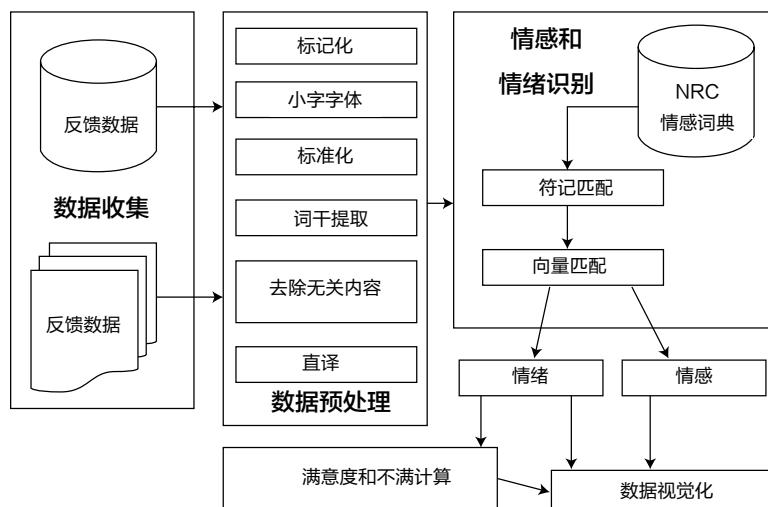


图1.本文所提情感分析系统的框架。在预处理输入数据之后，学生反馈的获得途径分为正式来源，如课程调查，和非正式来源，如博客和论坛，该系统使用自然语言处理，并结合《NRC 情感词典》，对情绪和情感进行分类。情感被分为两类：积极和消极；情绪则通过系统计算满意或不满分为八个类别：愤怒、期待、厌恶、恐惧、愉快、悲伤、惊讶及信任。该情感分析系统能够处理多语言内容，并包含了数据视觉化功能以协助分析。

的反馈。此外，这些研究并无处理多语种数据。最后，此前的研究人员没有尝试通过将他们分析的结果与传统的直接评估方法得出的结果进行比较，以验证该系统。

所提议的情感分析系统

我们所提议的情感分析系统帮助我们通过对多语种的学生反馈实行短暂的情感和情绪分析，主要关注教师表现和对课程的满意度，以帮助提升教学质量。该系统把情感分为两类——积极和消极，并根据Robert Plutchik的情绪分类把情绪分为八类——愤怒、期待、

厌恶、恐惧、快乐、悲伤、惊讶和信任，而且据此计算出满意或不满。

图1展示了系统的架构。该系统有五个主要组成部分：数据收集、数据预处理、情感和情绪区分、满意或不满计算以及数据视觉化。该系统运用了开放资源“R语言”(www.r-project.org)实行数据预处理和情感分类。

数据收集

我们的原始数据库包括学生对于“课程时代”课程的反馈以及一所大学的情感分析研究数据。“课程时代”的数据库包含了将近4,000名学生在课程中的评价，以及1,700学生在完成课程后的评价，时间跨度是从2015年8月至

相关研究

近年来，研究人员开始在教学领域应用情感分析，使用多种机器学习和自然语言处理技术。

2011年，Zied Kechaou, Mohamed Ben Ammar和Adel M. Alimi运用隐马尔科夫模型和支持向量机组合的混合技术对在线学习的博客和论坛进行情感分类。他们的实验使用三种具有不同特征的挑选方法——交互信息、信息获取和卡方检验，并确定卡方检验法优于其他两种方法。¹

两年后，Myriam Munezero和她的同事对学生的学习日志进行情绪分析，并根据Robert Plutchik的八大类情绪进行分类。他们同时还从这八大类情绪中计算出挫败和焦虑。²

2014年，Nabeela Altrabsheh, Mihaela Cocea和Sanaz Fallahkhair使用以一元模型为基础的朴素贝叶斯方法、朴素贝叶斯方法的补充、支持向量机和最大熵分类器，对学生反馈进行情感分析。他们总结认为对于实时反馈分析而言，支持向量机协同径向基核函数以及朴素贝叶斯方法的补充能达到良好的效果。他们同时观察到除去中立选项的研究会有更好的效果。³

第二年，Trisha Patel, Jaimin Undavia和Atul Patel使用文本工程的通用结构工具分析来自学生家长会的反馈，并使用ANNIE将评价分为积极或消极。⁴

2016年出版了许多研究。Francis F. Balahadia, Ma.

Corazon G. Fernando和Irish C. Juanatas提出使用情感分析系统让学生用英语和菲律宾语评价教师在课程中的表现。他们通过朴素贝叶斯算法并用图表形式呈现学生的积极及消极情感所占百分比，从定性和定量反应两方面进行评级，以帮助大学管理人员认识到学生所关注的方面。⁵

V. Dhanalakshmi, Dhivya Bino和A.M. Saravanan基于阿曼王国中东学院学生评价调查的反馈，进行情感分析。他们使用RapidMiner工具，从教师、考试、模块内容及资源四方面将评价分为积极和消极两大类。研究人员分别用朴素贝叶斯方法、支持向量机、最邻近规则算法和神经网络分类器进行研究，并对研究结果进行比较。⁶

Brojo Kishore Mishra和Abhaya Kumar Sahoo运用带有图形处理器的统一计算设备架构（CUDA）C语言编程评估全体教员的表现。他们根据学生在反馈表格上给出的平均分数把教员分为以下几类：卓越、很好、好、一般或差。研究人员顺利地对比了此研究的执行时间与运用中央处理器结构进行相似评估的执行时间。⁷

Guadalupe Gutiérrez Esparza和她的同事提出对学生在“推特”上用西班牙语对教师表现的评价进行情感分析。他们使用支持向量机算法将“推特”内容分为积极、消极和中立。他们也

2016年8月。学生反应系统数据库包括了过去十年间约500名学生在期中或期末考试后对于同一位教师教授的同一门课程的评价，从授课及实验两个维度进行评估。数据还包括不同教师在过去两年间教授的25门课程的学生调查和评价，对此，我们结合了对学生表现的直接评估，以验证该系统的可靠性。

数据预处理

在这个阶段，情感分析系统预备已收集的数据进行进一步的处理。这包括

了六个步骤。

标记化。把学生的评价分为一个一个的词语，或符号，在“R语言”中运用标记字符串功能。

小写字体。特征转化成小写字体以简化在将学生的评价与《NRC情感字典》⁸词语匹配的过程。该过程的实施运用了“R语言”的tm程序包里的tm_map功能。

标准化。通过使用字典查找常用的网络俚语缩写以标准化其内容。如：“gud”和“awsm”分别代表的是“good”和“awesome”。

词干提取。为了进一步促进词语匹配，学生评语中的词语会运用“R语言”的SnowballC程序包中的tm_map功能转化为他们的词根。如：“移动的”、“被移动的”和“移动”统统转化为“移动”。

去除无关内容。去除跟情感分析没有关系的标点符号和停止词，以提升系统反应的时间和效率。

直译。为解决学生评价中多种语言的混用问题，文本会直接使用谷歌翻译进行直译。

提出使用句法模式模型，并比较分别用支持向量机算法和句法模式得出的结果。⁸

参考文献

1. Z. Kechaou, M.B. Ammar, and A.M. Alimi, "Improving E-learning with Sentiment Analysis of Users' Opinions," Proc. IEEE Global Eng. Education Conf. (EDUCON 11), 2011, pp. 1032–1038.
2. M. Munezero et al., "Exploiting Sentiment Analysis to Track Emotions in Students' Learning Diaries," Proc. 13th Koli Calling Int'l Conf. Computing Education Research, 2013, pp. 145–152.
3. N. Altrabsheh, M. Cocea, and S. Fallahkhair, "Learning Sentiment from Students' Feedback for Real-time Interventions in Classrooms," Adaptive and Intelligent Systems, A. Bouchachia, ed., LNCS 8779, Springer, 2014, pp. 40–49.
4. T. Patela, J. Undavia, and A. Patela, "Sentiment Analysis of Parents Feedback for Educational Institutes," Int'l J. Innovative and Emerging Research in Eng., vol. 2, no. 3, 2015, pp. 75–78.
5. F.F. Balahadia, M.C.G. Fernando, and I.C. Juanatas, "Teacher's Performance Evaluation Tool Using Opinion Mining with Sentiment Analysis," Proc. IEEE Region 10 Symp. (TENSYMP 16), 2016, pp. 95–98.
6. V. Dhanalakshmi, D. Bino, and A.M. Saravanan, "Opinion Mining from Student Feedback Data Using Supervised Learning Algorithms," Proc. 3rd MEC Int'l Conf. Big Data and Smart City (ICBDSC 16), 2016; doi:10.1109/ICBDSC.2016.7460390.
7. B.K. Mishra and A.K. Sahoo, "Evaluation of Faculty Performance in Education System Using Classification Technique in Opinion Mining Based on GPU," Computational Intelligence in Data Mining, vol. 2, H. Behera and D. Mohapatra, eds., AISC 411, Springer, 2016, pp. 109–119.
8. G.G. Esparza et al., "Proposal of a Sentiment Analysis Model in Tweets for Improvement of the Teaching-Learning Process in the Classroom Using a Corpus of Subjectivity," Int'l J. Combinatorial Optimization Problems and Informatics, vol. 7, no. 2, 2016, pp. 22–34.

情感和情绪识别

在本阶段，情感分析系统分析预处理数据以识别情感和情绪实例。系统采用《NRC情感词典》⁵（即《情感词典》）联系积极或消极的情感词语以及八种基本的情绪。《情感词典》支持40种语言，包括部分印第安语，如北印度语、泰米尔语、古吉拉特语、马拉提语及乌尔都语。其包括14182个英语一元模式词语的注释和8116个北印度语一元模式词语注释。

词典里的每个词语都有一个情感向量（E），这包含每种情感(s)和情绪(e)

所带有的布尔值(b):

如果学生评价中的一个单词匹配字典里的一个词语，则得到对应的情绪矢量；如果字典里与该单词匹配的词语超过一个，则得到其所有对应的情绪矢量总和。通过这种方式，每一评价

中呈现的不同情绪和情感可以得到相应的情绪矢量。例如，句子“老师，你真棒！”在情感分析系统中则会得到下述情感矢量：

这等同于积极的情感，因为信任和积极参数有等于1的b值。

愤怒	焦虑	厌恶	恐惧	愉快	伤心	恐慌	信任	消极	积极
0	0	0	0	0	0	0	1	0	1

此结果相当于积极情感，因为信任和积极的参数 b=1。

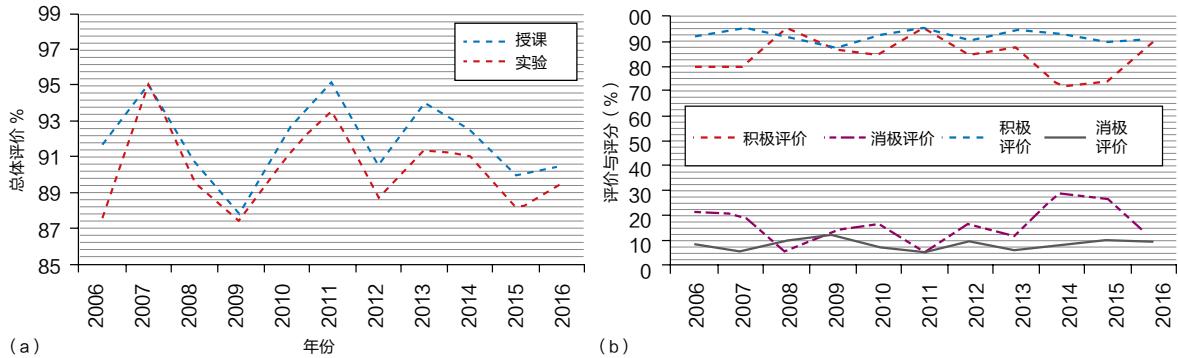


图 2: 短暂情感分析。(a) 学生对同一位教师上的同一门课的授课及实验环节在十年间的评级。学生对授课方面的评价稍高一些，在最后六年，学生的平均总体评价超过 90%。(b) 学生对同一位教师的积极和消极评价以及评级的百分比，平均而言，85% 的评价是积极的而 15% 的评价是消极的。

为了实现对情感和情绪的分析时间短，情感分析系统生成了每月及每年的平均情绪矢量(E_j)：

满意和不满意的计算

满意和不满在教育领域是关键的参数。该情感分析系统源于八种情绪中的六种，分别是愉快、信任、期待、愤怒、厌恶和悲伤。期待和信任显然意味着满意，但在某些情况下，愉快可能有消极的意味，例如：学生可能为逃掉一门无聊的课程感到愉快。因此，为了计算学生的满意度，我们把期待和信任的总和乘以一个常数 ($\alpha = 0.6$)，让上述参数占更多比例。在计算学生的不满意时，我们使用相同的机制，对比起伤心，增加愤怒和厌恶的比重。

计算方式如下：

$$\text{满意} = [\alpha(TA) + (1-\alpha)(J)] / n$$

$$\text{不满} = [\alpha(AD) + (1-\alpha)(S)] / n,$$

$$TA = \text{信任} + \text{期待}, J = \text{愉快}, AD = \text{愤怒} + \text{厌恶}, S = \text{悲伤}, n = \max(TA \text{ or } AD, J \text{ or } S).$$

思考以下两个例子。对于句子“他教学水平高”，情感分析系统会从

《NRC词典》中得到以下情绪矢量：

这里， $TA = 2, J = 1, n = \max(TA, J) = 2$ 。因此，满意度的计算为 $[0.6(2) + 0.4(1)]/2 = 1.6/2 = 0.8$ 。对于句子“他教学水平糟糕，每个学生都对课程有疑惑”，该系统得到以下情绪矢量：

愤怒	期待	厌恶	恐惧	愉快	伤心	惊讶	信任	消极	积极
0	1	0	0	1	0	1	1	0	1

这里， $TA = 2, J = 1, n = \max(TA, J) = 2$ 。因此，满意度的计算为 $[0.6(2) + 0.4(1)]/2 = 1.6/2 = 0.8$ 。对于句子“他教学水平糟糕，每个学生都对课程有疑惑”，该系统得到以下情绪矢量：

愤怒	期待	厌恶	恐惧	愉快	悲伤	惊讶	信任	消极	积极
1	0	1	2	0	2	0	1	2	0

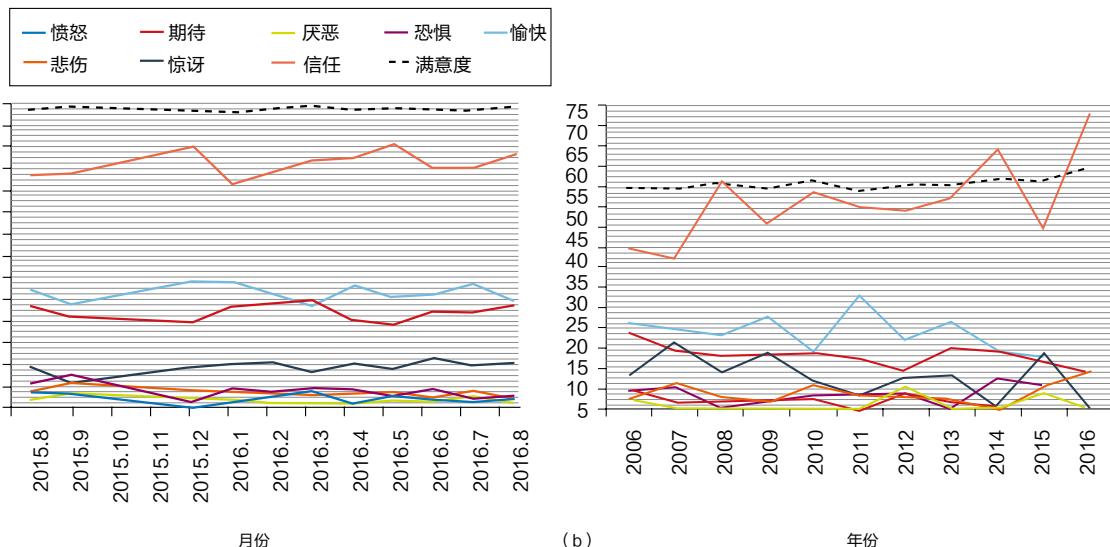


图3：短暂情绪分析。(a)从学生对为期一年的“课堂时代”课程反馈中提取的情绪百分比。表达积极情绪的学生比例比消极情绪的多，这显示他们满意该课程。(b)从学生对图2教师的评论中提取的情绪百分比。学生对授课人的信任度在逐步提升，而且每年表达积极情绪的学生都比消极多。

在这个例子中, $AD = 2$, $S = 2$, $n = \max(AD, S) = 2$ 。因此计算出的不满度为 $[0.6(2) + 0.4(2)]/2 = 2.0/2 = 1$ 。

数据可视化

为了让学生对课程满意度和教师表现的反馈分析更加方便, 我们的情感分析系统对数据进行了可视化, 生成了情感和情绪文字云及两者的变化曲线图。

情感和情绪词语云。学生在给出反馈时, 会利用许多词语来表达其情感或情绪。可视化通过词语云的形式表达积极词语(“棒”、“极好”、“有趣”等等)以及消极词语(“无聊”、“混乱不清”、“糟糕”等等), 这有助于判断学生的

学习行为, 例如: 他们是否在上有趣的课程或者进行有趣的实验等等。

短暂的情感或情绪分析。正如前文所言, 我们的情感分析系统组合了学生每月及每年的反馈中积极和消极的评价和评级。这使得人们可以随时追踪教师表现和课程的满意度。图2a呈现了从2006年至2016年间, 学生对同一教师的同一大学课程授课及实验方面的总体评级(从0-100%)。图表显示, 学生对教师在授课方面的表现评级稍高于对实验部分的评级, 并且过去六年的平均总体评级超过90%。图2b呈现了教师在同一时间得到学生的积极和消极评价和评级, 图表显示, 平均而言, 85%的评价是积极的, 而15%是消极

的。情感极性也可用于追踪不同的教师和课程的情况, 以分析在指定院校中总体的教学质量。

我们的情感分析系统每月及每年都收集学生关于教师和课程的评价所表达的情绪, 以提供更细致的观察。图3a呈现了学生每月对“课程时代”为期一年的某门课程的反馈, 从中提取各情绪所占的百分比, 比起消极情绪(悲伤、恐惧、厌恶和生气), 该图表所展示的积极情绪(信任、愉快、期待和惊讶)所占比例更高。图3b呈现的是学生对图2教师的评价中表达的情绪, 该图表显示, 过去十年间, 学生对教师的信任程度在逐步提升, 而且积极情绪所占比例每年都比消极情绪高。在两

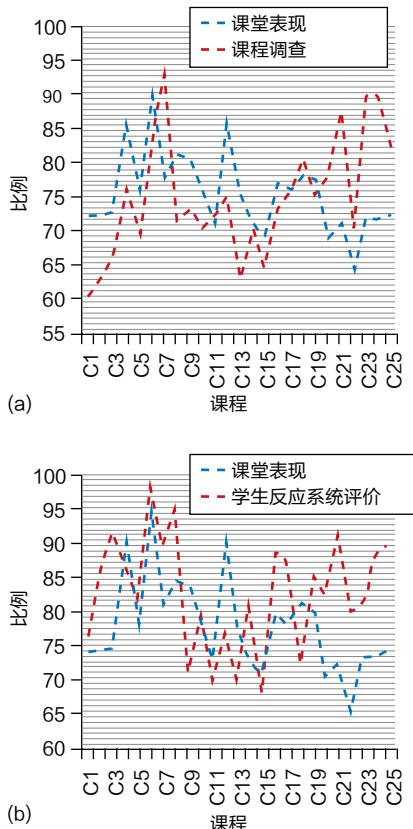


图 4. 对比学生表现，将平均成绩量化，比例值为 0–100。该图还呈现了(a) 调查中的积极情感比例，以及一所大学两年间从 25 门课程中获得的学生反应系统评价。两种方法的结果大致一致。

组数据集中，约 55% 的学生对教师感到满意。

系统评估

在教育领域，普遍认同关于教学质量和服务行为的直接和间接评估结果应该一致。例如，学生某门课程表现佳，那么他很有可能会给教师高评级以

及有利的评价；相反，如果学生表现不佳，则很有可能对课程不满意。

为了验证我们的情感分析系统，我们从一所大学的学生反应系统收集了两年间学生对 25 门课程的反馈，并比较学生对每门课程的积极情感所占比例，以及 100 分制的课程分数平均值。正如图 4 所示，结果是基本一致的，各方法间的绝对偏差为 20%。如果学生对一门课

程的分数超过了其满意度，则可能有许多解释：测验比较简单；该门课程有特别聪明或勤奋的学生修读；学生因个人原因不喜欢教师或认为他们并未从该课程中有很多收获。若学生对一门课程的满意度超过其分数，则可能是因为测试难度较大或是学生没有备考不充分。上述情况从两种方法中得出结果有相差，需要继续进行分析。

我们所提出的情感分析系统很有可能可以通过分析学生反馈中的情感、情绪和满意度参数改善大学的教育和学习，以帮助管理人员和教师明白问题所在并采取改善措施。学生对课程研究、论坛、博客及其他渠道的反馈信息量巨大，这是尚未被充分利用的资源，其可以通过使用机器学习技术得到有效利用，而该技术仍在不断发展。我们提出的系统对课程表现的分析结果与直接评估结果之间的比较，显示该系统可靠。

尽管该系统在此类研究领域有前途，但其仍有局限性。只有用正常的数据进行分析，该系统才有效，因此从学生处收集反馈时需要小心谨慎。学生反应系统必须通过良好的设计确保学生参与，教师必须通过协调，尽量确保学生提供完整准确的反馈。

未来，我们计划采用情感分析系统应用程序与学生反应系统及网上学习门户网站相结合，以实现学生反馈的实

时分析。我们也将再系统里加入更多的印地语，以扩展系统的多语种能力。■

参考文献

1. N. Altrabsheh, M.M. Gaber, and M. Cocea, “SA-E: Sentiment Analysis for Education,” *Intelligent Decision Technologies*, R. Neves-Silva et al., eds., FAIA 255, IOS Press, 2013, pp. 353–362.
2. M. Wen, D. Yang, and C.P. Rosé, “Sentiment Analysis in MOOC Discussion Forums: What Does It Tell Us?”, Proc. 7th Int’l Conf. Educational Data Mining (EDM 14), 2014; www.cs.cmu.edu/~mwen/papers/edm2014-camera-ready.pdf.
3. B.K. Mishra and A.K. Sahoo, “Evaluation of Faculty Performance in Education System Using Classification Technique in Opinion Mining Based on GPU,” *Computational Intelligence in Data Mining*, vol. 2, H. Behera and D. Mohapatra, eds., AISC 411, Springer, 2016, pp. 109–119.
4. A. Abdelrazeq et al., “Sentiment Analysis of Social Media for Evaluating Universities,” Proc. 2nd Int’l Conf. Digital Information Processing, Data Mining, and

关于作者

苏耶亚塔·兰尼是塔帕尔大学计算机科学及工程学系研究学者。她的研究兴趣包括自然语言处理 (NLP) 和机器学习。Rani 从塔帕尔大学获得计算机科学及工程获得工程学硕士研究生学位。她是美国计算机协会成员。联系方式: sujata.singla@thapar.edu。

帕蒂克·库马尔是塔帕尔大学计算机科学及工程学系副教授。他的研究兴趣包括自然语言处理、数据库和机器学习。Kumar 从塔帕尔大学获得自然语言处理博士学位。他是美国计算机协会成员。联系方式: bhatia@thapar.edu。

苏耶亚塔·兰尼和帕蒂克·库马尔在《用以提升教学效果的情感分析系统》一文中把情感分析定义为“用户对一篇文本的意识识别和分类的过程(如课程调查或网上论坛评价等)……用以判断用户对特定主体或实体的态度”并解释道“其对教学十分重要”。对比起单一的意见调查,情感分析事实上可以通过分析学生的作业得到更广阔的图景。Rani和Kumar通过以“课程时代”持续一年以上的学生评价和评级以及对一大学两年间不同教师所上的25门课程的评级为数据库,验证该情感分析模型有效。此类分析有助于教员根据学生的情感和其学习效果调整课程。

- | | |
|--|--|
| Wireless Comm. (DIPDMWC 15),
2015, pp. 49–62. | Emotion Association Lexicon,”
Computational Intelligence, vol. 29,
no. 3, 2013, pp. 436–465. |
|--|--|

5. S. M. Mohammad and P.D. Turney, “Crowdsourcing a Word-

利用“化学式”框架 管理网络教育资源

文 | 亨利·陈 (Henry C. B. Chan)，香港理工大学 (Hong Kong Polytechnic University)
译 | 林然

开放式教育资源 (*Open education resources, OERs*) 是指在网络上与传统教育机构中被广泛使用的公共学习资料。为了提高 *OERs* 的利用率，研究人员提出了一种新颖的“化学式”框架，通过在各类学习资料之间搭建相关联系，从而帮助教育机构制作、完善、定位这些宝贵的资源。

联合国教科文组织 (UNESCO，网址：www.unesco.org) 将开放式教育资源定义为“公共领域中或具备开放式许可协议的任何形式的教育资源。”此外，OER在线文库 (OER Common，网址：www.oercommons.org/about) 也给出了类似的定义：OER 是指“无需申请许可，即可反复免费使用的教学与学习资料。”

更广义地来说，能够通过互联网免费获取和使用的教育资源，以及具有某些特殊许可证书（比如 Creative

Commons 证书）的教育资源，都可以被称作开放式教育资源。全球各类教育机构已经在不同的学科中使用着这些开放式资源，以达成不同的教学目标。比如，西班牙国家远程教育大学 (National University of Distance Education, UNED) 的授课中，就采用了大量的开放式教学资源。得益于开源式制作工具，OER 式教学视频的制作成本十分低廉。*OERs* 不仅非常适用于本身就包含大量数字化内容的计算机软件课程，而且还能通过诸如模拟器等设备，将 OER 用于硬件课程教学，从而使学生们的学习时间和地点更加灵活。此外，*OERs* 还

能用于更为复杂的临床医学与医疗保健课程教学，并能制作成适合残障人士使用的教学资源。根据近期研究，影响 OERs 在社交网络环境中分享和使用的关键因素包括组织支持、语言、文化，以及质量。

本文对 OERs 和 IEEE 1484.12.1-2002 学习资料元数据 (Learning Object Metadata, LOM) 标准进行了介绍。为了弥补 IEEE 标准的不足，本文还提出了一种新颖的“化学式”框架，该工具能够在各类学习资料之间搭建相关联系——简而言之，教育者可以构造出“学习分子 (learning molecules)”。基于不同的教学与学习要求，这些“学习分子”由“连接键”连接而成。最后，为了进一步优化目前学习资料制作和利用的“生态”环境，本文提出了一种新的引用格式。

IEEE 学习资料与相关工作

使用 OER 的过程中，最重要的挑战在于如何对教育资源或学习资料进行描述。目前，主要方法包括针对数字资源和物理资源的都柏林核心元数据标准 (Dublin Core, 网址: dublincore.org)。此外，另一个重要的元数据标准——IEEE 学习资料元数据标准，针对的正是教育资源。该标准使用 XML 格式，对教育资源的结构与数据元进行统一而灵活的定义。一

表 1 学习资料管理方法

方法	简介
基于树形结构	» 基于资源目录储存搜索与注册架构 (Content Object Repository Discovery and Registration/Resolution Architecture, CORDRA) 的系统具有可重用性树 (reusability tree) 组织结构，能够高效地搜索和反复使用学习资料。 » 基于可重用性树的学习资料管理系统，具备加权与排序机制。
基于端对端 (P2P)	» 利用端对端系统分享学习资料
基于本体论	» 基于本体论方法的系统，并结合自然语言处理与文本挖掘技术对学习资料进行管理。
“化学式”方法	» “化学式框架”通过不同的连接键将学习资料连接起来，合成“学习分子”

^aF.H. Lin, T.K. Shih, and W. Kim, “An Implementation of the CORDRA Architecture Enhanced for Systematic Reuse of Learning Objects,” IEEE Trans. Knowledge and Data Eng., vol. 21, no. 6, 2009, pp. 925–938.

^bN.Y. Yen et al., “Ranking Metrics and Search Guidance for Learning Object Repository,” IEEE Trans. Learning Technologies, vol. 3, no. 3, 2010, pp. 250–264.

^cR. Santiago and A. Raabe, “Architecture for Learning Objects Sharing among Learning Institutions – LOP2P,” IEEE Trans. Learning Technologies, vol. 3, no. 2, 2010, pp. 91–95.

^dA. Zouaq and R. Nkambou, “Enhancing Learning Objects with an Ontology-Based Memory,” IEEE Trans. Knowledge and Data Eng., vol. 21, no. 6, 2009, pp. 881–893.

般来说，定义标准包括 9 个类别。

- 概要信息（比如标题），
- 发展历程（历史和现状，比如：版本号），
- 元数据信息（比如：制作时间），
- 技术要求与信息（比如：格式），
- 教育信息（比如：交互类型），
- 版权信息（比如：版权与使用限制），
- 关联信息（比如：与其他资源的关联性），
- 注释信息（比如：描述与评价）
- 分类信息（资源归属于何种类别系统，比如：来源）。

为了管理学习资料，人们通常将

其储存于资源库中。近期研究表明，主要有四种管理与储存学习资料的方法：学习管理系统、学习资料查阅库、学习资料储存库，以及公开课课件。除此以外，人们还提出了一些可用于学习资料管理的其他方法，如表 1 所示。基于资源目录储存搜索与注册架构 (Content Object Repository Discovery and Registration/Resolution Architecture, CORDRA) 的系统具有可重用性树 (reusability tree) 组织结构，能够高效地搜索和反复使用学习资料。另一个基于可重用

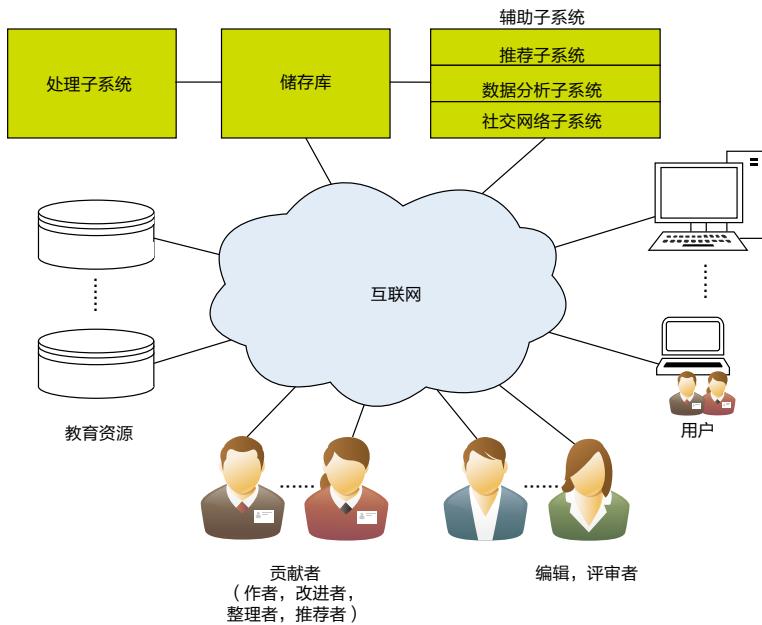


图 1. 系统构架图，储存库负责根据 IEEE 1484.12.1-2002 学习资料元数据 (Learning Object Metadata, LOM) 标准，将学习资料元数据，以及其宏观和微观属性存放在一起，而实际教育资源则存放在作者自己的系统中。

资料（学习原子）相互连接，形成“学习分子”，其与自然界中原子组合形成分子的过程十分相似。

系统架构

该系统的储存库与处理子系统的设计围绕着“化学式”框架展开，使学习资料的作者们（如教师）能够通过一个中央储存库将教育和学习资料分享出去。具体来说，储存库将基于 IEEE 学习资料元数据标准，兼备宏观与微观属性，仅将学习资料的元数据进行储存。而实际的教育资源（如视频课程）将被存放于作者自己的系统中。因此，该储存库仅根据教育资源的元数据和熟悉提供其连接或链接，而不储存资源本体。该系统的基本架构如图 1 所示。

为保证控制质量，提交的教育资源可以进行同行评审，与 Merlot 在线教学库以及期刊学术论文发表过程中的评审机制类似。用户可以通过网络使用该系统，搜索学习资料或“学习分子”。系统后端包括负责处理学习资料和用户要求的子系统。此外，还有用于实现其他功能的辅助子系统，比如向用户推荐学习资料和“学习分子”、通过数据分析用户行为，以及提供社交网络相关功能。

此系统的基本工作流程如图 2 所示，主要展示了其储存库和子系统处

图 2. 系统工作流程图，后端系统处理学习资料和用户需求。

性树的学习资料管理系统还具备加权与排序机制。通过嵌入相关插件和中间层使管理系统相互连接后，学习资料就可以通过点对点 (peer-to-peer, P2P) 的形式进行分享使用。基于本体论方法的系统，并结合自然语言处理与文本挖掘技术，同样也可以对学习

资料进行管理。

与上述各种方法相比，本文提出的“化学式”框架更具灵活性与创新性。其目的在于将各种学习资料通过不同的“连接键”相连，并以全新的视角研究它们。换而言之，通过类似于化学反应的计算处理过程，将多个学习

表 2 “化学式” 框架元素

元素	定义/类比
原子	学习资料的功能与原子类似
核子	宏观属性的功能与核子类似
电子	微观属性的功能与电子类似
连接键	连接键连接两个学习资料，并表明两者关系（比如互相强化、互补，或者其他复杂关系）
共价键	强化键的功能与共价键类似（共享电子）
离子键	互补键的功能与离子键类似（互补电子）
能量	一些电子更加活跃/能量更多，模型中引入能量系数表征这种现象
连接键强度	连接键强度决定着两个学习资料（原子）之间连接的紧密程度
分子	“学习分子”可以由学习资料（原子）通过不同类型的连接键组合而成
化合	通过化合处理计算过程赋予微观属性（电子）
化学反应	学习资料（原子）在特定条件与约束的情况下，通过类似于化学反应的计算过程结合在一起
其他连接键	还可以使用其他的模糊或概率连接键（比如量子理论中的连接键）

理流程。首先，作者或贡献者把学习资料的元数据提交至储存库中。随后，编辑团队对挑选出来的学习资料实施同行评审，对该资源进行认可。然后，用户即可通过特定标准——如利用元数据信息，进行学习资料的搜索。此外，用户还能够基于特定的目标、条件和约束，对学习资料或“学习分子”提出筛选要求。处理子系统随即对这些要求进行处理，并将筛选结果呈现给用户。最后，用户就能获得来自于网络的教育资源。

“化学式”框架

本文提出的“化学式”框架中，

学习资料是最关键的系统元素；它们如同原子一样，通过“连接键”相互连接后能够形成分子。“化学式”框架正是受到了该模型及其相关研究的启发，比如相似性/多样性、加权机制。所以，该框架的目的在于将学习资料连接起来（见表 2）。如同原子中的核子与电子一样，每个学习资料都分别被赋予了宏观属性和微观属性。比如，以下宏观属性提供了学习资料的核心信息。

· 储存库（学习资料元数据的储存位置）

· 身份编号（由储存库为学习资料提供的编号，用于识别不同学习资料）

- 版本（如：序列号）
- 领域（如：计算机学）
- 子领域（如：计算机网络学）
- 教育等级（如：大学水平）
- 类型（如：辅导教程）
- 格式（如：视频）

这些信息可以从作者或贡献者提供的元数据中提取，也可以由系统（比如学习资料元数据标准）生成。本质上来说，这些宏观属性和元数据将分别提供学习资料的定量和定性信息。

与此相比，微观属性提供的则是更加具体而特定的细节。简而言之，这种基于二进制的属性反映出的是匹

表 3 学习资料实例

学习资料	关键字	微观属性
O1	Dijkstra算法、OSPF算法	1100
O2	Dijkstra算法、OSPF算法	1100
O3	Dijkstra算法、Bellman–Ford算法、OSPF算法	1100
O4	Dijkstra算法、Bellman–Ford算法	1010
O5	OSPF算法、RIP	0101

配程度。举例来说，111 表示该学习资料与相关属性完全匹配，而 001 则表示仅与最后一个属性匹配。值得注意的是，二进制的微观属性可以转换为十进制——比如二进制中的 111 可以转换为十进制中的 7。与化学理论类比，微观属性就像围绕在原子核外特定轨道中的电子。比如，101 表示存在两个电子——分别位于第一和第三轨道中，而第二轨道中无电子存在。这些电子能够与其他学习资料（原子）的电子结合，形成连接。

可以根据不同的连接方式定义各种不同的连接种类，但这里为了便于理解，将其视为某种特定连接。一般来说，“学习分子”能够通过不同的连接而形成。与化学分子中的共价键（共享电子）类似，101 和 101 能够形成最强的强化 / 共价键。换而言之，所有电子被共享。与共价键相反的是离子键（互补电子），101 和 010 能够组成最强的互补 / 离子键。

考虑到一些电子或许比其他电子更加重要，这些电子带有更多能量，更为活跃。因此，该模型赋予了每个电子不同的能量系数（取值 0-1 之间），下文将详细解释。如此一来，两个学习资料（原子）之间的连接键强度就取决于电子能量系数与连接键类型。比如一个学习资料（原子）的电子为 101，最里和最外电子的能量系数分别为 1 和 0.5。若规定连接键强度等于电子能量的累加值，则两个这样的学习资料（原子）之间，就能形成强度为 1.5 (1+0.5) 的强化 / 共价键。一般来说，连接键的强度将表明两个学习资料（原子）之间的关联程度。

通过自动计算或化合处理过程 (fusion process)，可以为学习资料赋予微观属性（电子）。基本上来说，每个学习资料（原子）都能根据元数据或关键字中的 XML 标签内容进行描述。这些关键字可以通过众包形式 (crowdsourcing) 由贡献者提供，也

可以基于学习资料元数据信息，利用数据文本挖掘技术自动生成。

举个简单的例子，基本的化合处理过程如下：假设有 4 个学习资料(O1、O2、O3 和 O4)，各个资源关键字分别为 O1: {a, b, c}、O2: {a, b, d}、O3: {e, f}、O4: {b, e}。也就是说，O1 的关键字包括 a、b、c。请注意这些关键字是按照重要程度排列的(b 出现了三次，所以其重要程度最高)，同等重要的关键字则按字母排序。这些学习资料中，组成其微观属性的关键字包括 a、b、c、d、e 和 f。本例中，O1 的微观属性为 110100。换而言之，化合处理后的 O1 拥有三个电子。

其中，一些电子比另一些重要。正如前文所述，它们具有不同的能量系数。比如，本例中有 4 个学习资料，而关键字 b 出现了 3 次。因此，我们将关键字的活跃程度定义如下：b 的重要程度（即能量系数）为 3/4 或 0.75。类似的，a 和 e 的能量系数为 2/4 或 0.5，剩下的 c、d、f 的能量系数则为 1/4 或 0.25。如果两个微观属性为 110100 的“学习分子”相互结合，其强化 / 共价键强度则为 $0.75+0.5+0.25=1.5$ 。另一方面，当 110100 与 001011 的“学习分子”相互结合，其互补 / 离子键强度则为 $0.75+0.5+0.5+0.25+0.25=2.5$ 。可以看出，这两个“学习分子”完全互补。利用这种规则，用户就能根据

表 4 连接键强度

学习资料	O1	O2	O3	O4	O5
O1	n/a	(1.6, 0)	(1.6, 0.4)	(0.8, 1.2)	(0.8, 1.0)
O2	(1.6, 0)	n/a	(1.6, 0.4)	(0.8, 1.2)	(0.8, 1.0)
O3	(1.6, 0.4)	(1.6, 0.4)	n/a	(1.2, 0.8)	(0.8, 1.4)
O4	(0.8, 1.2)	(0.8, 1.2)	(1.2, 0.8)	n/a	(0, 2.2)
O5	(0.8, 1.0)	(0.8, 1.0)	(0.8, 1.4)	(0, 2.2)	n/a

表 5 不同的“学习分子”

目标/条件	约束	“学习分子”
最强的强化/共价键	无	O1和O2、O1和O3、O2和O3
最强的强化/共价键	互补/离子键不为零	O1和O3、O2和O3
最强的互补/离子键	无	O4和O5
最强的互补/离子键	强化/共价键不为零	O3和O5

自己的要求，将多个学习资料相互连接，组合成“学习分子”。

实例分析

该实例用到了 ACM 和 IEEE 计算机学会 2013 年的计算机科学课程表 (www.acm.org/education/CS2013-final-report.pdf)，该课程表涵盖了 18 个知识领域，全球很多计算机学院在制作课程表时都以此为参考。假设一个教授在系统中找到了 5 个有价值的学习资料（见表 3），并打算挑选其中两个（由两个学习资料形成的一个“学

习分子”），在自己讲授的《路线规划及其算法》课程中使用。

基于处理子系统中的化合处理过程，微观属性相关参数如下：Dijkstra 算法 ($4/5 = 0.8$)、开放式最短路径优先先 (Open Shortest Path First, OSPF) 算法 ($4/5 = 0.8$)、Bellman-Ford 算法 ($2/5 = 0.4$)、路由信息协议 (Routing Information Protocol, RIP) ($1/5 = 0.2$)，括号中的数值代表着其各自的能量系数。表 4 中计算了任意两个学习资料组合后的连接键强度。表 5 中提供了教授在系统中输入

不同目标、条件和约束情况下的最优“学习分子”备选项。比如，如果教授想得到最强的强化 / 共价键（学习资料相互加强），最优的选择有：O1 和 O2、O1 和 O3、O2 和 O3。这三个选择所获得的强化 / 共价键强度都为最大值——1.6。

然而，如果教授还要求互补 / 离子键强度不为零（学习资料之间要有一定程度的互补），最优选择则为：O1 和 O3、O2 和 O3。如果教授需要最强的互补 / 离子键，O4 和 O5 则是最优选择（其互补 / 离子键强度为 2.2）。

除了技术以外，内容贡献者无疑也在学习资料系统中扮演着重要作用。

因为两者之间完全互补。如果教授还要求强化 / 共价键不为零（学习资料之间要有一定程度的相互加强），最优选择则为 O3 和 O5。此时，两者的连接键总强度依然为 2.2。

利用同样的方法，可以组合出更复杂的“学习分子”。比如，从“核心”学习资料开始，为其组合上另一些学习资料，并在此基础上，利用不同的连接键，再组合上其他更多的学习资料。

贡献者与引用格式

除了技术外，人（内容贡献者）无疑也在学习资料系统中扮演着重要作用（见图 1）。因此，我们还需要一套激励机制，鼓励人们向该系统中提交学习资料。

受到期刊论文发表流程的启发，我们提出了一套能够鼓励提交学习资料的引用格式。如此一来，教职员的简历中不仅包括论文发表情况，还可以包括学习资料贡献情况。尽管学习资料发表与期刊论文发表之间存在相似之处，但也有明显不同。比如，学习资料比期刊论文更为动态，因为

其他人员（教师或用户）能够对教育资源进行不断地完善和改进。

因此，学习资料的贡献者包括四种：作者（author）、改进者（enhancer）、整理者（integrator）和推荐者（recommender）。在此，利用以下简单实例说明我们提出的引用格式：

此例中，一个贡献者 A（作者）准备了一段计算机学入门教学视频。他可以将其“发表”在编号为 1 的储存库中、按照适当的发表流程，该视频的学习资料编号被定为 1.100.1.1.3.3.1.4（宏观属性）。随后，贡献者 B 通过添加视频笔记的方式，改进并重新发表了该视频。此时，引用格式如 2 所示。其中，版本号码中的 1 改为了 2。接着，贡献者 D 将贡献者 A 提供的视频和贡献者 C 提供的补充学习资料整合成为一套新的学习资料，其中也可能包含 D 自己贡献的内容。此时，可将这套新资源发表在储存库中，其引用格式如 3 所示。此外，如果贡献者 A 没有发表该资源，那么其他人也可以作为推荐者将其发表。比如，贡献者 E 发现该教学视频

非常有用，所以将其发表在了储存库中，其中可能也包含 E 自己贡献的内容（如提供用户体验评价）。

上述例子较为简化，所以学习资料的网页链接并未包括在内。实际上，网页链接可以和其他重要信息，一并列入引用格式中。



文提出的“化学式”框架还可能延伸出有趣的研方向。比如，本文只提及两类基本的连接键和最简单的“学习分子”。还可以研究其他类型的连接键（概率键、模糊键），用于表示学习资料之间的连接。除此以外，还可以将上述的实例分析进行扩展延伸，研究“学习分子”复杂性的上限，同时考虑诸如用户评论和评分等其他因素，这些研究将不断完善该模型。

目前，电子商务中使用的筛选方法，或许也能用于向用户推荐学习资料或“学习分子”。因为该系统能够追踪记录用户行为，所以可为推荐系统提供有效的信息（比如基于不同的关联法则）。

除了标准网页，数据可视化工具也可以帮助用户浏览和管理学习资料，并针对不同结构的“学习分子”进行交互式审查。

一般来说，诸如进化算法等人工智能算法也能用于组成更为复杂的“学

关于作者

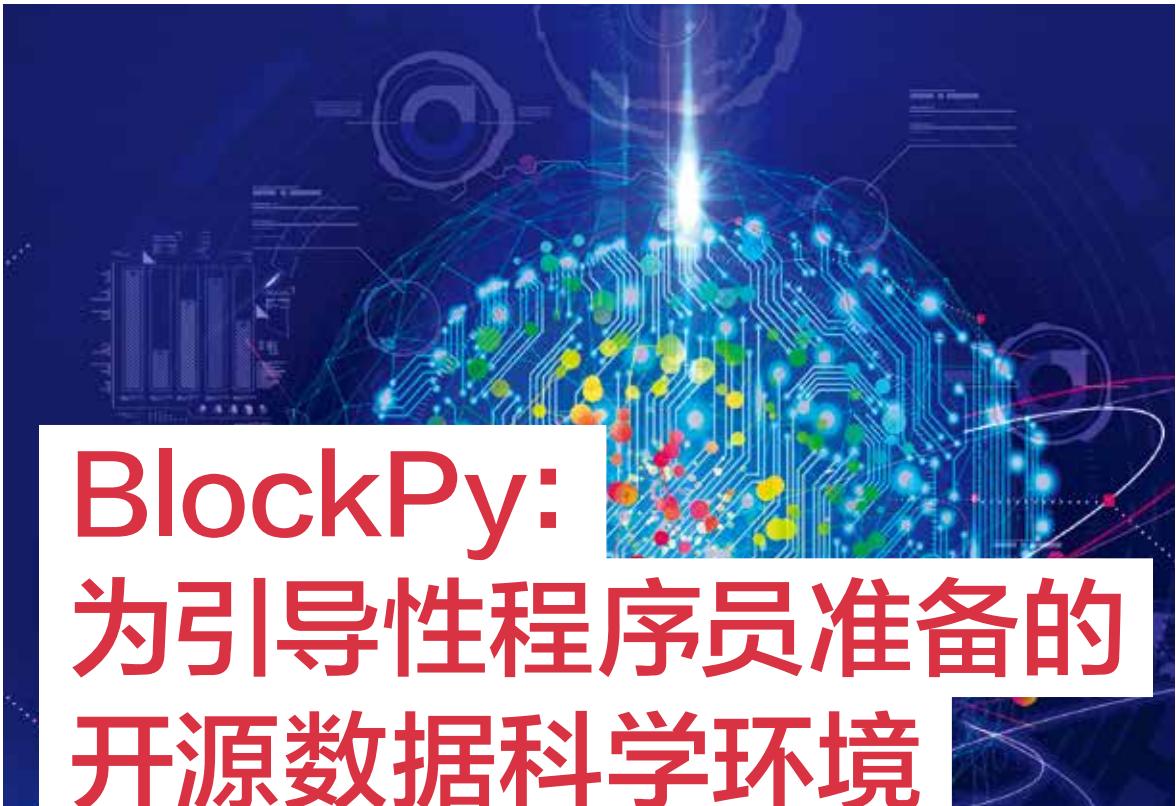
习分子”，以满足特殊的条件和约束。

利用众包的形式也有助于关键字的管理与教育资源的评分。然而，仍需要引入合理的质量管理机制（比如出现可疑评论时，向用户发出提示）。此外，采用同行评审和用户评分相结合的方式提高学习资料的质量，也将是有趣的研究。

最后，该系统还能提供另外一些辅助功能，比如结合社交网络合成“学习分子”，以及通过数据分析技术研究用户行为。■

参考文献

1. E. Tovar and N. Piedra, “Guest Editorial: Open Educational Resources in Engineering Education: Various Perspectives Opening the Education of Engineers,” *IEEE Trans. Education*, vol. 57, no. 4, 2014, pp. 213–219.
2. H. Rehatschek, U. Leopold, and S. Kerschbaumer, “Introduction of OER at the Medical University of Graz: Practical Tips to Introduce OER with a Minimum of Additional Budget and Efforts,” *Proc. Int'l Conf. Interactive Collaborative Learning (ICL 14)*, 2014, pp. 121–126.
3. R. Navarrete and S. Luján-Mora, “OER-Based Learning and People with Disabilities,” *Proc. 2015 Int'l Conf. Interactive Collaborative and Blended Learning (ICBL 15)*, 2015, pp. 25–34.
4. H. Pirkkalainen, J.P.P. Jokinen, and J.M. Pawlowski, “Understanding Social OER Environments: A Quantitative Study on Factors Influencing the Motivation to Share and Collaborate,” *IEEE Trans. Learning Technologies*, vol. 7, no. 4, 2014, pp. 388–400.
5. IEEE Std. 1484.12.1-2002, *Learning Object Metadata (LOM)*, IEEE, 2002.
6. X. Ochoa and E. Duval, “Quantitative Analysis of Learning Object Repositories,” *IEEE Trans. Learning Technologies*, vol. 2, no. 3, 2009, pp. 226–238.
7. F.H. Lin, T.K. Shih, and W. Kim, “An Implementation of the CORDRA Architecture Enhanced for Systematic Reuse of Learning Objects,” *IEEE Trans. Knowledge and Data Eng.*, vol. 21, no. 6, 2009, pp. 925–938.
8. N.Y. Yen et al., “Ranking Metrics and Search Guidance for Learning Object Repository,” *IEEE Trans. Learning Technologies*, vol. 3, no. 3, 2010, pp. 250–264.
9. R. de Santiago and A.L.A. Raabe, “Architecture for Learning Objects Sharing among Learning Institutions—LOP2P,” *IEEE Trans. Learning Technologies*, vol. 3, no. 2, 2010, pp. 91–95.
10. A. Zouaq and R. Nkambou, “Enhancing Learning Objects with an Ontology-Based Memory,” *IEEE Trans. Knowledge and Data Eng.*, vol. 21, no. 6, 2009, pp. 881–893.
11. C.M. Kelty, C.S. Burrus, and R.G. Baraniuk, “Peer Review Anew: Three Principles and Case Study in Post-publication Quality Assurance,” *Proc. IEEE*, vol. 96, no. 6, 2008, pp. 1000–1011.



BlockPy： 为引导性程序员准备的 开源数据科学环境

文 | 奥斯汀·科瑞·巴特 (Austin Cory Bart), 贾维尔·提巴乌 (Javier Tibau), 艾利·蒂勒维奇 (Eli Tilevich), 克利福德·A·沙弗尔 (Clifford A. Shaffer), 丹尼斯·卡福拉 (Dennis Kafura), 弗吉尼亚理工大学 (Virginia Tech)

译 | 宋杰

非计算机科学专业的人们通常难以在倾向于强调抽象概念、专注于非实践性娱乐、或依赖于非文本化设置的传统计算课程中找到相关性。BlockPy是一个基于Web的开源Python编程环境，该环境可以通过块／文本编程视图，为数据科学环境中的引导式程序员提供支持。

由

于计算在所有领域都普遍存在，除了自己领域的核心知识外，各行各业的专业人士越来越需要学习计算的技能。大学阶段的普通教育计算课程（例如计算思维课程）正在不断扩展，大量开放的在线课程也在蓬勃发展，许多专业人士和学生也在通过非正式的渠道进行学习。然而，许多学习者的计算经验很少，并且不确定计算对他们的长远职业目标有何裨益。学习者不仅需要与其能力相匹配的特殊工具，也不希望在获取这些资料的过程中遇到

太多的困难。我们提出了一个可以更好地服务于具有此类需求学习者的新工具——BlockPy。它是一个开源的，基于Web的Python数据科学环境，为引导式学习者提供指导和界面支持（www.Blockpy.com）。

引导性课程语境化的现代方法主要集中在体验性和趣味性，重点是游戏设计和媒体计算。然而，学生的学习动机具有复杂性，且不仅仅取决于情境兴趣。整体动机模型表明，学生还需要感觉到材料对学习是有用的，并且其长期的职

业目标得到满足。媒体计算的背景知识并不总是被非计算机科学专业认为有用。

我们相信, 数据科学是一种激励性的环境, 可以以不同的方式吸引学生, 这得益于其他专业对数据处理的广泛需求。在之前的工作中, 我们介绍了数据科学作为学习环境的可行性和影响。学生经常学习计算, 以学习如何管理在一个学科或特定的自我导出项目中存储和分析的令人眼花缭乱的数据量。在这方面的内容的基础上, 学生可以更容易地理解计算的相关性, 并更清楚地了解材料如何组合在一起。通过将上下文与学生的长期需求相结合, 学生将学习与这些需求更相关的技能。最后, 数据科学作为一种环境, 自然就适用于与结构化数据、迭代和其他核心材料相关的教学课题, 使其对计算机科学导师具有教学价值。

PYTHON和块

因为语法简单、功能强大, Python已经成为最受欢迎的入门编程语言。它提供对数据科学的强力支持, 正如我们在 Matplotlib 这样的流行图书馆中所看到的一样。Python 需要很少的代码就可以完成有趣的事情, 所以新手不会被复杂的语法细节所羁绊。它在引导性课堂和行业中的广泛使用进一步坚定

了我们对 Python 的选择。

任何类型的编程对初学者来说都是一个挑战, 因为编码被称为“曾经发明的最强大但可用性最差的人机界面”。基于块的语言(如 Scratch 和 AppInventor)已被证明可以缩短初学者开始编程和完成任务的时间。通过提供语言的整个用户界面的结构和即时视图, 将对引导性学习者大有帮助。

现今, 已经有几种环境可以让学生和导师在浏览器中编写 Python, 包括 CodeSkulptor, Pythy, 以及在线 Python 导师(Online Python Tutor, OPT)。BlockPy 站在巨人的肩膀上, 整合了由这些环境所启发的功能, 并引入更新颖的功能。但是没有一个现有的 Python 环境可以帮助学生过渡到文本编程语言。

BlockPy 旨在为基于块和基于文本的代码创作提供双重支持。在任何时候, 学生都可以在基于块的代码视图和传统的基于文本的视图之间进行自由的切换。这个强大的功能受到使用自己标识语言的铅笔代码(Pencil Code)的启发。类似的实现已经成功地成为学生的有力工具。

BlockPy 扩展了 Pythy 对“作业”的支持——将演示与评估相结合的问题。Pythy 只支持传统的单元测试来为学生提供反馈, 而 BlockPy 提供了一个用于代码分析和自由格式文本指导的

API, 导师可以配置这些 API 来向学生提供有用的建议。此外, Pythy 对数据科学的支持有限, 而 BlockPy 拥有丰富的数据源库和基于 Matplotlib 的绘图 API。

CodeSkulptor, Pythy 和 BlockPy 都使用相同的内部引擎来运行 Python 代码(Skulpt)。尽管 CodeSkulptor 拥有广泛的用于创建用户界面和游戏的 API, 但它提供了一个非标准的库, 这很难帮助学习者过渡到严肃的编程环境。BlockPy 的理念是保持与实际系统的大致兼容性。例如, BlockPy 模拟了 Matplotlib 界面, 而不是使用自定义绘图 API。

OPT 已被证明是可视化程序状态的有用工具。然而, 它包含了一些较难理解细节, 足以难倒入门的学生(例如, 它使用诸如框架和对象等术语, 这对初学者来说无异于外语一般)。BlockPy 的状态浏览器不会尝试匹配 OPT 的彻底性, 而是提供了一个有用且简单的程序状态图。此外, BlockPy 通过依赖于在浏览器中运行的 Skulpt 来避免 OPT 对服务器的依赖。

BlockPy概述

BlockPy 的主要设计目标是:

- › 减少学习编程的障碍;
- › 通过帮助学生解决现实问题, 提

封面文章：学习技术的发展

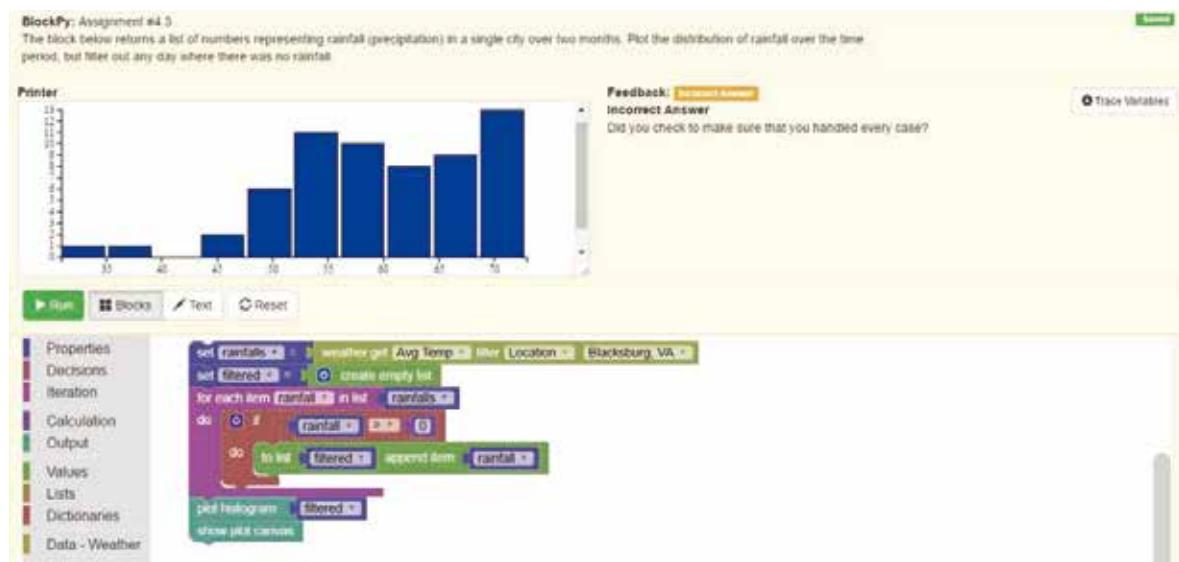


图 1. 正在完成编程作业的学生的 BlockPy 环境屏幕截图。请注意底部正在进行的当前学生解决方案以及右上方框中显示的最新反馈。

升真实性；

- › 通过实用工具提升成熟度（例如，从块转换为文本）；
- › 最大限度地减少人类导师的帮助。

开源, 开放访问

BlockPy 旨在成为一个高度开源的基于 Web 的平台, 供任何人学习如何编程。所有代码都是开源的, 并且利用了许多开源库。使用软件不需要注册, 尽管用户可以从免费注册中获益, 如可以管理教室。BlockPy 还提供教育者分享的指导性学习资料。

BlockPy 编辑器(见图 1)在输入时持续存储用户代码。日志存储在击键级别, 以便将来进行程序分析, 这一点我们将在下文进行详细阐述。因此, 会话之间可以使用最新版本的用户代码。当在离线模式下运行时, 代码存储在 `LocalStorage` 浏览器对象中; 当重新建立连接时, 执行同步操作。

Python 执行

BlockPy 系统设计为在脱机状态下工作, 这使其成为无法连接 Internet 情况下的理想编程环境。代码执行通过修改的 Skulpt JavaScript 库中的实例来实现。Skulpt 是一个完整的 Python 解析器和编译器, 通过生成 JavaScript 代码(包括对丰富的 Python 标准库的部分支持)来支持几乎所有的 Python 语言功能。Skulpt 执行环境完全位于用户浏览器中, 因此除初始页面加载之外, 不会依赖外部服务器。

基于块的 Python

为了支持引导性学习者处理 Python 语法, 最初的 BlockPy 接口是基于块的, 并使用流行的 Blockly JavaScript 库。语言功能(迭代、决定、变量分配和访问等)包含在界面左侧的工具箱中, 用户可以从中拖放块到画布之上。BlockPy 的块接口只生成语法上有效的 Python 代码, 由块的“捕捉”

连接器执行(尽管可能会生成语义不正确的代码, 这将在后面讨论)。该块的接口与文本界面同步。

Python 的一个重要的问题是, 基于 Python 的表示形式, 应该以多高的频率提供多少量的语言细节。Python 中 `for` 循环的一个很少使用的特性是, 包含一个在循环成功完成后执行的 `else` 子句(即使用 `break` 语句不会过早地转义)。此高级语言功能类似于具有例外的 `finally` 语句。然而, 如果一个 `else` 子句被提供给刚刚从迭代开始的那些语句, 那么它们可能会将概念与 `if` 语句中使用的条件 `else` 子句混淆。认知负载对于初学者来说是苛刻的, 用户界面应避免提供不必要的细节。

虽然在一个 `for` 循环中隐藏其他的物体是一个很清楚的例子, 但是有一些不那么清楚的例子。当学习者准备好使用并行分配时可能难以识别, 因此应该能够在分配块的左侧指定多个变量。基于块的语言迫使导师对如何公开

语言特征做出重要决定。对于未来的 BlockPy 工作，我们正在尝试以不同的频率显示语言特征，这可以由导师进行调整，因此系统可以为学生提供更准确的语言模型。

适应性指导实践

BlockPy 最强大的功能之一是其交互式的指导反馈。由于编程环境的限制，如 Snap!，一些编程环境无法做到教学互动——不能指导在系统中完成作业的学生走向成功。学习者必须决定何时完成课程，以及是否符合规范。对于独立的非正式学习者，这需要高水平的自我调节和认知。BlockPy 的自适应元素遵循专家模型；当学生运行代码时，会根据导师提供的逻辑来检查。如果学生的代码由于某种原因而失败，则提供一个建议。如果代码正确，系统会显示一个绿色的“完整”标记，这被认为具有激励的作用。

在 BlockPy 的导师模式中，导师提供了一个问题实例。首先，WYSIWYG (what you see is what you get, 所见即所得) 富文本编辑器编辑问题描述，支持任何有效的 HTML 内容（如图像和链接）。第二，导师提供影响学生体验的特殊画布中的代码，使用与学生使用的相同的文本 / 块界面。首先是开始的代码，当学生开始出现问题时（为避免空白的画布）即对学生进行展示。第

二个导师代码定义了交互式的反馈，可以访问学生的代码，最终输出对程序状态的完整跟踪。检查系统可以声明代码是正确的或显示一个作为用户反馈呈现的 HTML 字符串。导师可以自由地编写他们想要的任何逻辑，例如搜索特

问题特定的错误。初学者通常不能理解某些变量的真正目的，并通过模拟一个示例将它们错误地包含在代码中，但是 BlockPy 可以通过执行简单的变量活动分析来识别这些变量引发的错误。大多数现代编程环境都采用这种分析，但

**BlockPy 对于任何编程学习者来说
都是极易访问的网络平台。**

定的抽象语法树 (AST) 元素，测试控制台上的输出，或者通过程序状态来满足不变量。用于常见检查的 API（例如解析程序的 AST）是根据常见用例进行演进的，以确保它们不调用禁止内置的 Python 函数。

我们也正在探索导师的干预措施，不仅仅是提供文字反馈，例如显示一个带嵌入式教学视频的弹出式对话框，或者提醒导师为特定的学生提供即时教学。此外，我们正在尝试一个新的系统，用于解决学生与编程环境之间的误解，以提供更好的指导，解决学生的具体错误。

通常相信用户有添加变量的原因，从而被动地进行响应。

因为 BlockPy 安全地记录了用户程序的历史记录和日志界面交互，所以它可以挖掘这个代码库，以推断出不良学习者行为的常见模式。例如，如果频繁移动相同的块而不在问题目标中进行编程的用户解决该问题比其他用户需要更长的时间。或者，选择决定块来完成迭代问题的学生可能需要特别注意。通过证明或反驳这些假设，我们可以改进系统的自动反馈并提供更多的个性化支持。

学习技术互操作性

BlockPy 支持学习技术互操作性 (LTI) 协议——导师可以在现有课程管理软件（如 Canvas 和 OpenEdX）中嵌

深度学习的程序分析

BlockPy 使用简单的程序分析技术来发现新手往往会犯的一般错误和

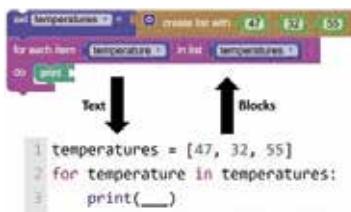


图 2. 块 / 文本转换的用户视角。块代码的顶部演示了创建一个列表并循环遍历的过程。底层代码片段是生成的 Python 文本代码。连接顶部和底部的箭头表示代码的双向性质。

入问题并接收分配结果如成绩的机制。一个典型的学习者使用 BlockPy 而没有意识到 LTI。使用系统的导师获得在其学习管理系统 (LMS) 中使用的秘密密钥和配置 URL。课程网站上的学生可能会使用 BlockPy 而不注册 BlockPy 帐户首次通过 LMS 提供的链接登录时，他们通过 LMS 的其他信息通过常规帐户在系统中进行无效的注册。随着学生完成工作，作业进度会报告给 LMS。导师使用特殊的交互式菜单来管理与课程相关的练习。

数据科学块

BlockPy 专注于数据科学作为其主要背景，因此它包括处理数据的块。BlockPy 支持流行的 Matplotlib 库的一个子集，并将其扩展到与 CORGIS(实时、巨型、有趣的位于数据集的集合)项目的连接。例如，Matplotlib API 提供了一个由“plot()”函数创建的简单线图。通过模拟 Matplotlib，学生可以无缝转

移到严格的 Python 编程环境，而不会丢失代码。

CORGIS 项目通过简单的编程库为入门学生提供了激励数据。这些数据集来自许多学科，这也导致学习材料具有普遍性和相关性。目前，BlockPy 支持多种不同的 CORGIS 库，包括天气数据、地震数据、美国犯罪统计数据和经典书籍数据集。

具有一元增量运算符。BlockPy 的语法和词汇来自于 Logo。

我们放弃了对其他语言的支持，这有利于更加全面地映射到 Python。为此，需要对系统进行微小的改动，引入以 Python 为中心的语法细节：功能块被标记为定义，赋值块有一个“=”符号，并且添加到列表块的项被重命名为追加。Blockly 还扩展了新的语言功能，包括字典访问和创建。

最终，BlockPy 界面应该提供与 Python 完全同构的映射。但是，首先需要解决一些随之而来的问题。例如，Python 对列表索引和字典访问都使用方括号。为了区分这些访问类型，将块视图分为两种不同类型的块（获取列表的元素而不是从 dict 块获取密钥）。然而，在计算上难以静态地识别给定的一对括号的使用；需要复杂的程序分析技术。

Parson 的问题

Parson 的问题是一种特殊类型的编码练习，其中所有必需的代码块都存在，但这些块之间断开了连接并进行混洗。这些问题通过提供完成问题所需的一切手段来支持初学者，减少开始入门的许多障碍。BlockPy 通过特殊的 Parson 模式支持这些问题，其中顶级块在块模式下进行混洗。

相互语言翻译

这个项目的技术贡献是 BlockPy 和 Python 之间的相互语言翻译。块输出有效的 Python 源代码，可以传递给 Skulpt 来提取 AST 的 JSON 表示。使用我们自己的 Py2Block 库解析此 AST，以生成 BlockPy 可以在块视图中呈现的 XML 表示。图 2 演示了用户的体验。当学生尝试用断开的块转换代码时，生成的 Python 代码将以三重下划线填充。这些下划线（通常是 Python 中的有效变量名称）将触发运行时的错误。

BlockPy 已经支持将它的块编译成 Python、JavaScript、PHP 和 Dart。然而，这种多语言支持导致缩小的同构性——每种语言对于它们的常见操作具有不同的语法，并且不可能创建具有一对一映射的全功能块语言。例如，JavaScript 不支持并行分配——Python 中常用的功能，而 Python 并不

状态管理器

BlockPy 提供了一个状态资源管理器, 用于跟踪程序随时间的执行。状态资源管理器显示有关变量的信息, 并允许用户逐步执行代码(这会影响当前打印 / 绘制的内容), 同时也显示导入的模块以及变量的值和类型。

模型用例

在这里, 我们会举出一些典型的 BlockPy 用例。BlockPy 希望在正式和非正式的情况下同时有用。

独立学习者

学习者独立登录 BlockPy 系统, 并选择使用迭代计算平均值的引导性问题“西雅图天气在华氏 60 度以上吗? 打印是或否。”新手用户不清楚在阅读问题描述后该怎么办。如果用户通过检查西雅图当前的天气并打印字面值来决定作弊, 则系统会智能地注意到用户缺少相关的气象块, 并说明他或她需要将编程决策逻辑与适当的数据源相结合。然后, 用户访问“天气”块获取信息, 但不确定下一步该怎么做。当用户运行程序时, 系统会注意到没有任何 if 语句, 并建议在线教科书中阅读链接的章节。如果用户继续努力整合片段, 系统可以提供越来越详细的提示, 直到他或

她获得成功。

课堂里的一课

导师正在使用 Canvas, 一个具有 LTI 能力的 LMS, 为当天创建一系列的作业。随着学生完成作业, 他们的成绩

示生成的 Python 代码。学生可以直观地看到代码块如何与块相关联, 这有助于他们理解代码不是一系列符号, 而是算法的结构化表示。

BlockPy 以数据科学为主要背景, 故包含与数据处理有关的块。

试点研究

报告给画布, 所以导师可以监控学生的进度, 并帮助那些正在努力完成任务的人。这些信息使导师能够以较早的干预措施来防止学生落伍。导师通过 BlockPy 提供的自动反馈越多, 他们需要关注的简单问题就越少(“您正在检查错误城市的温度”), 以便他们将注意力转向真正遇到困难的学生(“什么是迭代?”)。

一对一辅导

学生正在努力编写索引嵌套字典的必要语法(例如, 一个分为多个级别的犯罪报告, 一个嵌套在暴力犯罪分类下的城市的盗窃率)。学生没有弄清楚如何分析数据的分层结构。导师与学生一起, 通过将库访问块连接到数据块来构建访问数据的表达式, 在每个步骤显

BlockPy 在 2015 年春季有 35 名学生参加了大学级入门计算思维课程试点。这些学生来自各种专业, 包括文科(57%), 建筑学(17%)和科学(15%)。有 20 名女学生(57%)和 15 名男学生。大多数学生没有编程方面的经验, 而不到 17% 的学生在高中学习了计算机科学的课程。学生数量按年龄进行平均分布, 大四学生较多(29%), 大三和大二学生比例相同(26%), 大一新生较少(14%)。

课程内容侧重于教学抽象和算法。虽然编程不是一个主要的学习目标, 但是在讨论更高级别目标的过程中, 这是一个重要的课题。在课程上至三分之一的时候, 学生和 NetLogo 合作(虽然

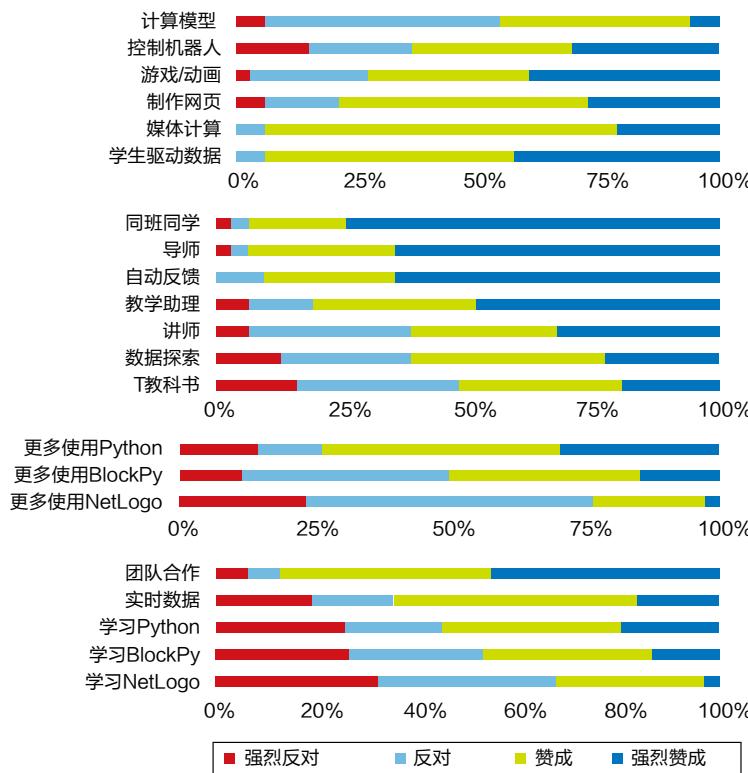


图 3. 关于 BlockPy 的调查问卷中学生们的部分回答。

他们没有编写程序，但是用它来读取代码)，并参加了解释性的活动。然后，使用 BlockPy 向学生介绍 Python，他们在完成指导实践问题时花了大约六个课时。接下来的两个课时致力于使用常规 Python 环境 (Spyder) 来完成小型编程分配（类似于使用 BlockPy 完成的任务）。最后，学生们使用八个课时，在 Spyder 的完成各自的大作业。

方法论

通过两次调查收集了学生对 BlockPy 的反馈，一个在 BlockPy 部分之后给出，另一个在课程结束时给出。这项调查是由 Likert 四点问题和开放的定性问题组成的。图 3 显示了一些特别有趣的结果。这项研究的所有结论应该被认为是初步的，因为它使用的是第

一个版本的 BlockPy。

对 BlockPy 的看法

第一个调查问题询问，学生希望在课程中有更多的时间使用哪个编程环境：NetLogo、BlockPy 还是 Spyder。请注意，BlockPy 被称为 Blockly，Spyder 环境被称为 Python。结果表明，相比 NetLogo，学生们认为他们在 BlockPy 的体验更具有价值，但大多数人认为他们得到足够的 Python 体验。这些结论是由高质量的数据支持的，有些学生说“更多的 Blockly，少一些 Python”，其他人则要求“更多 Blockly and more Python”。

BlockPy 用法

在使用 BlockPy 的六天中，学生

被赋予了 40 个课堂问题，19 个作业问题。学生每个问题平均运行四次代码（标准差 1.8）。

学生们被问及从 Blockly 到 Spyder 过渡的阶段是否成功。百分之六十五的学生表示同意或强烈同意，这表明还是有相当大的一部分同学在过渡期感到不适应。相互语言翻译的原始设计是同时具有并排的块和文字视图。然而，日志分析显示，大多数学生没有利用该功能。只有五名学生（约占 15%）使用了转换功能，而且一直使用得更少。学生们可能会随着代码的改变而进行观察，但是他们并没有写文字代码。很难说为什么学生没有利用这个功能。我们目前的假设是，学生们被界面所困惑，而这个操作需要手动进行从文本到块的转化。在我们新版本的 BlockPy 中，转换会自动发生，只需切换选项卡即可进行切换。初步数据表明，这种新界面使得学生过渡期的体验得到很大的改善。

我们同时也通过调查，向学生了解最有助于学习的内容。BlockPy 中给出的自动反馈与学生的同学和导师给出的反馈相当，这体现了系统具有的强大价值。尽管状态浏览器广受好评，但使用它的学生却相对较少（11 名学生，约 31%）。由于超过百分之五十的课程报告在数据浏览器中实现，所以即使他们自己没有利用这些工具，学生也可以从

使用引导工具的导师处获益。

数据科学背景

最后,我们就课程对学生们长期职业目标的影响,以及他们对入门计算课程的兴趣进行了调查。这些调查的内容都经过了简化——例如,媒体计算环境被列为“使用图片、声音和电影”。

结果表明,学生认为数据科学是很吸引人的——这个结果令人惊奇,因为大多数学生在不同编程环境中的经验可以忽略不计。然而,我们的初步结果表明,这是一个值得深入探讨的方法。

未来的工作

BlockPy 是一个正在不断发展的项目。有许多新的功能正在开发中,旨在扩展对 Python 的支持以及针对导师的指导反馈 API,例如利用更多的静态 / 动态推理技术来改进块渲染和错误报告。

基于块的环境的最大价值之一是它可以揭示 API 的广度,这大大地减少了学生对文档的依赖性。然而,这也可能是一个缺点,因为学生可能被界面的功能所迷惑。因此,一个开放的研究问题是决定以何种频率去展示语言特征。

游戏和动画设计作为引导性背景的主要优点之一是具有具体的抽象概念。需要进一步的分析来确定使用不同

关于作者

奥斯汀·科瑞·巴特是弗吉尼亚理工大学访问助理教授。他的研究方向包括计算机科学教育、软件工程和程序分析。Bart在弗吉尼亚理工学院获得计算机科学和科学学博士学位。请通过acbart@vt.edu与他联系。

贾维尔·提巴乌是弗吉尼亚理工学院计算机科学系的博士生,本文中的研究在该系进行。他还是ESPOL计算机科学助理教授。他的研究方向包括人机交互和计算机科学教育。Tibau在加泰罗尼亚大学获得计算机科学硕士学位。联系方式: jtibau@vt.edu。

艾利·蒂勒维奇是弗吉尼亚理工学院计算机科学系的副教授。他的研究方向包括分布式和移动计算和计算机科学教育的软件工程。Tilevich在Georgia Tech获得计算机科学博士学位。他是IEEE的高级会员。联系方式: tilevich@vt.edu。

克利福德·A·沙弗尔是弗吉尼亚理工学院计算机科学教授。他的研究方向包括计算生物学,特别是用于指定模型和计算的用户界面——算法可视化,以及计算机科学教育。他在马里兰大学获得计算机科学博士学位。联系方式: shaffer@vt.edu。

丹尼斯·卡福拉是弗吉尼亚理工大学计算机科学教授。他是两个荣获NSF IUSE奖项的主要参与者,他参与了使用BlockPy环境在大学开发计算思维的普通教育课程。Kafura的研究方向包括计算机科学教育,操作系统和软件工程。他从普渡大学获得计算机科学博士学位。联系方式: kafura@vt.edu。

上下文的必要性。BlockPy可以通过

乌龟图形和媒体计算库。

支持替代上下文来解决这些问题,例如

由于参加实验的学生数量较少,我

们的试点研究难以得出关于 BlockPy 的演变性质的结论性结果。进行中研究的初步结果表明，最近的改进已经克服了环境的一些限制，使得用户反馈大大改善。我们在最新的迭代中收集更多的数据，同时也在对已记录的学生代码进行后续的研究。我们希望 BlockPy 的用户群不断增加，为研究人员提供更多样本和更有意义的数据。

BlockPy 正在以按需的方式进行开发，由即时课程需求驱动，但这一点做得仍然不够。例如，块接口不支持许多高级 Python 功能，例如用于编写面向对象类的接口。这并不意味着学生不能编写具有类或其他高级功能的程序，因为 Python 代码使用这些功能在 BlockPy 中呈现为嵌入式文本块，并通过 Skulpt 正常执行。支持这些功能没有技术障碍——这个过程只受时间和学生兴趣的限制。

我们认为 BlockPy 是一个新入门学习者的范例，将交互式支持与成熟的编程途径相结合。通过在数据科学的背景下进行教学，BlockPy 可以提供真实性的体验，这使得它有助于学生向更严格的编程环境进行过渡。使用 BlockPy 的研究将有助于回答有关数据科学和数据块价值的关键问题。我们的希望是，BlockPy 的开放性质可以鼓励来自不同

领域的学习者从事计算，从而带领更多的人走向计算机科学的美好未来。■

致谢

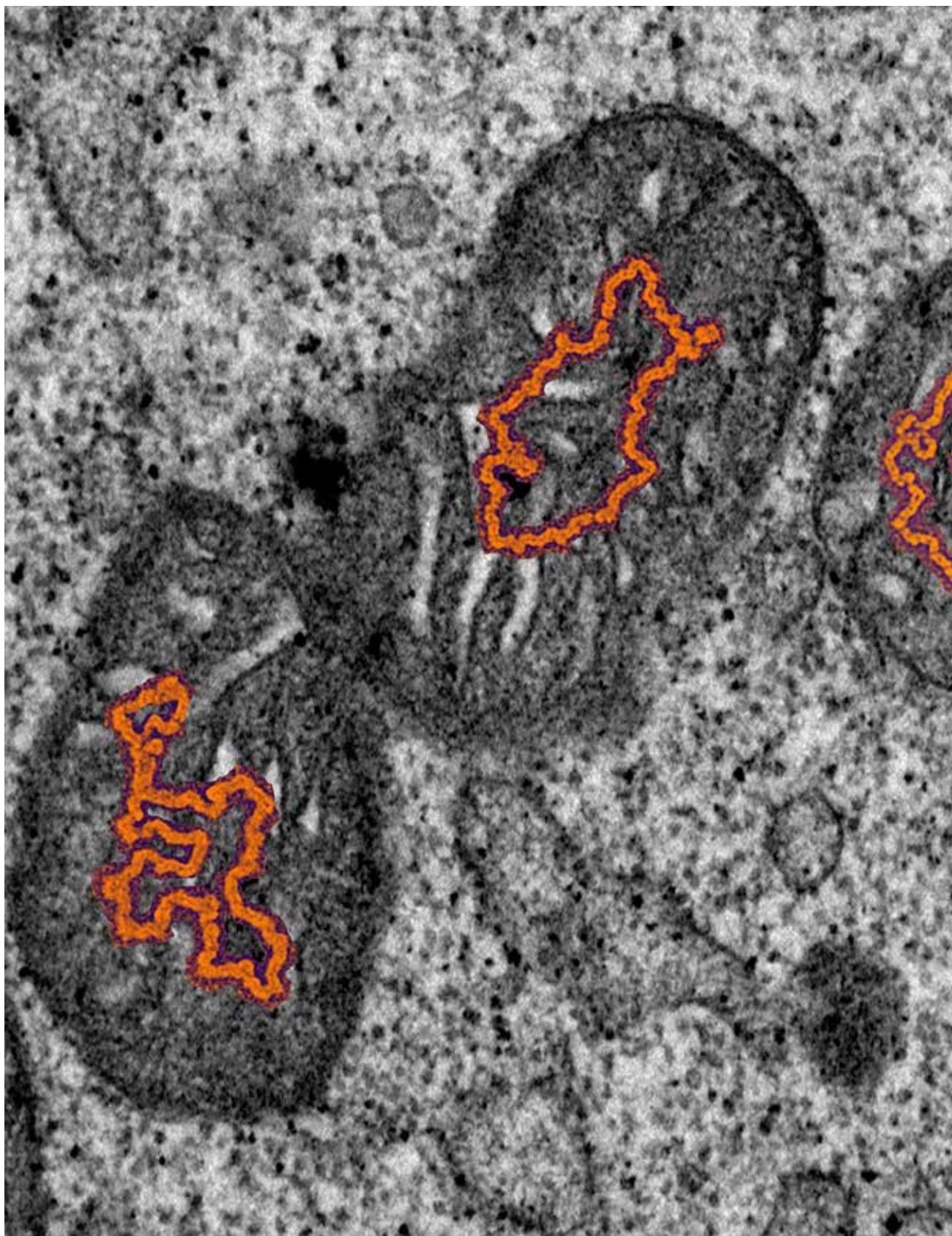
我们非常感谢国家科学基金会 DGE-0822220、DUE-1444094 和 DUE-1624320 的资助。

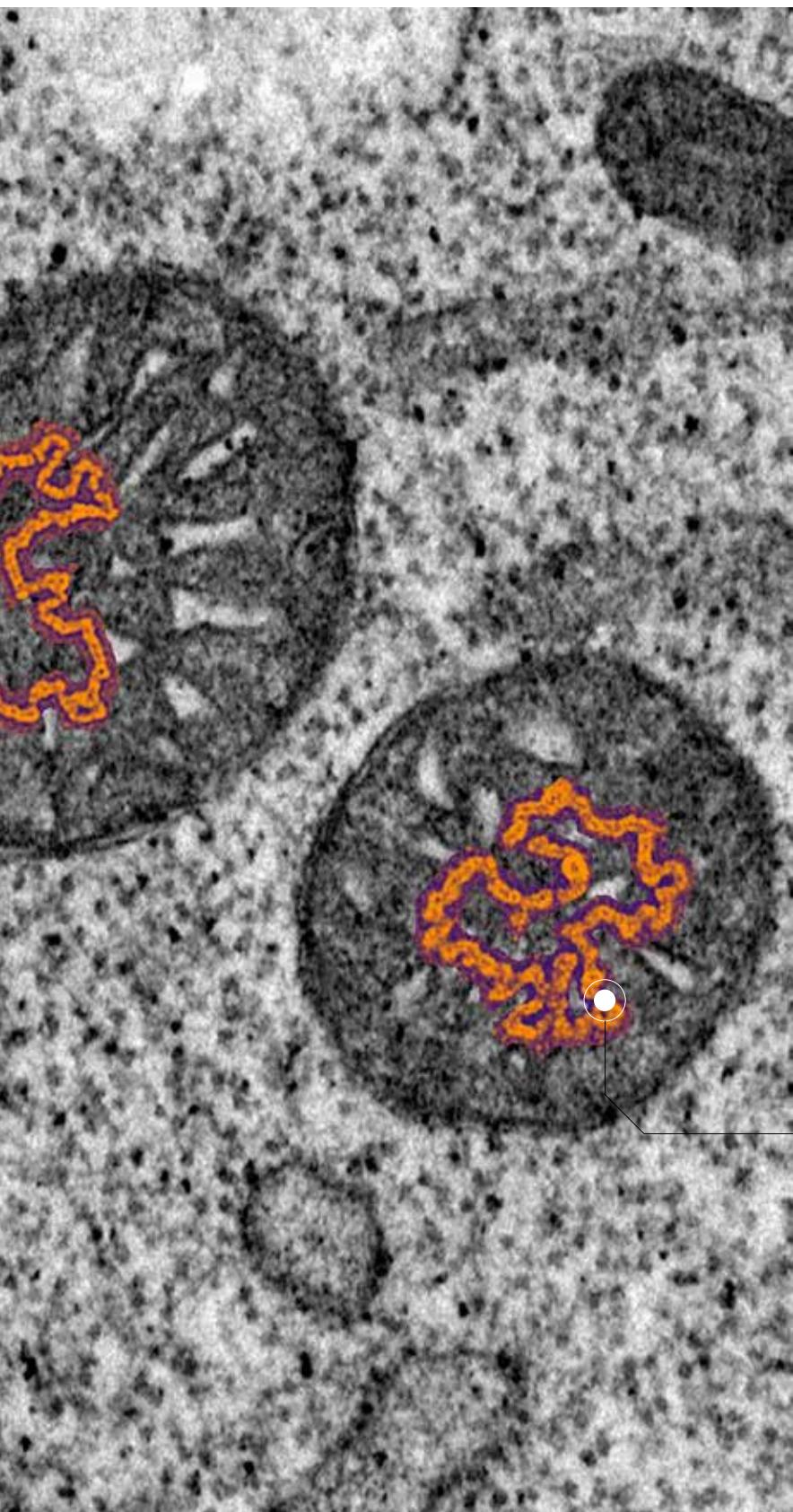
参考文献

1. B.D. Jones, “Motivating Students to Engage in Learning: The MUSICModel of Academic Motivation,” Int’l J. Teaching and Learning in Higher Education, vol. , no. , pp.-..
2. M. Guzdial and A.E. Tew, “Imagineering Inauthentic Legitimate Peripheral Participation: An Instructional Design Approach for Motivating Computing Education,” Proc. n d Int’l Workshop Computing Education Research (ICER 58), 58, pp. -.
3. A.C. Bart et al., “Computing with CORGIS: Diverse, Real-World Datasets for Introductory Computing,” Proc. th ACM Technical Symp. ComputerScience Education (SIGCSE), pp. -58.
4. P. Guo, “Python Is Now the MostPopular Introductory Teaching Languageat Top US Universities,” blog, Comm. ACM, Jul. 2014; c ACM.org/blogs/blog-c ACM/58-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext.
5. A. Ko, “Programming Languages Are the Least Usable, but Most Powerful HumanComputer Interfaces EverInvented,” blog, Mar. 2014; blogs.uw.edu/ajko/2014//programming-languages.
6. T.W. Price and T. Barnes, “Comparing Textual and Block Interfaces in a Novice Programming Environment,” Proc. th Ann. Int’l Conf. Int’l ComputingEducation Research (ICER), , pp. - .
7. D. Weintrop and U. Wilensky, “Using Commutative Assessments to Compare Conceptual Understandingin Blocks-Based and Text-Based Programs,” Proc. th Ann.

- Int'l Conf.on Int'l Computing Education Research(ICER), , pp.-.
8. T. Tang, S. Rixner, and J. Warren, "AnEnvironment for Learning InteractiveProgramming," Proc. th ACM Technical Symp. Computer Science Education(SIGCSE 14), 2014, pp. 59–5959.
9. S.H. Edwards, D.S. Tilden, and A.Allevato, "Pythy: Improving theIntroductory Python ProgrammingExperience," Proc. th ACM TechnicalSymp. Computer Science
- Education(SIGCSE 14), 2014, pp. 641–646.
- 10.P. Guo, "Online Python Tutor:Embeddable Web-Based ProgramVisualization for CS Education," Proc. th ACM Technical Symp. ComputerScience Education (SIGCSE 13),2013 pp. 579-584.
- 11.D. Bau, M. Dawson, and A. Bau, "Using Pencil Code to Bridge the GapBetween Visual and Text-Based Coding(Abstract Only)," Proc. 46th ACM Technical Symp. Computer Science
- Education(SIGCSE), , p. 59.
- 12.Y. Matsuzawa et al., "LanguageMigration in Non-CS IntroductoryProgramming Through Mutual LanguageTranslation Environment," Proc. 46th ACM Technical Symp. ComputerScience Education (SIGCSE),, pp. – .A.C. Bart, "Situating Computational Thinking with Big Data: Pedagogyand Technology (Abstract Only)," Proc. 46th ACM Technical Symp. ComputerScience



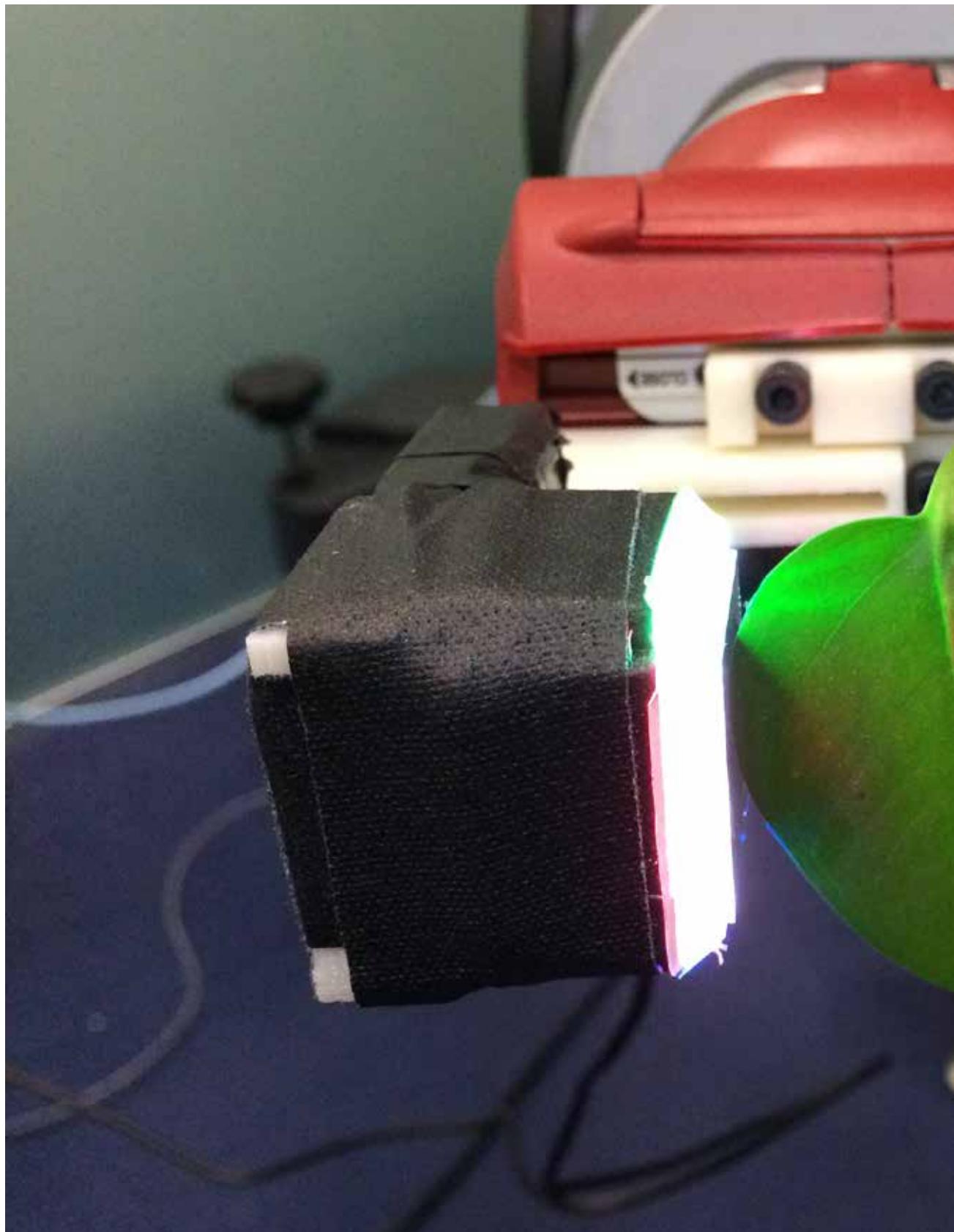




诊断好帮手

密歇根大学的生物化学家 Laurie Kaguni 开发了一套数据库系统，旨在更好地转化医学研究信息。这套系统名为 POLG 致病性预测服务器 (POLG Pathogenicity Prediction Server)，内含 681 名匿名的线粒体疾病患者的数据，包括他们的诊断年龄和症状描述。

图片来源：Cristina Nadalutti 和 Jack Griffith





触觉机器人

借助机器学习和新型高精度传感器的帮助，往日冷冰冰的机器人也有了精确而高效的触觉。

图片来源：Ruili.io



计算机教育改革的推手： 计算机科学部门和教师

文 | W·理查兹·阿德里安 (W. Richards Adrián)，马萨诸塞大学阿姆赫斯特分校 (University of Massachusetts Amherst)

译 | 刘大明

最近，有关计算机科学部门在“全民计算机科学计划”中的作用进行了小组讨论，尤其讨论了计算机科学的教职员为此做出的贡献，¹ 并对更广泛的社区兴趣活动提出了一些建议。

由奥巴马政府于2016年推出的“美国全民计算机科学计划”的目的是：……让所有美国学生从幼儿园到高中都能够学习计算机科学，并培养他们具备作为数字化经济的创造者（而不仅仅是消费者）所需的计算思维技能，并成为由技术驱动的世界的活跃公民。²

在芝加哥、纽约、旧金山，阿肯

色州、华盛顿州、犹他州、罗德岛州和其他几个城市和州，正在对该计划进行规划和实施。

这一计划为所有人提供了多种方法和选择，但在各州的教育机构（如教育部）或地方教育机构（如学区）很少拥有具备计算机科学专业知识的管理人员或顾问。目前，关于在计算机教育中向孩子教些什么、由谁来教

的问题，各个州正在解决过程中。这将改变进入高等教育计算机科学专业的学生的经验、准备情况和多样性。计算机科学教师可以提供有价值的专业知识，作为颁发教师执照、职前教师教育的标准，同时还可以加深所有的学生对计算机科学和计算思维概念、技能和实践的基本了解，在他们选择大学和职业时提供帮助。请记住，正如著名的科学教育顾问和研究人员 Megaran Garvin 所说的那样：“与决策者交谈时，谈及更广泛的选民时，影响决策者的可能性更大。”

标准

计算机科学的教职员如何才能为制定 K-12 计算机科学的标准做出贡

献呢？这些标准会影响所有学校，而不仅仅是资源丰富的地区，高效的国家战略有助于扩大民众的参与度和包容性，拓宽从学校到大学走向职业生涯的渠道。^{3,4}设计精良的标准有助于吸引更多准备就读计算机科学专业的学生，更能够让其他各个领域中具备实用计算机知识的非计算机专业学生进入大学。目前，国家和地方的教育工作人员、学校、校领导、老师、辅导员、家长和学生都很难理解计算机科学和计算思维。标准一旦订立就不会经常修改，因此计算机科学标准必须具备前瞻性，必须处于在新技术和应用的前沿。最后，制定标准有助于确定颁发教师执照所需的知识，培养出素质更高的教师，让更多的学生前来就读。

地方上的参与

有了标准之后，接下来要做些什么？我们需要让各个地区采用这一标准、制定课程，提高目前在职教师的专业技能，并制定教师执照制度和职前教师课程。每个州都有独特的限制条件和机会，成功推动全民计算机教育事业的策略必须解决这些问题。⁵比如阿肯色州州长阿萨·哈奇森（Asa Hutchinson）开始将计算机列入该州每个学校的课程，为此他成立了一个行政办公室，支持专业上的培训，并大力推广这一计划。相比之下，马萨诸塞州是则更为本地化，教育上的变革需要逐个学区地推行，很多时候需

要逐个学校地推行。

国家和地方教育部门需要做出许多决定和选择，无论计算机科学教育是在国家还是地方处理，都需要计算机教师的投入，包括如何提高教师的能力和教学效果（专业提升、正式课程、实习和共同教学）；用什么方式授课（独立式、综合式或二者结合）；如何选择课程（一站式方案还是 DIY）；哪种类型的老师适合教授计算机科学（是数学家/科学家、职业技术人员还是计算机专家）。

通常，计算机科学的教师会在本地学区、夏令营以及学校内举行一系列讲座或活动，作为校园课程或课外活动的一部分。这虽然令人钦佩，但这些授课途径通常是不可持续的。我们必须记住，“全民计算机科学”的目标需要得到各地区和学校的长期支持，才能制定课程、聘请和支持教师、鼓励学生，并将注意力集中在计算机科学的授课方式上。要将“全民计算机科学教育”活动推向前进，教师应作为计算机教育改革中的顾问，与大批利益相关人士进行合作。

随着招生人数的迅速增长，计算机科学部门和教职员为什么应该对“全民计算机教育”计划感兴趣呢？如果这个行动获得了成功，未来进入计算机专业学习的学生可能会具备更多的经验，准备也更为充分，我们可能会面对更加多元化的入学申请者，还会有更多的非计算机专业学生选修计算机课程。要在 K-12 学校中推广计算机科学课程，让所有学生都有机

会修读计算机课程，并鼓励他们这样做，可能尚需时日。高等教育面临的挑战是，如何以最优的方式管理入门课程，这些课程面向的学生在计算机科学方面具备的经验可能有深有浅，有些学生可能没有经验。教育部门可能会更愿意招收来自“全民计算机教育”中表现出色的学校的学生，这可能对生源的多样性和包容性产生影响。此外，如果 K-12 老师的准备工作没有跟上，可能会造成招录的学生对计算机课程准备不足，或者出于错误的原因学习这些课程。

入学要求

最近发布的计算研究协会（CRA）报告显示，为计算机科学专业做出更多的限制性入学要求，并没有像有些人担心的那样，对生源的多样性产生显著的负面影响。^{3,4}但是，机构多样性的增加，与这些机构是否通过特别措施努力招募和留住妇女和代表性不足的群体有关。

计算机科学院系应该在大学预修课程（AP）学分和录取权重上采取行动。将高中计算机课程纳入大学学分体系和招生标准，可以提高学生及其家长对 K-12 计算机科学的兴趣。通常，在 AP 计算机科学类考试“A”（AP CS-A）中获得高分（3-5 分）应予折抵大学学分，但不应作为录取要求（比如将其算作最低学历）。同样，在 AP 计算机科学原理（AP CSP）考试中的获得高分也可能抵扣学分，但是应将

其视作计算机科学学位、录取要求，还是一种普通教育的要求？各部门应与教育机构合作，向大学理事会提供关于 AP CS-A 和 AP CSP 考试的学分折抵和预修政策的信息。

K-12 计算机科学教育的发展与外延

由于计算机教师资格认证变得越来越普遍（全国 29 个州可以颁发某种形式的教师执照、认可或认证），高等教育教师预备（“预修”）计划需要修改和扩大。计算机科学院系与教育学院合作推广联合计划，有助于让计算机教育融入教师的教育体系。

而在参与 K-12 教育机构的学生、家长、教师和管理人员中，有越来越多的人认识到，计算机科学的相关职业和部门仍然有充分的理由继续与学校和社区进行沟通，并支持在当地的课外活动。在“全民计算机教育”全体会议期间，佐治亚理工大学计算外延部门的主管芭芭拉·埃里克森（Barbara Ericson）讨论了成功案例，以提高年轻妇女和代表性不足的少数群体中 AP CS-A 考试的合格率。

计算机科学教育研究

早在 20 世纪 60 年代，教育研究的先驱 Patrick Suppes 和 Richard C. Atkinson 就开始研究如何使用计算机来提升教学和学习的效果。到了 20 世纪 70 年代，随着这种教学方式的逐

步巩固，包括 John Seely Brown 和 Seymour Papert 在内的研究人员提出了智能计算机辅助教学（CAI）的思想。这些集体努力已经将这一思想体现在许多重要方向上：对智能导师的研究，创建面向的教育程序设计语言 LOGO，从提升技术能力转向流畅运用信息技术，以及提倡计算思维等。今天，计算机科学部门需要找到一个学习科学和计算机教育研究的地方，并像研究传统计算机科学一样，重视这些研究的价值。

作为一个具有广泛专业知识、可能在“全民计算机教育”中获得（或失去）大量机会的群体，计算机科学部门和教师都有责任积极推动计算机教育和政策的改革。计算机科学部门需要认识到教师与学校、地方进行合作的努力，并对此作出奖励，在全国范围内组织“全民计算机教育”活动，对政策的制定造成影响。■

参考文献

1. M. Guzdial et al., “The Role of CS Departments in the US President’s ‘CS for All’ Initiative,” Proc. 2017 ACM SIGCSE Technical Symp. Computer Science Education (SIGCSE 17), 2017, pp. 663–664.
2. M. Smith, “Computer Science for All,” Obama White House blog, 2016; obamawhitehouse.archives.gov /blog/2016/01/30/computer-science-all.
3. “Generation CS: Computer Science

Undergraduate Enrollments Surge Since 2006,” Computing Research Assoc., 2017.

4. W. DuBow et al., “Increasing Diversity in the Face of Enrollment Increases,” Proc. 2017 ACM SIGCSE Technical Symp. Computer Science (SIGSE 17), 2017, pp. 655–656.
5. B. Ericson et al., “State-Based Progress towards Computer Science for All,” ACM Inroads, vol. 7, no. 4, 2016, pp. 57–60.
6. “Arkansas Governor’s CS Initiative,” website; governor.arkansas.gov /promises/computer-science.
7. “AP Higher Education: Setting Your Policies,” The College Board, 2017; aphighered.collegeboard.org /setting-credit-placement-policy.
8. S. Cooper, Future Directions in Computing Education Summit, Part Two: Institutional Challenges to Supporting and Growing Computing Education Research, tech. report, Stanford InfoLab, 2014.

W·理查兹·阿德里安是马萨诸塞大学阿姆赫斯特分校信息与计算机科学学院名誉教授。联系方式：adrion@cs.umass.edu。



会议就在你的手中

IEEE计算机协会的会议发布服务（CPS）现在可以提供组织会议的移动应用了！让会议的日程、会议信息和论文列表在你的与会者手中的设备上显示。

会议的移动应用可在[安卓](#)设备、[iPhone](#)、[iPad](#)和[Kindle Fire](#)上运行。



欲知更多信息，请联系cps@computer.org





ENIAC 的第二生命

文 | 汉斯·诺伊康 (Hans Neukom)

译 | 丁家琦

世界上第一台通用电子计算机是如何诞生的，又是如何再次焕发活力的？



要是对计算机历史感兴趣的，都多多少少了解一些 ENIAC 在二战期间

由宾夕法尼亚大学开发出来的过程。我们熟悉它的开发者，J·普雷斯佩尔·埃克托 (J. Presper Eckert) 和约翰·毛赫利 (John Mauchly)，也很容易就能回想起老照片里 ENIAC 的样子：由一堆一堆的接线连接起来的一个个橱柜。ENIAC 在一开始只能通过插拔接线来进行编程，这在某种意义上可算不上真正的计算机。鲜有人知的是，战后，它又经历了一次“复兴”，终于能像现代计算机一样被编程了。这篇文章介绍的就是在二战之后，它最初的发明者毛赫利与约翰·冯·诺依曼都转向开发新的 EDVAC 之后，ENIAC 的团队为了提高它的操作效率所做的新的工作。他们找到了一些新方法，废除了插拔接线这一繁琐至极

的过程，同时还给 ENIAC 赋予了解决新问题的新能力。

反思 ENIAC 的设计

ENIAC 的原始设计完成于 1944 年 1 月，到 1944 年 6 月时，它已经初步成型，包含了两个累加器、电源，以及信号发生器，可以成功地进行一些初步计算。完整的 ENIAC 于 1945 年春天在宾大的摩尔电气工程学院投入使用，起初是为美国军方服务。在二战期间，这个设备是保密的，直到 1946 年 2 月 14 日才第一次公开亮相。

战争期间，美军急切地等待着 ENIAC 能提高运算速度，以满足战争中多弹道轨迹表计算的需求。因为军方的需求很紧迫，毛赫利和埃克托也没时间尝试其他的概念设计，因此，毛赫利

使用了一种简单的静态设计，让 ENIAC 采取同当时的人力计算类似的方法：首先把数学问题分解成最基本的加法或者减法，再让大量专门从事计算的人每人拿着机械计算器计算单个步骤，不断将中间结果传给下一个人力计算器，直到循环到足够的次数，可以得到原始问题一个令人满意的结果。

因为电子计算器，也即累加器 (accumulator) 的速度比人力计算快得太多，因此毛赫利在 ENIAC 中只用了 20 个累加器，它们之间通过总线连接起来，以将每一步计算的结果传送给下一个累加器。这种方法的好处在于，数学家只需沿用之前把复杂问题拆解成简单加减法迭代的思路就好，而 20 个高度准确又快速的并行累加器也比上百个可能出错的人力计算器更好控制。

要定义 ENIAC 的操作，就要设置单元之间的相互连接，而这些连接是通过可插拔的电线来实现的。“程序员”

们通过连成菊花链状的线所组成的编程盘与数字盘来形成 ENIAC 的三条连接线(总线): 数字总线用来传输带正负符号的 10 位十进制数, 编程总线用来引导各个单元以事先确定的顺序进行操作, 而同步总线则用来提供时钟脉冲, 以统一所有单元的操作进程。每个单元都有自己的编程控制开关, 可以手动调节, 以决定从哪里开始读取输入数据, 把结果输出到哪里。编程者也会使用这些开关来确定要重复多少次该单元的操作。

图 1 展示了 ENIAC 的数字盘、编程盘, 以及两位正在插接线以设定 ENIAC 完成一项任务的编程人员。

ENIAC 开发团队马上就意识到, 每次要解决新的问题时都得重新插拔一下这一大堆接线实在太麻烦了。当时在摩尔学院的美军联络官员赫尔曼·H·戈德斯坦(Herman H. Goldstine)就写到:

“到 1944 年, 摩尔学院的研究组都达成了共识, 认为对数字计算机的指令应当能够以数值的形式存储在纸带上。”他们重新综合分析了 ENIAC 的特性, 得出结论: ENIAC 不适合实行比计算多弹道轨迹表更一般性的任务。然而, 由于美军的需求极为紧迫, 此时已不容许研究组在 ENIAC 建造过程的半途更改设计。戈德斯坦由此向他的上级提出一项新的研发计划: 研制出一台比 ENIAC 更新、更好的计算机——这就是后来我们所说的 EDVAC。

1944 年夏天, 戈德斯坦在普林斯顿高等研究院偶遇了约翰·冯·诺伊曼, 并把 ENIAC 计划告诉了他。冯·诺伊曼

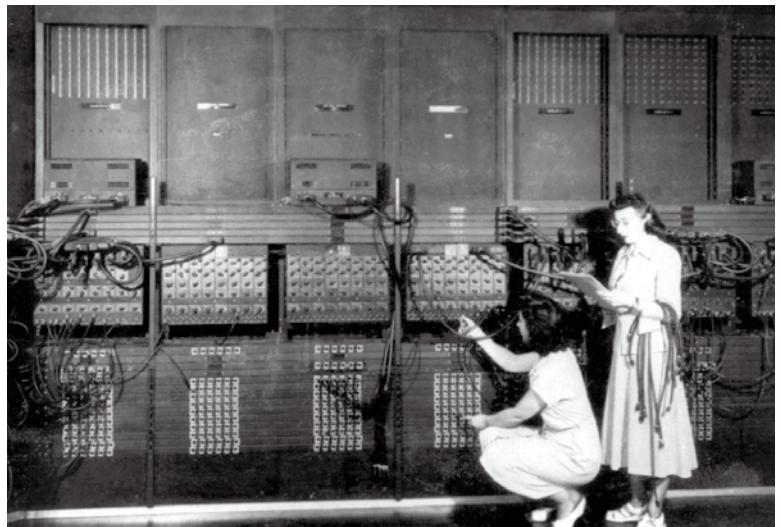


图 1. ENIAC 累加器的样貌。站着的程序员是爱斯特尔·格斯顿(Ester Gerston), 蹲着的是格洛丽亚·露丝·戈登(Gloria Ruth Gorden)。

立即产生了兴趣, 他参观了位于费城的摩尔学院, 并与已经存在的 ENIAC 新设计小组开展合作。冯·诺伊曼作为著名的 EDVAC 方案初稿作者而知名, 该方案描述了 EDVAC 的逻辑概念和结构, 这就是后来的冯·诺伊曼计算机结构。然而, 该方案中的所有想法并非冯·诺伊曼一人所提出, 他得到了 ENIAC 小组几位成员的帮助, 然而事到如今已经无法界定谁最初提出明确的想法了。戴维·艾伦格里尔(David Alan Grier)表示, J·普雷斯佩尔·埃克托、约翰·毛赫利、阿瑟·伯克斯(Arthur Burks)和赫尔曼·H·戈德斯坦对此都有贡献。例如, 戈德斯坦提到, 埃克托和毛赫利两人描述了用一种延迟线(delay line)来代替 ENIAC 的 20 个累加器, 以作为更大容量的存储器。此外, 埃克托 1944 年 1 月 29 日所写的三页备忘录表明, 他已经提出使用磁盘或磁鼓作为计算机存储介质的想法, 虽

然他并没有提出将数据存储与控制指令存储分开。他在备忘录(后来由赫尔曼·卢克夫出版)中清清楚楚地写道:

如果使用多轴系统(作者注: 即磁盘或磁鼓), 可用的设备数量将会大大增加, 设备及其运行过程的自动编程能力也将大大提升, 因为这类存储设备提供了更长的时间标度……这类编程可以通过在合金磁盘上以临时方式, 或是在刻蚀磁盘上以永久的方式来实现。

卢科夫与尼古拉斯·梅特罗波利斯(Nicholas Metropolis)和杰克·沃尔顿(Jack Worlton)都引用这段文字来表明戈德斯坦是第一个提出将数据和编程指令放在同一个存储介质上的人。因此, 可以说, 在冯·诺伊曼作为顾问加入

他们之前, ENIAC 的设计者心目中就已经有了存储程序计算机的基本理念了。

而现在, 问题显然就在于, 能否在不完全重建的前提下, 把 ENIAC 变成一台存储程序计算机。戈德斯坦在 1944 年 9 月 2 日写给弹道研究实验室副主任保罗·N·吉利翁 (Paul N. Gillion) 的一封信中提到, 他想采用机械继电器和电磁电话交换机, 通过电传打字带来阅读指令, 用机械而非人工的方式来操作 ENIAC 的开关和控制器。然而, 戈德斯坦的想法未曾得到实施, ENIAC 仍然按照原先的设计继续建造了。

在 1946 年 7 月到 8 月, 摩尔学院举办了一系列讲座(也就是著名的摩尔学院讲座), 内容包括最先进的电子计算机技术。其中几次讲座是包括 ENIAC 的, 但丝毫没有提到将 ENIAC 改造成存储程序计算机的想法。

1967 年, 毛赫利在写给莫里斯·V·威尔克斯 (Maurice V. Wilkes) 的信中宣称, 在冯·诺伊曼加入之前, ENIAC 的设计就已经能行使存储程序计算机的功能了。他还说:

原先的设计需要从转换装置来发射程序脉冲, 但后来加入了两块新的面板之后就可以触发更多的程序了。冯·诺伊曼直接的工作就是设计了一种特殊的 2 位指令码, 后来成为了 ENIAC 的“标准”冯·诺伊曼码。

毛赫利的这封信里没写日期, 但我们都知道冯·诺伊曼在 1947 年春天已经

意识到, ENIAC 可以被改造翻新成一台存储程序计算机, 两位相关人士都曾提到这一事实: 戈德斯坦和弹道研究实验室下属计算实验室主任理查德·F·克利平格 (Richard F. Clippinger)。戈德斯坦提到冯·诺伊曼提议把 ENIAC 当做一台原始的存储程序计算机来构造并操作, 建立一个所谓的“ENIAC 逻辑编程系统”, 并把这个任务交给了赫尔曼·戈德斯坦的妻子阿黛尔·戈德斯坦 (Adele Goldstein) 来完成, 她是当时的 ENIAC 程序员之一。

在那时, 冯·诺伊曼和戈德斯坦都在写一份名为“电子计算设备的设计和编程问题”的报告, 因此, 冯·诺伊曼的想法跟 ENIAC 有关也就不足为奇了。这个问题后来被交给克利平格来实现, 1948 年秋天, 最初的程序指令列表终于完成, 据戈德斯坦的记录, 1948 年 9 月 16 日, ENIAC 的新编码系统开始运作(与克利平格的记录相一致)。

ENIAC 焕然新生以后解决的第一个问题是首次蒙特卡罗模拟, 由数学家尼古拉斯·梅特罗波利斯 (Nicholas Metropolis) 和冯·诺伊曼所提出。与他们一同工作的数学家罗伯特·里克特迈耶 (Robert D. Richtmyer) 将此时的 ENIAC 描述为“首台拥有词汇量的机器”。

ENIAC 改造之后所运行的这一过程在我们今天被叫做仿真 (emulation), 它的速度比原先慢了六分之一(为什么会慢我们在后面会解释), 但它再也没有用回原先的工作方式。这一新的运作方式从 1946 年秋天开始, 一直沿用到 1955 年 10 月 2 日, 即它寿终正寝, 被

永久关闭的日子。

ENIAC 的逻辑编码系统

为了更好地了解 ENIAC 重建后的新功能, 需要深度研究其原始架构、编程过程, 以及新的特征。

ENIAC 的架构

ENIAC 填满了一整个 5×10 米的房间, 它包含 40 块电子面板, 沿三面墙摆放成 U 形(见图 2)。此外, 它还包含 3 个可移动的函数存储单元、一个 IBM 读卡机以及一个 IBM 卡纸打孔机。

此外, ENIAC 还包含了一个启动单元、一个循环单元、一个主程序控制单元、20 个累加器、一个高速乘法器, 以及一个除法与平方根计算单元。图 3 展示了 ENIAC 的逻辑框图。

ENIAC 中的 20 个累加器既是算术单元, 也是存储单元。ENIAC 的计算对象是带正负号的 10 位十进制数, 每个累加器都可以存储一个这样的数。有些累加器与乘法器和除法与平方根计算单元相连, 以存放输入值用于运算。

此外, 还有 3 个函数存储单元可以用来存储数据——只读存储器, 可以存放 104 个带正负号的 20 位十进制数(12 位可变, 8 位不变), 这些数字需要手动拨动开关来设置。原本功能存储单元是用来存储数学函数, 例如正弦函数或余弦函数, 以便于运算的, 但它们也可以用来存储在运算中一切需要存储的常数。

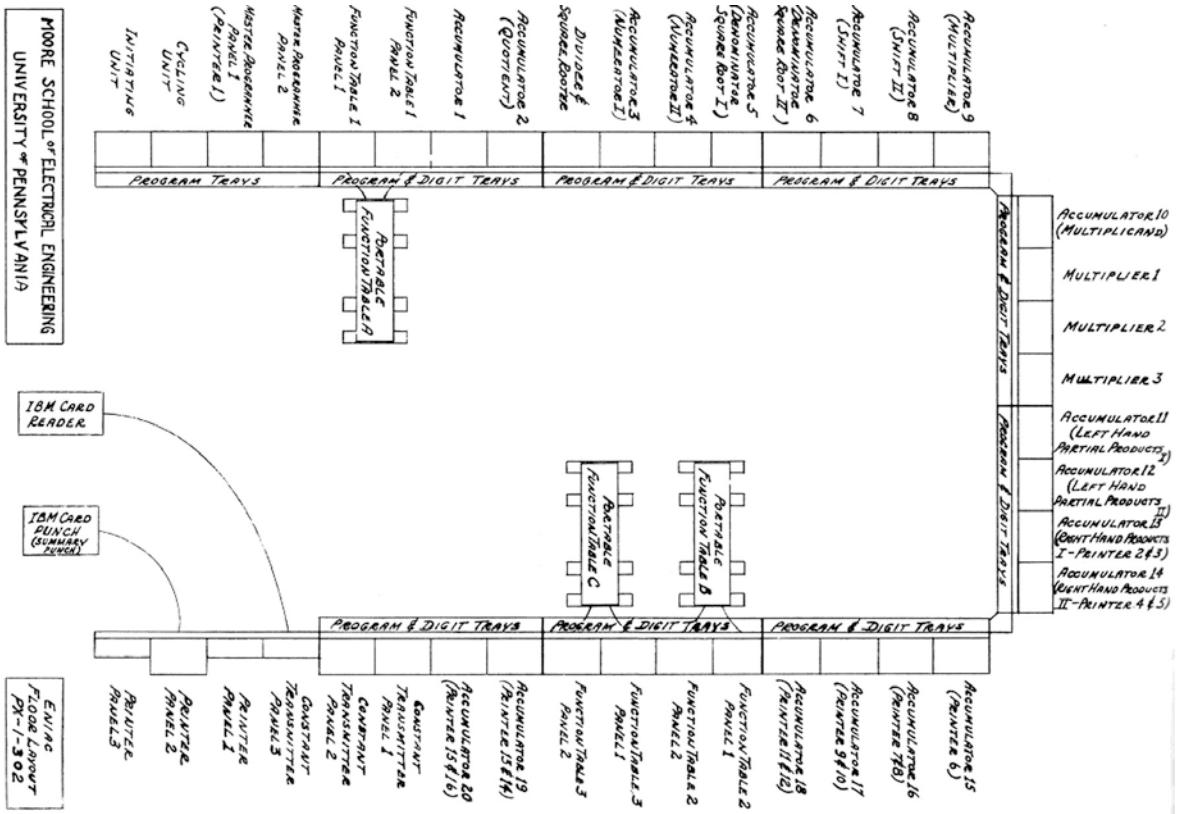


图 2. 费城宾夕法尼亚大学摩尔学院，ENIAC 楼层布局。

ENIAC Block Diagram

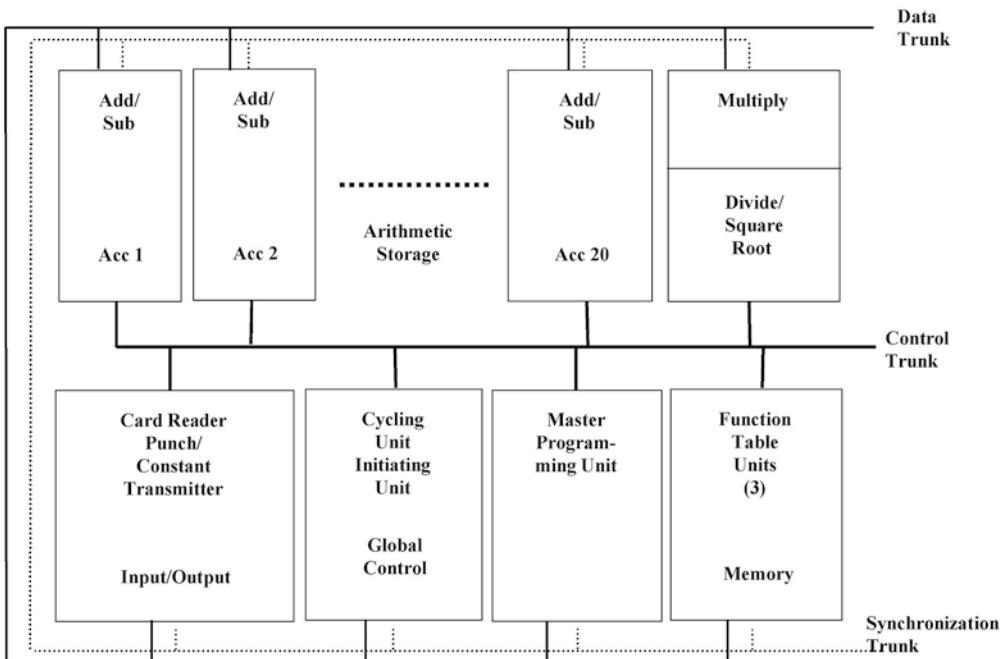


图 3. 作者根据《ENIAC：历史、运算与改造》中的图 4 重新绘制的 ENIAC 框图。

数据输入和输出则通过 IBM 打孔卡设备来完成。一张卡纸可以存储 80 位数字和 16 个符号，也就是说可以存储 8 个带正负号的 10 位十进制数或 16 个带正负号的五进制数。读卡机可以使用继电器将从卡纸上读出的数字存储在常数发送器上，计算结果则用打卡机打在类似的卡纸上。利用输出的打孔卡纸可以用离线打印机打印出结果。

启动单元、循环单元和主程序控制单元控制着 ENIAC 的运算。启动单元控制所有单元的通断电，同时负责清空累加器，让所有单元进入设定的状态，并启动 ENIAC 运算。

循环单元发射的脉冲序列统领着 ENIAC 操作的时间。一个 100 kHz 的石英振荡器产生 10 毫秒宽的脉冲，每个累加器通过加法循环完成一次加法需要 20 个基本脉冲，也即 200 毫秒来完成，这就是大家常说 ENIAC 每秒能做 5000 次运算的来源。

ENIAC 是一台同步并行的计算机——之所以说同步，是因为它的运算都是通过循环单元发射的脉冲来同意控

严格的串行工作。

以全新的方式编程

通过数字脉冲触发 ENIAC 的操作来将数字总线中的线同编程总线相连是可能的，这通常使用编程脉冲来完成。因此，ENIAC 的操作顺序是可以动态改变的。之所以能这样，是因为数字总线和编程总线的信号电平相同，这使得 ENIAC 成为当时首台能实现条件转移指令的计算机。毛赫利在写给威尔克斯的信中主张让 ENIAC 原始设计的一部分——函数存储单元来发射编程脉冲，也就是说要让数字脉冲来行使编程脉冲的功能，因为函数存储单元每读入一个数时就会发射一个数字脉冲。ENIAC 的存储程序计算机概念便是利用了这一特征来实现：函数存储单元成为了编程指令的只读存储器，ENIAC 开发者称其为指令。这一套 2 位数字指令码由冯·诺伊曼提出，他提供了一系列标准操作，以存储在 3 个函数存储单元中的编码顺序来触发，每个指令码都能触发一系列固定的事先设定好的 ENIAC 操作顺序，

资源。通过改变只读存储器，就可以在不改变硬件的情况下改变指令的行为。

如果能定义并初始化足够多的这类指令，程序员就再也不用每次编新的程序时都要重新插拔 ENIAC 的接线了，他们只需手动向函数存储单元键入一系列命令就行，ENIAC 会自动执行，这就成为了真正的程序，将 ENIAC 变成了一台虽然有些原始，但是真正的内部存储程序计算机。

源文件缺失

没有任何文献记录冯·诺伊曼是如何提出他计划书中的那些思想的，不仅如此，也没有任何文献记录阿黛尔·戈德斯坦如何将冯·诺伊曼的基本想法变成可以在 ENIAC 上操作的具体概念的。不过，ENIAC 最初 6 位女程序员（又被称为“ENIAC 女性”）中的一位——贝蒂·琼·詹宁斯（Betty Jean Jennings）回忆出了一些她当时与阿黛尔一道开发原始指令码的经历。詹宁斯说，冯·诺伊曼当时正在研究存储程序计算机最佳的指令集，并向克利平格提出了可以在 ENIAC 身上的一套。克利平格带领了一个小组研究如何实现它，每两周就去一趟普林斯顿大学寻求冯·诺伊曼的意见，冯·诺伊曼就在那里勾勒出他们需要实现的指令。詹宁斯强调，要在 ENIAC 上实现这些指令是很难的。

1982 年 7 月 24 日，威廉·F·阿斯普雷（William F. Aspray）在美国全国计算机大会中关于存储程序概念的一个分会场上做了一个报告，也提到了一件与此相关的轶事。阿斯普雷提到，克利

函数存储单元成为了编程指令的只读存储器， ENIAC 开发者称其为指令。

制的，之所以说并行，是因为它的所有计算单元都是同时运行的。然而，这一并行特征很难有效利用，因为研究人员找不到同步多个平行操作的方法。因此，他们将 ENIAC 改造成一台存储程序计算机，迫使其牺牲并行操作功能，变成

运行以指令码定义的函数。这一操作很像 20 世纪 70 年代中期的“微编程计算机”（microprogrammed computer）概念：微编程计算机中有 20 种指令，通过一个基于只读存储器（ROM）的微程序让机器执行指令，访问机器中的多个硬件

平格(他也在场大会的某个分会场做了报告,介绍了他是如何将 ENIAC 改造为存储程序计算机的)被弹道研究实验室送往摩尔学院是为了寻找让 ENIAC 解偏微分方程的方法。阿黛尔告诉了他如何从函数存储单元获取数字脉冲并用它来给 ENIAC 编程。有了这个方法,克利平格开发出了一套三地址的编码系统。冯·诺伊曼从阿黛尔那里学来了这套系统,然后在摩尔学院与克利平格讨论了一个星期以后,两人为 ENIAC 开发出单地址系统。不过,仍然没有任何文献记录下克利平格起初开发出的三地址系统,而要想象出这套系统也是很困难的,要知道 ENIAC 只有 20 个累加器,没有足够的地址容量。因此,指令无法像我们今天这样带上操作数地址,他们只得单用指令码来定义一个特定的指令,例如用不同的指令码来定义对不同累加器的同一操作。

将ENIAC变成一台存储程序计算机

1946 年的后几个月, ENIAC 被搬到弹道研究实验室新建的计算副楼, 位于马里兰的阿伯丁训练场, 并重新组装, 于 1947 年 7 月 29 日重新投入运行。ENIAC 的布置发生了些变化, 多出了新造的两个电子面板, 代表两个新的单元。这表明, 弹道研究实验室的研究人员在重建 ENIAC 的时候已经在计划提高它的性能了。

克利平格的报告则描述了 ENIAC 指令集建设过程中的技术实现问题。克利平格写到, 如果不增加或修改硬件的

话, 可以让 ENIAC 实现 51 种操作指令, 尽管他并没有解释这是怎么实现的。重建后的 ENIAC 增加了 4 个新硬件, 可以让它以更高的效率实现 60 种操作指令。这些新增的单元中, 一个是十步步进器, 作为指令码转换过程的一部分, 一个

中只有六步步进器, 因此 β 位只能取 0 到 5 这 5 个数, 故指令码共有 60 个。图 4 即为指令码转换的框图。

转换过程如下: 选择的指令码的 α 位(在图 4 中是 4)在函数存储单元的相应存储位置被读取出来, 产生相对应

重建后的 ENIAC 增加了 4 个新硬件, 可以让它以更高的效率实现 60 种操作指令。

指令选择器, 可以从函数存储单元中 12 位的存储位置里读取两位的指令码。由于总共有 3 个函数存储单元, 因此需要另一个新单元来告诉指令选择器读取哪一个, 这个新单元叫做函数存储单元选择器。第四个新增单元是一系列脉冲放大器, 可以用来将同一个数字脉冲送往不同单元的不同输入口。这些新单元一起组成了一个转换器, 装在两个新增电子面板之一上。

“转换”这一概念, 指的就是把函数存储单元当做程序存储器, 将 6 个两位指令码存储在函数存储单元的 12 位存储位置上。十进制的两位指令码从函数存储单元中被读取并解码, 产生一个特定的数字脉冲, 反过来再触发 ENIAC 硬件一系列的加法循环操作, 实施指令码所代表的函数操作。如果两位指令码左边一位的数字是 α , 右边一位的数字是 β , 那么指令码就是 $10 * \alpha + \beta$ 。通过新安装的十步步进器可以解码 α 位, β 位则由主程序控制单元中的 10 个六步步进器来解码。由于主程序控制单元

的数字脉冲, 送往十步步进器的输入端, 将步进器设到第 4 步。十步步进器每个输出都连接着六步步进器的输入, 因此, 十步步进器到达第 4 步以后就激活了与第 4 步相连的六步步进器, 随后读取指令码的 β 位(在图 4 中是 3), 将数字脉冲送往对应的六步步进器的第 3 步, 也即数字脉冲最终被送往第 4 十步步进器的六步步进器的第 3 步, 这就将这一数字脉冲结果与指令码 43 建立起了唯一的联系, 并送往编程总线以激活指令码为 43 的加法循环操作。

转换器的设置最多能容下 60 个指令码, 即 00 到 05, 10 到 15……直到 90 到 95。克利平格提到, 他们设计了两套指令码, 共使用了 67 个不同的指令。这两套指令码里有 53 个指令是共有的, 另外 7 个指令则代表着不同的函数。

表 1 详细描述了用第 13 号累加器作为从头寄存器来执行这样一条指令的过程(想知道全部的初始指令列表, 可以到《IEEE 计算史年鉴》在线版本的“Web Extras”部分查找。)

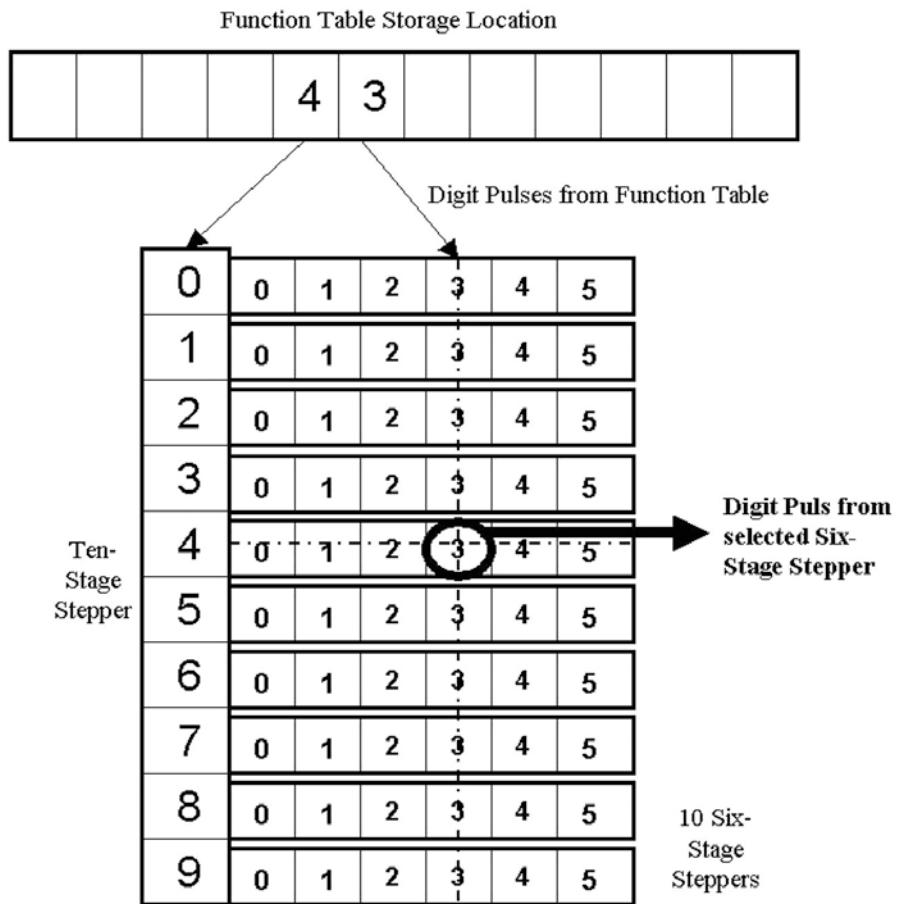


图 4. ENIAC 指令码转换过程的逻辑框图（由作者绘制）。

表 1. 理查德·克利平格记录下的指令细节。

代码号	代码号	代码	累加循环次数	命令执行前受影响的累加器中的内容	命令执行前受影响的累加器中的内容
30	X	不做四舍五入，10位乘数。累加器11中的数a与数d相乘，再与累加器13中的数相加，将结果保存在累加器15中。最终，累加器11、12、13中的值分别为a、d和0。	15	(11)a (12)b (13)c (15)d	a d 0 ad+c

* 克利平格用字母 a、b、c、d 来表示累加器中保存的数字，所以他在解释中使用了“数字 a”。最后两列表示按“命令”列中描述的命令代码 30 的执行中使用的累加器中的内容。(11)a 表示累加器 11 中包含数字 a，诸如此类。克利平格没有在最后一列中重复 (11)，但确保最后两列是一一对应的，使读者能够将最后一列与倒数第二列联系起来（即最后一列中的“a”即表示(11)a）。

破解编码

尽管没有任何文献记录下关于冯·诺伊曼和阿黛尔如何设计出最初的指令集的，我们仍然可以尝试还原出他们的思考过程。我们已经知道了 ENIAC 中有哪些硬件单元，但还要考虑数据在这些单元中是如何流动的。ENIAC 的数据来源有 3 个：IBM 读卡器、常数发送器中可通过开关手动设置的那一部分，以及同样可通过开关手动设置的函数存储单元。IBM 读卡器读取卡纸时，从卡纸读取的数字被送往与之相连的常数发送器，并存储在那里以便之后使用。随后，这些十进制数会一个接一个地从常数发送器被送往 20 个累加器。在常数发送器的手动设定部分设定的常数和来自函数存储单元的数据也可以被送往 20 个累加器中，数据还可以在累加器之间传输，最终，得出的结果可以从累加器送往 IBM 卡纸打孔机输出。此外，我们必须意识到，所有累加器并不是彼此平等的，其中一些特殊的累加器会通过固线连接到特定的 ENIAC 单元，例如乘法器，或是除法与平方根单元。

我们还需要记住的是，ENIAC 的累加器并没有寻址功能，指令码不仅需要定义操作本身，还需要定义要用到的累加器的“地址”。因此，这些指令也需要定义读取卡纸、从常数发送器和函数存储单元将数据传送到累加器，以及将数据在累加器之间传输（包括相加），设置好累加器以进行乘法、除法、平方根等运算，最后将结果传输至卡纸打孔

机。其他指令还要让累加器指向函数存储单元中特定的某一行，以使下一行的数据能从函数存储单元连接到累加器，不管这个数据所代表的是特定的值还是待执行的指令。

这样，ENIAC 中的累加器实际上行使了我们今天计算机中寄存器的功能。6 号累加器被用作指令计数器，8 号累加器被用作数据地址寄存器，尽管当时 ENIAC 的设计者还没发明出这些术语。6 号累加器包含了函数存储单元的数字及该函数存储单元中一个指令的列的地址，这样一来，一段程序就可以按顺序运行了。将 6 号累加器加载为一个新的值，就可以将一个分支移动到程序的另外一部分。8 号累加器也包含了函数存储单元的一个数和该函数存储单元中的一列地址，但这个地址指向的是存储在该位置的一个十进制数值，而非指令。此外，8 号累加器还会每次自动加 1，以读取一个列表数值——这是 ENIAC 最初设计时原本想让函数存储单元行使的功能。

如果更仔细地研究一下原始指令集，我们就能更深刻地理解其背后的概念：指令可能会占据 1 ~ 4 个指令位置。因为指令位置被定义成两位十进制数，所以指令可能会用到 2 位、4 位、6 位或 8 位十进制数。最开头的位置（即两位十进制数）是指令码，而第 2 ~ 4 个指令位置存储的是数据，而非地址，以待指令执行过程中使用。

占据两个位置的指令示例如下：

- 70 号指令码：将下一对指令位数送到 15 号累加器
- 90 号指令码：移位（第二个位置定义了移动的方向与数量）
- 占据 3 个位置的指令示例如下：
- 93a 号指令码：将接下来的 3 位指令数送往 8 号累加器
- 94 号指令码：将接下来的 4 位指令数送往 15 号累加器
- 94a 号指令码：将接下来的 3 位指令数送往 6 号累加器

占据 5 个位置的指令示例如下：

- 95 号指令码：将接下来的 6 位指令数送往 15 号累加器
- 95a 号指令码：将接下来的 6 位指令码送往 6 号累加器

除了移位指令以外，所有包含超过一个指令位置的指令都是用来将 2 位（或 3 位、4 位、6 位）数字根据函数存储单元中的指令码移动到累加器中以实施的。由于函数存储单元是只读存储，因此这些指令的使用方式只能通过散置其间的常数或地址来解释。这些移动指令就是用来从函数存储单元中读取这些数据，并将其送往累加器。从这些指令所送往的累加器（6 号、8 号和 15 号）可以看到，用到 6 号累加器的 94a 和 95a 号指令码组成了无条件的跳跃指令，可以用来跳过散置的常数。用到 8 号累加器的 93a 号指令码则设立了一个指向十进制数的指针，这一指针反过来又通过 72 号指令将位置信息

送往 11 号和 15 号累加器。用到第 15 号累加器的 70 号、94 号和 95 号指令码则用来读取散置在函数存储单元中的常数。

其他所有指令都只包含一个两位十进制数的指令位置。除了传送指令和数学指令之外，我们还找到了另外一种无条件跳跃指令：73/73a 号指令，以及 75 号有条件的跳跃指令。73/73a 号指令会将 15 号累加器的内容移到 6 号累加器的指令计数器中，从而影响了一个无条件跳跃操作；71 或 95a 号指令用于将两个可能的目标送往 6 号累加器，从而为一次有条件跳跃做准备；75 号指令则测试了 15 号累加器中所包含的数字，并根据这个数字是正是负设定 6 号累加器中的两个地址之一，以用于下一个即将执行的指令。

从以上简明的分析中，我们可以看到由冯·诺伊曼、阿黛尔·戈德斯坦和克利平格设计出的这份原始指令列表就已经包含了迭代以及在某个数达到特定值时终止迭代所需的指令。其中包含的跳跃指令也为子程序的建立提供了条件，尽管返回地址需要存储在一个未曾被子程序编码所占用的空闲累加器中。

升级是加速，而非减速

从函数存储单元中读取一个指令码、通过转换器将它解码并触发对应的 ENIAC 操作，共需要 7 个加法循环。由于这个基本操作顺序（如今我们称之为读取周期）与对应的 ENIAC 操作之间存在重叠，而每个操作都需要大约 7 个

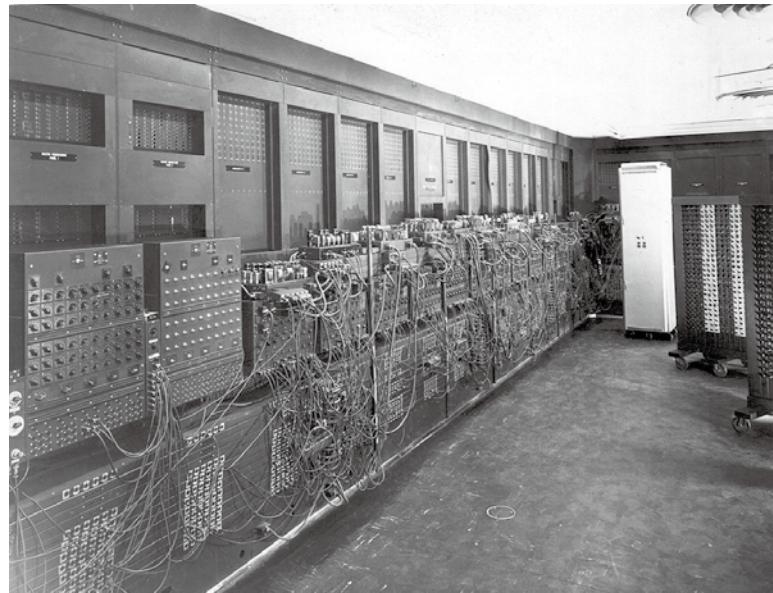


图 5. ENIAC 的照片，背景里是 Burroughs 磁芯存储器。图片来源：US Army photo。

加法循环因此与直接通过插拔线相比，用指令码编程大大降低了 ENIAC 的运行速度。然而，赫尔曼·戈德斯坦和冯·诺伊曼提到，原始 ENIAC 的最大速度几乎不可能达到。他们观察到，ENIAC 的累加器的运算速度与利用卡纸的输入 / 输出操作速度不匹配，这大大降低了执行一套完整的程序时的实际速度，只相当于累加器速度的三分之一。

戈德斯坦和冯·诺伊曼还引用了弹道计算的数据。弹道计算几乎全依赖于读卡与打卡，整台 ENIAC 的速度被限制在了卡片打孔上。他们把较为缓慢的存储程序计算机与较快的固线式计算机（如以前的 ENIAC）的速度进行了比较，当然也考虑了编程所需的时间：从前的 ENIAC 在解决一个问题之前所需要的插拔接线、设定开关等初始工作需要 8 个小时，而程序运行的时间则很短，只要 5 分钟左右；而改造成存储程序计算机后，ENIAC 的编程只需要通过在开

关的帮助下，手动向函数存储单元键入程序指令就行了，这一过程比此前插拔接线快得太多，以至于程序运行多花的那点儿时间已经完全不重要了。因此，新的编程方式绝对代表着重要的进步。

继续升级

然而，随着时间的推移，ENIAC 需要越来越多的技术上的改进以提高经济效益，并与弹道研究实验室新建造的计算机——EDVAC 相竞争。

升级的第一步就是将主程序控制单元中指令转换器的 10 个六步步进器换成 10 个新的十步步进器，这样就能解码从 00 到 99 的全部 100 个两位指令码了，同时也解放了主程序控制单元的步进器，让它们能行使原先的功能。克利平格的报告指出，这个转换器在 1948 年秋天他写报告的时候就已经安装完毕了，而多出来的指令可以让

ENIAC 在从函数存储单元读取指令之外,还能直接从打孔的卡纸上读取指令。

ENIAC 的程序员、数值分析师, 同时也是 1948-1955 年弹道研究实验室 ENIAC 分部主任的巴克利·W·弗里茨分别在 1949 年和 1951 年写过两篇报告, 描述了指令转换器升级后的全部指令列表(见在线版本“Web Extras”部分的表 2)。弗里茨可能是在 1951 年 ENIAC 新加入第 4 个函数存储单元时更新的报告, 此时 ENIAC 已经可以通过插接板来编程, 无需手动操作开关。想了解更多弗里茨的编程工作, 两次升级。1952 年初, 工作人员给它装上了一个高速电子移位寄存器, 以提高移位操作的速度。除此之外, 关于这台机器就没有更多的细节被揭露了。可以明白的是, 20 个累加器作为内部动态内存而言是不够的。Burroughs 公司做了一些关于磁芯存储器的研究, 也承包了为 ENIAC 开发磁芯存储器的任务。他们于 1953 年 7 月开发完成的 4100 比特的存储单元是世界上首个可操作的磁芯存储器, 它使用一个 4 比特的二进制码来表达一位十进制数字, 能存储 100 个带正负号的 10 位十进制数。弹道研究实验室增加了合适的界面电子电路, 以及三条额外的指令以访问这一新的存储器, 但除此之外也没有其他的细节披露了。图 5 展示了与 ENIAC 相连接的 Burroughs 磁芯存储器。

结论

改造过后的 ENIAC 是美国首批使用存储程序原理的计算机之一。全

世界首个存储程序计算机当然是英国曼彻斯特大学的曼彻斯特计算机(Manchester Machine), 于 1948 年 6 月 21 日首次运行, 比改造后的 ENIAC 早几个月。ENIAC 的故事启示我们, 我们不仅仅要关注一台新的计算机的初始建造过程, 也要关注它整个生涯中一点一滴的进展, 哪怕这些过程不那么为人所知。在计算机技术刚诞生的早期, 每天使用计算机的人也为整个计算机技术的发展做出了相当大的贡献, 他们也值得在历史中拥有一席之地, 哪怕只有很小的一席之地。

致谢

我要诚挚地感谢《IEEE 计算历史年鉴》的主编戴维·艾伦·格里尔(David Alan Grier), 是他鼓励了我钻研 ENIAC 这段鲜为人知的历史; 我还要感谢查尔斯·巴比奇研究所的工作人员, 尤其卡丽·塞布(Carrie Seib), 以及宾夕法尼亚大学档案馆的工作人员, 尤其南希·R·米勒(Nancy R. Miller), 他们帮我获得了弹道研究实验室的文献, 这对我的研究至关重要。我还要感谢论文匿名评审给了我宝贵的建议, 以及《年鉴》的工作人员, 让我的文章更具可读性。

参考文献

1. M.R. Williams, A History of Computing Technology, 2nd ed., IEEE CS Press, 2000.
2. H.H. Goldstine, The Computer from Pascal to von Neumann, Princeton Univ. Press, 1972, p. 184.
3. H.H. Goldstine, The Computer from Pascal to von Neumann.
4. J. von Neumann, “First Draft of a Report on the EDVAC,” IEEE Annals of the History of Computing, vol. 15, no. 4, 1993, pp. 27-75.
5. D.A. Grier, “Obituary to Herman H. Goldstine,” Goodman, “Soviet Computing,” pp. 552-553. Gerovitch, From Newspeak to Cyberspeak, pp. 284, 287-290. IEEE Annals of the History of Computing, vol. 26, no. 3, 2004, pp. 2-3.
6. H.H. Goldstine, The Computer from Pascal to von Neumann, p. 185.
7. H. Lukoff, From Dits to Bits: A Personal History of the Electronic Computer, Robotics Press, 1979.
8. N. Metropolis and J. Wrolton, “A Trilogy on Errors in the History of Computing,” Annals of the History of Computing, vol. 2, no. 1, 1980, pp. 49-59.
9. H.H. Goldstine, The Computer from Pascal to von Neumann, p. 199.
10. M. Campbell-Kelly and M.R. Williams, The Moore School Lectures: Theory and Techniques for Design of Electronic Digital Computers, Charles Babbage

ENIAC 编程小指南

[起这个标题是为了致敬马丁·坎贝尔·凯利 (Martin Campbell-Kelly)，他早年在英国写下的一系列关于早期编程的精彩文章中就用了类似的语句。]

编程思考

巴克利·弗里茨在他对ENIAC指令转换器指令列表的报告开头就写到，这一报告是给准备用ENIAC来解决问题的人写的手册指南，我写的这份小指南就是基于弗里茨在报告中的解释。

弗里茨给出了9个步骤，我在这里也详细地解释了要让ENIAC解决一个问题所需要经过的这9个步骤。

1. 第一步是数学上的：你要解决的物理问题必须能用数学公式表达出来。
2. 在对该问题进行数学建模以后，就需要找到解决该问题的数值计算方案。用弗里茨的话说，这类方案“最好是已知的，或者可以收敛成想要的解”。
3. 到此之前的准备工作都是纯数学上的，与ENIAC毫无关系。不过从现在开始，就需要考虑到ENIAC的特定特征了。首先，由于ENIAC只能提供-1到+1范围内数字的定点运算，所以所有的量都得是确定的，在需要的情况下还得按比例调节到-1到+1的区间内。
4. 到第4步，程序员就该画出让ENIAC解决数值问题的流程图了。流程图需要非常详细，以让每一部分都能轻易被转换成一系列ENIAC指令。
5. 下一步就是实际的编码工作了：将流程图的每一个框都换成相应的ENIAC指令码。
6. 到这里，看似这一系列指令可以直接应用于ENIAC来解决

问题了。然而，根据实际经验，弗里茨仍然建议先用台式计算机手动执行一个样本问题，这段试运行过程应当包含所有指令，并将结果与直接数学计算得出的解相比较。

7. 弗里茨还要求所有运行ENIAC的人员都应检查一下这段代码，以确信所有的ENIAC指令都正确应用，并与该报告中的描述完全吻合。常见的错误包括小数点位置不对、意外清空了一个累加器导致丢失了一个本当存储的数据、迭代的设置与控制错误，以及处理跟程序各种不同运行次序的操作地址方面。
8. 一旦前两项测试都成功通过，就可以开始准备ENIAC的输入数据了。输入数据需要手动在卡纸上打孔，还要把它放到打印机上确认。同时也需准备好函数存储单元的初始化纸张，包含ENIAC指令以及需要通过开关设置进函数存储单元的常数。
9. 弗里茨建议的最后一步，就是计算出ENIAC解决这个问题所需的大概时间长短。只要将所有指令中的加法循环累加起来就能得到总时长，如果有些迭代的次数不确定，就估计一下。

弗里茨在特定的指令码下也给出了一些评论，从细节方面体现了他对ENIAC编程的深刻见解。例如，第13号累加器本是作为大多数指令的工作寄存器，应该保持清空状态，不是用来存储数据的，然而，聪明的程序员会利用未清空的第13号累加器来达到一些本该多用一个指令来达成的效果。在当时，写出最小、最省时的程序比让代码整齐、结构清晰更重要。

ENIAC的使用者还必须面对如何表示负数的问题。ENIAC使用十补码来表示负数，这种表示形式是用来在常数传送器和函数存储单元的开关上设定负数用的，但在卡纸上打孔用的却是另一种形式：数字本身还是按照正常的位数打孔，但要在第一栏和第6

Institute Reprint Series for the History of Computing, vol. 9, M. Campbell-Kelly, ed., MIT Press, 1985.

11. M.V. Wilkes, "Mauchly's Position

on von Neumann's Role in Drafting the EDVAC Report," Annals of the History of Computing, vol. 2, no. 4, 1980, pp. 376-377.

12. H.H. Goldstine, The Computer

from Pascal to von Neumann, p. 233.
13. R.F. Clippinger, A Logical Coding System Applied to the ENIAC, BRL 673, report, Ballistic Research

栏打一个字母x以提示这个数为负数。读卡器在读取数字时，会将这种形式转化成十补码。

6号累加器包含了函数储存单元中待执行的指令码行的地址。第06号指令让6号累加器听候指令，将15号累加器中的内容送到6号累加器中，于是就形成了一个无条件跳跃指令。由于15号累加器的内容也可能是此前其他计算过程的结果，程序员可以动态地计算并操作地址，这就是弗里茨在该指令码上提出的评论。

一位ENIAC程序员不仅要跟指令打交道，还要考虑一些会影响到ENIAC操作的开关设置。每个累加器都有一个手动开关，用来防止累加器被清空。如果这个开关被打开，该累加器就会忽视一切将其清空的指令，将新的输入累加到它目前的内容之上。因此，如果这个开关被意外激活，势必会带来错误的计算结果。

将数字按比例缩小到-1到+1之间以适应ENIAC的计算操作，是通过移位指令来完成的。程序员必须清楚地知道计算过程中一个特定的数能达到怎样的大小，并适时运用移位指令，而在他们不确定数值算法在多次迭代的情况下会变成什么样的情况下，这可不是一个简单的任务。因此，这要求程序员对乘法、除法、平方根等更复杂的运算有着深刻的理解，以追踪输入的数据在这类操作下会变得多大。

还有一个问题，就是在多次迭代算法中可能出现的舍入效应。如果运算需要求用两个数乘以同一个数的和，弗里茨建议用 $(a + b) * c$ ，而非 $a * c + b * c$ ，这样可以减少四舍五入误差。在乘法操作中，如果计算机虚拟的第11位上得到的是5或大于5的数字，就会通过往第11位上加5的方式进位到第10位，因此，为了减小误差，应如此减少乘法的操作次数。同样的道理，在计算 $a * b$ 除以c的商时，也应该使用 $(a/c) * b$ 而非 $(a * b)/c$ 。

第69号条件跳跃指令（被称为“条件转移”）专门用来在结果已经达到适当的大小时终止迭代。由于它所依赖的条件是15号累加器中数字的正负号，程序员必须通过每次在结果上增加或减少

一个目标数量，来保证数字在正确的时候变号。另一个利用第69号指令的机智办法就是用它来从子程序中返回。返回地址必须存储在该子程序未曾使用的累加器中，在子程序的最后，这个地址必须被送往15号累加器，再从那里被送往6号累加器中的目标地址位置。69号条件转移指令测试到15号累加器中的数字为正（即返回地址是个正数），于是将目标地址移动到目前6号累加器中的地址子分区，从而强迫其跳跃到返回地址。

那么ENIAC的程序样例长什么样？

很遗憾，弗里茨并没有在报告中留下ENIAC已经解决的问题的编程样例，只通过引用的方式指引读者去阅读其他的报告。然而，不幸的是，不管是弗里茨提到的这些报告，还是我在研究中查阅的其他报告，都没有包含ENIAC的实际程序代码。不过，克利平格提供过一段ENIAC的测试程序，他还记载了计算超声波气流经过人的应用。为了原汁原味地给读者呈现ENIAC的代码，这段测试程序的第一部分，即检验其从函数存储单元读取能力的部分作为程序样例被放在在线版本“Web Extras”部分，附有对其指令及其影响的解释。

参考文献

1. B.W. Fritz, Description of the Eniac Converter Code, BRLM582, memorandum report, Ballistic Research Laboratories, Aberdeen Proving Ground, Md., Dec. 1951.
2. R.F. Clippinger, A Logical Coding System Applied to the ENIAC, BRL 673, report, Ballistic Research Laboratories, Aberdeen Proving Ground, Md., 29 Sept. 1948.
3. Readers interested in more details should refer to Table A in the online “Web Extras,” listing all ENIAC orders of the initial order set implemented and used by Clippinger.

Laboratories, Aberdeen Proving Ground, 29 Sept. 1948.

14. H.H. Goldstine and J. von Neumann, “Planning and Coding Problems for an Electronic

Computing Instrument,” Papers of John von Neumann on Computing and Computer Theory, vol. 12, Charles Babbage Inst. Reprint Series for the History of Computing,

W. Aspray and A. Burks, eds., MIT Press, 1986, pp. 151-306.
15. R.D. Richtmyer, “The Post-War Computer Development,” The American Mathematical

- Monthly, vol. 72, no. 2, 1965, pp. 8-14.
16. A.K. Goldstine, A Report on the ENIAC (Electronic Numerical Integrator and Computer), Univ. of Philadelphia, Moore School of Electrical Eng., June 1946.
17. J. Van der Spiegel et al., "The ENIAC: History, Operation and Reconstruction in VLSI," The First Computers: History and Architectures, R. Rojas and U. Hashagen, eds., MIT Press, 2000, pp. 121-178.
18. This feature is described in paragraph 4.5.3, Magnitude Discrimination Programs, in Goldstine, A Report on the ENIAC, Ref. 16.
19. Wilkes, "Mauchly's Position on von Neumann's Role in Drafting the EDVAC Report," IEEE Annals, p. 6.
20. I am referring to the Sperry Univac 9700 computer, which was based on this architecture, and with which I was working at that time as a support engineer for Sperry Univac in Switzerland.
21. B.W. Fritz, "The Women of ENIAC," IEEE Annals of the History of Computing, vol. 18, no. 3, 1996, pp. 13-28.
22. W. Aspray, "Pioneer Day, NCC '82: History of the Stored-Program Concept," Annals of the History of Computing, vol. 4, no. 4, 1982, pp. 358-361.
23. E.C. Berkeley, Giant Brains or Machines That Think, John Wiley & Sons, 1949.
24. The IBM card punch is normally referred to as "Printer" in the original source documents. It is, however, clear from these documents that cards were punched and then manually transferred to a printer for offline printing.
25. H.H. Goldstine and J. von Neumann, "On the Principles of Large Scale Computing Machines," Papers of John von Neumann on Computing and Computer Theory, vol. 12, Charles Babbage Inst. Reprint Series for the History of Computing, W. Aspray and A. Burks, eds., MIT Press, 1986, pp. 317-348.
26. It took 3,000 add-cycles to mechanically read a card in the IBM card reader and move its data to the Constant Transmitter.
27. B.W. Fritz, Description and Use of the Eniac Converter Code, TN 141, tech. note, Ballistic Research Laboratories, Aberdeen Proving Ground, Md., Nov. 1949.
28. B.W. Fritz, Description of the Eniac Converter Code, BRLM 582, memorandum report, Ballistic Research Laboratories, Aberdeen Proving Ground, Dec. 1951.
29. K. Kempf, "Electronic Computers within the Ordnance Corps," historical monograph, Aberdeen Proving Ground, Nov. 1961; <http://ftp.arl.mil/~mike/comphist/61ordnance/>.
30. M.H. Weik, "The ENIAC Story," Ballistic Research Laboratories, Aberdeen Proving Ground, 1961; <http://ftp.arl.mil/~mike/comphist/eniac-story.html>.
31. M.R. Williams, A History of Computing Technology, p. 322.

微信名：计算人
微信号：jisuanren



知乎

2017 香山杯 WSDM 挑战赛

对 3000,000 问题
进行语义理解
添加话题标签

时间

* 80,000
奖金

2017.5.15 - 2017.8.15



参赛入口





保持联系。

无论你在哪里，都能紧随IEEE计算机协会的脚步。

在Twitter、Facebook、Linkedin和YouTube上关注我们。



@ComputerSociety, @ComputingNow



facebook.com/IEEEComputerSociety
facebook.com/ComputingNow



IEEE Computer Society, Computing Now



youtube.com/ieeecomputersociety