

Alexandra I. Cristea
Christos Troussas (Eds.)

LNCS 12677

Intelligent Tutoring Systems

17th International Conference, ITS 2021
Virtual Event, June 7–11, 2021
Proceedings

 Springer

Founding Editors

Gerhard Goos

Karlsruhe Institute of Technology, Karlsruhe, Germany

Juris Hartmanis

Cornell University, Ithaca, NY, USA


Editorial Board Members

Elisa Bertino

Purdue University, West Lafayette, IN, USA

Wen Gao

Peking University, Beijing, China

Bernhard Steffen 

TU Dortmund University, Dortmund, Germany

Gerhard Woeginger 

RWTH Aachen, Aachen, Germany

Moti Yung

Columbia University, New York, NY, USA

More information about this subseries at <http://www.springer.com/series/7408>


Alexandra I. Cristea · Christos Troussas (Eds.)

Intelligent Tutoring Systems

17th International Conference, ITS 2021
Virtual Event, June 7–11, 2021
Proceedings

Editors

Alexandra I. Cristea 
Department of Computer Science
Durham University
Durham, UK

Christos Troussas 
University of West Attica
Aigaleo, Greece

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-030-80420-6

ISBN 978-3-030-80421-3 (eBook)

<https://doi.org/10.1007/978-3-030-80421-3>

LNCS Sublibrary: SL2 – Programming and Software Engineering

© Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The 17th International Conference on Intelligent Tutoring Systems (ITS 2021) was to be held in Athens, Greece, during June 7–11, 2021. The hosting institution of the ITS 2021 conference was the University of West Attica; however, due to the world-wide COVID-19 pandemic it took place online.

Conforming to the current move of education, work, and leisure online, the title of ITS 2021 was “Intelligent Tutoring Systems in an Online World”. Its objective was to present academic and research achievements in Computer and Cognitive Sciences, Artificial Intelligence, and, due to its recent emergence, specifically, Deep Learning in Tutoring and Education. The aim of ITS 2021 was to promote and improve learning technology systems, by combining novel and advanced technology with complex and nuanced research approaches. It offered a forum for exploring emerging and noteworthy progress in the field of Artificial Intelligence in Education.

The call for scientific papers focused on a plethora of topics of interest in the area of ITS and beyond, including the following:

- Intelligent Tutoring
- Learning Environments for Underrepresented Communities
- Artificial Intelligence in Education
- Human in the Loop, Understanding Human Learning on the Web in a Virtual (Digital) World
- Machine Behavior (MB), Explainable AI, Bias in AI in Learning Environments
- Emotions, Modeling of Motivation, Metacognition and Affect Aspects of Learning, Affective Computing and ITS
- Extended Reality (XR), Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR) in Learning Technologies
- Informal Learning Environments, Learning as a Side Effect of Interactions
- Collaborative and Group Learning, Communities of Practice and Social Networks
- Analytics and Deep Learning in Learning Systems, Educational Data Mining, Educational Exploitation of Data Mining and Machine Learning Techniques
- Sentiment Analysis in Learning Environments
- Data Visualization in Learning Environments
- Privacy, Security and Ethics in Learning Environments
- Gamification, Educational games, Simulation-based Learning and Serious Games
- Brain-computer Interface Applications in Intelligent Tutoring Systems
- Dialogue and Discourse During Learning Interactions
- Ubiquitous, Mobile, and Cloud Learning Environments
- Virtual Pedagogical Agents and Learning Companions
- Multi-agent and Service-oriented Architectures for Learning and Tutoring Environments
- Single and Group Wise Action Modeling in Learning Environments
- Ontological Modeling, Semantic Web Technologies, and Standards for Learning

- Empirical Studies of Learning with Technologies
- Instructional Design Principles or Design Patterns for Educational Environments
- Authoring Tools and Development Methodologies for Advanced Learning Technologies
- Domain-specific Learning Technologies, e.g. Language, Mathematics, Reading, Science, Medicine, Military, and Industry
- Non-conventional Interactions between Artificial Intelligence and Human Learning
- Personalized and Adaptive Learning Environments
- Adaptive Support for Learning, Models of Learners, Diagnosis and Feedback
- Recommender Systems for Learning
- Causal Modeling and Constraints-based Modeling in Intelligent Tutoring

The call for papers sought papers that presented significant new research findings in the use of advanced computing technology and interdisciplinary research to allow, promote, and enhance human learning. Full papers allowed for discussion of more mature and finalized research results, whilst short papers allowed discussions around brief novel findings. There was also a posters track, which included an excellent network for researchers to discuss research prototypes and work in progress to conference attendees.

The international Program Committee consisted of 63 leading members of the Intelligent Tutoring Systems community (20 senior and 43 regular), as well as highly promising younger researchers. Scientific papers were reviewed by three to five reviewers through a double-blind process. Only 25% of the submitted papers, were accepted as full papers, about 24% were accepted as short papers, and just 15% were accepted as posters. These percentages indicate that ITS 2021 was a top-flight, rather selective, high-quality conference.

A separate Doctoral Consortium (DC) offered a forum for Ph.D. students to present and discuss their research when it was still in the early stages of development, engage colleagues with similar goals, and collaborate with more senior members in the community (mentors). The Doctoral Consortium Chairs were Mizue Kayama, Shinshu University (Japan), and Mike Joy, University of Warwick (UK).

The full papers outlined some very important developments, the short papers explored some fascinating new theories, and the posters discussed research in progress that needs particular attention, all based on the ITS philosophy.

The main topics under which the accepted papers fall, on which basis we also structured this book, are as follows:

- **Theory** – comprising Theory and Reviews; Models; Concept Maps
- **Learner focus** – including Student Prediction; Learner Behavior; Feedback and Personalization; Groups, Teams, Social, Crowd and Communities Assessment
- **Future ITS orientation** – bringing together Games and Gamification; Emotions and Affect; and xtended Reality

A variety of new techniques had been introduced or revisited, including multi-modal affective computing, XR, mixed-compensation multidimensional item response, ensemble deep learning, cohesion network analysis, conversational agent, semantic web,

computer-supported collaborative learning, and social networking in education. The rigor of the research was high, and it revealed several generalizable findings. Furthermore, it created space for the use of approaches like retrospective trials, experimental research, and meta-analysis, which might include new insights at future ITS conferences.

The quality of a conference is reflected by the work of its participants as well as their ability to push the boundaries, and the rigor with which they encourage the rest of the research field to move beyond. The papers of ITS 2021 stretched the limits of intelligent tutoring, much as they had in the previous years. Reinforcement learning, artificial neural networks, semantic web technologies, natural language processing, social networking, digital assistants, and recommender systems were among the fields where they had documented remarkable work.

The ITS 2021 program was reinforced by the successful organisation of a Workshop: “Intelligent Tutor Demonstrations” by Mihai Dascalu, Amruth Kumar, and Daniela M. Romano, and two half-day Tutorials: “Learning Analytics Hands-On Tutorial” by Alexandra Cristea and “Data Science for Learning Process Management” by Filippo Sciarrone. They were all selected and managed by the Workshop and Tutorial Chairs, Amruth Kumar, Ramapo College of New Jersey (USA), Mihai Dascalu, University Politehnica of Bucharest (Romania), and Daniela Romano, University College London (UK).

We would like to express our thanks to many different contributors in the midst of the overwhelming and unforeseen circumstances of the COVID-19 pandemic.

The successful preparation and implementation of the ITS 2021 conference was secured by the original work of all the authors, the devoted contribution of the various Conference Chairs, the members of the Program Committee, and the Steering Committee, in particular its Chair, Claude Frasson. The organization, coordination, and online operation of ITS 2021 achieved by the Local Organizers and the Organization Chair, Kitty Panourgia. We would also like to address our special thanks to the Conference Sponsor, the “Education Sciences” journal (MDPI), for its support. Last but not least, we would like to acknowledge the Institute of Intelligent Systems (IIS) under the auspices of which this conference was held.

Rather than concluding this preface, we would like to emphasise that one of the main outcomes of the ITS 2021 conference is a fusion of new and established scholars, innovative and highly evolved subjects, theoretical developments and business interests, broadening of areas and deepening of subgenres. This equilibrium is an utterly necessary dimension. We hope you enjoy reading the papers and using them towards generating new ideas – and citing them in your own research!

April 2021

Alexandra Cristea
Christos Troussas

Organization

General Conference Chair

Cleo Sgouropoulou University of West Attica, Greece

Honorary Chair

Riichiro Mizoguchi Japan Advanced Institute of Science and
Technology, Japan

Program Committee Chairs

Alexandra Cristea Durham University, UK
Christos Troussas University of West Attica, Greece

Program Advising Chairs

Maiga Chang Athabasca University, Canada
Yugo Hayashi Ritsumeikan University, Japan

Workshop and Tutorial Chairs

Amruth Kumar Ramapo College of New Jersey, USA
Mihai Dascalu Politehnica University of Bucharest, Romania
Daniela Romano University College London, UK

Posters Chairs

Giora Alexandron Weizmann Institute, Israel
Jane Sinclair University of Warwick, UK

Doctoral Consortium Chairs

Mizue Kayama Shinshu University, Japan
Mike Joy University of Warwick, UK

Promotion, Publicity and Industry Chairs

Tatiana Gavrilova St. Petersburg University, Russia
Richard Tong Squirrel AI, China

Program Committee

Senior Program Committee

Roger Azevedo	University of Central Florida, USA
Bert Bredeweg	University of Amsterdam, The Netherlands
Stefano A. Cerri	University of Montpellier and CNRS, France
Maiga Chang	Athabasca University, Canada
Michel Desmarais	Ecole Polytechnique de Montreal, Canada
Claude Frasson	University of Montreal, Canada
Nathalie Guin	Université Claude Bernard Lyon 1, France
Yugo Hayashi	Ritsumeikan University, Japan
Kinshuk Kinshuk	University of North Texas, USA
Vivekanandan Kumar	Athabasca University, Canada
Amruth Kumar	Ramapo College of New Jersey, USA
Lewis Johnson	Alelo Inc., USA
Noboru Matsuda	North Carolina State University, USA
Gordon McCalla	University of Saskatchewan, Canada
Riichiro Mizoguchi	Japan Advanced Institute of Science and Technology, Japan
Roger Nkambou	Université du Québec à Montréal, Canada
Filippo Sciarrone	University Roma Tre, Italy
Stefan Trausan-Matu	Politehnica University of Bucharest, Romania
Christos Troussas	University of West Attica, Greece
Julita Vassileva	University of Saskatchewan, Canada

Program Committee

Giora Alexandron	Weizman Institute, Israel
Galia Angelova	Bulgarian Academy of Sciences, Bulgaria
Maria Bielikova	Kempelen Institute of Intelligent Technologies, Slovakia
Emmanuel Blanchard	IDU Interactive Inc., Canada
Jesus Boticario	National University of Distance Education, Spain
Tingwei Chen	Liaoning University, China
Chih-Yueh Chou	Yuan Ze University, Taiwan
Mark Core	University of Southern California, USA
Evandro Costa	Federal University of Alagoas, Brazil
Diego Dermeval	Federal University of Alagoas, Brazil
Philippe Dessus	Université Grenoble Alpes, France
Reva Freedman	North Illinois University, USA
Benjamin Goldberg	University of South Florida, USA
Sunčica Hadžidedić	Durham University, UK
Ella Haig	University of Portsmouth, UK
Elaine Harada Teixeira de Oliveira	Federal University of Amazonas, Brazil
Jason Harley	McGill University, Canada

Yusuke Hayashi	Hiroshima University, Japan
Gwo-Jen Hwang	National Taiwan University of Science and Technology, Taiwan
Seiji Isotani	University of Sao Paulo, Brazil
Patricia Jaques	Universidade do Vale do Rio dos Sinos, Brazil
Charalampos Karagiannidis	University of Thessaly, Greece
Mizue Kayama	Shinshu University, Japan
Akrivi Krouska	University of West Attica, Greece
Elise Lavoué	University of Lyon, France
Blair Lehman	Educational Testing Service, USA
Carla Limongelli	Roma Tre University, Italy
Chao-Lin Liu	National Chengchi University, Taiwan
Fuhua Lin	Athabasca University, Canada
Yang Long	Durham University, UK
Alvaro Ortigosa	Universidad Autónoma de Madrid, Spain
Kuo-Liang Ou	National Hsin-Chu University of Education, Taiwan
Elvira Popescu	University of Craiova, Romania
Valéry Psyché	Teluq University, Canada
Olga C. Santos	National Distance Education University, Spain
Lei Shi	Durham University, UK
Sergey Sosnovsky	Utrecht University, The Netherlands
Kaoru Sumi	Future University Hakodate, Japan
Thepchai Supnithi	National Electronics, and Computer Technology Center, Thailand
Marco Temperini	Sapienza University of Rome, Italy
Radu Vasiu	Politechnica University of Timisoara, Romania
Dunwei Wen	Athabasca University, Canada

Steering Committee Chair

Claude Frasson	University of Montreal, Canada
----------------	--------------------------------

Steering Committee

Stefano A. Cerri	University of Montpellier and CNRS, France
Maiga Chang	Athabasca University, Canada
Isabel Fernandez-Castro	University of the Basque Country, Spain
Gilles Gauthier	University of Quebec at Montreal, Canada
Guy Gouarderes	University of Pau and Pays de l'Adour, France
Yugo Hayashi	Ritsumeikan University, Japan
Amruth Kumar	Ramapo College of New Jersey, USA
Alan Lesgold	University of Pittsburgh, USA
James Lester	North Carolina State University, USA
Alessandro Micarelli	Roma Tre University, Italy

Roger Nkambou	Université du Québec à Montréal, Canada
Giorgos Papadourakis	Hellenic Mediterranean University, Greece
Elliot Soloway	University of Michigan, USA
John Stamper	Carnegie Mellon University, USA
Daniel Suthers	University of Hawaii, USA
Christos Troussas	University of West Attica, Greece
Stefan Trausan-Matu	Politehnica University of Bucharest, Romania
Beverly Woolf	University of Massachusetts, USA

Organizing Committee Chair

Kitty Panourgia	Neoanalysis, Greece
-----------------	---------------------

Organizing Committee

Aggelos Amarantos	Neoanalysis, Greece
Stefano Esposito	Neoanalysis, Greece
Elisavet Vasileiou	Neoanalysis, Greece
Isaak Tselepis	Neoanalysis, Greece
Rasa Tučinskaitė	Neoanalysis, Greece

Contents

Theory and Reviews

Difficulties and Disparities to Distance Learning During Covid-19 Period for Deaf Students –A Proposed Method to Eradicate Inequalities	3
<i>Konstantinos Karampidis, Athina Trigoni, Giorgos Papadourakis, Maria Christofaki, and Nuno Escudeiro</i>	
Wide-Scale Automatic Analysis of 20 Years of ITS Research	8
<i>Ryan Hodgson, Alexandra Cristea, Lei Shi, and John Graham</i>	
Exploring the Barriers of Educational Innovation	22
<i>Aivazidi Marina and Michalakelis Christos</i>	
A Brief Survey of Deep Learning Approaches for Learning Analytics on MOOCs	28
<i>Zhongtian Sun, Anoushka Harit, Jialin Yu, Alexandra I. Cristea, and Lei Shi</i>	

Models

DiKT: Dichotomous Knowledge Tracing	41
<i>Seoungyun Kim, Woojin Kim, Heeseok Jung, and Hyeoncheol Kim</i>	
CompPrehension - Model-Based Intelligent Tutoring System on Comprehension Level	52
<i>Oleg Sychev, Anton Anikin, Nikita Penskov, Mikhail Denisov, and Artem Prokudin</i>	
Learning Logical Reasoning : Improving the Student Model with a Data Driven Approach	60
<i>Roger Nkambou, Janie Brisson, Serge Robert, and Ange Tato</i>	
Checking Method for Fake News to Avoid the Twitter Effect	68
<i>Téo Orthlieb, Hamdi Ben Abdessalem, and Claude Frasson</i>	
Comparing Bayesian Knowledge Tracing Model Against Naïve Mastery Model	73
<i>Vanessa Getseva and Amruth N. Kumar</i>	

Exploring Bayesian Deep Learning for Urgent Instructor Intervention Need in MOOC Forums	78
<i>Jialin Yu, Laila Alrajhi, Anoushka Harit, Zhongtian Sun, Alexandra I. Cristea, and Lei Shi</i>	

Concept Maps

Creating and Visualising Cognitive Maps of Knowledge Diagnosis During the Processing of Learning Digital Footprint	93
<i>Viktor Uglev and Oleg Sychev</i>	

Integrating Knowledge in Collaborative Concept Mapping: Cases in an Online Class Setting	99
<i>Junya Morita, Yoshimasa Ohmoto, and Yugo Hayashi</i>	

An Evaluation of a Meaningful Discovery Learning Support System for Supporting E-book User in Pair Learning	107
<i>Jingyun Wang and Hiroaki Ogata</i>	

Towards Semantic Comparison of Concept Maps for Structuring Learning Activities	112
<i>Carla Limongelli, Carmine Margiotta, and Davide Taibi</i>	

Student Prediction

MOOC <i>Next Week</i> Dropout Prediction: Weekly Assessing Time and Learning Patterns	119
<i>Ahmed Alamri, Zhongtian Sun, Alexandra I. Cristea, Craig Stewart, and Filipe Dwan Pereira</i>	

Internet of Things (IoT) Based Support System for Diabetic Learners in Saudi Arabian High Schools	131
<i>Mona Alotaibi and Mike Joy</i>	

Training Temporal and NLP Features via Extremely Randomised Trees for Educational Level Classification	136
<i>Tahani Aljohani and Alexandra I. Cristea</i>	

Urgency Analysis of Learners' Comments: An Automated Intervention Priority Model for MOOC	148
<i>Laila Alrajhi, Ahmed Alamri, Filipe Dwan Pereira, and Alexandra I. Cristea</i>	

Early Predictor for Student Success Based on Behavioural and Demographical Indicators 161
Efthymoulos Drousiotis, Lei Shi, and Simon Maskell

Predicting Certification in MOOCs Based on Students’ Weekly Activities 173
Mohammad Alshehri, Ahmed Alamri, and Alexandra I. Cristea

Learner Behaviour

Recognizing Novice Learner’s Modeling Behaviors 189
Sungeun An, William Broniec, Spencer Rugaber, Emily Weigel, Jennifer Hammock, and Ashok Goel

Expert, Novice, and Intermediate Performance: Exploring the Relationship Between Clinical Reasoning Behaviors and Diagnostic Performance 201
Alejandra Ruiz-Segura and Susanne P. Lajoie

Agent-Based Simulation of the Classroom Environment to Gauge the Effect of Inattentive or Disruptive Students 211
Khulood Alharbi, Alexandra I. Cristea, Lei Shi, Peter Tymms, and Chris Brown

Investigating Clues for Estimating ICAP States Based on Learners’ Behavioural Data During Collaborative Learning 224
Yoshimasa Ohmoto, Shigen Shimojo, Junya Morita, and Yugo Hayashi

Behaviour Analytics - A Moodle Plug-in to Visualize Students’ Learning Patterns 232
Rita Kuo, Ted Krahn, and Maiga Chang

Toward a Webcam Based ITS to Enhance Novice Clinician Visual Situational Awareness 239
Komi Sodoké, Roger Nkambou, Issam Tanoubi, and Aude Dufresne

Feedback and Personalisation

Flexible Program Alignment to Deliver Data-Driven Feedback to Novice Programmers 247
Victor J. Marin, Maheen Riaz Contractor, and Carlos R. Rivero

Interaction of Human Cognitive Mechanisms and “Computational Intelligence” in Systems that Support Teaching Mathematics 259
Sergei Pozdniakov, Ilya Posov, and Chukhnov Anton

Learning Path Construction Using Reinforcement Learning and Bloom’s Taxonomy	267
<i>Seounghun Kim, Woojin Kim, and Hyeoncheol Kim</i>	
Customizing Feedback for Introductory Programming Courses Using Semantic Clusters	279
<i>Victor J. Marin, Hadi Hosseini, and Carlos R. Rivero</i>	
Voice Privacy with Smart Digital Assistants in Educational Settings	286
<i>Mohammad Niknazar, Aditya Vempaty, and Ravi Kokku</i>	
Selfit – An Intelligent Tutoring System for Psychomotor Development	291
<i>Laurentiu-Marian Neagu, Eric Rigaud, Vincent Guarnieri, Sébastien Travadel, and Mihai Dascalu</i>	
Assessment	
Automated Assessment of Learning Objectives in Programming Assignments	299
<i>Arthur Rump, Ansgar Fehnker, and Angelika Mader</i>	
Ex-Ante and Ex-Post Feature Evaluation of Online Courses Using the Kano Model	310
<i>Daniel Moritz Marutschke and Yugo Hayashi</i>	
Automated Summary Scoring with ReaderBench	321
<i>Robert-Mihai Botarleanu, Mihai Dascalu, Laura K. Allen, Scott Andrew Crossley, and Danielle S. McNamara</i>	
Automated Paraphrase Quality Assessment Using Recurrent Neural Networks and Language Models	333
<i>Bogdan Nicula, Mihai Dascalu, Natalie Newton, Ellen Orcutt, and Danielle S. McNamara</i>	
Groups, Teams, Social, Crowd and Communities	
XGBoost and Deep Neural Network Comparison: The Case of Teams’ Performance	343
<i>Filippos Giannakas, Christos Troussas, Akrivi Krouska, Cleo Sgouropoulou, and Ioannis Voyiatzis</i>	
Using Graph Embedding to Monitor Communities of Learners	350
<i>Fabio Gasparetti, Filippo Sciarrone, and Marco Temperini</i>	

Three Common Group Formations in Online Collaborative Learning	357
<i>Tao Wu and Maiga Chang</i>	
New Horizons on Online Tutoring System Inspired by Teaching Strategies and Learning Styles	364
<i>Karima Boussaha and Samia Drissi</i>	
A Comparative Evaluation of the Effect of Social Comparison, Competition, and Social Learning in Persuasive Technology on Learning	369
<i>Fidelia A. Orji and Julita Vassileva</i>	
Sovereignty by Personalization of Information Search: A Collective Wisdom May Influence My Knowledge	376
<i>Stefano A. Cerri and Philippe Lemoisson</i>	
Games and Gamification	
Confusion Detection Within a 3D Adventure Game	387
<i>Mohamed Sahbi Benlamine and Claude Frasson</i>	
Representation of Generalized Human Cognitive Abilities in a Sophisticated Student Leaderboard	398
<i>Christos Troussas, Akrivi Krouska, Filippos Giannakas, Cleo Sgouropoulou, and Ioannis Voyiatzis</i>	
Learning and Gamification Dashboards: A Mixed-Method Study with Teachers	406
<i>Kamilla Tenório, Bruno Lemos, Pedro Nascimento, Rodrigo Santos, Alexandre Machado, Diego Dermeval, Ranilson Paiva, and Seiji Isotani</i>	
Encouraging Teacher-Sourcing of Social Recommendations Through Participatory Gamification Design	418
<i>Elad Yacobson, Armando Toda, Alexandra I. Cristea, and Giora Alexandron</i>	
Automatic Adaptive Sequencing in a Webgame	430
<i>Tong Mu, Shuhan Wang, Erik Andersen, and Emma Brunskill</i>	
Towards Smart Edutainment Applications for Young Children. A Proposal	439
<i>Adriana-Mihaela Guran, Grigoreta-Sofia Cojocar, and Laura-Silvia Dioşan</i>	
Do Students Use Semantics When Solving Parsons Puzzles? – A Log-Based Investigation	444
<i>Amruth N. Kumar</i>	

Emotions and Affect

Tutorial Intervention’s Affective Model Based on Learner’s Error
 Identification in Intelligent Tutoring Systems 453
*Soelaine Rodrigues Ascari, Andrey Ricardo Pimentel,
 and Ernani Gottardo*

A Recommender System Based on Effort: Towards Minimising Negative
 Affects and Maximising Achievement in CS1 Learning 466
*Filipe D. Pereira, Hermino B. F. Junior, Luiz Rodriguez,
 Armando Toda, Elaine H. T. Oliveira, Alexandra I. Cristea,
 David B. F. Oliveira, Leandro S. G. Carvalho, Samuel C. Fonseca,
 Ahmed Alamri, and Seiji Isotani*

Evaluation Test Generator Using a List of Keywords 481
Doru Anastasiu Popescu, Gabriel Ciprian Stanciu, and Daniel Nijloveanu

Voice Emotion Recognition in Real Time Applications 490
Mahsa Aghajani, Hamdi Ben Abdessalem, and Claude Frasson

Affect-Aware Conversational Agent for Intelligent Tutoring of Students
 in Nursing Subjects 497
Moh’d Abuazizeh, Kristina Yordanova, and Thomas Kirste

Extended Reality

ARDNA: A Mobile App Based on Augmented Reality for Supporting
 Knowledge Exploration in Learning Scenarios 505
*Alessia Genovese, Federica Marino, Francesco Orciuoli,
 and Gennaro Zanfardino*

Extraction of 3D Pose in Video for Building Virtual Learning Avatars 512
Kodjine Dare, Hamdi Ben Abdessalem, and Claude Frasson

A Non-immersive Virtual Reality Application for Children with Autism
 Spectrum Disorder 519
Muhamad Irfan Rosli, Zarina Che Embi, and Junaidi Abdullah


Using Augmented Reality in Computing Higher Education 526
Sarah Alshamrani Alshaikhi and Mike Joy

Author Index 531

Theory and Reviews



Difficulties and Disparities to Distance Learning During Covid-19 Period for Deaf Students –A Proposed Method to Eradicate Inequalities

Konstantinos Karampidis¹ , Athina Trigoni¹, Giorgos Papadourakis¹,
Maria Christofaki¹, and Nuno Escudeiro²

¹ Hellenic Mediterranean University, 71410 Crete, Greece
{karampidis, papadour, mchristof}@hmu.gr

² Instituto Politécnico do Porto, 4000 Porto, Portugal
nfe@isep.ipp.pt

Abstract. During this critical period the humankind faces, one of the most affected daily activities is the education. Teachers and students of all education levels had to adopt new technological means to overcome the distance and to continue their education. Above and beyond the new prospects that this situation arises, also many problems were raised. For example, the special care education e.g., deaf or hard hearing students, face more difficulties than in ordinary situations. These students either read the lips or have to see their interlocutor's whole body, in order to understand what he/she is saying but this nowadays is almost inevitable. In the unlikely scenario that their school is open, all classroom participants including teacher must wear a face mask, while in distance learning they can see only a small window of their teacher and most likely only his/her face. This paper presents the ways of communication of deaf and hard hearing people and proposes a novel educational tool that with the proper use should help them overcome this difficult situation.

Keywords: Deaf students · Distance learning · Sign language · Educational tool

1 Introduction

The COVID-19 pandemic caused the largest disruption of education systems in history, affecting nearly 1.5 billion learners around the world. Depending on the wealth each country has, the closure of schools has affected almost 94% of the student's population, while this number rises up to 99% in middle- or low-income countries [1]. This health crisis intensified foregoing education disparities by decreasing the learning for many of the most vulnerable students i.e., students in poor countries, refugees and students with disabilities.

Different rules were applied for each country e.g., in one country the schools are partially open while in another country they are closed and operate only through distance learning. Closures of schools –especially in lower educational levels- prevents many parents to work. As financial pressures arise all over the world, the pandemic's economic

impact to education will be high. The dramatic loss of the Gross Domestic product (GDP) will obviously affect the funding of the education.

Conversely, COVID-19 pandemic has encouraged innovation within the education sector. New -or less used before crisis- methods like distance learning platforms were adopted in a very short time to substitute school teaching. These innovative solutions may become a permanent option as assistive technology when the world returns to normal situations, but flaws or disadvantages there were also highlighted when these solutions were offered to deaf or hard hearing students.

In this paper we address the difficulties these students face and propose a novel educational tool that with the proper use should help them overcome their disability and make them able to attend online classes. The rest of the paper is structured as follows. In Sect. 2 the ways of communication of the deaf people are presented. In Sect. 3 the proposed novel educational tool is presented and finally in Sect. 4 conclusions and future work are given.

2 Ways of Communication of Deaf People

Deaf people tend to sit opposite to each other and not next to each other in order to communicate. Moreover, in their conversations they keep a greater distance from each other than the non-deaf, since they need to watch not only the signer's face and his expressions but also the movements of his hands. Furthermore, deaf people use their peripheral vision a lot [2]. In their outings or meetings, they choose places with good lighting and a relatively quiet environment so that they can be perceived when they speak but also understand what they are saying, by looking at the lips of their interlocutor.

Deaf people are more diffuse and expressive in their reactions than the non-deaf. The role of extra linguistic elements of communication is more important for them than non-deaf. More specifically, the extra linguistic aspect is a nonverbal communication consisting of gestures, movements, face expressions which are fundamental for them in order to be understood to their interlocutors.

When deaf people want to communicate, they draw attention by touching someone's shoulder or foot. Moreover, a vital element in deaf people's communication is eye contact. When a deaf person wants to communicate with someone else, he makes sure to be in his field of vision and then greets him in Sign Language. Deaf people "touch" each other much more than non-deaf do. When a deaf person wants to participate in a discussion and say something he does so with a touch as in case of attracting attention, while non-deaf people usually use their names to get someone's attention.

The terms "deaf" and "hearing loss" are not used by the deaf people in the way they are used by the non-deaf. Acoustic ratings are irrelevant to deaf culture. The term "DEAF" used in the deaf community clarifies their identity and not their deafness [3].

Regarding deaf student's distance learning we must focus on two different aspects: a) to the course attendance and b) to the active participation of the course. For the first aspect it was mentioned that deaf or hard hearing students must see their interlocutor's face, body and hands. Active participation to the course means that a deaf student must try to draw attention or to join a conversation.

Distance learning platforms are not designed to meet these requirements. Therefore, deaf students cannot actively participate and the educational gap between non-deaf and

deaf students becomes greater. In order to overcome these obstacles and remove the barriers in deaf student's education we propose the adoption of Hercules as an integration to existing distance learning platforms.

3 Hercules – An Integration to Distance Learning Platforms

Hercules is the outcome of the two year EU funded program “International Assisted Communications for Education” [4–6]. The goal of the program was to implement a bi-directional translator from five (5) spoken languages (Portuguese, Slovenian, German, British, Cypriot and Greek) to their respective sign languages and vice versa. Hercules is also used to the EU funded program “InSign - Advancing inclusive education through International Sign”, where one of the outcomes will be an automatic bi-directional translator from International Sign Language to the aforementioned spoken languages [7]. InSign aims to promote the access of deaf students to education, international mobility, and global citizenship by raising awareness to International Sign as a lingua franca to communicate among deaf and non-deaf in international settings.

The translator consists of two modules: a) the text-to-sign module and b) the sign-to-text module. Two main tools were utilized for the text-to-sign translation module i.e., the translator and the configurator. Both tools were developed in Unity and built into WebGL in order to be used online through the most popular internet browsers, but also other builds can be easily implemented for other use.

In the configurator, deaf people and sign language experts add and validate the gestures either online or offline. Users simulate the gesture -through a 3D avatar- that corresponds to the given word by adjusting avatar's arms, hand configuration, body movement, head and facial expression.

All signs entered to the database by an editor, are to be validated by an expert –the validator- which can also correct the movement or send a request -to the editor- to change the animation. Once the words are validated, they become automatically available for the translator tool, which is updated in real time for all users. In the translator, users write text and this is translated to the selected sign language. Each word in the given text is searched in the validated signs database and the database returns – through the 3D avatar- the respective gesture according to the selected country sign language grammar rules.

In order to deploy this bi-directional translator to existing distance learning platforms, the interlocutor's voice must be captured and transformed to text. Therefore, the hearing persons can speak to their microphone and afterwards a voice-to-text module will produce the text required as input to the translator. The 3D avatar then will sign the gesture to the deaf student. Although there will be a time delay due to voice-to-text conversion and text to sign module, the proposed system can be considered as real time. The only limitation relies on the quality of the captured voice and whether the given words are already imported and validated into the database.

On the contrary when a deaf student wants to ask something he can “raise his hand” (built-in feature in most distance learning platforms) and text to everybody else who joins the same meeting. Therefore, the proposed model is shown in Fig. 1.

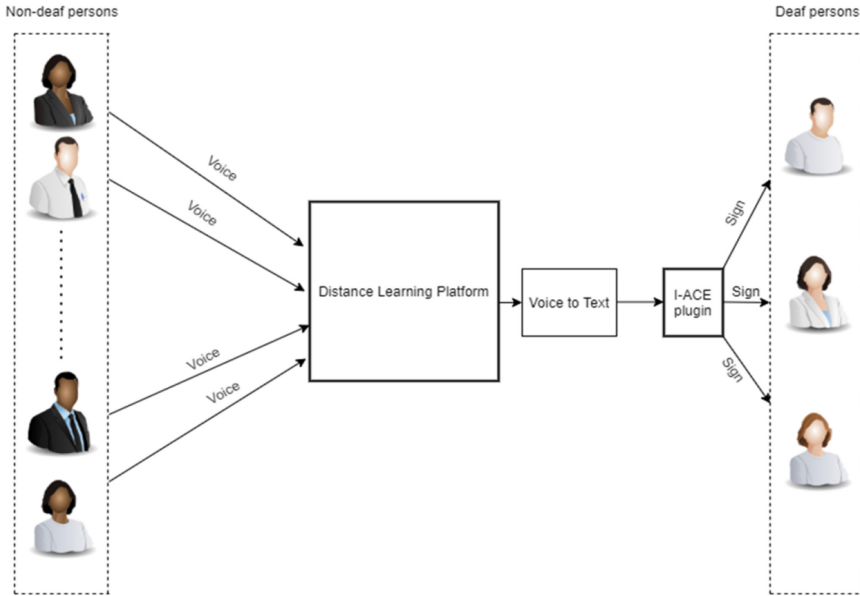


Fig. 1. The proposed architecture

4 Conclusions and Future Work

Nowadays where covid-19 pandemic influences our lives, education system has also been affected. Special care education suffers from this situation and even though distance learning was adopted as a countermeasure, deaf or hard hearing students are facing many difficulties or due to their impairment are excluded from education.

In this paper the difficulties that deaf students face in education due to Covid-19 pandemic were presented. Moreover, a novel educational tool that with the proper use should help them overcome this difficult situation was proposed as an integration to distance learning platforms. We strongly believe that this integration will enhance equality between non deaf and deaf students, minor the education gap that exists and give more learning opportunities to students with hearing disability.

In terms of future work, we plan to implement a plugin to capture voice and translate it – through the 3D avatar- into a sign and integrate it in the major distance learning platforms. The plugin will then be evaluated then by deaf students and sign language experts in order to enhance functionality and offer deaf community a friendly educational tool.




References

1. United Nations: Education during COVID-19 and beyond AUGUST 2020 (2020)
2. Michou, E.: Behaviors and Habits (2013). http://micro-kosmos.uoa.gr/gr/magazine/ergasies_foititon/ettap/2012-13/noimatiki/simperiforeskaisynitheies.htm. Accessed 31 Mar 2021

3. Lampropoulou, V.K.: First educational training package: society and the deaf. *Community and Culture Deaf* (1999)
4. Ward, A., et al.: The international assisted communications for education project IACE. In: 2017 27th EAEEIE Annual Conference EAEEIE 2017, pp. 1–6 (2017)
5. Lymperidi, D., et al.: International assisted communication for education (I-ACE): Greek contribution. In: ERACON Congress & CAREER-EU Conference, pp. 21–27 (2019)
6. Ulisses, J., Oliveira, T., Escudeiro, P.M., Escudeiro, N., Barbosa, F.M.: ACE assisted communication for education: architecture to support blind & deaf communication. In: IEEE Global Engineering Education Conference, EDUCON, Vol. 2018-April, pp. 1015–1023 (2018)
7. InSign (Advancing inclusive education through International Sign). <https://www.uni-siegen.de/zew/insign/insign/index.html.en?lang=en>. Accessed 23 Sep 2020



Wide-Scale Automatic Analysis of 20 Years of ITS Research

Ryan Hodgson¹ (✉) , Alexandra Cristea¹ , Lei Shi¹ , and John Graham²

¹ Department of Computer Science, Durham University, Durham DH1 3LE, UK
ryan.t.hodgson@durham.ac.uk

² Reveela Technologies, Carlisle Square, Newcastle Upon Tyne NE1 6UF, UK

Abstract. The analysis of literature within a research domain can provide significant value during preliminary research. While literature reviews may provide an in-depth understanding of current studies within an area, they are limited by the number of studies which they take into account. Importantly, whilst publications in hot areas abound, it is not feasible for an individual or team to analyse a large volume of publications within a reasonable amount of time. Additionally, major publications which have gained a large number of citations are more likely to be included in a review, with recent or fringe publications receiving less inclusion. We provide thus an *automatic methodology for the large-scale analysis of literature within the Intelligent Tutoring Systems (ITS) domain*, with the aim of *identifying trends and areas of research* from a corpus of publications which is significantly larger than is typically presented in conventional literature reviews. We illustrate this by a novel analysis of 20 years of ITS research. The resulting analysis indicates a significant shift of the status quo of research in recent years with the advent of novel neural network architectures and the introduction of MOOCs.

Keywords: Topic modelling · Epistemological engines · Automatic literature survey

1 Introduction

The considerable volume of research within Intelligent Tutoring Systems (ITS) presents challenges to the quantification of the various fields present within the domain. Conventional literature surveys are typically performed using manual analysis and filtering of available literature and as such are limited in the volume of publications. Additionally, researchers may fail to account for research, which is niche, but may be still important. Surveyors are furthermore likely to include main-stream research only, or research assisting in their argument. Thus, we propose to leverage the novel topic modelling algorithm Top2Vec [1] for the analysis of a large volume of ITS research. Advantages to such an analysis include the volume of ITS research processed, which exceeds that which may be feasible by even a large team of contributors. Additionally, the speed of topic analysis ensures ample time for the further analysis of temporal factors within the corpus and presentation of relationships between any identified topics.

Major contributions of this work are: 1) automatically identifying, for the first time, significant trends in ITS (e.g., temporally, ITS has observed a significant shift in research popularity from Adaptive Hypermedia towards online MOOC platforms; applied architectures and algorithms have shifted significantly towards Deep Learning and applications of Neural Networks); 2) automatically extracted relationships between several ITS topics indicate potential for novel areas of research; 3) we demonstrate the power of the recent Top2Vec algorithm to assist, for the first time, in large scale literature analysis, without limitations presented by conventional probabilistic topic models. Compared to existing studies within ITS, this research provides a *unique overview of the last 20 years of research*, without the bias presented by human-reviewers who may, arguably, ‘cherry-pick’ studies to argue their point. Our research accounts for *all research made available by API resources*.

2 Related Works

Traditional literature surveys in ITS follow a manual process, where identified publications are filtered, resulting in a significantly smaller batch of publications used in the final review. ITS reviews, such as [2], apply a Systematic Literature Review process. They analysed a total of 33 publications, filtered down from an initial corpus of 4,622 papers. These resulting papers are analysed in-depth; however, the exclusion of such a large volume of papers clearly indicates missed opportunities for obtaining insight from the excluded publications. Outside of manual literature surveys, we identified [3], which performed analysis of a larger volume of publications on a quantitative level. The scope of the publication addressed barriers and trends of ITS adoption rather than the trends and relations of the overarching field.

Top2Vec [1] provides a very recent alternative to Bayesian topic models such as PLSA and LDA [4, 5], eliminating the need for pre-defining the topics number and filtering stop words. It leverages language embeddings and enables pre-trained language models to be applied via Doc2Vec [6], BERT [7] or Universal Sentence Encoder [8]. A semantic embedding of joint document-word vectors is generated where the distance between document and word vectors represents semantic association. This ensures that semantically similar documents achieve a smaller distance between each other when compared to dissimilar documents. Resulting embeddings are clustered using the HDBSCAN [9] algorithm into topic clusters, with the hierarchical nature of HDBSCAN ensures automatic identification of the topic number. Given the high dimensionality of document embeddings, clustering requires prior dimensionality reduction through the UMAP [10] algorithm. Top2Vec has been demonstrated to outperform LDA and PLSA when applied to the benchmark 20NewsGroups [11] dataset [1].

3 Corpus Generation

3.1 Data Collection

We performed collection of publication data from the ITS domain via several API resources, over the past 20 years. These consist of the arXiv Preprint Repository, Springer

API, SAGE API, Elsevier API and CORE API [12–16]. We selected query terms based upon a sample of the key phrases presented in the 2000–2020 ITS Conference Proceedings; however, we avoided inclusion of low-level specific terms, to ensure the resulting document distributions were unbiased. To be comprehensive, we collected literature not limited to journals and conferences, but also included book chapters, preprints and academic theses. In total, we collected 5018 documents from 2000–2020. Research from 2000–2020 was selected to provide a wider-scale analysis, beyond that of only the most recent literature, which could assist in evaluating the changes in long-term trends of ITS quantitatively.

Following the collection of raw publication data, it was necessary to filter out results which were not relevant to ITS using Boolean word matching at the abstract level. Given that some terms used within ITS (e.g., adaptive learning) may be confused with general machine-learning terms by a partial matching system, it was deemed necessary to apply absolute string-matching during filtering. Documents were excluded if they failed to contain any instances of the terms within our search query. Given the large volume of research identified, it was necessary to perform this automatically with regex.

3.2 Preprocessing

We performed no preprocessing of the corpus prior to topic analysis, with the aim of maintaining contextual information within generated embeddings. Stop word removal, lemmatisation or stemming was not necessary as detailed by [1], in contrast to LDA, where stop word removal and additional filtering of highly frequent terms may be performed [17] to improve model performance. Language-checks were performed on the corpus to remove any non-English publications, which could impact model performance. For this we applied the `langdetect` [18] Python library. Following filtering of non-relevant results, the corpus size was of 3898 documents abstracts and titles, which we combined for our analysis. Given the limitations of access to publications, we were only able to collect abstracts, as access to full-text results was limited.

4 Analysis

The methodology for our analysis involves the modelling of topics within our corpus using the Top2Vec algorithm [1] and the subsequent analysis of the resulting topics in relation to temporal range and relationships between topics. Our work is available at¹.

4.1 Topic Modelling Approach

Top2Vec identifies semantic relationships through learning of a distributed representation via the Doc2Vec algorithm [6]. Alternatively, pretrained models may be applied including Universal Sentence Encoder [8] or the BERT [7] transformer network. However, our experiments identified that Doc2Vec embeddings fine-tuned to our corpus outperformed these, likely due to the presence of frequent domain-specific language

¹ <https://github.com/ryanon4/epistemological-topic-modelling>.

within ITS research. Additionally, the resulting clusters may be visualised to provide an understanding of clustering results as presented in Fig. 1.

HDBSCAN labels a portion of the documents within our corpus as noise, which we removed prior to presentation in Fig. 1, leaving all topics without any noise present. Given that ground truth labels were not available for clustering evaluation, we applied Silhouette scoring [19] using Euclidean distance and achieved a score of 0.37 when accounting for all 33 topics. When reducing this to only account for topics relevant to ITS, this increased to 0.42. In total HDBSCAN identified 33 separate topics.

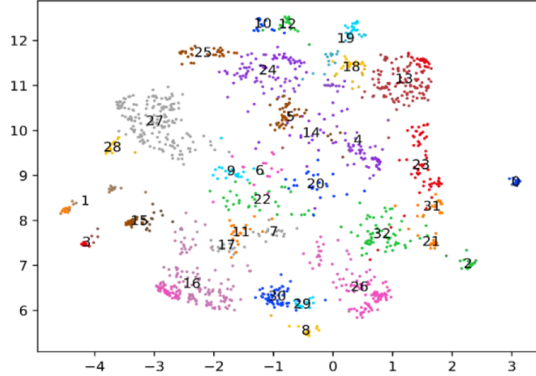


Fig. 1. Clustering results following noise removal with topic labels assigned

For higher accuracy, we further performed manual evaluation of the resulting topics at a qualitative level and filtered non-relevant topics which may have arisen, to ensure that only those relevant to ITS remain. Filtering consisted of analysis of the topic-word distributions for each topic, and exclusion was performed when a significant level of noise or non-informative terms were identified. A label was manually assigned to relevant topics, based upon the word distributions they entailed. These are detailed in Table 1.

Resulting topics indicate 11 highly coherent and relevant topics out of the 33 topics identified by Top2Vec. Topics were excluded where word-distributions contained unrelated terms and could not be clearly labelled, or the distributions were related, however contributed less to our analysis. Removal of non-relevant documents and noise reduced the total corpus size for our analysis to 1223 documents. Topic 13 indicates the types of architectures and models present within research and as such does not serve as a useful topic label for the corpus. We investigated this separately via a temporal analysis in Sect. 4.3.

4.2 Topic Analysis

Results from topic modelling with Top2Vec identified a range of ITS relevant topics within our corpus which we present ordered by the size of each identified topic in Table 1. These range from high-level areas which may entail different approaches within,

to specific areas of research relevant to ITS. Of the identified areas, high-level topics correlate to a larger volume of entailing documents with specific topics containing a lower volume of documents. The clustering of publications using the HDBSCAN algorithm leads to the assigning of single topic labels to each document, meaning that unlike probabilistic models like LDA, documents may not belong to multiple topics and therefore more specific or low-level topics typically contain fewer documents. Within this

Table 1. Identified topic-word distributions by Top2Vec, topics deemed relevant to ITS by qualitative analysis

Topic terms	Topic ID	Topic label (Manually-determined)
Hypermedia, aeh, aehs, ims, adaptive, adaptation, adaptivity, navigation, personalization, links, specification	0	Adaptive educational hypermedia
Dialogue, tutoring, natural, intelligent, language, tutorial, automatically, conversational, apos, corpus, medical, quot	2	Intelligent dialogue systems
Agent, animated, emotion, affective, emotional, pedagogical, agents, emotions, facial, conversational, apos	5	Pedagogical agents
Moodle, lms, source, management, open, centre, lectures, platforms, basic, dashboards, assignments	7	Learning Management Systems
Peer, assistance, collaborative, conditions, learned, tutor, collaboration, dialogue, actions, cscl	8	Computer-supported collaborative learning
Moocs, massive, mooc, dropout, forum, open, engaging, courses, videos, rates	12	MOOCs
Bayesian, networks, fuzzy, logic, artificial, diagnosis, intelligence, intelligent, neural, tutoring	13	Machine learning model types and algorithms
Simulations, simulation, intelligent, animated, virtual, training, multimedia, agents, reality	14	Simulations
Games, game, serious, play, agent, interact, intelligent, bring, initiative, metrics, simulation	17	Gamification
Essay, scoring, essays, automatic, grading, writing, automated, English, language, neural	19	Grading and assessment scoring
Recommender, recommendation, personalization, links, personalized, java, hypermedia, experiments, lecture, adapting	28	Recommender systems

section we ensure that all references are made using publications present within our corpus. Given the criteria applied during data collection, research discussed may include pre-print or thesis research which has not been peer-reviewed.

Topic 0 – Adaptive Educational Hypermedia

This topic represents the high-level area of Adaptive Educational Hypermedia Systems (AEHS). These may be defined as adapting content to fit the goals and needs of a user or student [20]. Documents within our corpus labelled under this topic typically investigate web-based approaches for adaptive tutoring [21] and relate closely to other ITS areas including Learning Management Systems (LMS) and MOOCs. We identified several approaches involving neural networks of which [22–24] are a sample, as well as framework proposals for the building of e-learning platforms [25, 26].

Topic 2 – Intelligent Dialogue Systems

Intelligent Dialogue Systems (IDS) typically investigate the application of conversational agents, applied to assisting in the pedagogical process. Sample documents involve the application of conversational agents to address tutoring of concepts and principles with students for both physics and programming [27, 28].

Topic 5 – Pedagogical Agents

We identified considerable interest in research related to the impact of pedagogical agents [29–32] and how the presentation of these agents may impact success within ITS. Other documents more closely correlate to emotion recognition through the assessment of learner feedback [33]. Adaptation of pedagogical agents in response to emotional queues are frequent within this topic [34], however research may alternatively investigate the impact of perceived emotions of pedagogical agents [35].

Topic 7 – Learning Management Systems

This topic entails the high-level area of Learning Management Systems. Within this topic, a significant volume of research relates to e-learning platforms such as Moodle [36] and includes proposals for the modification of such platforms to adapt to user learning styles and requirements. Interestingly, the majority of publications present within this topic avoid architectural specifications or computing-based terminology, and instead typically provide case studies of the implementation of existing LMS.

Topic 8 – Computer Supported Collaborative Learning

Documents assigned to this topic generally relate to Computer Supported Collaborative Learning (CSCL). Relations within this area include pedagogical agents [37], although generally there were fewer instances of bridging between the identified topics.

Topic 12 - MOOCs

Massive Open Online Courses (MOOCs) are a relatively recent aspect of ITS research, and we identify our earliest instance of this within our corpus in 2013 [38]. We identify 38% of research in this topic entailing learning analytics [39–41], which involves the

wealth of data provided by MOOC platforms. This data may be applied to dropout prediction and forecasting of MOOC platforms [42, 43], and we identify 11% of documents involving dropout prediction.

Topic 13 – Architectures and Algorithms

Documents assigned to this topic are more closely associated with implementation and architectures of models than ITS processes. We identify applications of fuzzy logic [44–46] comprising 25% of the topic, with 11% discussing or applying neural networks [47, 48] and 8% through clustering [49]. Given that algorithms and architectures will be likely present in the wider corpus we perform a temporal analysis of the entire corpus in Sect. 4.3.

Topic 14 - Simulations

This topic represents research involving the simulation of learning environments and simulated agents. We identify articles relating to pedagogical agents [50], CSCL [51] and adaptive hypermedia [52] within this topic. While documents relate to other ITS areas the majority discuss the simulation of environments to assist with learning. Instances of simulation include resource allocation training for police forces [53] and the use of virtual reality simulated environments [54, 55].

Topic 17 - Gamification

Publications applying gamification within ITS fall within this topic, with research contributing to the use of game mechanics for positive educational outcomes. Games may be applied to assisting learning in STEM subjects [56, 57] within virtual learning environments or in the tutoring of programming [58].

Topic 19 – Grading and Essay Scoring

This topic relates to the grading of work with ITS and is most directly associated with the area of Automatic Essay Scoring (AES), however other areas of grading exist within the topic. We identify 42% of documents discuss essay scoring directly, with research investigating short question grading [59] and essay scoring [60, 61]. The grading of text is not the only research within this topic however, and we identify unpublished research in the automated grading of map sketches [62] within our corpus sample. Within this topic we identify 18% of publications applying neural networks, while 6% apply ontologies and 5% applying Bayesian learning.

Topic 28 – Recommender Systems

Documents relating to recommender systems comprise this topic, which is the smallest relevant topic identified by our analysis. Research within this area present systems for the recommendation of courses in MOOC platforms [63] and adaption of learning environments using recommender systems [64] amongst others. This topic can be closely linked to several of our identified topics including adaptive educational hypermedia, computer supported collaborative learning and MOOC systems.

4.3 Temporal Analysis of ITS

We visualise the changes in resulting topics from our analysis in Fig. 2. These are normalised per-year to eliminate influence by changes in yearly publication volume.

Resulting temporal distributions generally correlate to the sizes of our identified topics, with documents assigned to Adaptive Educational Hypermedia forming the largest portion of research from 2001–2015. Other topics outside of these generally fail to form more than 20% of research interest prior to 2016, where research into MOOCs overtakes other topics to become the most dominant topic within our corpus. This considerable change in interest towards MOOC platforms may be influenced by the wealth of data obtained and provided by such platforms, with public datasets such as [65] allowing researchers easy access to data to contribute to the field, and the general trends towards ‘Big Data’ application and research. A decrease in popularity of AEH, IDS and several other topic types may further contribute to the adoption of MOOC type research, which may provide more easily accessible datasets and feature ranges.

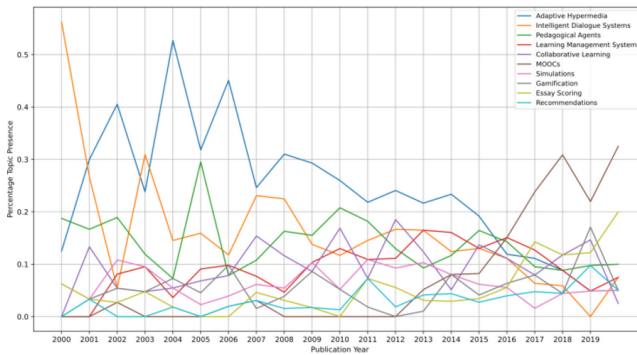


Fig. 2. Temporal changes in topics from 2000–2020. Normalised by number of publications each year

We present the occurrences of different algorithms and architectures in Fig. 3. Results indicate the consistent presence of ontologies within research throughout 2000–2020, while applications of other algorithms identified by Top2Vec fluctuate in popularity. Most notably, an increase in presence of both clustering and neural networks is observed from 2010–2020 within the entire corpus. In recent years (2019–2020), the volume of research discussing neural networks increases considerably. This may indicate the general trends of the wider computing field and may be attributable to recent novel algorithms such as the transformer network and BERT [7].

4.4 Topic Graph Relationships

For further analysis, we construct a network of relationships between ITS topics, as depicted in Fig. 4. These are constructed using the cosine similarity between average document embeddings of each topic. Average document embeddings were generated using the Doc2Vec document embeddings of all documents assigned to a topic by Top2Vec.

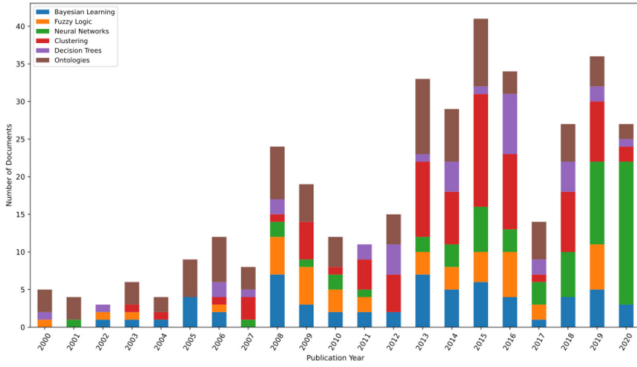


Fig. 3. Algorithmic and architectural presence based upon publication year.

We assign connections between topic nodes using the three highest scoring similarity relationships for each topic.

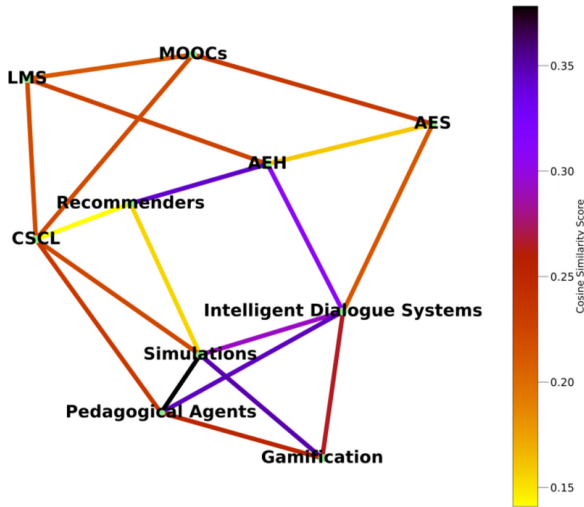


Fig. 4. Relationships of relevant topics based on cosine similarity between average of topic vectors.

Relations between Pedagogical Agents, Simulations, IDS and Gamification reflect the links present within the corpus, where agents may be presented to users in a graphical manner. The cosine similarity scores between these four topics are the highest within the network and demonstrate the linking themes and interoperability that these areas present. In the case of AEH and Simulation, research investigating the adaptation of simulated agents in response to user or student is present within both topic corpuses. Further high scoring similarities are observed between AEH and Recommender Systems, wherein publications may discuss the adaptation of recommender systems dynamically, based upon user responses and performance. Given that recommender systems may be

closely attributed to adaptation of systems to user input, we argue that this is represented through the association with Adaptive Hypermedia (AH) and Learning Management Systems (LMS) through the recommendation of course content.

LMS is additionally closest associated to CSCL and MOOC systems. We argue that LMS and MOOC systems are by nature closely linked (with MOOCs forming a subset of LMS) and therefore documents within these topics may share semantic terms. Both LMS and MOOC systems research incorporate aspects of collaborative learning within our corpus. A link between MOOCs and Automated Essay Scoring research is present, being the strongest link for AES, which is one of the weakest scoring topics, in terms of cosine similarity with other topics. This reflects how many of the topics present within the corpus offer a degree of interoperability, which is less so in the case of AES.

5 Discussion and Conclusion

The application of the novel Top2Vec [1] algorithm to topic analysis of the ITS literature enables an overview of the development as well as current research field. Contrary to well-known approaches, such as LDA [5], the algorithm requires fewer preprocessing steps and therefore demonstrates potential in application to a range of epistemological research without expert knowledge. Furthermore, this analysis approach ensures a significantly higher volume of research can be processed and analysed compared to manual review types. Our analysis of the resulting topics identified contributes to an understanding of the relationships between topics and the volume of research various areas contain.

General findings from our investigation indicate research involving Adaptive Hypermedia to comprise the highest volume of research overall. This area presents a high level of interoperability with others, such as with research applying Simulation and Recommender Systems in an adaptive manner, based on user input. Temporally, Adaptive Hypermedia entails the largest portion of ITS research up until 2016, where it is overtaken by MOOC research. Given that our analysis accounts for all research from 2000–2020, there exists further opportunity for a dedicated analysis of the more recent years publications only, in order to form a better understanding of reasons for MOOC research popularity, and identification of potential new areas of research within. Topics such as Automatic Essay Scoring are clearly underrepresented and may deliver promising avenues of future research – especially as some of this research seems yet unpublished. Temporally, we identify a shift in research in recent years (2016–2020) with a considerable increase in interest of MOOC systems, and applications of neural network architectures to research within these years. This, we argue, is likely the result of the increase in availability of data generated by MOOC systems, which achieve a considerable throughput of users and therefore volume of data. In the case of applications of neural network, we argue the interest spike follows the considerable improvements made in recent years for transformer-based and pre-trained networks. As a final note on our methodology, we identify limitations in the applications of abstracts only within our corpus, whereas structured full-text data may have provided valuable insight into topics of separate sections (e.g., related works, methodologies). We are considering analysing specifically further research targets in papers, both temporally, to understand to which

extent the targets have already been reached, or if they are open, as well as in terms or recent year gaps to fill. Key phrases for our search were based on the ITS Conference only. This may be a limitation, and other possible variations could be considered. However, given that the extraction was over the last 20 years, so conforming exactly to our target time period, we can say with some confidence that these results clearly show the progress of ITS research during the past 20 years from an ITS conference perspective.

References

1. Angelov, D.: Top2Vec: distributed representations of topics. <https://arxiv.org/abs/2008.09470>.
2. Dermeval, D., Paiva, R., Bittencourt, I., Vassileva, J., Borges, D.: Authoring tools for designing intelligent tutoring systems: a systematic review of the literature. *Int. J. Artif. Intell. Educ.* **28**, 336–384 (2018). <https://doi.org/10.1007/s40593-017-0157-9>
3. Nye, B.: Intelligent tutoring systems by and for the developing world: a review of trends and approaches for educational technology in a global context. *Int. J. Artif. Intell. Educ.* **25**, 177–203 (2014). <https://doi.org/10.1007/s40593-014-0028-6>
4. Hofmann, T.: Probabilistic latent semantic indexing. *ACM SIGIR Forum* **51**, 211–218 (2017)
5. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
6. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning, Beijing, China, pp. 1188–1196, *JMLR.org*. (2014)
7. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 4171–4186 (2019)
8. Cer, D., et al.: Universal sentence encoder (2018)
9. Campello, R., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7819, pp. 160–172. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37456-2_14
10. McInnes, L., Healy, J., Saul, N., Großberger, L.: UMAP: uniform manifold approximation and projection. *J. Open Source Softw.* **3**, 861 (2018)
11. Lang, K.: NewsWeeder: learning to filter netnews. In: *Machine Learning Proceedings 1995*, pp. 331–339 (1995)
12. arXiv API Access, arXiv e-print repository. <https://arxiv.org/help/api/index>
13. Springer API. <https://dev.springernature.com/>
14. Text and Data Mining on SAGE Journals: SAGE Journals. <https://journals.sagepub.com/page/policies/text-and-data-mining>
15. Elsevier Developer Portal. <https://dev.elsevier.com/>
16. CORE API. <https://core.ac.uk/services/api/>
17. Schofield, A., Magnusson, M., Thompson, L., Mimno, D.: Understanding text pre-processing for latent Dirichlet allocation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pp. 432–436 (2017)
18. Mimino666/langdetect. <https://github.com/Mimino666/langdetect>
19. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987)
20. Brusilovsky, P.: Adaptive hypermedia: from Intelligent tutoring systems to web-based education. In: Gauthier, G., Frasson, C., VanLehn, K. (eds.) ITS 2000. LNCS, vol. 1839, pp. 1–7. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45108-0_1

21. Noguera, J., Ayeni, F., Okuboyejo, S., Adusumi, S.: Towards a Web Based Adaptive and Intelligent Tutoring System. *Int. J. Comput.* **1** (2017)
22. Bayasut, B., Pramudya, G., Basiron, H.: ULUL-ILM: the design of web-based adaptive educational hypermedia system based on learning style. In: 13th International Conference on Intelligent Systems Design and Applications (2013)
23. Bayasut, B., Pramudya, G., Basiron, H.: The application of multi layer feed forward artificial neural network for learning style identification. *Adv. Sci. Lett.* **20**, 2180–2183 (2014)
24. Mota, J.: Using learning styles and neural networks as an approach to elearning content and layout adaptation (2008)
25. Chimalakonda, S., Nori, K.: An ontology based modeling framework for design of educational technologies. *Smart Learn. Environ.* **7**, 28 (2020). <https://doi.org/10.1186/s40561-020-00135-6>
26. Gouli, E., Kornilakis, H., Papanikolaou, H., Grigoriadou, M.: Adaptive assessment improving interaction in an educational hypermedia system. In: Proceedings of the PanHellenic Conference with International Participation in Human-Computer Interaction, pp. 217–222 (2001)
27. Jordan, P., et al.: Interactive event: the Rimac tutor - a simulation of the highly interactive nature of human tutorial dialogue. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS (LNAI), vol. 7926, pp. 928–929. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39112-5_148
28. Lane, H., VanLehn, K.: A dialogue-based tutoring system for beginning programming. In: FLAIRS Conference, pp. 449–454. AAAI Press, Menlo Park, CA (2004)
29. Moreno, R., Mayer, R.: Life-like pedagogical agents in constructivist multimedia environments: cognitive consequences of their interaction. In: Conference Proceedings of the World Conference on Educational Multimedia Hypermedia, and Telecommunications (ED-MEDIA), pp. 741–746 (2000)
30. Moundridou, M., Virvou, M.: Evaluating the impact of interface agents in an intelligent tutoring systems authoring tool. In: Advances in Human-Computer Interaction I: Proceedings of the Panhellenic Conference with International Participation in Human-Computer Interaction, pp. 371–376. Typorama Publications, Patras, Greece (2001)
31. Chou, C., Chan, T., Lin, C.: Redefining the learning companion: the past, present, and future of educational agents. *Comput. Educ.* **40**, 255–269 (2003)
32. Baylor, A., Kim, Y.: Pedagogical agent design: the impact of agent realism, gender, ethnicity, and instructional role. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 592–603. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30139-4_56
33. Akputu, O., Seng, K., Lee, Y., Ang, L.: Emotion recognition using multiple kernel learning toward E-learning applications. *ACM Trans. Multimed. Comput. Commun. Appl.* **14**, 1–20 (2018)
34. Sharma, P., Joshi, S., Gautam, S., Maharjan, S., Filipe, V., Reis, M.: Student engagement detection using emotion analysis, eye tracking and head movement with machine learning. <https://arxiv.org/abs/1909.12913>
35. Liew, T., Mat Zin, N., Sahari, N.: Exploring the affective, motivational and cognitive effects of pedagogical agent enthusiasm in a multimedia learning environment. *Hum. Cent. Comput. Inf. Sci.* **7**, 9 (2017). <https://doi.org/10.1186/s13673-017-0089-2>
36. Limongelli, C., Sciarone, F., Vaste, G.: Personalized e-learning in Moodle: the Moodle_LS system. *J. e-Learn. Knowl. Soc.* **7**, 49–58 (2011)
37. Mørch, A., Jondahl, S., Dolonen, J.: Supporting conceptual awareness with pedagogical agents. *Inf. Syst. Front.* **7**, 39–53 (2005)
38. Monson, R., Bunney, D., Lawrence, T.: MOOCs. learning analytics and learning advisors. *eCULTURE* **6**, 9–22 (2013)

39. Onah, D., Pang, E., Sinclair, J., Uhomoihibi, J.: Learning analytics for motivating self-regulated learning and fostering the improvement of digital MOOC resources. In: Auer, M.E., Tsiatsos, T. (eds.) IMCL 2018. AISC, vol. 909, pp. 14–21. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11434-3_3
40. Alexandron, G., Yoo, L., Ruipérez-Valiente, J., Lee, S., Pritchard, D.: Are MOOC learning analytics results trustworthy? With fake learners, they might not be! *Int. J. Artif. Intell. Educ.* **29**, 484–506 (2019). <https://doi.org/10.1007/s40593-019-00183-1>
41. Bystrova, T., Larionova, V., Sinitsyn, E., Tolmachev, A.: Learning analytics in massive open online courses as a tool for predicting learner performance. *Voprosy obrazovaniya/Educ. Stud. Mosc.* **4**, 139–166 (2018)
42. Alamri, A., et al.: Predicting MOOCs dropout using only two easily obtainable features from the first week's activities. In: Coy, A., Hayashi, Y., Chang, M. (eds.) ITS 2019. LNCS, vol. 11528, pp. 163–173. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22244-4_20
43. Gardner, J., Brooks, C.: Dropout model evaluation in MOOCs. <https://arxiv.org/abs/1802.06009v1>
44. Kornilakis, H., Papanikolaou, K., Magoulas, G.: Fuzzy inference for student diagnosis in adaptive educational hypermedia. In: Vlahavas, I.P., Spyropoulos, C.D. (eds.) SETN 2002. LNCS (LNAI), vol. 2308, pp. 191–202. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46014-4_18
45. Sgrò, F., et al.: A neuro-fuzzy approach for student module of physical activity ITS. *Procedia Soc. Behav. Sci.* **9**, 189–193 (2010)
46. Gumińska, M., Madejski, J.: Assessment of the didactic measurement results using FCM type networks. *Arch. Mater. Sci. Eng.* **39**, 45–52 (2009)
47. Rengasari, N., Venkatesh, R., Maheswari, N.: Intelligent tutoring system: predicting students results using neural networks. *JCIT* **3**, 22–26 (2008)
48. Maffon, H., et al.: Architecture of an intelligent tutoring system applied to the breast cancer based on ontology, artificial neural networks and expert systems. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.681.8541>
49. Hogo, M.: Evaluation of e-learning systems based on fuzzy clustering models and statistical tools. *Expert Syst. Appl.* **37**, 6891–6903 (2010)
50. Soliman, M., Guetl, C.: Simulating interactive learning scenarios with intelligent pedagogical agents in a virtual world through BDI-based agents. *Int. J. Eng. Pedagogy (iJEP)* **3**, 41 (2013)
51. Buche, C., Querrec, R., De Loor, P., Chevallier, P.: MASCARET: pedagogical multi-agents systems for virtual environment for training. In: Proceedings 2003 International Conference on Cyberworlds, pp. 423–430 (2003)
52. Huang, L., Ho, C.: Building and adaptive learning mechanism to assist eLearning students. In: AMCIS 2009 Proceedings, p. 201 (2009)
53. Furtado, V., Filho, J.: A multi-agent simulator for teaching police allocation. In: Proceedings of the National Conference on Artificial Intelligence, pp. 1521–1528 (2005)
54. Giuffra, C., Silveira, R.: An agent based model for integrating intelligent tutoring system and virtual learning environments. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) IBERAMIA 2012. LNCS (LNAI), vol. 7637, pp. 641–650. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34654-5_65
55. Papagiannakis, G., et al.: MAGES 3.0: Tying the Knot of Medical VR. ACM SIGGRAPH 2020 Immersive Pavilion. Association for Computing Machinery, New York (2020)
56. Terracina, A., Berta, R., Bordini, F., Damilano, R., Mecella, M.: Teaching STEM through a role-playing serious game and intelligent pedagogical agents. In: 2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT), pp. 148–152 (2016)
57. Steinmaurer, A., Pirker, J., Gütl, C.: sCool - game based learning in STEM education: a case study in secondary education. In: Auer, M.E., Tsiatsos, T. (eds.) ICL 2018. AISC, vol. 916, pp. 614–625. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11932-4_58

58. Hooshyar, D., Binti Ahmad, R., Wang, M., Yousefi, M., Fathi, M., Lim, H.: Development and evaluation of a game-based Bayesian intelligent tutoring system for teaching programming. *J. Educ. Comput. Res.* **56**, 775–801 (2018)
59. Saha, S., Dhamecha, T., Marvaniya, S., Foltz, P.: (PDF) Joint multi-domain learning for automatic short answer grading. https://www.researchgate.net/publication/331343422_Joint_Multi-Domain_Learning_for_Automatic_Short_Answer_Grading
60. Cozma, M., Butnaru, A., Ionescu, R.: Automated essay scoring with string kernels and word embeddings. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 2: Short Papers (2018)
61. Liu, M., Wang, Y., Xu, W., Liu, L.: Automated scoring of Chinese engineering students' english essays. *Int. J. Distance Educ. Technol.* **15**, 52–68 (2017)
62. Bhat, A.: Sketchography - automatic grading of map sketches for geography education (2017)
63. Hou, Y., Zhou, P., Wang, T., Yu, L., Hu, L., Wu, D.: Context-aware online learning for course recommendation of MOOC big data. *ArXiv. abs/1610.03147* (2016)
64. Demertzi, V., Demertzis, K.: A hybrid adaptive educational eLearning project based on ontologies matching and recommendation system. <https://arxiv.org/abs/2007.14771>
65. Kumar, S., Zhang, X., Leskovek, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1269–1278. ACM, New York (2019)



Exploring the Barriers of Educational Innovation

Aivazidi Marina (✉) and Michalakelis Christos

Harokopio University, Moschato, Greece
maivazidi@hua.gr

Abstract. The study of innovation has been thoroughly investigated over the past five decades by many researchers and organizations. Educational innovation, in particular, has been studied since the 1970s more systematically. Educational innovation, its adoption and implementation have been studied not only by various researchers, namely Fullan, Westera, Cohen and Ball, but different organizations, such as the Organization for Economic Cooperation and Development (OECD) as well. However, its implementation constitutes a very demanding task. This paper addresses the most crucial suspending factors that may hinder innovation implementation in education through recent and older literature review. The findings of our study include factors that are related to educators, parents, students and the educational context in general. This paper is part of a doctorate dissertation which is currently in progress.

Keywords: Educational innovation · Innovation barriers · Innovation adoption

1 Introduction

Innovation, as Rogers (2003) claims, is an idea, practice or product which is perceived as something new by any individual or institution wishing to adopt it. International scientific dictionaries attribute the term innovation the meaning of the introduction of a new idea, method, technology or product. Oslo and Frascati manuals describe innovation as a process which leads to the creation of new products and methods or the improvement of those already existing. The OECD and Mitchell (2003) adopt a similar approach to the definition of the term of educational innovation. Educational innovation, in particular, is defined as the implementation of new and upgraded ideas, methods and knowledge.

Educational innovation can be gradual when entirely educational and organizational changes take place (Westera 2004). Previous studies revealed that educational innovation should not be identical to educational reform or change that has not undergone new or improved ideas, methods or practices (King and Anderson 2002). In addition, according to Fullan (1991) and Dakopoulou (2008), implementation of new instructive approaches as well as the use of new instructive means that are conducive to the development of new attitudes as regards education, constitute educational innovation. It is, therefore, evident that the term does not reflect any educational change, but the adoption of novel or enhanced methods and technologies.

Another point that is worth mentioning is that innovation is not a fact but a process and it should be approached as such with a view to its effective diffusion (Spyropoulou et al. 2008). This is, undoubtedly, no easy task, given the fact that it is dependent both on the institutions adopting it and the context in which it is diffused. This study examines the factors that negatively affect the diffusion of educational innovation along with the way novel or enhanced practices and technologies are faced by educators and institutions where educators work.

2 Educational Innovation and Factors Inhibiting Its Adoption

It is common sense that the teachers and the learners are the protagonists of the Education Process and consequently of the Educational Innovation Process. Hinojosa et al. (2010) agreed with prior studies e.g. (Barber and Mourshed 2007) on the crucial role of teachers for the innovation's implementation. Specifically they state that teachers should have the ability to develop and apply the appropriate knowledge for problems solving and the necessary communication skills in order to prepare the learners for the knowledge society.

However we don't have to underestimate the significance of the learning environment and the learning methods and their impact on the evaluation of education and specifically on the educational innovation. Kearney et al. (2016) evaluated the classroom climate and its effect on academic activities and support that the classroom climate analysis contributes in the development of positive relationships within the learners and teachers. Troussas et al. (2020) applied Artificial Neural Network and the Weighted Sum Model in order to develop and test on intelligent tutoring system based on the collaborative learning styles recommendation.

In addition, new learning methods enhance the educational process and become a part of educational innovation. Troussas et al. (2020) observed that mobile learning and game based learning advance the knowledge level of students. Shulman and Shulman (2004) argued that the Vision, the motivation and Understanding, the practice, the reflection, the community constitute necessary characteristics for teachers who are oriented to the innovation.

According to literature review, educational innovation constitutes the development and adoption of novel or enhanced tools and technologies in education. Indeed the need for ongoing improvement in education is interrelated with the search and introduction of innovations in educational systems. The contribution of innovation to education is significant, as OECD report in one of their studies, for numerous reasons. First and foremost, it can underline the value of education and, in particular, be conducive to the improvement of pedagogic outcomes and the quality of education. It can further expand knowledge accessibility and benefits of pedagogic outcomes with the aim of facilitating the adjustment of educational systems to meet the demands of an ever-changing society. The need for continuous improvement is closely connected to the quest and introduction of innovation in educational systems. The development and implementation of educational innovation is a very demanding venture, though, as mentioned in recent and past papers.

Cohen and Ball (2006), who studied the barriers in the implementation of educational innovation, report that the designers and users are involved in the innovative process,

thereby focusing their study on them. Evers et al. (2002) also focus their study on educators, whom they view as users of innovation. Cohen and Ball (2006) added a few more to those involved in the process, namely those who adopt innovation and specific legal entities, such as schools, federal services and states. They also claim that the adoption and use of innovation are two very different things and the more complicated innovation becomes, the greater their difference. Our study mainly focuses on educators, who constitute the users of innovation in education, as they are the ones to implement any innovation in the educational process, either because it is required by educational institutions or due to the fact that implementation of such innovative processes is part of their options and initiatives.

Lack of adequate experience in implementation of innovation, lack of preparation for implementation on behalf of the teachers or lack of time on behalf of the educators to exchange ideas on the implementation of an educational innovation were the basic barriers in the implementation of an educational program in Holland, according to Evers et al. (2002). The educators who took part in the program were faced with some problems, which adversely affected the effectiveness of the program, and were primarily attributed to the psychological syndrome of exhaustion faced by some educators to a certain extent at some point in their careers because of the pressure felt to perform their duties. Another interesting finding by Evers et al. (2002) was that the older the teachers, the greater the emotional exhaustion they suffered from, despite their extensive experience. This was due to the implementation of innovation, which was a tremendous change to them. The negative attitude to the implementation of the innovative program resulted in the teachers' low self-esteem, which in turn, made them resort to traditional teaching practices.

Lack of motivation, according to Cohen and Ball (2006) is another factor that may contribute to the ineffectiveness of innovation implementation. On the contrary, motivating teachers could reduce their resistance to change, which, as previously mentioned, may be linked to excessive stress felt after any change in the educational process. Cohen and Ball (2006), though, maintain that the failure of innovation implementation is the result of ineffective organization as well as the complexity and heterogeneity of the educational context the innovation is aimed at. They continue saying that the different contexts of educational innovation can influence the design and diffusion of innovation, and that motivation for the adoption and implementation of educational innovation is inextricably connected to school success. The aforementioned factors that inhibit successful implementation of innovation are primarily related to educators. However, the role of parents, who indirectly participate in the educational process, cannot be disregarded, especially for pre-school and primary education.

Heich (2017) reports that the fact that nowadays both parents contribute financially to the family budget, as opposed to the past, has influenced parents' attitude to the introduction of educational innovation. In fact, parents are fully acquainted with educational methods they used when they were students, and due to lack of time and overloaded schedules, they are suspicious of novel educational methods, cannot understand the changes they could bring about and discourage their children from accepting their implementation. Heich's words: "if parents don't buy", then children will also face

innovation negatively, is characteristic of how parents negatively influence implementation. No matter how important the parents' role is, however, the decisive factor for the implementation of educational innovation is the teachers' attitude (Morris 1985).

More specifically, Hurst (1978) mentioned that availability of information for innovation, users' willingness, sustainability and resources for implementation of innovation, consequences, cost, efficiency and potential pilot test of innovation are the fundamental criteria for users of innovation to decide whether they will go along with its implementation or not. The cost criterion was introduced by Doyle and Ponder (1977) along with the conditions of different classes and effectiveness of their function. The degree of effectiveness of an educational innovation is related to the degree of meeting the needs of society (Long 1973) as well as the degree of understanding those needs and finding alternative solutions (Karmel 1973).

Cohen and Ball (2006) attribute the failure of implementation of innovation to their design. They believe that lack of meticulous design is a suspending factor. Another barrier to innovation is the inappropriate environment for implementation and this is something to be taken into consideration during the planning stage. Nevertheless, apart from the design, the designers' systematic support as well as the development of strategies that render innovation self-preserved, are equally important. To add to that, the problems of implementation should be examined and modifications should be made whenever required.

It is clear that the environment, teachers, parents and designers can pose potential barriers to innovation or positively contribute to its implementation. Yet, students should also be taken into account in the design stage (Evers et al. 2002), as they are the recipients of innovation and should accept it. It is blatantly obvious, therefore, that what may constitute a barrier in implementing educational innovation can be overcome, on condition the designers and promoters of innovation take all the factors which may prove inhibitive to its implementation into account. The study of these factors should take place during the planning stage, so that there is ongoing assessment of the implementation process and the possible problems associated with it be instantly dealt with.

Teachers' exhaustion, teachers' and parents' lack of time and their reaction to change, appropriate parents', students' and teachers' information, stress caused by changes to teachers, parents and students, environment and resources necessary for implementing educational innovation, as well as the usefulness of every innovation to users should be an indispensable part of the planning stage and seriously considered by those adopting it.

3 Conclusion

This study attempted to pinpoint the barriers posed in implementing educational innovations, whether they are related to technology or the introduction of novel or enhanced educational methods and tools. Looking into the literature we focused on the key factors of the learning process. Teachers and learners have the most important role and they participate in the educational procedure which takes places in the learning environment. Many years ago we located the school as the environment of the learning procedure.

However today, the new technologies developed on intangible environment based on the internet technologies. In this study we observed that the barriers of educational

innovation development are related with the teachers and learners and also with environment and technology. As we conclude from the prior and recent literature the lack of willingness and vision for communities learning prevents the innovative prospect of educational process. Moreover the inadequate training in new technologies blocks the innovation at learning methods which are based on information technologies. Learners and their parents – for young pupils- can be also a barrier of the orientation to innovation if they are not familiar with new technologies. Regarding the environment, no technological innovative teaching methods could be expected in a non appropriate equipped educational environment.

Future research will further examine the use of information technology as a tool for the diffusion of educational innovation and will focus on the contribution of ICT to the implementation of educational innovation and the creation of dynamic learning environments.

References

- Calvert, S.L.: Media effects on children. In: *International Encyclopedia of the Social & Behavioral Sciences*, pp. 9479–9483 (2001)
- Cohen, D.K., Ball, D.L.: Educational Innovation and the problem of scale, Ann Arbor (2006)
- Dakopoulou, A.: Educational change-reform-innovation. In: Athanasoula-Reppas, A., Dakopoulou, A., Koutouzis, M., Mavrogiorgos, G., Halkiotis, D. (eds.) *Educational Units Administration*, 2nd edn, vol. A, pp. 165–211. Educational Administration and Policy, Patras, EAP (2008)
- Doyle, W., Ponder, G.A.: The practicality ethic in teacher decision-making. *Interchange* **8**(3), 1–12 (1977)
- Evers, W.J.G., Brouwers, A., Tomic, W.: Burn out and self efficacy: a study on teachers' beliefs when implementing and innovative educational system in the Netherlands. *Br. J. Educ. Psychol.* **72**, 227–243 (2002)
- Fullan, M.G.: The meaning of educational change. In: Fullan, M.G. (ed.) *The New Meaning of Educational Change*, pp. 30–46. Teachers College Press, New York (1991)
- Fraser, B.J.: Classroom climate. In: *International Encyclopedia of the Social & Behavioral Sciences*, pp. 1983–1987 (2001)
- Heich, T.: 12 Barriersto Innovation in Education, Norsafe Academy – Training by specialists (2017)
- Hurst, P.: *Implementing Innovative Projects*. The British Council/World Bank, London (1978)
- Karmel, P.: *Schools in Australia Report of the Interim Committee for the Australian Schools Commission*. Australian Govt. Publishing Service, Canberra (1973)
- Kearney, S.W., Smith, P.A., Maika, S.: Asking students their opinions of the learning environment: an empirical analysis of elementary classroom climate. *Educ. Psychol. Pract.* **32**(3), 310–320 (2016)
- King, N., Anderson N.: *Managing Innovation and Change: A Critical Guide for Organizations* (2002)
- Long, D.D.R.: *Innovation in the primary school curriculum, a comparative study*. Unpublished dissertation for M.A. (Ed.), London University (1973)
- Morris, P.: Teachers' perceptions of the barriers to the implementation of a pedagogic innovation: a Southeast Asian case study. *Int. Rev. Educ.* **XXXI**, 3–18 (1985)
- Organization for Economic Co-operation and Development: *Frascati Manual: Guidelines for Collecting and Reporting Data on Research and Experimental Development*. OECD Publishing, Paris (2015)

- Organization for Economic Co-operation and Development & Statistical Office of the European Communities: *Innovating Education and Educating for Innovation: The Power of Digital Technologies and Skills*. OECD Publishing, Paris (2016)
- Organization for Economic Co-operation and Development & Statistical Office of the European Communities: *Oslo Manual: Guidelines for Collecting and Interpreting Innovation Data*, Paris (2005)
- Rogers, E.M.: *Diffusion of Innovation*, 5th edn. The free Press, London (2003)
- Westera, W.: *Higher Education* **47**(4), 501–507 (2004)
- Scheeners, J., Creemers, B.P.M., *Conceptualizing school effectiveness* (Chapter 1), pp. 691–706 (1989)
- Shulman, L.S., Shulman, J.H.: *How and what teachers learn: a shifting perspective*. *J. Curriculum Stud.* **36**(2), 257–271 (2004)
- Spyropoulou, D., Anastasaki, A., Deligianni, D., Koutra, H., Louka, E., Bouras, S.: *Innovative educational programmes*. *Qual. Educ. Pedagogical Inst.* **2008**, 197–239 (2008)
- Troussas, C., Krouska, A., Sgouropoulou, C. : *Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education*. *Comput. Educ.* 144/103698 (2020)
- Troussas, C., Giannakas, F., Sgouropoulou, C., Voyiatzis, I.: *Collaborative activities recommendation based on students' collaborative learning styles using ANN and WSM*. *Interact. Learn. Environ.* (2020)



A Brief Survey of Deep Learning Approaches for Learning Analytics on MOOCs

Zhongtian Sun^(✉), Anoushka Harit, Jialin Yu, Alexandra I. Cristea, and Lei Shi

Department of Computer Science, Durham University, Durham, UK
{zhongtian.sun, anoushka.harit, jialin.yu, alexandra.i.cristea, lei.shi}@durham.ac.uk

Abstract. Massive Open Online Course (MOOC) systems have become prevalent in recent years and draw more attention, a.o., due to the coronavirus pandemic's impact. However, there is a well-known higher chance of dropout from MOOCs than from conventional off-line courses. Researchers have implemented extensive methods to explore the reasons behind learner attrition or lack of interest to apply timely interventions. The recent success of neural networks has revolutionised extensive Learning Analytics (LA) tasks. More recently, the associated deep learning techniques are increasingly deployed to address the dropout prediction problem. This survey gives a timely and succinct overview of *deep learning techniques for MOOCs' learning analytics*. We mainly analyse the trends of feature processing and the model design in dropout prediction, respectively. Moreover, the recent incremental improvements over existing deep learning techniques and the commonly used public data sets have been presented. Finally, the paper proposes three future research directions in the field: *knowledge graphs with learning analytics*, *comprehensive social network analysis*, *composite behavioural analysis*.

Keywords: MOOCs · Deep learning · Dropout prediction · Learning analytics

1 Introduction

Although Massive Open Online Courses (MOOCs) have been deemed as a popular choice of online education [22], the low completion rate (7–10% on average) has become a primary concern [4, 8, 26]. To address the problem, researchers are interested in exploring why students drop out, by applying different approaches.

Concomitantly, deep learning is a major sub-domain of machine learning and has consistently obtained higher accuracy, compared with conventional statistical linear regression, including for student dropout prediction [6, 29].

Several previous papers surveyed the current progress of learning analytics in MOOCs [12, 41, 48], but none of them considered the application of deep

learning in the area. In this paper, we specifically focus on analysing deep learning in MOOCs, by differentiating deep learning from classical machine learning. Extensive studies [40, 54, 56, 57, 61, 62] have investigated the dropout problem.

In this paper, a brief, timely overview of deep learning techniques that have been used to tackle the dropout problem, is presented. This study aims to help researchers understand the trends of using deep learning in MOOCs better and gain future research insights. The main contributions of this study are:

1. First presentation of a succinct and timely overview of the application of deep learning in MOOC dropout prediction.
2. Summarising the trends of feature processing and development of applied deep learning models for MOOCs, for informing research and implementation decisions of the community.
3. Identifying the recent improvements of existing deep learning techniques in the area and informing on the publicly available data for experimentation.
4. Discussing and highlighting of three possible future research directions: knowledge graph with learning analytics, comprehensive social network analysis and composite behavioural analysis.

2 Method

This paper surveys recent literature, to outline and clarify the progress of deep learning approaches to learning analytics, mainly regarding student dropout prediction. Extensive databases, including Springer Link, Association for Computing Machinery (ACM) and IEEE Xplore have been searched, by indexing keywords in titles, such as “MOOC dropout prediction”, “deep learning in learning analytics”, “application of deep learning in MOOC dropout prediction” and “MOOC dropout prediction using deep learning”, and we found 570 papers published from 2015 to 2020. We only focused on primary research articles, surveys, evaluations or reviews were excluded. Papers that mainly used conventional machine learning for dropout prediction [19, 36] deep learning for other tasks e.g., grade prediction [17], learner interactions [15] or other purposes [33] rather than dropout prediction were also ruled out. We next examined the whole text of the 41 remaining papers to understand the content and focus of deep learning in MOOC dropout prediction. Furthermore, we analysed and evaluated the trends and improvements of deep learning on the topic of dropout prediction.

3 Deep Learning in Learning Analytics of MOOCs

As explained in Sect. 1, extensive studies [40, 54, 56, 57, 61, 62] have investigated the dropout problem. However, most of them only adopt the basic Recurrent Neural Networks (RNN) to process the sequential data. However, other advanced deep learning methods, such as Convolutional Neural Networks (CNN), Graph Neural Networks (GNN) and other deep learning models present promising alternatives. They are, however, less applied to dropout prediction in MOOCs, as analysed below.

Considering the temporal activities of students, existing dropout prediction researches in MOOCs mainly use RNN networks to obtain the temporal student behavioural information contained in the whole sequence [51]. Driven by the success in other prediction areas addressed before, CNN networks have started to be applied in learning analytics. CNN is a multi-layer neural network to extract features locally and could avoid the complex feature extraction. Currently, the main predictive features of existing dropout prediction studies using deep learning could be divided into *learning activity-based prediction* and *comment analysis on discussion forums*. We also present other *recent improvements* over existing deep learning techniques deployed in MOOC prediction.

3.1 Learning Activity-Based Prediction

In the **learning activity-based prediction** field, we categorise the research into the development of the feature process and model selection. Studies [7, 14, 31] treated learning activities as features, which generally include *number of learning sessions, video viewing and clickstream, quiz attempt, forum views and posts, page views* and so on. However, those features generally require manual processing by RNN models that are not flexible and time-consuming.

To address this problem, [51] proposed a ConRec Network model consisting of CNN and RNN, to automatically extract features from raw records, as CNN networks performs well at automatic feature extraction from raw input [16]. Despite the success of feature processing, the ConRec model is not outperforming the conventional feature engineering method. [39] then used a two-dimensional CNN directly, to learn the best features from the raw click-stream data, to reduce the complexity of feature extraction; they reached 86.75% dropout forecast accuracy.

Recently, [26] reported novel progress of applying RNN directly to raw log-line level click-stream data with no feature engineering, which was inspired by the success of one popular variant of RNN models, the Long Short-Term Memory (LSTM) Networks, on anomaly detection using raw texts [63]. They combined Gated Recurrent Unit (GRU-RNN) and dropout layers to solve long-term temporal dependencies and over-fitting, which could be further extended to other large-scale data sets.

To conclude, reducing the complexity of feature selection and processing is an important trend in the research area and both CNN and RNN methods could achieve the goal, although the former is more intuitive and robust for raw click-stream data [26].

From the **model perspective**, some researches use fundamental deep learning models, such as Feedforward neural networks (FFNN), which do not consider the connections among nodes within the same layer, and thus represent the simplest type of artificial neural network [43]. For instance, [3] used FFNN in a dropout prediction study, but their results could not identify the dropout students early enough [56].

Due to the temporal nature of attrition in MOOCs, the most popular models used for the dropout prediction are RNNs, as they are based on the temporal prediction mechanism with trace data. However, RNN models generally suffer

from long-term dependency on learning behaviours in these studies, leading to limited prediction performance. Additionally, all these studies [9, 14, 44, 50] stress more temporal characteristics, but neglect the fact that students are likely to exhibit similar learning patterns during a period.

Based on the above findings, [53] proposed a novel CNN model for dropout prediction, to capture the local correlation among the learning features, as a feature matrix. These studies mainly treat all the automatic extracted features equally, but fail in considering the relative influence of different features on the outcome of dropout prediction.

Addressing the problem, [65] developed a FWTS-CNN model to consider both feature weights and learning time series, by ranking the filtered key features. This idea is akin to the widely applied attention mechanism in deep learning introduced by [47], and improved accuracy by 2% compared to using CNN alone. [38] also reported 1.75% AUC (area under the curve) improvement by integrating an attention embedding of learners' behaviours. We refer the reader elsewhere [60] for more details on the attention mechanism.

3.2 Comment Analysis

Apart from the conventional predictors of learning, the analysis of informative comments and posts could provide an understanding of learners' satisfaction and attitudes [13]. This has become a current research hotspot in the area. For example, [59] found a positive correlation between the students' confusion posts in forums and dropout. [7] found a positive relation by applying an FFNN model to examine sentiments of the forum posts data to predict the student attrition in MOOCs in the following week, with an accuracy of 88%. [55] also found a negative correlation between participation in forums and dropout rate.

However, these existing studies simply explore the positive and negative emotions [32], while learners could have more complex achievement emotions, such as confusion, boredom, shame, which could affect the learning outcome [37]. A deeper understanding of the reasons of how students think and feel regarding the course could help to timely identify students at risk and help improve their engagement [21].

3.3 Recent Improvements

Several recent works made incremental improvements over existing deep learning techniques deployed in MOOC prediction. [24] incorporated a graph convolutional network (GCN), to consider more latent features of learners, such as social interaction with others and courses, as well as learners' embeddings (representations) for prediction. GCN is proposed by [25] to capture both global structural and local patterns by treating inputs as a graph. As graphs can be irregular, arbitrary, non-Euclidean in structure and contain rich values, they are arguably capable of representing the knowledge among entities (students and courses) in the real world [46, 64]. Additionally, most existing work focuses on performance prediction based on predefined fixed order of learning activities

(videos, readings and assessment) while the conventional sequential models are unable to be implemented when there are online question pools where students could select which question to answer [30]. As nodes (entities) in graphs can be order-invariant, the student-interaction-question could be represented by a graph neural network (GNN) model; [30] proposed a novel method, R²GCN, to fully model the knowledge evolution of students and predict their performance in interactive online question pools. The idea of representing the online learning system as a graph to leverage students’ relations, activities and courses could be extended further. Additionally, most studies mainly utilise the post-hoc prediction structures, while required features are not entirely knowable for predictive models in incomplete courses [23]. Addressing the problem, [23] proposed a new algorithm based on knowledge distillation, which only requires a few basic features, but still reaches promising forecasting results.

3.4 Dataset Summary

Popular MOOC platforms are EdX, FutureLearn, Udemy, Coursera, Khan Academy, Canvas and other open online study channels. Researchers collected data from those platforms or use public datasets summarised as in Table 1, below.

Table 1. MOOC data set summary

Datasets	Category	Source	Citation
Stanford MOOCPosts	Forum discussion	[2]	[52]
Act-Mooc	Social network	[27]	[58]
KDDCup 2015	Learning analytics	[31]	[5, 51]
OULAD	Learning analytics	[28]	[20, 24]
HarvardX person	Learning analytics	[34]	[23, 45]
Coursera forums	Forum discussion	[42]	[18]

4 Future Directions

Though deep learning techniques have proven their power for dropout prediction in MOOCs, there is still room for improvement, due to the complexity of learner behaviours and sentiments. As most studies only focus on prediction based on students’ learning behaviors or other interaction activities, respectively, we suggest three future directions including more features using deep learning models, based on our review and the literature gaps it exposed.

- **Knowledge Graphs with Learning Analytics** As aforementioned, graphs can be flexible and expressive. Knowledge graphs (KGs) are graphs that consist of facts and relations among entities [49], and have been applied to

MOOCs [11] to represent the online learning resources across several platforms and to represent the relations between students and courses [24]. In fact, these techniques could be implemented to account for more demographic information of learners, such as age, gender, nationality, working experience, educational background, disability, socio-economics and so on, studied to extend dropout prediction [35]. Advanced models, like GNN, could be deployed, to treat the demographic information as features of learners (nodes) and classify the type (active/neutral/passive) of learners, or cluster them into the same group with a similar background for further analysis. Additionally, by implementing GNN models (property of invariance to node order), the analysis of the knowledge evolution process could be extended to other learning analytics tasks, e.g., adaptive online learning and early intervention, without requiring the predefined learning curriculum.

- **Comprehensive Social Network Analysis** The social interactions of students could reflect their engagement and persistence in MOOCs. However, there is a lack of serious research efforts in analysing the influence of social network structures and other features of academic assessment (e.g., time of video watching, quiz attempts) comprehensively on students' engagement [10]. In addition to student-related factors, MOOC related factors like course design and lacking of isolation are crucial dropout predictors [1,21]. Researchers could thereby consider how to integrate structural-based social network analysis with other students' learning activities and MOOC related factors like course content using deep learning models in future.
- **Composite Behavioural Analysis** In addition to learning activity-based prediction and comments analysis, multi-modal analysis is a possible future direction. As deep learning techniques push the dropout prediction, one research goal is to understand learners more comprehensively and in-depth, regarding their satisfaction, attitudes, confusion, boredom as well as other possible sentiments, to enhance their learning in MOOCs. To do so, researchers could also incorporate more behavioural observations and analysis, such as face emotion detection and physiological reactions based on CNN and other advanced models, which also allows instructors to consider the information to adapt their course content.

5 Conclusion

This paper presents a succinct survey of deep learning techniques for analysing the student dropout problem. The survey draws several conclusions. First, unlike conventional feature engineering techniques, the current trend is to use end-to-end deep learning models, including the recent RNNs and CNNs to automatically extract features from raw data. Second, despite the nature of gradual attrition in MOOC, CNNs are increasingly deployed to avoid the long-term dependency problem and complex feature processing. Third, many studies focus on providing early prediction based on rough discrete emotions, rather than understanding more in-depth sentiments and problem inducing student dropout, and thus new

methods should be developed. Fourth, we identify the recent improved work over existing deep learning techniques applied in the area. Lastly, we summarise the commonly used public datasets and suggest three future directions for the study of deep learning in MOOCs: knowledge graphs with learning analytics, comprehensive social network analysis and composite behavioural analysis.

References

1. Adamopoulos, P.: What makes a great MOOC? An interdisciplinary analysis of student retention in online courses. In: International Conference on Information Systems (2013)
2. Akshay, A., Andreas, P.: The stanford MOOCPosts data set. <https://datastage.stanford.edu/StanfordMooCPosts/>. Accessed 28 Jan 2021
3. Al-Shabandar, R., Hussain, A., Laws, A., Keight, R., Lunn, J., Radi, N.: Machine learning approaches to predict learning outcomes in massive open online courses. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 713–720. IEEE (2017)
4. Alamri, A., Sun, Z., Cristea, A.I., Senthilnathan, G., Shi, L., Stewart, C.: Is MOOC learning different for dropouts? A visually-driven, multi-granularity explanatory ML approach. In: Kumar, V., Troussas, C. (eds.) ITS 2020. LNCS, vol. 12149, pp. 353–363. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_42
5. Ardchir, S., Talhaoui, M.A., Jihal, H., Azzouazi, M.: Predicting MOOC dropout based on learner’s activity. *Int. J. Eng. Technol.* **7**(4.32), 124–126 (2018)
6. Cazarez, R.L.U., Martin, C.L.: Neural networks for predicting student performance in online education. *IEEE Lat. Am. Trans.* **16**(7), 2053–2060 (2018)
7. Chaplot, D.S., Rhim, E., Kim, J.: Predicting student attrition in MOOCs using sentiment analysis and neural networks. In: AIED Workshops, vol. 53, pp. 54–57 (2015)
8. Cristea, A.I., Alamri, A., Kayama, M., Stewart, C., Alsheri, M., Shi, L.: Earliest predictor of dropout in MOOCs: a longitudinal study of futurelearn courses. In: Information Systems Development: Designing Digitalization. Association for Information Systems (2018)
9. Crossley, S., Paquette, L., Dascalu, M., McNamara, D.S., Baker, R.S.: Combining click-stream data with NLP tools to better understand MOOC completion. In: Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, pp. 6–14 (2016)
10. Dalipi, F., Imran, A.S., Kastrati, Z.: MOOC dropout prediction using machine learning techniques: review and research challenges. In: 2018 IEEE Global Engineering Education Conference (EDUCON), pp. 1007–1014. IEEE (2018)
11. Dang, F., Tang, J., Li, S.: MOOC-KG: a MOOC knowledge graph for cross-platform online learning resources. In: 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 1–8. IEEE (2019)
12. Dietze, S., Taibi, D., d’Aquin, M.: Facilitating scientometrics in learning analytics and educational data mining-the LAK dataset. *Semant. Web* **8**(3), 395–403 (2017)
13. Dmshinskaia, N.: Dropout prediction in MOOCs: using sentiment analysis of users’ comments to predict engagement. Master’s thesis, University of Twente (2016)

14. Fei, M., Yeung, D.Y.: Temporal models for predicting student dropout in massive open online courses. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), pp. 256–263. IEEE (2015)
15. Fotso, J.E.M., Batchakui, B., Nkambou, R., Okereke, G.: Algorithms for the development of deep learning models for classification and prediction of behaviour in MOOCs. In: IEEE Learning With MOOCs (LWMOOCs), pp. 180–184. IEEE (2020)
16. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 315–323. JMLR Workshop and Conference Proceedings (2011)
17. Guo, B., Zhang, R., Xu, G., Shi, C., Yang, L.: Predicting students performance in educational data mining. In: 2015 International Symposium on Educational Technology (ISET), pp. 125–128. IEEE (2015)
18. Guo, S.X., Sun, X., Wang, S.X., Gao, Y., Feng, J.: Attention-based character-word hybrid neural networks with semantic and structural information for identifying of urgent posts in MOOC discussion forums. *IEEE Access* **7**, 120522–120532 (2019)
19. Rahmani Hanzaki, M., Demmans Epp, C.: The effect of personality and course attributes on academic performance in MOOCs. In: Pammer-Schindler, V., Pérez-Sanagustín, M., Drachsler, H., Elferink, R., Scheffel, M. (eds.) EC-TEL 2018. LNCS, vol. 11082, pp. 497–509. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98572-5_38
20. He, Y., et al.: Online at-risk student identification using RNN-GRU joint neural networks. *Information* **11**(10), 474 (2020)
21. Hone, K.S., El Said, G.R.: Exploring the factors affecting MOOC retention: a survey study. *Comput. Educ.* **98**, 157–168 (2016)
22. Jordan, K.: Massive open online course completion rates revisited: assessment, length and attrition. *Int. Rev. Res. Open Distrib. Learn.* **16**(3), 341–358 (2015)
23. Kang, T., Wei, Z., Huang, J., Yao, Z.: MOOC student success prediction using knowledge distillation. In: 2020 International Conference on Computer Information and Big Data Applications (CIBDA), pp. 363–367. IEEE (2020)
24. Karimi, H., Derr, T., Huang, J., Tang, J.: Online academic course performance prediction using relational graph convolutional neural network. In: Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020), pp. 444–450 (2020)
25. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
26. Kőrösi, G., Farkas, R.: MOOC performance prediction by deep learning from raw clickstream data. In: Singh, M., Gupta, P.K., Tyagi, V., Flusser, J., Ören, T., Valentino, G. (eds.) ICACDS 2020. CCIS, vol. 1244, pp. 474–485. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-6634-9_43
27. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1269–1278 (2019)
28. Kuzilek, J., Hlosta, M., Zdrahal, Z.: Open university learning analytics dataset. *Sci. Data* **4**(1), 1–8 (2017)
29. Lang, C., Siemens, G., Wise, A., Gasevic, D.: Handbook of Learning Analytics. SOLAR, Society for Learning Analytics and Research, New York (2017)
30. Li, H., Wei, H., Wang, Y., Song, Y., Qu, H.: Peer-inspired student performance prediction in interactive online question pools with graph neural network. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2589–2596 (2020)

31. Li, W., Gao, M., Li, H., Xiong, Q., Wen, J., Wu, Z.: Dropout prediction in moocs using behavior features and multi-view semi-supervised learning. In: 2016 international joint conference on neural networks (IJCNN). pp. 3130–3137. IEEE (2016)
32. Liu, B., Xing, W., Zeng, Y., Wu, Y.: Quantifying the influence of achievement emotions for student learning in MOOCs. *J. Educ. Comput. Res.* **59**(3), 429–452 (2021)
33. Liu, L., et al.: Prerequisite relation learning for course concepts based on hyperbolic deep representation. *IEEE Access* **8**, 49079–49089 (2020)
34. HarvardX-MITx: HarvardX-MITx person-course academic year 2013 de-identified dataset, version 2.0. Harvard Dataverse (2014)
35. Morris, N.P., Swinnerton, B., Hotchkiss, S.: Can demographic information predict MOOC learner outcomes? In: Experience Track: Proceedings of the European MOOC Stakeholder, Leeds (2015)
36. Mulyani, E., Hidayah, I., Fauziati, S.: Dropout prediction optimization through smote and ensemble learning. In: 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 516–521. IEEE (2019)
37. Pekrun, R., Lichtenfeld, S., Marsh, H.W., Murayama, K., Goetz, T.: Achievement emotions and academic performance: longitudinal models of reciprocal effects. *Child Dev.* **88**(5), 1653–1670 (2017)
38. Pulikottil, S.C., Gupta, M.: ONet-a temporal meta embedding network for MOOC dropout prediction. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 5209–5217. IEEE (2020)
39. Qiu, L., Liu, Y., Hu, Q., Liu, Y.: Student dropout prediction in massive open online courses by convolutional neural networks. *Soft. Comput.* **23**(20), 10287–10301 (2019)
40. Raga, R.C., Raga, J.D.: Early prediction of student performance in blended learning courses using deep neural networks. In: 2019 International Symposium on Educational Technology (ISET), pp. 39–43. IEEE (2019)
41. Romero, C., Ventura, S.: Educational data mining and learning analytics: an updated survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **10**(3), e1355 (2020)
42. Rossi, L.A., Gnawali, O.: Language independent analysis and classification of discussion threads in Coursera MOOC forums. In: Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), pp. 654–661. IEEE (2014)
43. Schmidhuber, J.: Deep learning in neural networks: an overview. *Neural Netw.* **61**, 85–117 (2015)
44. Sun, D., Mao, Y., Du, J., Xu, P., Zheng, Q., Sun, H.: Deep learning for dropout prediction in MOOCs. In: 2019 Eighth International Conference on Educational Innovation through Technology (EITT), pp. 87–90. IEEE (2019)
45. Tang, J.K.T., Xie, H., Wong, T.-L.: A big data framework for early identification of dropout students in MOOC. In: Lam, J., Ng, K.K., Cheung, S.K.S., Wong, T.L., Li, K.C., Wang, F.L. (eds.) *ICTE 2015*. CCIS, vol. 559, pp. 127–132. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48978-9_12
46. Tang, S., Peterson, J., Pardos, Z.: Predictive modeling of student behavior using granular large scale action data from a MOOC. In: *Handbook of Learning Analytics and Educational Data Mining* (2017)
47. Vaswani, A., et al.: Attention is all you need. arXiv preprint [arXiv:1706.03762](https://arxiv.org/abs/1706.03762) (2017)
48. Viberg, O., Hatakka, M., Bälter, O., Mavroudi, A.: The current landscape of learning analytics in higher education. *Comput. Hum. Behav.* **89**, 98–110 (2018)

49. Wang, H., et al.: Exploring high-order user preference on the knowledge graph for recommender systems. *ACM Trans. Inf. Syst. (TOIS)* **37**(3), 1–26 (2019)
50. Wang, L., Wang, H.: Learning behavior analysis and dropout rate prediction based on MOOCs data. In: 2019 10th International Conference on Information Technology in Medicine and Education (ITME), pp. 419–423. IEEE (2019)
51. Wang, W., Yu, H., Miao, C.: Deep model for dropout prediction in MOOCs. In: Proceedings of the 2nd International Conference on Crowd Science and Engineering, pp. 26–32 (2017)
52. Wei, X., Lin, H., Yang, L., Yu, Y.: A convolution-LSTM-based deep neural network for cross-domain MOOC forum post classification. *Information* **8**(3), 92 (2017)
53. Wen, Y., Tian, Y., Wen, B., Zhou, Q., Cai, G., Liu, S.: Consideration of the local correlation of learning behaviors to predict dropouts from MOOCs. *Tsinghua Sci. Technol.* **25**(3), 336–347 (2019)
54. Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., Tingley, D.: Delving deeper into MOOC student dropout prediction. arXiv preprint [arXiv:1702.06404](https://arxiv.org/abs/1702.06404) (2017)
55. Xing, W., Chen, X., Stein, J., Marcinkowski, M.: Temporal predication of dropouts in MOOCs: reaching the low hanging fruit through stacking generalization. *Comput. Hum. Behav.* **58**, 119–129 (2016)
56. Xing, W., Du, D.: Dropout prediction in MOOCs: using deep learning for personalized intervention. *J. Educ. Comput. Res.* **57**(3), 547–570 (2019)
57. Xiong, F., Zou, K., Liu, Z., Wang, H.: Predicting learning status in MOOCs using LSTM. In: Proceedings of the ACM Turing Celebration Conference-China, pp. 1–5 (2019)
58. Xu, Z., Ou, Z., Su, Q., Yu, J., Quan, X., Lin, Z.: Embedding dynamic attributed networks by modeling the evolution processes. arXiv preprint [arXiv:2010.14047](https://arxiv.org/abs/2010.14047) (2020)
59. Yang, D., Wen, M., Howley, I., Kraut, R., Rose, C.: Exploring the effect of confusion in discussion forums of massive open online courses. In: Proceedings of the Second (2015) ACM Conference on Learning@ Scale, pp. 121–130 (2015)
60. Yin, S., Lei, L., Wang, H., Chen, W.: Power of attention in MOOC dropout prediction. *IEEE Access* **8**, 202993–203002 (2020)
61. Yu, C.H., Wu, J., Liu, A.C.: Predicting learning outcomes with MOOC click-streams. *Educ. Sci.* **9**(2), 104 (2019)
62. Zaporozhko, V.V., Parfenov, D.I., Shardakov, V.M.: Development approach of formation of individual educational trajectories based on neural network prediction of student learning outcomes. In: Hu, Z., Petoukhov, S., He, M. (eds.) AIMEE 2019. AISC, vol. 1126, pp. 305–314. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39162-1_28
63. Zhang, K., Xu, J., Min, M.R., Jiang, G., Pelechris, K., Zhang, H.: Automated it system failure prediction: a deep learning approach. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 1291–1300. IEEE (2016)
64. Zhang, Z., Cui, P., Zhu, W.: Deep learning on graphs: a survey. *IEEE Trans. Knowl. Data Eng.* (2020)
65. Zheng, Y., Gao, Z., Wang, Y., Fu, Q.: MOOC dropout prediction using FWTS-CNN model based on fused feature weighting and time series. *IEEE Access* **8**, 225324–225335 (2020)

Models



DiKT: Dichotomous Knowledge Tracing

Seounghun Kim, Woojin Kim, Heeseok Jung, and Hyeoncheol Kim^(✉)

Department of Computer Science and Engineering,
Korea University, Seoul, South Korea
{ryankim0409,woojinkim1021,poco2889,harrykim}@korea.ac.kr

Abstract. Knowledge tracing models the cognitive process of skill acquisition of a student to predict the current knowledge state. Based on cognitive processing theory, we regard student knowledge state in dichotomous view in alignment with Performance Factor Analysis (PFA). Assuming that a student's correct and incorrect responses are fundamentally different for modeling a student's knowledge state, we propose a Dichotomous Knowledge Tracing (DiKT), a novel knowledge tracing network with a dichotomous perspective on a student's knowledge state. We modify the network's value memory by dividing it into two memories, each encoding recallable and unrecallable knowledge to precisely capture the student knowledge state. With the proposed architecture, our model generates a knowledge trajectory that instantly and accurately portrays a student's knowledge level based on learning history. Empirical evaluations demonstrate that our proposed model achieves comparable performance on benchmark educational datasets.

Keywords: Knowledge tracing · Performance Factor Analysis · Dynamic Key-Value Memory Network · Deep learning · Student modeling · Learning analytics

1 Introduction

Computer-aided instruction (CAI) aims to provide individualized teaching with the awareness of who is being taught [16]. Among approaches to represent how much a student knows at any time, Knowledge Tracing (KT) aims to model the cognitive process of skill acquisition to predict the knowledge state of a student [1]. The migration of learning to an adaptive, large-scale online environment aspired the applications of knowledge tracing for modeling student knowledge acquisition. In the online learning setting, the gathered student data in the system records student responses to questions, mainly focusing on the student's factual and experiential evaluation.

Application of deep learning models on knowledge tracing for adaptive, large-scale online learning environment showed promising results in modeling student knowledge acquisition. Especially, Deep Knowledge Tracing (DKT) and Dynamic Key-Value Memory Network (DKVMN) demonstrated significant performance on the task of predicting future student performance [14, 22]. These approaches

are built upon the theory of skill knowledge and share the definition of student knowledge along with KT, where it aims to model the cognitive process of skill acquisition to predict the knowledge state of a student.

The student data in the online learning environment is the history of accumulated declarative knowledge that portrays the student’s mastery level on a concept [8]. Therefore, the student data conveys the attempts to recall knowledge to the surface of conscious awareness. The ability to recall knowledge is dependent on the effectiveness of the student’s encoding operations on the subject [7]. The difficulty of recall differentiates among degrees of encoding of knowledge to memory. The degree ranges from well encoded to barely adequate for recall, introducing recalled-unrecalled dichotomy on knowledge [17].

Performance Factor Analysis (PFA) regards student knowledge state in dichotomous view [12]. PFA is a variation of the probabilistic cognitive diagnosis model emphasizing flexibility in an adaptive online environment. PFA considers performance the strongest indicator of student ability and divides the student’s ability into a success variable and a failure variable. PFA finds their assumption on the categorization of practice to success and failure in the psychological literature that success may lead to student automaticity and less forgetting while failure uncovers the need for accumulation of declarative knowledge [13].

Adopting from student’s cognitive process and PFA on student knowledge, we assume that a student’s correct and incorrect responses are fundamentally different in characteristic for modeling a student’s knowledge state, affecting the predicted student mastery level on a concept. Thus, we regard the correct answer as recalled knowledge and the wrong answer as unrecalled knowledge. Based on our assumption on student knowledge, we propose a novel Dichotomous Knowledge Tracing network (DiKT). We modify the DKVMN’s value memory by dividing it into two memories, each encoding recallable and unrecallable knowledge, to better model the student knowledge state.

Our contribution is threefold: First, we revisit the definition of student knowledge defined in knowledge tracing and explores dichotomous views on student knowledge. Second, we propose a novel deep knowledge tracing architecture that considers two knowledge states, each encoding recallable and unrecallable concepts. Third, we demonstrate that our proposed model generates more comprehensible learning trajectories while achieving comparable performance with existing knowledge tracing models.

2 Related Work

2.1 Information Processing Theory

Information processing theory describes a student’s cognitive process to transform knowledge into long-term memory [3]. The final stage of the cognitive process is retrieval, which is referred to as the surfacing of the encoded knowledge in long-term memory to the level of consciousness. Easy retrieval of the knowledge requires well-encoded knowledge in memory and suitable retrieval cues.

As cognitive psychology literature suggests, the student’s memory encoding is characterized as levels of processing [2]. The basic notion is that memory duration is determined relative to the level of depth at which the information is processed [11]. A deep level of processing requires analysis in terms of semantics and is assumed to induce longer and stronger memory trace. Shallow processing, on the other hand, focuses on the non-semantic analysis of orthographic or phonological attributes [10]. The concept of levels of processing retains the assumption of the difference between deep and shallow processing, suggesting the dichotomous view of memory in learning.

2.2 Performance Factor Analysis

Performance Factor Analysis (PFA) is an educational data mining model that predicts accumulated learning of a student [12]. PFA is proposed to overcome the limitations of Learning Factor Analysis (LFA), which is insensitive to the student’s practice frequency. PFA modifies the LFA’s standard form by replacing the student ability term with the student’s prior success and failure on a particular knowledge concept. The PFA model is formulated as:

$$p(m) = \sigma \left(\sum_{j \in KCs} (\beta_j + \gamma_j s_j + \rho_j f_j) \right) \quad (1)$$

where $p(m)$ is the probability of accumulated learning of a student, β_j is the easiness of a knowledge concept j , and σ is a sigmoid function. The s_j is prior success attempts, and f_j is prior failure attempts, and together, they indicate the student ability. The γ and ρ are the weights of observed counts. PFA divides the student’s succeeded and failed attempts to make the model sensitive to the student’s failed attempts. Including the failure attempts to the model allows the incorrectness as a measure of learning, contrasting the successful attempts.

2.3 Deep Learning Based Knowledge Tracing

Knowledge Tracing (KT) aims to personalize the learning process for efficient skill mastery and predict future student performance by observing student progress on knowledge concepts [1]. KT is based on the theory that the process of skill acquisition is constructed into three stages: acquiring declarative knowledge, proceduralization of knowledge, and acquiring procedural knowledge [5]. KT further assumes that procedural knowledge is presumably represented as a probability estimate that associates a student’s current knowledge state and student actions of problem-solving. Based on this theory, KT is transformed into a task of predicting the mastery level on a knowledge concept given the historical academic achievement trajectories. The task attempts to model a student’s changing knowledge state on a concept during the process of learning.

A recent and state-of-the-art approach to predict the mastery level of a student utilizes deep learning models. Deep Knowledge Tracing (DKT) was the first to apply Long-Term Short Memory (LSTM) to model the changing knowledge

state of the student [14]. Current deep learning approaches have advanced to incorporating exercise text [4] or utilizing concept hierarchy as graphs [18].

Dynamic Key-Value Memory Network (DKVMN) was proposed due to limitations of DKT on the latent representation of the student knowledge [22]. The DKT compresses a student’s knowledge of all concepts into a single hidden state of fixed dimensions. As the sequence of student exercises gets longer, it becomes difficult to trace the student’s knowledge of a certain concept.

To overcome the limitation of DKT, DKVMN utilizes the Key-Value Memory Network structure. DKVMN’s architecture is based on a Memory-Augmented Neural Network, where it has an external memory module outside of the network to store the latent representation and interacts with the controller using read and write heads [9]. Adopting this architecture, DKVMN keeps two memories: a static key memory for storing knowledge concepts and a dynamic value memory for recording students’ knowledge state on each concept. DKVMN outperformed DKT with robustness to overfitting and fewer model parameters, introducing advantages of memory network architecture over standard LSTM.

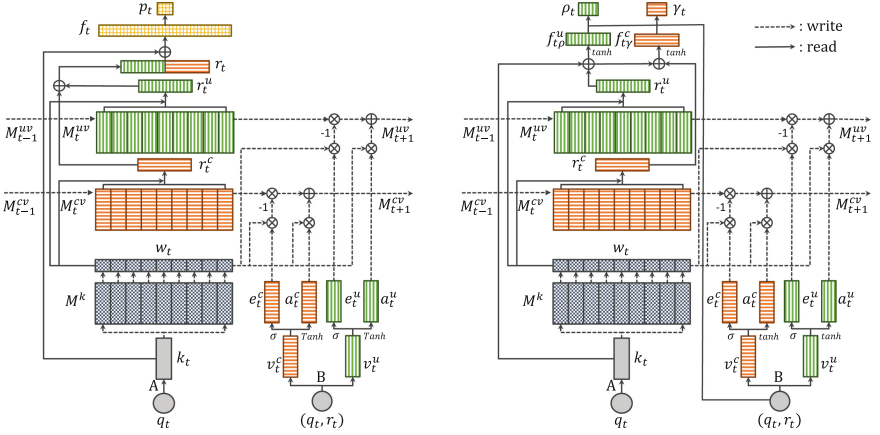
To the best of our knowledge, Deep-IRT is the first approach to merge the deep knowledge tracing models and the probabilistic knowledge tracing model [21]. Deep-IRT outputs the student ability and concept easiness variable using DKVMN architecture, and the final probability of correctness is calculated by passing those two values to the Item Response Theory function [15]. Taking on the attempt to consolidate IRT and DKVMN, we propose to integrate the essence of PFA into DKVMN.

3 DiKT: Dichotomous Knowledge Tracing Network

We propose to reflect these different knowledge states to the deep neural network architecture with our assumption on the fundamental differences between the recallable and unrecallable knowledge of a student. We divide the value memory into two value memory matrices, each encoding the recallable or correct exercise and unrecallable or incorrect exercise of a student. The overall architecture of our model is illustrated in Fig. 1a.

The problem formulation of KT is: given a student’s exercise history $\mathbf{X} = \{x_1, x_2, \dots, x_t\}$ where each x is a tuple of a question and the response to the question $x_t = (q_t, r_t)$ and student response is a binary variable $r \in 0, 1$, predict the probability that the student will correctly respond to a question in the next time step $p(r_{t+1} = 1 | \mathbf{X}, q_{t+1})$.

The first step of the model is generating a correlation weight vector, which indicates the degree of association between the question and each latent concept. First, the question embedding vector k_t is generated by multiplying the question q_t with an embedding matrix A . The correlation weight is computed by applying the softmax activation to the inner product of the question embedding vector and the static key matrix: $w_t = \text{Softmax}(k_t^T M^k(i))$.



(a) The architecture of DiKT, primary model of this paper. (b) The modified DiKT model outputting parameters for PFA.

Fig. 1. The overall architecture of proposed models. The key matrix and correlation vector are in dark gray. The read and write process of recallable and unrecallable value memory is in orange and green, respectively. The summary vector is in yellow. The modified DiKT model of Fig. 1b outputs a separate summary vector for each value memory. Best viewed in color. (Color figure online)

3.1 Recalled and Unrecalled Value Memory

In contrast to the DKVMN, we produce the read content for each of the value memory. The recallable read content r_t^c is produced as the weighted sum of the correlation weight to the recallable value memory where N is the number of latent knowledge concepts.

$$\mathbf{r}_t^c = \sum_{i=1}^N w_t(i) \mathbf{M}_t^{cv}(i) \quad (2)$$

The unrecallable read content is produced simultaneously.

$$\mathbf{r}_t^u = \sum_{i=1}^N w_t(i) \mathbf{M}_t^{uv}(i) \quad (3)$$

Then, the recallable and unrecallable read content and the question embedding vector \mathbf{k}_t are concatenated. The final concatenated vector goes through a fully connected layer \mathbf{W}_s and the hyperbolic tangent activation function to produce the summary vector \mathbf{f}_t .

$$\mathbf{f}_t = \tanh(\mathbf{W}_s^T [\mathbf{r}_t^c, \mathbf{r}_t^u, \mathbf{k}_t] + \mathbf{b}_s) \quad (4)$$

For the final step, the summary vector \mathbf{f}_t goes through the fully connected output layer \mathbf{W}_o to produce the final probability of response correctness on a question of the student.

$$p_t = \sigma(\mathbf{W}_o^T \mathbf{f}_t + \mathbf{b}_o) \quad (5)$$

The write process to the recallable and unrecallable value memory also occurs simultaneously. First the tuple (q_t, r_t) is divided into correct responses and incorrect responses of the student, (q_t^c, r_t^c) and (q_t^u, r_t^u) . Then, each tuple is embedded with the same embedding layer \mathbf{B} and produces recallable embedding vector \mathbf{v}_t^c and unrecallable embedding vector \mathbf{v}_t^u .

Each embedding vector goes through a fully connected erase layer and fully connected add layer, producing an erase vector and an add vector for each of recallable and unrecallable tuples.

$$\mathbf{e}_t^c = \sigma(\mathbf{E}^{cT} \mathbf{v}_t^c + \mathbf{b}_e^c) \quad (6)$$

$$\mathbf{e}_t^u = \sigma(\mathbf{E}^{uT} \mathbf{v}_t^u + \mathbf{b}_e^u) \quad (7)$$

$$\mathbf{a}_t^c = \tanh(\mathbf{D}^{cT} \mathbf{v}_t^c + \mathbf{b}_a^c)^T \quad (8)$$

$$\mathbf{a}_t^u = \tanh(\mathbf{D}^{uT} \mathbf{v}_t^u + \mathbf{b}_a^u)^T \quad (9)$$

The resulting recallable and unrecallable erase and add vectors are multiplied with the correlation weights and added to the previous time step's corresponding value memory. The erase-and-add mechanism of the write process mimics the behavior of the input gate and forget gate, allowing the value memory to stay up-to-date on the latest information.

$$\tilde{\mathbf{M}}_t^{cv}(i) = \mathbf{M}_{t-1}^{cv}(i)[1 - w_t(i)\mathbf{e}_t^c] \quad (10)$$

$$\tilde{\mathbf{M}}_t^{uv}(i) = \mathbf{M}_{t-1}^{uv}(i)[1 - w_t(i)\mathbf{e}_t^u] \quad (11)$$

$$\mathbf{M}_t^{cv}(i) = \tilde{\mathbf{M}}_{t-1}^{cv}(i) + w_t(i)\mathbf{a}_t^c \quad (12)$$

$$\mathbf{M}_t^{uv}(i) = \tilde{\mathbf{M}}_{t-1}^{uv}(i) + w_t(i)\mathbf{a}_t^u \quad (13)$$

3.2 DiKT Augmented with PFA

To further explore the possibility of integrating probabilistic and deep KT models, we adapt our model to the architecture of Deep-IRT [21]. The Deep-IRT augments DKVMN by attaching student ability and difficulty network that outputs the parameters for two-parameter logistic IRT model of the function $p_t = \sigma(3.0 * \theta_{tj} - \beta_j)$ [20].

The structure of the network is illustrated in the Fig. 1b. To incorporate this architecture, we do not concatenate the recallable read content and unrecallable read content. Instead, we extend our model to produce two separate summary vector for each value memory, $\mathbf{f}_{t\gamma}$ and $\mathbf{f}_{t\rho}$. The resulting recallable and unrecallable summary vector separately goes through an additional fully connected layer, producing success variable γ_{tj} and failure variable ρ_{tj} of PFA.

$$\gamma_{tj} = \tanh(\mathbf{W}_\gamma \mathbf{f}_{t\gamma} + \mathbf{b}_\gamma) \quad (14)$$

$$\rho_{tj} = \tanh(\mathbf{W}_\rho \mathbf{f}_{t\rho} + \mathbf{b}_\rho) \quad (15)$$

The difficulty level β_j is calculated in accordance with Deep-IRT. The final probability is calculated as $p_t = \sigma(\gamma_{tj}\sigma(s_{tj}) + \rho_{tj}\sigma(f_{tj}) - 2.0 * \beta_j)$. The difficulty level is doubled to scale the final output of the model before the sigmoid activation into range $[-4, 4]$ so that the maximum model output is $\sigma(4) = 0.982$.

4 Experiments

We evaluate our proposed models’ performance with three baseline models, DKT, DKVMN, and Deep-IRT, on four benchmark datasets, ASSISTments2009, ASSISTments2015, Statics2011, and Synthetic. We use accuracy and the Area Under the ROC Curve (AUC) for evaluation while considering AUC as the primary metric for model performance.

ASSISTments datasets¹ are collected from ASSISTment’s online tutoring system. We use skill-builder data gathered in school years 2009-10 and 2015. Statics2011² is from Carnegie Melon’s Open Learning Initiative (OLI) platform and contains 335 student data from a statics course of Fall 2011. Synthetic dataset³ consists of responses from 2,000 simulated students generated with Item Response Theory. The data consists of student responses on a sequence of 50 questions, where each question belongs to one of the five hidden concepts.

All networks are implemented with the Pytorch library with the distributed data-parallel method. For preprocessing, we limit the maximum sequence length of each instance to 200-time steps. We train all models for 100 epochs using the binary cross-entropy loss. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e - 8$ [6]. We use the Noam learning rate scheduler with 4,000 warm-up steps [19]. The dimensions of the key, value memory, summary vector, number of knowledge concepts are adjusted according to each dataset. We early stop the training when the model’s validation performance does not improve for 40 epochs. We provide the final model performance using weights with the highest validation AUC.

Table 1. The experiment results. Best results are highlighted in bold. Our proposed model achieves the best performance on most of the datasets.

	DKT		DKVMN		Deep-IRT		DiKT		DiKT+PFA	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
ASSISTments2009	81.55	76.74	76.37	81.07	75.76	79.76	76.56	81.33	76.34	80.30
ASSISTments2015	74.79	72.82	74.91	72.49	74.88	72.04	75.16	72.60	74.82	70.78
Statics2011	80.98	82.68	81.50	81.00	80.41	80.77	80.85	83.20	80.32	80.57
Synthetic	73.94	80.91	74.04	81.21	74.16	81.26	74.41	81.72	73.66	80.62

¹ <https://sites.google.com/site/assistmentsdata>.

² <https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507>.

³ <https://github.com/chrispiech/DeepKnowledgeTracing>.

4.1 Model Performance

Table 1 presents the prediction performance of our proposed models in accuracy and AUC. DiKT achieves the best performance compared to baseline models on four datasets. This suggests that separating students’ recalled and unrecalled knowledge is more effective in modeling student knowledge. Moreover, our DiKT model achieves better performance than PFA-augmented DiKT, implying the limitation of constructing the model’s final output based on a linear equation.

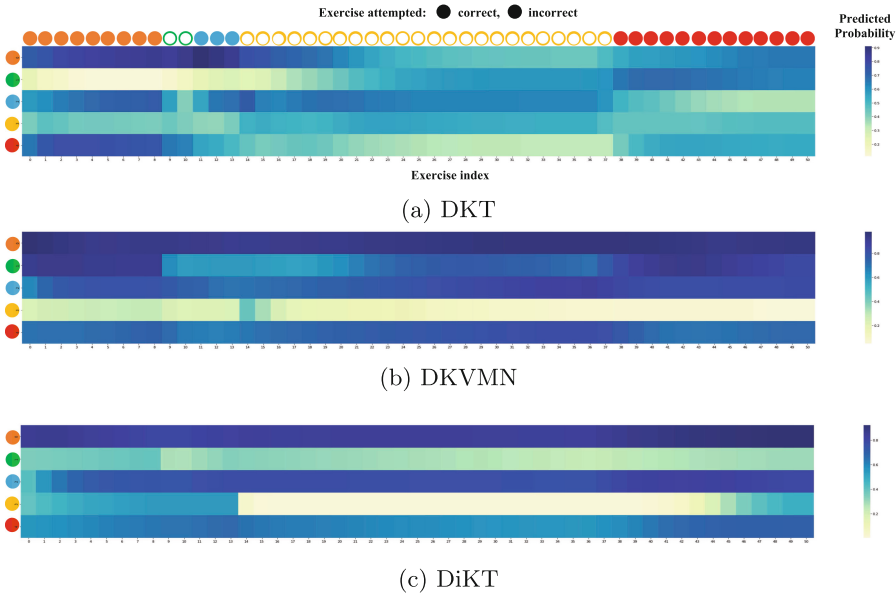


Fig. 2. Learning trajectories of DKT, DKVMN, and DiKT. Y-axis is knowledge concepts (color-coded) and x-axis is a student’s learning history. The learning trajectory of our proposed model, DiKT (2c) accurately captures the student’s knowledge state by distinctively representing knowledge level for each concept based on the learning history. Compared to DiKT, baseline models’ learning trajectories are incomprehensible or imprecisely tracks the student’s knowledge state for each concept. Best viewed in color. (Color figure online)

4.2 Model Comparison

We further evaluate and compare the model’s inference ability to capture the changing knowledge state of students. We draw an arbitrary student’s learning trajectories on a sequence of 50 questions using the ASSISTments2009 data. The learning trajectory visualizes a model’s prediction of giving the correct answer to each of the questions. Each row indicates a concept, and each cell’s color

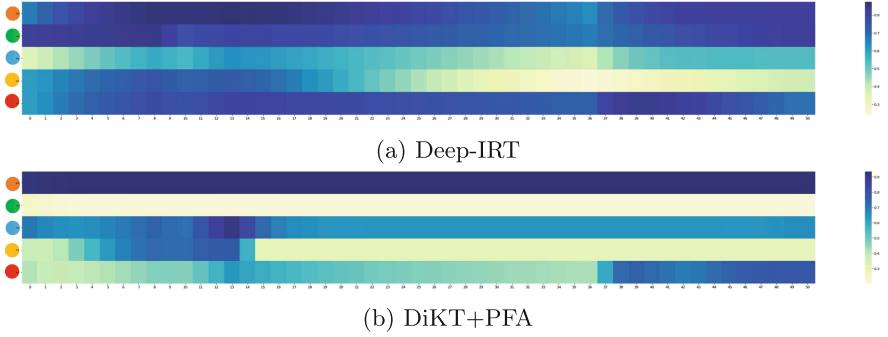


Fig. 3. Learning trajectories of Deep-IRT and DiKT+PFA. The learning trajectories are drawn with the same learning history of Fig. 2. The learning trajectory of DiKT+PFA (3b) better captures the knowledge state than Deep-IRT, immediately portraying student responses in representing knowledge state.

denotes a student’s estimated knowledge state of the concept. Darker the color, the higher the student’s estimated knowledge. The resulting learning trajectories are presented in Fig. 2 and Fig. 3. All trajectories are drawn using the same learning history.

The learning sequence of the student is presented in the top of Fig. 2a. A full circle and an empty circle denote student’s correct and incorrect answer, respectively. Among the five concepts, the student gave incorrect answers to the second and the fourth concept. We first compare the learning trajectories of DKT, DKVMN, and DiKT, presented in Fig. 2a, 2b, and 2c, respectively. The learning trajectory of the DiKT accurately estimates the student’s knowledge state compared to DKT and DKVMN. In DiKT’s trajectory, the estimated knowledge level decreases for the second and the fourth concept which correctly portrays the student’s learning history. However, the trajectory of DKVMN does not seem to consider the second concept. We believe that DiKT is more sensitive to right and wrong answers of the students due to separating the value memory, while DKT and DKVMN encode the information into a single cell or memory.

We also present learning trajectories of models that construct outputs with linear models in Fig. 3a and 3b. Compared to Deep-IRT, the DiKT+PFA correctly estimates low knowledge level for the second and the fourth concept. The DiKT+PFA is also more responsive to student answers than Deep-IRT, where the change of student knowledge state prediction occurs gradually.

Although our proposed model achieves similar performance with baseline models, our model provides more comprehensive learning trajectories. To deploy the model for supporting teaching and learning, the accurate model inference is more instrumental than model performance, that model performance only portrays the model’s ability on a sample population of students. Our models can present reasonable and comprehensible learning trajectories, broadening the application scope of deep learning models to aid individualized learning.

5 Conclusion

We propose a novel knowledge tracing deep neural network named Dichotomous Knowledge Tracing. DiKT constructs two separate value memories to integrate the level of cognitive processing of knowledge recall into the deep neural network architecture. With the independent modeling of student knowledge, our proposed model improves the network’s predictive power in precisely capturing the student’s changing knowledge state. Empirical evaluation of four datasets demonstrates that our model generates comprehensible learning trajectories with high performance. For future work, we plan to implement our model into a large-scale online learning platform, providing effective learning experience to students.

Acknowledgements. This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques).

References

1. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adap. Interact.* **4**(4), 253–278 (1994)
2. Craik, F.I., Lockhart, R.S.: Levels of processing: a framework for memory research. *J. Verbal Learn. Verbal Behav.* **11**(6), 671–684 (1972)
3. Flavell, J.: *Cognitive Development*. Prentice Hall, Upper Saddle River (2002)
4. Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., Hu, G., et al.: EKT: exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.* **33**, 100–115 (2019)
5. Jensen, J.C.: Skill acquisition and second language teaching. *Kinki Univ. Dept. Lang. Educ. J.* (3), 119–135 (2007)
6. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
7. Lovelace, E.A.: Metamemory: monitoring future recallability during study. *J. Exp. Psychol. Learn. Mem. Cogn.* **10**(4), 756 (1984)
8. Means, B., Toyama, Y., Murphy, R., Bakia, M., Jones, K.: *Evaluation of evidence-based practices in online learning: a meta-analysis and review of online learning studies*. US Department of Education (2009)
9. Miller, A., Fisch, A., Dodge, J., Karimi, A.H., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1400–1409 (2016)
10. Moeser, S.D.: Levels of processing: qualitative differences or task-demand differences? *Mem. Cogn.* **11**(3), 316–323 (1983). <https://doi.org/10.3758/BF03196978>
11. Morris, C.D., Bransford, J.D., Franks, J.J.: Levels of processing versus transfer appropriate processing. *J. Verbal Learn. Verbal Behav.* **16**(5), 519–533 (1977)
12. Pavlik, P.I., Jr., Cen, H., Koedinger, K.R.: Performance factors analysis- a new alternative to knowledge tracing. *Artif. Intell. Educ. Front. Artif. Intell. Appl.* **200**, 531–538 (2009)

13. Pavlik, P.I., Jr. Eglington, L.G., Harrell-Williams, L.M.: Generalized knowledge tracing: a constrained framework for learner modeling. arXiv preprint [arXiv:2005.00869](https://arxiv.org/abs/2005.00869) (2020)
14. Piech, C., et al.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems, pp. 505–513 (2015)
15. Rasch, G.: Probabilistic Models for Some Intelligence and Attainment Tests. ERIC (1993)
16. Self, J.A.: Student models in computer-aided instruction. *Int. J. Man-Mach. Stud.* **6**(2), 261–276 (1974)
17. Thompson, C.P., Wenger, S.K., Bartling, C.A.: How recall facilitates subsequent recall: a reappraisal. *J. Exp. Psychol. Hum. Learn. Mem.* **4**(3), 210 (1978)
18. Tong, H., Zhou, Y., Wang, Z.: HGKT: introducing problem schema with hierarchical exercise graph for knowledge tracing. arXiv preprint [arXiv:2006.16915](https://arxiv.org/abs/2006.16915) (2020)
19. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
20. Yang, F.M., et al.: Item response theory for measurement validity. *Shanghai Arch. Psychiatry* **26**(3), 171 (2014)
21. Yeung, C.K.: Deep-IRT: make deep learning based knowledge tracing explainable using item response theory. In: Proceedings of the 12th International Conference on Educational Data Mining (2019)
22. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th international conference on World Wide Web, pp. 765–774 (2017)



CompPrehension - Model-Based Intelligent Tutoring System on Comprehension Level

Oleg Sychev^(✉), Anton Anikin, Nikita Pensky, Mikhail Denisov,
and Artem Prokudin

Volgograd State Technical University, Lenin Ave, 28, Volgograd 400005, Russia
o_sychev@vstu.ru

Abstract. Intelligent tutoring systems become increasingly common in assisting human learners, but they are often aimed at isolated domain tasks without creating a scaffolding system from lower- to higher-level cognitive skills. We designed and implemented an intelligent tutoring system CompPrehension aimed at the comprehension level of Bloom's taxonomy that often gets neglected in favour of the higher levels. The system features plugin-based architecture, easing adding new domains and learning strategies; using formal models and software reasoners to solve the problems and judge the answers; and generating explanatory feedback and follow-up questions to stimulate the learners' thinking. The architecture and workflow are shown. We demonstrate the process of interacting with the system in the Control Flow Statements domain. The advantages and limits of the developed system are discussed.

Keywords: Bloom's taxonomy · Adaptive learning · Cognitive learning · Constraint-based tutors · Intelligent tutoring system

1 Introduction

Lately, a large number of Intelligent Tutoring Systems (ITS) is developed for different subject domains. Less effort is spent on categorising the learning tasks they use to provide adequate scaffolding for developing higher-level skills.

The popular Bloom's taxonomy [3] of educational objectives identifies six levels of cognitive skills that, ideally, should be developed in that order because higher-level skills rely on lower-level ones. Learning tasks on the first three levels are mostly simple: any sequence of correct steps leads to a solution. The higher-level tasks often require a strategy to reach the objective; not all correct moves will bring the learner to the solution. So on the knowledge, comprehension, and application levels the difference between cognitive (follow the learner step-by-step during problem solution) and constraint-based (checking snapshot solution states for not breaking the domain constraints) [10] ITSs is negligible.

The reported study was funded by RFBR, project number 20-07-00764.

© Springer Nature Switzerland AG 2021

A. I. Cristea and C. Troussas (Eds.): ITS 2021, LNCS 12677, pp. 52–59, 2021.

https://doi.org/10.1007/978-3-030-80421-3_6

While higher-level objectives are often the true goals of education, concentrating on high-level assessments may result in lessening learning gains as some students try to solve high-level problems without good knowledge and comprehension of the subject-domain concepts that limits their progress severely. E.g., students have difficulties writing code even given an algorithm as they do not make the connection between pseudo-code and programming-language code [16].

Knowledge can be taught using simple quizzes, but developing comprehension and learning to apply rules requires intelligent support. This makes developing ITSs aimed at comprehension and application levels an important challenge. The relative simplicity of comprehension-level assessments allows using formal domain models and software reasoners [1, 15]. This opens the way to developing a multi-domain ITS where subject domains can be attached as plugins. Comprehension-level tutoring systems are most useful when learners are introduced to new subject domains including large numbers of new concepts. We chose introductory programming as one such domain for in-depth study.

2 Related Work

Modern Intelligent Programming Tutoring Systems (IPTs) provide a wide range of features for learning-process support in different domains, such as programming tasks with feedback, quizzes, execution traces, pseudo-code algorithms, reference material, worked solutions, adaptive features, and many others [5].

Reviewing related works, we used the following criteria. (1) Domain-independence requires using an open model for representing domains, including concepts, their relations, and laws. (2) Levels of Bloom's Taxonomy [3] (cognitive domain) that IPTs is aimed at. If the high levels are targeted, how much they are supported by developing low levels first? (3) Question and task types. They affect learning objectives and the available feedback. (4) Feedback types provided.

Some IPTs allow to generate tasks using templates and domain models [8, 11], but most of the use only predefined tasks [4, 6, 7, 9, 12, 16]. They are aimed at different levels of Bloom's Taxonomy objectives - comprehension [8], application [4, 11, 16], analysis and synthesis [6, 7, 9, 12, 14]; some of them include limited support for underlying levels. The systems utilise different question types - single choice [16], multiple choice [6, 8, 16], drag and drop [6], drop-down choice [6], key in solution [6], and code fragments [16]. The feedback ranges from pass/fail result, errors, and the last correct line [16] to task-oriented predefined hints [6] and derived hints [7, 11]. Most of the systems use hardcoded models (e.g. [6, 8]); few systems like [11] use formal models to represent the subject-domain laws.

Most IPTs are built for a single domain [5, 13]. A common issue is missing levels of Bloom's taxonomy [16]. E.g., [7, 9] are aimed at the synthesis level so their feedback provides no or minimal information on the application and comprehension levels.

To fully utilise adaptive abilities, ITS needs a large bank of different tasks. Most of the existing systems provide a limited set of predefined tasks [6, 7, 16]. In some systems, this set can be extended by a teacher [6, 7]. To uncover the

full power of adaptive assessments, ITS needs either a way to generate new learning tasks according to the situation or a large bank of various tasks. Some works advocate problem generation on the formal domain model [8, 11], but the generated problems require human verification that limits their number.

3 CompPrehension: Models, Architecture, and Workflow

We propose a new architecture of a multi-domain ITS on the comprehension level. Its main goal is the flexibility along the four main axes, represented by the plugin types: (1) **Domain** plugin encapsulates everything related to a subject domain, making other modules domain-independent; (2) **Backend** plugin allows interaction with different software reasoners, solving tasks by using provided laws and facts; (3) **Strategy** plugin assesses the level of demonstrated knowledge for the learner and chooses the next question; (4) **Frontend** plugin encapsulates the user interface. Plugins exchange information through the core, making pedagogical strategies agnostic of domain knowledge and vice versa.

Generating explanatory feedback to foster understanding limits ITS to closed-answer questions because determining exact reasons for all possible mistakes in open-answer questions is impossible. Four types of questions were identified for our system: **single choice**; **multiple choice**; **match**; **order** (ordering elements of the given set using them zero or more times). While these questions seem basic, they can be used for complex tasks like determining program trace (order question) or finding subexpression data types (match question).

Subject Domain Modelling. Bloom defined comprehension as “the form of understanding that the learner knows what was communicated and is able to make use of the material or idea that was communicated without relating it to other topics or seeing all its implications” [3]. In a formal system, we cannot use the first part of this definition (“knows what was communicate”) but can rely on the second (“is able to make use of the material”). In [1] we showed that comprehension-level modelling requires an axiomatic definition of all the studied properties of domain concepts. So the domain ontology is enhanced with rules: **positive rules** or productions to infer the solution from the problem definition and **negative rules** or constraints to catch mistakes in learners’ answers.

In comprehension-level tasks a complete sequence of correct steps always leads to a correct answer, so the negative rules can be inferred from positive rules by applying negation, giving the full set of possible mistakes; each mistake has a template for explanation generation. One positive rule can spawn several negative rules, i.e. there may be several ways to break one law. This makes negative rules the best way to measure learners’ knowledge. However, negative rules can be complex if there are several reasons for making the same mistake in the given situation; in this case, the system can either give a complex explanation or generate follow-up questions to determine the particular fault reason.

While rules define all important properties of the concepts, one type of knowledge remains attached to the concepts: the learner’s ability to identify individuals

of the given concept in the human-readable form of problem definition. Everything else is taught using rules.

Architecture. Figure 1 shows the proposed architecture and the chief responsibilities of its components. Strategy plugins analyse the learning situation (a sequence of correct and incorrect applications of domain laws during previous interactions of the learner) and form the question request for the next question or decide that the exercise is complete. Backend plugins integrate external reasoners for domain use.

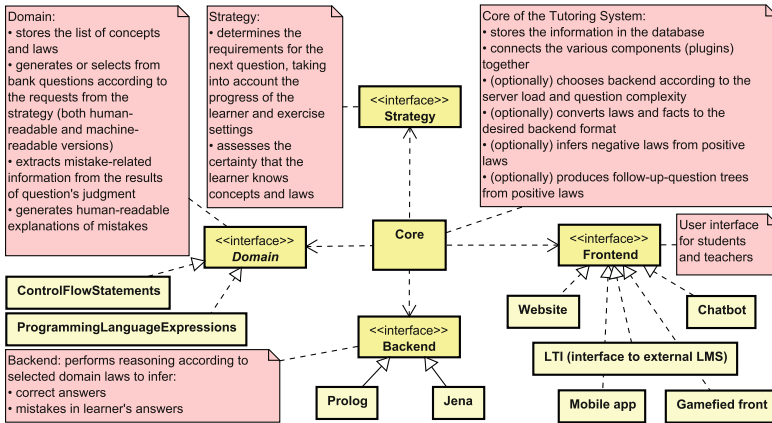


Fig. 1. CompPrehension architecture as a UML component diagram.

Domain plugins encapsulate everything related to a particular domain, including the formal models of the domain (i.e. its concepts and laws) and particular problem (individuals and facts), and also human-readable problem formulations and mistakes explanations. But domains are separated from both pedagogical decisions (the responsibility of strategies) and solving problems by logical inference (backends). This minimises the efforts required to add a new domain to the system and lets combine domains with different tutoring models and reasoners. Domain plugins also support tags, limiting the concepts and laws used in an exercise, This lets a single domain cover a group of similar tasks (e.g. the same tasks in different programming languages), re-using the rules.

Frontends encapsulate domain-independent user interface. This allows integration with modern learning management systems through standards like Learning Tools Interoperability (LTI), using providing accessibility to particular user categories or gamified experience, and using mobile interfaces or even messaging software to learn. Frontend plugins can transform complex questions into simpler ones if their user interface is restricted, e.g. ordering a set of elements can be transformed into a set of single-choice questions “choose the next element”.

Typical Workflow. As domains provide rules for finding a correct solution, the system can combine providing worked examples and guiding learners through the task-solving process. A typical workflow for task solving is as follows:

1. The strategy creates a question request based on the exercise settings, learner’s performance, and difficulty of the laws and concepts.
2. The domain generates a question based on the question request, including machine-solvable and human-readable versions.
3. The backend solves the question, finding the correct answer.
4. The learner provides a (possibly partial) answer through the frontend.
5. The core transforms the learner’s answer to backend facts.
6. The backend judges the answer, determining its correctness, mistakes, and their fault reasons (the sentence).
7. The domain interprets this sentence, transforms backend facts to domain law violations, and generates feedback.
8. The strategy adjusts feedback level.
9. The learner watches the feedback, possibly requesting more feedback.
10. The strategy chooses to ask a follow-up question, to continue with the current question, to generate a new question, or consider the exercise completed.

4 Scenario: Control Flow Statements

Control Flow Statements domain is aimed at developing comprehension of control-flow structures (sequences, alternatives, and loops) using the task of building an algorithm trace. The domain concepts are defined using an ontology – see Fig. 2a). The algorithm is represented as an abstract syntax tree, as in [2]. The trace is a linked list of `Act` instances connected by the `has_next` property. A pair of corresponding acts (`Act_begin` and `Act_end`) represents the boundaries of a control structure, containing the acts of the nested statements. The acts that evaluate control conditions of alternatives and loops contain the values of their conditions. Currently, the model contains 84 classes, including 29 algorithm elements, 5 trace acts, 33 kinds of mistakes, and 24 kinds of correct acts, 51 properties, 24 positive rules, and 35 negative rules (implemented as 28 and 55 Jena rules respectively).

The model uses `order` questions: learners must put execution acts of the control-flow statements in the correct order; any statement can be executed zero or more times. Figure 2b) shows an example of the problem text and a partial answer. The learner uses buttons to add new acts. The system can show a worked example, explaining the reasons for placing each step; ask the learner to build the entire trace; or show most of the trace and ask the learner to fill gaps.

$$\begin{aligned} & \text{conditional_loop}(L) \wedge \text{cond}(L, C) \wedge \text{end_of}(\text{loop_end}, L) \\ \Rightarrow & \text{on_false_consequent}(C, \text{loop_end}) \wedge \text{NormalLoopEnd}(\text{loop_end}) \end{aligned} \quad (1)$$

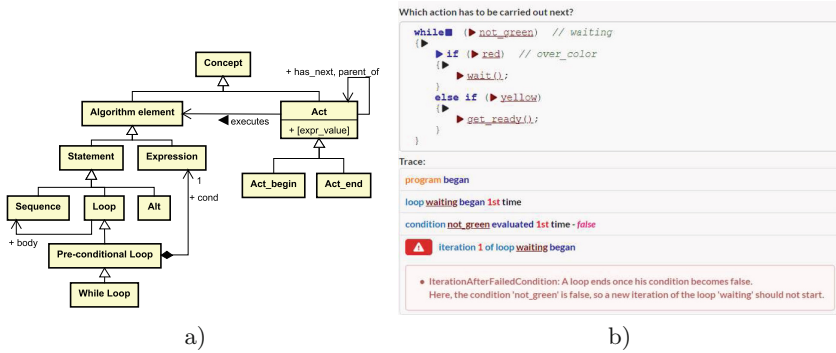


Fig. 2. Control Flow domain: a) partial hierarchy of concepts; b) question example.

$$\begin{aligned}
 & act_end(a) \wedge while_loop(L) \wedge cond(L, C) \wedge executes(a, C) \wedge \\
 & \quad expr_value(a, \mathbf{false}) \wedge student_next(a, b) \wedge executes(b, S) \wedge \quad (2) \\
 & body(L, S) \Rightarrow precursor(b, a) \wedge IterationAfterFailedCondition(b)
 \end{aligned}$$

In this example, the learner began the trace correctly, but then made a mistake, starting a loop iteration after its control condition evaluates to false. You can see the positive rule for finding the correct step at this point in (1) and the negative rule catching the mistake and its context (*precursor*) in (2). The system (strategy plugin) can then show a mistake explanation (as shown in Fig. 2b) or ask follow-up questions to find the exact fault reason.

5 Conclusion and Future Work

We designed a domain-independent comprehension-level ITS CompPrehension and developed a working prototype, including two subject domains (Control Flow Statements and Programming Language Expressions), four backends (Prolog, SWRL, Jena rules, SPARQL scripts), LTI Frontend, and a test strategy. The domains contain tag sets for C++ and Python programming languages.

The system is capable of selecting a problem from the problem base according to the learner's performance, solving it using software reasoning, grading answers, determining fault reasons, showing explanatory feedback, asking follow-up questions, and showing worked examples. Integrating a new subject domain requires developing one domain plugin.

The developed system exposes the properties of the domain concepts through simple questions, verifying and developing their comprehension for learners. This supports learners in doing higher-level tasks by ensuring that they understand the concepts they use. The system is limited to the comprehension level and is most effective for introductory courses in new domains like programming, mathematics, or natural languages when learners need to understand a significant number of new concepts and learn to handle them in typical situations.

The further work includes developing problem banks (by mining existing open-source code), implementing more strategies for different kinds of assessments, enhancing domain models to cover more tasks and programming languages, creating follow-up question trees based on learners' misconceptions, and verifying the domain-independence by developing English Word Order domain.

References

1. Anikin, A., Sychev, O.: Ontology-based modelling for learning on bloom's taxonomy comprehension level. In: Samsonovich, A.V. (ed.) BICA 2019. AISC, vol. 948, pp. 22–27. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-25719-4_4
2. Atzeni, M., Atzori, M.: CodeOntology: RDF-ization of source code. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2017). https://doi.org/10.1007/978-3-319-68204-4_2
3. Bloom, B.S., Engelhart, M.B., Furst, E.J., Hill, W.H., Krathwohl, D.R.: Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook 1: Cognitive Domain. Longmans Green, New York (1956)
4. Brusilovsky, P., Su, H.-D.: Adaptive visualization component of a distributed web-based adaptive educational system. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 229–238. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47987-2_27
5. Crow, T., Luxton-Reilly, A., Wuensche, B.: Intelligent tutoring systems for programming education. In: Proceedings of the 20th Australasian Computing Education Conference on - ACE 2018. ACM Press (2018). <https://doi.org/10.1145/3160489.3160492>
6. Fabric, G.V.F., Mitrovic, A., Neshatian, K.: Adaptive problem selection in a mobile python tutor. In: Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization. ACM (2018). <https://doi.org/10.1145/3213586.3225235>
7. Jeuring, J., Gerdes, A., Heeren, B.: A programming tutor for haskell. In: Zsók, V., Horváth, Z., Plasmeijer, R. (eds.) CEFP 2011. LNCS, vol. 7241, pp. 1–45. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32096-5_1
8. Kumar, A.N.: Generation of problems, answers, grade, and feedback—case study of a fully automated tutor. *J. Educ. Res. Comput.* **5**(3), 3 (2005). <https://doi.org/10.1145/1163405.1163408>
9. Lane, H.C., VanLehn, K.: Teaching the tacit knowledge of programming to novices with natural language tutoring. *Comput. Sci. Educ.* **15**(3), 183–201 (2005). <https://doi.org/10.1080/08993400500224286>
10. Mitrovic, A., Koedinger, K.R., Martin, B.: A comparative analysis of cognitive tutoring and constraint-based modeling. In: Brusilovsky, P., Corbett, A., de Rosis, F. (eds.) UM 2003. LNCS (LNAI), vol. 2702, pp. 313–322. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44963-9_42
11. O'Rourke, E., Butler, E., Díaz Tolentino, A., Popović, Z.: Automatic generation of problems and explanations for an intelligent algebra tutor. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) AIED 2019. LNCS (LNAI), vol. 11625, pp. 383–395. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23204-7_32

12. Papadakis, S., Kalogiannakis, M., Zaranis, N.: Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. *Int. J. Mob. Learn. Organ.* **10**(3), 187 (2016). <https://doi.org/10.1504/ijmlo.2016.077867>
13. Pillay, N.: Developing intelligent programming tutors for novice programmers. *ACM SIGCSE Bull.* **35**(2), 78–82 (2003). <https://doi.org/10.1145/782941.782986>
14. Price, T.W., Dong, Y., Lipovac, D.: iSnap. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. ACM (2017). <https://doi.org/10.1145/3017680.3017762>
15. Sychev, O., Denisov, M., Anikin, A.: Verifying algorithm traces and fault reason determining using ontology reasoning. In: Taylor, K.L., Gonçalves, R., Lécué, F., Yan, J. (eds.) Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 19th International Semantic Web Conference (ISWC 2020), Globally online, 1–6 November 2020 (UTC). CEUR Workshop Proceedings, vol. 2721, pp. 49–54. CEUR-WS.org (2020). <http://ceur-ws.org/Vol-2721/paper495.pdf>
16. Yoo, J., Pettey, C., Seo, S., Yoo, S.: Teaching programming concepts using algorithm tutor. In: EdMedia+ Innovate Learning pp. 3549–3559. Association for the Advancement of Computing in Education (AACE) (2010)



Learning Logical Reasoning : Improving the Student Model with a Data Driven Approach

Roger Nkambou¹(✉), Janie Brisson¹, Serge Robert¹, and Ange Tato^{1,2}

¹ Université du Québec à Montréal, Montreal, Canada

² BMU, Rohtak, India

Abstract. In our previous works, we presented Logic-Muse as an Intelligent Tutoring System that helps learners improve logical reasoning skills in multiple contexts. Logic-Muse components were validated and argued by experts throughout the designing process (ITS researchers, logicians and reasoning psychologists). A Bayesian network with expert validation has been developed and used in a Bayesian Knowledge Tracing (BKT) process that allows the inference of the learner's behaviour. This paper presents an evaluation of the learner components of Logic-Muse. We conducted a study and collected data from nearly 300 students who processed 48 reasoning activities. This data was used in the development a psychometric model, a key element for initializing the learner's model and for validating and improve the structure of the initial Bayesian network built with experts.

1 Introduction

Decades of research in cognitive science show that human reasoning does not function accordingly to the rules of formal logic (e.g. [6, 9, 14, 17]). When looking for solutions to improve human skills in this area, several questions arise: what's important in the assessment of logical competence? What are the phenomena involved in the acquisition of logical reasoning skills? What should be the characteristics of an Intelligent Tutoring System (ITS [15]) designed to support this learning? These questions cannot be answered without an appropriate understanding of human reasoning processes and the active participation of relevant experts, including logicians, psychologists, education professionals, and IT specialists. By bringing together specialists from these different fields, the Logic-Muse project aims to study the basics of logical reasoning skills acquisition, to understand the difficulties associated with this learning, and to create an STI that can detect, diagnose, and correct reasoning errors in various situations. Logic-Muse's architecture is based on classical ITS with its three usual components: the knowledge domain expert, the learner, and the tutor. Each of them has been previously elicited, validated, and argued by the experts. Logic-Muse in

NSERC Discovery Grant.

© Springer Nature Switzerland AG 2021

A. I. Cristea and C. Troussas (Eds.): ITS 2021, LNCS 12677, pp. 60–67, 2021.

https://doi.org/10.1007/978-3-030-80421-3_7

its current version implements these components for conditional reasoning and offers a set of activities allowing a learner to develop his logical reasoning skills in several situations defined by the domain experts. In the following sections, we describe how a data-driven approach was used to complement the bayesian network built by experts, leading to an effective predictive student model.

2 The Student Model in Logic-Muse

The main part of Logic-Muse’s learner model is made of a Bayesian network (BN) whose nodes are the 96 units of knowledge related to reasoning, as identified by the domain experts [16]. Some latent nodes are directly connected to the reasoning activities (items). The skills involved in the Bayesian network include the inhibition of exceptions to the premises, the generation of counterexamples to the conclusion, and the ability to manage all the relevant models for familiar, counterfactual, and abstract situations [13]. The system’s estimate of student skill acquisition is continually updated every time a student answers to an item, and that answer is used as evidence by the system to re-compute the probabilities in the network through the bayesian inference. The next item to be proposed to the learner is chosen by the tutor based on the current status of the network.

Item Bank for Conditional Reasoning. Items used for conditional reasoning in Logic-Muse are the four logical forms of conditional reasoning: the Modus Ponendo Ponens (MPP), the Modus Tollendo Tollens (MTT), the Affirmation of the Consequent (AC), and the Denial of the Antecedent (DA). Each of these four logical forms are declined in 3 levels of content, based on a developmental model of conditional reasoning. Markovits (2013) suggests that the more a premise has familiar content, the more rapid and easy the retrieval of a counter-example to invalid conclusions will be. For the AC and DA inferences, counterexamples are alternative antecedents, i.e. antecedents that differ from P but imply the consequent Q. For the MPP and MTT inferences, counterexamples are disabling conditions, i.e. a condition that prevents the antecedent P from implying the consequent Q. Many studies have shown that the number of potential counterexamples [6, 17] or the strength of association between them and the premise [8] determines the approval rate of the four forms of conditional inference. For example, with the premise “If a rock is thrown at a window, then the window will break”, reasoners will tend to accept the AC inference (a window is broken, therefore a rock was thrown at it) less often than with the premise “If a finger is cut, then it will bleed” (a finger bleeds, therefore it has been cut). The reason is that the former premise contains many alternative antecedents, like throwing a chair, a car accident, a tropical storm, etc., that are counterexamples to the putative conclusion, while the latter contains fewer of such antecedents (a finger is crushed, etc.). The counterfactual level contains a reversed causal rule known to be false. It allows the generation of an unrealistic category of alternative antecedents. For example, with the counterfactual premise “If a feather is

thrown at a window, then the window will break”, one could generate a counterfactual alternative antecedent like “throwing tissue on a window” or a disabling condition like “The window was strong enough to stay intact”. The abstract level contains if-then rules linking made-up words, e.g. “If one blops, then one will become plede”. This level requires an abstract representation of the premises that can generate abstract alternative antecedents. According to [13], when a reasoner reaches this level, he has a complete understanding of the implicative link: he understands that for a conditional premise (unknown or abstract), an alternative antecedent can be generated regardless of background knowledge. We thus have a total of 16 item classes (item nodes in the BN).

3 Training a CDM Model to Validate the BN and to Initialize the Learner Model

Logic-Muse’s learner model aims to represent user knowledge as accurately as possible using the Bayesian network we created. It allows for diagnosis and modeling of the learner’s current state of mastery for each identified skill. Validity and reliability are thus very important features of this model. The CDM model is built using the item bank, the Q-Matrix as well as data from all student responses to items [7]. The Q-Matrix (Fig. 1) connects items categories to the involved skills.

	Causal Familiar			Causal Contrary to Fact			Abstract		
	Inhibit	Generate	Manage	Inhibit	Generate	Manage	Inhibit	Generate	Manage
MPP FFD	1	0	0	0	0	0	0	0	0
MPP FMD	1	0	0	0	0	0	0	0	0
MIT FFD	1	0	1	0	0	0	0	0	0
MIT FMD	1	0	1	0	0	0	0	0	0
AC FMA	0	1	1	0	0	0	0	0	0
AC FFA	0	1	1	0	0	0	0	0	0
DA FMA	0	1	1	0	0	0	0	0	0
DA FFA	0	1	1	0	0	0	0	0	0
MPP CCF	1	0	0	1	0	0	0	0	0
MIT CCF	1	0	1	1	0	1	0	0	0
AC CCF	0	1	1	0	1	1	0	0	0
DA CCF	0	1	1	0	1	1	0	0	0
MPP A	1	0	0	1	0	0	1	0	0
MIT A	1	0	1	1	0	1	1	0	1
AC A	0	1	1	0	1	1	0	1	1
DA A	0	1	1	0	1	1	0	1	1

Fig. 1. Q-matrix for conditional reasoning

Data Collection and Preparation. Participants and procedures. A total of 294 participants were recruited online via the Prolific Academic platform. Materials. There are 3 items per each of the 16 items classes. This classification was done to obtain a more reliable measure for each competence class. For each of the 48 items, participants had to choose between three answers (the valid one, the invalid typical one, and the invalid atypical one). Answers were encoded as

“1” for valid and “0” otherwise. We then had to choose between three possible response matrices according to the number of valid responses for the three repeated measures for each of the 16 categories. Each category is encoded as 1 if the participant was successful for at least 2 out of 3 items. This particular threshold was chosen out of consistency with previous modeling choices: a majority of a successful response is the criteria to activate a competence node in our Bayesian Network.

CDM Model Type Selection and Training. We then trained a CDM model on the 294 response patterns, which allowed us to estimate various parameters: posterior probabilities, the goodness of fit indicator, guess (the probability that a learner could correctly answer an exercise without having the necessary skills), slip (the probability of a bad solution while the learner had the necessary skills), tetrachoric correlations, and marginal skill probabilities. We opted for a DINA (Deterministic Input, Noisy “And” gate) model since it makes the same assumption we made in our modeling of skills: the learner must have mastered all the related skills to be successful in an item.

3.1 Model Parameters

Goodness of Fit. As an absolute indicator of goodness of fit for the CDM model, we opted for the item pairwise χ^2 measure [3]. This measure indicates that the model is inadequate if the p-value of the maximal item pairwise χ^2 measure is above the 0.01 significance level [10]. In the present case, the χ^2 test results were $\chi^2 = 33.95$ with p -value = 6.793316×10^{-7} , which is clearly below any significance threshold and thus indicates that the present model is adequate.

Guess, Slip and Item Discrimination Index (IDI). We noted a high guess and low IDI for both MTT with familiar content. This can be explained by the fact that a biconditional interpretation of the major premise leads to the correct answer to the MTT, regardless of conditional skills. It is also interesting to note that all items involving few alternatives have a lower IDI than their many alternative counterparts. Beyond these remarks, however, all items have good IDI values. This is especially true for the MPP counterfactual, which suggests that successful completion of this task is the best way to ensure that conditional reasoning has been fully mastered. As discussed below, this finding is consistent with results obtained for marginal skill probabilities.

Tetrachoric Correlations. Tetrachoric correlations between skills are shown in Fig. 2. Based on our sample size, correlation scores over 0.33 are considered significant with $\alpha = 0.05$ [11]. Using this criterion for this analysis, one pattern that stands out is that skills with familiar content correlate highly with other skills of the same content level, but not with counterfactual and abstract level skills. However, counterfactual level skills correlate well with themselves

	Skill Correlations									α
	InhibitFam	GenerateFam	ManageFam	InhibitCF	GenerateCF	ManageCF	InhibitAbs	GenerateAbs	ManageAbs	
InhibitFam	1	0,85	0,9	-0,06	0,05	0,04	0,19	0,15	0,15	0,36
GenerateFam	0,85	1	0,78	0,05	0,06	0,16	0,2	0,16	0,17	0,38
ManageFam	0,9	0,78	1	0,22	0,06	0,04	0,2	0,15	0,16	0,39
InhibitCF	-0,06	0,05	0,22	1	0,64	0,6	0,67	0,5	0,52	0,46
GenerateCF	0,05	0,06	0,06	0,64	1	0,15	0,31	0,4	0,35	0,34
ManageCF	0,04	0,16	0,04	0,6	0,15	1	0,35	0,41	0,43	0,35
InhibitAbs	0,19	0,2	0,2	0,67	0,31	0,35	1	0,3	0,31	0,39
GenerateAbs	0,15	0,16	0,15	0,5	0,4	0,41	0,3	1	0,34	0,38
ManageAbs	0,15	0,17	0,16	0,52	0,35	0,43	0,31	0,34	1	0,38
α	0,36	0,38	0,39	0,46	0,34	0,35	0,39	0,38	0,38	0,38

Fig. 2. Skill correlations

and abstract level skills, and vice versa. Overall, these findings suggest a clear separation between the familiar level of content and the other two levels.

Marginal Skill Probabilities. The hardest skill to master seems to be “Inhibit counterfactual” (44.3%), which requires learners to inhibit disabling conditions to counterfactual conditional statements. This observation is consistent with the fact that the MPP counterfactual has the highest item discrimination index. However, these findings seem at odds with our initial psychological model. Indeed, the latter considers abstract skills to be the hardest and least mastered ones; consequently, abstract skills should be better indicators of mastery for conditional reasoning than counterfactual skills, not the other way around. With hindsight, one could however argue that inhibiting exceptions to a rule known to be false, such as “If I throw ketchup on a shirt, then the shirt will be clean”, might prove harder than simply inhibiting an imaginary one. This might also be the reason why performances with counterfactual content are similar, or even sometimes worse than performances with abstract content. Finally, the much higher probability of familiar level skills separates them from the other two levels’ skills, as the difference between abstract level skills probabilities and their lower-scoring counterfactual counterpart is not nearly as pronounced.

3.2 BN Initialization Through the CDM Model

The CDM allows us to predict a user’s probability of mastering the overall competence (root node) via its pre-test results. For this, we use the a posteriori probabilities obtained. To do so, from a learner’s vector of competence, we seek the line of the a posteriori matrix containing the same vector or a similar one. The joint probability matching this pattern, calculated based on the probabilities associated with each skill, is then used as the a priori likelihood (prior probability) of mastering the root node. The first matrix obtained indicated that the most probable vector was the “111 000 000”, which means a learner masters the three skills for the familiar level of content, but no competence for the counterfactual and the abstract levels. This seems to show a separation between the familiar content and the other two levels. However, given the very high number of possible combinations for these nine skills, the other vectors were very numerous, showed very small probabilities and a lot of them were equiprobable. We

thus decided to use vector classes based on the types of skills identified in our model.

		Skill Class Chunk Probabilities			
		Successful	Failed	OnlySuccessful	OnlyFailed
	Familiar	0,95	0,05	0,55	0,00
CausalCounterFactual		0,30	0,70	0,00	0,10
	Abstract	0,37	0,63	0,01	0,03
	Inhibit	0,39	0,61	0,07	0,01
	Generate	0,40	0,60	0,10	0,03
	Manage	0,41	0,59	0,10	0,02

Fig. 3. Skill class chunk probabilities

The a posteriori matrix based on our classification is shown in the Fig. 3. The left column represents the six vectors classes we created. Each content level (familiar, counterfactual, abstract) represents a triplet of skills (inhibit, generate, and manage) for this level, while each skill (inhibit, generate, manage) refers to the same skill in all three levels. In the first row, “successful” means all three skills for the corresponding level (or all three levels for the corresponding skill) are mastered, regardless of the performance in the other two levels or skills. For example, being successful in the familiar class may refer to vectors such as “111 111 111”, “111 011 111”, “111 011 011” or even “111 000 000”. In the second row, “failed” means the opposite: all three skills for the corresponding level (or all three levels for the corresponding skill) are not mastered, irrespective of a learner’s performance in the other two levels or skills. The “only successful” row means that, of all three skills or levels, only the corresponding one is fully (111) mastered. For example, being only successful in the familiar class may come from the vector “111 011 011” or even “111 000 000”, but not “111 111 010” or “111 000 111”. The “Only failed” row refers to the opposite situation: only the corresponding skill or level triple is failed (at least one zero in the triplet), while the two other triplets are fully mastered (111). Results for the different difficulty levels point to graduation in performance. As expected, results show that the “Familiar” level stands out in both “Successful” and “Only Successful” categories: this level is seldom failed and is usually the only one mastered by learners. On the other opposite, the ‘Causal Counterfactual’ level is the level most often failed and exclusively failed (‘Only failed’) by learners. While the difference in mastery level is not as pronounced as in the ‘Familiar’ case, it is nonetheless noteworthy, as both establish clear upper and lower boundaries for learning and performance. For instance, while the probability that learners exclusively fail the ‘Familiar’ case is close to zero, the same goes for the probability that learners exclusively master the ‘CausalCounterFactual’ level. As for the in-between ‘Abstract’ level, results show that it is nonetheless a difficult level to master, closer to the ‘CounterFactual’ level. Finally, results for the different skills (‘Inhibit’, ‘Generate’, ‘Manage’) are inconclusive, as no clear skill pattern can be found across the different difficulty levels.

Summary of the Results. In summary, analysis of CDM model data provided two major findings regarding the underlying developmental theory of our Bayesian Network. First, there seems to be a clear separation between the familiar level of content and the other two levels. Moreover, the hardest skill is not an abstract one, like our psychological model assumes, but a counterfactual one, i.e. the “inhibit counterfactual” skill. This surprising result, evidenced both by item discrimination indices and marginal skill probabilities, can be explained by the difficulty to inhibit exceptions, which are in fact realistic situations, to a false rule. Consequently, counterfactual level skills may very well prove harder to master than Abstract level counterparts.

3.3 The Bayesian Knowledge Tracing

Logic-Muse supports the Bayesian Knowledge Tracing process (BKT) [5]. BKT uses the Bayesian network to capture students’ knowledge which allows inferring the probability of mastering a skill from a specific response pattern [4]. Student performance is the observed variable and student knowledge is the latent data. The BKT is a special case of the Hidden Markov Model where student knowledge is represented as a set of binary variables (the skill is either mastered by the student or not). Observations are also binary: a student gets a problem either right or wrong [18]. However, there is a certain probability (G, the Guess parameter) that the student will give a correct response. Correspondingly, a student who does know a skill generally will give a correct response, but there is a certain probability (S, the Slip parameter) that the student will give an incorrect response. The standard BKT model is thus defined by four parameters [1]: initial knowledge, learning rate (learning parameters), slip and guess (mediating parameters). In general, the BKT applies prior knowledge (L0) and the probability to learn the applied concept ($p(T)$) to measure the progress of student learning. The process works well in Logic-Muse Intelligent Tutoring System. It has also been successfully used in a variety of systems including computer programming [12], reading skills [2], etc.

4 Conclusion

In this paper, we presented the student model in Logic-Muse Intelligent Tutoring System which aims at helps learners develop logical reasoning skills. A bayesian student model was first presented including its validation using a cognitive diagnosis model trained on data collected from 294 learners (each student solved 48 reasoning problems). The resulting CDM model was also used for initializing the Bayesian network for a new learner who goes through a pretest (composed of those 48 items) during its first connection to the system. The Bayesian network is the main support of the Bayesian knowledge tracing (BKT) in Logic-Muse.

References

1. Baker, R.S.J., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 406–415. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69132-7_44
2. Beck, J.E., Chang, K.: Identifiability: a fundamental problem of student modeling. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 137–146. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73078-1_17
3. Chen, J., de la Torre, J., Zhang, Z.: Relative and absolute fit evaluation in cognitive diagnosis modeling. *J. Educ. Meas.* **50**(2), 123–140 (2013)
4. Conati, C., Gertner, A., Vanlehn, K.: Using bayesian networks to manage uncertainty in student modeling. *User Model. User-Adapt. Interact.* **12**(4), 371–417 (2002)
5. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.* **4**(4), 253–278 (1994)
6. Cummins, D.D., Lubart, T., Alksnis, O., Rist, R.: Conditional reasoning and causation. *Memory Cogn* **19**(3), 274–282 (1991)
7. De La Torre, J.: A cognitive diagnosis model for cognitively based multiple-choice options. *Appl. Psychol. Meas.* **33**(3), 163–183 (2009)
8. De Neys, W., Schaeken, W., D’Ydewalle, G.: Inference suppression and semantic memory retrieval: every counterexample counts. *Memory Cogn.* **31**(4), 581–595 (2003)
9. Gilovich, T., Griffin, D., Kahneman, D.: *Heuristics and Biases: The Psychology of Intuitive Judgment*. Cambridge University Press, Cambridge (2002)
10. Groß, J., Robitzsch, A., George, A.: Cognitive diagnosis models for baseline testing of educational standards in math. *J. Appl. Stat.* **43**(1), 229–243 (2016)
11. Guilford, J.P., Lyons, T.C.: On determining the reliability and significance of a tetrachoric coefficient of correlation. *Psychometrika* **7**(4), 243–249 (1942)
12. Kasurinen, J., Nikula, U.: Estimating programming knowledge with bayesian knowledge tracing. In: ACM SIGCSE Bulletin, vol. 41, pp. 313–317. ACM (2009)
13. Markovits, H.: The development of abstract conditional reasoning. In: *The Development of Thinking and Reasoning*, pp. 83–104. Psychology Press (2013)
14. Markovits, H., Vachon, R.: Reasoning with contrary-to-fact propositions. *J. Exp. Child Psychol.* **47**(3), 398–412 (1989)
15. Nkambou, R., Mizoguchi, R., Bourdeau, J.: *Advances in Intelligent Tutoring Systems*, vol. 308. Springer, Heidelberg (2010). <https://doi.org/10.1007/978-3-642-14363-2>
16. Tato, A., Nkambou, R., Brisson, J., Robert, S.: Predicting learner’s deductive reasoning skills using a bayesian network. In: André, E., Baker, R., Hu, X., Rodrigo, M.M.T., du Boulay, B. (eds.) AIED 2017. LNCS (LNAI), vol. 10331, pp. 381–392. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61425-0_32
17. Thompson, V.A.: Interpretational factors in conditional reasoning. *Memory Cogn.* **22**(6), 742–758 (1994)
18. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized bayesian knowledge tracing models. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS (LNAI), vol. 7926, pp. 171–180. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39112-5_18



Checking Method for Fake News to Avoid the Twitter Effect

Téo Orthlieb^(✉), Hamdi Ben Abdesslem, and Claude Frasson

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal
H3C 3J7, Canada

{teo.orthlieb, hamdi.ben.abdesslem}@umontreal.ca,
frasson@iro.umontreal.ca

Abstract. The recent blocking of President Trump's twitter account has raised awareness of the danger of the impact of fake news and the importance of detecting it. Indeed, if one can doubt information, ignoring what is true or false it can lead to a loss of confidence in the decisions and other dangers. The objective of this paper is to propose an automatic method for fact checking using Knowledge Graphs, such as Wikipedia. Knowledge Graphs (KGs) have applications in many tasks such as Question Answering, Search Engines and Fact Checking, but they suffer from being incomplete. Recent work has focused on answering this problem with an abstract embedding of the KG and a scoring function, yielding results that are not easily interpretable. On the other hand, Path Ranking methods answer this problem with deductions represented by alternative paths in the KG, easily understood by a human. Favoring the Path Ranking approach for its interpretability, we propose an attention-based Path Ranking model that uses label information in the KG, making the model easily transferable between datasets, allowing us to leverage pretraining and demonstrate competitive results on popular datasets.

Keywords: Fact checking · Knowledge graphs · Deep learning

1 Introduction

Large Knowledge Graphs (KG) such as Wikidata, FreeBase or WordNet aim at storing facts of a domain in a structured manner, which is typically done with a multi-relational directed graph, where each node is an entity and each edge is labelled with a relation. We thus denote a fact by a triplet (h, r, t) , with h and t the head and tail entities, and r the relation, for example (Sean Connery, Profession, Actor).

In this context, the task of fact checking can be reformulated as a task of Link Prediction in the Knowledge Graph.

A popular approach to link prediction is Knowledge Graph Embeddings, which operate on nodes and edges as vectors in a latent space and use a scoring function to assert if a certain triplet is likely to be true. For instance, TransE [2] embeds both relations and entities in the same space, and ensures that $h + r \approx t$.

However, one inconvenient shared by Knowledge Graph Embeddings is that predictions are hard to explain, because the model operates only in a latent space that doesn't necessarily make sense for a human.

On the other hand, path ranking methods base their decisions on alternative paths in the KG.

For example, if the nationality of Sean Connery was not documented, a path ranking model could infer that because Sean Connery was born in Edinburgh, and because Edinburgh is the capital of Scotland, then Sean Connery was Scottish. These kinds of reasoning are understandable from a human perspective, and even if the model might be wrong, they still provide a useful insight to the user.

This article focuses on building a path ranking kind of model that is transferable between different Knowledge Graphs, and also proposes a new dataset that is more adapted for fact checking.

2 Related Work

A notable step for path ranking methods in Knowledge Graphs was [5] in which the Path Ranking Algorithm (PRA) from an earlier work [6] was adapted for KGs.

Simply put, the training procedure is as follows, for each (h, r, t) a number of depth-limited random walks are executed, and the associated meta-paths are then scored by how often they can reach the tail entity t . At prediction time given h and r a number of depth-limited random walks are performed again, and the end node of every paths weighed by the score of their associated meta-paths are combined to yield a distribution of end nodes. Predictions for the tail entity t are then ordered by their likelihood in this distribution.

The Path Ranking Algorithm is thus a purely statistical learning model, and provides a strong baseline that works well even with few data.

More recently, Path Ranking techniques incorporating deep learning have been introduced:

- DeepPath [7] instead frames the task of path ranking as a Reinforcement Learning task. For a given (h, r, t) the agent is tasked to find the most informative paths from h to t .
- Path Ranking with Attention to Type Hierarchies [10] takes another approach to path ranking by incorporating type hierarchies. The reasoning is that for any entity, different level of abstractions can be relevant. For instance, we can think of a Fork as Cutlery, Tableware or simply Ware.

However, they only result in a slight improvement over PRA on FB15k-237 [8].

All the path ranking methods above share one characteristic that we must detail. For any KG dataset, every nodes and relations are typically associated with a unique ID that differs between datasets. These IDs have the advantage of being independent of language and resilient to homonyms, but they have the disadvantage of not being transferable between datasets and they are not as informative as a label.

We explore representing nodes and relations with their labels directly instead of an ID, allowing the model make use of pretraining, which has been shown to be very effective in language tasks over the last decade [11].

3 Architecture

At training time, the model is given a fact triplet (h, r, t) as well as random depth limited meta-paths linking h to t . The model is tasked to output the probability that the fact triplet is valid. In order to do so the model uses various modules:

- **Label Embedding:** the model embeds all the nodes and relations in the input using the word level embedding on their labels. This transforms labels into vectors which the model can then use subsequently.
- **Meta-path Encoding:** every meta-path is encoded with a GRU, in order to transform each meta-path into a single vector.
- **Relation Encoding:** the model uses all the meta-path as a context in a transformer-type model [14] to get a new representation of r . In other words, this is the module in which the model thinks about all the relations he knows between the head and tail entity.
- **Classification Layer:** h , r , and t are fed through a final feedforward layer to yield the estimate probability that the fact represented by (h, r, t) is true.

Through experiments, we found the following hyperparameters to work well:

- Depth of paths = 4
- Embedding size = 50
- Learning Rate Cycling between [0.005; 0.1]

4 Datasets

Unfortunately, we couldn't find a dataset complete with negative examples used in *DeepPath* or *Path Ranking with Attention* to Type Hierarchies, and were not able to run the provided method to generate negative examples. So, we made our own method to generate negative examples as well as a new bigger dataset from WikiData, and published everything here: https://github.com/Inspirateur/KG_Datasets.

Since the previous papers don't use the same method to generate negative examples, comparing against them would be unfair, hopefully this will not be a problem for future papers since the complete datasets we use are public.

A random baseline yields 16% Maximum Average Precision (MAP) on FB15k-237 and WikiData, and our model (TAP-KG) reaches 78% MAP on FB15k-237 and 51% MAP on WikiData, which seems to be a harder dataset.

5 Visualization of Results

The ability to understand predictions from our model is a very important point, and we're able to do it by inspecting the **Relation encoding** part of the model to see which paths it relied the most on to make the prediction, illustrated in Fig. 1.

True: Wayne Knight people/nationality United States

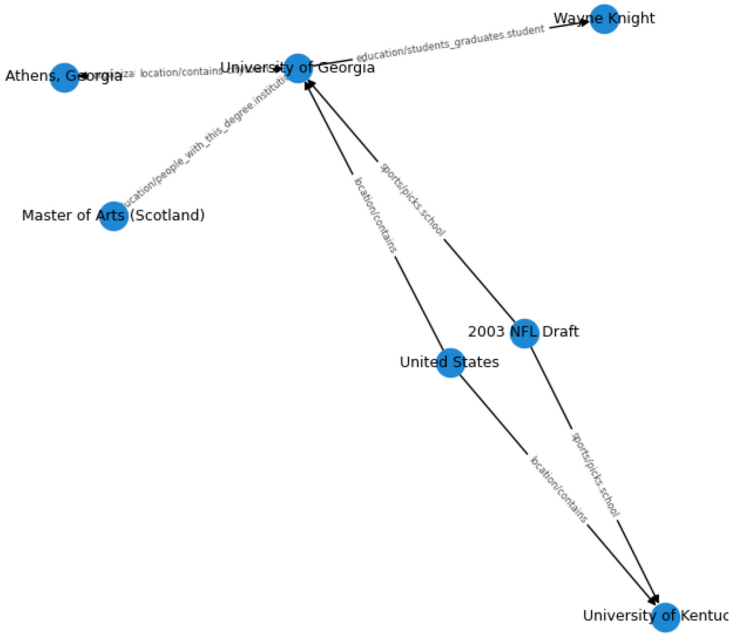


Fig. 1. The model deduces that Wayne Knight is a USA citizen because he graduated University of Georgia which is located in the USA

6 Conclusion

In order to help fighting against the rapid spread of fake news, we proposed an automatic fact checking model with interpretable results. Indeed, even though deep learning models have demonstrated great performance in recent years, they are often thought of as “black box” because it is challenging to understand what’s happening inside. However, the datasets used to train and test the model are not perfect. Since they are usually made automatically, some tasks like predicting that England contains Birmingham when given “England”, “contains” are nonsensical, because there are a lot of cities in England and the model could give many correct answers before giving “Birmingham”.

An improvement for the future would thus be to construct better tasks to train and test models on link prediction in knowledge graphs.

Acknowledgment. We acknowledge NSERC-CRD (National Science and Engineering Research Council Cooperative Research Development), Prompt, and BMU (Beam Me Up) for funding this work.

References

1. Shao, C., Ciampaglia, G.L., Flammini, A., Menczer, F.: Hoaxy: a platform for tracking online misinformation. In: Proceedings of the 25th International Conference Companion on World Wide Web - WWW 2016 Companion, Montréal, Québec, Canada, pp. 745–750 (2016)
2. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, Red Hook, NY, USA, pp. 2787–2795 (2013)
3. Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J.: RotatE: knowledge graph embedding by relational rotation in complex space (2019). [arXiv:1902.10197](https://arxiv.org/abs/1902.10197) [cs, stat]
4. Wang, R., Li, B., Hu, S., Du, W., Zhang, M.: Knowledge graph embedding via graph attenuated attention networks. *IEEE Access* **8**, 5212–5224 (2020)
5. Lao, N., Mitchell, T., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK, pp. 529–539 (2011)
6. Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.* **81**(1), 53–67 (2010)
7. Xiong, W., Hoang, T., Wang, W.Y.: DeepPath: a reinforcement learning method for knowledge graph reasoning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, pp. 564–573 (2017)
8. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D Knowledge Graph Embeddings (2017). CoRR, vol. abs/1707.01476
9. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an Architecture for Never-Ending Language Learning (2010)
10. Liu, W., Daruna, A., Kira, Z., Chernova, S.: Path ranking with attention to type hierarchies. *AAAI* **34**(03), 2893–2900 (2020)
11. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2018). CoRR, vol. abs/1810.04805
12. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1532–1543 (2014)
13. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches (2014). CoRR, vol. abs/1409.1259
14. Vaswani, A., et al.: Attention Is All You Need (2017). CoRR, vol. abs/1706.03762



Comparing Bayesian Knowledge Tracing Model Against Naïve Mastery Model

Vanesa Getseva and Amruth N. Kumar^(✉)

Ramapo College of New Jersey, Mahwah, NJ 07430, USA
{vgetseva, amruth}@ramapo.edu

Abstract. We conducted a study to see if using Bayesian Knowledge Tracing (BKT) models would save time and problems in programming tutors. We used legacy data collected by two programming tutors to compute BKT models for every concept covered by each tutor. The novelty of our model was that slip and guess parameters were computed for every problem presented by each tutor. Next, we used cross-validation to evaluate whether the resulting BKT model would have reduced the number of practice problems solved and time spent by the students represented in the legacy data. We found that in 64.23% of the concepts, students would have saved time with the BKT model. The savings varied among concepts. Overall, students would have saved a mean of 1.28 min and 1.23 problems per concept. We also found that BKT models were more effective at saving time and problems on harder concepts.

Keywords: Programming tutors · Bayesian Knowledge Tracing · Evaluation

1 Introduction

Student model is essential for facilitating adaptation in intelligent tutoring systems. Bayesian Knowledge Tracing (Corbett and Anderson 1992) is one of the more popular methods of modeling student's knowledge. The model consists of four parameters per concept. In the past, in order to estimate the four parameters, researchers have used baseline approach (Beck 2007), bounded guess and slip approach, Dirichlet Priors (Beck and Chang 2007), contextual estimation (Baker et al. 2008) and empirical probabilities (Hawkins et al. 2014). In this study, we present an empirical approach based on legacy data collected by intelligent tutors. Our approach differs from earlier attempts in that we calculate guess and slip parameters for each problem, not just each concept. We used the calculated BKT model to evaluate its effectiveness in terms of time and effort saved for the students represented in the legacy data.

Currently, our tutors use a naïve mastery model to determine whether the student has learned a concept during practice. In this model, a student is said to have mastered a concept if the student solves at least 2 problems on the concept and solves at least 60% of the problems correctly. For the concepts that do not occur as frequently in programming, the mastery criterion was set to at least 1 problem solved and at least 50% of the problems solved correctly. If the Bayesian Knowledge Tracing model could

determine that a student has learned a concept with fewer practice problems, using it would reduce the number of unnecessary problems solved and time spent by the student with our tutors.

For the current study we used legacy data collected by tutors on `while` loops and `for` loops from multiple institutions as shown in Table 1. In the table, multi-problem records are the records of students who solved more than one problem on a concept. Each tutor covers one topic, and each topic consists of multiple concepts. The tutors use pretest-practice-posttest protocol during every tutoring session (Kumar 2014).

Table 1. Statistics about the data collected by programming tutors.

Topic	Number of concepts	Number of semesters	Total records	Multi-problem records
while loops	9	9	4,933	2,030
for loops	10	9	40,124	5,817

The tutors presented only code-tracing problems wherein students were asked to identify the output of a given program. The student grade on each problem was normalized to $0 \rightarrow 1.0$: 0 when the answer was incorrect, 1.0 when it was correct and a value in between for partially correct answers. The tutors logged the grade and time spent on each problem by each student.

Bayesian Knowledge Tracing (Baker et al. 2008) uses four parameters: L_i , T , G , S . We calculated $P(L_0)$, the probability that a concept was mastered before using the tutor as the percentage of the users who solved the pretest problem on the concept correctly (among Total Records in Table 1). $P(L_0)$ was 0.80 or greater on 32% of the concepts across both tutors. Given the high values of $P(L_0)$, we used 0.98 instead of the traditional 0.95 as the mastery criterion for the BKT model. We computed $P(T)$, the probability of transferring from un-mastered to mastered state for a given concept as the percentage of students who solved the pretest problem on the concept incorrectly, and went on to solve the post-test problem correctly. These were the students who learned the concept by using the tutor.

We computed $P(G)$, the probability a student guesses the correct answer to a practice problem on an un-mastered concept (from Multi-problem Records in Table 1) as the percentage of students who solved the previous problem on the concept incorrectly or partially, but solved the current problem correctly. Similarly, we computed $P(S)$, the probability a student slips, i.e., solves a practice problem on a mastered concept incorrectly or partially as the percentage of students who solved the previous problem on the concept correctly, but solved the current problem incorrectly or partially. For the first practice problem, we approximated this to be 0.01 since the tutors never presented a practice problem unless the pretest problem was solved incorrectly.

Figure 1 illustrates the BKT model for a concept. Several attempts have been made to individualize BKT parameters per student with the aim of improving its fit (Bhatt et al. 2020). *Our approach is different in that we have tried to customize performance*

parameters G and S to the problems solved by the students because no two problems are alike in terms of the provided context or the expected answer.

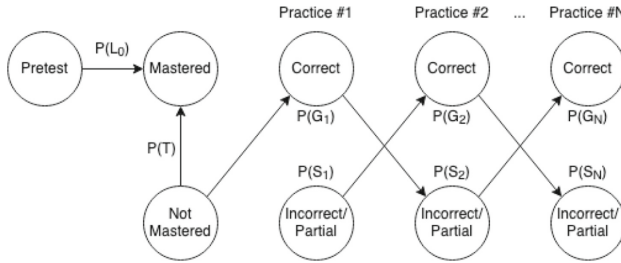


Fig. 1. BKT Model with two parameters (L,T) per concept and two parameters (G,S) per practice problem.

2 Evaluating the BKT Model

We used k -fold cross-validation to estimate the performance of our predictive BKT model: We used each of the k subgroups to find the number of students who would have saved time, made no difference, or lost time with the BKT model constructed using the other $k - 1$ groups. We used 25 as the size of each group and rounded up our sample size to the nearest multiple of 25 using stochastic oversampling. After cross-validation

Table 2. Results of evaluating BKT model on while loop data

Concept	Mean # of students who			Mean time saved (in Minutes)	% of total	Mean # of problems saved	% of total	Across k runs
	Saved time	Made no difference	Lost time					
1	16.60	6.60	1.80	0.96	24	1.13	21.8	10
2	16.64	4.73	3.64	1.27	31.75	1.08	15.25	11
3	19.50	5.50	0.00	0.41	13.67	0.86	25.93	2
4	20.71	3.57	0.71	0.90	30	1.31	31.31	7
5	19.65	4.47	0.88	1.35	33.75	1.91	44.59	17
6	17.64	5.14	2.21	1.92	48	1.58	26.63	14
7	19.33	5.33	0.33	0.65	21.67	1.33	42.45	3
8	14.00	5.50	5.50	2.28	57	1.04	11.82	12
9	14.11	7.89	3.00	2.05	51.25	1.09	11.64	9
Weighted mean	17.26	5.35	2.39	1.51	38.62	1.35	25.43	Total of 85

runs, we computed the mean of the time and practice problems saved per student across all the cross-validation runs.

The tutor on `while` loops covered 9 concepts. Table 2 lists the results for `while` loop tutor. Note that most students would have saved time with the BKT model on all the concepts. Concepts 8 and 9 are on nested loops and take longer to solve: those are the concepts on which students would have saved the most time with the BKT model.

The tutor on `for` loops covered 10 concepts. Table 3 lists the results for `for` loop tutor. Concepts 5 and 10 are minor variations of a regular loop – these were also the concepts on which nearly as many students had no difference as saved time with the BKT model. Concept 2 is on tracing the behavior of two loops, the second loop’s iterations dependent on the first. It takes longer to solve. Students saved the most time and problems on this concept.

Table 3. Results of evaluating BKT model on `for` loop data

Concept	Mean # of Students who			Mean time saved (in Minutes)	% of total	Mean # of problems saved	% of total	Across k runs
	Saved time	Made no difference	Lost time					
				(Per Student)				
1	24.08	0.92	0.00	0.90	30	1.52	27.22	13
2	16.03	6.31	2.67	2.12	53	1.78	27.53	36
3	13.00	7.13	4.88	0.26	8.67	0.62	20.9	8
4	12.13	8.10	4.77	1.48	37	1.07	15.93	31
5	10.00	9.59	5.41	0.62	20.67	0.54	8.22	17
6	24.41	0.59	0.00	1.03	34.33	1.81	50.99	37
7	12.88	6.42	5.67	1.68	42	0.93	11.8	43
8	13.80	9.80	1.40	0.35	11.67	0.92	30.67	10
9	14.20	7.97	2.83	0.63	21	0.95	18.51	30
10	12.50	11.14	1.36	0.52	17.33	0.67	16.14	14
Weighted mean	15.63	6.28	3.08	1.20	33.19	1.19	23.55	Total of 239

We found that, on average across the two tutors, students would have saved time with the BKT model on 64.23% of the concepts. This is similar to the results of another study that recently found that using BKT models saved time (Bhatt et al. 2020), although unlike them, our results were based on the use of legacy data. Students would have saved time/practice problems on some concepts more than others. The pattern that emerged is that students saved more time with BKT model on harder concepts on which it took longer to solve problems. When students neither saved nor lost time with BKT model compared to naïve mastery model, it was on simpler concepts. So, BKT model was found to be more beneficial for harder concepts than easier concepts.

For this study, we did not consider the relationships among the various concepts, i.e., we treated all the concepts as being independent and mutually exclusive. This is a fallible assumption in programming domain. In the future, we plan to use a Bayesian network to account for the relationships among these concepts.

Acknowledgements. Partial support for this work was provided by the National Science Foundation under grant DUE-1432190.

References

- Baker, R.S.J., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian Knowledge Tracing. In: Proceedings of the 9th International Conference on Intelligent Tutoring Systems, Berlin, Germany. pp. 406–415 (2008)
- Beck, J.: Difficulties in inferring student knowledge from observations (and why you should care). In: Educational Data Mining: Supplementary Proceedings of the 13th International Conference of Artificial Intelligence in Education, pp. 21–30 (2007)
- Beck, J.E., Chang, K.-M.: Identifiability: a fundamental problem of student modeling. In: Proceedings of the 11th International Conference on User Modeling (UM 2007), pp. 137–146 (2007)
- Bhatt, S., Zhao, J., Thille, C., Zimmaro, D., Gattani, N.: Evaluating Bayesian knowledge tracing for estimating learner proficiency and guiding learner behavior. In: L@S'20: Proceedings of the Seventh ACM Conference on Learning @ Scale, pp. 357–260. Association for Computing Machinery, New York (2020)
- Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adap. Inter.* **4**, 253–278 (1992)
- Hawkins, W.J., Heffernan, N.T., Baker, R.S.J.D.: Learning Bayesian knowledge tracing parameters with a knowledge heuristic and empirical probabilities. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 150–155. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07221-0_18
- Kumar, A.N.: A model for deploying software tutors. In: IEEE 6th International Conference on Technology for Education (T4E), Amritapuri, India, pp. 3–9 (2014)



Exploring Bayesian Deep Learning for Urgent Instructor Intervention Need in MOOC Forums

Jialin Yu^(✉), Laila Alrajhi, Anoushka Harit, Zhongtian Sun,
Alexandra I. Cristea, and Lei Shi

Department of Computer Science, Durham University, Durham, UK
{jialin.yu,laila.m.alrajhi,anoushka.harit,zhongtian.sun,
alexandra.i.cristea,lei.shi}@durham.ac.uk

Abstract. Massive Open Online Courses (MOOCs) have become a popular choice for e-learning thanks to their great flexibility. However, due to large numbers of learners and their diverse backgrounds, it is taxing to offer real-time support. Learners may post their feelings of confusion and struggle in the respective MOOC forums, but with the large volume of posts and high workloads for MOOC instructors, it is unlikely that the instructors can identify all learners requiring intervention. This problem has been studied as a Natural Language Processing (NLP) problem recently, and is known to be challenging, due to the imbalance of the data and the complex nature of the task. In this paper, we explore for the first time Bayesian deep learning on learner-based text posts with two methods: Monte Carlo Dropout and Variational Inference, as a new solution to assessing the need of instructor interventions for a learner's post. We compare models based on our proposed methods with probabilistic modelling to its baseline non-Bayesian models under similar circumstances, for different cases of applying prediction. The results suggest that Bayesian deep learning offers a critical uncertainty measure that is not supplied by traditional neural networks. This adds more explainability, trust and robustness to AI, which is crucial in education-based applications. Additionally, it can achieve similar or better performance compared to non-probabilistic neural networks, as well as grant lower variance.

Keywords: Deep learning · Artificial intelligence in education · Educational data mining · Bayesian modelling · Urgent instructor intervention · Natural language processing

1 Introduction

MOOCs are well-known for their high dropout rates [2,3]. Whilst learners may discuss their problems in the forums before actually dropping out, the sheer volume of posts renders it almost impossible for instructors to address them. Thus,

many of these urgent posts are overlooked or discarded. Hence, a few researchers proposed [4,9] automated machine learning models for need prediction based on learners' posts in MOOC forums. Such an approach would allow instructors to identify learners who require urgent intervention, in order to, ultimately, prevent potential dropouts (see our recent research, where we have shown only 13% of learners passing urgent intervention messages complete the course [27]).

More recently, techniques for applying deep neural networks to interpret texts from the educational field have emerged [17], including identifying learners' needs based on their posts in forums [5,16,36]. Despite their success, standard deep learning models have limited capability to incorporate uncertainty. One other challenge is that post data is notoriously imbalanced, with urgent posts representing a very low percentage of the overall body of posts - the proverbial 'needle in the haystack'. This tends to make a neural network overfit and ignore the urgent posts, resulting in large variance in model predictions.

To address the above two challenges, we apply Bayesian probabilistic modelling to standard neural networks. Recent advances in Bayesian deep learning offer a new theory-grounded methodology to apply probabilistic modelling using neural networks. This important approach is yet to be introduced in the Learning Analytics (LA) field.

Thus, the main contributions of this work are:

1. We present the first research on how Bayesian deep learning can be applied to text-based LA. Here, the aim is to predict instructor intervention need in the educational domain.
2. Hence, we explore, for the first time, not only one, but two Bayesian deep learning methods, on the task of classifying learners' posts based on their urgency, namely Monte Carlo Dropout and Variational Inference.
3. We show empirically the benefits of Bayesian deep learning for this task and we discuss the differences in our two Bayesian approaches.
4. We achieve competitive results in the task and obtain a lower variance when training with small size data samples.
5. We apply this approach to text-based processing on posts in MOOCs - a source generally available across all MOOC providers. Thus our approach is widely applicable - generalisable to foresee instructor's intervention need in MOOCs in general and to support the elusive problem of MOOC dropout.

2 Related Work

2.1 Urgent Intervention Need in MOOCs

Detection of the need for urgent instructor intervention is arguably one of the most important challenges in MOOC environments. The problem was first proposed and tackled [10] as a binary prediction task on instructor's intervention histories based on statistical machine learning. A follow-up study [9] proposed the use of $L1$ regularisation techniques during the training and used an additional feature about the type of forum (thread), besides the linguistic features

of posts. Another study [4] tried to build a generalised model, using different shallow ML models with linguistic features extracted by NLP tools, metadata and term frequency. In general, this problem was attempted based on two types of data format: text-only [6, 11, 16, 36, 39] or a mixture of text and post features [4, 9]. From a machine learning perspective, both traditional machine learning methods [4, 6, 9] and deep learning based methods [11, 16, 36, 39] were proposed and explored; with more recent studies being in favour of deep neural network-based approaches [17]. However, one critical problem for deep neural networks is that they do not offer a robust estimation over the prediction values. Also, we can not perform efficient learning on small sample size data. Thus, in this paper, we explore the benefits of Bayesian deep learning to predict learners who require urgent interventions from an instructor. We use text only features in our study, as it is the first study to explore the benefits of this new approach, and we leave future optimisation for further work.

To the best of our knowledge, this is the first study of Bayesian deep learning methods for learners' urgent intervention need classification. Our research sheds light on a new direction for other researchers in the fields of Educational Data Mining (EDM) and Learning Analytics (LA).

2.2 Bayesian Neural Networks

Modern neural networks (NNs) are self-adaptive models with a learnable parameter set W . In a supervised learning setting, given data $D = (x_i, y_i)_{i=1}^N$, we aim to learn a function through the neural network $y = f_{\text{NN}}(x)$ that maps the inputs x to y . A point estimation of the model parameter set W^* is obtained through a gradient based optimisation technique and with a respective cost function.

Bayesian neural networks (BNNs) [29, 30, 32], alternatively, consider the probability of the distribution over the parameter set W and introduce a prior over the neural network parameter set $P(W)$. The posterior probability distribution $P(W | D)$ is learnt in a data-driven fashion through Bayesian inference. This grants us a distribution over the parameter set W other than a static point estimation, which allows us to model uncertainty in the neural network prediction. In the prediction phase, we sample model parameters from the posterior distribution i.e. $w \sim P(W | D)$ and predict results with $f_{\text{NN}}^w(x)$ for the corresponding y . We marginalise the w samples and obtain an expected prediction. Due to the complexity and non-linearity of neural networks, an exact inference for BNNs is rarely possible, hence various approximation inference methods have been developed [12, 15, 18, 19]. The most widely adopted approximation method is the Monte Carlo Dropout [12], with applications in natural language processing, data analytics and computer vision [13, 22, 23, 40, 42]. In this paper, we adopt the same idea and use Monte Carlo Dropout [12] to approximate the neural network as a BNN.

2.3 Variational Inference

Variational inference (VI) [7, 21, 38] is a general framework for Bayesian statistical modelling and inference under a maximum likelihood learning scheme. It introduces an unobserved random variable as the generative component to model the probabilistic uncertainty. Given fully observed data $D = (x_i, y_i)_{i=1}^N$, we consider them as random variables and use capital letters X and Y to represent them. The unobserved random variable introduced with VI is denoted as Z and passes the information from X to Y . It can be marginalised out with Bayes' rule as:

$$P(X, Y; \theta) = \sum_Z P(X, Y, Z; \theta) = P(Y | Z; \theta)P(Z | X; \theta) \quad (1)$$

Under a mean-field assumption [37] over the unobserved random variable Z , we can factorise it as:

$$P(Z; \theta) = P(z_1, \dots, z_N; \theta) = \prod_{i=1}^N P(z_i; \theta) \quad (2)$$

Hence for each pair of data x and y , the maximum likelihood learning delivers the following objective with respect to θ :

$$\log P(y|x; \theta) = \log \int_z P(y|z; \theta)P(z|x; \theta)dz \quad (3)$$

Given observed data $D = (x_i, y_i)_{i=1}^N$, we can not directly model the distribution of unobserved z and hence the probability distribution $P(y|z; \theta)$ is intractable for data-driven models, such as neural networks. With VI, an additional variational parameter ϕ with its associated variational family distribution $q(z; \phi)$ is introduced, to approximate the real probability $P(y|z; \theta)$. During the learning process, we minimise the distance between $q(z; \phi)$ and $P(y|z; \theta)$ through the Kullback–Leibler divergence, a term that measures the distance between two probability distributions. Hence, the learning of the intractable probability distribution problem is converted to an optimisation problem over the evidence lower bound (ELBO), where \mathbb{D}_{KL} refers to the Kullback–Leibler divergence:

$$\log P(y|x; \theta) \geq \mathcal{L}(ELBO) = \mathbb{E}_{q(z; \phi)}[\log P(y|z; \theta)] - \mathbb{D}_{KL}[q(z; \phi) || p(z|x; \theta)] \quad (4)$$

VI was initially developed to solve a specific class of modelling problems where conditional conjugacy is presumed, and variational parameter ϕ is updated through closed-form coordinate ascent [14]. However, conditional conjugacy is not practical in most of the real world problems; thus further advancements [8, 20, 25, 34, 41] extend VI to large scale datasets and non-conjugate models.

3 Methodology

In this section, we first introduce the baseline model built based on recurrent neural networks (RNNs) and an attention mechanism. Then we present our two approaches for applying Bayesian deep learning with our baseline model: 1) Monte Carlo Dropout and 2) Variational Inference.

3.1 Baseline Deep Learning Model

In this section, we first introduce our non-Bayesian model, which serves as our baseline model. The model consists of three different components: an embedding layer, a two-layer recurrent neural network (RNN), and a prediction layer. We use attention based on the output of the RNNs to create a contextual representation over the RNN hidden outputs and then concatenate it with the last layer RNN outputs. The model architecture is presented in Fig. 1.

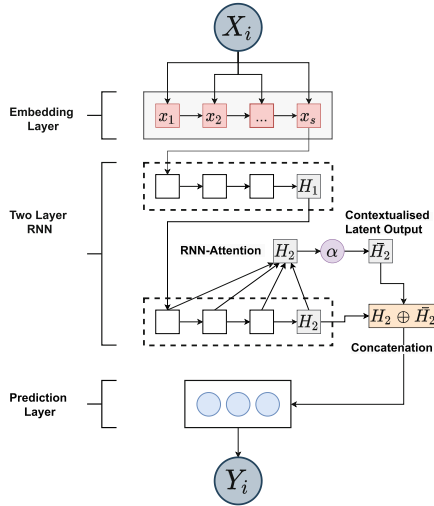


Fig. 1. Model architecture for baseline model (operation \oplus refers to the concatenation).

Given the data $D = (x_i, y_i)_{i=1}^N$, where each sentence x_i consists of a sequence of tokens $x_i^1, x_i^2, \dots, x_i^s$ where s denotes the sequence length. For our baseline model, given a sentence x_i , we first pass it through the embedding layer and obtain a sequence of word embeddings:

$$E = (\text{emb}(x_i^1), \text{emb}(x_i^2), \dots, \text{emb}(x_i^s)) \quad (5)$$

where emb is the embedding function we used for our experiment with d dimensions. Here, x_i^m denotes the m^{th} word in the sentence x_i . For the initial sentence $x_i \in \mathbb{R}^{s \times 1}$, we derive a sentence $x_i \in \mathbb{R}^{s \times d}$ after the embedding layer. Then we feed this as a sequence input through a two-layer long-short-term memory (LSTM) model as in [36]. The initial hidden state h_0 is set to 0 and we calculate the sequence of hidden states as:

$$h_m = \text{LSTM}(h_{m-1}, x_i^m) \quad (6)$$

where we have $m = 1, \dots, s$. The last layer of hidden states provides a sequence output $H \in \mathbb{R}^{s \times h}$, where h here represents the hidden dimension size. In order

to utilise the contextual information through the LSTM encoding process, we calculate the attention score α based on the last hidden state outputs H_2 ($H_2 = h_s$) and each hidden state in the sequence of H , as:

$$\alpha_m = \frac{H_2 * h_m}{\sum_{m=1}^s H_2 * h_m} \quad (7)$$

Then we calculate the contextual \bar{H}_2 as:

$$\bar{H}_2 = \sum_{m=1}^s \alpha_m h_m \quad (8)$$

Finally, we concatenate them and feed them through a fully connected layer with the output dimension equal to the number of classes for our task. This fully connected layer is represented as a prediction layer in Fig. 1.

3.2 Model Uncertainty with Monte Carlo Dropout

In this section, we present how to convert our baseline model into a Bayesian neural network. With Monte Carlo Dropout [12], we only need to use the dropout technique [35] before each layers containing the parameter set W . In our case, we add a dropout layer after the first and second LSTM layers, as well as after the fully connected layer, which takes the input as the concatenation of \bar{h}_2 and h_2 .

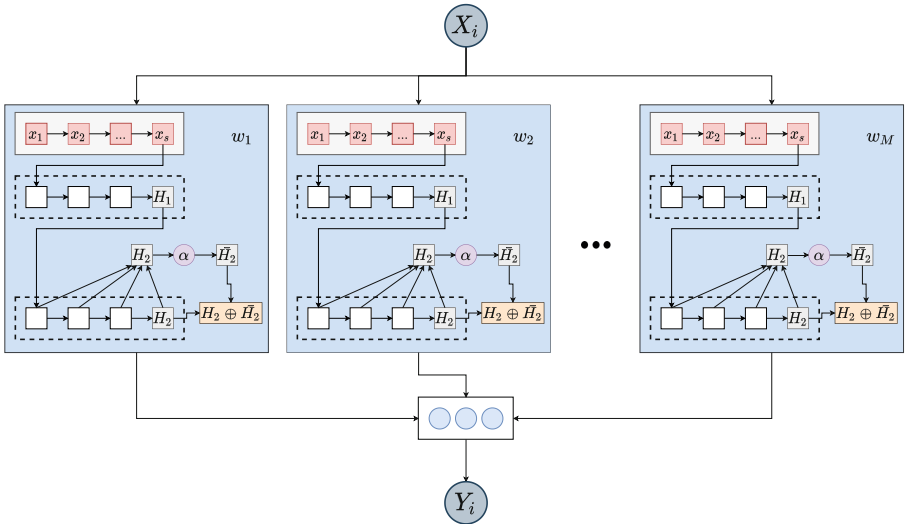


Fig. 2. A demonstration of the Monte Carlo Dropout in the test phase. We run the model for M times for M different prediction results and then calculate their average as the prediction layer output.

Compared with the standard dropout technique, which works as a regularisation technique in the training phase only, the Monte Carlo Dropout technique requires the dropout layer to be activated in both training and testing phases. This allows the standard neural network model to work as a BNN [12]. Each dropout works as a sample of w from its probabilistic distribution space and hence allows us to measure the uncertainty of the model, as shown in Fig. 2. In the testing phase, we predict the output through sampling M times [12] and the expectation of y can be calculated as:

$$\mathbb{E}(y | x) \approx \frac{1}{M} \sum_{i=1}^M f_{\text{NN}}^{w_i}(x) \quad (9)$$

We use this expectation as the final logits value and in our experiment we use a total sample M of 50 as in [23].

3.3 Model Uncertainty with Variational Inference

As discussed in the Sect. 2.3, Variational Inference (VI) introduces an additional random variable z with probability distribution $q(z; \phi)$ to the original model. This variational family $q(z; \phi)$ here approximates the posterior distribution $P(z | x; \theta)$ as $q(z|x, y; \phi)$. The model architecture is presented in Fig. 3. Following [31], we define $q_\phi(z|x, y)$ as:

$$q(z|x, y; \phi) = \mathcal{N}(z | \mu_\phi(x, y), \text{diag}(\sigma_\phi^2(x, y))) \quad (10)$$

we have:

$$\mu_\phi(x, y) = l_1(\pi_\phi) \quad (11)$$

and:

$$\log \sigma_\phi(x, y) = l_2(\pi_\phi) \quad (12)$$

where:

$$\pi_\phi = g_\phi(H_2, f_y(y)) \quad (13)$$

where $f_y(y)$ is an affine transformation from output $y \in \mathbb{R}^1$ to a vector space size $s_y \in \mathbb{R}^z$. The H_2 is the final latent state output of the second LSTM network layer as stated in Sect. 3.1. The latent variable $z \in \mathbb{R}^h$ can be reparameterised as $z = \mu + \sigma \cdot \epsilon$, known as the ‘‘reparameterisation trick’’ [26] with sample $\epsilon \sim \mathcal{N}(0, \mathbb{I})$. For the conditional distribution $p_\theta(z|x)$, we can model it as:

$$p(z|x; \theta) = \mathcal{N}(z | \mu_\theta(x), \text{diag}(\sigma_\theta^2(x))) \quad (14)$$

where we have:

$$\mu_\theta(x) = l_3(\pi_\theta) \quad (15)$$

$$\log \sigma_\theta(x) = l_4(\pi_\theta) \quad (16)$$

and:

$$\pi_\theta = g_\theta(H_2) \quad (17)$$

where l_1, l_2, l_3 and l_4 are four affine transformation functions. Since both $p(z|x; \theta)$ and $q(z|x, y; \phi)$ are multivariate Gaussian distributions, this allows us to have a closed-form solution for the Kullback–Leibler (KL) divergence term [25]. For the reconstruction term $\log p(y | z; \theta)$ with Monte Carlo approximation [31], the final reconstruction loss can be calculated as:

$$\mathbb{E}_{q(z)}[\log p(y|z; \theta)] \approx \frac{1}{M} \sum_{m=1}^M \log p(y|z_m \oplus H_2 \oplus \bar{H}_2; \theta) \quad (18)$$

where \oplus denotes the concatenation operation and M is the number of samples from the posterior distribution z . We use a single sample of $M = 1$ during training based on [24] and $M = 20$ during testing based on [31]. In the training phase, z is sampled from $q(z|x, y; \phi)$ and in the test phase, from $p(z|x; \theta)$.

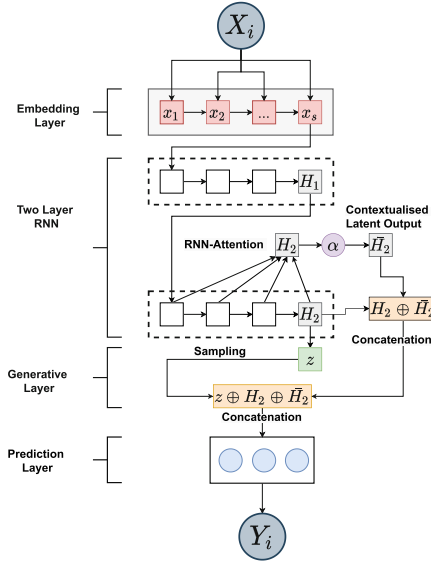


Fig. 3. Model architecture for the VI model.

4 Experiments

4.1 Dataset

Here, we used the benchmark posts dataset from the Stanford MOOC forum [1], containing 29604 anonymised posts collected from 11 different courses. Each post is manually labelled by three independent human experts and with agreements for the gold label. Apart from the text content, each post is evaluated based on six categories, amongst which urgency, which is the one we used here. Its range is

1 to 7; with 1 meaning no reason to read the post and 7 meaning extremely urgent for instructor interventions. An example urgent message is “I hope any course staff member can help us to solve this confusion asap!!!”; whilst a non-urgent would be “Good luck to everyone”. See more details on their website¹. Similar to [16], we convert the problem of detecting urgent posts to a binary classification task. A threshold of 4 is used as in [16] to create two need categories as: 1) *Need for urgent intervention* ($value > 4$) with label 1; and 2) *No need for intervention* ($values \leq 4$) with label 0. This allows us to obtain a total of 29,597 posts, with 23,991 labelled as 0 and 5,606 labelled as 1. We tokenise the text and create a vocabulary based on a frequency-based cutoff [28] and use the special token $\langle pad \rangle$ for padding and the unknown token $\langle unk \rangle$ for out-of-vocabulary words. We initialise the embedding layer with a 300-dimensional GloVe vector [33] if found in the pre-trained token list.

4.2 Experiment Setup and Evaluation

In this paper we have implemented 3 different models: a baseline model (*Base*), as shown in Fig. 1; a baseline model converted to a Bayesian neural network through Monte Carlo Dropout (*MCD*), as shown in Fig. 2; and a baseline model with variational inference (*VI*), as shown in Fig. 3. For the evaluation, we report mean accuracy; F1 score, Precision score, Recall score for all three models under each class (the higher the better); and entropy based on the prediction layer [23, 40] (the lower the better).

We conduct two sets of experiments. For the first set, we follow the setup in [5]. At each run of the experiment, we randomly split this data into training and testing sets each with a ratio of 80% and 20%, respectively, with stratified sampling on a random state. In the second set of experiments, we use less training examples, since the intervention case is rare compared with non-intervention, and we compare the robustness of our model given smaller size samples and we use a split of 40%, 60% for training and testing. The results for the two experiments are reported in Table 1 and Table 2, respectively, and we run both experiments 10 times. In Table 1, we report the best run of the model and in Table 2, we report the mean and variance. All the evaluation metrics results reported here in this paper are based on test dataset only. In the first table, we use bold text to denote the results that outperform results in [5] and in the second table, we use bold to denote results outperforming the (*Base*) model.

5 Results and Discussions

The results are presented in Table 1. The baseline model (*Base*) performs competitively against a strong model [5], especially in the recall and F1 score for the ‘urgent’ class and the precision score for the ‘non-urgent’ class. For the Monte Carlo Dropout (*MCD*) and Variational Inference (*VI*) models, we achieve better

¹ <https://datastage.stanford.edu/StanfordMoocPosts/>.

Table 1. Results compare baseline model and Bayesian deep learning approach in accuracy, precision, recall and F1 score.

			Non-urgent (0)			Urgent (1)		
	Accuracy	Entropy	Precision	Recall	F1	Precision	Recall	F1
Text [5]	.878	–	.90	.95	.93	.73	.56	.64
Base	.883	.095	.937	.918	.927	.677	.738	.697
MCD	.883	.085	.939	.915	.926	.675	.742	.698
VI	.873	.103	.940	.901	.919	.644	.752	.687

Table 2. Results compare mean and variance of Deep learning and Bayesian deep learning approach based on 10 runs.

			Non-urgent (0)			Urgent (1)		
	Accuracy	Entropy	Precision	Recall	F1	Precision	Recall	F1
Base	.870+-0.039	.1126+-0.041	.930+-0.039	.908+-0.088	.918+-0.030	.645+-0.159	.707+-0.215	.664+-0.052
MCD	.869+-0.013	0.101+-0.026	.929+-0.028	.908+-0.0319	.917+-0.0104	.652+-0.0574	.703+-0.0693	.660+-0.0042
VI	.867+-0.019	0.078+-0.0296	.924+-0.0028	.910+-0.0058	.916+-0.0017	.642+-0.0093	.680+-0.0164	.649+-0.0034

performance in these measurements against the baseline model (Base). Importantly, as an indication of the uncertainty measurement, we note that the entropy dropped for the MCD model. In Table 2, we can see that Bayesian deep learning methods generally achieve similar or better performance compared to the non-Bayesian base model, but hold lower variance and lower entropy against small sample size data. This is often the case in real life scenarios, where the label ‘need intervention’ is scarce. A probabilistic approach works as a natural regularisation technique, when neural network models are generally over-parameterised. We can conclude that Bayesian deep learning mitigates this issue of over-parametrisation with lower variance and entropy. This is especially clear for the VI methods. The result from a Wilcoxon test shows that, compared with the Base model, the experiment results of the VI model are statistically significant at the .05 level, with $p=.022$ for the entropy value and with $p=.007$ for the recall, in the ‘urgent’ case. Comparing MCD and VI models, the latter achieves better performance in most metrics, as shown in both tables, especially with a higher recall score. The recall score is preferable to precision in this task, where we have a comparatively small number of positive examples. However, the implementation of MCD models is more accessible to researchers interested in introducing uncertainty into their neural networks. This should be considered in using them in practice.

6 Conclusion

Identifying the need of learner interventions for instructors is an extremely important issue in MOOC environments. In this paper, we have explored the benefits of a Bayesian deep learning approach to this problem for the first time. We have implemented two different approaches to Bayesian deep learning, namely Monte Carlo dropout and variational inference. Both offer a critical probabilistic measurement in neural networks. We have demonstrated the effectiveness

of both approaches in decreasing the epistemic uncertainty of the original neural network and granting equivalent or even better performance. We have thus provided guidelines for researchers interested in building safer, more statistically sound neural network-based models in the Learning Analytics (LA) field. Entropy measures a classifiers' confidence level. In intelligent tutoring systems, high confidence (thus low entropy) is essential. With Bayesian deep learning, we turn NN models into probabilistic models, allowing more explainability and trust. For future research, these can be extended and applied in more areas.

References

1. Agrawal, A., Venkatraman, J., Leonard, S., Paepcke, A.: Youedu: addressing confusion in MOOC discussion forums by recommending instructional video clips (2015)
2. Alamri, A., et al.: Predicting MOOCs dropout using only two easily obtainable features from the first week's activities. In: Coy, A., Hayashi, Y., Chang, M. (eds.) ITS 2019. LNCS, vol. 11528, pp. 163–173. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22244-4_20
3. Alamri, A., Sun, Z., Cristea, A.I., Senthilnathan, G., Shi, L., Stewart, C.: Is MOOC learning different for dropouts? a visually-driven, multi-granularity explanatory ML approach. In: Kumar, V., Troussas, C. (eds.) ITS 2020. LNCS, vol. 12149, pp. 353–363. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_42
4. Almatrafi, O., Johri, A., Rangwala, H.: Needle in a haystack: Identifying learner posts that require urgent response in mooc discussion forums. *Comput. Educ.* **118**, 1–9 (2018)
5. Alrajhi, L., Alharbi, K., Cristea, A.I.: A multidimensional deep learner model of urgent instructor intervention need in MOOC forum posts. In: Kumar, V., Troussas, C. (eds.) ITS 2020. LNCS, vol. 12149, pp. 226–236. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_27
6. Bakharia, A.: Towards cross-domain mooc forum post classification. In: Proceedings of the Third (2016) ACM Conference on Learning@ Scale, pp. 253–256 (2016)
7. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, New York (2006)
8. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* **112**(518), 859–877 (2017)
9. Chandrasekaran, M.K., Kan, M.Y., Tan, B.C., Ragupathi, K.: Learning instructor intervention from mooc forums: Early results and issues. arXiv preprint [arXiv:1504.07206](https://arxiv.org/abs/1504.07206) (2015)
10. Chaturvedi, S., Goldwasser, D., Daumé III, H.: Predicting instructor's intervention in MOOC forums. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (vol. 1, Long Papers), pp. 1501–1511 (2014)
11. Clavié, B., Gal, K.: Edubert: Pretrained deep language models for learning analytics. arXiv preprint [arXiv:1912.00690](https://arxiv.org/abs/1912.00690) (2019)
12. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp. 1050–1059 (2016)
13. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 1019–1027 (2016)

14. Ghahramani, Z., Beal, M.: Propagation algorithms for variational bayesian learning. *Adv. Neural Inf. Process. Syst.* **13**, 507–513 (2000)
15. Graves, A.: Practical variational inference for neural networks. *Adv. Neural Inf. Process. Syst.* **24**, 2348–2356 (2011)
16. Guo, S.X., Sun, X., Wang, S.X., Gao, Y., Feng, J.: Attention-based character-word hybrid neural networks with semantic and structural information for identifying of urgent posts in mooc discussion forums. *IEEE Access* **7**, 120522–120532 (2019)
17. Hernández-Blanco, A., Herrera-Flores, B., Tomás, D., Navarro-Colorado, B.: A systematic review of deep learning approaches to educational data mining. *Complexity* **2019**, (2019)
18. Hernández-Lobato, J.M., Adams, R.: Probabilistic backpropagation for scalable learning of bayesian neural networks. In: *International Conference on Machine Learning*, pp. 1861–1869 (2015)
19. Hernández-Lobato, J.M., Gelbart, M., Hoffman, M., Adams, R., Ghahramani, Z.: Predictive entropy search for bayesian optimization with unknown constraints. In: *International Conference on Machine Learning*, pp. 1699–1707. PMLR (2015)
20. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. *J. Mach. Learn. Res.* **14**(1), 1303–1347 (2013)
21. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. *Mach. Learn.* **37**(2), 183–233 (1999)
22. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint [arXiv:1511.02680](https://arxiv.org/abs/1511.02680)* (2015)
23. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: *Advances in Neural Information Processing Systems*, pp. 5574–5584 (2017)
24. Kim, Y., Wiseman, S., Rush, A.M.: A tutorial on deep latent variable models of natural language. *arXiv preprint [arXiv:1812.06834](https://arxiv.org/abs/1812.06834)* (2018)
25. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114)* (2013)
26. Kingma, D.P., Salimans, T., Welling, M.: Variational dropout and the local reparameterization trick. In: *Advances in Neural Information Processing Systems*, pp. 2575–2583 (2015)
27. Laila, A., Ahmed, A., Filipe, D.P., Alexandra, I.C.: Urgency analysis of learners' comments: an automated intervention priority model for mooc. Presented at the (2021)
28. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. *arXiv preprint [arXiv:1508.04025](https://arxiv.org/abs/1508.04025)* (2015)
29. MacKay, D.J.: A practical bayesian framework for backpropagation networks. *Neural Comput.* **4**(3), 448–472 (1992)
30. MacKay, D.J.: Probable networks and plausible predictions—a review of practical bayesian methods for supervised neural networks. *Netw. Comput. Neural Syst.* **6**(3), 469–505 (1995)
31. Miao, Y., Yu, L., Blunsom, P.: Neural variational inference for text processing. In: *International Conference on Machine Learning*, pp. 1727–1736 (2016)
32. Neal, R.M.: *Bayesian Learning for Neural Networks*, vol. 118. Springer, New York (2012) <https://doi.org/10.1007/978-1-4612-0745-0>
33. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014)

34. Ranganath, R., Gerrish, S., Blei, D.: Black box variational inference. In: *Artificial Intelligence and Statistics*, pp. 814–822. PMLR (2014)
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
36. Sun, X., Guo, S., Gao, Y., Zhang, J., Xiao, X., Feng, J.: Identification of urgent posts in mooc discussion forums using an improved RCNN. In: *2019 IEEE World Conference on Engineering Education (EDUNINE)*, pp. 1–5. IEEE (2019)
37. Tanaka, T.: A theory of mean field approximation. In: *Advances in Neural Information Processing Systems*, pp. 351–360 (1999)
38. Wainwright, M.J., Jordan, M.I.: *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc, Boston (2008)
39. Wei, X., Lin, H., Yang, L., Yu, Y.: A convolution-LSTM-based deep neural network for cross-domain mooc forum post classification. *Information* **8**(3), 92 (2017)
40. Xiao, Y., Wang, W.Y.: Quantifying uncertainties in natural language processing tasks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7322–7329 (2019)
41. Zhang, C., Bütepage, J., Kjellström, H., Mandt, S.: Advances in variational inference. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(8), 2008–2026 (2018)
42. Zhu, L., Laptev, N.: Deep and confident prediction for time series at Uber. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 103–110. IEEE (2017)

Concept Maps



Creating and Visualising Cognitive Maps of Knowledge Diagnosis During the Processing of Learning Digital Footprint

Viktor Uglev¹(✉)  and Oleg Sychev² 

¹ Siberian Federal University, Zheleznogorsk, Russia
uglev-v@yandex.ru

² Volgograd State Technical University, Volgograd, Russia

Abstract. The paper describes the problem of creating a single mapping of the learning situation during the processing of digital footprint and decision making in intelligent tutoring systems. We propose creating a Cognitive Map of Knowledge Diagnosis as a way of summarising data in the digital footprint. The elements of this cognitive map and the details of their visualisation are described. The results of the use of cognitive maps are demonstrated on the example of personalising the content of an online learning course and providing detailed feedback.

Keywords: Intelligent tutoring systems · Visualisation · Cognitive Maps of Knowledge Diagnosis · Learning digital footprint · Decision making

1 Introduction

The management of the learning process in intelligent tutoring systems (ITS) is based on the connection between the models of an e-learning course and its actors i.e. learners, teachers, and tutors. Machine Learning and Data Mining approaches allow good prediction performance for learner's progress [12], but they require vast amounts of data in learning digital footprint to train, are vulnerable to the cold-start problem with new learners [13], and have poor explainability [9]. These problems are even more important for low-throughput courses like specialised courses, master's-level, and post-graduate courses. In this work, we propose an approach to decision making for the intelligent planner component of ITS using Cognitive Maps of Knowledge Diagnosis (CMKD).

The models of a description of teaching materials can be divided into hierarchical and semantic models. Hierarchical models organise learning units by grouping them into topics, modules, and, eventually, courses. The model can be formalised as a tree, whose leaves must be presented to the learner sequentially.

A part of the reported study was supported by the Ministry of Science and of Higher Education the Russian Federation (research theme code FSRZ-2020-0011). A part of the reported study was funded by RFBR, project number 20-07-00764.

© Springer Nature Switzerland AG 2021

A. I. Cristea and C. Troussas (Eds.): ITS 2021, LNCS 12677, pp. 93–98, 2021.

https://doi.org/10.1007/978-3-030-80421-3_11

This defines the basic learning trajectory; the actual trajectory is more complex and non-linear for most students. This approach was first proposed by Skinner for programmed learning [14]; nowadays almost all Learning Management Systems (LMS) implement this model. But when combining this structure with the data from the learner’s digital footprint, the stream of events captured according to SCORM standards [11] or Tin Can API [8] doesn’t allow to capture the connections between the elements of domain knowledge; this results in lowering the value of the accumulated data and lowering the incentive for their processing.

The semantic approach involves creating a graph whose edges represent semantic dependencies between the subject-domain concepts that often involves developing ontologies or conceptual models of these domains. This allows analysing learner’s actions according to the structure of the domain reflected in the domain semantic links (ontology [3]). However, developing high-quality ontologies requires more time and effort than creating hierarchical course structure because ontologies are tied more to the subject domain than to the course’s learning goals (see [2]). Ontologies capture well the dependencies between domain concepts, but these don’t always coincide with learning-unit boundaries that makes the representation of course structure in ontologies problematic. While using lightweight [4] and heavyweight [15] ontologies as the basis of ITS allows generating personalised feedback for low-level tasks, the decision making on the strategic level often involves machine-learning approach [1], lacking the necessary data for it. This limits the scope of developed ITS’s and creates problems with the attempts to leverage them to the level of the learning-process management.

For agile learning-process management in ITS, only the dialectical unity of the two approaches allows preserving the completeness of connections between subject domain concepts and course’s learning units; this allows generalising information about the learner’s progress in digital footprint and making informed strategic decisions about learning trajectories.

2 Method

Combining the hierarchical structure of the learning units and their semantic dependencies is necessary for decision making in ITS, concerning the interaction of the learner, teacher, and tutor models [5]. These models use the same digital footprint as the basis for representation, evaluation, and analysis of the learning situation. To make this information understandable, we need a way to represent our knowledge about the course on an appropriate generalisation level so that we will keep course structure and semantic dependencies between its elements.

We propose enhancing the visual cognitive map (see [16]) by grounding it in the hierarchical and semantic representations of the course. Cognitive Map of Knowledge Diagnosis is a map that represents the information summarised by ITS (using metric concentration methods) in order to simplify the expert analysis of learning digital footprint, evaluate learning situation, and determine an efficient reaction to learner’s actions [17]. The basis of this map is the sequence of learning units (see Fig. 1), grouped by their topics and modules according to

the basic learning trajectory for the course as a small-world network [10]. The map contains a priori semantic causal dependencies, showing the prerequisites for learning units as the edges inside the circle. A learning unit is represented as a square with the unit number; it can be enhanced with the information from the digital footprint. Figure 1 shows that CMKD allows tracing incoming (from predecessors) and outgoing (from successors) semantic dependencies.

The basics features of course elements represented in CMKD are importance levels of learning units, aspects, and points of view. Learning units can be divided by their important level into the units belonging to the course core, local (module) core, miscellaneous (non-core), and reference (additional) material. Also, one learning unit can be implemented in several variants, depending on the unit representation, difficulty, or used method (tool); this results in creating a set of elements for one position that is shown as a stack of squares. Aspects mean different kinds of data from the digital footprint that is represented on the map after generalisation: structural and normative data, knowledge, competencies, etc. Any aspect can be shown in the initial, current, and target states. Point of view, in accordance with the approach [6], is a kind of model (a part of ITS), representing the learning situation: the models of learner, teacher and tutor may form different “opinions” (evaluation results) about the same situation. This allows modelling reflection in ITS and using methods of compromise solutions (according to Reflexive Management theory [7]).

The process of creating CMKD consists of several stages: the planner of ITS chooses a decision-making task and determines relevant aspect and point of view; data are extracted from the digital footprint and generalised for the map, according to the process described in [18]; visualising (if necessary) the invariant part of the map; applying the information from the footprint to the map according to the aspect and the point of view; enhancing map (if necessary) including adding interactive features; passing the map to the decision-making algorithm and (if necessary) creating maps for the other aspects.

CMKD can be used to solve the following tasks, requiring metric concentration of digital footprint: generating or revising a personalised online course and basic personalised learning trajectory; adaptive knowledge assessment, results analysis and hints generation; generating messages explaining ITS decisions; advising to the learner about using the online course; aggregating information for human teachers and tutors, allowing analysis of assessments’ results.

While CMKD is a part of the automatic decision-making process, its visualised form can be used by human teachers (for studying the ITS performance),

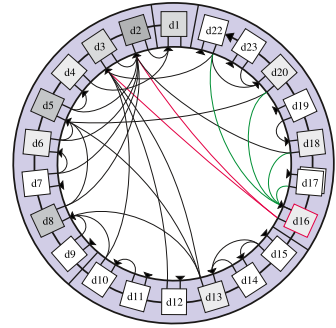


Fig. 1. CMKD for the online course “Decision-making theory” (structural aspect).

tutors (summarising the data about the course), and even to the learners (for understanding the system’s behaviour). The ability to visualise CMKD allows producing human-readable maps for every stage of learning, including producing a set of maps showing the progress of a learner or a group of learners.

3 Use Case

Our example is based on the course “Simulation Modelling” at the master’s level program “System analysis and management” in the experimental “EASU” ITS. The basic (a priori) course model as CMKD contains 19 learning units and semantic dependencies between them (see Fig. 2 a). Personalised course structure for two students who didn’t show much interest in the course is shown at Fig. 2 b, c. The learning units that are outside the circle were made optional. Colours represent current results of summative assessments in the knowledge aspect. The process of personalising the course content is described in detail in [18].

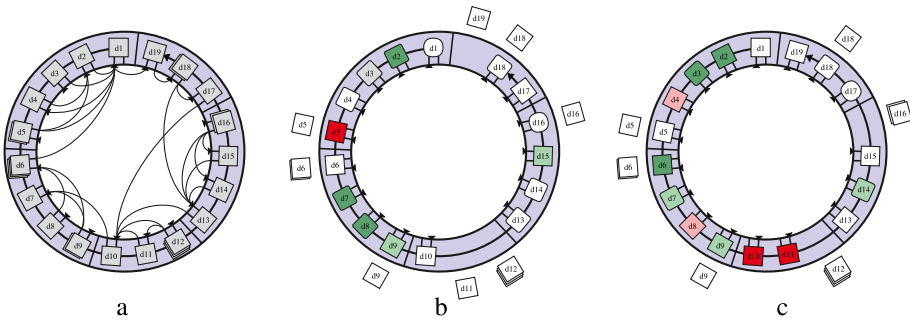


Fig. 2. Cognitive Map of Knowledge Diagnosis for the course “Simulation Modelling”: a) a priori map, structural aspect; b, c) personalised maps, knowledge aspect.

During personalising of the course’s content and learning trajectory, for 37.5% out of 28 students the composition and number of learning units was similar to the basic (a priori) course while for 12.6% of the students it was increased (up to 17.5%) and for 49.1% of the students it was decreased (up to 24%) along with personalising the degree of control of student’s resulting knowledge.

This example shows that if the course-level model contains semantic dependencies between learning units its possible to produce a decision tree, allowing to take into account both the regulations and personal learning goals using past experience, current achievements, and learning objectives. Creation and processing of CMKD allow personalised approach in planning components for ITS.

4 Conclusion

In this paper, we propose a model for generalising and visualising of learning digital footprint as a Cognitive Map of Knowledge Diagnosis that allows enhancing

planning in Intelligent Tutoring Systems. The maps can be used both for human-computer interaction and automatic decision making; they also can be exported to a student's portfolio.

References

1. Aissaoui, O., Oughdir, L.: A learning style-based ontology matching to enhance learning resources recommendation. In: 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), pp. 1–7 (2020). <https://doi.org/10.1109/IRASET48871.2020.9092142>
2. Brusilovsky, P., Rus, V.: Social navigation for self-improving intelligent educational systems, pp. 131–145. Army Research Laboratory (12 2019). <https://www.pitt.edu/~peterb/papers/SocNav4SIS.pdf>
3. Chanaa, A., El Faddouli, N.-E.: Predicting learners need for recommendation using dynamic graph-based knowledge tracing. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) AIED 2020. LNCS (LNAI), vol. 12164, pp. 49–53. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52240-7_9
4. Grévisse, C., Rothkugel, S.: An SKOS-based vocabulary on the swift programming language. In: Pan, J.Z., et al. (eds.) ISWC 2020. LNCS, vol. 12507, pp. 244–258. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62466-8_16
5. Karpenko, A., Dobryakov, A.: Model for automated training systems. overview. *Sci. Educ.* **7**, 1–63 (2011). <https://doi.org/10.7463/0715.0193116>
6. Kossiakoff, A., Sweet, W., Seymour, S., Biemer, S.: *Systems Engineering Principles and Practice*. Wiley-Interscience, Hoboken (2011)
7. Lefebvre, V.: *Lectures' about the Theory of Reflexive Games*. Cogito-Tsentr, Moscow (2009)
8. Lim, K.C.: Using the xAPI to track learning. In: Li, K.C., Yuen, K.S., Wong, B.T.M. (eds.) *Innovations in Open and Flexible Education*. EIS, pp. 233–242. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-7995-5_21
9. Lu, Yu., Wang, D., Meng, Q., Chen, P.: Towards interpretable deep learning models for knowledge tracing. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) AIED 2020. LNCS (LNAI), vol. 12164, pp. 185–190. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52240-7_34
10. Newman, M.: *Networks: An Introduction*. Oxford University Press Inc, USA (2010). <https://doi.org/10.5555/1809753>
11. Parmar, A.: Paper review on sharable content object reference model (scorm): Framework for e-learning standard. In: 2012 Second International Conference on Advanced Computing Communication Technologies, pp. 409–411 (2012). <https://doi.org/10.1109/ACCT.2012.95>
12. Piech, C., et al.: Deep knowledge tracing. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 28, pp. 505–513. Curran Associates, Inc. (2015). <https://stanford.edu/~cpiech/bio/papers/deepKnowledgeTracing.pdf>
13. Pliakos, K., Joo, S., Park, J., Cornillie, F., Vens, C., Van den Noortgate, W.: Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems. *Comput. Educ.* **137**, 91–103 (2019). <https://doi.org/10.1016/j.compedu.2019.04.009>
14. Skinner, B.: Teaching machines. *Science* **128**(3330), 969–977 (1958)

15. Sychev, O., Penskoy, N.: Ontology-based determining of evaluation order of C expressions and the fault reason for incorrect answers. In: Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 19th International Semantic Web Conference (ISWC 2020). CEUR Workshop Proceedings, vol. 2721, pp. 44–49. CEUR-WS.org (2020). <http://ceur-ws.org/Vol-2721/paper494.pdf>
16. Tolman, E.: Cognitive maps in rats and men. *Psychol. Rev.* **55**(4), 189–208 (1948)
17. Uglev, V.: Implementation of decision-making methods in intelligent automated educational system focused on complete individualization in learning. *AASRI Procedia* **6**, 66–72 (2014). <https://doi.org/10.1016/j.aasri.2014.05.010>
18. Uglev, V., Zakharin, K., Baryshev, R.: Cognitive maps of knowledge diagnosis as an element of a digital educational footprint and a copyright object. In: Silhavy, R., Silhavy, P., Prokopova, Z. (eds.) *CoMeSySo 2020. AISC*, vol. 1295, pp. 349–357. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63319-6_31



Integrating Knowledge in Collaborative Concept Mapping: Cases in an Online Class Setting

Junya Morita¹(✉), Yoshimasa Ohmoto¹, and Yugo Hayashi²

¹ Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu 432-8011, Japan
j-morita@inf.shizuoka.ac.jp

² Ristumeikan University, 2-150 Iwakura-cho, Ibaraki, Osaka 567-8570, Japan

Abstract. The rapid spread of online learning has increased demand for promoting and grasping collaborative learning processes. In this paper, we present a multi-channel process analysis of collaborative knowledge building, using a custom-made concept map tool and the application of conventional videoconferencing. The analysis focused on a process of copying and merging elements from individually created maps to a collaboration map. The sequential calculation of edit distance between maps revealed characteristics of group dynamics. The group who successfully integrated concept maps deepened their understanding of the topic, while the other group engaging shallow cooperation failed to build mutual understanding. The result shows the effectiveness of collaborative concept mapping in grasping online collaborative learning.

Keywords: Collaboration · Concept map · Online learning

1 Introduction

Due to the worldwide COVID-19 pandemic, online learning in higher education spread rapidly in 2020. Such a new learning environment ensures educational opportunities without geographical constraints, while it simultaneously reduces the chance for equal collaborative learning between learners. Online video conferencing tools, which are used frequently in such educational settings, usually concentrate on one specific speaker, namely the lecturer, hiding reactions from class attendees. These tools also restrict turn-taking behavior due to poor multi-modal conversational cues [6]. Therefore, there is widespread concern that online environments are detrimental to building a learning community in the educational environment [7].

The challenge is to build a collaborative environment in online education. One tradition solution in learning science is to build a concept map. Since classic cognitive science studies in the 1980s (e.g., [3]), many researchers have considered that knowledge in one's mind can be represented as nodes and edges networks and have built supporting tools for collaborative concept mapping [2].

The effect of such applications in the actual classroom have also been examined [13,14]. Dividing target phenomena into discrete elements located in a 2-D space is believed to uncover a knowledge structure in the internal mind of learners. Based on such visual representations, collaboration among learners is potentially promoted by finding commonalities or differences of understandings between participants, eventually leading to a deeper exploration on a learning topic.

This study follows assumption—introducing collaborative concept mapping in the online learning environment and examines cases of this process to find utilities and problems for future learning. In this study, we focus particularly on the analysis of edit distance between individual and collaborative maps.

2 Method

2.1 Participants and Target Class

This study targeted a class titled “Theories of Learning Process,” where participants aimed to acquire theories and technologies regarding the collaborative learning process. This class was an elective subject in a curriculum of the informatics department at a Japanese university. In the class, participants engaged in group work by reading research papers on collaborative learning and applying several techniques for analyzing the learning process, such as verbal-text analysis or multi-modal communication analysis. In the 2020 fall semester, the overall process was conducted in an online setting without face-to-face lectures.

Participants of the class were seven third-year undergraduate students. All were male and Japanese native speakers. Among them, we analyzed five participants whose target activities were successfully recorded. These participants belonged to two groups (two in G2, three in G3) and engaged in collaborative concept mapping to summarize papers related to the topic of the class.

2.2 Materials

Papers. The participants themselves selected papers from Japanese paper repositories (e.g., [5,15]). They were told that selected papers should be refereed and deal with human learning (i.e., a change in the cognitive state over time), which had occurred in interactive situations. G1 selected a short paper reporting a questionnaire survey on university students’ attitudes toward collaborative learning [12]. G2 selected a full paper presenting a computational model showing the mutual learning process of joint attention by caregiver and infant agents [10]. Note: We did not control for the difficulties of these papers; the page length and prerequisite knowledge of the papers were different.

Tool for Collaborative Concept Mapping. Despite many collaborative concept map tools (e.g., [2]), we could not find one with a detailed logging function. Therefore, we originally developed a concept map tool based on an existing open

source library of network visualizations [17]. Adding a function of communication with a server to the library, our environment makes it possible to operate on a single concept map simultaneously from different client Web browsers. The client-side script (JavaScript) regularly posts the status of maps (nodes and edges with labels) to a server as a JSON file to synchronize each client.

Video Conference Tool. We used Zoom [16], as it was a de facto standardized application when the study was conducted. Every participant in the study was familiar with the tool. During the class, participants were required to turn on their cameras, and the screen-sharing function and breakout rooms were utilized.

2.3 Procedure

The collaborative concept mapping was conducted in the fifth day of the class; however, the activities conducted before and after that day were also related in the latter analysis as follows:

1. Reading paper:

On the fourth day of the class, the participants were grouped based on their interest in a reading paper and were required to make a single-page summary of the paper before the target day of the class.

2. Individual concept mapping:

The participants were instructed to create concept maps eliciting important concepts from the above single-page summary. They were also provided an explanation of concept maps with a description based on Japanese Wikipedia. Following the instruction, the participants engaged in building concept maps individually for approximately 20 min.

3. Collaborative concept mapping:

The participants were instructed to integrate individual maps into one collaborative map through a group activity. They were presented an interface of collaborating concept maps, which were divided into three or four panels (one left panel for constructing a collaboration map and two or three right panels for presenting individual maps). Following the instruction, the participants were assigned breakout rooms according to their group. In each breakout room, they recorded their own process by themselves and freely used the screen-sharing function in Zoom. This collaborative concept mapping continued for approximately 30 min.

4. After the collaboration:

After the above target class, the participants were given an assignment to elaborate on their own individual map for the sixth class; the map was used as a tool to explain the reading paper to another participant who had not read it before. Following this paired work, the participants individually summarized the paper in a 10-min presentation format. During the seventh class, each participant gave a presentation and all of the class participants rated each presentation, including their own presentation.

3 Results and Discussion

The data obtained from the two groups (G1 and G2) were analyzed to demonstrate how the mutual understanding on the topic was developed through the visualization of a knowledge structure. Before presenting this process, we will show results relating to learning outcomes and a summary of a collaborative map to characterize each group.

3.1 Outcomes of the Collaboration in Each Group

After the collaboration activity, the participants' understandings were rated with the presentation introducing the paper (procedure 4). Table 1 and Table 2 show the mutual rating scores (ratings given by self, group members, and overall class (averaged with $n = 7$)).

Both for G1 and G2, the differences of average ratings did not reach significance level (G1: $t(6) = 1.44, p > .10$, G2: $F(2, 6) = 3.13, p < .10$). The mutual rating within the groups, however, revealed different characteristics for each group. In G1, the mutual scoring between two participants was not obviously asymmetrical, though P2 rated his partner higher than himself. To the contrary, the rating by the group members of G2 clearly divided them into low- and high-understanding participants. P5 was rated as the highest by the other participants, while he rated the lowest to be P3. That is, these members had an agreement between them as to the order of their understanding level.

Table 1. Scores of understanding rated on the presentation in G1

		Rated by		
		P1	P2	Class
Rated on	P1	4	4	4.14
	P2	5	4	4.57

Table 2. Scores of understanding rated on the presentation in G2

		Rated by			
		P3	P4	P5	Class
Rated on	P3	4	4	3	3.85
	P4	4	4	4	4.14
	P5	5	5	4	4.42

3.2 Collaborative and Individual Maps

The characteristics of each group can also be illustrated by the final collaborative maps. Figure 1 shows the maps constructed by the participants in each group. The left map shows the final collaboration map (procedure 3), while the right two maps were individually created by each participant (procedure 2). The map constructed by P4 is omitted because of the absence of commonalities with G2's collaborative map.

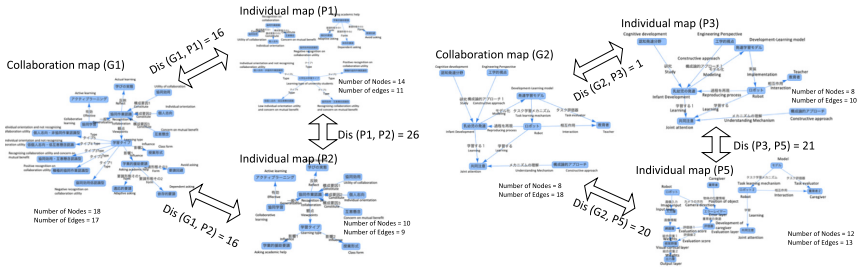


Fig. 1. Concept maps created by G1 (left) and G2 (right).

The figures also show several indices featuring each map computed with a network analysis library [11]. In particular, we focus on edit distance (noted as “Dis” in the figure), displaying the number of operations needed to transform one map to another [1]. This seems to be suitable for describing the collaborative editing process by examining the copied elements from individual maps.

From the values of edit distance, it is observed that an equal contribution to individual maps was made by P1 and P2 to G1’s collaborative map. On the other hand, G2’s collaborative map is dominantly based on P3’s individual map. From this result, G1 is characterized as having reached a sufficient mutual understanding. To the contrary, G2 failed to integrate individual maps, based primarily on the map created by the lowest understanding participant. This difference in the group could not be attributed to the initial commonalities of the individual maps; the two maps in G1 have larger distances than those in G2.

3.3 Process of Collaborative Concept Mapping

To reveal how the success and failure of collaborative concept mapping occurred, Fig. 2 summarizes the edit process in each group. These figures were created with ELAN [8] to annotate operators who edit the map. The upper line graph in each figure indicates edit distance from the collaborative maps to each individual map at each time point. The bottom several rows indicate the timing of the edition made by each participant.

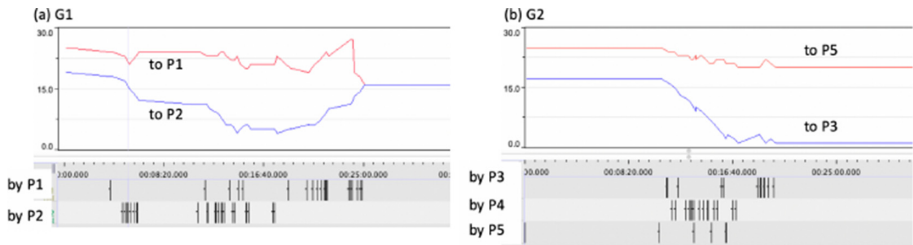


Fig. 2. Edit distance from individual maps to collaborative map.

In G1, the line representing “to P2” is described as U-shaped: at first, the collaborative map is close to P2’s map, then it moves away from P2’s map while adding intrinsic elements of P1’s individual map. Finally, the two individual maps contribute equally to the collaborative map. This result shows a successful example of mutual knowledge building in online collaborative concept mapping. In contrast, the editing process shown in G2’s graph does not present sufficient integration. The figure contains two lines (to P3, to P5), omitting a line that has no commonalities with the collaborative map. The distance to P3 is always smaller than P5, showing dominant dependence on P3’s map.

To explore a more detailed process of collaboration, Fig. 3 summarizes the relationship between the number of editions closing to or going away from each individual map and operators of these editions. In Fig. 3a, we can observe a clear relationship. Each edit of P1 and P2 created an element of the collaboration map, closing to his individual map. On the other hand, Fig. 3b shows different collaboration styles. P4, whose map did not contribute to the collaborative map, had the highest number of manipulations in the creation of this graph. P5, who was rated highest in understanding the topic, had the smallest contribution in constructing the collaboration map.

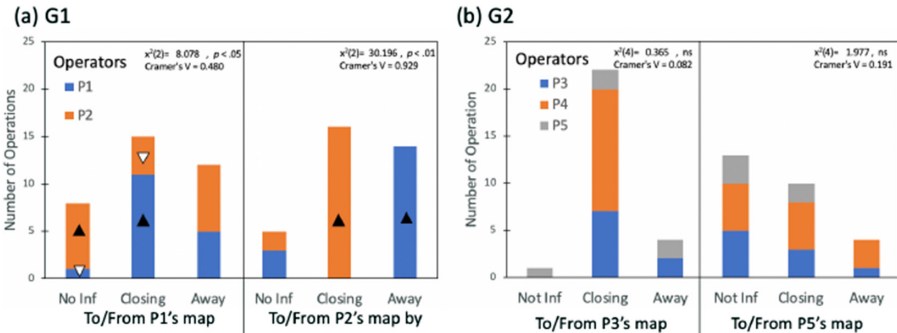


Fig. 3. The number of operations divided into types and operators. The triangles on the bars indicate significant results in the residual analysis of chi-square test ($p < .05$)

4 Conclusion

This study presents a learning task applicable to online collaborative learning. The two groups targeted in this study exhibited a contrasting process: one successfully integrated knowledge while the other failed. We contend that this study can contribute to the development of intelligent tutoring systems for online collaboration learning. In the proposed task, success and failure of the process could be easily detected with the relationships of edit distance. By using features obtained in this study, future studies can construct learning supports that will intervene in collaborative concept mapping.

However, this study did not determine the specific reason for the failure of collaboration. All members of G2 engaged in some role in the collaboration. Yet, this collaboration did not contribute to a sufficient mutual understanding. The reason for this failure may be attributed to several factors in the process. To clarify causal factors, we need to conduct further experimental studies to control for these variations.

Further, the process shown by G1 also has limitations. The two individual maps actually merged, but there is no evidence that this collaboration led to the construction of new knowledge. Many learning scientists [4, 9] promote the importance of conflict resolution in collaborative learning. Further study is needed to explore the conditions that occur during successful online collaboration using collaborative mapping from individual mapping.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number JP20H04299.

References

1. Abu-Aisheh, Z., Raveaux, R., Ramel, J.Y., Martineau, P.: An exact graph edit distance algorithm for solving pattern recognition problems. In: 4th International Conference on Pattern Recognition Applications and Methods 2015 (2015)
2. Cañas, A.J., et al.: CmapTools: a knowledge modeling and sharing environment. In: Concept Maps: Theory, Methodology, Technology, Proceedings of the First International Conference on Concept Mapping, Pamplona, Spain (2004)
3. Chi, M.T.H., Feltovich, P., Glaser, R.: Categorization and representation of Physics problems by experts and novices. *Cogn. Sci.* **5**, 121–152 (1981)
4. Chi, M.T., Wylie, R.: The ICAP framework: linking cognitive engagement to active learning outcomes. *Educ. Psychol.* **49**(4), 219–243 (2014)
5. CiNii [Web site] National Institute of Informatics (2021). Retrieved <https://ci.nii.ac.jp/>
6. Doherty-Sneddon, G., Anderson, A., O'Malley, C., Langton, S., Garrod, S., Bruce, V.: Face-to-face and video-mediated communication: a comparison of dialogue structure and task performance. *J. Exp. Psychol. Appl.* **3**(2), 105–125 (1997)
7. Dung, D.T.H.: The advantages and disadvantages of virtual learning. *IOSR J. Res. Method Educ.* **10**(3), 45–48 (2020)
8. ELAN (Version 6.0) [Computer software]. Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive (2020). Retrieved <https://archive.mpi.nl/tla/elan>
9. Miyake, N.: Constructive interaction and the iterative process of understanding. *Cogn. Sci.* **10**(2), 151–177 (1986)
10. Nagai, Y., Asada, M., Hosoda, K.: Acquisition of joint attention by a developmental learning model based on interactions between a robot and a caregiver. *Trans. Jpn. Soc. Artif. Intell.* **18**, 122–130 (2003)
11. NetworkX [Computer software] (2021). Retrieved <https://networkx.org/>
12. Nonaka, Y.: The examination of student' learning type based on belief of cooperation in university students. *Jpn. J. Educ. Technol.* **41**(Sulli.), 217–220 (2018)

13. Shimojo, S., Hayashi, Y.: Prompting learner-learner collaborative learning for deeper interaction: conversational analysis based on the ICAP framework. In: Proceedings of the 28th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education, pp. 177–182 (2020)
14. Tohyama, S., Miyake, N.: The evaluation of ReCoNote summaries for learner-centered integration. In: Third International Conference on Intelligent Networking and Collaborative Systems, Fukuoka, 2011, pp. 855–856 (2011)
15. JSTAGE [Web site] Japan Science and Technology Agency (2021). <https://www.jstage.jst.go.jp>
16. Zoom [Computer software] (2021). Retrieved <https://zoom.us>
17. vis.js [Computer software] (2021). Retrieved <https://visjs.org>



An Evaluation of a Meaningful Discovery Learning Support System for Supporting E-book User in Pair Learning

Jingyun Wang¹  and Hiroaki Ogata² 

¹ Durham University, Durham, UK
jingyun.wang@durham.ac.uk

² Kyoto University, Kyoto, Japan

Abstract. In this paper, an experiment was conducted to study the learning performance when learning new knowledge in groups with an e-book system and a meaningful discovery learning support environment. The participants studied target new knowledge with an e-book in pairs; at first, all the knowledge points that appear in the e-book were displayed and learners in each pair were encouraged to actively create relations between the knowledge concepts together; after completing the task, they can compare their learner-generated relations with expert-generated relations. The learning perception of one hundred and forty-three participants are analyzed and discussed.

Keywords: Meaningful learning · Discovery learning · Topic map · E-book

1 Introduction

Advance organizers are presented by Ausubel [1] to facilitate meaningful learning which refers to the non-arbitrary substantive incorporation of new concepts or propositions into the existing hierarchical framework of cognitive structure. When new material is presented, with the support of the advance organizer, the learner's attention can be directed to the key concepts and key relations, even to relevant prior concepts. It favors the understanding of new concepts and support the knowledge.

To support meaningful learning in e-book systems, a cache-cache comparison mode which encourages the learner to actively process the e-book information and discover the relation between the given key concepts is implemented in a visualization support system (VSSE) [2, 3], in addition to a reception comparison mode which provide complete versions of expert-generated topic maps to learners. A series of experiments had been conducted to examine the learner behaviors and performance while they did review activities with the support of e-books and VSSE. In previous work [3], we found that after review activities learners with low prior knowledge showed greater increases in performance than learners with high prior knowledge when encouraged to actively discover the relations between key concepts appearing in target learning content. This suggests that cache-cache comparison mode is more appropriate than reception comparison mode for learners with low prior knowledge. On the other hand, for learners

with high prior knowledge, learning mode made no significant difference to learning achievement; however, learners felt significantly less pressure and more satisfaction in reception comparison mode than in cache-cache comparison mode. In light of the above, for learners with high prior knowledge, reception comparison mode is indicated.

Instead of studying the effectiveness of VSSE on the review activity, the experiment in this paper studied the learning performance of learners who did self-study in pairs with e-book involving new knowledge under the support of the cache-cache comparison mode.

2 The Cache-Cache Comparison Mode in VSSE System

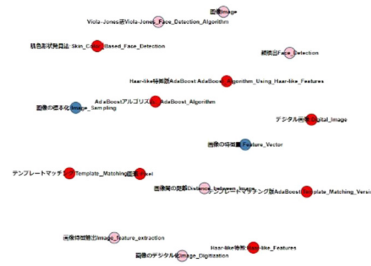


Fig. 1. The cache-cache comparison mode.

Figure 1 shows an instance of cache-cache comparison mode: the content of interest to the learner is pages 1–43 of an e-book titled “Face detection”. First, “cache-cache comparison” mode displays all the knowledge points (KPs, a KP is defined as “a minimum learning item which can independently describe the information constituting one given piece of knowledge in the content of a specific course.”) that appear in the page range of interest in red; the related KPs that do not appear in the pages of interest in ranges in blue; and their upper concepts in pink. Then firstly the learner is required to classify the KPs by connecting them to their pink upper concepts; next, the learner is encouraged to find out the relations between KPs by connecting red nodes or connecting red nodes to blue nodes. The descriptions of the relation arcs made by the learner can be modified and saved anytime. After the learner completes the relation map, she/he can click the “Compare with experts” button. Finally, all the relations extracted from the ontology will be displayed as red lines. The learner can easily compare the red lines with the black lines that she/he has made.

3 Experimental Description

One hundred and forty-three first-year undergraduates from the same class at a University in Japan participated in this study. These students were all taught by the same instructor, who had taught computer science for more than twenty years. Before the experiment, all the participants had studied Information Science for 13 weeks with learning support

system environments (Moodle, Mahara and an e-book system). Learning performance measurement techniques in this experiment included learning achievement tests (pre- and post-test), and a questionnaire for measuring learning related perceptions. Both test sheets had been developed by experienced teachers. The questionnaire consisted of 11 questions involving responses on a seven-point Likert scale (1–3: strongly to slightly disagree, 4: neutral, 5–7: slightly to strongly agree). Question content was related to learning perception, specifically technology acceptance [4, 5] cognitive load [6], and satisfaction with learning mode [7]. The reliability of this Japanese version questionnaire has discussed in the previous work [3].

Firstly, all the participants took the pre-test consisting of 19 multiple-choice questions which aimed at evaluating their prior knowledge of Information Science. Subsequently, they received a training about how to use Japanese version of VSSE [3], which can be opened in the browser of any PC, tablet or smart phone. During the 15-min training, the study procedures were demonstrated using one sample map in cache-cache comparison mode; participants were then encouraged to repeat the demonstrated actions so as to familiarize themselves with system operation. After the training, they were assigned randomly in pairs to study with the support of e-book systems and VSSE for 60 min. An e-book titled “Face detection” was chosen as target learning content. Since the target learning materials are new for everyone, so they allowed to discuss with their peers in the same group and complete the topic map on VSSE (as Fig. 1) together (only one map will be submitted per group). After that, they took a post-test consist of 3 multiple-choice questions and a questionnaire.

4 Learning Perception Results

System evaluation and feedback about the learning activity are shown in Table 1.

Table 1. Results for learning perception.

Item	Satisfaction	Mental effort		Mental load		Technology acceptance	
		Understand the purpose (1–7)	Learn the KPs (1–7)	Distraction (1–7)	Pressure (1–7)	Easiness (1–3:no 4–6:yes)	Usefulness (1–3:no 4–6:yes)
Mean	3.75	4.77	4.61	3.52	3.74	3.86	3.90
S.D	1.26	1.48	1.26	1.52	1.72	1.69	1.50

In terms of “mental effort,” the average rating for “effort required for understanding the purpose of the learning activity” was higher than 4 (the neutral point) but still lower than 5, indicating that most participants felt that it is a bit difficult to understand the purpose of the activity. The average ratings of “effort required for learning the target KPs” was 4.61; this suggests that the difficulty of the learning activity slightly difficult for the participants. In terms of “mental load,” the average rating for degree of distraction

and degree of pressure was less than 3; this implies that the participants felt little pressure while concentrating on learning with VSSE. In terms of “technology acceptance” measures, the average rating on the “perceived ease of use” item and the average rating of “perceived usefulness” was slight lower than 4; most participants reported that VSSE was a bit difficult to operate and become familiar with and did not believe VSSE was useful for improving their learning performance in studying new knowledge. In terms of the average ratings (using the mean rankings for the five related items) for “satisfaction with learning mode” were 3.75 (slightly lower than 4); this implies that most participants were slight dissatisfied with the learning mode.

5 Discussion, Conclusion and Future Work

The learning perception result differences between the previous experiments and the one in this study lie in “mental effort,” “satisfaction with learning mode” and “technology acceptance”. In previous finding [3] which compared the learning effectiveness of two different VSSE modes in supporting review activity, the average rating for “effort required for understanding the purpose of the learning activity” was less than 4 (the neutral point) for both groups, indicating that most participants in both groups felt that they could easily understand the purpose of the activity; The average ratings of “effort required for learning the target KPs” were 4.28 and 3.78 for learner who studied with cache-cache mode and those who studied with reception mode, respectively; this suggests that the difficulty of the learning activity was moderate (neither too easy nor too difficult) for the participants in both groups. Furthermore, the average rating on the “perceived ease of use” item was 4.08 for the cache-cache mode and 4.47 for the reception mode; most participants reported that VSSE was easy to operate and become familiar with. The average rating of “perceived usefulness” was 4.50 for the cache-cache mode and 4.76 for the reception mode, which implies that in both groups, most participants thought that VSSE was useful for improving their learning performance in review activity. Finally, the average ratings (using the mean rankings for the five related items) for “satisfaction with learning mode” were 4.51 and 4.96 for the cache-cache mode and the reception mode, respectively; this implies that most participants in both groups were satisfied with the provided learning mode.

In the other word, the self-study activity with cache-cache mode is a bit difficult than the review activity with either cache-cache mode or reception mode. Without sufficient prior knowledge and guidance, the unassisted discovery learning in the experiment in this research lead to additional cognitive load, lower technology acceptance and dissatisfaction, this finding has also been cautioned by many previous researches [1, 8, 9]. Although the cache-cache mode hid all the key relation and only presented the key concepts, there were still too much new information for novices in a self-study activity. In summary, compared to the previous experiments [3] where participants attended lecture of the target content weeks before the review activity, the participant precepted higher mental effort, lower satisfaction with learning mode and lower technology acceptance. Due to the small amount of the learner data, further experiment still needs be conducted to confirm those conclusions.

References

1. Ausubel, D.: In defense of advance organizers: a reply to the critics. *Rev. Educ. Res.* **48**(2), 251–257 (1978)
2. Wang, J., Ogata, H., Shimada, A.: A meaningful discovery learning environment for e-book learners. In: *IEEE Global Engineering Education Conference 2017*, pp. 1158–1165 (2017)
3. Wang, J., Shimada, A., Oi, M., Ogata, H., Tabata, Y.: Development and evaluation of a visualization system to support meaningful e-book learning. *Interact. Learn. Environ.* (2020). <https://doi.org/10.1080/10494820.2020.1813178>
4. Chu, H.C., Hwang, G.J., Tsai, C.C., Tseng, J.C.R.: A two-tier test approach to developing location-aware mobile learning systems for natural science courses. *Comput. Educ.* **55**(4), 1618–1627 (2010)
5. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly* **13**(3), 319–340 (1989)
6. Sweller, J., Ayres, P.L., Kalyuga, S., Chandler, P.A.: The expertise reversal effect. *Educ. Psychol.* **38**(1), 23–31 (2003)
7. Chu, H.C., Hwang, G.J., Tsai, C.C.: A knowledge engineering approach to developing Mindtools for context-aware ubiquitous learning. *Comput. Educ.* **54**(1), 289–297 (2010)
8. Alfieri, L., Brooks, P.J., Aldrich, N.J., Tenenbaum, H.R.: Does discovery-based instruction enhance learning? *J. Educ. Psychol.* **103** (1), 1–18 (2011)
9. Mayer, R.: Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. *Am. Psychol.* **59**(1), 14–19 (2004)



Towards Semantic Comparison of Concept Maps for Structuring Learning Activities

Carla Limongelli¹(✉), Carmine Margiotta¹, and Davide Taibi²

¹ Department of Engineering, Roma Tre University, Rome, Italy
limongel@ing.uniroma3.it, car.margiotta@stud.uniroma3.it

² Institute for Educational Technology, National Research Council of Italy,
Palermo, Italy
davide.taibi@itd.cnr.it

Abstract. Concept maps are significant tools able to support several tasks in the educational area such as curriculum design, knowledge organization and modeling, students' assessment and many others.

Algorithms for comparing graphs have been extensively studied in the literature, but they do not appear appropriate for concept maps. In concept maps, concepts exposed are at least as relevant as the structure that contains them. Neglecting the semantic and didactic aspect inevitably causes inaccuracies and the consequently limited applicability in automated systems.

In this work, starting from an algorithm which compares didactic characteristic of concept maps, we present an extension which exploits a semantic approach to catch the actual meaning of the concepts expressed in the nodes of the map. We also present experimental results.

Keywords: Concept maps · Natural Language Processing · Similarity measures

1 Introduction

Concept Maps (CMs) are a significant tool able to support several tasks in the area of education, for instance with applications in curriculum design, knowledge organization and understanding, and also evaluation of learning achievements. For this reason, CMs are suitable to contribute to the development of ITS especially concerning domain knowledge representation and learner model. CMs facilitate knowledge organisation by making explicit the structure of the relationships between concepts. Authors in [1] proposed the use of Natural Language Processing approaches to automatically extract from text concepts that constitute the nodes of a CM representing the text. The application of this method to teaching materials provides useful hints on knowledge organisation thus supporting the assessment of course content based on active reflection.

CMs are mainly used within ITS to organize system knowledge, in the assessment of students' learning processes, and to model domain and learner knowledge. To this aim, it is relevant to elaborate methods and approaches to automatically compare CMs to detect similarities and to identify whether and in

which context the same concepts are used by different knowledge representations. The purpose of this work is to design a method for automatic comparison (and thus management) of concept maps. As graphs, concept maps are characterized by their structure and the relationships between nodes. We thus need to deal with both the structural and semantic recognition algorithms for the maps. An algorithm for comparing concept maps that consider relations between arcs as prerequisite relations has been proposed in [2]. In this contribution, we extend the algorithm with a method that allows us to identify similar nodes at the semantic level, making an automatic comparison between two CMs effectively applicable.

2 Theoretical Background

To deal with the semantic comparison between concept maps, we analyze the most significant comparison techniques.

Algorithms for Word Comparison. The use of a knowledge base such as Wordnet [4] has the great advantage of producing results totally explicable and with a high degree of control, being possible to explore the entire knowledge base to make the best choices, especially in the disambiguation phase. Lin metric [3], applied to Wordnet, provides very reliable results with the best efficiency among those taken into analysis. The experimental results observed in [7] indicate Lin's metric as one of the most reliable and, at the same time, more efficient metric than Li's and Jiang-Conrath's ones.

For comparing a pair of nodes of two different CMs, we must consider that they represent a well-defined semantic context. Clustering algorithms are best suited to take advantage of this property, because of their ability to group similar elements (in this case the word senses). Since word senses are non-numerical data, we use k-medoids [5] clustering that is similar to k-means but assumes, as a constraint, that the centroid of the cluster must be in the corpus of senses of the words that are to be clustered.

Metrics for Sentence Similarity. In concept maps, concepts represented in the node can be expressed by short sentences, and in this case, it is necessary to apply different methods suitable to compare sets of words.

In the approach proposed in [6], authors use a similarity matrix which allows calculating the similarity between a pair of vectors representing the semantic structures. This leaves a high degree of freedom on the choice of the algorithms for words comparison as well as on those of disambiguation.

Metrics for Structural Graph Similarity. Structural comparison of a pair of concept maps is not a trivial task. An effective approach for computing structural similarity between two graphs is based on their spectra [9]. An important property of this distance measure is that it can be used to compare graphs of different sizes and it is particularly efficient on CMs. Two efficient distant measures based on this principle, "Spectral Distance of Degree Matrices", and " ℓ_2 norm of difference of distance matrices" are presented in [8]. We have considered these two measures for comparing our approach as discussed in Sect. 4.

3 A New Measure for Maps Comparison

The algorithm we propose starts from considering the Prerequisite Constraint Measure (PCM) presented in [2] which is designed for capturing the similarity of two concept maps from the point of view of their common prerequisites. For each common concept of two CMs, this measure computes the amount of common knowledge expressed by its predecessors. If the same concept is presented with different prerequisites in two CMs, it means that from the point of view of prerequisites the two maps are different. For example, given two CMs presented in the left-hand side of Fig. 1 the set of common nodes is given by $CN = \{Nucleus, Cytoplasm\}$ and the value calculated with the PCM method is 0.514. The new algorithm is augmented by the semantic comparison of the concepts expressed into the maps. We compute the semantic similarity of all the nodes in the two maps (see right-hand side of Fig. 1) and we discover that there are similar concepts that have been ignored by the PCM (Cells and Eukaryotes). In this case the value computed by PCM with semantic similarity (PCMS) is 0.926 that is much higher than the value of 0.514 calculated without the inclusion of semantic comparison.

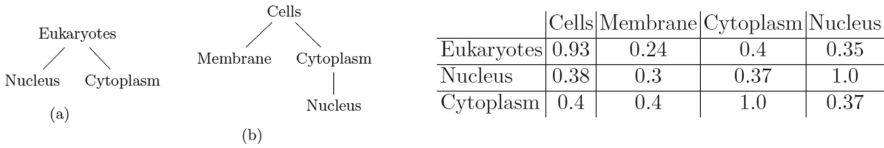


Fig. 1. Two simple concept maps on the left-hand side. The semantic distances of all their concepts on the left-hand side.

PCMS algorithm first computes all the similarities between the concepts of the two maps and then it considers *similar* those concepts that exceed a given threshold. The threshold has been determined with the following procedure: (i) we generate a set of 15000 pairs of concepts maps through a pseudo-random algorithm; (ii) we compute the distance between these pairs, by applying the PCM measure; (iii) we compute the spectral distance of the degree matrix [9]; (iv) we compute the distance of the distance matrices; (v) we apply a *hill-climbing* algorithm to exclude peak values i.e., values computed by PCM that are too far from the values computed in the two previous steps.

4 Tests and Results

We have pseudo-randomly generated 15,000 pairs of graphs with size ranging from 7 to 30 nodes. The content of the nodes is a number between $[0, \dots, 1]$ so that the comparison between two nodes of the graphs returns the average semantic similarity of each pair of nodes. This value is consistent with the measure

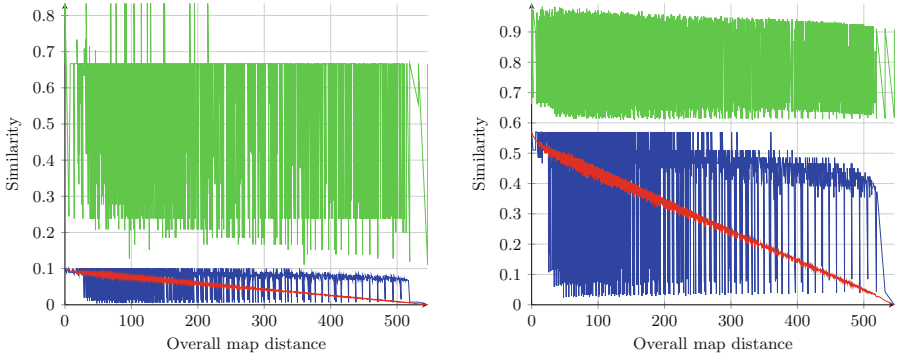


Fig. 2. Test on CMs with the 10% (left) and 57% (right) of similar nodes. The green line represents PCMS and the red and blue lines the structural distances presented in Sect. 2. (Color figure online)

resulting from the semantic comparison between two “real” nodes (i.e. a number between $[0, \dots, 1]$). For the two tests illustrated in Fig. 2, we have considered the generation of 15,000 pairs of graphs. For the first generation (left-side of Fig. 2), we have set 10% of semantically similar nodes, for the second test (right-side of Fig. 2), we have generated 57% of semantically similar nodes. PCMS is compared with the spectral distance of degree matrices and the ℓ_2 norm of difference between distance matrices, both associated with the average semantic similarity of every node. On the x -axis there is the overall map distance between the compared CMs and on the y -axis there is the result produced by the similarity measures. The distance between distance matrices, in red, and spectral distance, in blue, show lower similarity degree than PCMS. In fact, these measures are not able to capture the prerequisites relationships between concepts as the PCMS does.

5 Conclusions

The introduction of the semantic layer has significantly improved the existing *didactic* measure for CM comparison and the resulting method, as a whole, yields very significant results in detecting similarities between CMs. The experimentation, at the moment, has been carried out by using CM randomly generated due to the lack of adequate datasets of CMs suitable for testing however, these first encouraging results pave the way for experimentation on real concept maps.

Even though there are evidences that the use of CMs within ITS is undoubtedly effective, there are still many progresses to be done, both for what concerns the generation of algorithms that work on CMs (especially to identify pedagogical characteristics), and regarding standardization in structuring CMs. On the one hand, standardization would constrain the flexibility that is one of the most important characteristic of CMs but, on the other hand, it would support

the global sharing of CMs and, consequently, their extensive use as well as the development of methods and algorithms to elaborate them automatically.

References

1. Atapattu, T., Falkner, K., Falkner, N.: A comprehensive text analysis of lecture slides to generate concept maps. *Comput. Educ.* **115**, 96–113 (2017). <https://doi.org/10.1016/j.compedu.2017.08.001>
2. Limongelli, C., Sciarone, F., Lombardi, M., Marani, A., Temperini, M.: A framework for comparing concept maps. In: 2017 16th International Conference on Information Technology Based Higher Education and Training (ITHET), pp. 1–6. IEEE (2017)
3. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 296–304. ICML 1998, Morgan Kaufmann Publishers Inc., San Francisco (1998)
4. Miller, G.A.: Wordnet: a lexical database for English. *Commun. ACM* **38**, 39–41 (1995)
5. Park, H.S., Jun, C.H.: A simple and fast algorithm for k-medoids clustering. *Expert Syst. Appl.* **36**(2), 3336–3341 (2009). <https://doi.org/10.1016/j.eswa.2008.01.039>
6. Stevenson, M., Greenwood, M.A.: A semantic approach to IE pattern induction. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 379–386. ACL 2005, Association for Computational Linguistics, Stroudsburg (2005). <https://doi.org/10.3115/1219840.1219887>
7. Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E.G., Milios, E.E.: Semantic similarity methods in wordnet and their application to information retrieval on the web. In: Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, pp. 10–16 (2005)
8. Wills, P., Meyer, F.G.: Metrics for graph comparison: a practitioner’s guide. *PLoS One* **15**(2), 1–54 (2020). <https://doi.org/10.1371/journal.pone.0228728>
9. Wilson, R.C., Zhu, P.: A study of graph spectra for comparing graphs and trees. *Pattern Recogn.* **41**(9), 2833–2841 (2008). <https://doi.org/10.1016/j.patcog.2008.03.011>

Student Prediction



MOOC *Next Week* Dropout Prediction: Weekly Assessing Time and Learning Patterns

Ahmed Alamri¹(✉), Zhongtian Sun¹, Alexandra I. Cristea¹(✉), Craig Stewart¹,
and Filipe Dwan Pereira²

¹ Department of Computer Science, Durham University, Durham, UK
{ahmed.s.alamri, alexandra.i.cristea}@durham.ac.uk

² Institute of Computing Federal, University of Roraima, Boa Vista, Brazil

Abstract. Although Massive Open Online Course (MOOC) systems have become more prevalent in recent years, associated student attrition rates are still a major drawback. In the past decade, many researchers have sought to explore the reasons behind learner attrition or lack of interest. A growing body of literature recognises the importance of the early prediction of student attrition from MOOCs, since it can lead to timely interventions. Among them, most are concerned with identifying the best features for the entire course dropout prediction. This study focuses on innovations in predicting student dropout rates by examining their *next-week-based learning activities and behaviours*. The study is based on multiple MOOC platforms including 251,662 students from 7 courses with 29 runs spanning in 2013 to 2018. This study aims to build a generalised early predictive model for the weekly prediction of student completion using machine learning algorithms. In addition, this study is the first to use a ‘*learner’s jumping behaviour*’ as a feature, to obtain a high dropout prediction accuracy.

Keywords: Learning analytics · Early dropout prediction · Machine learning · Behavioural pattern

1 Introduction

Massive Open Online Courses (MOOCs) offer open access courses to unlimited learners in an online learning manner. MOOC as a term was first coined in 2008, followed by the naming of 2012 as the ‘Year of the MOOC’, when MOOC providers, such as Coursera, Udacity, edX and FutureLearn, were all launched and they have reached to millions of learners across the world [14, 26], MOOCs have proven a popular education choice and become a critical mainstream approach to democratise knowledge [12]. However, it should be noticed that only 3–15% of participants complete their courses [5]. Such a situation undermines the initial purpose of MOOC that provides free access for massive numbers of students. Therefore, academics are interested in exploring why participants drop out and how to improve their engagement with the course until completion [2].

Researchers intend to find the most predictive feature(s) of students’ dropout activity and thus enable early intervention. One usual way is to identify learning behaviour

indicators to raise precision and recall of MOOCs' completion prediction [5]. However, data is not always available for such indicator analysis. For instance, non-completion can be predicted by a linguistic analysis of discussion forum data [24]. Nevertheless, as students' comments only amount to 5–10% of posts in discussion forums, this feature is not applicable universally [21]. Additionally, numerous variables can be considered for non-completion analysis, such as student profile data (e.g., country, age, gender) and course-attended related data (e.g., reading, watching, writing, taking quizzes). To the best of the authors' knowledge, this study is the first to consider participants' learning paths and associated behaviours in weekly dropout prediction. According to [1], a learning path is an insightful dropout prediction feature as successful learners will follow the instructed path and exhibit the so-called catch-up learning behaviours. Conversely, learners may jump forward and backward in their learning sessions [7], defined as exhibiting jumping behaviour and they are more likely to quit in the process. This study also considers other features such as number of learning activities, to predict student completion in the following week. Hence, our research question and its respective sub-questions are formulated as follows:

1. *Are there (high) differences in the prediction of weekly dropout and whole course dropout?*
2. *Will the weekly predictive model be more accurate after considering student jumping behaviours and catch-up learning patterns during the course?*

The main contributions of this paper are:

- We compare the prediction of *weekly dropout* and dropout of the whole course.
- New feature: we are the first to incorporate students' learning patterns, specifically *jumping behaviours* into the weekly predictive model and demonstrate the effectiveness of it.
- We implement seven machine learning algorithms and demonstrate that our proposed method outperforms the current best-in-class.

2 Related Work

Under the context of MOOCs' rapid spread to millions of people, the low completion ratio encourages researchers to explore, reason and build prediction models for dropout since 2014 [8]. The prediction of MOOC completion, especially at an early stage, has been the primary concern of researchers in learning analytics. Existing studies mainly analyse long-term learner behaviours, i.e., discussion activity, clickstream data, and time spent, based on different machine learning (ML) methods. For instance [14] examined learners' study pattern under a predictive ML framework for a 12-week-long psychology MOOC course. They improved 15% in prediction accuracy (70% up to 85%), compared to baseline methods. However, the proposed model did not perform well at early dropout detection.

[25], targeted struggling learners who need early intervention to keep the engagement, by designing a prioritising at-risk student temporal model. They illustrated the

necessity of building an effective and robust ensemble stacking prediction model for such analysis. [36], used data from the first two weeks of study, to allow for early intervention and they achieved accuracy of 80%.

Another study, [10], generated an average of 92% precision and 88% accuracy result of dropout prediction based on a two-layer cascading classifier structure. Additionally, [16] built an ML-based sliding window model based on Support Vector Machine for course completion prediction, which allowed MOOC instructors and designers to track potential dropouts.

However, all the above studies mainly focus on predicting participants' dropout activities for the entire course rather than in the upcoming week.

This paper focuses on predicting students' weekly completion, which we define (following the overall completion in other studies [13], applied to the week level) as accomplishing 80% of learning activity in the following week during the entire course. For example, we will predict students' completion of the second week by using their previous learning behaviours in the first week only. In addition, the model will predict students' completion of the fifth week by using their previous four weeks learning pattern. [11] demonstrated that clickstream-based features are much more predictive for drop out study. This paper will mainly use clickstream-based learning topics accessed for prediction. Additionally, according to [1], participants' learning patterns (linear learning behaviour followed by instructed learning path or jumping learning behaviour opposite the former) are an insightful feature for drop out prediction. We are the first to incorporate the students' learning patterns into our weekly drop out predictive model by reviewing their previous behaviours.

3 Methodology

Future Learn is one of the youngest massive online learning platforms (since 2012), and the European counterpart to USA's Coursera, EdX, etc., which now supports 327 courses created by 83 partners and reached 3 million students by 2018 [7]. As it is a newer platform, there are fewer studies performed on it. We fill this gap by selecting courses delivered through it. This study analyses a massively large dataset of 29 runs (Each course has run several times over years) of 7 multidisciplinary courses which falls under four main categories: Computer Science, Literature, Business and Psychology. The courses have been delivered through FutureLearn by two universities in the United Kingdom (University of Warwick and Durham University) between 2013 and 2018. The studied courses have a length of 4 to 10 weeks. The structure of these courses is based on a weekly learning unit. Every learning week includes so-called 'steps', which cover images, videos, articles and quizzes. Having joined a given course, learners can access these steps and optionally mark them as completed or solved quizzes. These steps also allow comments, replies and likes on these comments, from different users enrolled within the course. Moreover, quizzes can be frequently attempted, until the correct answer is obtained.

We use raw data and aggregate data, i.e., data composed from different raw data sources. We use data for early prediction as well as for general descriptive analysis. We employ data generated with various techniques: e.g., generated applying sentiment

analysis on student information exchange, and to limit it somewhat for the current paper, we have decided to perform a first aggregation step based on the weekly learning unit, which is used as a **synchronisation** point in instructor led FutureLearn courses.

In total, we have obtained interactional educational data (not publicly available) for 251,662 students shown as below in Table 1. Enrolled refers to registered students and accessed refers to students who have accessed the course at least once. It can be seen from the data in Table 1 that about half of enrolled students in MOOC do not access the course contents after the course has started. Each course has several runs as they are popular and held for more than one term. ‘The Mind Is Flat’ is the largest course among others in term of enrolled students, accessed students and number of runs see Table 1.

Table 1. Courses’ summary

Course	Enrolled	Accessed	Run
Open Innovation in Business (OI)	6071	2798	3
Leading and Managing People-Centred Change (LMPCC)	10417	6575	3
Babies in Mind (BIM)	48771	26175	6
Big Data (BD)	33427	16272	3
Shakespeare (SHK)	63625	29432	5
Supply Chains (SUP)	5808	2912	2
The Mind is Flat (THM)	83543	39894	7
Total	251662	124058	29

As we have used a massive dataset for different courses, we have prepared the training and testing sets based the last Run. For example, in the The Mind Is Flat course, we extracted data from several runs (1–6), with students activities between 2013 to 2017, to train our models, and to test the model, we used a new data set from a different Run (Run 7) that contains students’ activities in 2018 - see Fig. 1 - which is similar to some extent to transformer models [23].

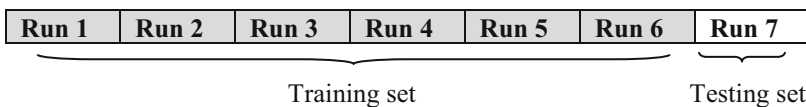


Fig. 1. The Mind Is Flat course

Moreover, we have incorporated students’ learning patterns, specifically *jumping behaviours*, for now, into the weekly predictive model, by adding a new column that presents number of jumping activates for each student in each week. To demonstrate the effectiveness of jumping behaviours, we compared the performance of weekly prediction models with and without students’ jumping behaviours. In addition, we run the features’

importance to identify the best indicators (features) to predict student dropout in each week.

3.1 Sentiment Analysis

In this research, the power of Natural Language Processing (NLP) has been used to analyse student comments and use them as features to predict their dropout activities. A tool called Textblob¹ has been employed, in order to classify students' comments into three categories: *positive*, *neutral* and *negative*. TextBlob is an NLP-oriented Python library, which measures polarity and subjectivity of a textual dataset for certain tasks, such as sentiment analysis, classification, part-of-speech tagging, extraction and more complex text processing tasks [20].

3.2 Weekly Prediction

Although a considerable amount of literature has been published on the prediction of MOOC dropout rates, there is no formal definition of student dropout [22]. Researchers in the domain have been using a variety of definitions. In this current research, we have prepared the dataset based on the *weekly prediction technique*, to determine at-risk students at an early stage. It is believed that predicting at-risk students from their previous weeks' activities may improve the model prediction performance. Therefore, in this study, we have implemented seven predictive models, to provide early intervention for learners at-risk in the following week. Each week, we predict the students who do not access 80% of the topics in the coming week, by using previous week/weeks activities as input for our model. The results are generated by seven chosen ML algorithms. We compared our *weekly prediction method* (see Eq. 1) with the more traditional method of predicting students' dropout from the whole course (the students who do not access 80% of the whole course, see Eq. 2).

$$Dropout = \sum_{week=2}^{\infty} accessedsteps < \sum_{week=2}^{\infty} totalsteps \times 0.8 \quad (1)$$

Weekly dropout prediction (WP)

$$Dropout = \sum_{allweeks} accessedsteps < \sum_{allweeks} totalsteps \times 0.8 \quad (2)$$

Whole course dropout prediction (CP)

Although, about 3–15% of participants complete their courses in MOOC [5], dropout is a gradual process. We are interested in analysing and predicting those weekly dropouts. Figure 2 presents the number of weekly dropouts over time. Clearly, participants are most likely to drop out in the first few weeks. Therefore, identifying those early dropouts is important for prediction. Moreover, using the jumping behaviour feature, we can capture those early dropouts and thus improve the accuracy.

3.3 Proposed Machine Learning Model

We present an overview of the proposed model to predict students' future activities, such as *next week dropout*. The first phase is to clean the datasets, by removing the blank

¹ <https://textblob.readthedocs.io/en/dev/>.

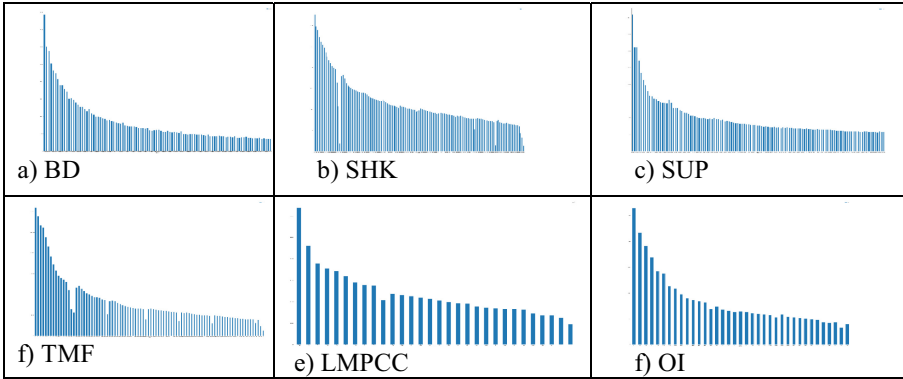


Fig. 2. Remaining students over time in different courses (a–f)

values and missing data. Still, the literature has reported that class imbalance can affect ML algorithms' performance. Due to the massive different completers' ratio to non-completers in our dataset, we set the class weight [5, 28] to the inverse of the frequency of different classes. In terms of best performing learning algorithms, the use of random forest (RF) (e.g., [15, 29–31]) has appeared in the literature among the most frequently used approaches for the student classification tasks. Additionally, Ensemble Methods, such as boosting, error-correcting have been shown to often perform better than single classifiers, such as SVM, KNN and Logistic Regression [28, 32]. In addition, KNN is an instance-based method, whilst logistic regression is a functional model.

To build our model, we employed several competing ML ensembles methods, as follows: Random Forest (RF) [3], Gradient Boosting Machine (Gradient Boosting), [33] Adaptive Boosting (AdaBoost) [34] and XGBoost [32] to proceed with exploratory analysis. Ensembles refers to those learning algorithms that fit a model via combining several simpler models and converting weak learners into strong ones [26]. In cases of binary classification (like ours), Gradient Boosting uses a single regression tree to fit on the negative gradient of the binomial deviance loss function [24]. XGBoost, a library for Gradient Boosting, contains a scalable tree boosting algorithm, which is widely used for structured or tabular data, to solve complex classification tasks [32]. AdaBoost is another method, performing iterations using a base algorithm. At each interaction, AdaBoost uses higher weights for samples misclassified, so that this algorithm focuses more on difficult cases [34]. Random Forest is a method that uses a number of decision trees constructed via bootstrapping resampling and then applying majority voting or averaging to perform the estimation [3].

The current study used a balanced accuracy score (BA) to evaluate the performance of the models; this metric is widely used to calculate accuracy for *imbalanced* datasets, by preventing the majority of negative samples from biasing the result [9]. Moreover, we used the McNemar's [35] test to measure the significance of any improvement in the models after considering student jumping behaviours. Significance levels were set at the 5% level ($P \leq 0.05$).

4 Results and Discussion

This section shows the performance results generated by our seven chosen ML algorithms: Random Forest (RF), Adaboost Classifier (AdaB), XGBoost (XG), Gradient-Boosting (GBoost), k-nearest neighbour (KNN), Logistic Regression (LR), and extra-Trees Classifier. We examine students' learning pattern, accessing time and registration date as mentioned before, for the coming week dropout prediction. Figure 3 shows that participants are more likely to complete the weekly learning activities at the beginning and dropout as time has passed. Around 7500 students have completed the first week in Big Data course. In contrast, only 2223 completed week 5. Therefore, weekly prediction is a reasonable approach to determine at-risk students at an early stage.

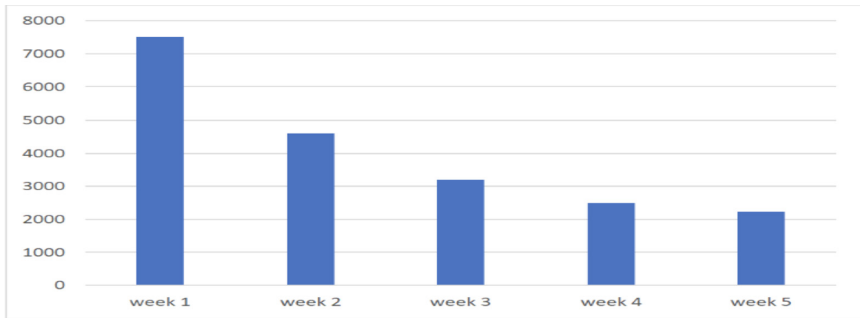


Fig. 3. Number of completers students in each week (Big Data course)

4.1 Weekly Prediction

Table 2 shows the performances of models for courses, evaluated by Balanced Accuracy score, a commonly used metric for binary classification of unbalanced dataset. In general, the most robust model is Random Forest (RF), as it outperforms in four courses: 'Supply Chain', 'The Mind is Flat', 'Big Data', and 'Babies in Mind'. Table 2 also shows the performance of several predictive models for both **CP** (whole course dropout, which means if the learner did not access 80% of the topics in the whole course) and **WP** (weekly dropout prediction which means if the learner did not access 80% of the topics in the next week). The input data was extracted from the first week of each course. The two results show that all seven models performed better with weekly predictions using the same number of previous weeks' data and achieved higher accuracy.

The prediction accuracy differences between weekly dropout prediction and the whole course dropout prediction are highlighted in Table 2. The results show clearly how the method of weekly prediction has contributed in terms of accuracy for dropout prediction from early stage (week 1). Figure 4 shows the most robust prediction models of weekly drop out and drop out of the whole course.

Table 2. Results (balanced accuracy score (BA)) for prediction models in week 1 for both “weekly dropout prediction” and “dropout from the whole course”

Testing Balanced Accuracy score (BA)								
		AdaB	ExTrees	GBoost	KNN	LR	RF	XG
BIM	CP	50.00%	72.50%	58.13%	59.39%	78.22%	80.32%	54.78%
	WP	84.48%	79.07%	67.48%	69.50%	82.52%	83.05%	73.59%
BD	CP	50.00%	78.97%	53.77%	51.47%	86.77%	87.28%	51.44%
	WP	91.98%	90.76%	89.31%	89.39%	92.05%	92.03%	91.84%
SHK	CP	50.00%	69.17%	60.47%	61.42%	80.01%	81.02%	59.65%
	WP	87.15%	85.90%	81.90%	82.79%	84.07%	87.23%	86.97%
SUP	CP	50.00%	78.61%	67.69%	65.47%	88.53%	86.59%	61.85%
	WP	93.28%	90.26%	84.50%	81.33%	92.09%	91.45%	91.27%
TMF	CP	50.00%	74.31%	57.03%	58.77%	85.58%	86.77%	53.98%
	WP	91.27%	90.27%	84.77%	81.72%	90.07%	91.43%	90.66%
IMPCC	CP	89.12%	86.34%	78.39%	80.24%	88.12%	86.86%	84.08%
	WP	90.63%	88.98%	84.48%	85.11%	90.12%	88.40%	90.14%
OI	CP	90.67%	81.16%	77.50%	71.18%	83.00%	84.07%	78.87%
	WP	91.41%	87.21%	77.78%	74.75%	81.99%	85.02%	86.53%

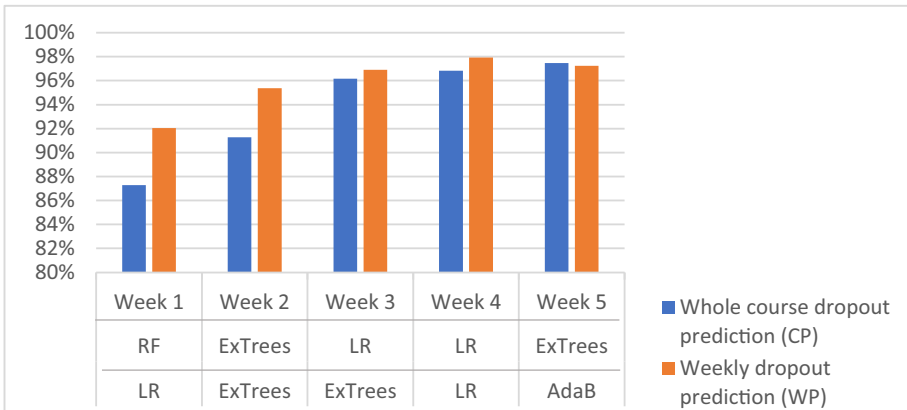


Fig. 4. Big data: weekly prediction vs entire course prediction per week with the best performing model

Weekly dropout prediction and the whole course dropout prediction are highlighted in Table 2 and Fig. 4. It has been shown how the method of weekly prediction has contributed to the increase in the accuracy of dropout detection from early stage.

4.2 Weekly Prediction with Jumping Activities

We have verified the improved performance of prediction after considering learners' jumping behaviour in four courses. For example, after incorporating the jumping learning pattern as a new feature to the dataset, accuracy rises by nearly 4% - from 86.9% to 91.3% in the XGBoost models in the Shakespeare course. In the Big Data course, the accuracy improves by nearly 3.3%, to 94% for the ExtraTrees Classifier. This weekly dropout prediction improvement is even more generalised in the Open Innovation in Business course, where all seven models implemented are more insightful and the highest accuracy is 94.95%, after considering the jumping learning behaviours. In addition, Table 3 shows that these results were statistically significant between WP and WPWJ (p value ≤ 0.05). Based on this analysis, module instructors could implement early interventions, judged on a weekly basis, to improve students' engagement at risk for the upcoming week dropout.

Table 3. Results (BA) of prediction models in week 1 for both weekly dropout prediction (WP) and weekly dropout prediction with jumping activities (WPWJ)

Testing Balanced Accuracy score (BA)															
Course	AdaB	P.V	ExTrees	P.V	GBoost	P.V	KNN	P.V	LR	P.V	RF	P.V	XG	P.V	
BD	WP	91.98%	$P \leq 0.05$	90.76%	$P \leq 0.05$	89.31%	$P \leq 0.05$	89.39%	$P \leq 0.05$	92.05%	$P \leq 0.05$	92.03%	$P \leq 0.05$	91.84%	$P \leq 0.05$
	WPWJ	94.73%	$P \leq 0.05$	94.07%	$P \leq 0.05$	92.52%	$P \leq 0.05$	92.01%	$P \leq 0.05$	94.66%	$P \leq 0.05$	94.56%	$P \leq 0.05$	94.68%	$P \leq 0.05$
SH	WP	87.15%	$P \leq 0.05$	85.90%	$P \leq 0.05$	81.90%	$P \leq 0.05$	82.79%	$P \leq 0.05$	84.07%	$P \leq 0.05$	87.23%	$P \leq 0.05$	86.97%	$P \leq 0.05$
	WPWJ	87.15%	$P \leq 0.05$	89.74%	$P \leq 0.05$	87.38%	$P \leq 0.05$	86.15%	$P \leq 0.05$	87.58%	$P \leq 0.05$	87.16%	$P \leq 0.05$	91.32%	$P \leq 0.05$
IM	WP	90.63%	$P \leq 0.05$	88.98%	$P \leq 0.05$	84.48%	$P \leq 0.05$	85.11%	$P \leq 0.05$	90.12%	$P \leq 0.05$	88.40%	$P \leq 0.05$	90.14%	$P \leq 0.05$
	WPWJ	94.62%	$P \leq 0.05$	93.81%	$P \leq 0.05$	92.12%	$P \leq 0.05$	88.03%	$P \leq 0.05$	94.60%	$P \leq 0.05$	91.29%	$P \leq 0.05$	94.38%	$P \leq 0.05$
OI	WP	91.41%	$P \leq 0.05$	87.21%	$P \leq 0.05$	77.78%	$P \leq 0.05$	74.75%	$P \leq 0.05$	81.99%	$P \leq 0.05$	85.02%	$P \leq 0.05$	86.53%	$P \leq 0.05$
	WPWJ	94.95%	$P \leq 0.05$	92.76%	$P \leq 0.05$	85.86%	$P \leq 0.05$	79.97%	$P \leq 0.05$	93.27%	$P \leq 0.05$	92.7609	$P \leq 0.05$	90.07%	$P \leq 0.05$

P.V: P-value to show significant difference between WP and WPWJ perfection

Figure 5 shows the Feature importance [26] in Random Forests (our most robust model for this cohort) for the Big Data course. The *Number of Jumping activities* feature is ranked as number one in terms of the importance in predicting students' dropout. However, Fig. 5 also shows that the *Number of accesses feature* is the second most important one.

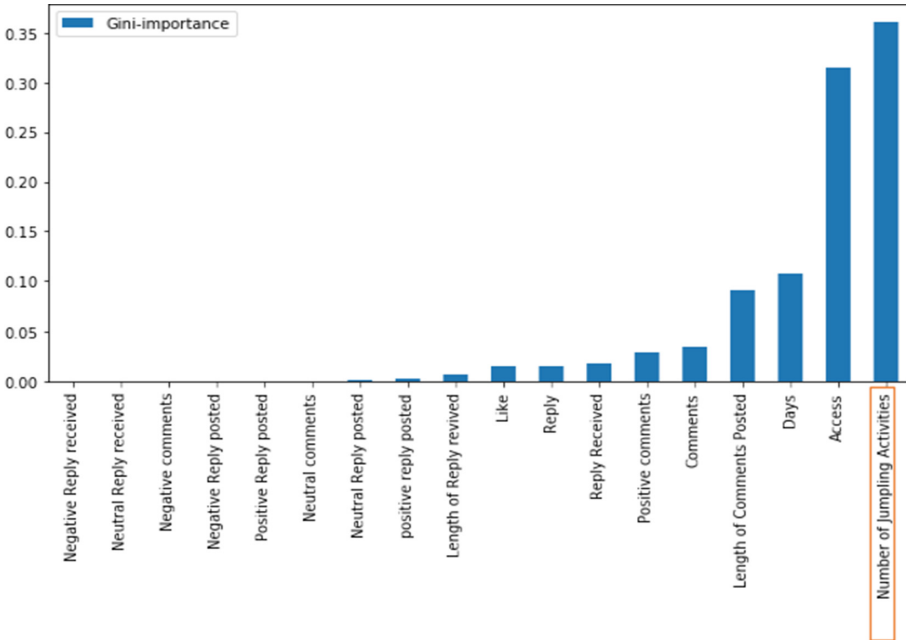


Fig. 5. Importance of predictive features

5 Conclusion

This study implements seven predictive models to provide information to enable early interventions for learners at-risk of drop out in the following week from MOOCs. To solve the imbalance dataset problem characteristic for MOOCs (in that successful, completing learners are usually much fewer than non-completers), we set the class weight to the inverse of the frequency of different classes. By reviewing students' learning patterns, particularly *jumping learning behaviours* and previous total course accessing activities, we propose robust machine learning algorithms to build predictive models across seven courses accessed by 251,662 students from 2013 to 2018. Our best model's accuracy (AdaBoost) for the next week dropout learner's detection ranges from 91.41% to 94.95% in the Open Innovation in Business course, after considering participants' jumping behaviours which could be utilised to personalise and prioritise assistance at-risk learners. Researchers can further add more learners' features (i.e., educational background, age, gender, nationality) to examine further improvements in prediction accuracy in a broad educational context. Additionally, researchers may also deploy the state of art language modelling like Bidirectional Encoder Representations from Transformers (BERT) and XLNet for natural language processing task (Yang et al. 2019) for comment analysis and sentiment analysis in MOOC prediction. Future studies can also explore knowledge representation learning methods based on students' knowledge background.

References

1. Alamri, A., Sun, Z., Cristea, A.I., Senthilnathan, G., Shi, L., Stewart, C.: Is MOOC learning different for dropouts? A visually-driven, multi-granularity explanatory ML approach. In: Kumar, V., Troussas, C. (eds.) ITS 2020. LNCS, vol. 12149, pp. 353–363. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_42
2. Balakrishnan, G., Co etzee, D.: Predicting student retention in massive open online courses using hidden Markov models. *Electr. Eng. Comput. Sci. Univ. Calif. Berkeley* **53**, 57–58 (2013)
3. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
4. Brinton, C.G., Chiang, M., Jain, S., Lam, H., Liu, Z., Wong, F.M.F.: Learning about social learning in MOOCs: from statistical analysis to generative model. *IEEE Trans. Learn. Technol.* **7**(4), 346–359 (2014)
5. Coates, A., et al.: Text detection and character recognition in scene images with unsupervised feature learning. In: 2011 International Conference on Document Analysis and Recognition, pp. 440–445. IEEE (2011)
6. Drummond, C., Holte, R.C., et al.: C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on Learning from Imbalanced Datasets II, vol. 11, pp. 1–8. Citeseer (2003)
7. Gardner, J., Brooks, C.: Student success prediction in MOOCs. *Model. Adap. Interact.* **28**(2), 127–203 (2018)
8. Fox, C.: Futurelearn has 3 million learners, March 2016. <https://www.futurelearn.com/info/press-releases/futurelearn-has-3-million-learners>
9. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 3121–3124. IEEE (2010)
10. Hong, B., Wei, Z., Yang, Y.: Discovering learning behavior patterns to predict dropout in MOOC. In: 2017 12th International Conference on Computer Science and Education (ICSE), pp. 700–704. IEEE (2017)
11. Jeon, B., Park, N.: Dropout prediction over weeks in MOOCs by learning representations of clicks and videos. arXiv preprint [arXiv:2002.01955](https://arxiv.org/abs/2002.01955) (2020)
12. Jordan, K.: Massive open online course completion rates revisited: assessment, length and attrition. *Int. Rev. Res. Open Distrib. Learn.* **16**(3), 341–358 (2015)
13. Khalil, H., Ebner, M.: MOOCs completion rates and possible methods to improve retention—a literature review. In: EdMedia+ Innovate Learning, pp. 1305–1313. Association for the Advancement of Computing in Education (AACE) (2014)
14. Kloft, M., Stiehler, F., Zheng, Z., Pinkwart, N.: Predicting MOOC dropout over weeks using machine learning methods. In: Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs, pp. 60–65 (2014)
15. Liang, J., Li, C., Zheng, L.: Machine learning application in MOOCs: dropout prediction. In: 2016 11th International Conference on Computer Science & Education (ICCSE), pp. 52–57. IEEE (2016)
16. Lu, X., Wang, S., Huang, J., Chen, W., Yan, Z.: What decides the dropout in MOOCs?. In: Bao, Z., Trajcevski, G., Chang, L., Hua, W. (eds.) DASFAA 2017. LNCS, vol. 10179, pp. 316–327. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55705-2_25
17. Pereira, F.D., et al.: Using learning analytics in the amazonas: understanding students’ behaviour in introductory programming. *Br. J. Educ. Technol.* **51**(4), 955–972 (2020)
18. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Learning with class skews and small disjuncts. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS, vol. 3171, pp. 296–306. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28645-5_30

19. Ren, Z., Rangwala, H., Johri, A.: Predicting performance on mooc assessments using multi-regression models. arXiv preprint [arXiv:1605.02269](https://arxiv.org/abs/1605.02269) (2016)
20. Robinson, C., Yeomans, M., Reich, J., Hulleman, C., Gehlbach, H.: Forecasting student achievement in MOOCs with natural language processing. In: Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, pp. 383–387 (2016)
21. Rose, C., Siemens, G.: Shared task on prediction of dropout over time in massively open online courses. In: Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs, pp. 39–41 (2014)
22. Sunar, A.S., White, S., Abdullah, N.A., Davis, H.C.: How learners' interactions sustain engagement: a MOOC case study. *IEEE Trans. Learn. Technol.* **10**(4), 475–487 (2016)
23. Vaswani, A., et al.: Attention is all you need. arXiv preprint [arXiv:1706.03762](https://arxiv.org/abs/1706.03762) (2017)
24. Wen, M., Yang, D., Rosé, C.: Linguistic reflections of student engagement in massive open online courses. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 8 (2014)
25. Xing, W., Chen, X., Stein, J., Marcinkowski, M.: Temporal predication of dropouts in MOOCs: reaching the low hanging fruit through stacking generalization. *Comput. Hum. Behav.* **58**, 119–129 (2016)
26. Yang, D., Sinha, T., Adamson, D., Rosé, C.P.: Turn on, tune in, drop out: anticipating student drop outs in massive open online courses. In: Proceedings of the 2013 NIPS Data-Driven Education Workshop, vol. 11, p. 14 (2013)
27. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. arXiv preprint [arXiv:1906.08237](https://arxiv.org/abs/1906.08237) (2019)
28. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) *MCS 2000*. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45014-9_1
29. Sharkey, M., Sanders, R.: A process for predicting MOOC attrition. In: Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs, pp. 50–54 (2014)
30. Nagrecha, S., Dillon, J.Z., Chawla, N.V.: MOOC dropout prediction: lessons learned from making pipelines interpretable. In: International World Wide Web Conferences Steering Committee Proceedings of the 26th International Conference on World Wide Web Companion, pp. 351–359 (2017)
31. Bote-Lorenzo, M.L., Gómez-Sánchez, E.: Predicting the decrease of engagement indicators in a MOOC. In: Proceedings of the Seventh International Learning Analytics and Knowledge Conference on LAK 2017, pp. 143–147. ACM Press, New York (2017)
32. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
33. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001)
34. Hastie, T., Rosset, S., Zhu, J., Zou, H.: Multi-class adaboost. *Stat. Interface* **2**, 349–360 (2009)
35. Everitt, B.: *The Analysis of Contingency Tables*. Chapman and Hall, London (1977)
36. Pereira, F.D., et al.: Early dropout prediction for programming courses supported by online judges. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds) *AIED 2019*. LNCS, vol. 11626, pp. 67–72. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23207-8_13



Internet of Things (IoT) Based Support System for Diabetic Learners in Saudi Arabian High Schools

Mona Alotaibi^(✉)  and Mike Joy^(✉) 

Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK
{mona.alotaibi,m.s.joy}@warwick.ac.uk

Abstract. In this workshop position paper, we identify the importance of a proposed system to monitor and assess diabetic students using Internet of Things (IoT) technology and a decision support system. We survey the current studies on the application of IoT in the Saudi Arabian educational system and related work. The model of Unified Theory of Acceptance and Use of Technology will be used to specify the critical factors that affect the use of the system for diabetic students in Saudi Arabian high schools. Finally, our research is at the beginning phase, so future work will identify the academic issues of the diabetic students and factors that affect the system usage by using a mixed method approach. In addition, the proposed decision tree algorithm will be implemented and evaluated.

Keywords: UTAUT · Internet of Things · Decision support system · High schools · Diabetic students · Technology Acceptance

1 Introduction

Schools find it challenging to use the Internet of Things (IoT), which has the potential to dramatically change learning, teaching, and monitoring. The learning, management, and relationship processes between all those involved in education may benefit from the IoT, because the associated physical devices ensure that people are connected and active. Recently, the importance of IoT has been explicitly reported in the medical field, but implementing IoT in education, unlike in other fields, is extremely challenging. The IoT should ensure the creation of an environment that supports the acquisition of knowledge in a natural, novel, and effective manner such that it is consistent with learners' expectations.

The importance of a system for monitoring and assessing diabetic students is evident, since in Saudi Arabia many critical issues could be faced by diabetic students and teachers. First, students could experience severe health situations during their school day, such as fainting caused by diabetes [7]. Furthermore, in Saudi Arabian schools, teachers have faced dealing with such problems and assessing these students. Therefore, it will be a great achievement if, using IoT technology, they can be alerted to read abnormal vital signs using a decision support system prior to the occurrence of a situation that causes

hyperglycemia or hypoglycemia. Second, diabetic students experience difficulties in learning and demonstrating academic achievements more than non-diabetic learners [3]. They have problems with academic performance, participate less in social and dynamic activities, and are often less independent than non-diabetic students [6]. Another factor that can affect students' academic performance is irregular attendance. According to Holmes et al. [5], diabetic students demonstrate lower academic performance because they miss more days of school than their healthy peers.

Students' assessments traditionally depend on tests, assignments, and activity scores, and based on these scores, we measure their educational improvement; however, other factors, specifically for students with chronic diseases such as diabetes, are seldom considered.

Owing to the aforementioned issues and the lack of previous studies that address the issues of diabetic learners in classrooms, it would be more helpful if teachers and administration offices would be notified prior to the occurrence of crises such as hypoglycemia and hyperglycemia.

The proposed system is a new innovative smart environment to monitor and assess the achievements of diabetic students in Saudi Arabian schools. The targeted participants are teachers, administrators, and students with diabetes, the proposed system comprises IoT sensors to collect students' vital signs and all relevant information (concerning diabetes), to be wearable by a student either inside or outside the classroom. We will create an AI model, and we initially propose a decision tree (DT) algorithm because it is suitable for multiple decisions based on different situations. In the true negative case, the system will neglect these readings, but they will be saved for the future use and analysis. In terms of the false positive or false negative case, according to the precision percentage, an alert will be sent to users to check a student with a different color which indicates that there is a doubt. In the true positive case, an alert will be sent to teachers, healthcare providers in a school, and an administration office. The decision support system will enhance the assessment process by allowing teachers to visualize the students' attendance and scores.

2 Current Research on the Internet of Things in the Saudi Arabian Educational System and Related International Work

No previous study has been found that utilized IoT technology and sensors for diabetic students in the Saudi Arabian education system. Abed et al. [2] provided a study to examine the user acceptance toward IoT technology in Saudi Arabian universities and educational institutions using the Technology Acceptance Model (TAM). In addition, it demonstrated some of the practicalities of this technology and determined its potential for transition to IoT technology in the Saudi Arabian educational environments. This type of research targets helping and promoting Saudi Arabian educational institutions to utilize this technology. Using IoT at a university will assist teachers and students to improve communication, enhance the learning process, develop student experiences, and even save money (because IoT reduces the overuse of water and conditioning). The use of IoT technology on campus will improve classroom and campus environments, monitor safety and student health, and enhance student engagements.

Regarding the ease of use and the perceived usefulness, the researchers conclude that there is a strong consensus among the individuals on the future of the Internet in Saudi Arabia, and of the IoT in particular.

Owing to the lack of applications for ensuring the students' safety on school buses, Abbas et al. [1] proposed a safety sensor and tracking system for school buses in Saudi Arabia by utilizing GPS and passive infrared (PIR) sensors. The system allows the school to supervise the bus drivers and to follow up with the records of students' attendance. Furthermore, the efficient operation of the sensor and issuing of accurate alerts in the mobile application enables drivers to keep track of the students still in the bus. A total of 150 people used this system, and it was observed when evaluating the school bus safety tracking and sensor system that 65% of the users were satisfied.

Facchinetti et al. showed the importance of Continuous Glucose Monitoring (CGM) sensors in their application and how the real-time algorithms improve CGM sensors through decreasing the uncertainty and inexactitude and enhancing their ability to warn about decreasing or increasing blood glucose levels [4]. The smart CGM sensor includes a commercial CGM sensor that incorporates three real-time software units for noise reduction, improvement, and prediction, that was implemented on glucose control for the Dexcom SEVEN Plus monitor. They evaluated the execution of the CGM from data gathered at 2 experiments with 12 type 1 diabetic patients in each one. The results showed that the noise reduction unit improves the efficiency of the CGM series by an average of approximately 57%, the optimization module minimizes the absolute proportional difference from 15.1 to 10.3%, which raises the value pairs in the Clark error grid region A by 12.6%, and it could predict hyperglycemia and hypoglycemia events 14 min earlier.

3 The Unified Theory of Acceptance and Use of Technology Model (UTAUT)

The critical factors that affect the use of the system for diabetic students are based on a well-established theory of acceptance and use of technology, the Unified Theory of Acceptance and Use of Technology Model (UTAUT). The UTAUT model includes a total of four key constructs defined by [8] as follows.

- Performance expectancy (PE) is the level of individuals believing that using the system will benefit them in their job performances. It has a significant impact on adopting the use of the IoT and sensor based support systems. If users perceive that using the systems can improve their job performance, they are likely to be motivated to use the systems.
- Effort expectancy (EE) is the level of ease associated with the use of the system. Designing an easy to use system which meets users' requirements is likely to motivate using the system.
- Social influence (SI) is the level of individuals perceiving that important others believe they should use the new system. It is also described as the degree of social pressure from others to use the new system.
- Facilitating conditions (FC) represent the level of an individual believing that an organizational and technical infrastructure exists to support the use of the system, i.e., to what extent is the infrastructure available.

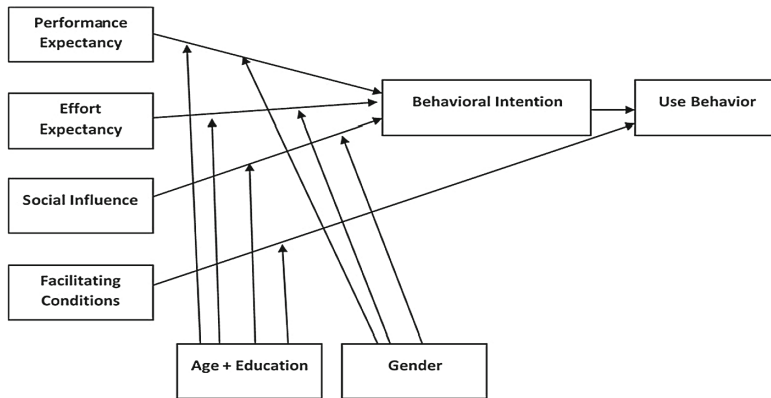


Fig. 1. Theoretical model

4 Future Work

4.1 Research Design

The research design is based on mixed methods comprising quantitative and qualitative research methods. The quantitative study will explore the diversity of specific behaviors or perceptions of adoption by diabetic students to assist in the educational process, by teachers, administration staff, and parents of diabetic children. In addition, it will explore the academic issues of diabetic students. In terms of the qualitative study, it will explore the perception of adoption of the system and the cognitive functioning issues from the viewpoint of diabetes specialists. This research is at the early stage, the proposed decision tree algorithm will be implemented and evaluated later.

5 Conclusion

The importance of the proposed system for monitoring and assessing diabetic learners was discussed, and examples of education related research on the IoT in Saudi Arabia and related work were presented. The UTAUT model will be used to investigate users' perception of the system adoption. Future work will involve the use of mixed-research methods to further develop and evaluate the system.

References

1. Abbas, S.A., Mohammed, H., Almalki, L., Hassan, M., Meccawy, M.: A safety tracking and sensing system for school buses in Saudi Arabia. *Period. Eng. Nat. Sci.* 7(2), 500–508 (2019)
2. Abed, S., Alyahya, N., Altameem, A.: IoT in education: its impacts and its future in Saudi universities and educational environments. In: Luhach, A.K., Kosa, J.A., Poonia, R.C., Gao, X.-Z., Singh, D. (eds.) *First International Conference on Sustainable Technologies for Computational Intelligence*. AISC, vol. 1045, pp. 47–62. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-0029-9_5

3. Dahlquist, G., Källén, B.: Swedish childhood diabetes study group: school performance in children with type 1 diabetes—a population-based register study. *Diabetologia* **50**(5), 957–964(2007)
4. Facchinetti, A., et al.: Real-time improvement of continuous glucose monitoring accuracy: the smart sensor concept. *Diabetes Care* **36**(4), 793–800 (2013)
5. Holmes, C.S., Fox, M.A., Cant, M.C., Lampert, N.L., Greer, T.: Disease and demographic risk factors for disrupted cognitive functioning in children with insulin-dependent diabetes mellitus (IDDM). *Sch. Psychol. Rev.* **28**(2), 215–227 (1999)
6. Shiu, S.: Issues in the education of students with chronic illness. *Int. J. Disabil. Dev. Educ.* **48**(3), 269–281 (2001)
7. Varni, J.W., Curtis, B.H., Abetz, L.N., Lasch, K.E., Pault, E.C., Zeytoonjian, A.A.: Content validity of the PedsQLTM 3.2 Diabetes Module in newly diagnosed patients with type 1 diabetes mellitus ages 8–45. *Qual. Life Res.* **22**(8), 2169–2181 (2012). <https://doi.org/10.1007/s11136-012-0339-8>
8. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: toward a unified view. *MIS Q.* **27**(3), 425–478 (2003)



Training Temporal and NLP Features via Extremely Randomised Trees for Educational Level Classification

Tahani Aljohani^(✉) and Alexandra I. Cristea^(✉)

Durham University, Durham, UK
{tahani.aljohani,alexandra.i.cristea}@durham.ac.uk

Abstract. Massive Open Online Courses (MOOCs) have become universal learning resources, and the COVID-19 pandemic is rendering these platforms even more necessary. These platforms also bring incredible diversity of learners in terms of their traits. A research area called Author Profiling (AP in general; here, Learner Profiling (LP)), is to identify such traits about learners, which is vital in MOOCs for, e.g., preventing plagiarism, or eligibility for course certification. Identifying a learner's trait in a MOOC is notoriously hard to do from textual content alone. We argue that to predict a learner's academic level, we need to also be using other features stemming from MOOC platforms, such as derived from learners' actions on the platform. In this study, we specifically examine *time stamps*, *quizzes*, and *discussions*. Our novel approach for the task achieves a *high accuracy* (90% in average) even with a simple shallow classifier, *irrespective of data size*, outperforming the state of the art.

Keywords: Learner Profiling · MOOC metadata · Data size · Decision trees

1 Introduction and Related Works

MOOCs attract tremendous numbers of users, due to their free cost, creating a rich diversity of user demographics - like age, gender, education level, etc. However, many face-to-face courses suddenly stopped during the current pandemic of COVID-19 [25], so the majority of new MOOC users this year are those who are trying to find replacements for their suspended classes [23] - making MOOCs an optimal alternative, as they offer classes from the world's top institutions [22]. According to a recent statistical report [23], enrollments at Coursera, a USA MOOC provider, have increased by 640% just between mid-March to mid-April 2020 (10.3 million in 30 d), compared with the same interval in 2019. Another example in the UK is FutureLearn, which has now 13.5 million users [10]. Due to these statistics, having personalised recommendations when delivering these courses to learners, based here on their demographics, becomes vital. Moreover, Learner Profiling (LP) is not only required during the current pandemic, but at all times, since demographic information is in demand for many types of

MOOC research. Although MOOC providers ask users to specify their demographic information during enrollment, the majority of users seem unaware of its value to their learning, and only about 10% fill it in [3]. The main motivation for this study is to offer thus an automatic method for MOOC researchers to extract users' demographics without relying on these, often incomplete, surveys. Specifically, a majority of users who benefit from MOOCs are education seekers. According to a Chinese study [27], investigating reasons behind student motivations in learning in MOOCs, 55% of the participants find MOOCs more interesting for receiving knowledge, 61% of the participants noticed that the repeatability of courses in MOOCs helps them understanding courses' content even deeper, 28% of the participants benefit from MOOC discussion forums for sharing knowledge, 27% of participants prefer MOOCs over other traditional modes of teaching, and 19% of the participants mentioned that the video lectures motivated their enrollment. One of the advantages of MOOCs is providing college credits, via a certificate. The first attempts started in October 2013, when a contract has been entered between Antioch University and Coursera, to license several of the University courses on the Coursera platform, as credits for part of a Bachelor's degree program. Also, in the same year, a course offered as a MOOC, "Innovation and Design Thinking", by the University of Cincinnati, was announced to provide credit for all students on Master's degree tracks [14]. The current pandemic promotes the demand for the online education in the future, as it breaks any spatial or temporal limitations. However, many obvious challenges appear in these platforms. Checking for plagiarism or authorship are some ways that increase trust in online education accrediting. Thus, our study is a step toward achieving such trust in MOOCs.

Natural Languages Processing (NLP) provides an approach for predicting user characteristics, called Author Profiling (AP). AP is data-driven computational linguistics that attempts to extract a user's attributes automatically, and is well-known as a challenging task in the NLP area. AP needs deeper linguistic analysis, typically with many training samples, because the hypothesis of AP is to explore similar linguistic patterns amongst authors who share the same demographics [5]. Moreover, works that have achieved state-of-the-art results in AP usually utilise a large number of linguistic features [20]. This complicates the AP task in practice. Also, online AP research in prior works mainly focused on social networks, and targeted few characteristics such as, gender, age, or native language [20]. Yet, other demographics, such as education level, and some important domains, like education, have received less attention from the online AP community [4,9]. In MOOCs, traditionally, 61% of the enrollments are education seekers [26], and the education level is well-known to influence learning in learning systems. As claimed by Kaati and his team [15], AP models that were trained on a content of a particular domain significantly underperform when applied to another domain, which means that AP models primarily rely on data used for training. What is more, content-based features usually used hundreds or even thousands of features for classification; ranging from lexical, semantical, to syntactical, and based on grammars, n-grams, frequencies,

token levels, etc. This should be very effective when two objectives are met: enormous text samples from a specific domain; and sufficient power of computational resources. When this is not the case, we propose that AP tasks can be also solved by other approaches, that is, by an in-depth examination of other potential features and metadata available in a specific domain. Regarding the used classifiers in the area, Support Vector Machines (SVM) are the most used for training textual features. Although deep learning models became state-of-art in the NLP field, especially the new generation of deep learning called Transformers, shallow classifiers have outperformed deep learning classifiers like the Bidirectional Encoder Representations from Transformers (BERT), or Long Short-Term Memory (LSTM), accordingly to recent AP studies [19]. Based on results obtained from an AP competition on predicting the gender of authors from their written texts, the best proposed technique was combining character n-grams, word n-grams, and function words, then trained them via an SVM classifier [19]. In this study, we address the learner profiling (LP) task, namely, predicting learners' level of education in MOOCs. The main contributions of this study are: i) we are the first to predict the educational status in MOOCs using NLP/ML approaches; ii) we investigate available MOOC metadata comprehensively for the task; iii) this is the first time the AP approach is linked with MOOC domain-related data, not only based on NLP features; iv) in spite of the simplicity of the applied features, we obtain a high accuracy regardless of data size, even with inexpensive classifiers.

2 Data Set

We have collected a large scale dataset [1, 2] from FutureLearn, extracted from 4 courses delivered by the University of Warwick from 2013 to 2015. These courses bring together different topic domains (Computer Science, Psychology, Literature). Each course has been offered multiple times (called 'runs'), with 21 runs in total, and are of different durations, as follows: Big Data (BG): three runs and nine weeks duration each. Babies in Mind (BM): six runs and four weeks duration each. The Mind is Flat (MF): seven runs and six weeks duration each. Shakespeare (SH): four runs and ten weeks duration each. In each week, learners learn a 'learning unit' that includes several tasks (called 'steps'), which can be a video, article, quiz, or discussion. The system generates a unique ID for each learner, and also timestamps which are: time of enrollment, time of submission of an answer, and time of accessing a step; The first time visiting a step (Visited), and when learners press the "Mark as Completed" button (Completed). The system also stores numerical and Boolean data related to learners' responses to different questions during a course. Learners in our data collection have accessed 2,794,578 steps. For our experiment, we have 12934 learners (who declared their level of education) out of the total of learners in our data set (245,255 learners), categorised as: Bachelor (B), Master (M), and Doctorate (D). We have collected the metadata from enrollments, quizzes, steps, and comments. Thus, we obtained very different data sizes, as there were different case scenarios of users' activities.

Table 1. Courses: BD, MB, MF, SH; levels: (B)achelor, (M)aster, (D)octor

Course	Enrollments			Quiz			Time spent			Comments		
	B	M	D	B	M	D	B	M	D	B	M	D
BD	870	737	160	5250	4860	1576	544	458	117	2326	2052	526
BM	1561	932	156	10065	6522	971	980	653	98	4650	2300	298
MF	2237	1424	269	48761	31015	6668	1249	836	187	9232	5844	2717
SH	2503	1747	388	136919	93311	22547	1802	1328	312	21363	14997	5887

For example, some users watched videos but did not answer quizzes, some wrote comments while others did not, and so on, see Table 1. However, we fixed this issue by filling in missing data, as will be explain in Sect. 3.5. The size and richness of this data arguably allows for generalisability of our study.

3 Methodology

3.1 Feature Extraction

We have, comprehensively, studied potential features that can be extracted from our rich MOOC data, and can contribute to the level of education prediction. This feature extraction process was based on three conditions:

1. **Existence of Labels.** Features should belong to learners who have declared their education level. This is essential because our study is basically based on supervised learning techniques.
2. **Size of Feature’s Samples.** Some metadata are available in our dataset, but they do not meet the current condition. For example, the time at which a comment was moderated for inappropriate or offensive content, is available in our data; however, when we tried to extract it, we found that it reflected upon only three learners; which is not adequate for the training process.
3. **Relatedness.** Some available metadata in our dataset has not been extracted, such as question number or comment ID, since they obviously are not predictors for our task.

As a result, our extracted features can be classified into four categories:

- **Enrollment Features.** We extracted date of enrollment for each learner (enrolled-at [timestamp] – when the learner enrolled).
- **Quiz Features.** We extracted date of submitting answers (submitted-at [timestamp]), responses data (which is the answer number selected, reflecting their ordered position [numerical]), and correctness data (for the correctness of the responses [true or false]).
- **Time Spent Features.** We extracted two types of dates related to steps: first visited-at (when the step was first viewed by the user[timestamp]), and last-completed-at (when the step was last marked as complete by the user[timestamp]).

- **Comment Features.** We extracted comments written by a learner ([text]), date of post comments ([timestamp]), and number of likes attributed to each comment.

3.2 Feature Engineering

All the extracted features are in raw format, so we normalised these features before feeding them into machine learning models. For example, we removed URLs, since URLs have a standard structure that is not influenced by a user’s writing style. Also, we dropped any duplicated comments and kept only the first comments (the original comment written by a learner). This was because we have found some learners copy and paste other learners’ comments, which meant that these copied comments were not written in their own personal writing style. In addition, we applied simple and advanced NLP techniques to the comments to convert them into textual representations that are commonly utilised for AP. All features have been converted to numerical forms, as follows:

1. Temporal Features (5 Feature Sets):

- Hour:* value of time hour within a day (values between 0 to 23).
- Month:* value of that month within a year (values between 1 to 12).
- Week Day:* value of that day within a week (values between 1 to 7).
- Month Day:* value of that day within a month (values between 1 to 31).
- Year Day:* value of that day within a year (values between 1 to 365).

See Table 2 for temporal features symbols.

2. Simple Textual Features (9 Feature Sets):

- Character Count:* Total number of characters in a comment.
- Word Count:* Total number of words in a comment.
- Word Density:* Average length of words in a comment.
- Sentence Count:* Total number of sentences in a complete comment.
- Sentence Density:* Average length of a sentences in a complete comment.
- Punctuation Count:* Total number of punctuation marks in a comment.
- Upper Case Count:* Total number of upper count words in a comment.
- Title Word Count:* Total number of proper case (title) words in a comment.
- Stopword Count:* Total number of stop words.

3. Advanced NLP Features (2 Feature Sets):

The advanced NLP features are extracted by pythonic NLP libraries:

- *Part of Speech (POS):* To extract the part of speech tags [24], we used the standard Textblob library. Then, we have calculated the total number of nouns, verbs, adjectives, adverbs, and pronouns in each comment. See Table 2 for Tag symbols.
- *Sentiment Analysis (SA):* To extract the SA polarity [18], we used the standard NLTK which assigns three polarities: positive (1), negative (-1), and neutral(0).

4. Time Spent Feature:

This is computed via the difference between the time when the a learner has fully completed the step (C), and the first time that learner visited that step (V), in seconds:

$$TimeSpent = C - V \quad (1)$$

Table 2. Description of POS and temporal features symbols in our study

Name	Symbol
Hour	hour[enrolment(e_hour), quiz(q_hour), comment(c_hour)]
Month	month[enrolment(e_month), quiz(q_month), comment(c_month)]
Week day	week_day[enrolment(e_week_day), quiz(q_week_day), comment(c_week_day)]
Month day	month_day[enrolment(e_month_day), quiz(q_month_day), comment(c_month_day)]
Year day	year_day[enrolment(e_year_day), quiz(q_year_day), comment(c_year_day)]
Noun	noun_count['NN', 'NNS', 'NNP', 'NPS']
Verb	verb_count['VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ']
Adjective	adj_count['JJ', 'JJR', 'JJS']
Adverb	adv_count['RB', 'RBR', 'RBS', 'WRB']
Pronoun	pron_count['PRP', 'PRP\$', 'WP', 'WP\$']

3.3 Models

One of our study objectives is to consider less expensive computational classifiers rather than expensive and complex models like deep learning algorithms. This is practically possible since our approach has included a feature engineering step. We have trained our labeled examples on many different supervised shallow learning algorithms. We have employed models that have been commonly used in the AP area: Support Vector Machine, Naïve Bayes, Decision Trees, Random Forests, Logistic Regression, Multilayer Perceptron, and K-Nearest Neighbors [4]. We are presenting in this paper only results of the top performing model, which is the Decision Trees model, particularly, the Extra Trees (ET) Classifier; a decision tree-based classifier that learns in an ensemble way, which is standing for Extremely Randomised Trees. This algorithm is fundamentally an ensemble of decision trees, similar to other DT-based models, such as the random forest. However, ET is built by more unpruned decision trees, more than random forest, and the prediction is based on majority voting if the task is a classification [11]. The advantage of this algorithm is it fits every single decision tree to the whole training dataset, inside of a bootstrap sample of the training dataset, which is the case in the random forest. This creates more robust and better generalisation performance [11]. The maximum size of tree depth in ET, by default, is none, which means trees keeping expanding till all nodes are pure - see ET Algorithm 1.

3.4 Baseline Models

For comparison purpose, we employed three baseline models that are commonly used for text classification tasks (texts are comments of learners), which are:

Algorithm 1: Extremely Randomised Trees Algorithm Procedure

```

1 begin
2   Split a node(S):
3   - inputs are the local learning subset (S) corresponding to the node we want to split
4   - output: a split [ $a < a_c$ ] or nothing; (a= attribute)
5   if Stop split(S) is False, then
6     -Select K attributes:  $a_1, \dots, a_K$  among all non constant (in S) candidate attributes
7     -Draw K splits  $s_1, \dots, s_K$ , where  $s_i = \text{Pick a random split}(S, a_i), \forall i = 1, \dots, K$ 
8     -Return a split  $s^*$  such that  $\text{Score}(s^*, S) = \max_{i=1, \dots, K} \text{Score}(s_i, S)$ 
9     else
10      | return nothing
11    end
12  end
13  begin
14    Pick a random split(S,a)
15    - Inputs: a subset S and an attribute a
16    - Output: a split.
17    -Let  $a_{max}^S$  and  $a_{min}^S$  denote the maximal and minimal value of a in S;
18    -Draw a random cut-point  $a_c$  uniformly in [ $a_{min}^S, a_{max}^S$ ]
19    - Return the split [ $a < a_c$ ]
20    begin
21      Stop split(S):
22      - Inputs: a subset S
23      - Output: a boolean
24      if  $|S| < n_{min}$ , then
25        | return TRUE;
26        if all attributes are constant in S: then
27          | return TRUE;
28        end
29        if the output is constant in S: then
30          | return TRUE;
31        end
32        else
33          | return FALSE
34        end
35      end
36    end
37  end
38 end

```

- **Term Frequency-Inverse Document Frequency (TF-IDF):** Simple and old-fashioned, but also NLP state-of-the-art. It is a lexically-dependent, but semantically independent technique. For our study, we applied both character n-grams ($n = 3,6$) and word n-grams ($n = 1,2$), which are the best performing n-grams settings employed for AP in the PAN evaluation campaign [6]. The next equation explains a standard TF-IDF technique mathematically:

$$TF-IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (2)$$

- **Word2vec:** First neural network-based modeling approach in NLP [17], which is a semantics-dependent, but context-independent embedding. We used the skip-gram-600 model (one of the word2vec algorithms), which has two layers of shallow neural networks. It consists of average word vectors, that are built based on training on a corpus of 50-million tweets [12]. In the skip-gram model, the conditional probability P is calculated for context words w_o

and for a central (target) word w_c , by a softmax operation on the vector v inner product.

- **Bidirectional Encoder Representations from Transformers (BERT):** A transformer model, which is the cutting edge language model in NLP nowadays. It is a context-dependent embedding. BERT is a complex neural network architecture, and its large version includes: 24 layers, 1024 hidden states, 16 heads, 340M parameters [8].

3.5 Data Normalisation

After merging the features for each learner, we noticed some missing values, because not all learners have done all activities in each step. So, by using a module from Scikit-Learn (SimpleImputer [21]), we filled the missed values by adding the average value of each feature. This step is important for creating vectors with fixed lengths for machine learning classifiers. Also, our data are not in balance, so we balanced them via the popular Synthetic Minority Oversampling Technique (SMOTE) [7]. Next, we split our data set into training (80%), and testing (20%). We further shuffled and stratified for better learning performance [16]. Finally, we examined the extracted features individually, and as combinations. Figure 1 represents the general workflow of our experiments, visually.

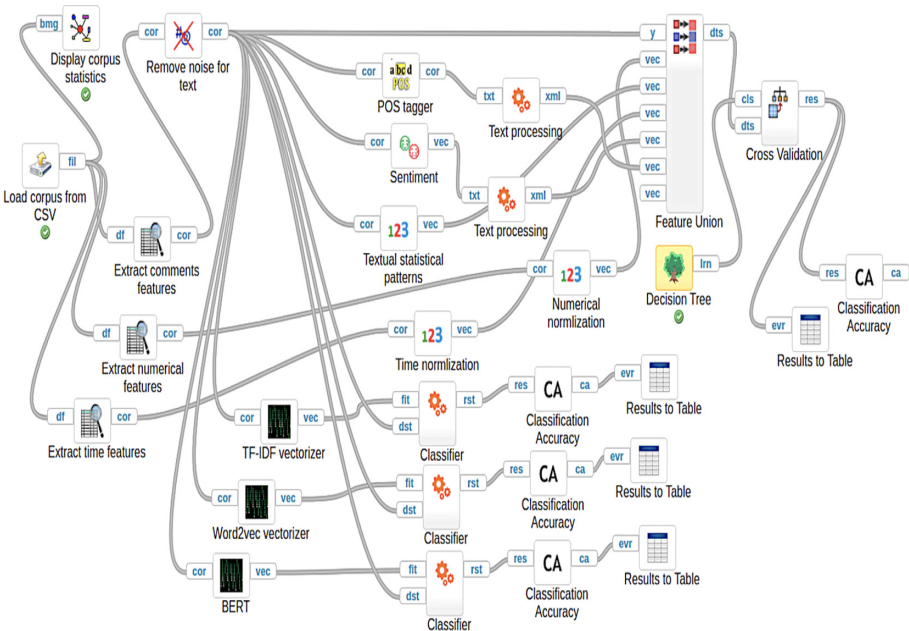


Fig. 1. General workflow of our level of education prediction approach in MOOC

4 Results and Discussion

Firstly, we applied the baseline models to predict the level of education of learners only based on comments (traditional way of solving NLP tasks, in general). We found that using comments alone, based on these models, did not provide satisfactory results. This could be because comments in our data were dominated by course context, which is more representative of courses rather than learners. This may have affected the performance of these NLP state-of-the-art algorithms in our study. BERT’s performance in classifying the learners was the lowest. Word2vec performed better than BERT and TF-IDF at character-level, but similar to TF-IDF at word-level. Next, we examined the extracted features in terms of their performance. Despite the simplicity of textual representations that we applied in our experiments, they performed significantly better than the text representations via BERT, TF-IDF, or Word2vec. This supports our initial assumption that using simple and basic textual features could solve our research problem.

Table 3. Overall accuracy per feature category and course, in addition to baseline models

Approach	BD	BM	MF	SH	Average
TF-IDF (char)	0.75	0.78	0.62	0.68	0.71
TF-IDF (word)	0.80	0.84	0.75	0.66	0.76
Word2vec	0.76	0.83	0.75	0.68	0.76
BERT	0.76	0.87	0.80	0.81	0.81
Enrollment + ET	0.67	0.78	0.74	0.68	0.72
Comment + ET	0.85	0.94	0.90	0.88	0.89
Quiz + ET	0.84	0.89	0.84	0.72	0.83
Time Spent + ET	0.82	0.85	0.87	0.84	0.85
Time Spent + Comment + ET	0.87	0.91	0.84	0.97	0.90

Furthermore, we found that MOOC metadata also outperforms baseline models, except for enrollment features. Time-spent features, as well as comment features both achieved highest accuracies, thus we combined them, and this combination obtained the best accuracy compared to all models and settings in our experiments. With respect to machine learning models, the Extra Trees (ET) achieved highest performance for all of our experimental settings, so we are discussing in this paper only results obtained by the ET classifier - see Table 3, which reports ET overall accuracy per course and per feature category. These results are validated by using 10-fold Cross-Validation (CV), which is well known to be used for avoiding the over-fitting issue [13]. In each iteration (k), a single accuracy is estimated, then all accuracies are averaged to get the final accuracy (A). 10-Fold CV is given by the following formula (k -Fold CV accuracy; $k = 10$):

$$Accuracy = \frac{1}{k} \sum_{i=1}^k A_i \quad (3)$$

Finally, we evaluated the best obtained results, after combining time spent and textual features, comprehensively and realistically. So, we applied three popular performance measurements: F1-score, precision, and recall. This is an important evaluation step, since our data is not balanced, and it is necessary to not only consider overall accuracy results, which could be strongly biased. We reported results of these three evaluation measurements per category, allowing clear exposure of minority classes, see Table 4).

Table 4. Detailed results (F1, precision, and recall) per course/class, and based on time spent and comment combined features

Course	Acc.	Precision			Recall			F1-score		
		B	M	D	B	M	D	B	M	D
BD	0.87	0.86	0.87	0.90	0.91	0.86	0.76	0.89	0.87	0.82
BM	0.91	0.91	0.91	0.91	0.95	0.87	0.79	0.93	0.89	0.84
MF	0.84	0.85	0.87	0.72	0.91	0.76	0.76	0.88	0.81	0.74
SH	0.97	0.97	0.96	0.98	0.97	0.96	0.95	0.97	0.96	0.96
Average	0.8975	0.8975	0.9025	0.8775	0.935	0.8625	0.815	0.9175	0.8825	0.84

5 Conclusion

We solve our Learner Profiling (LP) text classification problem, even though presented in a domain with weak textual representation about authors (MOOCs), with very simple metadata available in the domain. Our new proposed LP model doesn't only obtain high performance, but also shows that this task can be performed via inexpensive computational algorithms, regardless of data size. Our results demonstrate that the selected features are so representative that they work well even with extremely unbalanced data. We also show that using state of the art NLP models is not supportive enough for what is supposed to be mainly a text classification task, due to the domain conditions. For future work, we plan to experiment with other traits of LP in MOOCs, due to the specific domain-related challenges it poses.

Acknowledgment. We gratefully acknowledge funding support from the Ministry of Education of Saudi Arabia for this research.

References

1. Aljohani, T., Cristea, A.I.: Predicting learners' demographics characteristics: Deep learning ensemble architecture for learners' characteristics prediction in moocs. In: Proceedings of the 2019 4th International Conference on Information and Education Innovations, pp. 23–27 (2019)

2. Aljohani, T., Pereira, F.D., Cristea, A.I., Oliveira, E.: Prediction of users' professional profile in MOOCs only by utilising learners' written texts. In: Kumar, V., Troussas, C. (eds.) ITS 2020. LNCS, vol. 12149, pp. 163–173. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_20
3. Aljohani, T., Yu, J., Cristea, A.I.: Author profiling: prediction of learners' gender on a mooc platform based on learners' comments. *Int. J. Comput. Inf. Eng.* **14**(1), 29–36 (2020)
4. Alroobaea, R.: An empirical combination of machine learning models to enhance author profiling performance. *Int. J. Adv. Trends Comput. Sci. Eng.* **9**, 2130–2137 (2020). <https://doi.org/10.30534/ijatcse/2020/187922020>
5. Argamon, S., Koppel, M., Pennebaker, J.W., Schler, J.: Automatically profiling the author of an anonymous text. *Commun. ACM* **52**(2), 119–123 (2009)
6. Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., Nissim, M.: N-gram: new groningen author-profiling model. CoRR abs/1707.03764 (2017). <http://arxiv.org/abs/1707.03764>
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *J. Artif. Int. Res.* **16**(1), 321–357 (2002)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of ACL, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (June 2019). <https://doi.org/10.18653/v1/N19-1423>, <https://www.aclweb.org/anthology/N19-1423>
9. Estival, D., Gaustad, T., Hutchinson, B., Pham, S., Radford, W.: Author profiling for English and Arabic emails. In: Association for Computational Linguistics (2007)
10. FutureLearn: Online courses and degrees from top universities. <https://www.futurelearn.com/>
11. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. *Mach. Learn.* **63**, 3–42 (2006). <https://doi.org/10.1007/s10994-006-6226-1>
12. Hsieh, F., Dias, R., Paraboni, I.: Author profiling from Facebook corpora. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018). <https://www.aclweb.org/anthology/L18-1407>
13. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning. STS, vol. 103. Springer, New York (2013). <https://doi.org/10.1007/978-1-4614-7138-7>
14. Jaschik, S.: Moocs for credit. *Inside Higher Ed* **23**, (2013)
15. Kaati, L., Lundeqvist, E., Shrestha, A., Svensson, M.: Author profiling in the wild. In: 2017 European Intelligence and Security Informatics Conference (EISIC), pp. 155–158 (2017). <https://doi.org/10.1109/EISIC.2017.32>
16. Khamsan, M., Maskat, R.: Handling highly imbalanced output class label. *Malaysian J. Comput.* **4**(2), 304 (2019). <https://doi.org/10.24191/mjoc.v4i2.7021>
17. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. vol. [arXiv:1301.3781](https://arxiv.org/abs/1301.3781) (2013)
18. NLTK: nltk.sentimentpackage — nltk 3.5 documentation (2020). <https://www.nltk.org/api/nltk.sentiment.html#module-nltk.sentiment>
19. Pardo, F.M.R., Rosso, P., Charfi, A., Zaghouni, W., Ghanem, B., Sánchez-Junquera, J.: Overview of the track on author profiling and deception detection in Arabic. In: FIRE (2019)
20. Pardo, F.M.R., Rosso, P.: Overview of the 7th author profiling task at pan 2019: Bots and gender profiling in twitter. In: CLEF. Psychology, Computer Science (2019)

21. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
22. Robson, D.: Online learning: how to acquire new skills during lockdown. *The Guardian* (2020). <https://www.theguardian.com/education/2020/apr/19/online-learning-how-to-acquire-new-skills-during-lockdown>
23. Shah, D.: Highlights from coursera partners conference 2020. *The Report by Class Central* (2020). <https://www.classcentral.com/report/coursera-conference-2020-highlights/>
24. TextBlob: Textblob: Simplified text processing (2020). <https://textblob.readthedocs.io/en/dev/>
25. WHO: Coronavirus, world health organization (2020). <https://www.who.int/health-topics/coronavirus#tab=tab1>
26. Zhenghao, C., Alcorn, B., Christensen, G., Eriksson, N., Koller, D., Emanuel, E.J.: Who’s benefiting from MOOCs, and why. *Harvard Business Review* (2015). <https://hbr.org/2015/09/whos-benefiting-from-moocs-and-why>
27. Bayeck, R.: Exploratory study of MOOC learners’ demographics and motivation: The case of students involved in groups. *Open Praxis* **8**(3), 223–233 (2016)



Urgency Analysis of Learners' Comments: An Automated Intervention Priority Model for MOOC

Laila Alrajhi¹(✉), Ahmed Alamri¹, Filipe Dwan Pereira², and Alexandra I. Cristea¹

¹ Computer Science, Durham University, Durham, UK
{laila.m.alrajhi,ahmed.s.alamri,
alexandra.i.cristea}@durham.ac.uk

² Computer Science, Federal University of Roraima, Boa Vista, Brazil
filipe.dwan@ufrr.br

Abstract. Recently, the growing number of learners in Massive Open Online Course (MOOC) environments generate a vast amount of online comments via social interactions, general discussions, expressing feelings or asking for help. Concomitantly, *learner dropout*, at any time during MOOC courses, is very high, whilst the number of learners completing (*completers*) is low. Urgent intervention and attention may alleviate this problem. Analysing and mining learner comments is a fundamental step towards understanding their need for intervention from instructors. Here, we explore a dataset from a FutureLearn MOOC course. We find that (1) learners who write many comments that need urgent intervention tend to write many comments, in general. (2) The motivation to access more steps (i.e., learning resources) is higher in learners without many comments needing intervention, than that of learners needing intervention. (3) Learners who have many comments that need intervention are less likely to complete the course (13%). Therefore, we propose a *new priority model for the urgency of intervention* built on learner histories – past urgency, sentiment analysis and step access.

Keywords: MOOCs · FutureLearn · Comments · Priority in intervention

1 Introduction

Today, with the successful development of MOOC environments, they are playing a vital role in education. In an online world, learners can access knowledge and numerous high-quality resources [1]. This attracts a large learner cohort with different abilities. At the same time, the dropout rate is high enough to be a serious problem. There are many reasons for dropping out, including *learners' need for instructor intervention* [2].

MOOC platforms have an asynchronous discussion forum tool that provides a venue for learners to communicate with others [3]. It is a crucial component and can be utilised in different ways, involving social interaction, discussion, or as an essential part of a teaching strategy [4]. Also, it is the main communication tool between learner and instructor [5] for feedback, support, and encouragement [6].

Instructors' interventions are an essential teaching activity in MOOCs, to help learners [7]. However, due to the high ratio of learners-to-instructors, it is very hard to monitor all learners' comments. Thus, the problem of detecting urgent posts has stirred researchers to solutions primarily framed towards a text classification problem [8–10]. However, such approaches did not consider the study of the learner's behaviour.

We conjecture it as essential to understand learners' behaviours before proposing intervention. Hence, after analysing the distribution of comments that need intervention, we additionally explore the relation between high-frequency commenters and their behaviours, in terms of their access and completion rates. We define *high-frequency (HF) commenters* as learners who have many comments that need intervention, and formulate the following research questions:

RQ1: What is the behaviour of learners who need an urgent intervention?

- *RQ1.1: Is there a relationship between the number of comments written by the learners that need urgent intervention and the average number of comments?*
- *RQ1.2: Is there a relation between high-frequency (HF) commenters and their number of steps accessed?*
- *RQ1.3: Is there a relation between HF commenter number and completion-rates?*

RQ2: Can we design an effective intervention priority framework based on behaviour?

2 Related Work

Before the era of MOOCs, researchers were already analysing the need of *instructor intervention* in discussion forums in asynchronous virtual learning environments [11]. Recently, instructor intervention is one of the hot research directions for MOOCs [12]. The most common approaches focused on the use of text classification methods [8–10, 13]. Some were based only on Natural Language Processing (NLP), others involved other features. They deployed different types of machine learning algorithms (shallow and deep neural networks models). Other relevant attempts [14–16] predicted urgency as one of three different tasks (confusion, sentiment and urgency), but they also involved only text-based methods.

In [17], the instructor intervention problem in MOOC forums was tackled by using the sequence of posts and combined features from these posts. They considered instructor posts as intervention. Chandrasekaran et al. [18] proposed several studies on instructor intervention in Coursera forums. For instance, [19] proposed a taxonomy of pedagogical interventions for automated guidance to instructors. Moreover, [20] investigated discourse relations and used PDTB (Penn Discourse Treebank) based features to predict the need for instructor interventions. For position bias in intervention context [5] they showed that there is a bias in instructor intervention. They improved intervention classifier performance when they removed bias from the training data. In [3] they studied instructor intervention based on a deep learning model, and thread structure.

While these works provide solutions to the instructor's intervention problem, learner behaviours' relation to urgent intervention need remains unstudied. Specifically, we want

to analyse how HF commenters behave on MOOC platforms. The main idea in this paper is enhancing intervention by prioritising it as an intelligent filtering system. This priority is generated based on learner behaviour. To the best of our knowledge, the priority in intervention shown in this paper has not been seen before in the literature.

3 Methodology

This section presents our dataset and methodologies.

3.1 Dataset

The raw corpus dataset we utilised was provided by the FutureLearn platform [21], namely, the ‘Big Data’ course, Run 2. The course was conducted during 2016 on an over 9 weeks scale and, a.o., it contains English comments text. We then focus only on the first half of the course (5 weeks), with its subset of 5790 comments. This is done as early intervention on urgent comments is considered more appropriate than late intervention, as most learners tend to drop out in the first stages of the course [22, 23].

Gold Standard Corpus Creation. The collected 5790 text comments were manually labelled to assign urgency and they were annotated by domain experts. From these experts, two are instructors at the Department of Computer Science at the University; in addition, one is an author of this paper. We gave Agrawal et al. [24] instructions to annotators, to manually classify comments onto the urgency scale (1–7), (1: no reason to read the post – 7: extremely urgent: instructor definitely needs to reply). After completing the annotations, we excluded (four) comments containing anything other than (1–7). To validate these labels we used Krippendorff’s α [25]. However, we found that the agreement between these annotators was very low. To alleviate this, we converted the scale to binary (1 to 3 \rightarrow 0, 4 to 7 \rightarrow 1). Then, we applied a voting process between the three annotators, resulting in a binary-class label as: 0 \rightarrow Non-Urgent; 1 \rightarrow Urgent.

As this is real data, possibly unsurprisingly, the resulting data is biased towards the (Non-Urgent) class, with 883 comments as Urgent (15%) and the rest as Non-Urgent.

Dataset Statistics. The 5786 comments were created by 873 unique learners (commenters) in 5 weeks. Number of steps and comments per week appear in Table 1.

Figure 1 (left) illustrates the number of comments written over 5 weeks. This number decreased gradually, dropping to 180 comments in the last week, from 2130 comments in the first week (–99.9%). Every week has a different number of steps to complete. Thus, we also represented the number of comments per steps (Fig. 1, right) on the temporal axis. These numbers oscillate more – showing some topics to trigger more comments than others – although the overall numbers follow the downwards trend.

Who is, however, writing these comments? To inspect the distribution of the number of, what we call, *active* learners (commenters) who wrote the comments every week and step, we visualised them as shown in Fig. 2.

Next, we observed comments that need urgent intervention, to focus on their trend. Hence, we visualised a line graph over the 5 weeks, to explore how urgency changed over time (Fig. 3, left). Overall, the first weeks had a higher percentage of comments needing

Table 1. Statistics of the gold standard corpus.

Week	# of steps (week)	# of comments (week)	# of active learners (week)	Average comments per learner (week)
1	11	2130	749	2.84
2	12	1600	419	3.81
3	15	1123	236	4.75
4	11	753	180	4.18
5	4	180	92	1.95

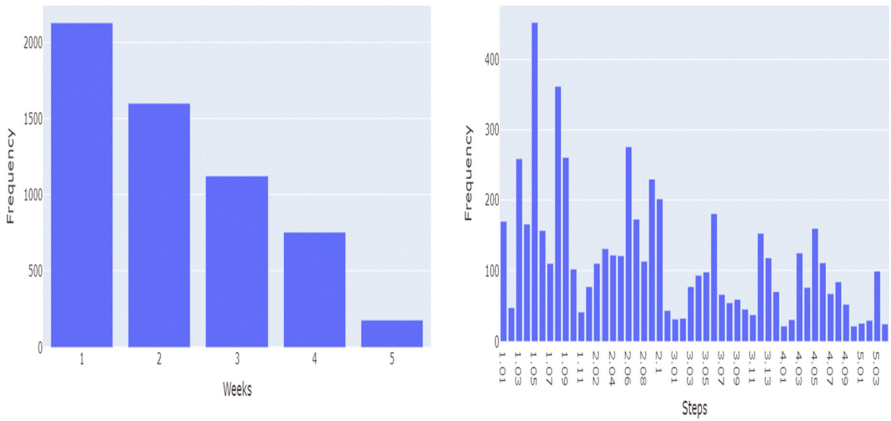


Fig. 1. The number of comments in every week (left) and in every step (right).

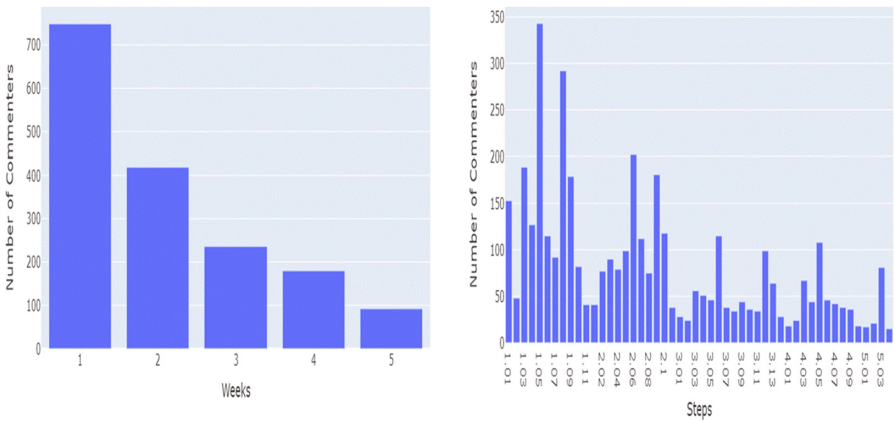


Fig. 2. Active learners (commenters) in every week (left) and in every step (right).

intervention (Fig. 3, left), drawn from a higher number of comments (Fig. 1, left). The fluctuation from week 4 to 5 is due to the drastic drop in overall comments. We also visualised percentages of urgent comments for every step, (Fig. 3, right), which showed high fluctuation. We further graphically compared results between Urgent and Non-Urgent comments number across (weeks, steps) in Fig. 4 (left, right).

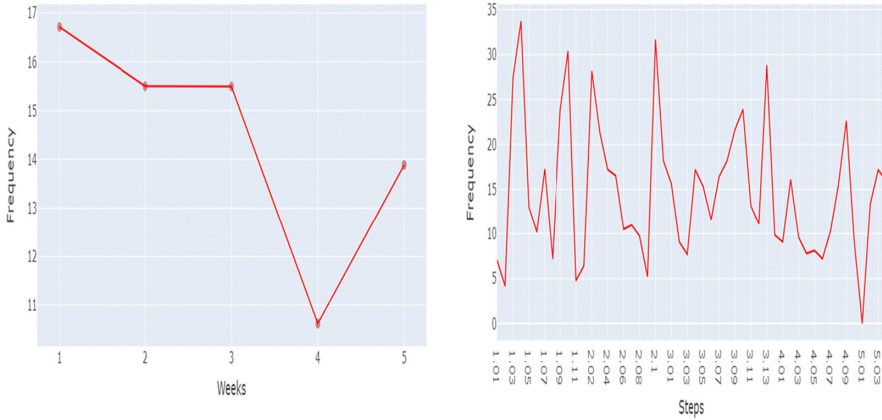


Fig. 3. The percentage of urgent comments for every week (left) and for every step (right).

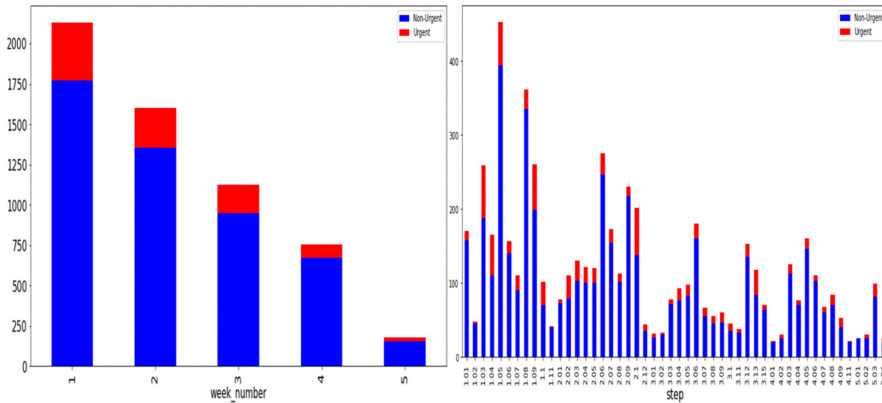


Fig. 4. Comparing Urgent and Non-urgent comment numbers for every week (left) and every step (right).

3.2 Exploring Urgency and Learner Behaviour

As an initial step, to understand learners' behaviour in writing comments, we explored the relationship between the number of comments written by the learners who need

urgent intervention with the average number of comments. Then, to explore the effect of urgency on learner behaviour, we explored the relationship between HF commenters and their learning behaviour – here we simply compared it to the number of step accesses. We defined a learner who needs urgent intervention (*HF commenters*) as per Eq. 1; let n : number of comments, $u(c)$: urgent comments and c : a comment.

$$HF\ Commenters = \frac{\sum_{n=1}^{\infty} u(c)}{\sum_{n=1}^{\infty} (c)} = 1 \quad (1)$$

We calculated the average number of step access for each group (Non-Urgent) and HF commenters (Urgent) to track how every group behaves on the platform.

Finally, we addressed completers with respect to their need of intervention. We defined completers according to Eq. 2; where, *total access steps*: number of total access per learner, *total course steps*: total number of steps in a course.

$$Completer = total\ access\ steps \geq total\ course\ steps * 0.80 \quad (2)$$

We define completers as in Eq. 2 because, in spite of the large number of previous studies, a formal definition of learners dropout is lacking [26]. Therefore, we went with the definition in [23], we defined completers are learners whose number of steps accessed is equal or higher than 80%.

3.3 Priority in Urgent Intervention

In this study we propose a new intervention framework designed to add prioritising to urgent comments based on learners' history, to assist instructors' decision, optimise their time and ability to adapt their intervention. We begin by supposing that, when the instructor intervened, some of these comments were potentially urgent. Then, for these potentially urgent instances we add *priority* (high-, mid- or low), depending on the *learner risk level*. The idea is to focus on learners, understand their behaviours and do a segmentation based on 3 variables (urgency, sentiment analysis and number of accesses).

Our model includes two phases (see Fig. 5), first phase (prediction phase): using a supervised classifier to predict if the comments need a response urgently or not. Second phase (intervention priority phase): takes the output of the previous phase (urgent comments) as input. Then, adds a priority to these comments based on the history of learners who wrote these comments using unsupervised machine learning (clustering). Therefore, based on these groups we assign different priorities to comments.

Prediction Phase. Here we apply the state-of-the-art in text classification, Bidirectional Encoder Representations from Transformers (BERT) [27] to predict urgency.

Intervention Priority Phase. We study the behaviour of learners based on three variables (urgency, sentiment analysis and step access). We selected these three variables because they address RQ1.2 and RQ1.3. Moreover, a sentiment analysis study [13] found a negative correlation between urgency and sentiment analysis; meaning urgent comments correlate with negative sentiments. The processing was as follows:

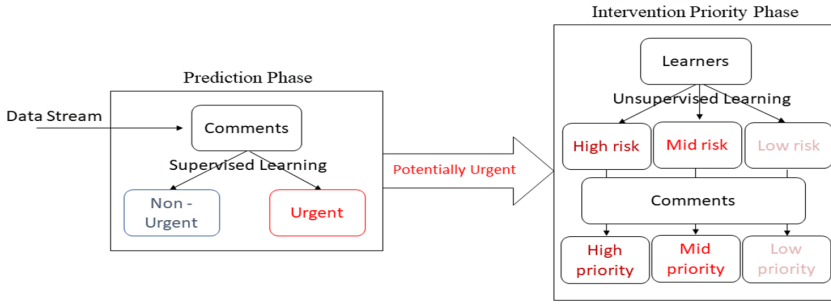


Fig. 5. Priority in urgent intervention framework.

Urgency. To find the learners for whom most of their comments need intervention, we calculate the number of urgent comments for each learners. After that, we clustered in an unsupervised manner all the learners into three groups, by assigning each learner based on the number of urgent comments, to a specific cluster.

Sentiment Analysis. We analysed every comment to extract sentiment polarity into three categories (positive, negative, and neutral) sentiment using the VADER tool. We selected this tool because it is a well-known tool and some researches proved that VADER outperforms Text Blob in social media [28, 29]. Then we found the overall average value of sentiments for each learner and created sentiment clusters, low sentiment number indicating high-risk learners.

Steps Access. For each learner, we calculated the number of step accesses. Then we clustered learners into three groups, based on these values. A high step access number is an important indicator of learning activity, possibly connected to high motivation.

For every variable (urgency, sentiment analysis and step access), we clustered all learners into three groups, by applying natural breaks optimisation with the Fisher Jenks algorithm [30] as it works on one dimensional data. Therefore, every learner has three scores that represent the three clusters’ variables (urgency, sentiment analysis and steps access). We calculated an overall score for every learner as in Eq. 3.

$$Overall_{score} = urgency_{cluster-score} + sentiment\ Analysis_{cluster-score} + step\ Access_{cluster-score} \quad (3)$$

Thus, the overall score will be between (0–6). Then, we mapped the overall score onto different levels of risks: Higher than 3 → High risk; Higher than 1 → Mid risk; Others → Low risk. Then, we segmented learners as below:

- *High risk:* learners who have high overall score from three variables (urgency, sentiment analysis and access steps).
- *Mid risk:* learners who have middle overall score from three variables.
- *Low risk:* learners for whom overall score from three variables is low.

Based on these levels of risks we computed the priority to the intervention for all potentially urgent comments – see Algorithm 1.

Algorithm 1. Priority of Intervention (C, U, S, M)**Input:**

- i) C: Stream of potentially urgent comment instances.
- ii) U: Number of urgent comments for each learner.
- iii) S: Average value of comments' sentiment for each learner.
- iv) M: Number of steps access for each learner.

Output:

- i) Urgent comments with the priority intervention results.

Method:

Build 3 learner clusters for Urgency.

Build 3 learner clusters for Sentiment Analysis.

Build 3 learner clusters for Steps Access.

Compute the Overall Score.

if Overall Score is higher than 3 **then** High risk learner.

Urgent comment = high priority intervention.

else if Overall Score is higher than 1 **then** Mid risk learner.

Urgent comment = mid priority intervention.

else

Low risk learner. Urgent comment = low priority intervention.

end if

End Algorithm

4 Results and Discussions

RQ1.1: Is there a relationship between the number of comments written by the learners that need urgent intervention and the average number of comments?

To inspect learners' writing behaviour, we transformed an average number of comments into an urgency bar chart (1 urgent comment, 2 urgent comments, etc.), as shown in Fig. 6. Interestingly, we observed that, usually (but not always), if a learner writes more comments that need intervention, they tend to write more comments in total. This is useful in that they do not 'give up' and present longer time to be 'dealt with'.

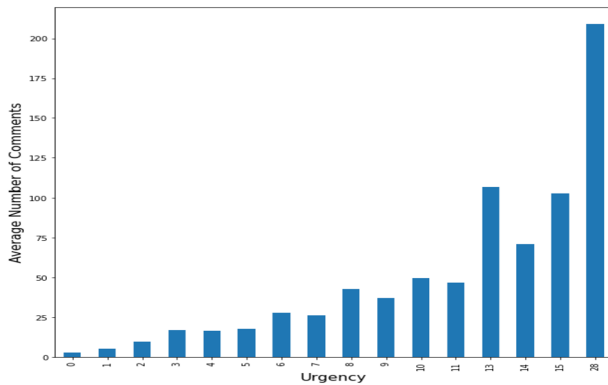


Fig. 6. Relation between urgent comments (urgency) and average number of comments.

RQ1.2: Is there a relation between high-frequency (HF) commenters and their number of steps accessed?

As (Fig. 7, left) shows, we calculated the average number of steps accessed for the HF learners or Urgent and Non-Urgent group. We found that, in general, both groups access learning materials, but the average number of steps access in the Urgent group was lower (33 steps). This difference is statistically significant (Mann-Whitney U test: $p < 0.05$). Consequently, the key observation indicates more learning activity and thus potentially increased motivation for learners with comments not needing intervention.

RQ1.3: Is there a relation between HF commenter number and completion-rates?

The result of the relation between urgency and completion is shown in (Fig. 7, right). As we can see, HF learners who require urgent intervention are less likely to complete the course only (13%). This difference is statistically significant (Mann-Whitney U test: $p < 0.05$). From this result, we conclude that learners who need intervention tend not to complete the course. We think this is one of the reasons for the high dropout rate. This confirms the need for intervention for urgent comments.

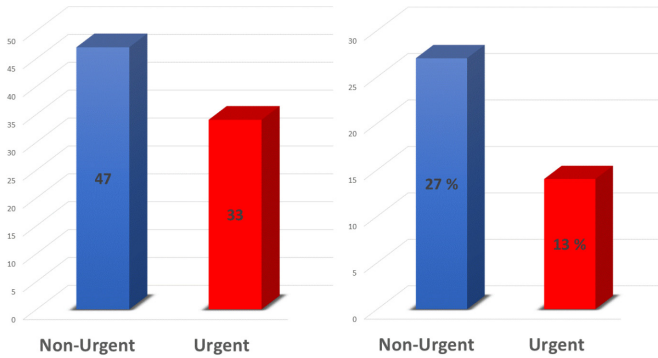


Fig. 7. For each group: average number of steps accessed (left), completion rate (right).

RQ2: Can we design an effective intervention priority framework based on behaviour?

As per Sect. 3.3, we proposed a framework containing two phases. We suppose that the instructor can decide to intervene after 5 weeks (our data). In the prediction phase, we used a stratified 5-fold cross validation to estimate the performance of classification model. To evaluate BERT, we measured accuracy averaged over two classes (Urgent, Non-Urgent), Recall, Precision and F1-score for the (important, minority) Urgent class (Table 2). We prioritise the Recall metric that gave us the rare Urgent cases rather than Precision – preferring to ensure we are capturing all urgent cases.

Table 2. The results of BERT model (Precision, Recall, F1-score for the Urgent class).

Accuracy	Precision	Recall	F1-score
0.90	0.65	0.72	0.68

In the intervention priority phase, there are 387 commenters who have at least one comment that needs Urgent intervention. Table 3 shows the minimum (min) and maximum (max) for each variable in every cluster. For Urgency labelling, we used the label resulting from our manual annotators with voting mechanism, not the one predicted by a classifier, to increase accuracy.

Table 3. The minimum (min) and maximum (max) for each variable in every cluster.

Cluster	Urgency 'min:max'	Sentiment analysis 'min:max'	Steps access 'min:max'
0	'1:3'	'27:75'	'35:52'
1	'4:9'	'7:24'	'15:34'
2	'10:28'	'-3:6'	'0:14'

Finally, to further validate the effectiveness of this proposed model, we computed the relation between different risk groups of learners identified (high, mid, low) and their completion-rates. The distributions are visualised in Fig. 8. From this box plot we note that most of completion-rates of high-risk learners are very low, whilst mid risk learners have average ones and for low risk learners, the completion ration is very high. This is further confirmation that our risk model, based on data from the first half of the course, and refining our potential urgency model, can correctly find learners at risk for not completing their course, and separate them from the other two milder risk groups.

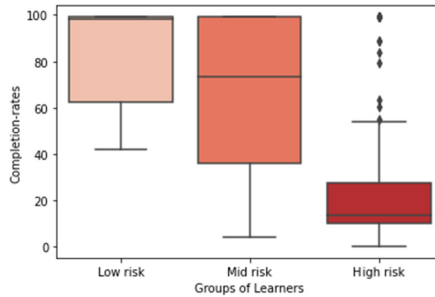


Fig. 8. Boxplot for groups of learners' risk and their completion-rates.

We need here to discuss what constitutes a good classifier of urgent intervention, in terms of best trade-off between false positives (incorrectly identifying learners requiring urgent intervention) and false negatives (failing to identify learners who require urgent intervention). We arguably interpreted it here by giving priority to intervention on urgent cases; hence false negatives were more problematic than false positives. Please also note that learners who need intervention but do not use the comments as communication means are not a target of this research; they would need other means of identification. We also do not compare with work associating comments to participation in MOOCs

[31, 32]— as the focus here is on intervention. Further work can link with the work on pedagogical interventions for automated guidance to instructors [19], as well as evaluating how interventions guided by our procedure presented impact on learner progression.

5 Conclusion

In this paper we addressed the *automatic, intelligent intervention problem* in MOOCs. We offer an analysis of learner comments for *urgency*. We demonstrate that learners with high step access rate require less intervention to their comments, whilst step access of *high-frequency commenters* are less than that of other commenters. This might be due to a decrease in learners' motivation to continue accessing the course material, when they have many comments that need intervention. In addition, we confirmed that most course completers did not need much intervention to their comments. Based on these findings, we have constructed a *framework and algorithm for priority of intervention*, to encourage instructors to help their learners and support them by focusing on learners with high risk first, to improve the potential outcomes of the intervention. This framework can be used in intelligent system in MOOC environments. Future work can look into interventions guided by our procedure and its effect on learner progression, as well as using coefficients to allocate different importance to the three criteria (urgency, sentiment analysis and number of accesses) and other optimisation means for the performance of the intervention procedure.

References

1. Kay, J., et al.: MOOCs: so many learners, so much potential. *IEEE Intell. Syst.* **28**(3), 70–77 (2013)
2. Hone, K.S., El Said, G.R.: Exploring the factors affecting MOOC retention: a survey study. *Comput. Educ.* **98**, 157–168 (2016)
3. Chandrasekaran, M.K., Kan, M.-Y.: *When to reply? context sensitive models to predict instructor interventions in mooc forums*. arXiv preprint [arXiv:1905.10851](https://arxiv.org/abs/1905.10851) (2019)
4. Mazzolini, M., Maddison, S.: Sage, guide or ghost? The effect of instructor intervention on student participation in online discussion forums. *Comput. Educ.* **40**(3), 237–253 (2003)
5. Chandrasekaran, M.K., Kan, M.-Y.: Countering position bias in instructor interventions in MOOC discussion forums. In: *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications* (2018)
6. Sokolovskaya, A.: *Connectivist Knowledge Building, Collaborative Learning, and Social Presence in a Connectivist Massive Open Online Course: A Study of PLENK2010*. Concordia University (2015)
7. Chandrasekaran, M.K., et al.: Using discourse signals for robust instructor intervention prediction. In: *Thirty-First AAAI Conference on Artificial Intelligence* (2017)
8. Guo, S.X., et al.: Attention-based character-word hybrid neural networks with semantic and structural information for identifying of urgent posts in MOOC discussion forums. *IEEE Access* **7**, 120522–120532 (2019)
9. Sun, X., et al.: Identification of urgent posts in MOOC discussion forums using an improved RCNN. In: *2019 IEEE World Conference on Engineering Education (EDUNINE)*. IEEE (2019)

10. Almatrafi, O., Johri, A., Rangwala, H.: Needle in a haystack: identifying learner posts that require urgent response in MOOC discussion forums. *Comput. Educ.* **118**, 1–9 (2018)
11. Lin, F.-R., Hsieh, L.-S., Chuang, F.-T.: Discovering genres of online discussion threads via text mining. *Comput. Educ.* **52**(2), 481–495 (2009)
12. Adnan, M., et al.: Predicting at-risk students at different percentages of course length for early intervention using machine learning models. *IEEE Access* **9**, 7519–7539 (2021)
13. Alrajhi, L., Alharbi, K., Cristea, A.I.: A multidimensional deep learner model of urgent instructor intervention need in MOOC forum posts. In: Kumar, V., Troussas, C. (eds.) *ITS 2020. LNCS*, vol. 12149, pp. 226–236. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_27
14. Clavié, B., Gal, K.: EduBERT: pretrained deep language models for learning analytics. arXiv preprint [arXiv:1912.00690](https://arxiv.org/abs/1912.00690) (2019)
15. Wei, X., et al.: A convolution-LSTM-based deep neural network for cross-domain MOOC forum post classification. *Information* **8**(3), 92 (2017)
16. Bakharia, A.: Towards cross-domain MOOC forum post classification. In: *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*. ACM (2016)
17. Chaturvedi, S., Goldwasser, D., Daumé III, H.: Predicting instructor's intervention in MOOC forums. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2014)
18. Chandrasekaran, M.K., et al.: Learning instructor intervention from MOOC forums: early results and issues. arXiv preprint [arXiv:1504.07206](https://arxiv.org/abs/1504.07206) (2015)
19. Chandrasekaran, M., et al.: Towards feasible instructor intervention in MOOC discussion forums (2015)
20. Chandrasekaran, M.K., et al.: Using discourse signals for robust instructor intervention prediction. arXiv preprint [arXiv:1612.00944](https://arxiv.org/abs/1612.00944) (2016)
21. FutureLearn. <https://www.futurelearn.com>. Last Accessed 24 June 2021
22. Cristea, A.I., et al.: Earliest predictor of dropout in MOOCs: a longitudinal study of FutureLearn courses. *Association for Information Systems* (2018)
23. Alamri, A., et al.: Predicting MOOCs dropout using only two easily obtainable features from the first week's activities. In: Coy, A., Hayashi, Y., Chang, M. (eds.) *ITS 2019. LNCS*, vol. 11528, pp. 163–173. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22244-4_20
24. Agrawal, A., Paepcke, A.: The stanford MOOC posts data set. <https://datastage.stanford.edu/StanfordMoocPosts/>.
25. Antoine, J.-Y., Villaneau, J., Lefevre, A.: Weighted Krippendorff's alpha is a more reliable metrics for multi-coders ordinal annotations: experimental studies on emotion, opinion and coreference annotation. In: *EACL 2014* (2014)
26. Sunar, A.S., et al.: How learners' interactions sustain engagement: a MOOC case study. *IEEE Trans. Learn. Technol.* **10**(4), 475–487 (2016)
27. Devlin, J., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
28. Bonta, V., Janardhan, N.K.A.N.: A comprehensive study on lexicon based approaches for sentiment analysis. *Asian J. Comput. Sci. Technol.* **8**(S2), 1–6 (2019)
29. Min, W.N.S.W., Zulkarnain, N.Z.: Comparative evaluation of lexicons in performing sentiment analysis. *JACTA* **2**(1), 14–20 (2020)
30. North, M.A.: A method for implementing a statistically significant number of data classes in the Jenks algorithm. In: *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE (2009)

31. Saadatdoost, R., et al.: Understanding MOOC learners: insights from participation in coursera MOOC. *Int. J. Web-Based Learn. Teach. Technol. (IJWLTT)* **14**(1), 93–112 (2019)
32. Wong, J.-S., Pursel, B., Divinsky, A., Jansen, B.J.: An analysis of MOOC discussion forum interactions from the most active users. In: Agarwal, N., Xu, K., Osgood, N. (eds.) *SBP 2015. LNCS*, vol. 9021, pp. 452–457. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16268-3_58



Early Predictor for Student Success Based on Behavioural and Demographical Indicators

Efthymou Drousiotis¹(✉), Lei Shi²(✉), and Simon Maskell¹(✉)

¹ Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK
{e.drousiotis,s.maskell}@liverpool.ac.uk

² Department of Computer Science, Durham University, Durham, UK
lei.shi@durham.ac.uk

Abstract. As the largest distance learning university in the UK, the Open University has more than 250,000 students enrolled, making it also the largest academic institute in the UK. However, many students end up failing or withdrawing from online courses, which makes it extremely crucial to identify those “at risk” students and inject necessary interventions to prevent them from dropping out. This study thus aims at exploring an efficient predictive model, using both behavioural and demographical data extracted from the anonymised Open University Learning Analytics Dataset (OULAD). The predictive model was implemented through machine learning methods that included BART. The analytics indicates that the proposed model could predict the final result of the course at a finer granularity, i.e., classifying the students into Withdrawn, Fail, Pass, and Distinction, rather than only Completers and Non-completers (two categories) as proposed in existing studies. Our model’s prediction accuracy was at 80% or above for predicting which students would withdraw, fail and get a distinction. This information could be used to provide more accurate personalised interventions. Importantly, unlike existing similar studies, our model predicts the final result at the very beginning of a course, i.e., using the first assignment mark, among others, which could help reduce the dropout rate before it was too late.

Keywords: MOOCs · Virtual learning environment · Learning analytics · Behavioural analytics · Machine learning · Prediction · BART

1 Introduction

Online learning offers a convenient alternative for everyone to learn on-demand. According to Class Central Report [1], more than 180 million students have enrolled in online learning courses, in particular, MOOCs (Massive Open Online Courses). Yet, one of the well-known challenges in online learning, especially in the context of MOOCs, is student retention. Studies, e.g. [2], show normally only 5%–15% of the students who have registered for a MOOC finally complete it. Luckily, the massive data tracked on online learning platforms, so-called *Educational Big Data*, offers great opportunities to explore how students learn online thus providing insight into (dis)engagement patterns.

In fact, many studies have been conducted to predict student dropout, using techniques through statistical modelling [3] to machine learning [4, 5].

However, most studies, e.g. [4, 6–8], proposed their predictive models using the learning activity data of a whole course, which are not particularly useful in terms of helping the *current* students, as the predictions are only made after the course has completed. A few studies did aim at an earlier prediction using the very first/early data available. For example, Cristea, *et al.* [9] attempted to use the date of registration (in terms of distance from the course start) of students to predict their completion of the course; Alamri, *et al.* [10] used the student's number of accesses and time spent per access in the first week of the course to predict their completion. However, only activity data, i.e., behavioural data, e.g., access to learning materials and discussion forums, were considered; whilst the demographical data, e.g., gender and educational level, might also be available at the start of the course, which might be considered as well to improve the prediction. Additionally, most existing studies, e.g., [11, 12], classified students only into completers and non-completers (two categories), which might hide the differences amongst the students who completed a course, and the differences amongst the students who did not, even though a finer classification might be useful to understand why a student completes or drops out thus providing personalised interventions towards reducing the dropout rate as well as improving their participation and engagement.

Therefore, with the aim of moving towards bridging the gap, this study took into consideration both behavioural and demographical data. The objective was earlier prediction of finer classification of students in online learning especially within the context of MOOCs.

2 Related Work

Along the emergence of big data with the advances in computation, the areas of Learning Analytics (LA) and Educational Data Mining (EDM) have been rapidly developed in recent years, aiming at understanding how people learn online and improving the online learning process. While LA and EDM overlap with each other in similar attributes and goals, they are also different from each other in many aspects [13]. The former is stated as “the process of measuring and collecting data about learners and learning with the aim of improving teaching and learning practice” [14]; the latter is defined as “an emerging discipline, concerned with developing methods for exploring the unique and increasingly large-scale data that come from educational settings and using those methods to better understand students, and the settings which they learn in” [15]. Both aim at improving the analysis of large-scale educational data to support practice in the educational context. In terms of their major differences, according to Siemens and Baker [13], in LA, leveraging human judgement is key, and automated discovery is a tool to accomplish this goal, while in EDM, automated discovery is key, and human judgment is a tool to accomplish this goal; LA has a stronger emphasis on understanding systems as a whole in full complexity, while EDM has a stronger emphasis on reducing components and analysing individual components and the relationships between them.

The main techniques and methods applied in LA and EDM include statistics, machine learning, and data mining, seeking usage patterns of learning resources including video

lectures, forums, assessments, and so on, to compose useful models that can be smoothly adapted to educational data [16]. In particular, three techniques are often used in both LA and EDM: (1) prediction, to find a relationship between known and unknown data using simple statistical methods such as regression, non-linear statistics, and neural [17]; (2) clustering analysis, to create a collection of similar data objects within the same cluster [18]; and (3) relation mining, to classify various relationships that may occur between two or more variables [19].

While most studies, e.g. [20–22], focus on predicting completion and/or dropout rate, e.g., classifying students into completers and non-completers (two categories), we extend the predictive model and further classify students into four categories, including Withdrawn, Fail, Pass and Distinction. Besides, there are only a few similar studies, e.g. [9, 23], that tried predicting as early as possible student completion and dropout rate using limited data gathered. Our study also uses registration date as in previous studies [9] yet associated with also other parameters, as explained below in Sect. 3, with the aim of producing a predictive model with better performance. Moreover, our predictive model aims to enhance the early predictive accuracy by introducing the BART (Bayesian Additive Regression Trees) model.

3 Method

3.1 Dataset

The dataset used in this study is the anonymised OULAD (Open University Learning Analytics Dataset)¹, which contains data about 7 courses and 32,593 registered students (55% males, 45% females), as well as their 10,655,280 interactions (clicks on webpages) with these 7 courses in the Virtual Learning Environment (VLE), operated by the Open University². The dataset is in the format of 7 csv files, connected using unique identifiers including Student_ID, Assessment_ID, and Code_Module (ID of a course).

When joining the Open University for the first time, the students were directly prompted to complete an online form asking about their personal details such as gender and age. While using the VLE to study an online course, students' activity logs were generated, linked by unique Student IDs with timestamps, and recorded in the database.

In total, these 7 courses provided 3,635 learning items, each of which was presented on a webpage in the VLE; there were 196 different assessments, and the students made 173,740 submissions. Interestingly, as Fig. 1 shows, out of 32,593 registered students, only 15,385 (42.78%) passed the courses, highlighting the fail/non-completion issue in MOOCs, which is in consistence with many reports, e.g., [4, 11, 24].

3.2 Study Settings and Data Preparation

The courses under study were organised in weekly learning units, each of which consisted of a collection of learning blocks that might contain one or a few steps. Steps were the fundamental learning items which might include articles, pictures, videos, and quizzes.

¹ https://analyse.kmi.open.ac.uk/open_dataset.

² The OULAD dataset is released under CC-BY 4.0 licence.

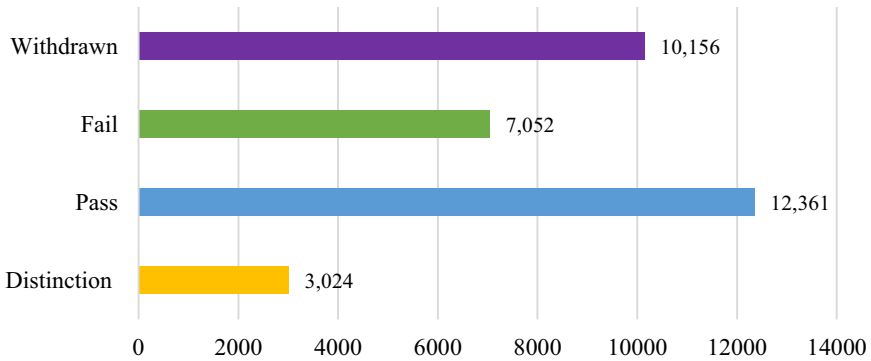


Fig. 1. Number of students in 4 categories: withdrawn, failed, pass, distinction

Figure 2 shows an example of the navigation page of a course, where a student might click one of the WEEK buttons to navigate to the weekly learning unit or click a step title to access a step page (learning item).

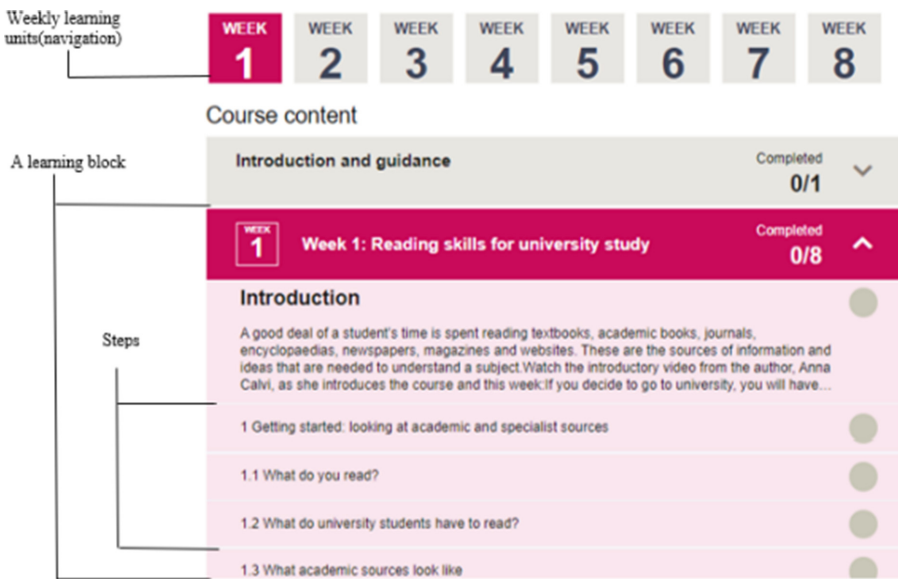


Fig. 2. Navigation page of a MOOC

It’s worth mentioning that the courses in this study were “synchronous” – having official starting and finishing dates and running over an exact number of weeks [11]. In different courses, there were different numbers of assessments during a certain period of time (week); additionally, at the end of each course, there was a final exam. Each course might change slightly, in different runs (i.e., years), the number of weekly learning units and steps, as well as assessment types (tutor marked assessment, computer

marked assessment, and final exam). We used data from all 7 csv files as described in Sect. 3.1. During a course, each student completed several assessments which had different weights summing up to 100%. We used the total number of clicks until a course started, for an *earlier prediction*. Each course had different durations and first assignment submission days, as shown in Table 1. We also converted the categorical variables including Educational Level and Age, into dichotomous variables.

Table 1. Information about MOOCs.

Course	1 st assignment submission day	# of registered students	Year (run)
AAA	Day 19	748	2013 & 2014
BBB	Day 54	7,909	2013 & 2014
CCC	Day 18	4,434	2014
DDD	Day 23	6,272	2013 & 2014
EEE	Day 33	2,934	2013 & 2014
FFF	Day 19	7,762	2013 & 2014
GGG	Day 61	2,534	2013 & 2014

3.3 Analysis

For the analysis, seven variables were defined, as below.

- **First Assignment Mark:** the mark of a student's submission to the first assignment. On the StudentsAssessments csv file, it is called score.
- **Educational Level:** the highest level of education that a student has achieved; including 4 categories: Lower than A level, A level or equivalent, HE Qualification, and Post Graduate Qualification. On the StudentInfo csv file, it is represented as highest_education.
- **Clicks till Course Starts:** the number of clicks made by a student until a course started. Clicks are represented as sum_click on the studentVle csv file.
- **Registration Date:** the date of a student registered for a course, in terms of distance (the number of days) from the start of the course. On the studentRegistration csv file, it is represented as date_registration.
- **Age:** the band of a student's age (0–35, 35–55, >55). On the StudentInfo csv file, it is represented as age_band.
- **Disability:** whether a student has declared a disability. On the StudentInfo csv file, it is represented as disability.
- **Gender:** a student's self-reported gender (male/female). On the StudentInfo csv file, it is represented as gender.
- **Previous Attempts:** times that a student has failed a particular course. On the StudentInfo csv file, it is represented as num_of_pred_attempts.

We used the Pearson chi-square statistical hypothesis to test whether the output (Final Mark Classification) was dependent upon the categorical input variables (Educational level, Age, Gender, Disability), i.e., whether the input variables were relevant to the prediction tasks. The p-value was $<5\%$, which is within the acceptable range [25], indicating that the categorical variables we used were relevant to the output. Moreover, to ensure that the variables were not only dependent upon the output, we also conducted Pearson's correlation tests to measure the strength of the association between the variables (results shown in Table 2), in terms of selecting variables which were not tightly related, in order to improve the predictive models' efficiency. Table 2 shows that the variables were correlated at a very low level showing that it was appropriate to use them as the input variables for our predictive models. The result of the two statistical tests shows that the selected variables fulfilled all the requirements in order to implement efficient and robust predictive models. The chosen variables for the resulting csv file used to train our learning algorithms included the First Assignment Mark, Educational Level, Clicks till Course Starts, Registration date, Age, and Gender. 70% of the data were used as the training data, and 30% as the test data. The majority of the algorithms we used relied on the default settings of the sklearn version 0.24.0, which can be found in the documentation for reference and reproduction³. The learning algorithms we used include Decision Tree, Random Forest, and BART, as they are known for their strong predictive power on binary classification problems.

Table 2. Pearson's correlation test result

	Gender	Educational level	Age	Previous attempts	Disability	First assign. mark	Registration date	Clicks till course start
Gender	1.00							
Educational level	-0.03	1.00						
Age	0.02	0.15	1.00					
Previous attempts	0.04	0.00	0.00	1.00				
Disability	0.04	-0.06	-0.02	0.04	1.00			
First assignment mark	-0.05	-0.01	0.04	-0.04	-0.04	1.00		
Registration date	0.02	0.04	0.03	-0.02	-0.01	0.08	1.00	
Clicks till course starts	-0.10	0.03	0.12	-0.03	0.01	0.24	-0.07	1.00

³ <https://pypi.org/project/scikit-learn/>.

Decision Tree is a supervised learning method which splits the population or sample into two or more homogeneous sets (or sub-populations) based on the most significant splitter/differentiator in input variables that predict the value of the target variable [26].

Random Forest is a supervised learning algorithm that takes randomly selected data to build multiple decision trees merged together to generate more accurate and solid predictions. Specifically, Random Forest gets a prediction from each tree and selects the best solution using voting.

Bayesian Additive Regression Trees (BART), compared to Random Forest and Decision Tree, is the least used algorithm, so it is described in detailed. BART is a Bayesian version of tree ensemble methods where the estimation is given by the variable Y which is a sum of Bayesian CART trees [27]. We used the basic BART model which is shown in (1) below.

$$Y_k = \sum_{j=1}^m g(x_k; T_j, M_j) + \varepsilon_k \quad (1)$$

In Eq. (1), T_j symbols the j^{th} decision tree $j = 1 \dots m$ and M_j is a vector holding the terminal node parameters of T_j , while x_k is an $n \times p$ matrix of variables x , with $x_k = [x_{k1}, \dots, x_{kp}]$, and $\varepsilon_k \sim N(0, \sigma^2)$, where σ^2 is the net variance (bias). In order to create a Bayesian model, we used a prior for the parameters, which in our case is the same as Chipman *et al.* [28] used:

$$P(T_1, M_1, T_2, M_2, \dots, T_m, M_m, \sigma) = [\prod_j^m \{\prod_k^{b_j} P(\mu_{kj} | T_j)\} P(T_j)] P \quad (2)$$

From Eq. (2), we set distributions for the priors $\mu_{kj} | T_j$, σ , and T_j which are $\mu_{kj} | T_j \sim N(\mu_\mu, \sigma_\mu^2)$, $\sigma^2 \sim \text{IG}(\frac{\nu}{2}, \frac{\nu\lambda}{2})$ and $\text{IG}(\alpha, \beta)$ respectively (α : the shape parameter, and β : the rate parameter). For ν , the default value is 3, and λ the value is determined in BART with the quantile set to 0.90.

To evaluate our predictive model's performance, we used the following four metrics.

- **Precision:** the ratio of the correctly predicted positive observations to the total predicted positive observations.
- **Recall:** the ratio of correctly predicted positive observations to all observations in the actual positive class.
- **F1-score (3):** the weighted average of Precision row and Recall row. Therefore, this score takes both false positives and false negatives into account.
- **Accuracy:** the most intuitive performance measure and it is simply a ratio of correctly predicted observations to the total observations.

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

We used the “one-vs-rest” strategy, which fits a binary classifier for each class against all the rest of the classes, in particular – Withdrawn versus the rest, Fail versus the rest, Pass versus the rest, Distinction versus the rest. This allows binary classifiers (Decision

Tree, Random Forest, BART (purely binary classifier)) to apply the already trained algorithm to an unseen sample x and predict the label y and calculating the performance of the algorithm with specific metrics. In our case, those metrics were Precision, Recall, F1-score and Accuracy. Specifically, we used precision and recall metrics as those are better at characterising performance in the context of imbalance data (see Fig. 1).

4 Results and Discussions

Table 3 compares the performance of three similar tree-based algorithms that we used in the analysis, including Decision Tree, Random Forest, and BART. As mentioned in Sect. 3, we explored the BART model with the aim of improving our results and enhance the prediction accuracy. Interestingly, we found BART could give the optimum prediction accuracy on every “one-vs-rest” pair. Specifically, we achieved a relatively high accuracy of 81% for identifying students who might Withdraw from a course, 80% accuracy identifying students who might Fail, 69% accuracy identifying students who would get a Pass mark for the course, and 92% accuracy identifying students who might get a Distinction mark.

Table 3. Performance comparisons between three predictive models

	Metric	Decision tree	Random forest	BART
Withdrawn	Precision	0.65	0.75	0.81
	Recall	0.65	0.71	0.91
	F1	0.65	0.72	0.86
	Accuracy	0.69	0.78	0.81
Fail	Precision	0.68	0.69	0.79
	Recall	0.67	0.75	0.98
	F1	0.67	0.71	0.87
	Accuracy	0.67	0.76	0.80
Pass	Precision	0.63	0.65	0.72
	Recall	0.62	0.65	0.74
	F1	0.62	0.65	0.73
	Accuracy	0.63	0.65	0.69
Distinction	Precision	0.85	0.86	0.92
	Recall	0.84	0.90	0.98
	F1	0.85	0.87	0.96
	Accuracy	0.84	0.89	0.92

Table 4 shows the reason for a relatively low accuracy (yet, higher than Decision Tree and Random Forest), i.e., 69%, for the “Pass-vs-rest” pair classification, as the misclassified cases between the two classes is fairly high. As the Pass class is between the Fail class and the Distinction class, it seems that the algorithms tend to misclassify the Pass class as Fail or Distinction which is not happen.

Table 4. Confusion matrix for pass versus the rest

	Pass	Rest
Pass	1,582	1,529
Rest	1,436	3,879

Moreover, Fig. 3 shows the performance of the algorithms for the “Distinction-vs-rest” classification task, where we can observe the improved ability of the BART algorithm in comparison with Random Forest and Decision Tree algorithms to correctly classify the data.

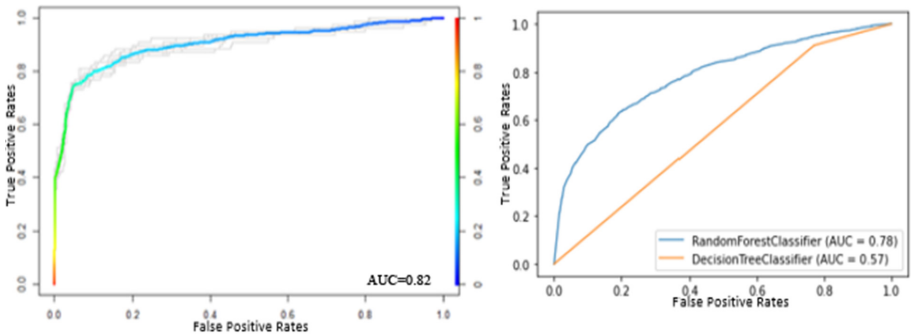


Fig. 3. BART (Left) Random Forest (Right) Decision Tree (Right) AUC graphs for Distinction versus the rest

Our results suggest that combining demographical data (such as educational level, gender, age, and disability) and behavioural data (such as student’s daily activity (clicks), the number of previous attempts in a course, first assignment mark, and registration date) can produce a predictive model with good performance.

The results obtained are worthy of discussion - as we observe that among the tree-based machine learning algorithms we used, the BART outperforms the others. To begin with, our results show that BART produced the optimal predictive accuracy for every “one-vs-rest” pair (i.e., Withdrawn, Fail, Pass, Distinction, respectively, with the rest of the classes). Our model could predict the final result classification (Withdrawn, Fail, Pass, Distinction), so the lecturers, after the first assignment, can use it to identify who is more likely to Fail, Pass, etc., thus being able to provide early interventions to these students, with tailored reminders, as the students were classified into finer-grained categories (comparing to other methods that classified them into only two categories – completers and non-completers).

It is very important to highlight the strong predictive power of the number of clicks (resource, glossary, URL, forum, homepage, etc.) on the VLE, which we should aim to raise in order to improve students’ performance. Figure 4 shows that students who failed (green dots) exhibit significantly a smaller number of clicks on the VLE compared to those with a pass (blue dots) or a distinction (yellow dots) mark. This suggests that high

scores are associated with more frequent access to the VLE, and that, in order to have a better result of the course, students should be using the VLE more often.

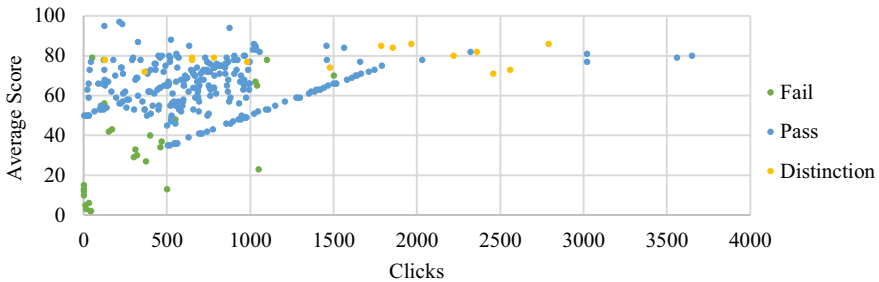


Fig. 4. Relationship between the number of clicks and the overall student outcome (Color figure online)

5 Conclusions

In summary, this paper presents the results of a study aiming to discover whether it is possible to predict and identify, as early as possible, which students might withdraw from a course, and, possibly, make earlier interventions to reduce their withdrawal or failure, and to improve students' final marks. This is different from most previous studies that analysed the data after the completion of the whole course which is not very useful for the **current** students. To produce and validate the predictive models, we have examined 8 independent variables in total, including both demographical variables (*Educational Level, Gender, Age, Disability*), and behavioural variables (*Registration Date, Clicks until Course Starts, First Assignment Score, and Previous Attempts* on Open University's (OU) VLE). This is different from most previous studies where only behavioural variables are included.

The main limitation, however, was the strict scope of the dataset. The daily interaction with the VLE, i.e., clicks, plays an important role but the virtual learning system (VLE) is not integral. For example, the results of the final written exams were not included in the csv files. Besides, on the independent variable *Clicks till Course Starts*, we could not take into consideration the students' educationally relevant discussions outside of the OU's VLE or the private discussion forums, and it is worth noting that not all learning behaviour could be fully captured through online platforms.

Future work may include investigating and validating efficient strategies for the use of the proposed predictive model. For example, it could be used in 3 different stages of a MOOC. Firstly, use the model to identify, as early as possible, the students who are likely to withdraw. For example, in order to keep the student remaining in a course, the lecturer could send personalised messages reinforcing the usefulness and objectives of the course. Secondly, after a couple of weeks, when more data is collected such as the second assignment mark, the lecturer could use the model to identify students who might fail with improved accuracy and provide them with necessary supports. Finally,

at the final stage of the MOOC (previous assignments marks could have been added to the model as an additional input) before the final examination, the model can be used to identify the students with Pass or Distinction marks and provide the lecturer with a precise overview of the students' benchmarks. Importantly, the first assignment mark is suggested to be a very strong predictor of students' performance. Thus, the lecturer is recommended to periodically send students reminders with evidence, to emphasise the importance of participation and engagement to be successful in a course.

References

1. By The Numbers: MOOCs in 2020 — Class Central. The Report by Class Central, 30 November 2020. <https://www.classcentral.com/report/mooc-stats-2020/>. Accessed 04 Jan 2021
2. Study offers data to show MOOCs didn't achieve their goals | Inside Higher Ed. <https://www.insidehighered.com/digital-learning/article/2019/01/16/study-offers-data-show-moocs-didnt-achieve-their-goals>. Accessed 04 Jan 2021
3. Gomez-Zermeno, M.G., Garza, L.A.D.L.: Research analysis on MOOC course dropout and retention rates (2016). <https://doi.org/10.17718/tojde.23429>
4. Dalipi, F., Imran, A.S., Kastrati, Z.: MOOC dropout prediction using machine learning techniques: review and research challenges. In: 2018 IEEE Global Engineering Education Conference (EDUCON), pp. 1007–1014, April 2018. <https://doi.org/10.1109/educon.2018.8363340>
5. Borrella, I., Caballero-Caballero, S., Ponce-Cueto, E.: Predict and intervene: addressing the dropout problem in a MOOC-based program. In: Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale, Chicago, IL, USA, June 2019, pp. 1–9. <https://doi.org/10.1145/3330430.3333634>
6. Kloft, M., Stiehler, F., Zheng, Z., Pinkwart, N.: Predicting MOOC dropout over weeks using machine learning methods. In: Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs, Doha, Qatar, October 2014, pp. 60–65. <https://doi.org/10.3115/v1/w14-4111>
7. Liang, J., Li, C., Zheng, L.: Machine learning application in MOOCs: dropout prediction. In: 2016 11th International Conference on Computer Science Education (ICCSE), August 2016, pp. 52–57. <https://doi.org/10.1109/iccse.2016.7581554>
8. Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., Tingley, D.: MOOC dropout prediction: how to measure accuracy? In: Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale, Cambridge Massachusetts USA, April 2017, pp. 161–164. <https://doi.org/10.1145/3051457.3053974>
9. Cristea, A., Alamri, A., Stewart, C., Alshehri, M., Shi, L.: Earliest predictor of dropout in MOOCs: a longitudinal study of FutureLearn Courses Mizue Kayama, August 2018
10. Alamri, A., et al.: Predicting MOOCs dropout using only two easily obtainable features from the first week's activities. In: Coy, A., Hayashi, Y., Chang, M. (eds.) ITS 2019. LNCS, vol. 11528, pp. 163–173. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22244-4_20
11. Wang, Y., Baker, R.: Content or platform: why do students complete MOOCs? **11**(1), 14 (2015)
12. Uden, L., Sinclair, J., Tao, Y.-H., Liberona, D. (eds.): LTEC 2014. CCIS, vol. 446. Springer, Cham (2014). <https://doi.org/10.1007/978-3-319-10671-7>
13. Baran, E., Siemens, Baker: Learning analytics and educational data mining: towards communication and collaboration. In: Learning Environments Design Reading Series

14. Learning analytics | Advance HE. <https://www.advance-he.ac.uk/knowledge-hub/learning-analytics>. Accessed 29 Mar 2021
15. Educationaldatamining.org. <https://educationaldatamining.org/>. Accessed 29 Mar 2021
16. Liñán, L.C., Pérez, Á.A.J.: Minería de dades educatives i anàlisi de dades de l'aprenentatge: diferències, semblances i evolució en el temps. RUSC. Univ. Knowl. Soc. J. **12**(3) (2015). Article no. 3. <https://doi.org/10.7238/rusc.v12i3.2515>
17. Madigan, C.D., Daley, A.J., Kabir, E., Aveyard, P., Brown, W.: Cluster analysis of behavioural weight management strategies and associations with weight change in young women: a longitudinal analysis. *Int. J. Obes.* **39**(11), 1601–1606 (2015). <https://doi.org/10.1038/ijo.2015.116>
18. 4 - Prediction.pdf. <http://www.cs.stir.ac.uk/courses/ITNP60/lectures/1%20Data%20Mining/4%20-%20Prediction.pdf>. Accessed 29 Mar 2021
19. Klapaftis, Ioannis P., Pandey, S., Manandhar, S.: Graph-based relation mining. In: Dziech, A., Czyżewski, A. (eds.) MCSS 2011. CCIS, vol. 149, pp. 100–112. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21512-4_12
20. Guo, P.J., Reinecke, K.: Demographic differences in how students navigate through MOOCs. In: Proceedings of the first ACM conference on Learning @ scale conference, New York, NY, USA, March 2014, pp. 21–30. <https://doi.org/10.1145/2556325.2566247>
21. Shi, L., Cristea, A.: Demographic indicators influencing learning activities in MOOCs: learning analytics of FutureLearn Courses, August 2018
22. Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., Tingley, D.: Delving deeper into MOOC student dropout prediction. *arXiv:1702.06404* [cs], February 2017. <http://arxiv.org/abs/1702.06404>. Accessed 28 Jan 2021
23. Brinton, C.G., Chiang, M.: MOOC performance prediction via clickstream data and social learning networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM), April 2015, pp. 2299–2307. <https://doi.org/10.1109/infocom.2015.7218617>
24. Liyanagunawardena, T.R., Williams, S.A.: Dropout: MOOC participants' perspective', p. 8
25. Bolboacă, S.D., Jäntschi, L., Sestraş, A.F., Sestraş, R.E., Pamfil, D.C.: Pearson-Fisher chi-square statistic revisited. *Information* **2**(3) (2011). Article no. 3. <https://doi.org/10.3390/info2030528>
26. 1.10. Decision Trees — scikit-learn 0.24.1 documentation. <https://scikit-learn.org/stable/modules/tree.html>. Accessed 29 Mar 2021
27. Chipman, H.A., George, E.I., McCulloch, R.E.: BART: Bayesian additive regression trees. *arXiv:0806.3286* [stat], October 2010. <https://doi.org/10.1214/09-aos285>



Predicting Certification in MOOCs Based on Students' Weekly Activities

Mohammad Alshehri^{1,2}(✉), Ahmed Alamri^{1,3}, and Alexandra I. Cristea¹

¹ Department of Computer Science, Durham University, South Road, Durham DH1 3LE, UK
{mohammad.a.alshehri, a.s.alamri,
alexandra.i.cristea}@durham.ac.uk

² College of Business, University of Jeddah, Jeddah, Saudi Arabia

³ College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia

Abstract. Massive Open Online Courses (MOOCs) have been growing rapidly, offering low-cost knowledge for both learners and content providers. However, currently there is a very low level of course purchasing (less than 1% of the total number of enrolled students on a given online course opt to purchase its certificate). This can impact seriously the business model of MOOCs. Nevertheless, MOOC research on learners' purchasing behaviour on MOOCs remains limited. Thus, the umbrella question that this work tackles is *if learner's data can predict their purchasing decision (certification)*. Our fine-grained analysis attempts to uncover the latent correlation between learner activities and their decision to purchase. We used a relatively large dataset of 5 courses of 23 runs obtained from the less studied MOOC platform of FutureLearn to: (1) statistically compare the activities of non-paying learners with course purchasers, (2) predict course certification using different classifiers, optimising for this naturally strongly imbalanced dataset. Our results show that learner activities are good predictors of course purchasability; still, the *main challenge was that of early prediction*. Using only student *number of step accesses, attempts, correct and wrong answers*, our model achieve promising accuracies, ranging between 0.81 and 0.95 across the five courses. The outcomes of this study are expected to help design future courses and predict the profitability of future runs; it may also help determine what personalisation features could be provided to increase MOOC revenue.

Keywords: Learner analytics · MOOCs · Certification prediction

1 Introduction

Online courses have been revolutionising and reforming education for decades. More recently, massive open online courses (MOOCs) were developed, specifically to reach a massively unlimited number of potential learners from around the world. This modern age of e-learning commenced with the commercially successful introduction of Stanford's Coursera in 2011 [1]. The following year witnessed the launch of many of today's MOOC platforms, coining 2012 as "the year of the MOOCs" [2]. Many providers, such

as *FutureLearn*, *edX*¹, *Udemy*² and *Coursera*³ have started offering scalable online courses to the public, with a diverse set of learning content for learners from all over the world [3, 4]. This has resulted in 16.3 thousand MOOCs delivered via more than 950 university partners to more than 180 million learners by the end of 2020 [5].

Although MOOCs have been successful, attracting many online learners, the staggeringly *low completion and certification rates* is still one of the more concerning aspects to date, a funnel with students “leaking out” at various points along the learning pathway [6, 7]. While the high dropout rate has been the focus of many studies [8–10], the race towards identifying precise predictors of completion as well as the *predictors of course purchasing*, continues. Importantly, although MOOCs have started being analysed more thoroughly in the literature, few studies have investigated the characteristics and temporal activities for the purpose of modelling learners’ certification decision behaviours. Concomitantly, the literature shows that user purchasing behaviour has been widely studied on pure e-commerce platforms [11]. To date, this kind of behaviour has not been extensively considered in the educational domain, even though MOOC providers have been struggling to build their own sustainable revenues [12]. Considering the recent MOOCs’ transition towards paid macro-programmes and online degrees with affiliate university partners, this paper presents a fine-grain exploration of student behaviours from a different point of view, non-paying learners versus certificate purchasers. Specifically, this paper attempts to answer the following research questions:

- **RQ1:** *Do MOOC non-paying learners behave differently to course purchasers as to their activities of access and question answering (attempts, correct/wrong answers)?*
- **RQ2:** *Can MOOC learner’s logged data predict course purchase decisions (certification)?*

It is worth mentioning that the first research question attempts to compare the activities of non-paying learners (NL) versus certificate purchasers (CP) using a systematic statistical methodology as shown in Sect. 3.5. Subsequently, the second research question examines whether learner activities can be used to predict later certification behaviour. This goes beyond comparing samples to employing some state-of-art ML algorithms to predict students’ decisions of purchasing a certificate after finishing the course. This type of prediction seems essential keeping in mind that the certificate purchasing decision is usually taken after the end of the course i.e. after attending the whole course’ weekly content.

2 Related Work

Looking through the few studies that investigated MOOC certification, [13] studied the relationship between intention of completion, actual completion, and certificate earning. The study applied on 9 HarvardX MOOCs showed that the correlation between the first two variables was a stronger predictor of certification than any demographic traits.

¹ www.edx.org.

² www.udemy.com.

³ www.coursera.org.

[14] studied MOOC learners' subsequent educational goals after taking the course, by using consumer goal theory. They showed that MOOC completers satisfied with the course delivery were more likely to progress to the course-host institution, than the non-completers. It also showed that having a similar pedagogical and delivery approach in a university for both conventional and online courses can encourage learners to join further academic online study. It thus became a roadmap for tertiary institutes on how to design an effective MOOC to target potential future students.

Using the only the first week behaviour, [15] predicted MOOC certification via an asset of features. This includes average quiz score, number of completed peer assessments, social network degree and being either a current or prospective student at the university offering the course. Their Logistic Regression classifier model was trained and tested on one MOOC run only under certain conditions and incentives, by the provider; therefore, it might need to be replicated, for the results to be generalisable. Qiu et al. [16] extracted factors of engagement in XuetangX (China, partner of edX) on 11 courses, to predict grades and certificate earning with different methods (LRC, SVM, FM, LadFG); their performance was evaluated using the area under the curve (AUC), precision, recall, and F1 score. However, the number of features used, i.e. demographics (gender, age, education), forums (number of new posts and replies), learning behaviour (chapters browsed, deadlines completed, time spent on videos, doing assignments, etc.), courses delivery windows (delivered within 8 months only) and study learners (around 88,000) are relatively low. [17] used four different algorithms (RF, GB, k-NN and LR) to predict student certification on one edX-delivered course. They used a total of eleven independent variables to build the model and predict the dependent variable—the acquisition of a certificate (true or false).

More recently, [18] used behavioural and social features of one course “Big Data in Education”, which was first offered on Coursera and later on edX, to predict dropout and certification. Table 1 below summarises the surveyed certification prediction models. Data used included Click Stream (CS), Forum Posts (FP), Assignments (ASSGN), Student Information Systems (SIS), Demographics (DEM) and Surveys (SURV).

Table 1. Certification prediction models versus our model.

Refs.	Data source	#Courses	#Students	Data description
[19]	Coursera	1	826	CS; FP
[15]	Coursera	1	37,933	ASSGN; FP; SIS
[13]	HarvardX	9	79,525	DEM; SURV
[20]	edX	1	43,758	CS
[21]	Coursera	1	84,786	FP
[16]	XuetangX	11	88,112	CS
[22]	HarvardX-MITx	10	n/a	CS; FP
[18]	Coursera; edX	1	65,203	CS; FP
Our model	Future learn	9	245,255	CS; ASSGN; FP

Unlike previous studies on certification, our proposed model aims to predict the financial decisions of learners on whether to purchase the course certificate. Also, our work is applied to a less frequently studied platform, FutureLearn (Table 1). Another concern we address is study size, with 6 out of the total 9 studies conducted on one course only. As students may behave differently based on the course attended, previous models' generalisability is unclear. Instead, we used a variety of courses from different disciplines: Literature, Psychology, Computer Science and Business. Another novelty of our study is predicting the learner's real financial decision on buying the course and gaining a certificate. Most course purchase prediction models identify certification as an automatic consecutive step to the completion, making them not different from completion predictors. Our study additionally identifies the most representative factors for certification purchase prediction. It also proposes tree-based and regression classifiers to predict MOOC purchasability using relatively few input features.

3 Methodology

3.1 Data Collection

When a learner joins FutureLearn for a given course, the system generates logs to correlate unique IDs and time stamps to learners, recording learner activities, such as weekly-based steps visited, completed, comments added, or question attempted [23]. The current study is analysing data extracted from a total of 23 runs spread over 5 MOOC courses, on 4 distinct topic areas, all delivered through FutureLearn, by the University of [university name removed]. These topic areas are: Literature (with course Shakespeare and his World [SP]; with course duration 10 weeks); Psychology (with courses The Mind is Flat [TMF]: 6 weeks, and Babies in Mind [BIM]: 4 weeks); Computer Science (Big Data [BD]: 9 weeks) and Business (Supply Chains [SC]: 6 weeks).

These courses were delivered repeatedly in consecutive years (2013–2017), thus we have data on several 'runs' for each course. Table 2 below shows the number of enrolled, non-paying learners (NL), as well as those having purchased a certificate (CP). Our data shows that students *accessed 3,007,789 materials* in total and declared *2,794,578 steps completed*. Regarding these massive numbers, Table 2 clearly illustrates the low certification rate (less than 1% of the enrolled students).

3.2 Data Preprocessing

The obtained dataset went through several processing steps, in order to be prepared and fed into the learning model. Since some students were found to be enrolled on more than one run of the same course, the run number was attached to the student's ID, to avoid any mismatch during joining student activities over "several runs" with their current activities.

The pre-processing further contained some standard data manipulations, such as processing (replacing) missing values with zeros, applying *lambda* and *factorize* functions along with Pandas [24] and NumPy [25] to render the data format as machine-feedable. The pre-processing further contained eliminating irrelevant data generated by organisational administrators (455 admins across the 23 runs analysed). Table 3 shows the main four features analysed in this study.

Table 2. The number of non-paying learners and certificate purchasers on 5 FutureLearn courses.

Course	#Runs	#Weeks	#Non-paying learners	#Certificate purchasers
BIM	6	4	48,777	676
BD	3	9	33,430	268
SP	5	10	63,630	750
SC	2	6	5810	71
TMF	7	6	93,608	321
Total	23	35	245,255	2086

Table 3. The features utilised for comparing student activities and predicting course purchasability

Activity source	Activities (per week)
Step access (<i>a</i>)	# Accessed steps
Attempts (<i>t</i>)	# Attempts
Correct answers (<i>r</i>)	# Correct answers
Wrong answers (<i>f</i>)	# Wrong answers

3.3 Feature Extraction

The preliminary data shape is a timestamp log spread on different data frames based on the data log source (access log, question answering log, comments and responses log). As MOOCs are usually delivered on a weekly basis, it was essential to compute the various weekly activities of each learner generating a temporal matrix of their weekly activities. The newly processed Students Activities matrix of each course is as follows:

$$sa = \begin{bmatrix} s_1 & a_{w(1-n)} & t_{w(1-n)} & r_{1-n} & f_{1-n} \\ s_2 & a_{w(1-n)} & t_{w(1-n)} & r_{1-n} & f_{1-n} \\ \dots & \dots & \dots & \dots & \dots \\ s_n & a_{w(1-n)} & t_{w(1-n)} & r_{1-n} & f_{1-n} \end{bmatrix}$$

where *s* = student, *a* = access, *t* = attempt, *r* = correct answers, *f* = wrong answers, *w* = week, *n* = the number of the weeks in a given course.

3.4 Features Selection

Our pre-processed number of features as can be seen in the *sa* matrix above is considerably high due to multiplying the total number of the main extracted features (4) by the total number of weeks *w* in a given course *c*. This resulted in a large array of features, especially for long courses like SP, where the number of weeks was 10, hence generating 40 features. This would on one hand allow for: (1) a temporal fine-grain analysis

of the course's content, (2) a timely and early prediction of student's behaviours. However, in order to highlight the most representative features, feature selection techniques were applied, as below. As algorithms employed include tree-based and regression, the features for the tree-based algorithms were selected using Mean Decrease in Impurity (MDI), whereas Variance Inflation Factor (VIF) was used to detect and reduce the multilinearity for the regression algorithms as further explained below [26].

Mean Decrease in Impurity (MDI)

MDI counts the times a feature is used to split a node, weighted by the number of samples it splits. It calculates each feature importance as the sum over the number of splits (across all trees) that include the feature, proportionally to the number of samples it splits. MDI is defined as the total decrease in node impurity (weighted by the probability of reaching that node—which is approximated by the proportion of samples reaching that node) averaged over all trees of the ensemble [27].

Variance Inflation Factor (VIF)

Prior to doing regression, multicollinearity among our input features should be taken into consideration. We use VIF (Variable Inflation Factor) to analyse multicollinearity.

$$vif_i = \frac{1}{1 - R_i^2}$$

where R_i^2 is the R^2 value obtained by regressing the i^{th} predictor on the remaining predictors. Dropping variables after calculating VIF was an iterative process, starting with the variable having the largest VIF value, as its trend is highly captured by other variables. It was noticed that dropping the highest VIF feature has sequentially reduced the VIF values for the remaining features.

3.5 Statistical Analysis

Normality Test

Our first step of exploring our dataset was examining whether it comes from a specific distribution. The three common procedures of normality verification procedures of: graphical method (Quantile-Quantile plot), numeric method (skewness and kurtosis) and formal normality tests (Shapiro-Wilk) were applied [28]. This has revealed that our data comes from non-Gaussian (normal) distribution and therefore nonparametric tests were conducted as below.

Mann-Whitney U Test

As our data is not normally distributed as well as the variables we are analysing are independent, we used Mann-Whitney U test (Mann-Whitney-Wilcoxon (MWW) [29]), a nonparametric test for testing the statistical significance of the difference of distributions. We use it here to compare the activities of non-paying learners with certificate purchasers.

3.6 Classification Algorithms

Further to the statistical inference, the current study applied four different classification and regression algorithms to predict MOOC learners' purchasing behaviour: Random Forest (RF), ExtraTree (ET), Logistic Regression (LR) and Support Vector Classifier (SVC). These algorithms were chosen due to the fact that they were able to predict course purchasability well, by dealing with massively imbalanced datasets and using at the same time only very few features, as shown in Table 3. These input features exist in any standard MOOC system, which further promotes our model as generalisable. There are some further features that can be utilised for learner behaviour prediction, e.g. demographics or leaving surveys; these features are either not generated by every MOOC platform, or logged later after the end of the course, making early prediction of purchasing behaviour challenging.

To simulate the real-world issue of the low certification rate in MOOCs, we fed the imbalanced data to the classification models as-is. We have initially used many other classification algorithms for this prediction tasks. However, the algorithms that do not deal well with imbalanced data, i.e. have a parameter to define the class weight during learning were excluded.

To deal with our imbalanced dataset, we used the Balanced Accuracy (BA), also known as the Area Under the ROC Curve, which is defined as the average of recall obtained on each class [30]. BA equals the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate) as follows:

$$ba = \frac{1}{2} \left(\frac{tp}{tp + fn} + \frac{tn}{tn + fp} \right)$$

Having applied the above preprocessing steps, the shape of X and Y passed to the prediction model was as depicted in Table 4.

Table 4. Number of observations in each class of 0 and 1 by number of selected features

Course	Class_0	Class_1
BIM	(25508, 18)	(625, 18)
BD	(16010, 30)	(232, 30)
SC	(2840, 26)	(59, 26)
SH	(28920, 42)	(497, 42)
TMF	(39533, 26)	(308, 26)

4 Results and Discussion

The results explore how our processed features can temporally identify course buyers based on their activity data. Our temporal analysis showed some statistical significance at various levels when comparing Non-paying Learners and Certificate Purchasers'

behaviours across the five courses analysed. Due to the paper limit, we are reporting the most important results here only ordered by the activity categories as shown in Tables 5, 6, 7 and 8, where **bold** values mean the most significant value in a given course. As the courses analysed spanned over different numbers of weeks, we have selected the first, middle and last weeks to report the results, for fairness of comparison and easy visualisation. For courses with an even number of weeks, we have selected the middle week closer to the end of the course. Our results show that paying learners were generally more engaged with the course content, in terms of accessing the content more frequently, answering more questions correctly and being more socially interactive, i.e. having more comments and responses over their learning journey.

4.1 Access

Purchasers seem to have a higher number of accessed steps towards the end of the course. With the SC course as an exception, the purchasers' weekly number of access is increasing at different level of significance, but with the last week being the most significant for the majority of the courses.

Table 5. Comparison of the number of Access for non-paying learners and purchasers at three different time points of the course.

C	M	(NL)			(CP)			<i>p-value</i> <i>1st week</i>	<i>p-value</i> <i>Mid week</i>	<i>p-value</i> <i>Last week</i>
		1st week	Midweek	Last week	1st week	Mid week	Last week			
BIM	μ	18.25	11.09	10.58	18.30	12.64	14.20	<i>3.1E-06</i>	<i>2.2E-12</i>	<i>8.4E-26</i>
	σ	2.09	6.16	8.35	2.30	5.27	7.38			
SH	μ	15.60	11.55	13.23	15.63	11.59	14.26	<i>3.7E-01</i>	<i>2.3E-01</i>	<i>3.3E-08</i>
	σ	1.30	2.18	5.75	1.04	2.11	4.94			
TMF	μ	14.71	12.06	15.53	14.70	11.88	15.61	<i>4.7E-01</i>	<i>7.1E-03</i>	<i>6.5E-03</i>
	σ	1.62	3.11	6.65	1.62	3.16	6.92			
SC	μ	18.66	16.48	21.17	18.24	17.10	21.49	<i>2.3E-01</i>	<i>3.4E-01</i>	<i>4.8E-01</i>
	σ	2.23	4.78	8.81	3.36	3.59	8.29			
BD	μ	11.72	9.32	7.83	11.73	9.61	10.03	<i>1.8E-01</i>	<i>1.2E-02</i>	<i>8.4E-08</i>
	σ	1.58	3.56	6.57	1.53	3.41	6.05			

4.2 Correct Answers

The students who purchased a certificate at the end of course have generally answered more correct answers compared to non-paying learners. Contrary to the trend, TMF has shown different results for both statistical analysis and the number of correct answers.

Table 6. Comparison of the number of Correct Answers for non-paying learners and purchasers at three different time points of the course.

C	M	(NL)			(CP)			<i>p-value</i> <i>1st week</i>	<i>p-value</i> <i>Mid week</i>	<i>p-value</i> <i>Last week</i>
		1st week	Midweek	Last week	1st week	Mid week	Last week			
BIM	μ	4.70	4.13	3.34	4.76	4.12	3.54	<i>1.9E-02</i>	<i>3.7E-15</i>	<i>7.4E-31</i>
	σ	1.18	1.92	2.37	1.14	1.99	2.30			
SH	μ	11.41	11.11	9.24	11.45	11.30	9.80	<i>1.7E-01</i>	<i>1.3E-02</i>	<i>1.2E-03</i>
	σ	1.26	2.69	4.53	1.16	2.38	4.14			
TMF	μ	9.54	8.86	7.76	9.38	8.77	7.53	<i>3.3E-04</i>	<i>3.5E-01</i>	<i>1.1E-0</i>
	σ	1.40	2.54	3.82	1.51	2.65	3.94			
SC	μ	4.85	4.49	4.09	4.83	4.58	4.15	<i>4.8E-01</i>	<i>3.1E-01</i>	<i>4.6E-01</i>
	σ	0.83	1.50	1.87	0.91	1.40	1.81			
BD	μ	4.58	3.39	2.02	4.67	4.01	2.93	<i>1.8E-01</i>	<i>1.3E-05</i>	<i>1.5E-08</i>
	σ	1.38	2.32	2.26	1.25	2.03	2.18			

4.3 Number of Comments

The number of comments posted by learners seem to be the most effective predictor of course purchasability. We can see from Table 7 below that purchasers have commented more than non-paying learners across all weeks and all courses.

Table 7. Comparison of the number of Comments for non-paying learners and purchasers at three different time points of the course

C	M	(NL)			(CP)			<i>p-value</i> <i>1st week</i>	<i>p-value</i> <i>Mid week</i>	<i>p-value</i> <i>Last week</i>
		1st week	Mid week	Last week	1st week	Mid week	Last week			
BIM	M	1.71	0.76	0.63	3.17	1.95	2.01	<i>1.0E-35</i>	<i>1.3E-47</i>	<i>8.6E-68</i>
	Σ	2.81	1.87	1.97	3.74	2.98	3.35			
SH	M	1.56	1.17	1.14	2.70	2.06	2.23	<i>1.3E-22</i>	<i>2.6E-22</i>	<i>1.7E-21</i>
	Σ	2.89	2.26	2.23	3.76	2.91	3.23			
TMF	M	1.46	0.89	1.02	1.73	1.17	1.30	<i>4.9E-02</i>	<i>2.8E-02</i>	<i>1.3E-01</i>
	Σ	2.65	2.01	2.47	2.86	2.39	3.06			
SC	M	1.51	0.92	1.33	2.58	2.02	3.31	<i>8.5E-03</i>	<i>1.1E-03</i>	<i>8.9E-04</i>
	Σ	2.84	2.44	3.76	3.90	3.44	5.74			
BD	M	0.62	0.32	0.36	1.03	0.59	0.84	<i>2.6E-07</i>	<i>8.4E-06</i>	<i>9.2E-09</i>
	Σ	1.55	1.02	1.19	2.02	1.30	1.84			

4.4 Number of Replies

The number of replies posted by both non- and paying learners have similar pattern to the number of comments discussed above. However, non-paying learners in SH and TMF courses have responded more during the first weeks only.

Table 8. Comparison of the number of Replies for non-paying learners and purchasers at three different time points of the course.

C	M	(NL)			(CP)			<i>p-value</i>	<i>p-value</i>	<i>p-value</i>
		1st week	Mid week	Last week	1st week	Mid week	Last week	<i>1st week</i>	<i>Mid week</i>	<i>Last week</i>
BIM	M	0.41	0.26	0.14	0.68	0.67	0.43	$5.0E-09$	$1.3E-16$	$1.6E-23$
	Σ	1.65	1.39	0.82	2.27	2.59	1.56			
SH	M	1.28	0.95	0.82	0.98	1.05	0.85	$1.9E-04$	$5.5E-02$	$2.0E-02$
	Σ	10.47	4.89	5.54	3.56	4.81	3.52			
TMF	M	0.85	0.72	0.83	0.79	0.82	0.94	$4.5E-01$	$2.5E-01$	$4.3E-01$
	Σ	3.52	4.33	5.10	3.67	3.25	4.57			
SC	M	0.32	0.14	0.27	0.66	0.12	0.41	$1.9E-01$	$1.1E-01$	$3.3E-02$
	Σ	1.39	0.83	1.23	2.21	0.38	1.05			
BD	M	0.55	0.28	0.16	0.92	0.41	0.27	$3.4E-05$	$1.4E-04$	$2.5E-02$
	Σ	2.68	1.70	1.07	2.80	1.21	1.33			

4.5 Prediction Performance

The results as shown in Table 9 achieved promising balanced accuracies (BA) across the five domain-varying courses. Keeping numbers of students from Table 2 in mind, it can be seen that there is an inverse relationship between the number of times a course is delivered *#Runs* and the model performance. This suggests that learner activities may be different between runs of the same course, even though the content of each different run of a given course is almost the same—hence generating noisier data for the model to learn. This may also explain why the CS course, with the lowest number of purchasers, has achieved the highest results on both classes' recalls, compared to the other courses. Class-wise, it is worth mentioning that Recall_1 prediction *our main target, paying students* was greater than Recall_0 over all the five courses.

Table 9. Learner classification results distributed by course, class 0 = non-paying learners, class 1 = paid learners.

Course	Classifier	1st week			Mid week			Last week		
		Rec_0	Rec_1	BA	Rec_0	Rec_1	BA	Rec_0	Rec_1	BA
BIM	RF	0.61	0.95	0.78	0.79	0.85	0.82	0.80	0.85	0.83
	ET	0.60	0.95	0.77	0.80	0.82	0.81	0.81	0.82	0.81
	LR	0.60	0.95	0.78	0.78	0.86	0.82	0.80	0.86	0.83
	SVC	0.59	0.96	0.78	0.79	0.87	0.83	0.80	0.87	0.84
BD	RF	0.78	0.96	0.87	0.87	0.86	0.86	0.87	0.95	0.91
	ET	0.76	0.98	0.87	0.85	0.90	0.88	0.86	0.95	0.91
	LR	0.76	0.98	0.87	0.86	0.88	0.87	0.86	0.95	0.91
	SVC	0.76	0.98	0.87	0.85	0.90	0.87	0.85	0.95	0.90
CS	RF	0.78	1.00	0.89	0.90	0.90	0.90	0.90	1.00	0.95
	ET	0.78	1.00	0.89	0.89	0.90	0.89	0.89	1.00	0.95
	LR	0.78	1.00	0.89	0.90	0.90	0.90	0.90	1.00	0.95
	SVC	0.78	1.00	0.89	0.90	0.85	0.87	0.89	1.00	0.95
SP	RF	0.55	0.98	0.77	0.79	0.96	0.87	0.84	0.91	0.87
	ET	0.55	0.98	0.77	0.79	0.96	0.88	0.84	0.92	0.88
	LR	0.58	0.95	0.76	0.84	0.90	0.87	0.84	0.90	0.87
	SVC	0.55	0.98	0.77	0.79	0.96	0.87	0.84	0.91	0.87
TMF	RF	0.66	0.96	0.81	0.80	0.93	0.86	0.85	0.86	0.86
	ET	0.66	0.98	0.82	0.81	0.89	0.85	0.84	0.86	0.85
	LR	0.66	0.98	0.82	0.80	0.93	0.86	0.84	0.86	0.85
	SVC	0.66	0.98	0.82	0.81	0.89	0.85	0.84	0.86	0.85

5 Conclusion and Future Work

In this study, we found that students who paid for the course certificate were in general more engaged with the course content and interactive with their peers. We further compared four tree-based and regression classifiers to predict course purchasability based on learners' logged activities. Our proposed model achieved various balanced accuracies, ranging between 0.81 and 0.95. Taking into consideration the real-life challenge of the massively imbalanced classes in MOOCs, our method aimed to solving this issue using the data as-is, without further balancing. This is particularly competitive when considering that there could be many other factors influencing the financial decision, such as financial resources, need to document certification, which may have little to do with how students do during the course.

There are few experiments we are planning to conduct in the future. We will investigate the students' sentiments during the course, in order to infer if they correlate with the decision to purchase the certificate. This would be a promising research topic, taking advantage of recent developments in textual data analysis. This could further help in classifying students as early as possible, to provide them with timely intervention and guidance. Another avenue for further research is what to do when students have been categorised, if (and how) to lead them to certification.

References

1. Ng, A., Widom, J.: Origins of the modern MOOC (xMOOC). In: Hollands, F.M., Tirthali, D. (eds.) *MOOCs: Expectations and Reality: Full Report*, pp. 34–47 (2014)
2. Gardner, J., Brooks, C.: Student success prediction in MOOCs. *User Model. User Adap. Inter.* **28**(2), 127–203 (2018)
3. Alamri, A., et al.: Predicting MOOCs dropout using only two easily obtainable features from the first week's activities. In: Coy, A., Hayashi, Y., Chang, M. (eds.) *ITS*, pp. 163–173. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22244-4_20
4. Cristea, A.I., et al.: Earliest Predictor of Dropout in MOOCs: A Longitudinal Study of FutureLearn Courses. Association for Information Systems, Atlanta (2018)
5. Shah, D.: *By the Numbers: MOOCs in 2018* (2018)
6. Clow, D.: MOOCs and the funnel of participation. In: *Proceedings of the Third International Conference on Learning Analytics and Knowledge*. ACM, New York (2013)
7. Breslow, L., et al.: Studying learning in the worldwide classroom research into edX's first MOOC. *Res. Pract. Assessm.* **8**, 13–25 (2013)
8. Castaño-Muñoz, J., Kreijns, K., Kalz, M., Punie, Y.: Does digital competence and occupational setting influence MOOC participation? Evidence from a cross-course survey. *J. Comput. High. Educ.* **29**(1), 28–46 (2016)
9. Pursel, B.K., et al.: Understanding MOOC students: motivations and behaviours indicative of MOOC completion. *J. Comput. Assist. Learn.* **32**(3), 202–217 (2016)
10. Hansen, J.D., Reich, J.: Socioeconomic status and MOOC enrollment: enriching demographic information with external datasets. In: *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*. ACM, New York (2015)
11. Zhang, K.Z.K., Haiqin, X., Zhao, S., Yugang, Y.: Online reviews and impulse buying behavior: the role of browsing and impulsiveness. *Int. Res.* **28**(3), 522–543 (2018)
12. Dellarocas, C., Van Alstyne, M.W.: Money models for MOOCs. *Commun. ACM* **56**(8), 25–28 (2013)
13. Reich, J.: MOOC completion and retention in the context of student intent. *EDUCAUSE Rev.* **8** (2014)
14. Howarth, J., et al.: MOOCs to university: a consumer goal and marketing perspective. *J. Mark. High. Educ.* **27**(1), 144–158 (2017)
15. Jiang, S., et al.: Predicting MOOC performance with week 1 behavior. In: *Proceedings of the 7th International Conference on Educational Data Mining*, pp. 273–275 (2014)
16. Qiu, J., et al.: Modeling and predicting learning behavior in MOOCs. In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, New York (2016)
17. Ruipérez-Valiente, J.A., Cobos, R., Muñoz-Merino, P.J., Andujar, Á., Kloos, C.D.: Early prediction and variable importance of certificate accomplishment in a MOOC. In: Kloos, C.D., Jermann, P., Pérez-Sanagustín, M., Seaton, D.T., White, Su. (eds.) *EMOOCs*, pp. 263–272. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59044-8_31

18. Gitinabard, N., et al.: Your actions or your associates? Predicting certification and dropout in MOOCs with behavioral and social features. [arXiv:1809.00052](https://arxiv.org/abs/1809.00052) (2018)
19. Ramesh, A., et al.: Modeling learner engagement in MOOCs using probabilistic soft logic. In: Proceedings of the NIPS Workshop on Data Driven Education (2013)
20. Coleman, C.A., Seaton, D.T., Chuang, I.: Probabilistic use cases: discovering behavioral patterns for predicting certification. In: Proceedings of the Second ACM Conference on Learning@ Scale (2015)
21. Joksimović, S., et al.: Translating network position into performance: importance of centrality in different network configurations. In: Proceedings of the Sixth International Conference on Learning Analytics and Knowledge (2016)
22. Bin, X., Yang, D.: Motivation classification and grade prediction for MOOCs learners. *Comput. Intell. Neurosci.* **2016**, 1–7 (2016)
23. Alshehri, M., et al.: On the need for fine-grained analysis of gender versus commenting behaviour in MOOCs. In: Proceedings of the 3rd International Conference on Information and Education Innovations. ACM, New York (2018)
24. McKinney, W.: Data structures for statistical computing in Python. In: Proceedings of the 9th Python in Science Conference, Austin, TX (2010)
25. Oliphant, T.E.: A Guide to NumPy, vol. 1. Trelgol Publishing, New York (2006)
26. Agarwal, R.: The 5 Feature Selection Algorithms Every Data Scientist Should Know (2019). <https://towardsdatascience.com/the-5-feature-selection-algorithms-every-data-scientist-need-to-know-3a6b566efd2>. Accessed 30 Mar 2021
27. Perrier, A.: Feature Importance in Random Forests (2015). <https://alexisperrier.com/datascience/2015/08/27/feature-importance-random-forests-gini-accuracy.html>. Accessed 30 Mar 2021
28. Razali, N.M., Wah, Y.B.: Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *J. Stat. Model. Anal.* **2**(1), 21–33 (2011)
29. Mcnight, P.E., Najab, J.: Mann-Whitney U test. In: Weiner, I.B., Edward Craighead, W. (eds.) *The Corsini Encyclopedia of Psychology*. Wiley, Hoboken (2010)
30. Developers, S.-L.: Metrics and Scoring: Quantifying the Quality of Predictions (2007–2020). https://scikit-learn.org/stable/modules/model_evaluation.html#balanced-accuracy-score. Accessed 30 Mar 2021

Learner Behaviour



Recognizing Novice Learner's Modeling Behaviors

Sungeun An¹(✉), William Broniec¹, Spencer Rugaber¹, Emily Weigel¹,
Jennifer Hammock², and Ashok Goel¹

¹ Georgia Institute of Technology, Atlanta GA 30308, USA
sungeun.an@gatech.edu

² Smithsonian Institute, Washington DC 20002, USA

Abstract. Modeling is an important aspect of scientific problem-solving. However, modeling is a difficult cognitive process for novice learners in part due to the high dimensionality of the parameter search space. This work investigates 50 college students' parameter search behaviors in the context of ecological modeling. The study revealed important differences in behaviors of successful and unsuccessful students in navigating the parameter space. These differences suggest opportunities for future development of adaptive cognitive scaffolds to support different classes of learners.

Keywords: Ecological modeling · Modeling behaviors · Parameterization · Cognitive scaffolds · Learning analytics

1 Introduction

Scientific modeling is a complex cognitive process that requires integrating a variety of thinking skills and background knowledge in an investigative process [13]. Thus, studies examining middle school, high school, and college students' engagement with scientific modeling have highlighted a broad range of issues, including parameterization [15, 24, 27, 29]. Parameterization is the task of selecting values for a model's parameters and equations to define and/or test traits of a system's key behaviors [15, 24]. The parameterization task is often difficult due to a lack of domain knowledge and the high dimensionality of the parameter search space [26]. First, domain knowledge is required to constrain a range of possible values for a parameter (e.g., a sheep will usually give birth to between 1 and 2 litters). Second, parameter search strategies are required to systematically test the hypothetical changes in a model with the large number of parameters and the large range of values. As the parameterization in modeling is an important and difficult skill for novice learners, it is necessary to understand why it is difficult for them and how they struggle with it in order to provide them with cognitive support.

This paper is a preliminary step towards the creation of a learner model and technology-based cognitive scaffolding for scientific modeling. Thus, the goal of this

paper is to understand how novices explore the parameter space and identify successful/unsuccessful parameter search behaviors. The research questions associated with this effort were: 1) How do novices explore the parameter space? 2) How do parameter search behaviors relate to the success/unsuccess of the modeling task? Answering these questions requires discovering learning behavior patterns and developing a learner's mental model, which can be learned from data mining, especially learning analytics [8, 9].

In this study, we collected log data of 50 college students to observe their parameter search behaviors and identify modeling behavior patterns by comparing the differences between the groups who completed the task successfully and those who were unsuccessful. The publicly and freely available modeling environment called VERA was used in the experiment (<https://vera.cc.gatech.edu/>, [1, 2]). Although many studies have identified novices' difficulties in parameterization [15, 24, 27, 33] and developed cognitive scaffolds to support the parameterization task [4, 5, 12], they were limited due to the dependence on the predefined expert models, reference models, or data. Instead, we posit that interactive cognitive support should recognize the modelers' differing intentions and strategies as well as give personalized feedback according to the recognized behaviors to help them test various hypotheses and ideas.

Our contributions are threefold. First, the results complement the body of research on modeling behaviors for novice learner's success and struggles. Second, our log data study provides quantitative evidence for the model-fitting behaviors found in other protocol studies (for example, [15, 24]) and suggests that general-purpose cognitive support may be insufficient for many students. Lastly, insights about the novices' unproductive modeling behaviors suggest useful directions for designing adaptive scaffolds.

2 Related Work

2.1 Understanding Novices' Difficulties in Parameterization

Prior research has shown a number of difficulties for students doing quantitative modeling by presenting a detailed analysis of their cognitive processes [15, 24, 27]. The students typically struggled with defining and manipulating the system parameters and deciding what parameter values to use in their equations. Most students had a strong focus on adjusting model parameters to fit the empirical data or the given simulation output graph without deeply thinking about the system [24, 27]. Many students had a hard time understanding the indirect effects of manipulating the large number of simulation parameters and the large range of values that can appear in a model [15]. Consequently, the students tended to focus on the individual parameters separately instead of understanding the direct and/or indirect interactions among the components of a system as a whole [15, 24]. The students' difficulties in exploring the parameter search space have negative correlations with the quality of the model that students created [24]. Therefore, previous research emphasized the importance of adequate scaffolding that takes a top-down approach during parameterization so that students can focus on explaining the underlying mechanism [15, 24, 27].

Although these studies examined novices' difficulties during model parameterization due to the high dimensionality of the parameter search space, they did not necessarily

investigate why such difficulties emerge and how novices explore the parameter space. In this study, we investigate how learners manipulate the parameter values and how they use the output to guide adjustments to their models in detail to identify behavioral signals and build a learner model. In addition, previous studies typically used directed observations and verbal protocols to identify the difficulties of novices while working on a modeling task. In this study, we used students' interaction log data for detailed analysis that provides a more objective analysis.

2.2 Adaptive Scaffolding During the Parameterization Process

Cognitive scaffolding provides support to learners while they are learning a new task and enables them to do certain tasks that they may not be able to do without the support [10]. Adaptive scaffolding recognizes learners' behaviors, intervenes when they are in need of help, and reacts to different behaviors and issues during the task [19, 20]. Various scaffolding strategies have been proposed to help learners develop models, such as giving feedback and hints on the student's model as well as the student's modeling process. For example, sample equations have been given to support students' quantification of models along with the model diagrams they match and the output they yield [11, 15]. Real-world datasets have been given to help them set real-world quantities to use for parameters in their models [5, 15]. Expert models and reference models have served as ground truths to assess students' models and give feedback by comparing against behaviors generated by a correct expert model [4, 5, 28].

Most scaffolds only provide support with regard to setting up and defining the parameter values as defined scenarios, datasets, or reference models [4, 5, 11]. Typically, the system monitors modelers' models and gives corresponding feedback when there is a mismatch between the learners' models and the correct expert model or dataset [4, 5]. However, students may have different modeling goals, which sometimes do not match the example model (e.g., students may want to explore ecological collapse rather than stability). To support testing of new ideas or making novel hypotheses, adaptive scaffolding should also be provided during parameter exploration to support various modeling trajectories.

3 VERA for Ecological Modeling

VERA is an intelligent web-based ecological modeling application that allows learners to explore ecological systems and perform "what-if" experiments [1, 2]. In Fig. 1, the top image shows a screenshot of the model canvas in VERA where a learner can build a conceptual model by adding biotic, abiotic, and habitat components and defining the relationships among them. Conceptual models of ecological phenomena in VERA are expressed in the Component Mechanism Phenomenon (CMP) language [17] that derive from the Structure-Behavior-Function theory of modeling complex systems [14].

On the model canvas, the simulation parameters of each component that can affect its simulation behavior can be changed in the right panel. To help learners quantify the model, VERA uses the Smithsonian's Encyclopedia of Life (EOL) digital library to retrieve the structured data about the species and suggest the parameter values of its

lifespan, body mass, offspring count, reproductive maturity, etc. [2, 21]. VERA also uses genetic algorithms for parameter optimization to fit the model to the data. [7].

After constructing the conceptual model in the model canvas, VERA generates an agent-based NetLogo simulation (<https://ccl.northwestern.edu/netlogo/>, [31, 32]) based on the model and the simulation parameters (See the bottom image in Fig. 1) [18]. In this way, VERA integrates both qualitative reasoning in the conceptual model and quantitative reasoning in the agent-based simulation on one hand, and explanatory reasoning (conceptual model) and predictive reasoning (simulation) on the other. At the start of the COVID-19 pandemic, VERA Epidemiology (VERA-Epi) was created to support agent-based versions of compartmental epidemiology models [6]. Thus, the infrastructure of VERA has a degree of domain generality.

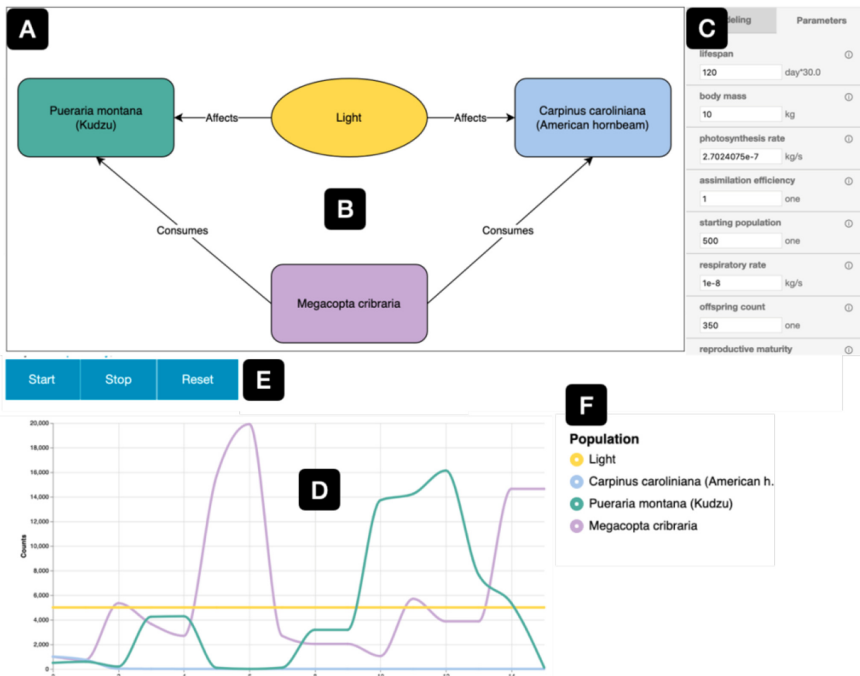


Fig. 1. The VERA system. (A) The model canvas, which provides a CMP model of the kudzu food web. (B) Model components. (C) Simulation parameters. (D) The simulation output graph – x axis: Time (months); y axis: Population. (E) Start, Stop, and Reset of the simulation output. (F) The model components on the simulation results screen.

4 Study

We conducted an experiment in a live classroom setting to understand how novices navigate the modeling parameter space while interacting with the modeling system. The study was conducted during one 50-min class period in an undergraduate biology class at Georgia Institute of Technology, a large, public R1 institution in the southeastern US.

4.1 Participants

A complete log data of 50 students who are enrolled in an Introductory Biology course in Fall 2019 was recorded ($N = 50$). Given the nature of the course and the students' self-assessments, the students were novice biologists and modelers who had limited biology knowledge or experience in modeling. On a 1–5 Likert scale, the average familiarity with biology was 2.80. The average self-perceived familiarity with modeling was only 2.22. The students did not receive any extra monetary compensation or course credit for their time. The students were asked to do this as an in-class exercise relevant to what they were learning for the course. Three researchers motivated students by moving around the classroom checking how they are doing and answering their questions. Additionally, two instructors of the course were sitting back in the classroom to observe the study. While the number of students enrolled in the class was 220, in our analysis we included only the students attended the class on the day of our intervention, performed the class activity, consented to study, and completed all of the assignments related to the intervention (e.g., pre-test, in-class test, and training session). Students who missed any of these steps were eliminated from our analysis.

4.2 Procedure

Before the day of the class intervention, the students took a biology pre-test as a class assignment to assess their baseline biology knowledge. During the intervention, we spent approximately 15 min training the students on the concept of scientific modeling and the use of the system. We introduced each of the modeling and simulation tabs and the meaning of each simulation parameter, and then walked through one scenario of building and revising a model. Next, the students were instructed to spend 25 uninterrupted minutes to complete a modeling task on a pre-built (kudzu) model (Fig. 1). The experiment instructions were given through a Qualtrics survey. After the exploration, students took an in-class biology test. All the students in the class used the modeling application on their own laptops during the study.

4.3 Modeling Task

Without knowing the effects of the values of the kudzu bug population (KBP) in advance, the students were asked to manipulate the population to select the best value for the ecosystem stability (e.g., making sure that kudzu, the kudzu bug, and American hornbeam all survive, creating a predator-prey cycle). The students were first asked to observe the simulation results of the initial model that manifests a fast-growing kudzu population. Then they answered three multiple-choice questions to test their understanding about the phenomenon. Then they were asked to alter the KBP between 1 and 1000 to provide what they thought to be the optimal value for the KBP for the stability of the ecosystem (in terms of kudzu, kudzu bug, and American hornbeam) and explain their reason in a short text. The initial model given to students manifested a fast-growing kudzu population when KBP is 1.

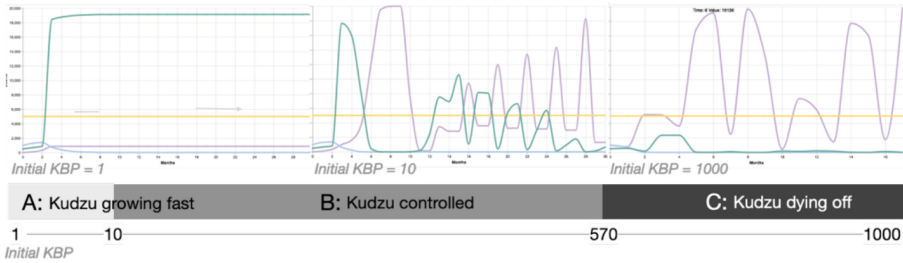


Fig. 2. The parameter spaces of the Kudzu Bug Population (KBP) and the simulation output graphs for each space.

4.4 Data

We analyzed the 50 students' log data and their submitted answers through Qualtrics. To use the students' biographic and school performance data, we obtained institute records to de-identify and pair the data obtained during the study and the class performance data. This was done in accordance with an Institute Review Board protocol (H18258). The class performance data included students' pre-class biology test and in-class biology test scores, and the score on the exercise questions that were given about the kudzu behaviors during the modeling task.

The students' log data during the modeling task was analyzed to create a set of features that were considered important and commonly used in prior work on analyzing and assessing behaviors [3, 8, 9, 25]. Along with the features derived from prior work, we created three new features to get additional information about the modeling behaviors. In particular, we selected 10 features to analyze different modeling behaviors including 1) *the total number of attempts*, 2) *time spent on simulation* (e.g., observing the simulation results), 3) *time spent on revision* (e.g., changing the parameter values for each iteration), 4) *the number of simulation pauses*, 5) *the median of the attempted values*, 6) *the number of the attempted values in false ranges* (e.g., out of the success range), 7) *redundancy* (e.g., revisiting previously explored ranges), and 8–10) *three test scores*.

Deviation was used to identify how evenly the students explored the space by calculating the standard deviation of the frequency of each space. For example, if student A tried three numbers in range between 10 to 570 (Parameter Space B in Fig. 2) and student B tried three numbers in range between 1–570 (Space A and B), student A will have a higher *deviation* than student B. *The number of explored spaces (num explored)* was created to identify how broadly the students explored the space by counting whether they explored each of the three result spaces. For example, *num explored* is 1 if he/she explored only one space (either A, B, or C), 2 if two spaces were explored, and 3 if all three spaces (A, B and C) were explored. *Success/Unsuccess* was created to determine whether the modeling task was successful or unsuccessful based on the students' answers. If the selected KBP value was between 10–570, it was given a score of 1 (Success); otherwise, 0 (not a success). Consequently, a total of 13 features were used for analysis.

4.5 Results

Dimension Reduction. We used lda as a dimension reduction technique to find a linear combination of the modeling features that were predictive of task success [13]. The first component of the lda model and the biology knowledge feature were used as a new set of features for the analysis. We scaled the feature values as few values have different quantities which would impact the linear regression algorithm. Figure 3 shows all students plotted with the biology knowledge and the first lda component with their respective success labels.

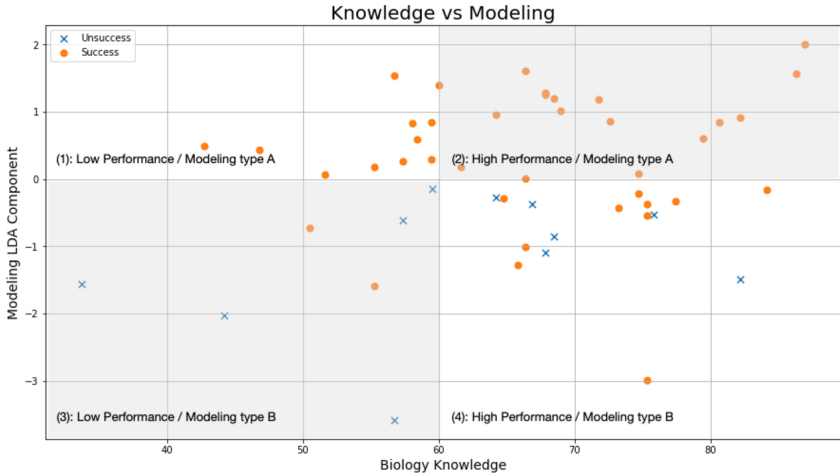


Fig. 3. The scatter plot based on the LDA component and biology knowledge.

As shown in Fig. 3, 78% of the students (N = 39) found the parameter value that fits in the successful range (expressed by orange “o”); 22% of the students (N = 11) did not find the right parameter range (expressed by blue “x”). The students are divided into four different categories and represented in each quadrant based on the performance and the modeling type: (1) low performance/modeling type A, (2) high performance/modeling type A, (3) low performance/ modeling type B, and (4) high performance/ modeling type B.

The task success strongly correlates with the modeling type ($r = 0.5265, p < 0.0001$) while it does not strongly correlate with the biology knowledge ($r = 0.1939, p = 0.1771$). Specifically, the modeling behavior A presented in quadrants 1 and 2 is considered a more successful behavior than that in quadrants 3 and 4. For example, among the modeling type A in quadrant 1 and 2, 100% of the students successfully completed the task whereas among the modeling type B, only 47.6% of the students successfully completed the task. Among the high-performance group, 81.5% of the students successfully completed the task. Among the low-performance group, only 72.22% were successful.

LDA Features and Modeling Features Correlations. We looked into how the original modeling features are correlated with the values of the first lda component and how much they contributed to the different modeling types a and b. All the features showed statistically significant correlations except for *deviation* ($r = -0.24$, $p < .5$). The most significant features predictive of task success were *the total number of attempts* ($r = -0.65$, $p < .001$), *the number of attempted values in false ranges* ($r = -0.72$, $p < .001$), and *the number of explored spaces* ($r = -0.56$, $p < .001$). Which are all negatively correlated with task success. The positively correlated modeling features are *the time spent on simulation* ($r = 0.29$, $p < .05$) and *the time spent on revision* ($r = 0.35$, $p < .05$), and *redundancy* ($r = 0.42$, $p < .005$).

Table 1. Summary of the parameter search patterns and descriptive statistics for successful and unsuccessful students. Values are means (std error in brackets).

Pattern	Relevant Feature	Successful	Unsuccessful
The students iterated more times	<i>The total number of attempts</i>	4.28 (1.50)	5.81 (2.56)
	<i>The number of simulation pauses</i>	2.92 (1.46)	3.30 (1.52)
The students spent less time in observing the simulation results and changing the parameter values	<i>The time spent on revision (normalized)</i>	134.72 (85.02)	105.61 (55.36)
	<i>The time spent on simulation (normalized)</i>	21.09 (17.09)	14.32 (4.87)
The students navigated in false ranges	<i>The number of explored spaces</i>	1.74 (0.63)	2.18 (0.40)
	<i>Deviation</i>	1.27 (0.63)	1.48 (0.88)
	<i>The number of the attempted values in false ranges</i>	0.89 (0.88)	1.81 (1.16)
The students revisited the already explored values and spaces	<i>Redundancy</i>	46.15% (18 out of 39)	72.72% (8 out of 11)

Modeling Behaviors. From the results, some patterns of parameter search can be derived. The unsuccessful modeling behavior type b was more wandering. This means that the students who fall into the modeling type b category iterated many times, and their attempted values were more likely to be concentrated in the false ranges as they navigated different parameter spaces. Table 1 is the summary of the parameter search patterns of unsuccessful students who show modeling type b (e.g., all unsuccessful students showed modeling type b, see Fig. 3). Note that the patterns of successful and unsuccessful students are in complete contrast to each other.

Doing many iterations is a commonly found behavior of novice search strategies in modeling [15, 24, 29]. Our study additionally reveals how and why the students' parameter search behavior is inefficient. The model-fitting behaviors observed by [15] and [24] were quantitatively observed (e.g., trying similar values on a certain space). In web search studies, [30] found two extreme learner groups: explorers and navigators, one being highly variable and one being highly consistent. Our results indicate somewhere in between showing both variability and consistency in their search interaction. For example, the students of model type A were consistent in that their attempted values were well balanced and less redundant, but also variable in that they tried broader space than the students of model type B. Nonetheless, we expect that results can be varied by task (e.g., well-defined task and complex sense-making tasks) and interface affordance (e.g., numeric input and slide bar).

4.6 Design Implications for Adaptive Cognitive Scaffolding

The above results provide insight into adaptive scaffolding for modeling based on the recognized parameter search behaviors and issues. The following design implications may also be applicable to other quantitative modeling tools or systems that require parameterization, including defining and adjusting the parameter values.

First, one common problem identified among unsuccessful students is that they repeatedly explore the similar values that produce similar simulation outputs. Agent-based models are stochastic, and the system behavior emerges out of interactions among a large number of components [23]. Consequently, the students have to test similar values many times to see their expected outcomes as it is difficult to predict which component and parameter value changes the system behavior significantly. In other words, the students heuristically have to learn the sensitivity of the parameters through trial and error as each parameter has a different degree of effect on the simulation results (e.g., some simulation parameters react more sensitively than the other parameters). For example, Fig. 2 shows discrete spaces for KBP that produce significantly different simulation behaviors. Such discrete spaces can be identified by automatically comparing the simulation outputs and using them to suggest different spaces.

Second, the students that were unsuccessful in the modeling task often explored a non-valid parameter search space. For example, we provided the students with a range of numbers with which to explore the parameter space, but without this constraint, it is more likely that students would take more time trying more numbers to find the valid space. While the learner can freely explore the parameter space by experimenting with various parameter values, the constraints can help the learner know whether his or her model makes sense in the real world and explore the parameter space more efficiently and effectively. In this process, the domain knowledge, such as the notion of exponential growth, logistic growth, and carrying capacity in ecology, can be leveraged to help narrow the parameterization space.

Third, the parameter values tried by the students were concentrated in one specific region of the space. In this paper, the parameter space was divided into three meaningful regions based on the kudzu behaviors (Fig. 2). Along with helping learners to search the parameter space, it is also important to have them understand the model as a whole by

having them exploring the three different parameter spaces rather than focusing on one space. Although these spaces were divided manually by the researchers, the similarities of the simulation results in different regions can be calculated to identify the distinct spaces. Then, the interactive tool can encourage learners to observe the unexplored spaces or to revisit the space to compare results, gain a deeper insight into structure of the parameter spaces, and see the meaningful patterns.

Last, while previous studies such as [4, 28] assumed that there were right or wrong models based on the expert or reference models, the learner can also try different values just to test new ideas or make new predictions, for example, to probe whether the model responds in predicted ways across a range of values. The system thus should be able to recognize what the learner is trying to achieve in the model to give appropriate guidance. For example, when searching the parameter space, increasing values can be a signal to suggest the range of values of the next parameter search space; decreasing values can be a signal to suggest the range of values of the previous search space.

5 Conclusion

We draw three preliminary conclusions from this research. First, our work confirms several findings from earlier work reported in the literature: parameterization in scientific modeling of complex phenomena is difficult because of the large number of parameters in a model and the large ranges of the values of the parameters, and that many learners struggle with parameterization. We observed this struggle even with college-level biology students. Second, general-purpose cognitive scaffolding in intelligent modeling environments like VERA is not sufficient for many students. The incompleteness and imprecision of the default values of the system parameters still leaves a large problem space of parameter values to be searched. Third, this suggests that intelligent learning environments need adaptive cognitive scaffolding to help learners navigate the large search spaces. The provision of heuristics for the search might be one such scaffolding yet to be evaluated.

In our study, we explored the parameter search strategies with one model component and parameter. Having the learners explore far more complex space (many components and many parameters) may give us different insights into parameter search strategies. This work is an early step in understanding learners' parameterization search patterns and leaves many exciting questions to be answered with further research. The ability to classify a learner's search strategy is an interesting problem on its own, but a learner-specific adaptive interface could test the feasibility of applying this type of results in real-time and build learner profiles that will enable personalized interaction.

Acknowledgements. This research was supported by an US NSF grant #1636848 (Big Data Spokes: Collaborative: Using Big Data for Environmental Sustainability: Big Data + AI Technology = Accessible, Usable, Useful Knowledge!) and the NSF South BigData Hub. This research was conducted in accordance with the Institute Review Board protocol #H18258. We thank Robert Bates for his help in constructing the VERA system. We thank anonymous reviewers of previous drafts for their helpful comments and suggestions.

References

1. An, S., Bates, R., Hammock, J., Rugaber, S., Goel, A.: VERA: popularizing science through AI. In: Proceedings of the International Conference on Artificial Intelligence in Education, pp. 31–35. Springer, Cham, June 2018
2. An, S., Bates, R., Hammock, J., Rugaber, S., Weigel, E., Goel, A.: Scientific modeling using large scale knowledge. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) AIED 2020. LNCS (LNAI), vol. 12164, pp. 20–24. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52240-7_4
3. Aula, A., Khan, R.M., Guan, Z.: How does search behavior change as search becomes more difficult? In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 35–44 (2010)
4. Basu, S., Biswas, G., Kinnebrew, J.S.: Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Model. User Adap. Inter.* **27**(1), 5–53 (2017)
5. Bridewell, W., Sanchez, J.N., Langley, P., Billman, D.: An interactive environment for the modeling and discovery of scientific knowledge. *Int. J. Human Comput. Stud.* **64**(11), 1099–1114 (2006)
6. Broniec, W., An, S., Rugaber, S., Goel, A.K.: Using VERA to explain the impact of social distancing on the spread of COVID-19. arXiv preprint [arXiv:2003.13762](https://arxiv.org/abs/2003.13762) (2020)
7. Broniec, W., An, S., Rugaber, S., Goel, A.K.: Guiding parameter estimation of agent-based modeling through knowledge-based function approximation. In: Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021) - Stanford University, Palo Alto, California, USA, 22–24 March 2021
8. Buckley, B.C., Gobert, J.D., Horwitz, P.: Using log files to track students' model based inquiry. In: Proceedings of the 7th International Conference on Learning Sciences, pp. 57–63 (2006)
9. Buckley, B.C., Gobert, J.D., Horwitz, P., O'Dwyer, L.M.: Looking inside the black box: assessing model-based learning and inquiry in BioLogica. *Int. J. Learn. Technol.* **5**(2), 166–190 (2010)
10. Collins, A., Brown, J.S., Newman, S.E.: Cognitive apprenticeship: teaching the craft of reading, writing and mathematics. *Thinking. J. Philos. Child.* **8**(1), 2–10 (1988)
11. De Jong, T., Van Joolingen, W.R.: Scientific discovery learning with computer simulations of conceptual domains. *Rev. Educ. Res.* **68**(2), 179–201 (1998)
12. Duque, R., Bollen, L., Anjewierden, A., Bravo, C.: Automating the analysis of problem-solving activities in learning environments: the co-lab case study. *J. Univ. Comput. Sci.* **18**(10), 1279–1307 (2012)
13. Flick, L.B.: Cognitive scaffolding that fosters scientific inquiry in middle level science. *J. Sci. Teacher Educ.* **11**(2), 109–129 (2000)
14. Goel, A.K., Rugaber, S., Vattam, S.: Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language. *AIEDAM* **23**(1), 23–35 (2009)
15. Hogan, K., Thomas, D.: Cognitive comparisons of students' systems modeling in ecology. *J. Sci. Educ. Technol.* **10**(4), 319–345 (2001)
16. Hunter, E., Mac Namee, B., Kelleher, J.D.: A comparison of agent-based models and equation based models for infectious disease epidemiology. In: Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science (AICS), pp. 33–44 (2018)
17. Joyner, D.A., Goel, A.K., Rugaber, S., Hmelo-Silver, C., Jordan, R.: Evolution of an integrated technology for supporting learning about complex systems. In: Proceedings of the IEEE 11th International Conference on Advanced Learning Technologies, pp. 257–259. IEEE July 2011
18. Joyner, D.A., Goel, A.K., Papin, N.M.: MILA--S: generation of agent-based simulations from conceptual models of complex systems. In: Proceedings of the 19th International Conference on Intelligent User Interfaces, pp. 289–298, February 2014

19. Joyner, D.A., Goel, A.K.: Improving inquiry-driven modeling in science education through interaction with intelligent tutoring agents. In: Proceedings of the 20th International Conference On Intelligent User Interfaces, pp. 5–16, March 2015
20. Liu, J., Wong, C.K., Hui, K.K.: An adaptive user interface based on personalized learning. *IEEE Intell. Syst.* **18**(2), 52–57 (2003)
21. Martinez, A.M., Kak, A.C.: PCA versus LDA. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(2), 228–233 (2011)
22. Parr, C. S., et al.: The encyclopedia of life v2: providing global access to knowledge about life on earth. *Biodivers. Data J.* (2) (2014)
23. Railsback, S.F., Grimm, V.: *Agent-Based and Individual-Based Modeling: a practical introduction*. Princeton University Press (2019)
24. Sins, P.H., Savelsbergh, E.R., vanJoolingen, W.R.: The difficult process of scientific modelling: an analysis of novices' reasoning during computer-based modelling. *Int. J. Sci. Educ.* **27**(14), 1695–1721 (2005)
25. Tabatabai, D., Shore, B.M.: How experts and novices search the web. *Libr. Inf. Sci. Res.* **27**(2), 222–248 (2005)
26. Thiele, J.C., Kurth, W., Grimm, V.: Facilitating parameter estimation and sensitivity analysis of agent-based models: a cookbook using NetLogo and R. *J. Artif. Soc. Soc. Simul.* **17**(3), 11 (2014)
27. VanLehn, K.: Model construction as a learning activity: a design space and review. *Interact. Learn. Environ.* **21**(4), 371–413 (2013)
28. Vattam, S.S., et al.: Understanding complex natural systems by articulating structure-behavior-function models. *J. Educ. Technol. Soc.* **14**(1), 66–81 (2011)
29. White, B.Y., Frederiksen, J.R.: Inquiry, modeling, and metacognition: making science accessible to all students. *Cogn. Instr.* **16**(1), 3–118 (1998)
30. White, R.W., Drucker, S.M.: Investigating behavioral variability in web search. In: Proceedings of the 16th International Conference on World Wide Web, pp. 21–30 (2007)
31. Wilensky, U., Resnick, M.: Thinking in levels: a dynamic systems approach to making sense of the world. *J. Sci. Educ. Technol.* **8**(1), 3–19 (1999)
32. Wilensky, U., Reisman, K.: Thinking like a wolf, a sheep, or a firefly: learning biology through constructing and testing computational theories—an embodied modeling approach. *Cogn. Instr.* **24**(2), 171–209 (2006)
33. Wu, H.K.: Modelling a complex system: using novice-expert analysis for developing an effective technology-enhanced learning environment. *Int. J. Sci. Educ.* **32**(2), 195–219 (2010)



Expert, Novice, and Intermediate Performance: Exploring the Relationship Between Clinical Reasoning Behaviors and Diagnostic Performance

Alejandra Ruiz-Segura^(✉)  and Susanne P. Lajoie 

McGill University, 3700 McTavish Street, Montreal, QC, Canada
alejandra.ruizsegura@mail.mcgill.ca

Abstract. Understanding the diagnostic process and the interplay between gathering and interpreting information can reduce the inaccuracies that lead to medical errors. In this study, we examined the relationship between medical students' ($n = 46$) performance profiles and the type of clinical reasoning behaviors they executed while diagnosing a clinical patient in the context of an intelligent tutoring system, BioWorld [2]. Performance was measured by efficiency (similarity to an expert solution), confidence, and time. We found three groups: high, low, and intermediate performance. High-performing students were characterized by high efficiency, intermediate students had average efficiency and confidence, and low performing students were more characterized by low confidence rather than their efficiency score. We found that the high performers put more effort in integrating elements of the clinical case, a deep learning strategy. Unexpectedly, the high and intermediate groups additionally selected more information from the patient history, a shallow learning strategy. Our findings contribute to understanding of learning of clinical reasoning skills using an intelligent tutoring system.

Keywords: Diagnostic performance · Clinical reasoning · Medical students · Expertise

1 Introduction

Medical errors are the 3rd cause of death in the US [3]. The cause of medical errors in 74% of cases are associated with inaccurate synthesis and interpretation of information [4]. These mistakes are more likely to occur among novice physicians who engage in simple steps in the diagnostic process instead of connecting and contrasting pieces of information [5, 6]. Therefore, understanding the diagnostic process and accounting for the interplay between gathering and interpreting information of the clinical cases can significantly reduce the inaccuracies that lead to medical errors. In an attempt to contribute to this understanding, we explore the relationship between engagement in clinical reasoning behaviors and diagnostic performance.

1.1 Clinical Reasoning Behaviors

Clinical reasoning is the process by which health professionals diagnose patients [7]. It implies a complex interplay between gathering and interpreting information [8]. Previous research demonstrated that successful diagnosis is influenced by executing different steps in the clinical reasoning process [9, 10].

When approaching a clinical case, physicians need to review the description of the patient history and detect critical symptoms and relevant background events [5, 7, 9]. Additionally, physicians collect supporting information from literature and order lab tests to confirm their diagnoses [7, 10]. Usually, physicians consider more than one diagnosis and compare hypotheses by linking the patient symptoms, literature information, and results from the lab tests [6, 10]. During this process, physicians classify the information as supportive, neutral, or contradicting, and prioritize which factors are the most relevant for making a final diagnosis [5, 9, 11].

The clinical reasoning actions taken by medical students reflect different learning strategies used when approaching a patient [6]. It is more common to see shallow learning strategies among novice students which include information acquisition behaviors, that involve collecting evidence to support a diagnosis from the patient history, literature search, and ordering lab tests [5, 7, 10]. Students with expert like behaviors execute deep learning strategies, i.e., information transformation behaviors, that involve differentiating the relevant from irrelevant information, generating appropriate hypotheses, linking evidence, categorizing and prioritizing the most relevant information [6, 10, 12]. Low performing medical students tend to rely on shallow learning strategies, and high performing students rely on deeper learning strategies which leads them to more accurate diagnosis [6].

1.2 Expert-Like and Novice Performance

Experts are characterized as being fast and accurate, consistently demonstrate superior performance, and are able to distinguish relevant from irrelevant information [8, 13, 14]. During the process of clinical reasoning, expert physicians compare potential diagnoses, and prioritize the most critical signs and symptoms, relying on experience [6, 10, 15].

Novice medical trainees, on the other hand, are less strategic and engage in less metacognitive effort: they focus more on collecting data from patient history and reviewing medical literature [5, 6], without connecting these pieces of information. Moreover, novice physicians commonly make an error that can be termed “premature closure” in which they stick with their first diagnosis, instead of contrasting this diagnosis with other possibilities that could result in a different explanation [15].

In the clinical reasoning research students are usually classified a priori based on performance results and the qualities of the clinical reasoning process are explored afterwards [i.e., 12]. As such, performance is normally divided into two groups high and low performance, mimicking expert-like or novice performance [6, 15, 16]. However, recent empirical research showed that intermediate groups reflect relevant differences that could lead to a better understanding of medical students’ diagnostic performance [17]. Moreover, literature has reinforced the importance of accounting for different measures of performance to have a better understanding of the factors that influence the medical students’ diagnostic skills [12, 18].

1.3 Intelligent Tutoring Systems to Understand Clinical Reasoning

Medical education increasingly uses simulators and computer-based learning environments to teach medical skills such as clinical reasoning. Intelligent tutoring systems (ITS) are particularly designed to support instructional and learning processes [1]. The uses of these technologies allow medical students to practice in authentic environments, with no risk of damaging a patients' life [2]. With the growing use of simulators and ITS in learning, theories like the cognitive theory of multimedia learning [19] expose the need to understand how the student cognition is impacted by the system characteristics. Thus, this study aims to contribute to the understanding of medical students clinical reasoning performance as they interact with an ITS.

Moreover, we examine the question of how an ITS for clinical reasoning can impact medical students confidence when diagnosing virtual patients, since deliberate practice [13] is expected to increase knowledge and expert like skills, thus augmenting confidence when executing the activity. However, previous research has identified mixed patterns of medical students' confidence when learning clinical reasoning with ITS. On one study, confidence was high in cases of different levels of difficulty, as such, it was aligned for easy cases with correct diagnosis, however, in difficult cases students had a misalignment of overconfidence confidence with an incorrect diagnosis [20]. Another study did not show significant changes in confidence despite this association was expected [21].

We argue that it is relevant to understand differences in diagnostic performance along a continuum to better measure and understand the clinical reasoning process in the context of an ITS. To contribute to this understanding, the current study aims to classify students based on different measures of performance. Particularly, based on similarity to an expert solution, confidence in final diagnosis, and time solving the case. We aim to understand differences in execution of different clinical reasoning behaviors while diagnosing a virtual patient in an ITS.

2 Research Questions and Hypotheses

We pose the following research questions when diagnosing a virtual patient in an ITS:

RQ1. Do performance measures during a clinical reasoning task cluster (i.e. group) medical students in a meaningful way?

RQ2. Is there a significant difference in frequency of clinical reasoning behaviors between groups of students clustered by different measures of performance in a clinical reasoning task?

H1. We expect to find three clusters: two opposing clusters and a third intermediate cluster. One cluster will reflect low performance with low similarity to an expert solution (z-scores around -1 in efficiency score), low confidence (z-scores around -1 in confidence), and high time investment (z-scores > 1). A second cluster will reflect high performance with high similarity to an expert solution (z-score around 1), high confidence (z-score around 1) and low time investment solving the clinical case (z-score around -1). A third cluster will group intermediate students (z-scores around ± 0.5 on all behaviors). We expect a third group since earlier research showed intermediate clusters present different performance characteristics [17].

H2. We expect that the low performance cluster will engage in more shallow learning strategies reflected by higher execution of patient history, literature search, and ordering lab tests, whereas the high-performance will cluster will engage in more behaviors related to deep learning strategies, which will be inferred from occurrence of the behaviors labeled as hypothesis generation, linking evidence and integration. We take an exploratory approach on the intermediate cluster and thus do not have a specific hypothesis.

3 Methods

The data used in this analysis was part of a larger project, which obtained REB approval from a North American University.

3.1 Participants

46 medical students volunteered to take part in the study (23 female, 13 males, and 10 not reported), with an average age of 23.61 ($SD = 3.11$). Gender among groups did not have a significant statistical difference.

3.2 Task: Diagnosing a Virtual Patient in BioWorld

Participants diagnosed virtual patients in BioWorld [2], an ITS [22] (see Fig. 1). BioWorld allows students to deliberately practice [13] clinical reasoning skills in a safe and authentic virtual environment [2]. The BioWorld interface is a research and learning tool designed to scaffold students to follow expert-like steps to diagnose virtual patients. Medical students start by reading the patient case, in which they can highlight relevant symptoms and background information to set potential diagnoses. Students can obtain supporting literature in an online library and request lab tests that can aid them in formulating a differential hypothesis. Students can also compare different potential diagnoses, link evidence from the collected information from the patient history, library, and laboratory tests, and they set a confidence percentage for each diagnosis. When the students select a final diagnosis, they categorize and prioritize the supporting evidence that led them to their choice and write a case summary that synthesizes the most relevant information. BioWorld as an ITS, individualizes feedback by using a novice-expert overlay system, highlighting similarities and differences comparing evidence items of the student and the expert solutions [22]. The system shows the case accuracy and efficiency compared to an expert solution. Accuracy refers to correctness of final diagnosis, students see a green checkmark if their final diagnosis is correct, or a red cross in case it is different from the expert solution. Efficiency is calculated by similarity to an expert solution accounting for categorized and prioritized evidence, students see the percentage of similarity calculated by comparing the information selected by them and by the expert. Students can read the expert's summary to understand their reasoning process.

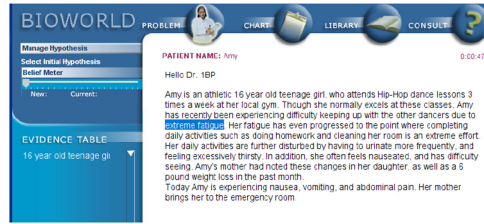


Fig. 1. Screenshot of BioWorld interface

3.3 Measures

Student interaction with BioWorld is saved in a system log file. The log file records students' actions, which were used to categorize clinical reasoning behaviors, time stamps, and performance measures such as final diagnosis accuracy (correctness), confidence, and efficiency.

Performance. For the current study, the selected measures of performance were efficiency, confidence, and time. The students select their confidence in final diagnosis on a scale of 0 to 100. Efficiency is calculated by similarity of evidence compared to an expert solution in a score from 0 to 100. Time was calculated by subtracting the time of final diagnosis minus start of the case.

Clinical Reasoning Behaviors. Clinical reasoning behaviors were coded based on the actions in the log file reflecting six behaviors: adding patient history, library search, ordering lab tests, selecting hypothesis, linking evidence, and integrating (categorizing and prioritizing) case information. For controlling for time difference, relative frequencies were calculated as if students spent one hour solving the case. Raw frequency of the participants' behaviors was divided by time on the case and the resulting number was multiplied by 60.

3.4 Data Analyses

Data Screening. For finding univariate outliers the behaviors and performance measures were transformed to z-scores in IBM SPSS. The behaviors of adding patient history, laboratory tests, library search, linking evidence, and integration had univariate outliers, identified by z-scores above ± 3.2 . Outliers were replaced with the closest non-outlier score [23].

Performance-Clusters Extraction and Differences in Clinical Reasoning Behaviors.

For answering RQ1, a K-clusters analysis was conducted on three measures of performance while solving a clinical reasoning task: efficiency, confidence, and time. Measures of performance were converted to z-scores to conduct a K-means cluster analysis in IBM SPSS. The number of clusters (i.e., three) was selected based on earlier empirical work [17]. To answer RQ2 and find group differences in frequency of clinical reasoning behaviors, a series of analysis of variance (ANOVA) were conducted.

4 Results

Table 1 shows descriptive statistics of performance measures and clinical reasoning behaviors.

Table 1. Descriptive statistics of clinical reasoning performance and behaviors

Variable name	Mean	SD
Efficiency	53.24	17.78
Confidence	74.65	17.82
Time	23.81	10.26
Hypothesis	37.61	21.39
Integrate	105.8	58.30
Library	17.89	18.18
Linking	37.96	35.11
Patient history	41.61	16.25
Order lab test	37.02	17.28

For answering RQ1 a three-cluster k-means cluster analysis was conducted. The analysis required three iterations to create differentiated clusters, which is considered acceptable as it successfully showed different clusters in a small number of iterations. Cluster centers were found by z-scores and were used to interpret and label each cluster. Students in cluster one ($n = 17$) had the highest efficiency scores ($z = .83$), average confidence ($z = .50$), and spent the lowest time solving the case ($z = -.68$). Students in the second cluster ($n = 19$) had an average efficiency ($z = -.46$), average confidence ($z = .33$), and were the ones that invested the most time in the case compared to the other groups ($z = .68$). Finally, the third cluster ($n = 10$) had the lowest, yet average, efficiency ($z = -.54$), the lowest confidence ($z = -1.48$), and average time ($z = -.13$). In conclusion, cluster one was labeled as high performance, cluster two as intermediate performance, and cluster three as low performance.

Analyses of variance (ANOVA) were conducted to find group differences (IV: cluster membership) in frequency of the different clinical reasoning behaviors (DV: frequency of clinical reasoning behaviors). Equal variances were assumed with a non-significant homogeneity of variance for all the behaviors with a *Levene's F* (2, 43) ranging from 1.21 to 1.87, and a p -value ranging from .213 to .317. The results of the ANOVAs revealed a significant difference in the behaviors of integrating information ($F(2, 43) = 1.05, p < .001$) and adding patient history ($F(2, 43) = 17.19, p < .001$).

Post-hoc analyses using LSD were conducted for the clinical reasoning behaviors that had a significant difference across groups. For integrating information, participants in the high performing group ($M = 154.82, SD = 40.67$) had a higher frequency of this behavior compared to the intermediate ($M = 76.52, SD = 54.03; p < .001$), and low performing group ($M = 105.87, SD = 52.30; p < .001$). For adding patient history,

the high-performance group ($M = 54.47$, $SD = 12.29$) had a higher frequency of this behavior compared to the intermediate group ($M = 30.21$, $SD = 10.41$; $p < .001$), and the low performing group ($M = 41.40$, $SD = 15.76$; $p = .011$). Additionally, the intermediate group significantly added more patient information compared to the low performing group ($p = .026$).

5 Discussion

The findings successfully supported the hypothesis that medical students can be clustered by different measures of performance. As expected (H1), three groups were found: high performance, low performance, and intermediate performance. In line with our hypothesis, the high-performance cluster had the highest efficiency scores, and invested the lowest time in the case. However, the findings did not support the expectation of high confidence. Similarly, the intermediate cluster met the hypothesized measures, with average scores in confidence and efficiency; contrary to our hypothesis, this cluster spent more time solving the case. Finally, the low performance cluster had the lowest confidence and efficiency score, yet spent average time.

The expectations about group differences in frequency of clinical reasoning behaviors were partially supported (H2). As expected, the high performing group engaged more in integration, a deep learning behavior, compared to the intermediate and low performing clusters. Contrary to our hypothesis, the high performing group added more items from the patient history, a shallow learning behavior, compared to the intermediate and low performance clusters. Moreover, the intermediate cluster additionally had a higher frequency of adding patient history items compared to the low performance group. This finding might be explained by the fact that medical students in the high and intermediate clusters collected more information from the patient history, and were capable of connecting patient history data with the final result. The high performing group also invested more effort categorizing and prioritizing information, thus, students in this group took more information from the patient history, yet put efforts to connect the gathered information in a more expert-like manner. On the contrary, students in the low performing cluster likely focused less relevant information and put less effort in connecting information [6, 10, 15, 16].

Our findings revealed that low performing students had low confidence, which characterized them more than their efficiency score. In medical fields, confidence has been associated to more effective use of clinical skills and success in clinical practice [24]. Our findings do not align with previous research that identified overconfidence with poor performance levels when diagnosing virtual patients [20]. In a high-stakes field as medicine, confidence is a relevant factor when approaching a patient and, in this case, diagnosing correctly; therefore, more research in this area is needed to clarify medical students' confidence when diagnosing virtual patients in an ITS. We suggest medical instructors to create interventions to increase medical students' metacognitive skills to assess critically their knowledge and confidence when approaching a clinical case, and improve their diagnostic skills.

A significant limitation of our study is the small sample size ($n = 46$). Future research should account for a larger sample that can potentially identify clearer differences among clusters and among the different clinical reasoning behaviors. Beyond the small sample size, case-correctness could not be calculated. Future research should also explore if the proposed clusters differ in diagnostic accuracy. Unfortunately, we could not access previous experience of the participants. However, future studies should control for years of education and practice to identify if these factors influence the results.

6 Conclusion

This study confirms that different measures of clinical reasoning performance (i.e., confidence, efficiency, and time) can account for student differences and can be used to classify medical students. Particularly, we successfully identified a high, intermediate, and low performance group. Such groups differed in the way they approach the clinical reasoning task, reflecting different engagement in learning strategies when diagnosing a virtual patient in an ITS. The findings support the relevance for medical students to improve diagnostic skills by management of time, efficient learning strategies and confidence in their own clinical reasoning skills.

Our findings contribute to understanding the clinical reasoning process medical students use when learning to diagnose a patient in an ITS. Particularly, we confirmed that medical students with more expert-like performance engaged in more synthesizing and interpreting information, which is likely to lead to fewer medical errors. Moreover, we identified that novice students had lower confidence. In an attempt to improve medical students' diagnostic skills, we suggest that medical instructors create interventions for medical students to deliberate practice [13] clinical reasoning skills, particularly integrating case information, and increasing confidence about their knowledge and skills. Intelligent tutoring systems [1] can serve as a mean for medical students to practice in safe authentic environments.

References

1. Lajoie, S.P., Azevedo, R.: Teaching and learning in technology-rich environments. In: Alexander, P.A., Winne, P.H. (eds.) *Handbook of Educational Psychology*, pp. 803–821. Lawrence Erlbaum Associates Publishers, Mahwah (2006)
2. Lajoie, S.P.: Developing professional expertise with a cognitive apprenticeship model: examples from avionics and medicine. In: Ericsson, K.A. (ed.) *Development of Professional Expertise: Toward Measurement of Expert Performance and Design of Optimal Learning Environments*, pp. 61–83. Cambridge University Press, Cambridge (2009)
3. Makary, M.A., Daniel, M.: Medical error—the third leading cause of death in the US. *BMJ* **353**, i2139 (2016). <https://doi.org/10.1136/bmj.i2139>
4. Graber, M.L., Franklin, N., Gordon, R.: Diagnostic error in internal medicine. *Arch. Intern. Med.* **165**(13), 1493–1499 (2005). <https://doi.org/10.1001/archinte.165.13.1493>
5. Artino, A.R., Cleary, T.J., Dong, T., Hemmer, P.A., Durning, S.J.: Exploring clinical reasoning in novices: a self-regulated learning microanalytic assessment approach. *Med. Educ.* **48**(3), 280–291 (2014). <https://doi.org/10.1111/medu.12303>

6. Zheng, J., Li, S., Lajoie, S.P.: The Role of achievement goals and self-regulated learning behaviors in clinical reasoning. *Technol. Knowl. Learn.* **25**(3), 541–556 (2019). <https://doi.org/10.1007/s10758-019-09420-x>
7. Kuiper, R.A.: Integration of innovative clinical reasoning pedagogies into a baccalaureate nursing curriculum. *Creat. Nurs.* **19**(3), 128–139 (2013). <https://doi.org/10.1891/1078-4535.19.3.128>
8. Lajoie, S.P.: Learning science applications for research in medicine. In: Lin, L., Spector, J.M. (eds.) *The Sciences of Learning and Instructional Design*, pp. 108–118. Routledge (2017)
9. Durning, S.J., et al.: The feasibility, reliability, and validity of a post-encounter form for evaluating clinical reasoning. *Med. Teach.* **34**(1), 30–37 (2012). <https://doi.org/10.3109/0142159X.2011.590557>
10. Lubarsky, S., Dory, V., Audétat, M.-C., Custers, E., Charlin, B.: Using script theory to cultivate illness script formation and clinical reasoning in health professions education. *Can. Med. Educ. J.* **6**(2), e61–e70 (2015)
11. Jarrell, A.: The emotional twists and turns of problem solving: an examination of learners' behavioral, psychological and experiential emotion responses to unexpected events during problem solving. Masters thesis, McGill University, Montreal (2015)
12. Li, S., Zheng, J., Poitras, E., Lajoie, S.: The Allocation of time matters to students' performance in clinical reasoning. In: Nkambou, R., Azevedo, R., Vassileva, J. (eds.) *ITS 2018. LNCS*, vol. 10858, pp. 110–119. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91464-0_11
13. Ericsson, K.A.: The influence of experience and deliberate practice on the development of superior expert performance. In: Ericsson, K.A., Charness, N., Feltovich, P.J., Hoffman, R.R. (eds.) *The Cambridge Handbook of Expertise and Expert Performance*, pp. 683–704. Cambridge University Press, Cambridge (2006)
14. Glaser, R.: Changing the agency for learning: acquiring expert performance. In: *The Road to Excellence: The Acquisition of Expert Performance in the Arts and Sciences, Sports, and Games*, pp. 303–311. Lawrence Erlbaum Associates, Inc, Hillsdale (1996)
15. Eva, K.W.: What every teacher needs to know about clinical reasoning. *Med. Educ.* **39**(1), 98–106 (2005). <https://doi.org/10.1111/j.1365-2929.2004.01972.x>
16. Lajoie, S.P., Zheng, J., Li, S., Jarrell, A., Gube, M.: Examining the interplay of affect and self regulation in the context of clinical reasoning. *Learn. Instr.* **72**, 101219 (2019). <https://doi.org/10.1016/j.learninstruc.2019.101219>
17. Jarrell, A., Harley, J.M., Lajoie, S., Naismith, L.: Success, failure and emotions: examining the relationship between performance feedback and emotions in diagnostic reasoning. *Educ. Tech. Res. Dev.* **65**(5), 1263–1284 (2017). <https://doi.org/10.1007/s11423-017-9521-6>
18. Lajoie, S.P., Zheng, J., Li, S.: Examining the role of self-regulation and emotion in clinical reasoning: implications for developing expertise. *Med. Teach.* **40**(8), 842–844 (2018). <https://doi.org/10.1080/0142159X.2018.1484084>
19. Parong, J., Mayer, R.E.: Cognitive and affective processes for learning science in immersive virtual reality. *J. Comput. Assist. Learn.* **37**(1), 226–241 (2020). <https://doi.org/10.1111/jcal.12482>
20. Lajoie, S.P., et al.: Technology-rich tools to support self-regulated learning and performance in medicine. In: Azevedo, R., Alevan, V. (eds.) *International Handbook of Metacognition and Learning Technologies*. SIHE, vol. 28, pp. 229–242. Springer, New York (2013). https://doi.org/10.1007/978-1-4419-5546-3_16
21. Kim, J.Y., Kim, E.J.: Effects of simulation on nursing students' knowledge, clinical reasoning, and self-confidence: a quasi-experimental study. *Korean J. Adult Nurs.* **27**(5), 604–611 (2015). <https://doi.org/10.7475/kjan.2015.27.5.604>

22. Poitras, E.G., Lajoie, S.P., Doleck, T., Jarrell, A.: Subgroup discovery with user interaction data: an empirically guided approach to improving intelligent tutoring systems. *Educ. Technol. Soc.* **19**(2), 204–214 (2016)
23. Meyers, L.S., Gamst, G., Guarino, A.J.: *Applied Multivariate Research: Design and Interpretation*. SAGE Publications, Thousand Oaks (2016)
24. Lundberg, K.M.: Promoting self-confidence in clinical nursing students. *Nurse Educ.* **33**(2), 86–89 (2008). <https://doi.org/10.1097/01.NNE.0000299512.78270.d0>



Agent-Based Simulation of the Classroom Environment to Gauge the Effect of Inattentive or Disruptive Students

Khulood Alharbi¹(✉), Alexandra I. Cristea¹, Lei Shi¹, Peter Tymms²,
and Chris Brown²

¹ Computer Science, Durham University, Durham, UK
{khulood.o.alharbi, alexandra.i.cristea, lei.shi}@durham.ac.uk

² School of Education, Durham University, Durham, UK
{p.b.tymms, chris.brown}@durham.ac.uk

Abstract. The classroom environment is a major contributor to the learning process in schools. Young students are affected by different details in their academic progress, be it their own characteristics, their teacher's or their peers'. The combination of these factors is known to have an impact on the attainment of young students. However, what is less known are ways to accurately measure the impact of the individual variables. Moreover, in education, predicting an end-result is not enough, but *understanding the process* is vital. Thus, in this paper, we simulate the interactions between these factors to offer education stakeholders – administrators and teachers, in a first instance – the possibility of understanding how their activities and the way they manage the classroom can impact on students' academic achievement and result in different learning outcomes. The simulation is based on data from Performance Indicator in Primary Schools (PIPS) monitoring system, of 65,385 records that include 3,315 classes from 2,040 schools, with an average of 26 students per class collected in 2007. The results might serve teachers in solving issues that occur in classrooms and improve their strategies based on the predicted outcome.

1 Introduction

Young students form the bases of our societies. The way they interact with their environment and how it affects their achievement has been an interest of literature for years [4, 5, 29]. It is important to provide young students at such a young age with a respectful and suitable environment for learning, to eliminate the disturbances or minimise them when they occur. Creating this desired environment requires the full understanding of the interactions and their anticipated consequences in classrooms.

Interestingly, however, the literature on classroom simulation is limited. A relatively recent attempt by Ingram and Brooks [15] aimed to understand specifically the effect of seating and friendship groups on attainment. Their model calculates a weight for a number of influences, e.g. proximity to teacher, peers' state and student's own inclination to be either *productive* or *disruptive*. Their model takes into consideration the effect of

teacher proximity to a student, as well as the student's friends' state. Specific types of disruptive behaviour was not addressed in this work, but, importantly, simulation of *attainment* was.

In this paper we aim to move further and understand the effect of having disruptive students in a classroom through simulating *Inattentiveness* and *Hyperactivity* behaviours. According to the World Health Organization [37], Inattentiveness indicates moving between tasks, leaving one unfinished before losing interest, while Hyperactivity implies excessive movements, particularly in a situation where calmness is expected, such as remaining in one's seat. The two types are symptoms of the Attention-deficit hyperactivity disorder (ADHD) that has a prevalence in 5.9% to 7.1% of the children and adolescents [24]. Our work considers a student's achievement and the influence of teachers' as well as peers' characteristics. We use a fixed positioning of students, as in a regular classroom setting [16], therefore a friend's state (assuming they are not proximal to the student in question) cannot be considered an influence, as in the case depicted by Ingram and Brooks [15]. However, due to our agent-based approach, this could be generalised to classrooms with more movement. Importantly, we take into consideration the level of *teacher quality* and *control* as an added influence on student state transitions. Specifically, we aim to answer the following research questions:

R1. *To what extent does the existence of (different types of) disruptive students affect other students? (specifically, inattentive or hyperactive students)*

R2. *How does teaching quality and teacher control along with peer characteristics contribute to the achievement of young students in a disruptive classroom?*

2 Related Work

2.1 Disruptive Behaviour in Classrooms

The issue of disruptive behaviour of students from different age groups has been addressed in several studies [13, 14, 30]. In classrooms, we usually find a number of students, up to a quarter of a class, who display some form of disruptive behaviour [10]. Such students regularly show lower academic performance than their peers in the same class [7]. Additionally, the presence of disruptive behaviour in a classroom can increase the general disruptive level in that class. Shin and Ryan [28] explored whether the provision of emotional support by teachers could ameliorate high levels of disruptiveness in classrooms. They found that classes low in teacher emotional support had higher level of disruptiveness by the end of the year compared to classes high in teacher emotional support. It was found that students in classes with low teacher emotional support were more likely to have similar disruptive behaviour as their friends, which shows the effect of a teacher against peers' influence. Therefore, emotional support by teachers showed to be effective in reducing disruptive behaviour. Taking measures to ensure stability in classrooms and reduce disruptive behaviour is vital, as such behaviour is linked to low achievement of the whole classroom [25]. Bourne [3] used 'economy tokens' as a measure of reducing unwanted behaviour in the classroom, which decreased some disruptive behaviours to over 50% by the 7th week of the experiment.

2.2 Agent Based Modelling in Education

Agent based modelling (ABM) is a tool for modelling systems through software agents and their interactions in an environment. Agents interact with other agents and with the environment based on a set of behaviours driven from their defined characteristics. An agent can represent an individual or a group and their relationships in a simulation are represent social relations [19].

Agent based modelling has been adopted in the field of education, to serve different purposes. Some utilised it as a support of the learning activity, by modelling games for younger students, such as the case with Ponticorvo et al. [27], where they introduced a general ABM framework for designing digital games for young students by capturing the common features of educational materials and describing them in terms of interacting agents. A model of student behaviour [26] focused on cheating in assignments. Their model showed a strong connection between cheating and participating in extracurricular activities, as students who participated more in extracurricular activities had less time to finish their homework. Mauricio et al. [22] used a multi-level model, as well as ABM, to explain the differences in effectiveness between schools using social ties. The model presents peers' effect in the form of friendships that affects a student's learning attitude and teacher's effect through feedback and attention given to each group based on their academic performance. It assumed that more attention is given by teachers to higher performance groups than lower performance ones. Not enough attention has been given to the simulation of factors of a learning environment, thus, we simulate the effect of disruptive behaviour of young students and peers in the classroom. We use a disruptive score range defined by scales with items that is almost identical to the diagnostic criteria for ADHD in the American Psychiatric Association's Diagnostic and Statistical Manual of Mental Disorders (American Psychiatric Association, 1994) [23]. The model also takes into consideration different technical backgrounds of education practitioners, by providing a user-friendly front-end that allows them to easily use the model and observe its output during the simulation run. Validation process is complicated and requires sufficient real data to compare with [17]. We use the correlation coefficients comparison between input variables and output variables from real data and the model's simulated data.

3 Data

The main source of data was obtained from the Performance Indicators in Primary Schools (PIPS) monitoring system [33, 34]¹, in which young students were assessed at the start of their first year in elementary school and again at the end of that year. Specifically, assessments were carried out at the start and end of the academic year 2007/8. PIPS was run by the Centre for Evaluation and Monitoring (CEM) (www.cem.org) at Durham University, UK [12, 36]. The assessment process also provided a score, given by the teacher, for symptoms of disruptive behaviour (i.e. Inattentiveness in a range from 0 to 9, Hyperactivity with a range of 0 to 6) for each student at the end of the school year. The data contains 3,315 classes from 2,040 schools with an average of 26 students per class.

¹ RR344_-_Performance_Indicators_in_Primary_Schools.pdf (publishing.service.gov.uk).

The dataset has 65,385 records of students that include the mentioned Inattentiveness and Hyperactivity scores, as well as gender, and scores with a mean of 19.7 and 39.3 for the initial and end of year assessments of Math, respectively.

4 Methodology

As noted, we used Agent Based Modelling (ABM) to create a simulation of the learning process interactions. This is because the target stakeholders for our research question are human stakeholders in education, such as educational researchers, teaching administrators, teachers and, ultimately, students. We need to not only predict a fixed-point outcome (e.g. end of year results), but also be able to simulate how changing variables (e.g. the way of teaching a class) influence the outcome at different points in time (e.g. during a class, at the end of a class, at the end of a given number of classes, etc.).

From a technical point of view, the model was built using Mesa, which is an ABM framework in Python licensed by Apache2 [21]. Mesa provides a browser-based interface to visualise the model, which allows the use of interactive tools while running the model. This is especially useful during this COVID-affected time, when most interaction has moved online. Moreover, as it is coded in Python, it also has access to Python's large analysis tool library, such as SciPy for scientific computing, Pandas for data analysis and Matplotlib for visualisation.

From a visualisation point of view, a classroom is presented in the simulation as a 5×6 grid to satisfy the limit of class size being 30 students per class in the UK [8]. Shown as coloured circles, students start the class session in a random state of either learning, passive, or disruptive. The state becomes a *learning state* (in green) when the student has a low disruptive behaviour score. It turns into a *disruptive state* (in red) if the student has a high disruptive behaviour score or the student's Disruptive Tendency score exceeds the threshold (Disruptive Tendency and Disruptive threshold are defined in Sect. 4.1), where 1 tick in the model represents 1 min. When a student is being disruptive, he or she may affect the state of their neighbours, depending on the neighbours' disruptive score and the level of *Teacher Control* and *Teaching Quality*. As previously stated, every student has two disruptive behaviour scores: *Inattentiveness* and *Hyperactivity*, ranging from 0 to 9 and 0 to 6, respectively (as per PIPS). These values could in the future be set at the start of a class; for now, our model initialises each randomly. Students also have other attributes that will be explained in Sect. 4.1.

A Math lesson lasts for 45 min (as recommended by the Department for Education and Skills, 2002), where a student will be moving between the three states: passive, learning and disruptive (as modelled using the PIPS data). Figure 1 shows a flow chart of the model we have created to illustrate the change of the student state.

4.1 Variable Definition

The model offers first *switch variables* that can be manually altered for each run, as described below. These are partially informed by variables recommended by PIPS researchers, and partially self-derived. We discuss implications of choices in Sect. 7.

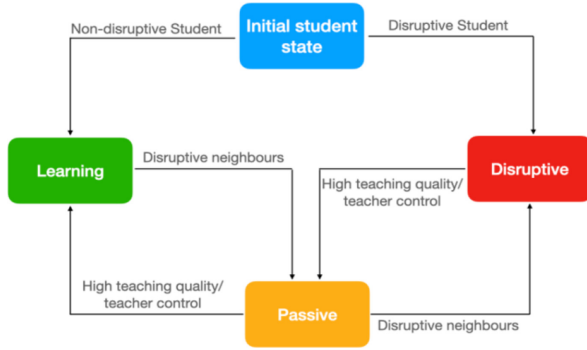


Fig. 1. SimClass model flow chart

Inattentiveness and Hyperactivity Switch: This variable switch can be tuned to indicate a high or low level of Inattentiveness/Hyperactivity behaviour in a class.

Teaching Quality/Teacher Control Switch: This switch varies the quality/control of teaching, ranging from 1 (weak) to 5 (excellent) this scale is defined for this model and was not taken from PIPS, as it is not available; its purpose is understanding the effect of this variable as a part of the learning environment factors.

Attention Span Switch: This variable represents the length of simulation time (ticks) the student maintains their learning state.

The model also computes a number of *derived variables* during the simulation runs, defined as follows below.

Initial Disruptive Tendency: Students will be allocated this value based on their Inattentiveness. We propose to compute it using the following formula:

$$DT_{initial}(s, c) = \frac{I(s) - \mu(s, c)}{\sigma(s, c)} \tag{1}$$

Where $I(s)$ is the Inattentiveness score of student s ; μ and σ are the mean and standard deviation values of Inattentiveness’ scores for class c of student s that is taken from PIPS data for a realistic setting.

Disruptive Tendency: This variable will change over time - students who are disrupted frequently will be affected and their disruptive tendency will increase. The length of time a student will be in a disruptive or a learning state will be affected by a student’s own characteristics, as well as that of the teacher’s and peers’:

$$DT(s, c, T_{current}) = \left(\frac{D(s, c, (T_{current} - 1)) - L(s, c, (T_{current} - 1))}{T_{current} - 1} \right) + DT_{initial}(s, c) \tag{2}$$

Where $D(s, c, T_{current})$ represents the number of ticks (minutes) when the student s was in a disruptive state till $T_{current}$, while $L(s, c, T_{current})$ represents their learning state’s

ticks until $T_{current}$. The higher the disruptive tendency becomes, the higher the chance that the student will change to a disruptive state; $T_{current}$ represents the number of ticks that passed since the beginning of the school year.

Math Attainment Level: This variable accounts for individual differences between students; it is derived from their initial score in Math as follows [31]:

$$A(s, c) = \frac{Smath(s, c) - \mu_{smath(c)}}{\sigma_{smath(c)}} \quad (3)$$

Similar to disruptive tendency, we use the z-score of student s 's initial assessment in the Math subject, Start Math, $Smath(s)$, defined below, because we wish to obtain information on varying from an average value, as opposed to absolute values. μ and σ are the mean and standard deviation values of Start Math scores for class c of student s that are computed before the simulation is initialised, either from PIPS data or model generated random data for the Start Math variable.

Start Math: This variable can be taken from PIPS or produced randomly by the model for each student. Its range (0–69) corresponds to the PIPS data range. Here, we took the values from PIPS, to simulate a realistic environment.

Start Math Scaled: As number of ticks the students learn indicate here their final score in Math, we have rescaled the Start Math score to represent *minutes of learning*:

$$Smath_{scaled}(s, c) = \left(e^{Smath(s,c)} \right)^{\frac{1}{n}} \quad (4)$$

We use n in the exponent to fit the logarithmic function to map the ‘learning Minutes’ into ‘Score’ in a similar manner as the work of [22], who used the logarithmic function to map ‘Teacher feedback’ into ‘Score’. To fit the logarithmic function, we use the total number of minutes the students would possibly have in a school year, which equals to $end-time = 8550$. Since $\log 8550^n = 69$, we calculate n to be ≈ 7.621204857 .

End Math: The simulated End Math score is shown in Eq. 5, where $L(s, c, T_{end-time})$ represents the total learning time student s had throughout the simulated year:

$$Emath(s, c) = \log(L(s, c, T_{end-time}) + Smath_{scaled}(s, c))^n + A(s, c) \quad (5)$$

Disruptive Threshold: Represents one standard deviation above the mean disruptive tendency of the class [2, 11].

4.2 Functionality

As per Fig. 1, students would be in a **learning state** if one of the following occurs:

- Disruptive Tendency is lower than the Disruptive Threshold of class.
- Disruptive Behaviour is low, and Teaching Quality or Teacher Control is high [35].

- Current state is passive, and more than half of the neighbours are in a learning state.

Students will be in a **passive state** if one of the following situations occurs:

- Disruptive Tendency is higher than the Disruptive Threshold, but Teacher Control or Teaching Quality is high.
- Current state is disruptive, but Teacher Control is high.
- Disruptive Behaviour is low, and Teaching Quality is low.
- Two neighbours are disruptive.
- Ticks of learning state exceed the attention span value.

Students will be in a **disruptive state** if one of the following situations occurs:

- Disruptive Tendency is higher than the Disruptive Threshold, and Teacher Control or Teaching Quality is low.
- Disruptive Behaviour is high, and previous state is passive.
- Previous state is disruptive, and Teacher Control is low (regardless of disruptive score) [35].
- Four or more neighbours are disruptive. The threshold of four is arbitrary defined for our current model, but can be further set by simulation requirements.

An ABM agent is a self-directed independent entity with attributes and protocols of interaction with other agents and their environment [20]. Our agent representing a student will remember its previous state and choose the next state based on earlier states. For example, if a student is *disruptive* for long, they can change to either *passive* or *learning*, based on characteristics or statuses of the teacher and neighbours. The model's

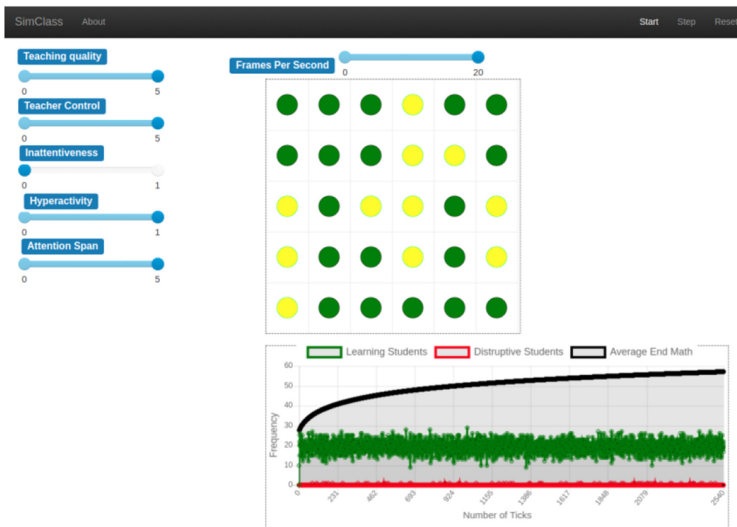


Fig. 2. Running the SimClass model with inattentiveness = 0

simulation visualisation (Fig. 2) will display the changes in student states during a minute (tick) in a lesson, with a line graph (below) that updates as the model runs. The graph follows the total number of disruptive students and learning students in every tick of the model. The black line represents the average End Math score of the class computed on every tick, while the red and green line represents the total number of disruptive and learning students.

5 Data Analysis

To answer the first research question, R1, and understand the effect of *disruptive* students on other students (here, the whole class), we explore the relationship between **disruptive behaviour** and **End Math scores** (here, representing general attainment – see Sect. 1). Specifically, we compute this End Math average score in classes with high number of disruptive students and then compare it with classes with lower number of *disruptive* students. We define the (set of) *disruptive* students as $DS \subseteq S$:

$$DS = \{s \in S, \text{ where } ds(s) \geq M\} \tag{6}$$

$$M = \{\text{median}(ds(s)) | s \in S\} \tag{7}$$

Where S is the set of all students, s is an individual student, $ds(x)$ is the disruptive score function, and M is the median. The median, rather than mean, was chosen to define the threshold, because the data, according to Shapiro’s test, is not normally distributed [1]. According to the data from PIPS, Inattentiveness has a median of 5, while Hyperactivity has a median of 3.

Out of 3,315 classes in the data set there were 2,337 classes with students categorised as *disruptive*. To have a deeper look into the data, we calculated the percentage of disruptive students per class and the average of the End Math score for that class and compared the two. Table 1 shows the correlation test results, where we can see that the percentage of disruptive students has a higher negative correlation (of -0.16) with the average of End Math. This suggests an effect of the number of disruptive students in a class over the general attainment - represented by End Math scores - in that class.

Table 1. Correlation test between disruptive behaviour and math scores

	Start Math	End Math	Average End Math
Inattentiveness	-0.27	-0.33	-0.07
Hyperactivity	-0.14	-0.18	-0.06
Percentage of disruptive students	-0.04	-0.06	-0.16

6 Results

Running the simulation model for 8,553 *ticks* represents a 45 min Math lesson a day for 190 days in a year [18]. We here present 3 *runs* with different parameter inputs, to observe their effect on student End Math scores. Results are shown in Table 2.

Run 1: In the first simulation run, we set all parameters with the maximum value for each (Teaching Quality and Teacher Control = 5, Inattentiveness/Hyperactivity = 1 and Attention span = 5). We chose this setting to be the baseline, to allow us to explore the different impact of each parameter in other runs.

Run 2: In this run of the model, we switched off Inattentiveness and kept the rest of the parameters at maximum value, in order to understand the effect of Inattentiveness variable over the results when compared with the baseline.

Run 3: Here, we aimed to observe the impact of Teaching Quality; therefore, all parameters had the maximum possible values of their ranges, except Teaching Quality, which was given the lowest possible value from its range, i.e., 1 out of 5.

Table 2. Results of End Math and disruptive tendency variables of three runs

	Math		Disruptive Tendency	
	First tick (Start Math)	Last tick (End Math)	First tick	Last tick
Run 1	27.43	43.08	1.16	0.12
Run 2	27.43	66.16	0.73	-0.53
Run 3	27.43	36.45	1.05	-0.07

7 Discussion

Three different parameter inputs into the simulation model provided different results. Therefore, we computed Cohen's *d* to present the effect size between the three runs (see Table 3). An effect size of .2 is considered small, .5 medium and .8 large [6]. We can see that the effect size is large between the runs. We used t-test and found the differences between End Math scores of the three runs to be statistically significant.

Table 3. Cohen's *d* and t test between End Math scores of all runs

	End math (Run 1)	End math (Run 2)	End math (Run 3)
End math (Run 1)	-	1.43 (p = 4.13e-42)	7.81 (p= 6.41e-07)
End math (Run 2)	-	-	9.12 (p = 3.09e-37)
End math (Run 3)	-	-	-

In the case of the third simulation, when Teaching Quality was reduced, the End Math results produced by the model were the lowest, with an average of 36.45, indicating that

students made the least progress in Maths of all runs. This means that Teaching Quality as a characteristic of the teacher influenced the attainment of the class by the end of the year. Additionally, we can see that students had also the highest disruptive tendency in this run. In contrast, the highest average of End Math scores was seen in the second run, when the Inattentiveness switch was off, resulting in 66.16 for the average End Math score, which presents an answer to Run 2 showing a negative effect of disruptive students in a class over their attainment. An average of 43.08 falls in between the previous two in the baseline run, when all variables used in the model had the maximum value allocated for each range. To compare with the real-world PIPS data², we ran a Pearson correlation test for the three different simulation runs.

Table 4. Correlation test between simulation runs results and model variables (8,553 ticks)

	End Math (Run 1)	End Math (Run 2)	End Math (Run 3)	End Math (PIPS)
Start Math	0.71	0.74	0.66	0.70
Inattentiveness	-0.31	-0.09	-0.38	-0.34
Hyperactivity	-0.13	-0.11	-0.12	-0.18

Table 4 shows that the correlation results of the three runs are close to End and Start Math of PIPS data, which was (computed separately to be) 0.70. The nearest correlation score to PIPS data can be seen in the first run, with 0.71, where all parameters had the maximum values possible. Therefore, we computed the correlation between this run’s simulated End Math and PIPS End Math and found the correlation to reach 0.68. These results can be used for finding adjustment of the model such as adding elements of learning, changing ticks representation and adjusting neighbours’ affect.

Next, we consider our various parameters in more details. We have used here inattentiveness as disruptive, but this may not be the case. It can be passive, such a daydreaming. But impulsivity can be disruptive. As we do not have a direct measure of disruption, anything in the model is a proxy. Follow-up work can look into the relation between Disruptive Tendency and its impact on personality. We have here simulated, analysed and compared results at classroom level, and compared averages. We showed the link between pupil disruption and Math attainment for pupils and for classes, i.e. at *two levels*. This naturally leads to *multi-level models* for future simulations. Beside the 3 runs presented here, we have run simulations with various parameters. More structured experiments are planned with models with slight variations, gradually moving toward each of the extremes represented here as Run 1, Run 2, Run 3, and graph the results. A related issue, to be addressed by multiple runs, is the stability of the models – how much variation there is when parameters hardly change. Start Math scaled, introduced here, is currently rather deterministic – if we know how much time has been devoted to Maths we will know the score in Maths. But children’s Maths scores rise and flatten and rise again

² Please note however that PIPS data is only available for Start Math and End Math, thus only the start and end of the simulation process.

and stagnate in unexpected ways. Future work could contain an element of randomness, to note if results change significantly. The model can then be applied by teachers to understand the effect of the disruptive students in each classroom depending on their numbers, positions and work towards minimising this effect through management styles or rearrangements. Teachers can use the model by uploading their own dataset for initial scores or have the model generate these scores randomly. They can then set the range of available parameters to the setting they would like to explore and run the model. The simulation will display in real-time output showing the changing variables over time.

Limitations include addressing only *Inattentiveness* and *Hyperactivity* as factors influencing disruptive behaviour in class, while other student characteristics, such as gender or social economic status, might impact on disruptive behaviour. Also, data are from only one country (UK), and are from 2007. Society's evolution means young students are more digital natives than ever, social interactions have evolved. Finally, more student characteristics could be modelled and simulated to further fine-tune the results.

8 Conclusion

This paper has presented an ABM model design to understand the *effect of disruptive young students in a classroom environment using the PIPS data*. The model simulates the interactions for one school year. The results show an increase in average End Math scores when the *Inattentiveness* variable is reduced, which confirms the effect of disruptiveness in a class over *attainment*, conforming to the PIPS data. In contrast, a decrease in the average End Math scores was seen when the *Teaching Quality* was reduced, showing the effect of teacher characteristics over students' attainment. The model was created using a user-friendly front, which allows users to make adjustments to the model easily to find how to apply pedagogical strategies. Future work includes exploring and validating further additions to this model, such as teacher intervention using rewards or adding a teacher assistant to observe the impact over attainment.

References

1. Agnihotri, L., et al.: Mining login data for actionable student insight. In: Proceedings 8th International Conference Educational Data Mining, pp. 472–475 (2015)
2. Blank, C., Shavit, Y.: The association between student reports of classmates' disruptive behavior and student achievement. *AERA Open* **2**(3),(2016). <https://doi.org/10.1177/2332858416653921>
3. Bourne, P.A.: A token economy: an approach used for behavior modifications among disruptive primary school children. *MOJ Public Heal.* **7**(3) (2018). <https://doi.org/10.15406/mojph.2018.07.00212>
4. Cardoso, A.P., et al.: Personal and pedagogical interaction factors as determinants of academic achievement. *Procedia Soc. Behav. Sci.* **29**, 1596–1605 (2011). <https://doi.org/10.1016/j.sbspro.2011.11.402>
5. Cobb, J.A.: Relationship of discrete classroom behaviors to fourth-grade academic achievement. *J. Educ. Psychol.* **63**(1), 74–80 (1972). <https://doi.org/10.1037/h0032247>
6. Cohen, J.: *Statistical Power Analysis for the Behavioral Sciences*. Academic press (2013)

7. Finn, J.D., et al.: Disruptive and inattentive-withdrawn behavior and achievement among fourth graders. *Acad. Manag. Rev.* **31**(2), 386–408 (2006). <https://doi.org/10.1097/EDE.0b013e3181>
8. Department for Education: Class Size and education in England evidence report. Research Report DFE-RR169, pp. 34–35 (2011)
9. Department for Education and Skills: Mathematical activities for the Foundation Stage Early years (2002)
10. Esturgó-Deu, M.E., Sala-Roca, J.: Disruptive behaviour of students in primary education and emotional intelligence. *Teach. Teach. Educ.* **26**(4), 830–837 (2010). <https://doi.org/10.1016/j.tate.2009.10.020>
11. Finn, J.D., Pannozzo, G.M., Voelkl, K.E.: Disruptive and inattentive-withdrawn behavior and achievement among fourth graders. *Elem. Sch. J.* **95**(5), 421–434 (1995). <https://doi.org/10.1086/461853>
12. Fitz-Gibbon, C.T.: Official indicator systems in the UK: examinations and inspections. *Int. J. Educ. Res.* **25**(3), 239–247 (1996)
13. Haroun, R., O’Hanlon, C.: Teachers’ perceptions of discipline problems in a jordanian secondary school. *Pastor. Care Educ.* **15**(2), 29–36 (1997). <https://doi.org/10.1111/1468-0122.00053>
14. Houghton, S., et al.: Classroom behaviour problems which secondary school teachers say they find most troublesome. *Br. Educ. Res. J.* **14**(3), 297–312 (1988). <https://doi.org/10.1080/0141192880140306>
15. Ingram, F.J., Brooks, R.J.: Simulating classroom lessons: an agent-based attempt. In: Proceedings Operational Research Society Simulation Work SW 2018, pp. 230–240 (2017)
16. Koneya, M.: Location and interaction in row-and-column seating arrangements. *Environ. Behav.* **8**(2), 265–281 (1976)
17. Koster, A., Koch, F., Assumpção, N., Primo, T.: The role of agent-based simulation in education. In: Koch, F., Koster, A., Primo, T., Guttman, C. (eds.) CARE/SOCIALEDU -2016. CCIS, vol. 677, pp. 156–167. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-52039-1_10
18. Long, R.: The School Day and Year (England). House Commons Library, Vol. 07148 (2019)
19. Macal, C., North, M.: Introductory tutorial: agent-based modeling and simulation, pp. 6–20 (2014)
20. Macal, C.M., North, M.J.: Agent-based modeling and simulation. In: Proceedings 2009 Winter Simulation Conference, pp. 86–98 (2009). <https://doi.org/10.1109/WSC.2009.5429318>
21. Masad, D., Kazil, J.: Mesa: an agent-based modeling framework. In: Proceedings 14th Python Science Conference, pp. 51–58, April 2015. <https://doi.org/10.25080/majora-7b98e3ed-009>
22. Mauricio, S., et al.: Analysing differential school effectiveness through multilevel and agent-based modelling Multilevel Modelling and School Effectiveness Research. **17**, 1–13 (2014)
23. Merrell, C., et al.: A longitudinal study of the association between inattention, hyperactivity and impulsivity and children’s academic attainment at age 11. *Learn. Individ. Differ.* **53**, 156–161 (2017). <https://doi.org/10.1016/j.lindif.2016.04.003>
24. Merrell, C., Tymms, P.B.: Inattention, hyperactivity and impulsiveness: their impact on academic achievement and progress. *Br. J. Educ. Psychol.* **71**(1), 43–56 (2001)
25. Müller, C.M., et al.: Peer influence on disruptive classroom behavior depends on teachers’ instructional practice. *J. Appl. Dev. Psychol.* **56**, 99–108 (2018). <https://doi.org/10.1016/j.appdev.2018.04.001>
26. Paul, R., et al.: Using agent-based modelling for EER experimental design: preliminary validation based on student cheating behaviours. In: Canadian Engineering Education Association, pp. 1–8 (2020)

27. Ponticorvo, M., et al.: An agent-based modelling approach to build up educational digital games for kindergarten and primary schools. *Expert Syst.* **34**(4), 1–9 (2017). <https://doi.org/10.1111/exsy.12196>
28. Shin, H., Ryan, A.M.: Friend influence on early adolescent disruptive behavior in the classroom: teacher emotional support matters. *Dev. Psychol.* **53**(1), 114–125 (2017). <https://doi.org/10.1037/dev0000250>
29. Smith, D.P., et al.: Who goes where? The importance of peer groups on attainment and the student use of the lecture theatre teaching space. *FEBS Open Bio* **8**(9), 1368–1378 (2018). <https://doi.org/10.1002/2211-5463.12494>
30. Stephenson, J., et al.: Behaviours of concern to teachers in the early years of school. *Int. J. Disabil. Dev. Educ.* **47**(3), 225–235 (2000). <https://doi.org/10.1080/713671118>
31. Swing, S.R., Peterson, P.L.: The relationship of student ability and small-group interaction to student achievement. *Am. Educ. Res. J.* **19**(2), 259–274 (1982). <https://doi.org/10.3102/00028312019002259>
32. Tomasevic, N., et al.: An overview and comparison of supervised data mining techniques for student exam performance prediction. *Comput. Educ.* **143**, 103676 (2020). <https://doi.org/10.1016/j.compedu.2019.103676>
33. Tymms, P.: *Baseline Assessment and Monitoring in Primary Schools* (1999)
34. Tymms, P., Albone, S.: Performance indicators in primary schools. In: *School Improvement through Performance Feedback*, pp. 191–218 (2002)
35. Tymms, P., Brown, C.: *Modelling educational systems: an interactive approach to integrating theory and policy*
36. Tymms, P., Coe, R.: Celebration of the success of distributed research with schools: the CEM centre. *Durham. Br. Educ. Res. J.* **29**(5), 639–667 (2003)
37. World Health Organization: The ICD-10 classification of mental and behavioural disorders. *World Heal. Organ.* **55**(1993), 135–139 (1993). <https://doi.org/10.4103/0019>



Investigating Clues for Estimating ICAP States Based on Learners' Behavioural Data During Collaborative Learning

Yoshimasa Ohmoto¹(✉), Shigen Shimojo², Junya Morita¹, and Yugo Hayashi²

¹ Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu 432-8011, Japan
ohmoto-y@inf.shizuoka.ac.jp

² Ristumeikan University, 2-150 Iwakura-cho, Osaka, Ibaraki 567-8570, Japan

Abstract. Interactions based on the learners' state of understanding and their attitudes toward tasks are considered important for realising a support system for collaborative learning. In this study, as a first step, we tried to detect whether the learner's state is Passive in the ICAP theory from the data obtained during collaborative learning. We actually conducted an experiment of collaborative learning between participants and obtained data on facial features, gaze directions, and speech state during the experiment. Based on these data, we investigated clues to classify the status of ICAP as either Passive or not. As a result, we were able to find several candidates. On the other hand, in the state classification of participants' states using these independent variables, it was not possible to show high accuracy. In future experiments, we plan to simultaneously measure physiological indices as a clue to estimate participants' internal state.

Keywords: ICAP framework · Collaborative learning · Learning support

1 Introduction

Collaborative learning is effective at promoting one's own understanding by incorporating different perspectives [3, 9, 12]. Several important interactions are required for the success of collaborative learning. Interactions based on the learners' state of understanding and their attitudes toward tasks are considered important [7]. Computer Supported Cooperative Learning (CSCL) and Intelligent Tutoring Systems (ITS) have been proposed to provide such interactions [6, 11]. Realising a support system for collaborative learning requires estimating the learner's state in real-time during learning and return appropriate feedback based on it. The cognitive tutoring system (e.g., [1]) uses a pre-built model to detect the learner's state and provide relevant and appropriate feedback.

ICAP theory is a framework of human collaborative learning [4]. In ICAP theory, learners' activity states in collaborative learning are categorised as follows: Passive state, Active state, Constructive state, and Interactive state. When

collaborating with agents on tasks, the goal should be to attain the interactive state in ICAP theory. However, many cooperative agents have been unable to effectively change the human state from Passive to Active [8,10]. Some studies have classified learning states based on the ICAP framework and investigated the effects of the support. A previous study [18] have investigated whether the ICAP framework could predict learning performance in STEM classes.

Our final goal is to realise an ITS using cognitive tutor agent (s) who provide appropriate feedback according to the learner's state. As a first step in this study, we tried to detect whether a learner's state was Passive in the ICAP from data obtained during collaborative learning. Implementing a cognitive tutor agent requires both estimating the learner's status at a certain point in time and predicting the learner's state in the near future. We conducted an experiment in which participants collaboratively learned and obtained data on facial features, gaze directions, and utterance states during the experiment. Based on these data, we investigated clues to classify the ICAP states as Passive or other.

2 Method

2.1 Participants

The present study used 16 Japanese university students (eight pairs: four male and 12 female) from a previous study as a sample. Here, pairs of learners worked on a collaborative learning task that required explaining a specific psychological phenomenon while creating a concept map. One participant participated in the experiment only once. For details, refer to [13–15].

2.2 Materials and Systems

Two PCs and two monitors were used by participants, two video recorders filmed their conversations and facial expressions and two Tobii systems [16] recorded their eye gaze. Cmap Tools [5] was used in the experimental task. A monitor and video recorder were placed in front of each participant in pairs. They sat across from each other with a partition placed between them so they could not see each other. Each participant was free to talk.

2.3 Procedure

In the experiment, the task was to make inferences about a certain psychological phenomenon. Before the task, participants read a text passage about causal attribution. In the story, the characters participated in school counselling with Michael Peter, who was discussing that he is worried about the new semester. The participants need to explain why the student (Michael Peter) was worried about the new semester based on the story by causal attribution. At the end of this phase, a comprehension test was performed (pre-test). Next, in the individual phase (10 min), the participants conducted the task [17] of applying the

causal attribution of success and failure to the episode (10 min). In this study, dyads were asked to build concept maps about causal attribution and create them individually at that time. Finally, in the collaboration phase (15 min), the participants created a concept map collaboratively with reference to their individual maps. At the end of the experiment, another comprehension test was performed (post-test).

2.4 Independent Variables

Data Segmentation. During the experiment, facial features and gaze direction data were acquired at approximately 30 fps. The moving average was calculated by shifting the data by 5 s with 10-s window. We analyzed the data of the moving average.

Utterance. Participants' utterances were annotated with the start and end times from the video by the experimenter. Then, the number of seconds in utterance(s) for each window of the moving average was calculated. We used the number of seconds of utterance(s) by the own and other as independent variables (self_u, other_u).

Facial Features. The facial movements were analyzed by OpenFace. This automatically calculated whether an Action Unit (AU) appeared. The numerical value output by OpenFace indicated the strength of the AU and was obtained by a formula described in a "toolkit" for using the software [2]. There were 18 types of AUs observed among the participants as follows. AU01: Inner Brow Raiser, AU02: Outer Brow Raiser, AU04: Brow Lowerer, AU05: Upper Lid Raiser, AU06: Cheek Raiser, AU07: Lid Tightener, AU09: Nose Wrinkler, AU10: Upper Lip Raiser, AU12: Lip Corner Puller, AU14: Dimpler, AU15: Lip Corner Depressor, AU17: Chin Raiser, AU20: Lip stretcher, AU23: Lip Tightener, AU25: Lips part, AU26: Jaw Drop, AU28: Lip Suck, and AU45: Blink.

Gaze. Gaze was acquired by the Tobii system. A monitor was placed in front of each participant to construct concept maps during the experiment that showed three concept maps: their own concept map and the partner's concept map that they had created in the experiment's preliminary stage and the concept map on which they had collaborated during the experiment. In the analysis, we used the location (own concept map, partner's concept map, or the collaborative concept map) and duration of participants' attention on this screen as independent variables (gaze_self, gaze_other, gaze_c).

2.5 Analysis

As a first step toward implementing a cognitive tutoring agent that provides adaptive feedback based on the learner's state, two analyses were conducted in

this study. One is Sequential Pattern Mining (SPM), which was conducted to obtain clues for predicting the participant's state in the near future. The other was the Generalised Linear Mixed Model (GLMM), which was conducted to obtain clues with which to estimate the participant's state among many variables. The participants' ICAP states annotated by the experimenter were the dependent variable and the analyses were conducted using the abovementioned independent variables.

Sequential Pattern Mining. This article uses Sequential Pattern Mining to extract features from the data during the experiment. SPM is an analysis method that extends the basket analysis to sequence databases. SPM can extract frequently occurring patterns based on the order in which multiple elements emerge. Using this, we considered that we could find clues to estimate learners' state by extracting patterns immediately before the change in the learner's state that is non-Passive (i.e., Active, Constructive, or Interactive; ACI states).

After calculating the moving average of each independent variable, it was encoded 1 if it exceeded a certain threshold and 0 otherwise. The data sequence was converted into transaction data for SPM. From the transaction data, the data for the 20s immediately before the change to an ACI state in the ICAP was extracted and common patterns contained in the data were extracted. To perform SPM analysis, we used the "arulesSequences" package included in R v4.0.2. The extraction conditions in this analysis were: $\text{maxlen} = 4$, $\text{maxsize} = 4$, $\text{confidence} > 0.7$, $\text{lift} > 1.0$, and $\text{support} > 0.15$.

As a result, we were able to extract 822 patterns as sequence patterns before the change to ACI states, some of which were sub-patterns of other sequence patterns. Frequent independent variables appearing in these patterns are candidates for clues that estimate the learner's state in the near future.

The independent variables with the highest occurrence frequency were; *gaze_c* (549), *AU04* (179), *AU07* (597), *self_u* (433), and *other_u* (483). The numbers in parentheses are the number of occurrences. The only variable other than these that was included in the sequence patterns was *AU25* (2).

The results show that there were a limited number of independent variables that appear with some degree of commonality before the change to ACI states. In addition, there is no ever-present independent variable. On the other hand, by using these highly common variables as clues, we may be able to estimate the state of a learner in the near future. However, since these variables appear frequently even in the Passive state, it is difficult to estimate the learner's state using just these clues.

Generalised Linear Mixed Model. The degree of freedom of communication is high in real sequences of communication. The high level of freedom means that we can assume that there exist a number of hidden variables that are necessary to estimate the learner's state. This study uses a hierarchical Bayesian model to include such hidden variables in the analysis. The generalised linear mixed model (GLMM) is commonly used for this purpose. Aside from the explanatory

variables’ fixed effects coefficient, GLMM introduces another coefficient called the random effect; we can include hidden variables as a random effect by using GLMM; we used the “MCMCglmm” package included in R v4.0.2.

In this analysis, the results of the comprehension tests conducted before and after the experiment (pre-test and post-test) and the labels indicating individuals (userID) were included as random effects. Three variables were included as random effects; GLMM was applied to all combinations of random effects (eight patterns), including the case where no random effect was included whatsoever, and the variables of fixed effects used for discrimination were reduced by the variable reduction method.

Table 1. Results of GLMM analysis.

	–	UI	Pre	UI+Pre	Post	UI+Post	Pre+Post	UI+Pre+Post
self_u	0.00015	0.00015	0.00015	0.00015	0.00015	0.00010	0.00015	0.00015
other_u	0.0067	0.0024	0.0015	0.0027	0.0054	0.0068	0.0042	0.020
AU05	0.0042	0.033	0.024	0.0018	0.015	0.0012	0.0015	0.00091
AU06	–	0.010	–	–	–	–	–	0.0058
AU07	0.013	–	0.00061	–	–	–	0.00030	–
AU10	0.0018	0.0012	0.029	–	–	0.13	0.10	0.00015
AU12	0.00015	–	0.0012	–	0.00015	–	0.16	–
AU14	0.0048	–	0.00015	–	0.00015	–	0.013	–
AU15	0.0045	–	0.10	–	–	–	–	–
AU17	–	–	–	–	–	–	0.031	–
AU26	0.0030	–	0.0054	–	–	–	–	–
gaze_c	0.081	0.0024	–	–	–	0.015	–	0.027
gaze_o	–	0.012	–	–	–	0.042	–	0.020
Accuracy	0.593	0.594	0.608	0.599	0.621	0.593	0.591	0.604

Table 1 shows the results of analysis; the topmost row shows the relevant random effect combinations. UI stands for userID, Pre for pre-test, and Post for post-test. For example, UI+Pre signifies that userID and pre-test were applied as random effects. We use “–” when a random effect was excluded. The leftmost column shows the names of the finally remained independent variables that were variable choice candidates. The numerical values in the table show the results of the variable reduction where the p-values represent selected independent variables and “–” represents removed variables. The accuracy of the discrimination is shown below the double line (Passive state or ACI states) and is calculated with the MCMC algorithm after selecting the variables.

The accuracy of the classification is at most 61%, which is above the chance level but not sufficiently high. In addition, even if various random effects were taken into account in the classification, the accuracy does not necessarily improve. In any case, the results suggest that the independent variables on which we are currently focusing are not sufficient to estimate the state of the learner.

However, the variables that remained in the analysis results can be considered candidates for clues to estimate the learner's state. The speech duration is a index that shows positive involvement because the task of this experiment required a relatively long utterance to improve the complex concept map, while a relatively short utterance was sufficient to approve another participant's opinion. AU05 is always remained so this is important facial feature for estimating the participant's state. AU10, AU12, and AU14 are changes around the mouth. It is thought that a learner's state appears in the interaction with speech. In the experimental data used in this analysis, the changes around the eyes may not have been used in communication because the learners could not see each other's faces. In addition, the results show that the variables that discriminate the learner's state do not necessarily include gaze_c and AU07, which appeared frequently in SPM. This suggests that the clues for estimating the learner's state in the near future may differ from clues for estimating the learner's state at a certain point in time.

3 Discussions and Conclusion

The main contribution of this study is to suggest that there may be differences between the clues for estimating the current state of a learner and for estimating the learner's future state. In addition, we attempted to discriminate between Passive and the rest in ICAP states based on data acquired in the experiment; although the accuracy was not sufficiently high, the probability of discrimination was higher than random chance. As an additional analysis, we tried to discriminate each participant's data in the experiment using the discriminant model with the highest accuracy constructed from the overall data and found that some participants were able to discriminate >80% while some of the others were below the chance level. To take these individual differences into consideration, we analysed the GLMM Random effects by including individual effects, but the accuracy improvements were small. Since the number of participants analyzed in this study was small, there may be variables that are not covered by our findings.

In the analysis of this study, we found that there were only a few clues to distinguish Passive from others in the available information. One reason for this is that, while various human behaviours observed in the state are judged as Passive, there are situations in which a person's internal state may be Active, Constructive, or Interactive but they show no explicit attitude (mainly speech). Previous studies analysing the same data have mainly analysed the differences between the Active, Constructive, and Interactive states out of the ICAP states. The analysis is also important so we will try to distinguish the three states (Active, Constructive, and Interactive) using much more data.

From the results of this study, it can be considered difficult to estimate the state of the ICAP with high accuracy from the information that can be observed in a simple manner. Therefore, in future experiments, we plan to simultaneously measure physiological indices as a clue to estimate participants' internal state. Physiological indices such as heart rate variability and skin conductance response are said to reflect the activities of the sympathetic and parasympathetic nervous systems in humans and can provide clues to estimate human conditions.




References

1. Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive tutors: lessons learned. *J. Learn. Sci.* **4**(2), 167–207 (1995)
2. Baltrušaitis, T., Robinson, P., Morency, L.: OpenFace: an open source facial behavior analysis toolkit. In: 2016 IEEE Winter Conference on Applications of Computer Vision, pp. 1–10 (2016)
3. Chi, M.T.H., Leeuw, N., Chiu, M., Lavancher, C.: Eliciting self-explanations improves understanding. *Cogn. Sci.* **18**(3), 439–477 (1994)
4. Chi, M.T., Wylie, R.: The ICAP framework: linking cognitive engagement to active learning outcomes. *Educ. Psychol.* **49**(4), 219–243 (2014)
5. <https://cmap.ihmc.us/>
6. Dillenbourg, P., Fischer, F.: Computer-supported collaborative learning: the basics. *Zeitschrift für Berufs- und Wirtschaftspädagogik* **21**, 111–130 (2007)
7. Hayashi, Y.: Towards supporting collaborative learning with an intelligent tutoring system: predicting learning process by using gaze and verbal information. *Cogn. Stud.* **26**(3), 343–356 (2019)
8. Misu, T., et al.: Modeling spoken decision support dialogue and optimization of its dialogue strategy. *ACM Trans. Speech Lang. Process. (TSLP)* **7**(3), 10. 221–224 (2011)
9. Okada, T., Simon, H.A.: Collaborative discovery in a scientific domain. *Cogn. Sci.* **21**(2), 109–146 (1997)
10. Raux, A., Langner, B., Bohus, D., Black, A.W., Eskenazi, M.: Let's go public! taking a spoken dialog system to the real world. In: Ninth European Conference on Speech Communication and Technology (2005)
11. Rummel, N., Weinberger, A., Wecker, C., Fischer, F., Meier, A., Voyiatzaki, E., et al.: New challenges in CSCL: Towards adaptive script support. In: Kanselaar, G., Jonker, V., Kirschner, P. A., Prins, F.J. (eds.) *Proceedings of ICLS 2008*, pp. 338–345. International Society of the Learning Sciences, Utrecht (2008)
12. Shirouzu, H., Miyake, N., Masukawa, H.: Cognitively active externalization for situated reflection. *Cogn. Sci.* **26**(4), 469–501 (2002)
13. Shimojo, S., Hayashi, Y.: An experimental investigation on collaborative dyads' explanation activities using conceptual maps: analysis on learning performance based on understanding and the use of different perspectives. *IEICE Tech. Rep.* **119**(39), 87–91 (2019a)
14. Shimojo, S., Hayashi, Y.: Relation between dialog activity and learning performance on collaborative learning visualized other knowledge: An analysis of turn-taking and knowledge convergence. *Japanese Cogn. Sci. Soc.* **36**, 2–46 (2019b)
15. Shimojo, S., Hayashi, Y.: How shared concept mapping facilitates explanation activities in collaborative learning: an experimental investigation into learning performance in the context of different perspectives. In: *Proceedings of the 27th International Conference on Computers in Education(ICCE2019)*, pp. 172–177 (2019c)

16. <https://www.tobiipro.com/>
17. Weinberger, A., Fischer, F.: A framework to analyze argumentative knowledge construction in computer supported collaborative learning. *Comput. Educ.* **46**(1), 71–95 (2006)
18. Wiggins, B.L., Eddy, S.L., Grunspan, D.Z., Crowe, A.J.: The ICAP active learning framework predicts the learning gains observed in intensely active classroom experiences. *AERA Open* **3**(2), 1–14 (2017)



Behaviour Analytics - A Moodle Plug-in to Visualize Students' Learning Patterns

Rita Kuo¹ , Ted Krahn², and Maiga Chang²  

¹ New Mexico Institute of Mining and Technology, Socorro, NM, USA

² Athabasca University, Edmonton, AB, Canada

maigac@athabascau.ca

Abstract. Learning Management System (LMS) is widely used in higher education. How to track students' learning behaviours in a LMS like Moodle becomes an important issue. This research designs a Moodle plug-in that not only can visualize students' learning behaviour patterns from the log but also can cluster students into different groups based on their behaviour patterns. Teachers can easily see how students went through one learning object to another; review the common learning pattern that students in the same group have; and, annotate the learning pattern a group or an individual student has based on his or her observations on the pattern's details. The annotations made by the teacher can be a support for researchers to further analyse and design mechanism and algorithm to automatic recognize and identify a student's characteristics and conditions like learning styles, preferences, at-risk, and potential required assistances via the features extracted from a learning pattern and notify the teacher or administrative staff automatically.

Keywords: Behaviour analysis · Pattern · Clustering · Learning Management System · Visualization · Learning path · Graph theory

1 Introduction

Based on Papamitsiou and Economides' study, most of the learning analytics are using classification or clustering methods [4]; only a few studies are working on visualizing students' learning behaviours. Graph structure is used for representing learning objects in the learning environments. For example, Cui and Yu use a knowledge graph tool in the Learning Cell System and ask students to create the knowledge graph based on the relations among the learning cells [1]. This research's goal is to design a Moodle plug-in that can visualize the relations among the learning objects and activities in a Moodle course as well as visualize the students' learning behaviours of accessing particular learning objects in the course. Moreover, the plug-in is capable of clustering students into different groups based on their learning behaviour patterns.

The rest of the paper is organized as following. The methods of generating knowledge structures and representing the learning behaviour in the LMS in this study are explained in Sect. 2. Section 3 demonstrates the Moodle plug-in the research team has designed based on the proposed methods. At the end, the summary of the works in this research and the proposed future works are described in Sect. 4.

2 Method of Representing and Clustering Learning Behaviours

Teachers usually organize the learning resources/activities in sections. The plug-in designed in this study retrieves the section information to group the resources/activities and create a graph structure. Figure 1 shows an example of how to organize the learning resources/activities in a graph structure. In the example a course has two sections: the section “Topic 1” has two webpages and one assignment, and the section “Topic 2” has two webpages. The plug-in constructs a graph with three nodes for sections, four nodes for webpages, and one node for the assignment. The learning resources/activities are directly linked to the section they belong to.

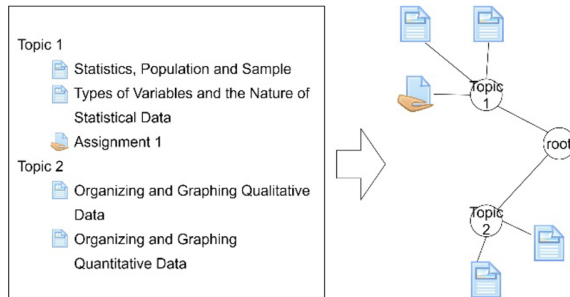


Fig. 1. A graph structure constructed from the learning resources/activities in a course.

At next stage, the plug-in retrieves students’ learning behaviours from Moodle’s log and generate the graph structure. For example, Fig. 2 shows that Amy prefers to work on the assignment first and then walks through the materials; on the other hand, Betty likes to read the materials in the beginning and at the end she works on the assignment. Their behaviours generate different learning paths.

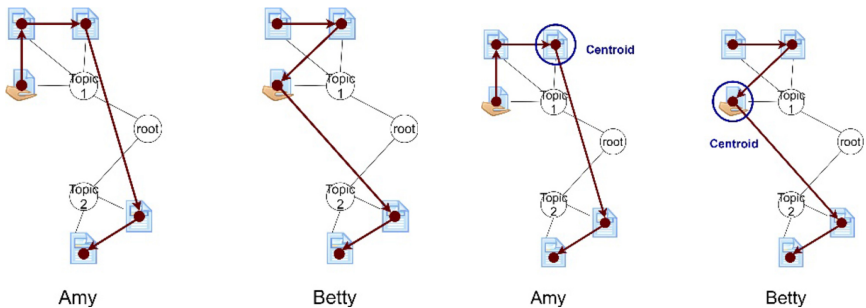


Fig. 2. Learning paths of two students.

Fig. 3. Centers of the two learning paths

After students’ learning paths are discovered, the teacher can use the plug-in to cluster students into different numbers of groups according to his or her needs. This research uses centroid subgraph [5] based on the centroid decomposition [2] to find the

center of a learning path. The centers of Amy and Betty’s learning paths in previous example then can be found in Fig. 3.

The learning path centroids’ coordinates are used to present students’ learning behaviours. The plug-in uses k-means algorithm [3] to cluster students. Based on the design, the research team developed the Moodle plug-in, Behaviour Analytics¹, for showing teachers their students’ learning behaviours in their Moodle courses as well as the common behaviour patterns that different student groups have.

3 Behaviour Analytics Moodle Plug-in

After Moodle system administrator installs the Behaviour Analytics plug-in, the course teacher could enable the plug-in through the course management interface. Figure 4 shows the page that displaying the relations among learning resources and activity. The node colors represent the different types of resources/activities in the course. When moving the cursor over a node in the graph, the plug-in shows which learning object the node represents for. It would be helpful when the teacher reviews his or her students’ learning behaviours. If the teacher believes a learning resource/activity in the course is not useful for clustering students’ behaviours, he or she can remove the node from the graph by unchecking the resource/activity at the right-hand side panel or simply clicking the node and choose to hide it.

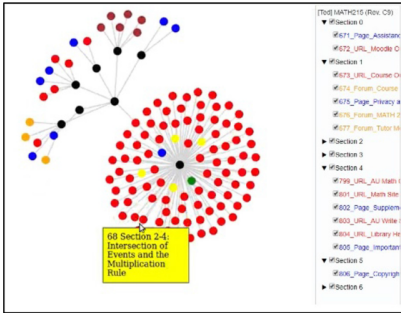


Fig. 4. Graph representation of the relations among learning objects in a course.

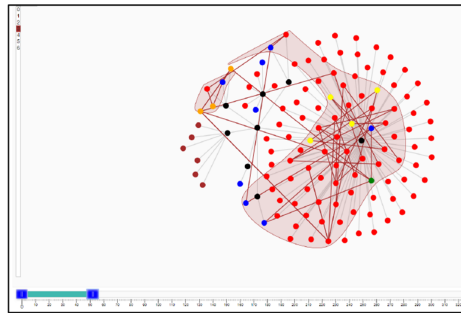


Fig. 5. Student’s learning behaviours and the correspondent learning-object coverage.

Teachers can also review students’ behavior through the plug-in as Fig. 5 shows. At left-hand side of the page, the plug-in lists all of the students (with anonymous number) and hides students’ information in order to protect student privacy. The teacher can select one (or more) student(s) to check their correspondent learning behaviours. The plug-in not only shows the learning behaviours of selected student(s) but also presents the learning resources/activities coverage for the selected student(s). The teacher can also review the learning behaviours in a specific time period by adjusting the timeline

¹ <https://studyguide.athabascau.ca/networkgraph/download.php>.

slider at the bottom. When the teacher selects more than one student as Fig. 6 shows, he or she can compare the learning resources/activities coverages among them easily in the plug-in.

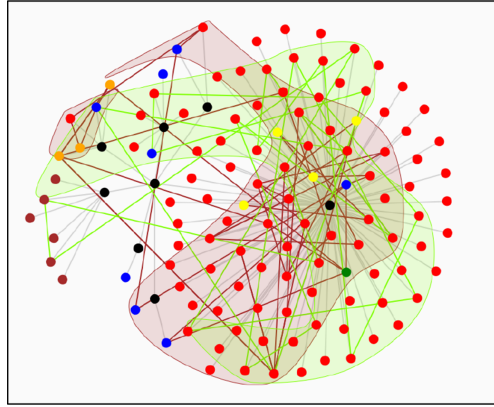


Fig. 6. Comparing two students' learning behaviours and their learning-object coverages.

At the next stage, the teacher can use the plug-in to cluster students into groups. As Fig. 7(a) shows, the plug-in first finds the centroids of students' learning behaviours with triangles symbols in different colors. The teacher can decide how many groups he or she would like the plug-in to group students. When the teacher decides to cluster the students into three groups – assuming each group will have significant different academic achievements, learning preferences, learning styles, or personality traits, the plug-in randomly puts three cluster centers (i.e., the cross symbols in different colors) on the plane as shown in Fig. 7(a). After a few runs, students #0, #1, #2, #3, and #5 are clustered into the same group which is presented by the blue cross; student #4 (i.e., the dark green triangle) is clustered into another group presented by the red cross; the last group presented by the orange cross includes student #6 (i.e., the orange triangle) as Fig. 7(b) shows.

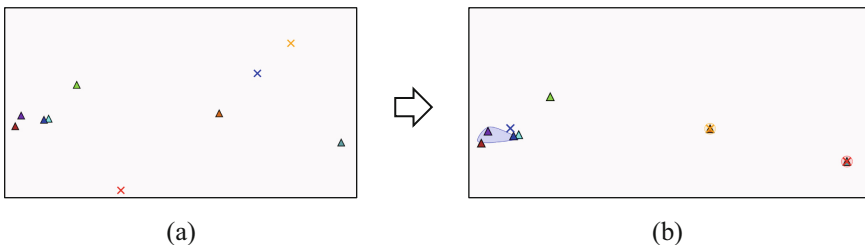


Fig. 7. Clustering students' learning behaviours into three groups. (Color figure online)

The teacher can still review a student's learning behaviour at this moment. Figure 8 shows student #1's learning behaviour in the center of the screen when the teacher moves

the mouse cursor over the dark blue triangle. The teacher can also move the mouse cursor over other triangles to see the differences.

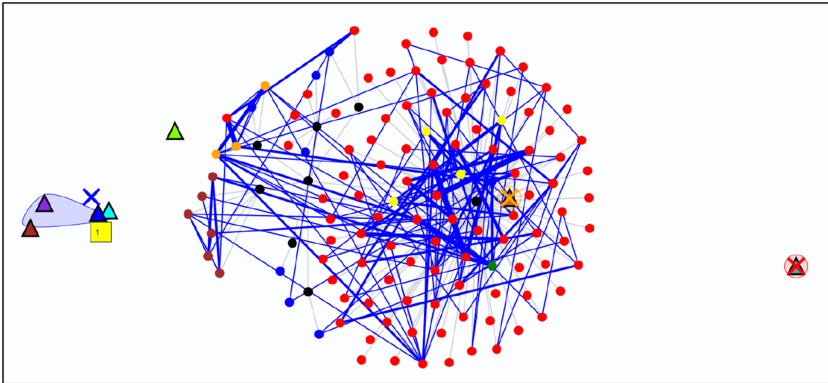


Fig. 8. Moving cursor over a student (i.e., a triangle) to see his/her learning behaviours. (Color figure online)

The teacher can also check the common learning behaviour pattern that students in the same group have. Figure 9 shows the common learning behaviour pattern that the group of students #0, #1, #2, #3, and #5 has, when the teacher moves the mouse cursor over the blue cross on the screen. If the teacher clicks the blue cross, he or she can add a note for the group based on his or her observation and thoughts on the group's learning behaviour pattern; for instance, he or she may identify that some learning objects or activities are always overlooked by this group of students. He or she can also do the same for individual student by clicking any triangle. After he or she reviews all students' learning behaviours in a group, the teacher can easily drag the triangle from one group and drop it to another one if he or she believes any of the students should be classified into another group.

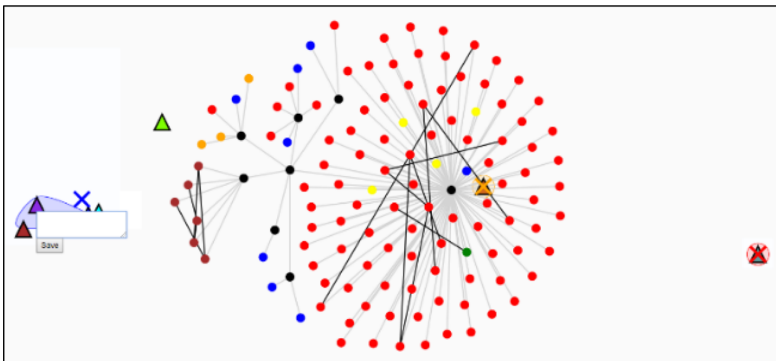


Fig. 9. Blue cross group's common learning behaviour pattern and the note taking textbox for the teacher. (Color figure online)

4 Conclusion

This research designs a graph-based student behaviour representation and analysis Moodle plug-in. The plug-in first analyzes the relations among learning resources/activities in a Moodle course and then constructs the graph structure for the resources/activities. The plug-in next retrieves the students' learning behaviours from Moodle's log and visualizes students' learning behaviours and the correspondent learning object coverage in graph. To cluster students into groups based on their learning behaviours, the plug-in uses centroid subgraph algorithm to find the centroids of the learning paths and applies k-means algorithm to cluster students after the teachers decide how many groups they would like to divide students into. The plug-in has been reviewed and approved by Moodle community on September 22, 2020 and can be found, downloaded, and installed from Moodle plug-in repository. As of April 1, 2021, it has been downloaded 180 times and installed in 94 Moodle sites around the world.

This research has some limitations. First, the learning resources/activities graph structure is based on the section organization in the Moodle course. If a course doesn't have its materials organized in sections, the learning resources/activities will be formed as a one-level tree structure. Another limitation is that the plug-in currently uses the centroids coordinates of learning behaviour patterns to cluster students into groups. When teachers change the positions of nodes in the learning resources/activities graph, it might also affect the clustering results. It is better to arrange the learning resources/activities nodes in a more meaningful and reasonable way automatically based on the relationship among the nodes.

To overcome both above-mentioned limitations, nature language processing has been considered in analyzing the content and/or meta-data of the resources/activities to construct a more meaningful graph. Currently the research team is developing the LORD² (Learning Object Relation Discovery) Moodle plug-in with the help of WordNet and Natural Language Processing. The LORD plug-in can measure the similarity between two learning objects according to their content in English and create a more reasonable and objective Learning Object Graph (LOG) that represents students' behaviours among learning objects.

Last but not the least, the research team plans to analyze the annotations that the professors made for the common learning behaviour patterns according to their observations and thoughts. Based on the analysis results, a mechanism could be designed to recognize and identify the features from students' learning behaviours. The plug-in then can provide teachers more information about their students and remind them when specific learning behaviour traces/features are detected.

References

1. Cui, J., Yu, S.: Fostering deeper learning in a flipped classroom: effects of knowledge graphs versus concept maps. *Br. J. Educ. Technol.* **50**(5), 2308–2328 (2019)
2. Jordan, C.: Sur les assemblages de lignes. *Journal für die reine und angewandte Mathematik* **70**, 185–190 (1869)

² <https://studyguide.athabascau.ca/networkgraph/access-lord.php>.

3. Lloyd, S.: Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982)
4. Papamitsiou, Z., Economides, A.: Learning analytics and educational data mining in practice: a systematic literature review of empirical evidence. *Educ. Technol. Soc.* **17**(4), 49–64 (2014)
5. Smart, C., Slater, P.J.: Center, median, and centroid subgraphs. *Netw. Int. J.* **34**(4), 303–311 (1999)



Toward a Webcam Based ITS to Enhance Novice Clinician Visual Situational Awareness

Komi Sodoké^{1(✉)}, Roger Nkambou^{1(✉)}, Issam Tanoubi², and Aude Dufresne²

¹ Université du Québec à Montréal, Montréal, Canada
sodoke.komi_sepeli@courrier.uqam.ca, nkambou.Roger@uqam.ca

² Université de Montréal, Montréal, Canada

Abstract. This research focus specifically on the eye-gaze movement of novice vs. expert clinicians to perform their clinical reasoning. The eye gaze data are spatiotemporal sequences representing the dynamic of the clinician's eye movements in the visual space to perform a clinical reasoning tasks. The objective is to do a comparative analyses of the eye movements fixations inside some areas of interest, the saccades trajectory and the scanpath. Taken together, the outcome of those analysis has provided us insights to build a webcam based ITS (Intelligent tutoring system). The aim of the ITS is to help reinforcing gradually the learning stages of novice clinicians with some cues from the behavioral implicit expert knowledge in terms of visual attention to perform a clinical reasoning in critical anesthesiology clinical case.

Keywords: Eye tracking · Scanpath comparison · ITS · Behavioral expertise transfer · Medical situational awareness tutoring

1 Introduction

Researches have produced interesting achievements on identifying expert/novice clinicians differences [3]. From those researches, the hypothesis generation is among the earliest phases to support the clinical reasoning and it involves a formulation of hypotheses derived from clinical observations such as vital signs, symptoms, physical examinations, etc. So, during that phase, the clinician's visual perception must be a proactive process of evidences collection. In some medical field such as anaesthesiology, visual perception is a key skills of the iceberg known as: visual situational awareness. In fact, the clinician must to develop the skills to see efficiently the patient vital signs such as heart rate, arterial saturation, blood pressure, respiratory rate, etc. In order to build his understanding and interpretation of the clinical situation. Therefore, the visual perception can modulate their expertise, and influences their practice and performance.

2 Experimental Design and Analysis

2.1 Methodology

After the institutional review board approval and a written informed consent, a sample of 12 experts and 12 novices from the department of Anaesthesiology at the Université de Montréal were recruited on a voluntary basis. A [Novice] is a resident clinician within the first or second year of the residency program. An [Expert] is a hospital staff member with more than 8 years experience. Eye tracking has been largely used in laboratory settings with different eye tracking devices such as the Tobii TX-300. However, to make eye tracking research available to a wider audience, webcams have started to be used to perform some eye tracking research [1]. The objective of this research is to build an ITS that will be used by novice clinicians not in a lab settings (with an eye tracker) but using their laptop. Therefore, the experiments to collect the eye gaze data and the ITS are performed using a webcam based online platform developed during the project and named EyeLab. For the experiment, participants were asked to visualize a video simulation inside EyeLab and verbalize toward its progression what they observe in the identified zones (areas of interest: AOI) (Fig. 1) to perform their clinical reasoning (think-aloud protocol). The simulation based on the CICO (Cannot Intubate/ Cannot Oxygenate) algorithm from the Difficult Airway Society to manage unanticipated difficult intubation on adults [4]. The seven AOI have been defined by the medical instructors based on their key importance for a situational awareness (ex. healthcare providers interactions, patient vital signs). For the data collection, each data point represents the AOIs seen by a participant at a given timestamp. A frame rate of 8 frames per second is set by default in EyeLab which means that it gets an eye fixation every 125 ms.

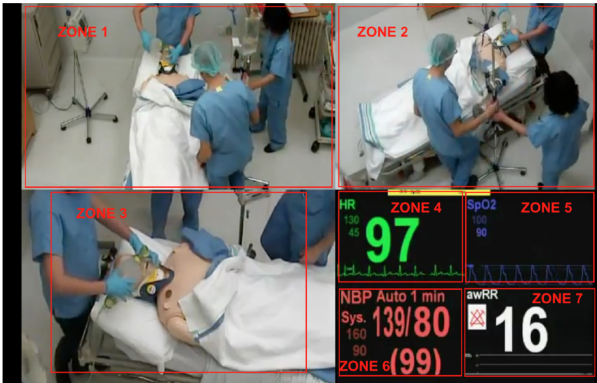


Fig. 1. Areas of interest in the simulation.

2.2 Analysis

Each data point represents the participant eye fixation inside an AOI at a given timestamp t_i . The consecutive eye movements data points of a given participant represent the dynamic of his eye movement trajectory within AOIs and it's called the *scanpath*. The scanpath is represented as a time series sequence like: AOI-1, AOI-3, AOI-5, AOI-5. We performed the following four analysis:

1. Preliminary analysis: to compare novices to experts using descriptive statistics on fixations (count means, duration, etc.)
2. Eye behavioural analysis: to make a finer comparative analysis of the novices and experts scanpaths around key events that are important for medical or situational awareness. From there, we generated incremental fixations heatmaps and observed more salient differences. For instance, at 01:37 (patient desaturation) novices focused on AOI 5, whereas experts focused on AOI 3. At 09:39 (Blue Code initiation) novices focused on AOIs 1 and 3, whereas experts focused in addition on AOIs 4 and 5.
3. Scanpaths comparison: to get objective metric of the similarity between the clinicians scanpath. One of the well-established sequences comparison algorithm is the Smith-Waterman algorithm [2]. It is used to compare the clinician scanpath to each other and the scores are normalized. The expert Vs expert scores have a mean of 0.8 and standard deviation of 0.1, while the novice Vs Novice have a mean of 0.4 and a standard deviation of 0.2. Those results indicate a more consistent and similar eye movement among the experts therefore could be used to built a typical expert scanpath.
4. Scanpath patterns classification: to create a deep convolutional LSTM for eye gaze spatiotemporal data sequence classification aiming to predict whether a given scanpath is a novice or an expert eye movements behavior [9].

3 EyeLab Pedagogical Services

EyeLab is built as a modern interactive and dynamic web application with modular architecture developed using open source technologies. On the client size, EyeLab captures the learner's eye movements via the webcam and sends them to processing modules on the server size where face detection and pupil's movements detection are implemented. At the end, the successive image frames with the eye position on screen are used to generate a video recording for each learner. A learner will replay his video recordings with a visual trace of his scanpath during the tutoring session.

Few studies have focused on improving learner eye movement towards high performance eye movement behavior. One of the interesting result was obtains by Litchfield and Al [8]. They conducted three experiments to observe how guiding attention via other people's eye movements would improve the radiographer performance in reading chest X-rays. They obtained the most significant improvement from novices when they were provided with the expert's eye movements. These results suggests that guiding novice attention toward other person's eye

movements can have a short-term effect on helping them to scaffold their decision using other expert's search behavior.

For the EyeLab ITS, we adopted an example-based learning approach. In fact, the example-based strategy is beneficial for novices who does not have the knowledge to support the resolution steps. In our case, there is no well-established domain knowledge about adequate eye movements that can be used by the novice in all clinical situations. So, it will be helpful to provide them customized insights into the way experts allocate their visual attention using their experience. The tutoring services is done by using only visual cues to guide novices' attention to relevant AOI since combining eye movement guidance with a verbal explanation could have negative cognitive load impact on learning [5]. The visual cues and guidance are provided interactively as metacognitive knowledge. By analyzing the learners' gaze during complex problem solving, [6] showed that the metacognitive support is efficient for learning and it helps novices to develop a deeper conceptual knowledge. The tutoring approach is done in two phases. In phase 1, the novices clinicians visualize the simulation video based on the experimental protocol. In phase 2, they visualize the recorded video with their scanpath and the customized tutoring hints and feedback.

The domain knowledge modelling is typically based on explicit formalized knowledge. Since there no formal domain knowledge about eye movement during clinical reasoning, the solution adopted is to use machine learning techniques to extract partial domain knowledge (scanpath) from the resolution traces. The experts scanpath sequence is used as the resolution traces since we observed constancy and high similarity. From there, the PhARules [7] patterns mining algorithm is used to extract the experts frequent eye movement patterns specifically when the key situational awareness events occurs. In fact, PhARules provided the advantages for the mining of sequential rules considering the resolution phases.

The learner model is represented as an overlay on the expert scanpath with some errors. Those errors are the learner scanpath misbehaviors using the expert as reference. The scanpath misbehaviors are obtained using sequence alignment to get differences derived from the number of mismatches and gaps. Those customized messages are triggered during the key-events when the learner scanpaths alignment with the experts scanpaths is bellow a predefined similarity threshold. The similarity is calculated using the Smith-Waterman algorithm. The tutor hints and feedback message includes spatiotemporal dimensions that indicate the AOIs involved as well as the chronological order to execute the expected scanpath behavior.

4 Discussion and Future Works

The research allowed us to collect eye gazed data from clinicians during their clinical reasoning. Several analyses have been performed to identify key differences between the clinicians' visual attention. Taken together, the outcome of those analysis have provided us insights to build a webcam based ITS. An initial

validation from the novice clinicians and the medical instructors showed a high level of satisfaction for the eye tracking functionalities and the tutoring services. An in-situ experiment is planned as a second phase of the project in a larger scale (more participants and clinical cases) to measure the behavioral learning gain and the learning persistence, to add an emotional state to the learner model using the facial expressions, etc.

References

1. Agustin, J.S., et al.: Evaluation of a low-cost open-source gaze tracker. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp. 77–80. ACM, Austin (2010)
2. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences (PDF). *J. Mol. Biol.* **147**(1), 195–197 (1981)
3. Cuthbert, L., duBoulay, B., Teather, D., Teather, B., Sharples, M.: Expert/novice differences in diagnostic medical cognition - a review of the literature. *Cognitive Sciences* (1999). Research paper 508
4. Frerk, C., et al.: Difficult airway society 2015 guidelines for management of unanticipated difficult intubation in adults. *Br. J. Anaesth.* **115**(6), 827–848 (2015)
5. Van Gog, T., Jarodzka, H., Scheiter, K., Gerjets, P., Paas, F.: Attention guidance during example study via the model's eye movements. *Comput. Hum. Behav.* **25**(3), 785–791 (2009)
6. Schwonke, R., Ertelt, A., Otieno, C., Renkl, A., Alevin, V., Salden, R.J.C.M.: Metacognitive support promotes an effective use of instructional resources in intelligent tutoring. *Learn. Instr.* **23**, 136–150 (2013)
7. Toussaint, B.-M., Luengo, V.: Mining surgery phase-related sequential rules from vertebroplasty simulations traces. In: Holmes, J.H., Bellazzi, R., Sacchi, L., Peek, N. (eds.) *AIME 2015. LNCS (LNAI)*, vol. 9105, pp. 35–46. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19551-3_5
8. Litchfield, D., Ball, L.J., Donovan, T., Manning, D.J., Crawford, T.: Viewing another person's eye movements improves identification of pulmonary nodules in chest x-ray inspection. *Exper. Psychol. Appl.* **16**(3), 251–262 (2010)
9. Sodoké, K., Nkambou, R., Dufresne, A., Tanoubi, I.: Toward a deep convolutional LSTM for eye gaze spatiotemporal data sequence classification. In: EDM (2020)

Feedback and Personalisation



Flexible Program Alignment to Deliver Data-Driven Feedback to Novice Programmers

Victor J. Marin, Maheen Riaz Contractor, and Carlos R. Rivero^(✉)

Rochester Institute of Technology, Rochester, NY, USA
vxm4964@rit.edu, mc1927@rit.edu, crr@cs.rit.edu

Abstract. Supporting novice programming learners at scale has become a necessity. Such a support generally consists of delivering automated feedback on what and why learners did incorrectly. Existing approaches cast the problem as automatically repairing learners' incorrect programs; specifically, data-driven approaches assume there exists a correct program provided by other learner that can be extrapolated to repair an incorrect program. Unfortunately, their repair potential, i.e., their capability of providing feedback, is hindered by how they compare programs. In this paper, we propose a flexible program alignment based on program dependence graphs, which we enrich with semantic information extracted from the programs, i.e., operations and calls. Having a correct and an incorrect graphs, we exploit approximate graph alignment to find correspondences at the statement level between them. Each correspondence has a similarity attached to it that reflects the matching affinity between two statements based on topology (control and data flow information) and semantics (operations and calls). Repair suggestions are discovered based on this similarity. We evaluate our flexible approach with respect to rigid schemes over correct and incorrect programs belonging to nine real-world introductory programming assignments. We show that our flexible program alignment is feasible in practice, achieves better performance than rigid program comparisons, and is more resilient when limiting the number of available correct programs.

Keywords: Automated program repair · Data-driven feedback

1 Introduction

The recent worldwide interest in computer science has originated an unprecedented growth in the number of novice programming learners in both traditional and online settings [2, 14, 18, 21]. A main challenge is supporting novice programming learners at scale [5, 10, 17], which typically consists of delivering feedback

This material is based upon work supported by the National Science Foundation under Grant No. 1915404.

© Springer Nature Switzerland AG 2021
A. I. Cristea and C. Troussas (Eds.): ITS 2021, LNCS 12677, pp. 247–258, 2021.
https://doi.org/10.1007/978-3-030-80421-3_27

explaining what and why they did incorrectly in their programs [7]. Different than traditional settings, online programming settings often have a large proportion of novice learners with a variety of backgrounds, who usually tend to need a more direct level of feedback and assistance [1]. A common practice is to rely on functional tests; however, feedback generated based solely on test cases does not sufficiently support novice learners [3, 9, 17].

Current approaches cast the problem of delivering feedback to novices at scale as automatically repairing their incorrect programs [3, 12, 13, 15, 17, 18]. Note that, similar to existing approaches, we consider a program to be correct if it passes a number of predefined test cases [3, 18]; otherwise, it is incorrect. Once a repair is found, it can be used to determine pieces of feedback to deliver to learners [17]. Non-data-driven approaches aim to find repairs by mutating incorrect programs until they are correct, i.e., they pass all test cases [11]. Data-driven approaches exploit the fact that repairs can be found in existing correct programs and extrapolated to a given incorrect program [18]. This paper focuses on the latter since, in a given programming assignment, there is usually a variety of correct programs provided by other learners that can be exploited to repair incorrect programs [3, 13, 15, 18].

The “search, align and repair” [18] framework consists of the following steps: 1) Given an incorrect program p_i , search for a correct program p_c that may be useful to repair p_i ; 2) Align p_i with respect to p_c to identify discrepancies and potential modifications in order to repair p_i ; and 3) Apply those modifications to p_i until the resulting program p'_i passes all test cases. Current approaches instantiating the “search, align and repair” framework use rigid comparisons to align incorrect and correct programs, i.e., they require the programs to have the same or very similar control flows (conditions and loops), and they are affected by the order of program statements [3, 13, 15, 18]. As a result, such approaches may miss a potentially valuable set of correct programs that can repair incorrect programs using flexible program comparisons.

In this paper, we explore a new alignment step that relies on a flexible program comparison based on approximate graph alignment of program dependence graphs. On one hand, a program dependence graph combines information about the control and the data (use of variables) flows of a program [4]. On the other hand, approximate graph alignment finds a correspondence between the nodes of two graphs [8]. Such a correspondence takes both topology and semantics of the graphs into account. When applied over program dependence graphs, topology implies programs that approximately match with respect to their control and data flow information, while semantics are modeled as operations and calls performed in the programs. Each pair that belongs to an alignment has a node similarity associated to it. Replacement suggestions are computed as those pairs whose similarities substantially deviate from the average similarity of a given alignment. Furthermore, addition and removal suggestions are those non-aligned nodes that are connected to replacement suggestions.

We evaluate our alignment step using nine real-world introductory programming assignments from a popular online programming judge (CodeChef). We collected publicly-available correct and incorrect programs in these assignments from real-world learners. We compare our flexible alignment with respect to existing rigid alignments: CLARA [3], Refazer [15], and Sarfgen [18]. For a fair comparison, we evaluate program comparison schemes using a common repair framework. We use different scenarios in which we vary the number of correct programs available. We show that our flexible program comparison achieves a better repair performance than other rigid program comparisons, and that it is more resilient to provide repairs when the number of correct programs available is reduced.

The paper is organized as follows: Sect. 2 summarizes previous approaches and ours; Sect. 3 presents background information; Sect. 4 describes our flexible comparison; Sect. 5 discusses our experimental results; Sect. 6 presents the related work; and Sect. 7 recaps our conclusions and future work.

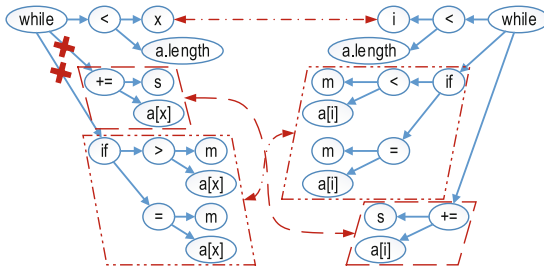
2 Overview

We consider CLARA [3], Refazer [15], Sarfgen [18], and `sk_p` [13] the state of the art in searching, aligning and repairing programs. CLARA and Sarfgen compare variable traces between an incorrect and a correct programs that share the same control statements like `if` or `while`. Refazer uses pairs of incorrect/correct program samples to learn transformation rules, which aid a program synthesizer to transform incorrect into correct programs. Finally, `sk_p` uses partial fragments of contiguous statements to train a neural network to predict possible repairs.

In the alignment step, these approaches compare an incorrect program with respect to a correct program based on rigid schemes, which limits their repair potential. To illustrate our claim, the Java programs presented in Fig. 1 aim to compute the minimum value in an array and the sum of all its elements, and print both minimum and sum values to console. Note that the values of the input array are assumed to be always less or equal than 100. In Sarfgen, an incorrect program will be only repaired if its control statements match with the control statements of an existing correct program. This is a hard constraint since: a) It requires a correct program with the same control statements to exist, and b) Such a correct program may not “naturally” exist. For instance, the control statements of the correct program in Fig. 1a do not match with the incorrect program in Fig. 1b; in order to match, the correct program should “artificially” contain an `if` statement before or after line 7, and such a statement should not modify the final output of the program. CLARA relaxes these constraints such that, outside loop statements (`while` or `for`), both programs can have different control statements, but they need to be the same inside loops. This relaxation still forces a correct program with the same loop signature to exist.

<pre> 1 void f(int[] a){ 2 int x = 0, m = 101, s = 0; 3 while (x < a.length) { 4 s += a[x]; 5 if (m > a[x]) 6 m = a[x]; 7 x++; 8 } 9 print(s + ", " + m); 10 } </pre>	<pre> 1 void g(int[] a){ 2 int i = 0, m = 101, s = 0; 3 while (i < a.length) { 4 if (m < a[i]) 5 m = a[i]; 6 s += a[i]; 7 i++; 8 if (m == 0) 9 i--; 10 } 11 print(s + ", " + m); 12 } </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(a) Correct program (b) Incorrect program



(c) Excerpt of (simplified) abstract syntax tree edits

```

1 while (x < a.length) {
2   s += a[x];
3   if (m > a[x])
                
```

(d) Lines 3–5 in Figure 1a

```

1 while (x < a.length) {
2   if (m > a[x])
3     m = a[x];
                
```

(e) Fragment needed to fix Figure 1b

Fig. 1. Comparison of different existing methods

Refazer uses the tree edit distance to find discrepancies between two programs. The tree edit distance between two equivalent abstract syntax trees with different order of statements implies multiple edits. For example, Fig. 1c shows an excerpt of the edits to transform the abstract syntax tree of the correct into the incorrect program in our example, which implies removing and adding full subtrees; however, only two edits would be necessary, i.e., changing “<” by “>” and removing the subtree formed by lines 8–9 in Fig. 1b. In sk_p, different order of statements result in different partial fragments, so additional correct programs will be required to train the program repairer. For instance, Fig. 1d shows a fragment extracted from the correct program; however, the incorrect program will only be fixed by a fragment like the one in Fig. 1e.

We propose an alignment step based on approximate alignment of program dependence graphs. Figure 2 shows an excerpt of the program dependence graphs derived from the programs in Fig. 1. Each node corresponds to a statement in such programs, e.g., u_3 corresponds to line 3 in Fig. 1a. The first step consists of transforming programs into program dependence graphs that are further annotated with semantic labels. For example, $Ctrl$ in u_3 summarizes that the corresponding statement is a control statement. In addition, u_3 is also annotated with Lt that represents the “less than” operation of the statement. We apply approximate graph alignment over two (correct and incorrect) program dependence graphs G_1 and G_2 . For each pair of nodes (u_i, v_j) such that u_i and v_j belong to G_1 and G_2 , respectively, we compute a node similarity based on topology and semantic labels. Having all pairwise node similarities, we compute an alignment from the nodes in G_1 to the nodes in G_2 . We cast this problem as finding a matching with maximum similarity in bipartite graphs [16]. Bold, double-headed edges in Fig. 2 represent a sample alignment with maximum similarity. Finally, we discover individual pairs in a given alignment that are useful for repairing an incorrect program. We rely on the node similarities in a given alignment to make this decision, i.e., each pair of nodes whose similarity deviates k standard deviations from the mean of the node similarities in the alignment are selected as repair suggestions. The intuition behind this is that the similarity of such pairs is smaller than the rest of the similarities in the alignment, i.e., they are less similar than others. In our example, we suggest (u_5, v_4) as a repair to fix the incorrect program in Fig. 1b. There may be nodes in the larger program dependence graph that are not present in the alignment. These nodes are suggested to be added or removed depending on whether they belong to a correct or an incorrect program, respectively. In our example, both v_8 and v_9 belong to an incorrect program, so they are suggested to be removed.

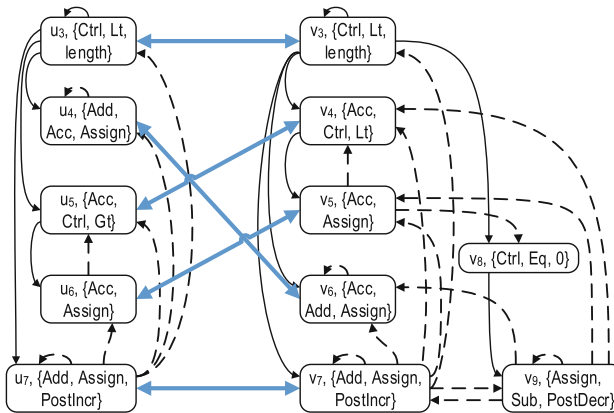


Fig. 2. Alignment of program dependence graphs derived from Figs. 1a and 1b

3 Background

A program dependence graph $G = (V, E, l_s, l_e)$ of a program p is a directed, labeled multigraph, where V is a set of nodes representing statements in p , $E : V \times V$ is a set of directed edges, $l_s : V \rightarrow (L_s, m)$ is a node labeling function, and $l_e : E \rightarrow \{Ctrl, Data\}$ is an edge labeling function. Let $(v_s, v_t) \in E$, $l_e((v_s, v_t)) = Ctrl$ indicates that the execution of node v_t depends on node v_s evaluating to true; furthermore, $l_e((v_s, v_t)) = Data$ represents that v_t uses a variable declared or re-assigned by v_s . L_s contains labels that summarize the semantics of a program statement, such as *Assign*, *Call*, and *Ctrl* to denote variable assignments, calls to other methods, and condition or loop statements, respectively. In addition, L_s includes labels to represent operation semantics and constants of a program statement, which include *Acc*, *Add*, and *Sub* to encode array access, addition, and subtraction, respectively. Since a program statement may contain multiple operations that are the same, e.g., multiple array accesses, $m : L_s \rightarrow \mathbb{N}$ is a function to support multisets. Note that, for the sake of simplicity, we omit the details of multi-method program dependence graphs, i.e., programs that contain multiple methods. In such cases, we extend L_s and $\{Ctrl, Data\}$ allowing nodes denoting method entry points, method calls, and parameters and result of a method call [4].

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two directed graphs such that $|V_1| \leq |V_2|$. Subgraph isomorphism consists of finding all non-induced solutions $\phi : V_1 \rightarrow V_2$ such that $\forall (u_i, u_j) \in E_1 \Rightarrow (\phi(u_i), \phi(u_j)) \in E_2$. The problem of approximate graph alignment consists of finding an injective function $\varphi : V_1 \rightarrow V_2$ such that $|V_1| \leq |V_2|$. This problem is a relaxation of the subgraph isomorphism problem in which we assume that $|V_1| \approx |V_2|$, and G_1 is approximately contained in G_2 .

4 Flexible Program Alignment

We wish to compute an alignment between the statements in a correct program with respect to the statements in an incorrect program. The computation of such an alignment takes topological and semantic information into account. On one hand, topological information encodes the context of each statement regarding its control and data dependencies, i.e., what are the statements that must be fulfilled in order for a given statement to be executed, and what are the variable uses of such a statement. On the other hand, semantic information allows us to distinguish statements that are performing different operations, such as addition or an API call. Let $G_1 = (V_1, E_1, l_s^1, l_e^1)$ and $G_2 = (V_2, E_2, l_s^2, l_e^2)$ be two program dependence graphs. At this stage, we are agnostic to correctness and incorrectness of the programs evaluated, so G_1 can be either correct (and G_2 is then incorrect), or incorrect (and G_2 is then correct). Our first goal consists of computing all the pairwise similarities between nodes in V_1 and V_2 . The similarity of nodes $u_i \in V_1$ and $v_j \in V_2$ is measured as follows [6, 8, 20]: $Sim(u_i, v_j) = \alpha Top(u_i, v_j) + (1 - \alpha) Sem(u_i, v_j)$, where *Top* and *Sem* are topological and semantic similarities, respectively, and $\alpha \in [0, 1]$ is a parameter to

balance the contribution of each type of similarity. $Sim(u_i, v_j) = 1$ entails that both nodes are identical.

We compute similarities $Sim(u_i, v_j)$ for every pair of nodes $u_i \in V_1$ and $v_j \in V_2$. The next step consists of computing an alignment between both graphs, i.e., $\varphi : V_1 \rightarrow V_2$ ($|V_1| < |V_2|$). We cast the problem of finding an alignment as finding a maximum weight matching in bipartite graphs. Let $B = (V_1, V_2, E, \omega)$ be a bipartite graph where V_1 and V_2 are the sets of nodes of G_1 and G_2 ($V_1 \cap V_2 = \emptyset$), respectively, $E : V_1 \times V_2$ is a set of undirected edges, and $\omega : E \rightarrow \mathbb{R}$ is a function that assigns weights to the edges as follows: $\omega(u_i, v_j) = Sim(u_i, v_j)$. There are several algorithms in the literature to compute maximum weighted matchings that find augmenting paths that alternatively connect edges in V_1 and V_2 , ensuring that the final similarity weight is maximized and, thus, producing the alignment φ with maximum similarity [16].

Once we have computed an alignment $\varphi : V_1 \rightarrow V_2$ between two program dependence graphs G_1 and G_2 , our goal is to discover repair suggestions, i.e., statements in the correct program that can be used to fix the incorrect program. To discover these statements, we rely on the node similarities of the pairs available in the approximate graph alignment φ . Recall that approximate graph alignment assumes that $|V_1| \leq |V_2|$, which leads to two different situations: 1) If G_1 is correct and G_2 is incorrect, non-aligned nodes belong to the incorrect program and we aim to remove them. This is the case of superfluous/inadequate statements. 2) Otherwise, non-aligned nodes belong to the correct program and we aim to add them to the incorrect program. This is the case of missing statements. Programs in Figs. 1a and 1b are an example of the former situation since the first one is correct and smaller than the second one, which is incorrect. In such a case, we aim to remove lines 8 and 9 from the incorrect program.

First, we address the problem of finding replacement suggestions, i.e., which pairs of nodes in a given alignment φ are appealing to repair incorrect statements replacing them by correct statements. Intuitively, we analyze which node similarities in φ significantly deviate from the rest of the node similarities, for which we rely on mean and standard deviation. Let μ_φ and σ_φ be the mean and standard deviation of the node similarities in φ , respectively. We establish a similarity threshold that, for all pairs whose node similarities are below such threshold, we will consider them as replacement suggestions, i.e., they significantly deviate from the rest. Therefore, we consider a pair $(u_i, \varphi(u_i))$ to be a replacement suggestion if $Sim(u_i, \varphi(u_i)) < \mu_\varphi - k \sigma_\varphi$, where $k \in \mathbb{R} \geq 0$.

Second, we address the problem of finding suggestions of statements to be added or removed. Recall that we suggest statements to be added when the size of the incorrect program is less than the size of the correct program (total number of nodes). Otherwise, if the correct program is smaller than the incorrect, we suggest statements to be removed. Note that, in practice, the number of addition or removal suggestions can be large if the core of both programs are similar but they have differences in implementation. A common example in our experiments is learners who reutilize their own implementation of a console manager for reading from and writing to console. Other learners exploit utility classes to

achieve the same behavior, e.g., `java.util.Scanner`. In such cases, even when the core of both programs is similar, there are a large number of non-aligned nodes that correspond to the ad-hoc console manager. If we remove such nodes, it is very unlikely that the resulting program would be correct. As a result, we only suggest nodes to be added or removed if they are directly or indirectly (one hop) connected with nodes suggested as replacements without taking direction into account. More formally, $v \in V_2 | v \notin \text{ran } \varphi$ is suggested as an addition or removal if $\exists u \in V_1 | (u, \varphi(u)) \in P \wedge |\text{Path}_U(\varphi(u), v, G_2)| \leq 1$, where P is the set of replacement suggestions and $\text{Path}_U(u, v, G)$ is the shortest path between nodes u and v in the undirected version of graph G .

We adapt edge correctness [8] (EC) to compute a global similarity between program dependence graphs that measures the number of edges that are preserved in an approximate alignment φ , which is defined as follows: $EC(\varphi) = \sum_{(u_i, u_j) \in E_1} IP(u_i, u_j, \varphi, E_2) / |E_1|$, where $IP(u_i, u_j, \varphi, E_2) = 1$ iff $(\varphi(u_i), \varphi(u_j)) \in E_2 \wedge l_e^1((u_i, u_j)) = l_e^2((\varphi(u_i), \varphi(u_j)))$; otherwise, $IP(u_i, u_j, \varphi, E_2) = 0$.

Table 1. Summary statistics of CodeChef assignments

	Id	#C	#I	LOC	#V	#E
BUYING2	BU	861	741	43.4 ± 29.9	45.2 ± 25.7	108.7 ± 62.4
CARVANS	CA	719	1,122	36.6 ± 28.0	37.0 ± 23.7	91.0 ± 57.7
CLEANUP	CL	1,650	889	55.4 ± 29.0	57.4 ± 23.1	154.6 ± 66.9
CONFLIP	CO	1,203	450	41.8 ± 30.7	39.3 ± 25.1	81.4 ± 62.3
JOHNY	JO	1,534	454	39.3 ± 28.3	40.3 ± 24.4	99.3 ± 65.0
LAPIN	LA	561	288	49.6 ± 32.3	53.6 ± 28.3	125.4 ± 78.3
MUFFINS3	MU	2,394	527	23.6 ± 27.4	20.5 ± 24.0	40.2 ± 63.0
PERMUT2	PE	1,890	1,083	41.7 ± 28.4	38.3 ± 22.4	89.1 ± 55.8
SUMTRIAN	SU	1,883	1,032	49.3 ± 28.4	52.5 ± 23.2	147.5 ± 60.7

5 Evaluation

We focus on nine assignments from CodeChef (<https://codechef.com>) shown in Table 1, where #C and #I entail the total number of correct and incorrect programs, respectively; LOC, #V and #E stand for average and standard deviation of lines of code, and nodes and edges in the program dependence graphs, respectively. Programs were collected in Nov, 2017. All CodeChef assignments follow the same structure: each test case must be read from console by a given program, and such test case consists of a single block of text that requires parsing and, usually, involves more than one loop before performing any computations to solve the assignment at hand.

We aim to compare our flexible alignment approach with respect to rigid alignments used in state-of-the-art approaches: CLARA [3], Refazer [15] and Sarfgen [18]. We implemented a common repair framework with a number of variations in the search step as follows:

- In SameCDG (*SC*), a correct and an incorrect programs are considered only if there exists a graph isomorphism between their control nodes (Sarfgen).
- SameLoop (*SL*) is a relaxation of SameCDG such that any combination of control statements are allowed unless they are included in a loop (CLARA).
- Flexible (*FL*) ranks correct programs based on edge correctness with respect to an incorrect program and selects top- t correct programs.
- Rigid (*RI*) is more restrictive than *FL* since only edge correctness that belongs to the interval $[1, .85)$ are considered (Refazer).

We evaluate the power set of suggestions in an incremental way starting from the empty set with an upper limit l [19]. The repair process takes as input the lines of the incorrect program that are impacted, and adds, removes, and/or replaces them by lines in the correct program. When adding or replacing lines, variables can be different since we may compare programs coming from different learners. As a result, the repair process also evaluates all possible combinations of variables in the original incorrect program [19]. We perform this repair step for all possible combinations of correct-incorrect programs resulting from the search step depending on each approach (*SC*, *SL*, *RI* and *FL*). We evaluate all possible pairs of correct-incorrect programs; however, CLARA and Sarfgen use variable traces to select a single correct program.

We consider two scenarios in which we limit the number of correct programs available to repair incorrect programs sorted by submission date, ascending: 100% (C_{100}) and 25% (C_{25}). These sets simulate different stages in time of a given assignment in which we have only collected a partial number of correct programs. We deem $\alpha = .5$ and $k = 1$ as proper parameter values. Comparing C_{100} in Fig. 3a with respect to C_{25} in Fig. 3b, we observe a performance drop

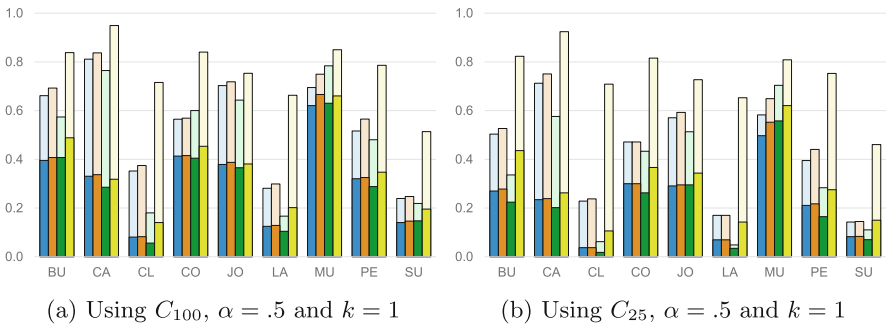


Fig. 3. Repairs achieved by the different approaches. From left to right, bars correspond to *SC*, *SL*, *RI* and *FL*, respectively. The Y axis presents the percentage of incorrect programs fully (darker color) and partially (lighter color) repaired.

for all approaches except for *FL*, which keeps competitive performance in both full and partial repairs. These results match our hypothesis that our flexible program alignment is appealing when there exist fewer correct programs for a given assignment. In C_{100} , we observe that *FL* is able to achieve more full repairs than *SC* and *SL* except in the CA, JO and MU assignments, where *SC* and *SL* perform better. In these assignments, there exist correct programs with the same loop structure that are suitable to repair incorrect programs; however, our ranking based on edge correctness does not promote these in favor of other correct programs with a different loop structure but similar semantics.

6 Related Work

CLARA [3] compares variable traces to cluster correct programs based on test cases. A single correct program is selected as a cluster representative. Each incorrect program is compared to every cluster representative based on variable traces to find the minimal repairs to transform from incorrect to correct. Sarfgen [18] searches for similar correct programs that share the same control flow structure as the incorrect program. To identify the best correct program to repair an incorrect program, it summarizes variable traces into vectors that are compared using Euclidean distance, so the correct program with the smallest distance is selected. Incorrect and correct programs are fragmented based on their control flows, and, for each fragment pair that is matched, potential repairs are computed using abstract syntax tree edits. CLARA and Sarfgen only consider pairs of programs whose control flow match, which is a hard constraint since such a pair may not currently be present in the set of correct programs, or such a control flow may not even be possible.

Refazer [15] proposes “if-then” rules to match and transform abstract syntax subtrees of a program. Such rules are synthesized from sample pairs of correct/incorrect programs, in which tree edit distance comparisons between correct and incorrect programs help identify individual transformations. A clustering algorithm finds transformations that can be abstracted away into the same rule. sk_p [13] relies on neural networks to repair incorrect programs. First, all variables in each program are renamed to tokens, and sk_p constructs partial fragments of three consecutive statements using these renamed tokens. The middle statements in fragments are removed and fed to the repairer for training, i.e., each training pair consists of the partial fragment without the middle statement and the full fragment. Given an incorrect program, sk_p computes all candidate statements to be fixed, which form a search space that needs to be explored to find all the necessary repairs. The order of statements is one of the main drawbacks of Refazer, Sarfgen, and sk_p: Refazer and Sarfgen rely on edit distances of abstract syntax trees, while sk_p treats programs as documents. Our approximate alignment allows to account for more implementation variability and flexible comparison of programs.

7 Conclusions

Nowadays, programming is perceived as a must-have skill. It is thus not surprising that the number of learners have scaled to millions, especially in online settings. Delivering feedback is addressed by repairing learners' incorrect programs. The trend in data-driven approaches is to perform a rigid matching between correct and incorrect programs to discover snippets of code with mending capabilities. The downside is that potential repairs that could be captured by looser alignments may be missed. This paper explores using a flexible alignment between statements in pairs of programs to discover potential repairs. We compare flexible with respect to rigid program comparisons. The former is capable of repairing more programs than rigid schemes, which supports our hypothesis that rigid approaches might be missing valuable code snippets for reparation that could be discovered by an approximate method otherwise. As a result, we claim that "search, align and repair" approaches should rely on flexible alignments to improve their repair capabilities.




References

1. Coetzee, D., Fox, A., Hearst, M.A., Hartmann, B.: Should your MOOC forum use a reputation system? In: CSCW, pp. 1176–1187 (2014)
2. Garcia, D.D., Campbell, J., DeNero, J., Dorf, M.L., Reges, S.: CS10K teachers by 2017?: Try CS1K+ students now! coping with the largest CS1 courses in history. In: SIGCSE, pp. 396–397 (2016)
3. Gulwani, S., Radicek, I., Zuleger, F.: Automated clustering and program repair for introductory programming assignments. In: PLDI, pp. 465–480 (2018)
4. Horwitz, S., Reps, T.W.: The use of program dependence graphs in software engineering. In: ICSE, pp. 392–411 (1992)
5. Jawalkar, M.S., Hosseini, H., Rivero, C.R.: Learning to recognize semantically similar program statements in introductory programming assignments. In: SIGCSE, p. 1264 (2021)
6. Khan, A., Wu, Y., Aggarwal, C.C., Yan, X.: NeMa: fast graph search with label similarity. PVLDB **6**(3), 181–192 (2013)
7. Kirschner, P.A., Sweller, J., Clark, R.E.: Why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educ. Psychol.* **41**(2), 75–86 (2006)
8. Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., Pržulj, N.: Topological network alignment uncovers biological function and phylogeny. *RSIF* **7**(50), 1341–1354 (2010)
9. Marin, V.J., Pereira, T., Sridharan, S., Rivero, C.R.: Automated personalized feedback in introductory Java programming MOOCs. In: ICDE, pp. 1259–1270 (2017)
10. Marin, V.J., Rivero, C.R.: Clustering recurrent and semantically cohesive program statements in introductory programming assignments. In: CIKM, pp. 911–920 (2019)
11. Monperrus, M.: Automatic software repair: a bibliography. *CSUR* **51**(1), 17:1–17:24 (2018)
12. Piech, C., Huang, J., Nguyen, A., Phulsuksombati, M., Sahami, M., Guibas, L.J.: Learning program embeddings to propagate feedback on student code. In: ICML, pp. 1093–1102 (2015)

13. Pu, Y., Narasimhan, K., Solar-Lezama, A., Barzilay, R.: sk_p: a neural program corrector for MOOCs. In: SPLASH, pp. 39–40 (2016)
14. Rodriguez, C.O.: MOOCs and the AI-Stanford like courses: Two successful and distinct course formats for massive open online courses. *EURODL* **15**(2) (2012)
15. Rolim, R., et al.: Learning syntactic program transformations from examples. In: ICSE, pp. 404–415 (2017)
16. Sankowski, P.: Maximum weight bipartite matching in matrix multiplication time. *TCS* **410**(44), 4480–4488 (2009)
17. Singh, R., Gulwani, S., Solar-Lezama, A.: Automated feedback generation for introductory programming assignments. In: PLDI, pp. 15–26 (2013)
18. Wang, K., Singh, R., Su, Z.: Search, align, and repair: data-driven feedback generation for introductory programming exercises. In: PLDI, pp. 481–495 (2018)
19. Xin, Q., Reiss, S.P.: Leveraging syntax-related code for automated program repair. In: ASE, pp. 660–670 (2017)
20. Zhang, S., Tong, H.: FINAL: fast attributed network alignment. In: KDD, pp. 1345–1354 (2016)
21. Zweben, S., Bizot, B.: 2015 Taulbee Survey. Tech. rep, Computing Research Association (2016)



Interaction of Human Cognitive Mechanisms and “Computational Intelligence” in Systems that Support Teaching Mathematics

Sergei Pozdniakov¹(✉) , Ilya Posov^{1,2}(✉) , and Chukhnov Anton¹(✉) 

¹ Saint Petersburg Electrotechnical University LETI, Saint Petersburg, Russia

² Saint Petersburg State University, Saint Petersburg, Russia

Abstract. The article presents an analysis of a number of projects in which the authors of the article participated. All projects are related to computer support for the productive activities of students in the process of studying mathematics. All analyzed systems can be considered as systems that make human cognitive mechanisms interact with machine “computational intelligence”. Based on the analysis of projects, various ways of coordinating the user’s mental activity with the pseudo-intellectual behavior of systems, which provide an intelligent dialogue in the process of solving mathematical problems, are identified. It is shown that the support of productive activity of students, including research and constructive activity, requires non-standard computer systems, and that the separation of intellectual operations between a learning system and a person supports the productive components of the learning process and initiates the development of new pedagogical approaches.

Keywords: Productive learning · Meaning transfer · Intelligent dialogue · Computational intelligence · Verification · Different representations of mathematical knowledge · Human cognitive functions

1 Introduction

When designing an intelligent learning support system (LSS), one should answer the following question: how to distribute cognitive mechanisms between a human and an LSS? If the system has poorly developed means to support a subject dialogue (in its expanded understanding), it would not make significant changes in the learning process. If the system, on the contrary, is capable of performing meaningful actions, there is a danger that its users will not show sufficient

The study was carried out with the financial support of the Russian Foundation for Basic Research within the framework of the scientific project No. 19-29-14141: study of the relationship between conceptual mathematical concepts, their digital representations and meanings as the basis for the transformation of school mathematics education.

© Springer Nature Switzerland AG 2021

A. I. Cristea and C. Troussas (Eds.): ITS 2021, LNCS 12677, pp. 259–266, 2021.

https://doi.org/10.1007/978-3-030-80421-3_28

intellectual initiative, because the system will take it upon itself, and intellectual functions of users will degrade. In recent years, the problem of transferring conceptual mathematical knowledge through the use of a computer tool has attracted more and more attention. Some theoretical generalizations are presented in the work [8].

Another important aspect of the development of an intelligent LSS is its focus on the users, and the desire of developers to use their psychophysical characteristics. There is a danger that the system will know too much about a person interacting with it, which can potentially become a source of negative impacts, either because of the possible unauthorized access to this information, or because of an insufficiently competent use of the system by the authors: it is enough to recall the first computer training systems: sometimes, according to the points scored in the test, the systems gave offensive reactions to their user.

If we accept this limitation about collecting sensitive data, then the design of intelligent LSS should be based only on modeling the subject area, and the system should only analyze the user's actions within this subject area. So the development of such systems should be based on the field of teaching methods, and not on the field of psychology. The other corollary of this limitation is a principle, stating that an intellectual LSS should be based on cognitive mechanisms of a user instead of substituting them.

The report will consider the so-called “computational intelligence” and its use in intelligent LSS. By computational intelligence, we mean the ability of a computer to perform calculations with various mathematical objects much faster than a person, and an ability to present results in such a way, that users consider them as results of intellectual activity. As an example, let us mention one of examples that will be considered later: the ability of the system to determine the correctness of the solution of a human-made problem without specifying the correct answer.

2 Using “Computational Intelligence” in Mathematical Research

The paper [20] demonstrates how mathematicians interact with computational abilities of computers to make progress in the number theory. At first glance, it seems that the interaction of a mathematician with a computer is obvious in this case: instead of inventing new mathematical constructions, a researcher makes a huge amount of computations with already known constructions. However, this is not quite true. Here is what the authors write about this [20] (pp. 20–23): “It is interesting that today we can repeat all the classical calculations on this topic, performed in the 18th–19th centuries, in a few minutes on a household computer”. “But in order to repeat the calculations by Western, you need to think a little about the algorithms used” [20] (p. 18).

The reason is that with a straightforward approach, the operating time of a computer is unrealistic to solve the problem. When human cognitive mechanisms are included in the process, they make it possible to reduce the search

so much that it is performed in a reasonable time. The authors note that the presence of computational capabilities, which, unlike the first computers, operate not just with numbers, but with more complex objects, for example, polynomials in symbolic form, stimulate mathematicians to look for new ways of operating with mathematical structures that are already “adapted” for further processing by “computational intelligence”, which in this case means computer algebra systems. One might question whether they may be considered as intelligent systems, but the example of WolframAlpha convinces that computational power can be interpreted by humans as an intelligent response.

3 Using “Computational Intelligence” to Support Solving Mathematical Tasks

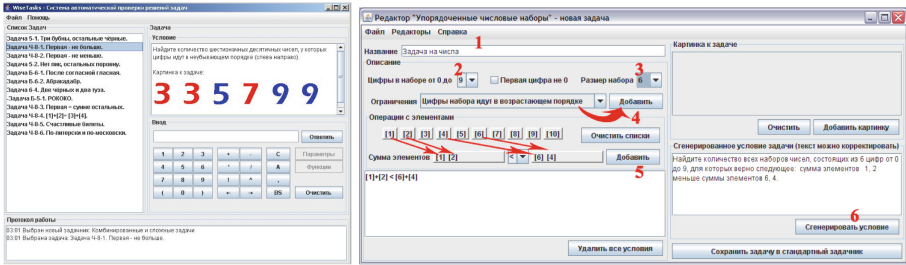
At first glance, the above example may seem far from education. Let us consider the systems of self-checking tasks, united by the common name Wise Tasks. They are described in articles [2,3,17]. The main idea of these systems is that they allow the user to describe a statement of a combinatorial, geometric, etc. problem so that this statement is a formal description of the problem. After entering such a description into the system, its author or any other user may solve this problem, and the system will check the correctness of the answer. The feature of the system is that the author of the problem does not enter its reference solution; moreover, he or she usually does not know the solution of the problem. At the same time, the system allows checking the entered answers using the description of the problem, presented in a special form, close to the usual human description.

Let’s consider the steps that a users makes while working the system. At the first step, a user prepares the task for processing. For this he or she actually composes an exact mathematical description of the problem in a convenient interface. Note that such work is not reproductive and requires mathematical conclusions, different from those that are supposed by the problem books in mathematics, but no less thoughtful and meaningful. This step, new for mathematical teaching, can be called modeling. After entering the task into the system, the user begins to solve it like the above-mentioned tasks from school problem books. The system test whether the answer conforms to the task description and generates a response to the answer Fig. 1.

Thus, the solution to the problem is carried out by the interaction of human intelligence with the “computational intelligence”. This opens up new possibilities for creativity: the user can vary the task, simplify it, consider similar ones and solve them by interacting with the system.

4 Research Story Problems in STEM Education

A number of articles by the authors of this paper are devoted to the development of various methods of applying the verification principle [7,11,12]. Let us consider one typical example.



(a) The task solving interface (b) The task authoring interface, simplified for a certain tasks type

Fig. 1. The WiseTasks combinatorics system

The use of the idea of verification in the project described below is associated with using computational intelligence for solving NP-hard problems in real time, which are auxiliary for solving more complex NP-hard problems.

Let us consider the interaction of human and computational intelligence in solving mathematical research problems [6, 19]. The articles present a system for supporting research tasks within the framework of the remote International Competition for the Application of ICT in the Natural Sciences, Technologies and Mathematics “Construct, Test, Explore” (CTE). An example problem “Mathematical rock climbing” is taken from [18]. The question is to place 16 points in a square so that the shortest path connecting them is as long as possible. Note that the problem of constructing the shortest path is NP-hard. To solve the task, participants were supposed to use various intuitive considerations, including considerations of symmetry and analyzing the results of experiments. The Fig. 2 from [18] shows the best solution found by only two participants out of several thousands. It should be noted that the optimal solution was unknown not only by the participants, but also by the jury of the competition.

Thus, this example shows how a computer, and a participant interact in conducting an educational research, splitting intellectual processes. Without fast processing of the participant’s hypothesis by the system, the participant would not be able to build the required path needed for further steps of the study.

5 Supporting Constructive Work as a Way to Introduce a Theoretical Problem

As the last project, we will consider a support system for the competition in discrete mathematics and theoretical computer science, in which the idea of multiple representation of mathematical objects is combined with the idea of verification.

Consider the system that supports the competition in discrete mathematics and theoretical computer science (DMaTCS) [1, 4] Let us discuss in detail one example of using different interpretations to help students understand related

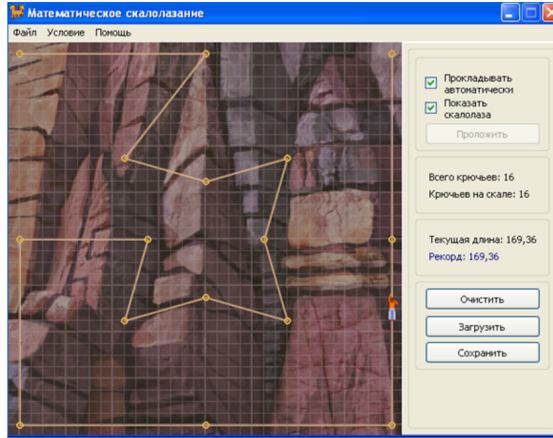


Fig. 2. An example of the best solution to the minimax problem found by the participant with the support of “computational intelligence” [18]

ways of representing sets of words, such as: regular expressions, finite-state machines, Turing machines.

A Task Example. Describe the set of words consisting of letters a and b , that may be split into alternating blocks of similar letters of the odd length, for example, $abbbaaaaab$.

This task can be supported by three different environments: the environment to work with regular expressions, the environment to work with finite state machines, the environment to work with turing machines.

The tool for verifying the proposed solution is used here as computational intelligence. Moreover, the interface uses various methods of generating examples for verification: it can either use examples provided by the user or use automatically generated examples. In this work, the tasks are used independently, however, developing the idea of integrating the human and machine intelligences, it is possible to propose a variant of combining different representations with a single interface, then verification can be done without comparing the answer with the reference, but comparing the answers for different representations with each other. This part will be done by “computational intelligence”. The user will only have to analyze examples on which there different representations don’t agree and look for an error in his or her mathematical reasoning and constructions.

6 Discussion

In order to check the validity of the generalizations made above, let us analyze the class of widespread and well-known systems: the dynamic geometry software. This class of learning systems is one of the few that have earned worldwide recognition and the idea of which is implemented in dozens of different systems,

united by the generic concept of “dynamic geometry”. If in the basis of these systems we find the features highlighted above, this can be considered a serious argument in favor of the validity of the theoretical analysis performed.

Dynamic geometry systems make it possible to verify the proposed solutions to geometric problems. Moreover, the verification mechanism is based not so much on the capabilities of the system as on the capabilities of a person. The solution created in the form of a geometric structure allows to visually see the properties of this solution, and by moving the structural elements, one can clearly see the existence of invariants that are specified by the problem statement [9]. It should be noted that the point that geometry studies is precisely invariants under various transformations, and it is the determination of invariants that is an important but difficult task for artificial intelligence systems. In connection with this, one may note an example that Seymour Papert gives using another tool (a turtle) to derive the invariance of the inscribed angle [16].

Thus, an important element of such systems is the transformation of the machine (internal) representation of an object into a form that can be considered as a preprocessing for the subsequent activation of human cognitive mechanisms.

This approach provides one of the possible solutions to the problem associated with the danger of introducing computer systems that will duplicate the intellectual functions of a person and thereby hinder the development of intellectual mechanisms of humans.

7 Conclusion

The use of machine intelligence in its current state in combination with human intelligence makes it possible to create intelligent learning systems that do not hinder the development of human intelligence, but give it new fields for development.

Support for intellectual dialogue in the learning process can be achieved through the splitting of information processing functions between a person and a system. When creating a system, the following restrictions must be taken into account: 1) the initiative to interact with the system must belong to the user; 2) the system should not collect information that may harm the user or may be indirectly used for such purposes.

To support productive learning, the separation of functions between the user and the system can be implemented as follows: 1) the system performs computational operations related to the generation and preprocessing of information, including verification on a set of specific examples of theoretical hypotheses formulated by the user; 2) the system presents the results of calculations in a form that initiates human cognitive mechanisms associated to a large extent with vision.

References

1. Akimushkin, V.A., Pozdniakov, S.N., Chukhnov, A.S.: Constructive problems in the structure of the Olympiad in discrete mathematics and theoretical informatics. *Olympiads Inform.* **11**, 3–18 (2017)
2. Bogdanov, M., Pozdnyakov, S., Pukhov, A.: Multiplicity of the knowledge representation forms as a base of using a computer for the studying of the discrete mathematics. *PEDAGOGIKA* **96**, 136–142 (2009)
3. Bogdanov, M.S.: Avtomatizaciya proverki resheniya zadachi po formal'nomu opisaniyu ee usloviya/zh. *Komp'yuternye instrumenty v obrazovanii* **4**, 51–57 (2006)
4. Chukhnov, A.S.: Constructive tasks as a tool of invasive and non-invasive assessment of knowledge. *Comput. Tools Educ.* **3**, 96–104 (2019). <https://doi.org/10.32603/2071-2340-2019-3-96-10>
5. Chukhnov, A., Maytarattanakhon, A., Posov, I., Pozdniakov, S.: Constructive graph tasks in distant contests. *Inform. Educ.* **19**(3), 343–359 (2020). <https://doi.org/10.15388/infedu.2020.16>
6. Konkurs, E.S.B.: “Konstruiruj, Issleduj, Optimiziruj” i ego vliyanie na razvitie tvorcheskih sposobnostej shkol'nikov. *Komp'yuternye instrumenty v shkole* **2**, 18–31 (2020)
7. Ivanov, S.G.: Rabota na urokah matematiki so sredoj Verifier. *Komp'yuternye instrumenty v obrazovanii* **1**, 58–66 (1998)
8. Kynigos, S., Lagrange, J.-B.: Cross-analysis as a tool to forge connections amongst theoretical frames in using digital technologies in mathematical learning. *Educ. Stud. Math.* **85**, 321–327 (2014). <https://doi.org/10.1007/s10649-013-9521-3>
9. Laborde, C.: Relationships between the spatial and theoretical in geometry: the role of computer dynamic representations in problem solving. In: Tinsley, D., Johnson, D.C. (eds.) *Information and Communications Technologies in School Mathematics*. ITIFIP, pp. 183–194. Springer, Boston, MA (1998). https://doi.org/10.1007/978-0-387-35287-9_22
10. Maier, V.R.: Programmirovaniye kak instrument poznaniya v kurse geometrii. *Informatika i obrazovanie* **5**, 15–18 (1997)
11. Mancеров, D.I.: Sreda Verifier-KD: verifikaciya reshenij zadach po matematike. *ZH. “Komp'yuternye instrumenty v obrazovanii”* **4**, 36–41 (2006)
12. Mancеров, D.I.: Sistema verifikacii dlya parametricheskikh klassov zadach po matematike. *ZH. Nauchno-tehnicheskie vedomosti SPbGPU. Informatika, telekommunikacii, upravlenie*, **5**(65), 183–189 (2008)
13. Maytarattanakhon, A., Posov, I.A.: Automation of distance contests based on research problems in mathematics and informatics. *Comput. Tools Educ.* **6**, 45–51 (2014). (in Russian)
14. Marvin, M.: A framework for representing knowledge. In: Winston, P.H. (ed.) *The Psychology of Computer Vision*. McGraw-Hill, New York (1975)
15. Minsky, M.: *The Society of Mind*. Simon and Schuster, New York (1987)
16. Papert, S.: An exploration in the space of mathematics educations. *Int. J. Comput. Math. Learn.* **1**(1), 95–123 (1996). <https://doi.org/10.1007/BF00191473>
17. Perchenok, O.V., Pozdnyakov, S.N., Posov, I.A.: Avtomatizaciya proverki resheniya geometricheskikh zadach po opisaniyu ih uslovij na predmetno-orientirovannom yazyke. *Komp'yuternye instrumenty v obrazovanii*. - SPb. **1**, 37–44 (2012)

18. Posov, I.A.: Razbor zadach “Zadacha otshel’nikov” i “Matematicheskoe skalozazanie” konkursa KIO-2009. *Komp’yuternye instrumenty v shkole.* - SPb. **4**, 20–27 (2009)
19. Pozdniakov, S.N., Posov, I.A., Puhkov, A.V., Tsvetova, I.V.: Science popularization by organizing training activities within the electronic game laboratories. *Int. J. Digital Literacy Digital Competence* **3**(1), 17–31 (2012)
20. Vavilov, N.A.: Komp’yuter kak novaya real’nost’ matematiki. *Komp’yuternye instrumenty v obrazovanii* **3**, 1–45 (2020)
21. Wing, J.M.: Computational thinking. *Commun. ACM* **49**, 33–35 (2006)



Learning Path Construction Using Reinforcement Learning and Bloom's Taxonomy

Seounghun Kim, Woojin Kim, and Hyeoncheol Kim^(✉)

Department of Computer Science and Engineering, Korea University,
Seoul, South Korea

{ryankim0409,wojinkim1021,harrykim}@korea.ac.kr

Abstract. Massive Open Online Courses (MOOC) often face low course retention rates due to lack of adaptability. We consider the personalized recommendation of learning content units to improve the learning experience, thus increasing retention rates. We propose a deep learning-based learning path construction model for personalized learning, based on knowledge tracing and reinforcement learning. We first trace a student's knowledge using a deep learning-based knowledge tracing model to estimate its current knowledge state. Then, we adopt a deep reinforcement learning approach and use a student simulator to train a policy for exercise recommendation. During the recommendation process, we incorporate Bloom's taxonomy's cognitive level to enhance the recommendation quality. We evaluate our model through a user study and verify its usefulness as a learning tool that supports effective learning.

Keywords: Personalized learning · MOOC · Knowledge tracing · Reinforcement learning · Learning path construction · Bloom's taxonomy

1 Introduction

The Massive Open Online Courses (MOOCs) reduces traditional education's limitations on time and location, increasing the accessibility and opportunity of an education. Students without background knowledge may easily register for a course and learn the subject. However, MOOCs suffer from a long-lingering problem of low retention rates. The main elements affecting the low retention rate are lack of personalized learning, complexity, and MOOC participants' diversity [4]. The low retention rate is also found to be affected by the low quality of MOOC content and structure, lack of usability in tools and learning environments, and high levels of self-organization [19]. Accordingly, MOOC course design, which includes course content, structure, and information delivery technology, is considered another predictor of student retention [10]. Diversity of learners also affects retention rate, that one-size-fits-all content of MOOCs is insufficient to

adapt to individual learner’s needs [12]. By assigning learning resources based on the student’s need, the resulting versatile adaptation of the learning environment achieves personalization in MOOCs [3]. MOOC literature addresses that adaptive and personalized learning should be considered essential and incorporated into MOOC services [3, 21].

There are attempts to provide personalized learning through learning path construction using knowledge tracing models and reinforcement learning [7]. Here, we regard a learning path as a sequence of recommended exercises to accelerate learning. However, existing approaches only recommend exercise sequences that maximize the student’s predicted probability of giving the correct answer, without considering the exercise difficulty [14, 15]. Exercises on a concept may require different cognitive levels. An exercise could test a student’s ability to remember knowledge that requires a low cognitive level or applying the knowledge that requires a high cognitive level. Therefore, constructing personalized learning paths should regard the exercises’ cognitive levels for a high-quality recommendation.

We propose a learning path construction model that generates an optimal learning sequence based on knowledge tracing and deep reinforcement learning, augmented with Bloom’s taxonomy. We utilize Dynamic Key-Value Memory Network and Trust Region Policy Optimization (TRPO) to accomplish this task [16, 22]. We model student knowledge with the Dynamic Key-Value Memory Network based on the student history data and generate an optimal learning path filtered with Bloom’s taxonomy, suggesting an efficient learning plan. We verify the adequacy and value of practically deploying the proposed model by conducting a user study on MOOC stakeholders, teachers, and students.

Our work’s contribution is threefold: first, our work is the first approach to integrate deep learning-based knowledge tracing, deep reinforcement learning-based learning path construction, and Bloom’s taxonomy to improve the generation quality of learning paths. Second, we build a bridge between deep learning technology and educational theory by joining deep knowledge tracing, deep reinforcement learning, and Bloom’s taxonomy. Third, we analyze the meaning and expectations of applying learning technology from the perspective of educational professionals.

2 Related Work

In this section, we focus works on deep learning application to education and introduce Bloom’s taxonomy. We present deep knowledge tracing and deep reinforcement learning approaches to learning path construction.

2.1 Deep Learning-Based Knowledge Tracing

Knowledge Tracing (KT) attempts to model a student’s knowledge state based on the observed performance of the student [6]. The observed performance in the system is mainly the student’s response to a question on a certain concept. The

goal of the knowledge tracing is to model the student's latent knowledge state and use the information to predict student behavior or to provide appropriate assistance in learning [11].

Current knowledge tracing methods adopt deep learning approaches, improving performance in general. Deep Knowledge Tracing (DKT) models the knowledge state in the form of student response prediction. DKT uses the Long-Short Term Memory (LSTM) architecture to convey the sequential nature of student interaction data [13]. Dynamic Key-Value Memory Network (DKVMN) develops upon the DKT using memory network architecture [22]. DKVMN's static key memory stores knowledge concepts, and dynamic value memory stores students' knowledge states on the corresponding concepts. DKVMN models the student's knowledge states of latent concepts with a key-value memory, demonstrating superior performance compared to DKT.

However, knowledge tracing methods are limited to estimating student knowledge to predict student responses to questions. This limitation requires humans to comprehend and interpret the prediction for meaningful utilization of models for educational implementation. For instance, the model could produce a student's estimated knowledge level on knowledge concepts, but it is up to humans to instruct appropriate learning materials.

2.2 Personalized Learning Path Construction with Reinforcement Learning

Constructing a personalized learning path is achieved by determining how to adaptively sequence various instructional activities to assist in learning. With the student's estimated knowledge state, the personalized recommendation of the learning resources is formulated as generating an optimal sequence of instructions in the system. The goal of learning path construction is to compute a policy with reinforcement learning methods that select the best learning materials that maximize the student's knowledge state on a particular knowledge concept.

Applications of reinforcement learning-induced methods are effective at generating meaningful instructional policies in various learning tasks that benefit student learning [7]. Rafferty et al. find the optimal learning policy by formulating the task as a Partially Observable Markov Decision Process (POMDP) to accelerate learning [14]. Reddy et al. investigate the model-free reinforcement review scheduling algorithm that learns flexible and scalable teaching policies to maximize learning [15].

Recent works merge deep knowledge tracing models with deep reinforcement learning with consideration of knowledge concepts. Ai et al. proposes a concept-aware knowledge tracing model with exercise recommendation using reinforcement learning, demonstrating better performance in maximizing student's knowledge level [1]. Liu et al. accomplish adaptive learning through joining the Knowledge Tracing module, Cognitive Navigation module, and Actor-Critic Recommender module [11]. However, these works only consider knowledge concepts and its hierarchical structure of a question, overlooking the question complexity that assesses students' cognitive skills.

Table 1. Bloom’s Taxonomy and its corresponding action verbs [2]

Objective	Verbs
Remembering	Choose, Define, Find, Recall, Relate, Select, Show, Spell, Tell
Understanding	Classify, Compare, Contrast, Demonstrate, Explain, Relate, Interpret, Rephrase, Show, Summarize, Translate
Applying	Apply, Build, Choose, Construct, Develop, Identify, Interview, Make use of, Model, Organize, Plan, Select, Solve, Utilize
Analyzing	Analyze, Assume, Categorize, Classify, Compare, Contrast, Dissect, Distinguish, Divide, Examine, Function, List, Simplify, Survey
Evaluating	Agree, Appraise, Assess, Award, Compare, Conclude, Decide, Deduct, Disprove, Estimate, Evaluate, Measure, Perceive, Prove
Creating	Adapt, Combine, Compile, Construct, Create, Design, Discuss, Estimate, Imagine, Improve, Invent, Modify, Predict, Propose

2.3 Bloom’s Taxonomy in Learning Path Recommendation

Bloom’s taxonomy classifies cognitive skills of students into six categories, ranging from low-level skills to high-level skills [5]. The hierarchy of revised Bloom’s taxonomy and its corresponding action verbs are presented in Table 1 [2]. The classification of progressive complexity of cognitive levels is widely used to construct learning objectives that students are expected to master [18]. Bloom’s taxonomy is further utilized to classify learning contents associated with learning objectives for cognitive skill acquisition [17]. A number of approaches have integrated Bloom’s taxonomy during the process of learning path recommendation. Yang et al. formulate learning activities and assessments by Bloom’s taxonomy, proposing an outcome-based learning path model [20]. Govindarajan et al. use Bloom’s taxonomy to determine the proficiency level and cluster students into groups [9]. However, these learning path construction approaches are limited to rule-based algorithms and do not use Bloom’s taxonomy during learning path recommendation.

3 Learning Path Construction with Bloom’s Taxonomy

We propose a deep learning approach to recommend exercises, generating a personalized learning path. Figure 1 illustrates the overall framework of our model. We use all students’ learning histories of the MOOC and the exercise level information, labeled with Bloom’s taxonomy. For modeling student knowledge, we

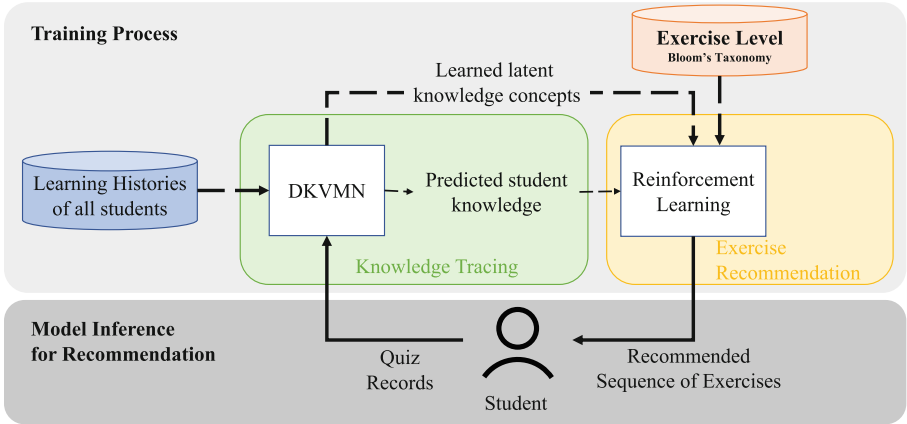


Fig. 1. The training process and inference process for learning path construction.

use Dynamic Key-Value Memory Network. We train the DKVMN model with student learning history, which results in learned latent knowledge concepts to predict student knowledge state. Then, the recommendation policy is trained using the DKVMN model as the reward function. Based on the student's estimated knowledge state, the learning path construction task is framed as a Partially Observable Markov Decision Process (POMDP) planning problem, and the optimal policy is found by TRPO algorithm [16]. Finally, the recommended exercises are filtered by the complexity of the learning content classified based on Bloom's taxonomy, depending on the student's current level. The trained model is then used to construct a sequence of exercises based on a new student's quiz records.

Our work mainly builds up from the model proposed by Ai et al.[1]. The model presents a reinforcement learning approach using the concept-aware Dynamic Key-Value Memory Network as the reward function. The knowledge concepts are predefined and then fed into the process of knowledge tracing and learning path construction. A difference between the model and our proposed method is integrating exercise hierarchy based on Bloom's taxonomy. Our proposed method first learns the latent knowledge concepts based on learning history data, then incorporates the exercise hierarchy information for learning path construction.

We divide Bloom's taxonomy into two categories, 'Remembering & Understanding' and 'Applying & Analyzing', each corresponding to 'Low' and 'High' to apply our model to the Edwith e-learning platform¹. The Edwith is a MOOC for software development education created by the Connect Foundation, Naver Corporation of South Korea. The platform hosts 320 online courses on software development and targets job seekers who did not pursue a major in computer science or software development.

¹ <https://www.edwith.org>.

Table 2. Exercise filtering algorithm using Bloom’s taxonomy

Step 1	Given a student quiz history, calculate a cognitive level (low or high) as weights by complimenting correct answers and penalizing incorrect answers
Step 2	For high cognitive level students, calculate the length of low level exercises to be recommended by dividing high cognitive weight by low cognitive weight
Step 3	Using the learned policy and a student simulator, generate a recommended exercise until a desired length of learning path is reached
Step 4-a	For high-cognitive level student, incorporate low-level exercise to the learning path until the low-level exercise length from step 2 is reached. High-level exercises are added for the remaining length of the learning path
Step 4-b	For low-cognitive level student, incorporate only low-level exercises to the learning path
Step 5	As the student solves the recommended exercise, repeat the process from step 1 by re-evaluating the cognitive level of the student

Exercises of Edwith courses rarely include questions that require ‘Evaluating’ or ‘Creating’ level of cognitive skills. Instead, most of the exercises fall under the ‘Understanding’ or ‘Applying’ level. Therefore, we disregard ‘Evaluating’ and ‘Creating’ level of Bloom’s taxonomy in our model and use ‘Low’ and ‘High’ for labeling each exercise’s required cognitive level during the recommendation process.

Our algorithm for incorporating Bloom’s taxonomy is as follows. Based on a student’s quiz history, the cognitive level of the student is determined based on a threshold derived from student score distribution. Then, given the student’s cognitive level, the recommended exercises are filtered. For instance, a student with a low cognitive level receives exercise recommendations of low cognitive levels, and a student with a high cognitive level receives learning paths with high-level exercises. Furthermore, for students with medium cognitive levels, the number of low cognitive exercises is determined based on the cognitive level’s degree. As the student progresses the course, the model keeps track of the student’s changing cognitive level. If the student’s cognitive level reaches ‘High’, the model recommends high cognitive level exercises. The complete process of the proposed algorithm is elaborated in Table 2.

We assume that students achieve cognitive skill when students give correct answers to questions corresponding to the skill. By incorporating the cognitive level of exercises, we consider a student’s cognitive skill development process during the recommendation process. Further, we acknowledge a different level of complexity in the learning contents on a single knowledge concept and

incorporate it into our model. Therefore, our model recommends learning paths that better considers the student’s current cognitive skill.

4 Experiment

4.1 Dataset

To evaluate the performance of the proposed model, we use data from Edwith. The Edwith is a MOOC service provider created by the Connect Foundation, Naver Corporation of South Korea. The platform hosts 320 online courses on software development and targets job seekers who did not pursue a major in computer science or software development.

The Edwith provides its specialized content, named Boost Course. Unlike lecture-based courses, Boost Course consists of lectures and their corresponding quizzes. To be certified for course completion, students must receive a perfect score in all questions in five trials. We used the student log data of the first trial for the experiment. Among the offered courses, we choose the learning history data from the CS50, an introductory computer science course taught at Harvard University. Edwith hosts the lectures with Korean translated subtitles and a self-developed quiz for each lecture. The concepts defined in the CS50 course are data structures, algorithms, control flows, file I/O, and basic computer architecture.

Table 3. Student log data statistics of CS50 course.

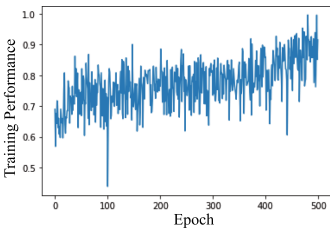
Statistics	Value
Number of students	1,984
Max sequence length	60
Average sequence length	32.25
Course completion rate	34.77%
Average sequence length of course completed students	51.59/60
Average quiz score of course completed students	84.36/100
Average sequence length of course uncompleted students	21.94/60
Average quiz score of course uncompleted students	33.87/100

Table 3 presents statistics of 1,984 unique student log data who are enrolled in CS50 course. The quiz sequence’s total length is 60, and the average number of quiz sequence solved by students is 32.25. The course completion rate is 34.77%. The average number of quiz sequences solved by students who received course certification is 51.59, with an average quiz score of 84.36 out of 100. The average number of quiz sequences solved by students who did not complete the course is 21.94, with a 33.87 average quiz score, confirming the relationship between student engagement and course completion.

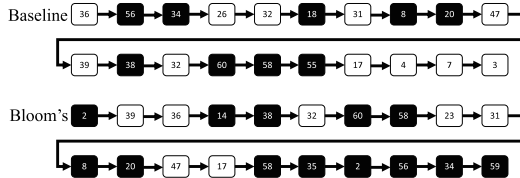
Table 4. Example concepts and exercises and its corresponding Bloom’s level.

Concept	Exercise	Bloom’s Level
Overflow	What is a problem caused by storing a numeric value outside of the range of a variable limit?	Low
Data structure	What is a data structure that follows first-in-first-out (FIFO) method?	Low
Algorithm	What is the Big-O of finding ‘John Doe’ with linear search in a telephone dictionary?	High
Algorithm	Given a list of [5, 6, 7, 3, 2], what is the first iteration of selection sort for sorting in ascending order?	High

We preprocess the data by breaking down each time step of CS50 student response history into a data instance to predict each student response at each time step. We also preprocessed skipped exercises as wrong answers. To use Bloom’s taxonomy information, we further labeled each exercise to a low or high cognitive level initially based on the exercise’s action verb. For exercises which its learning objective differs from its action verb, we relabeled the cognitive level of the exercise to match its learning objective to correct cognitive level. Table 4 describes sample exercises and Bloom’s cognitive level. The exercises are translated into English by authors.



(a) Training performance of reinforcement learning.



(b) Sample learning paths generated by the baseline model(top) and our proposed model(bottom). Each number indicates an exercise question.

Fig. 2. The training performance of solving with TRPO and sample learning paths generated from the baseline model and our model based on an average student’s learning history. The learning path generated by using Bloom’s taxonomy recommends high-level exercises at the end of the learning path to increase the cognitive level of the student.

4.2 Model Implementation

We construct the DKVMN with 50 key dimensions, 100 value dimensions, 50 summary dimensions, and 20 concept dimensions. The DKVMN is implemented

with the Pytorch library and trained with the distributed data-parallel method on a single GPU machine. We train the network for 200 epochs and early-stop when the validation performance does not increase for 40 epochs. We set the batch size to 512 and the learning rate to 0.001. The network is trained to minimize binary cross-entropy loss. The evaluation metrics for DKVMN are accuracy and Area under the ROC Curve (AUC).

Our model’s reinforcement learning system is implemented with the rllab’s off-the-shelf implementation in OpenAI Gym environment [8]. The model solves the POMDP problem with TRPO using a Gated Recurrent Unit (GRU) architecture [15]. We train the policy for 500 epochs.

4.3 Model Performance and Evaluation

We evaluate the performance of the DKVMN network using 5-fold cross-validation. The average accuracy is 0.9681, and the average AUC is 0.9929, verifying that the network successfully predicts future student response. We present the process of reinforcement training performance in Fig. 2a. The performance gradually increases to 90% while training for 500 epochs.

We present the generated learning paths from the baseline model of Ai et al. [1] and our proposed model in Fig. 2b. The learning paths are generated to recommend ten exercises, and each square of a learning path indicates an exercise. Black and white square indicate high and low Bloom’s level, respectively. We selected a random student from the test data to generate the learning paths. The student demonstrated average performance, scoring 30 out of 60 exercises. The baseline model’s recommended learning path is portrayed as somewhat random, recommending low cognitive exercise at the end of the learning path. Recommending low-level exercises at the end of the recommendation could result in frustration or boredom for a student. However, our model demonstrates a higher quality of learning path, recommending the appropriate ratio of low cognitive level exercises and suggesting high-level exercises at the end of the sequence.

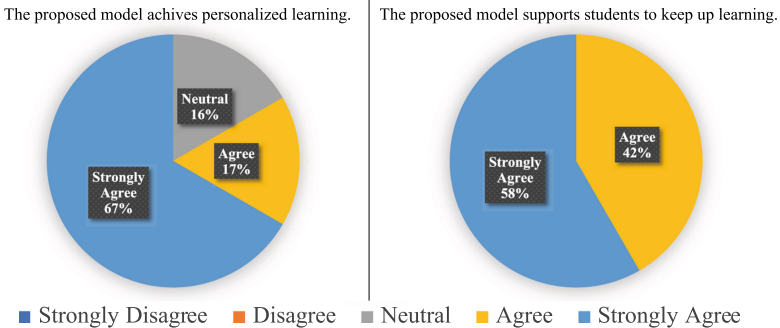


Fig. 3. Survey responses on the model’s value in providing personalized learning and supporting the continuation of learning.

5 User Study

We conducted a user study to explore the educational value of our model. The goal is to verify our model to MOOC stakeholders, teachers, and students. Although model performance could be a measure of effectiveness, we consider educational meaningfulness of the model as instrumental when applied to actual learning. Since the MOOC platform is for a vast range of audiences, we emphasize the expertise of educational professionals for in-depth observation and insights on different interests and viewpoints.

We construct the survey questions as per the respondent's profession. Having personalized learning as the main theme, we asked teachers about teaching methods using online platforms, course-level learning, and potential applications of our model in teaching methods. To students, we asked for a continuation of learning through MOOC, self-directed learning, and potential applications of our model in their learning. We asked MOOC stakeholders on retention and customer satisfaction. We surveyed a total of nine respondents, and survey responses are translated from Korean to English by the authors.

Figure 3 illustrates survey responses on two questions on a Likert scale. We first asked whether our model achieves personalized learning in MOOCs. 67% of respondents answered 'Strongly Agree' and 17% of respondents answered 'Agree'. Second, we asked whether our model supports students to keep up their learning in MOOCs. 58% answered 'Strongly Agree' and 42% of the respondents answered 'Agree', demonstrating our model's practicality in MOOC learning.

The survey participants were positive towards the model's application to achieve personalized learning and self-directed learning. Teacher A described that *"learning results from a continuous process. Current learning experience consists of previous learning and everyday learning experience leads to future learning. During this process, the main reason for dropout is the consumption of excess energy in determining the learning direction. If the model can reduce the burden of setting learning plans, it would be a great help"*. Teacher B commented that *"What determines the continuation in learning is students' intrinsic reward during the process of learning itself. The reward comes from the experience of success. Since the model is constructed to provide students with an experience of success, it is expected that the model will have a positive influence in increasing the continuation of learning"*. Student C responded that the *"Most crucial step of self-directed learning is constructing the sequence of learning contents. This level of thought requires a high level of metacognition. When the model replaces this process, students are expected to focus more and put more energy into the learning process itself"*.

We were able to confirm the needs of MOOC stakeholders on personalized recommendations. Stakeholder A stated that *"Providing appropriate information in the right place at the right time is the primary role of service providers. Therefore, the proposed service meets the user's need for a content recommendation based on their current state"*. The stakeholder B answered that *"We're analyzing how other similar service providers are implementing recommendation feature. I believe the model may be used for the personalized content recommendation"*

engine". Overall, the stakeholders addressed the necessity of a recommendation system for individualized learning and anticipate that the model will improve the learning experience. However, there exist pessimistic comments on the stability of the model in the real application. Primarily, teachers and MOOC stakeholders were cautious about deploying the model for practical learning, addressing the need for further verification of the model for safe implementation.

6 Conclusion

In this work, we proposed a learning path construction model using Bloom's taxonomy. We used a deep learning-based knowledge tracing model for estimating student knowledge state and used reinforcement learning to generate a sequence of recommended exercises. During the construction process, we integrated the cognitive level information of exercises categorized by Bloom's taxonomy, enhancing recommended exercises' quality. We evaluated that recommendation based on Bloom's taxonomy's cognitive hierarchy better meets the needs of students than recommendations without considering cognitive exercise levels. We anticipate that the model will effectively solve the absence of a personalization function in MOOCs that eludes its purpose as a practical learning tool.

Acknowledgements. This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques).

References

1. Ai, F., et al.: Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. International Educational Data Mining Society (2019)
2. Anderson, L.W., Bloom, B.S., et al.: A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. Longman(2001)
3. Assami, S., Daoudi, N., Ajhoun, R.: Personalization criteria for enhancing learner engagement in mooc platforms. In: 2018 IEEE Global Engineering Education Conference (EDUCON), pp. 1265–1272. IEEE (2018)
4. de Barba, P.G., Kennedy, G.E., Ainley, M.: The role of students' motivation and participation in predicting performance in a mooc. J. Comput. Assisted Learn. **32**(3), 218–231 (2016)
5. Bloom, B.S.: Taxonomy of educational objectives: the classification of educational goals. Cognitive domain (1956)
6. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. User Modeling User-adapted Interaction **4**(4), 253–278 (1994)
7. Doroudi, S., Aleven, V., Brunskill, E.: Where's the reward? Int. J. Artif. Intell. Educ. **29**(4), 568–620 (2019)
8. Duan, Y., Chen, X., Houthoof, R., Schulman, J., Abbeel, P.: Benchmarking deep reinforcement learning for continuous control. In: International Conference on Machine Learning, pp. 1329–1338 (2016)

9. Govindarajan, K., Kumar, V.S., et al.: Dynamic learning path prediction—a learning analytics solution. In: 2016 IEEE Eighth International Conference on Technology for Education (T4E), pp. 188–193. IEEE (2016)
10. Hone, K.S., El Said, G.R.: Exploring the factors affecting mooc retention: a survey study. *Comput. Educ.* **98**, 157–168 (2016)
11. Liu, Q., et al.: Exploiting cognitive structure for adaptive learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 627–635 (2019)
12. Onah, D.F., Sinclair, J., Boyatt, R.: Dropout rates of massive open online courses: behavioural patterns. In: EDULEARN14 Proceedings 1, pp. 5825–5834 (2014)
13. Piech, C., et al.: Deep knowledge tracing. *Adv. Neural Inf. Process. Syst.* **28**, 505–513 (2015)
14. Rafferty, A.N., Brunskill, E., Griffiths, T.L., Shafto, P.: Faster teaching by POMDP planning. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS (LNAI), vol. 6738, pp. 280–287. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21869-9_37
15. Reddy, S., Levine, S., Dragan, A.: Accelerating human learning with deep reinforcement learning. In: NIPS Workshop: Teaching Machines, Robots, and Humans (2017)
16. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: International Conference on Machine Learning, pp. 1889–1897 (2015)
17. Thomas, B., Chandra, J.: The effect of bloom’s taxonomy on random forest classifier for cognitive level identification of e-content. In: 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) pp. 1–6. IEEE (2020)
18. Ullah, Z., Lajis, A., Jamjoom, M., Altalhi, A., Saleem, F.: Bloom’s taxonomy: a beneficial tool for learning and assessing students’ competency levels in computer programming using empirical analysis. *Computer Applications in Engineering Education* (2020)
19. Vitiello, M., Walk, S., Hernández, R., Helic, D., Gütl, C.: Classifying students to improve mooc dropout rates. *Research Track*, p. 501 (2016)
20. Yang, F., Li, F.W., Lau, R.W.: A fine-grained outcome-based learning path model. *IEEE Trans. Syst. Man Cybern. Syst.* **44**(2), 235–245 (2013)
21. Yu, H., Miao, C., Leung, C., White, T.J.: Towards ai-powered personalization in mooc learning. *npj Sci. Learn.* **2**(1), 1–5 (2017)
22. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web, pp. 765–774 (2017)



Customizing Feedback for Introductory Programming Courses Using Semantic Clusters

Victor J. Marin¹, Hadi Hosseini², and Carlos R. Rivero¹(✉)

¹ Rochester Institute of Technology, Rochester, NY, USA
vxm4964@rit.edu, crr@cs.rit.edu

² Pennsylvania State University, State college, PA, USA
hadi@psu.edu

Abstract. The number of introductory programming learners is increasing worldwide. Delivering feedback to these learners is important to support their progress; however, traditional methods to deliver feedback do not scale to thousands of programs. We identify several opportunities to improve a recent data-driven technique to analyze individual program statements. These statements are grouped based on their semantic intent and usually differ on their actual implementation and syntax. The existing technique groups statements that are semantically close, and considers outliers those statements that reduce the cohesiveness of the clusters. Unfortunately, this approach leads to many statements to be considered outliers. We propose to reduce the number of outliers through a new clustering algorithm that processes vertices based on density. Our experiments over six real-world introductory programming assignments show that we are able to reduce the number of outliers and, therefore, increase the total coverage of the programs that are under evaluation.

Keywords: Graph clustering · Approximate graph alignment

1 Introduction

The number of novice programming learners has been steadily increasing for the last years in both traditional and online settings [1,4]. Traditional methods mainly rely on manual grading, and, as a result, are tedious, particularly for providing effective feedback to many novice learners [1]. There is also a need to assist instructors with the current “boom” in computing courses while continuing to provide a quality educational experience [9]. Current techniques to analyze learner programs mainly focus on automating feedback delivery, and they usually do not support an active role of the instructor [6]. Such an active

This material is based upon work supported by the National Science Foundation under Grant No. 1915404.

role can be reflected in many forms, e.g., by enabling a flexible grading scheme that is refined during the actual grading [3]. Additionally, the delivered feedback is internally decided by the tools and the instructor, which can provide very useful information regarding an assignment, has almost no opportunity to customize their feedback and attune it according to each learner’s particular needs [6,7]. These tools perform several tasks over the set of available programs, for example, automated analysis and repairing [10]. These tasks can help an instructor to gain insights regarding learners’ strengths and weaknesses; however, these opportunities have not been fully exploited by existing tools.

To address these issues, Marin and Rivero [8] presented a technique to analyze correct programs that pass a set of test cases. The individual statements of these programs are clustered according to their semantic intent, e.g., checking whether an integer i is greater than another integer m (to keep track of the maximum number) can be accomplished in several ways: `if (i > m)`, `if (m < i)`, `if (!(i <= m))`, or `if (a[j] > m)`. These individual statements can be clustered together as “Check whether current number is greater than maximum.” Statement clusters are promising to enable customized, automated feedback delivery [5,8]. The existing technique relies on a structural graph clustering algorithm that imposes strong graph connectivity restrictions to form clusters. When individual program statements do not clearly belong to a cluster, they are categorized as outliers. As a result, this technique discovers a small number of statement clusters in real-world programs that are very cohesive, but misses to classify many other individual statements. In the reported experiments, outliers are between 30% and 70% of the total number of program statements.

In this paper, we identify opportunities to cluster additional statements under the same semantic intents without compromising the cohesiveness of the discovered clusters. We assume that a graph that connects all individual program statements of all the programs that are under evaluation is created [8]. Then, we discard edges in such a graph between individual program statements whose similarities are below a certain threshold and, therefore, are noisy. To discover statement clusters, we process vertices in the graph based on their density, which serves to resolve “clear cuts” first, i.e., clusters of individual program statements that are homogenous in their semantic intent, leaving the difficult cases for latter stages. Finally, we avoid clusters that may contain different program statements belonging to the same program. The assumption is that each individual program statement has a semantic intent, and that semantic intent must be different within a certain program. Taking all of these into account, we propose a new statement clustering algorithm that, according to our experiments, is more efficient than the previous technique, and is able to cluster, in the worst case, more than 93% of the individual program statements.

The rest is as follows: preliminaries (Sect. 2), our proposed algorithm (Sect. 3), experimental results (Sect. 4), and conclusions (Sect. 5).

2 Preliminaries

We wish to analyze programs solving introductory programming assignments. Assume three programs presented in Fig. 1 that solve the following assignment: Read the total number of test cases. Each test case contains the number of cars and a list of space-separated integers, each of which denotes the maximum speed of a car in the order they enter a straight segment. For each test case, output the number of cars which are moving at their maximum speed.

<pre> 1 Scanner sc = new Scanner(System.in); 2 int t = sc.nextInt(); 3 while (t-- > 0) { 4 int n = sc.nextInt(); 5 int[] mx = new int[n]; 6 for (int i = 0; i < n; i++) 7 mx[i] = sc.nextInt(); 8 int x = mx[0]; 9 int s = 1; 10 for (int i = 1; i < n; i++) { 11 x = Math.min(mx[i], x); 12 if (x == mx[i]) 13 s++; 14 } 15 System.out.println(s); 16 }</pre>	<pre> 1 Scanner sc = new Scanner(System.in); 2 int t = sc.nextInt(); 3 while (t-- > 0) { 4 int n = sc.nextInt(); 5 int[] ar = new int[n]; 6 for (int i = 0; i < n; i++) 7 ar[i] = sc.nextInt(); 8 int c = 1; 9 int small = ar[0]; 10 for (int i = 1; i < n; i++) 11 if (ar[i] <= small) { 12 c++; 13 small = ar[i]; 14 } 15 System.out.println(c); 16 }</pre>	<pre> 1 Scanner sc = new Scanner(System.in); 2 int t = sc.nextInt(); 3 while (t-- > 0) { 4 int n = sc.nextInt(); 5 int[] a = new int[n]; 6 for (int i = 0; i < n; i++) 7 a[i] = sc.nextInt(); 8 int count = 1; 9 for (int i = 1; i < n; i++) 10 if (a[i - 1] >= a[i]) 11 count++; 12 else 13 a[i] = a[i - 1]; 14 System.out.println(count); 15 }</pre>
(a) p_1	(b) p_2	(c) p_3

Fig. 1. Three programs solving CARVANS (<https://www.codechef.com/problems/CARVANS>)

First, we model programs as program dependence graphs to be analyzed. A program is represented by a program dependence graph $G = (V, E, L_V, L_E)$ such that V is a set of vertices, each of which is a program statement, $E : V \rightarrow V$ is a bag of directed edges (there can be multiple edges connecting the same vertices), $L_V : V \rightarrow \mathcal{P}(\mathcal{V})$ is a vertex labeling function from each vertex to the power set of possible vertex labels \mathcal{V} , and $L_E : E \rightarrow \{Ctrl, Data\}$ is an edge labeling function that determines whether an edge is control (*Ctrl*) or data (*Data*).

The program dependence graph representing p_1 contains a vertex for each program statement, for instance, a vertex associated to line 7 where a position of a previously declared array is updated with the speed of a car. As a result of these operations, L_V of this specific vertex contains several labels, such as array access, assignment and `nextInt`. All these labels form \mathcal{V} that help identify the semantics of the statements. Additionally, edges between vertices indicate the relationships between the statements in the code. For example, there is a *Ctrl* edge between the statement in line 6 (`for` loop) and the vertex previously discussed. This edge indicates that the statement is executed only if the condition of the loop is true. There is a *Data* edge between the statement that declares variable `i` and the statement that uses `i` to access the array.

A statement cluster C is a set of vertices (statements) that have the same semantic intent but can be implemented in different ways. Programs in Fig. 1

initialize console using **Scanner**, which form a statement cluster. They read the total number of test cases using **nextInt**, which form another statement cluster.

The technique by Marin and Rivero [8] discovers statement clusters using a distance $d(v_i, v_j)$ between vertices v_i and v_j , which considers both $L_V(v_i)$ and $L_V(v_j)$ as well as their context. Having two program dependence graphs $G_i = (V_i, E_i, L_{V_i}, L_{E_i})$ and $G_j = (V_j, E_j, L_{V_j}, L_{E_j})$, it finds a correspondence between their vertices, a.k.a. alignment, $A : V_i \rightarrow V_j$ ($V_i \subseteq V_j$). To find A , a weighted bipartite graph $B = (V_i, V_j, E_B, W_B)$ is used, where $E_B : V_i \rightarrow V_j$ and $W_B : V_i \times V_j \rightarrow \mathcal{R}$ is an edge weight function such that $W_B((v_i, v_j)) = 1 - d(v_i, v_j)$. B is complete: every vertex in V_i is related to every vertex in V_j by an edge in E_B . An alignment A is a maximum weighted matching in B .

The next step consists of finding alignments between all programs under evaluation (for n programs, the total number of alignments is $1/2(n-1)n$). The union of the alignments form the pairwise alignment graph $P = (V_P, E_P, W_P)$, where V_P is the union of all vertices in the program dependence graphs, $E_P : V_P \times V_P$ is the set of edges such that each edge belongs to a specific alignment A (maximum weighted matching), and $W_P : V_i \times V_j \rightarrow \mathcal{R}$ is an edge weight function such that $W_P((v_i, v_j))$ corresponds to $W_B((v_i, v_j))$ in the bipartite graph B from which A is computed. Statement clusters are discovered by exploiting structural graph clustering over P , discerning between statement clusters, hubs and outliers. Hubs and outliers are vertices that are connected to other vertices in different statement clusters, but they do not belong themselves to any cluster. A hub is connected to vertices that belong to more than one statement cluster; an outlier is connected to vertices that belong to the same statement cluster.

In Fig. 1a, the statement in line 11 in p_1 is a hub since it relates statements in the cluster formed by statements checking the current speed (lines 11 and 10 in p_2 and p_3 , respectively), and statements in the cluster formed by statements updating the current minimum speed (lines 13 in both p_2 and p_3).

3 A New Clustering Algorithm

Low weights in alignments introduce noise [8]. These weights are the distance between two statements in an alignment graph. The algorithm to compute maximum weighted matchings focuses on large weights first, i.e., statements that are very related and, therefore, it is desirable to have correspondences between them. Unfortunately, since the algorithm aims to compute a maximum matching, there are certain vertices (the “leftovers”) that are forced to match, even though their weight is low, i.e., they are probably not semantically related. We propose a user-defined threshold δ to avoid low weights in alignments as follows: let v_i and v_j be two vertices, (v_i, v_j) is discarded from an alignment A if $W_B((v_i, v_j)) < \delta$. By introducing δ , we expect to mitigate such noisy correspondences.

The processing order of the vertices may have an impact in the clustering process. Depending on which vertex is selected first for processing, statement clusters may contain a different set of statements. We propose to rely on the

Algorithm 1: Mine statement clusters

Input: $P = (V_P, E_P, W_P)$, δ , β , ι
Output: A statement cluster function $X : V_P \rightarrow \mathbb{N}$

```

1  $E_P := E_P \setminus \{(v_i, v_j) \mid (v_i, v_j) \in E_P \wedge W_P((v_i, v_j)) < \delta\}$ 
2  $clnumber := 0$ 
3 foreach  $v \in V_P$  sorted by core number do
4    $N := \hat{N}(v, P)$ ,  $N' := \emptyset$ 
5   foreach  $v \in N$  do
6      $N' := N' \cup \hat{N}(n, P)$ 
7   if  $overlap(N, N') \geq \beta$  then
8      $X(v) := clnumber$ 
9     foreach  $n \in N \cap N'$  do
10    |  $X(n) := clnumber$ 
11    |  $clnumber := clnumber + 1$ 
12  else
13  |  $X(v) := -1$ 
14 foreach  $i \in ran X$  do
15 |  $V := \{v \mid X(v) = i\}$ 
16 | if  $|V| < \iota$  then
17 | | foreach  $v \in V$  do
18 | | |  $X(v) := -1$ 

```

concept of the core number to determine such processing order. A k -core is a maximal subgraph of a graph in which all vertices have at least k neighbors [2]. The core number of v is the largest k such that v belongs to the k -core but not to the $(k + 1)$ -core. We thus process first vertices that are expected to be dense, i.e., they are semantically cohesive. These vertices should be “clear cuts” and the unraveling of posterior vertices should benefit from these early decisions.

A duplicated statement cluster contains at least two vertices that belong to the same program [8]. Since our goal is to detect statements across programs that have the same semantic intent, duplicated statement clusters are thus harmful. For instance, lines 11 and 12 in p_1 can be part of the same statement cluster. As a result, we avoid duplicated statement clusters by defining a $\hat{N}(v, G)$ function that receives a vertex v and a graph G as input, and outputs all the neighbors of v in G such that every neighbor belongs to a different program than v . Forming clusters based on $\hat{N}(v, G)$ prevents duplicated statement clusters.

Algorithm 1 uses all of these ingredients to discover statement clusters.

4 Experiments

We evaluate our technique over six different introductory programming assignments. Five of them are from CodeChef (BUYING2, CARVANS, CONFLIP, LAPIN and STONES), which were also studied by Marin and Rivero [8]. The sixth assignment corresponds to P327A from Codeforces¹. Table 1 presents

¹ <https://codeforces.com/problemset/status/327/problem/A>.

Table 1. Statement clusters and program coverage obtained for six different introductory programming assignments using $\delta = .5$, $\beta = .8$ and $\iota = .05 |P|$

	$ P $	$ V_P $	$ E_P $	$ C $	$ U $	Cov	μ_V	T (s)
BUYING2	861	24,566	8,350,147	80	2,700	95.10%	273.33 ± 249.05	44
CARVANS	719	17,487	4,855,564	62	2,270	94.32%	245.44 ± 222.60	22
CONFLIP	1,203	26,685	12,155,327	68	3,555	93.77%	340.15 ± 340.27	75
LAPIN	561	18,126	3,856,061	107	1,890	94.77%	151.74 ± 135.09	21
P327A	750	22,384	6,948,266	93	1,116	96.26%	228.69 ± 201.79	34
STONES	152	4,312	252,174	98	405	96.67%	39.87 ± 40.03	1

our results, where $|P|$ represents the total number of correct programs available, $|V_P|$ is the total number of statements in the pairwise alignment graph, $|E_P|$ is the total number of edges that meet the weight threshold criterion ($W_P((v_i, v_j)) < \delta = .5$) in the pairwise alignment graph, $|C|$ is the number of statement clusters discovered that meet both overlap and pervasiveness criteria based on $\beta = .8$, $|U|$ is the number of vertices that are non-clustered, Cov is the mean coverage of the program statements under evaluation, μ_V is the mean (and standard deviation) number of program statements that are contained in each statement cluster, and T is the total time in seconds to discover statement clusters. We set ι to 5% of the total number of programs ($\iota = .05 |P|$). The timings presented in Table 1 were obtained using commodity hardware.

Comparing our results with those obtained by Marin and Rivero [8], we observe that the coverage we obtain with the statement clusters computed by our technique significantly outperforms the previous coverage. For instance, in the LAPIN assignment, the previous coverage was above 30% based on 20 statement clusters. In our experiments, we obtain a coverage of 94% using 107 statement clusters. LAPIN has a fewer number of programs that have more implementation variability than other assignments. This can be determine by measuring the number of statement clusters as well as the average number of program statements per cluster. Because of this variability, the technique by Marin and Rivero [8] marks many program statements as outliers or hubs since there is no enough evidence to include them in a specific cluster. In our technique, these program statements are “forced” to belong to a given cluster, which will result in more diverse program statements clustered together.

5 Conclusions

Introductory programming learners need to receive constant feedback to improve their computational problem solving skills. It is currently a challenge to deliver feedback to the large number of learners in both traditional and online settings. Existing techniques focus on the automated analysis and delivery of feedback, and do not generally support an active role of the instructor neither in the feedback nor in its delivery. A promising direction to enable instructor-on-the-loop feedback delivery is to group program statements into clusters with a similar




semantic intent. A previous technique focused on guaranteeing the semantic cohesiveness of the clusters rather than covering a large number of individual program statements. As a result, many program statements in the long tail are not clustered and, therefore, do not receive feedback. In this paper, we analyze several opportunities to increase the coverage of individual program statements with the goal of delivering feedback to the long tail. Our experiments show that we are able to cover, in the worst case, more than 93% of the program statements available for the assignments under evaluation. This increasing coverage comes with the penalty of less semantically-cohesive statement clusters.

References

1. Camp, T., Zweben, S.H., Buell, D.A., Stout, J.: Booming enrollments: survey data. In: ACM Technical Symposium Computing Science Education, SIGCSE 2016, pp. 398–399 (2016)
2. Cheng, J., Ke, Y., Chu, S., Özsu, M.T.: Efficient core decomposition in massive networks. In: IEEE International Conference on Data Engineering, ICDE 2011, pp. 51–62 (2011)
3. Fitzgerald, S., Hanks, B., Lister, R., McCauley, R., Murphy, L.: What are we thinking when we grade programs? In: ACM Technical Symposium on Computer Science Education, SIGCSE 2013, pp. 471–476 (2013)
4. Huang, J., Piech, C., Nguyen, A., Guibas, L.J.: Syntactic and functional variability of a million code submissions in a machine learning MOOC. In: Workshops at the International Conference on Artificial Intelligence in Education, AIED Workshops 2013 (2013)
5. Jawalkar, M.S., Hosseini, H., Rivero, C.R.: Learning to recognize semantically similar program statements in introductory programming assignments. In: ACM Technical Symposium on Computer Science Education, SIGCSE 2021, p. 1264 (2021)
6. Keuning, H., Jeuring, J., Heeren, B.: A systematic literature review of automated feedback generation for programming exercises. *ACM Trans. Comput. Educ. (TOCE)* **19**(1), 3:1–3:43 (2019)
7. Marin, V.J., Pereira, T., Sridharan, S., Rivero, C.R.: Automated personalized feedback in introductory Java programming MOOCs. In: IEEE International Conference on Data Engineering, ICDE 2017, pp. 1259–1270 (2017)
8. Marin, V.J., Rivero, C.R.: Clustering recurrent and semantically cohesive program statements in introductory programming assignments. In: ACM International Conference on Information and Knowledge Management, CIKM 2019, pp. 911–920 (2019)
9. Singh, R., Gulwani, S., Solar-Lezama, A.: Automated feedback generation for introductory programming assignments. In: ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2013, pp. 15–26 (2013)
10. Wang, K., Singh, R., Su, Z.: Search, align, and repair: data-driven feedback generation for introductory programming exercises. In: ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2018, pp. 481–495 (2018)



Voice Privacy with Smart Digital Assistants in Educational Settings

Mohammad Niknazar^(✉) , Aditya Vempaty^{}, and Ravi Kokku^{}

Merlyn Mind, Inc., New York, NY, USA
{mohammad, aditya, ravi}@merlyn.org

Abstract. The emergence of voice-assistant devices ushers in delightful user experiences not just on the smart home front, but also in diverse educational environments from classrooms to personalized-learning/tutoring. However, the use of voice as an interaction modality could also result in exposure of user's identity, and hinders the broader adoption of voice interfaces; this is especially important in environments where children are present and their voice privacy needs to be protected. To this end, building on state-of-the-art techniques proposed in the literature, we design and evaluate a practical and efficient framework for *voice privacy at the source*. The approach combines speaker identification (SID) and speech conversion methods to randomly disguise the identity of users right on the device that records the speech, while ensuring that the transformed utterances of users can still be successfully transcribed by Automatic Speech Recognition (ASR) solutions. We evaluate the ASR performance of the conversion in terms of word error rate and show the promise of this framework in preserving the content of the input speech.

Keywords: Privacy · Voice-enabled device · Speech conversion · De-identification

1 Introduction

There has been an explosion in voice-assistant device market over the past few years, especially due to the broader movement towards voice-enabling IoT devices in homes and enterprises [8]. Although voice-assistant devices in education are not meant to replace the meaningful and necessary human interaction between teachers and students, they can help teachers do many things more efficiently and excitingly. An example of this application would be a digital assistant that sits in the classroom to support the teacher (and sometimes students) in their everyday tasks such as providing access to vetted content including text, image, video, etc., or browsing web pages through voice.

However, the use of voice also raises practical privacy concerns [6, 7, 10]. These concerns are more serious in environments such as schools where children are present and they may also interact with the device. For instance, the speech files can be *processed* for profiling after appropriate speaker identification (SID).

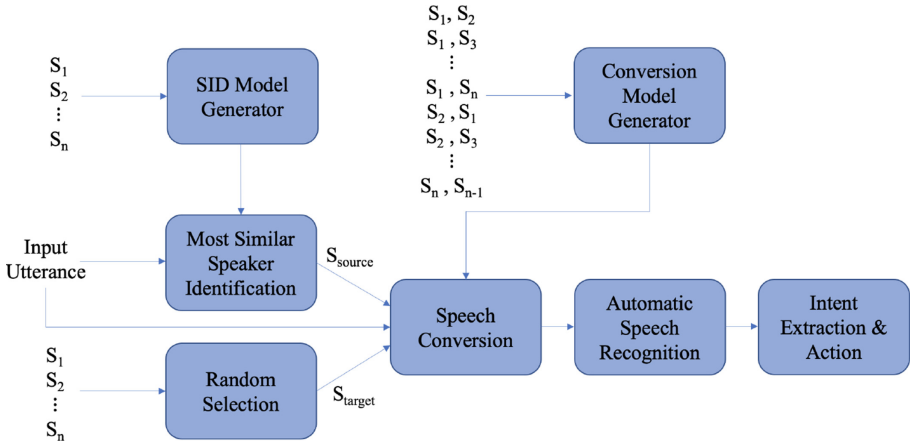


Fig. 1. Block diagram of the proposed framework.

Although recent privacy laws (including COPPA and GDPR) have forced major technology companies to establish privacy policies [12], a typical user is still unaware of and lacks enough forethought on ways in which the speech data could be used then or in the future (including sharing with third-parties). We believe that privacy should not be an after-thought, but should be provided by design [13], and as close to the source of recording as possible. To address this problem as well as several drawbacks of the previous related works, we develop an efficient framework for voice privacy that is practical to implement in real-time.

2 Methodology

In a preliminary work [9], some early observations were reported for voice privacy protection based on cycle-consistent adversarial networks voice conversion (CycleGAN-VC) [4], which can be used to convert speech utterances. CycleGAN-VC2 [5] is an improved version of CycleGAN-VC that outperforms CycleGAN-VC in terms of naturalness and similarity for different speaker pairs, including intra-gender and inter-gender pairs [5]. We used CycleGAN-VC2 based on the implementation in [1] to create multiple source-target conversion models.

The steps of the proposed framework are as follows (refer to Fig. 1). First a training dataset with n number of speakers is selected. This dataset is used to create n SID models, one for each speaker, as well as $n(n-1)$ conversion models based on CycleGAN-VC2, two for each pair of the speakers (one per direction). For SID, we adopted a typical SID method that uses mel-frequency cepstral coefficient (MFCC) and linear predictive coding (LPC) features along with GMM and universal background model (UBM) based on the implementation in [2] to model each speaker. As Fig. 1 shows, the input utterance is first mapped to the closest speaker using the SID models trained for the speakers in the dataset to ensure a high-quality conversion. Then, the target is chosen at random

Table 1. WER values for different conversion combinations. The first row corresponds to the original utterances with no conversion. The median value for all cases was 0.

Source	Target	p75 Truncated mean	Mean	STD
–	–	0	0.019	0.107
All	All	0.028	0.219	0.363
All	Male	0.006	0.171	0.332
All	Female	0.044	0.256	0.389
Male	Male	0.010	0.186	0.342
Female	Female	0.016	0.193	0.346
Male	Female	0.158	0.369	0.436
Female	Male	0.032	0.235	0.378

from the remaining speakers in the training dataset. Since part of the model selection process is random, it ensures there is no reversibility in identifying the speaker of any given utterance. Finally, the input utterance is converted using the pre-generated conversion models and the output is sent to the ASR and intent extraction modules to take an action.

3 Results

In order to assess the performance of the proposed framework, 880 utterances were recorded from 1 woman and 4 men (176 each), where 3 men were native English speakers with American accent. The recorded utterances were typical examples of commands given to voice assistant systems in an educational setting such as “*go back to the beginning of a video*”, “*mute*”, “*speak louder*”, “*yes*”, etc. For the training dataset, VCC2016 [11], which has high-quality utterances from 5 men and 5 women ($n=10$) was used. Different combinations of source-target subsets were evaluated to identify the subset with the lowest Word Error Rate (WER). Table 1 reports the WER statistics for different subsets on the outputs obtained from a major technology vendor’s cloud ASR platform [3] when an appropriate list of commands, which goes beyond the commands used in the dataset was provided as the input context to the ASR. The p75 results for the WER show success of the conversion in most cases. As seen, the lowest WER was obtained when both male and female speakers were used as candidates to map the source, and only male speakers were used as the random target. This phenomenon may be caused by the inherent characteristics of the CycleGAN-VC2 or possible unbalanced (male dominated) data used to train the ASR engines.

Different approaches may be adopted to evaluate the de-identification performance of the proposed method. To simplify things in favor of a cloud SID system aiming at identifying speakers, we assumed that the system already has some training data on original voices of each speaker with labels and generated SID models based on them (same SID approach as discussed in Sect. 2). Then, we

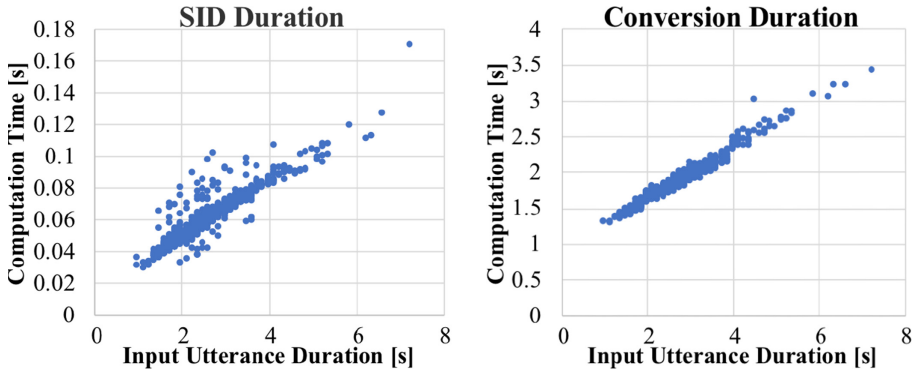


Fig. 2. Computational time for speaker identification (SID) and voice conversion.

compared the performance of SID with unconverted and converted test data. In order to do so, we randomly divided the data of 880 utterances to train and test sets with 70% and 30% portions, respectively. The SID on the original unconverted test data was 100% accurate with all utterances being perfectly classified. However, when the converted version of the same utterances were tested with the SID, the accuracy was only 20.45% with 210 out of 264 utterances misclassified (essentially random selection of speaker). This shows the effectiveness of the proposed framework in disguising the identity of the speakers.

Finally, the inference computation time of the proposed framework was calculated to determine the feasibility of the method in real-time applications. Figure 2 shows the computation time for speaker identification and voice conversion on a machine with an Nvidia GPU (1080 GTX), with less than 10% GPU utilization. As seen, speaker identification is performed very fast and remains under 0.2 s even for an utterance of 7 s. Voice conversion takes longer and a delay of 0.5 s is observed for short utterances of 1 s, which can be negligible in most applications. Nevertheless, as the duration of utterances increases, the delays become smaller and for any utterances of 2 s and longer, the conversion can be run real-time.

4 Conclusion

In this paper, we proposed a practical framework for voice privacy protection, while preserving the content of an utterance, using a combination of speaker identification and speech conversion models. In educational settings, due to laws and regulations, solutions may need extra level of privacy before gaining adoption. We believe that if smart assistants provide such voice privacy by design (at the source), the adoption of voice interfaces would accelerate in both one-one personal tutoring and public spaces such as schools and lead to ubiquity of voice as an interaction modality sooner.

References

1. GAN-voice-conversion. <https://github.com/njellinas/GAN-Voice-Conversion/>
2. Speaker-recognition. <https://github.com/ppwwyyxx/speaker-recognition/>
3. Google: Cloud speech-to-text. <https://cloud.google.com/speech-to-text/>
4. Kaneko, T., Kameoka, H.: Parallel-data-free voice conversion using cycle-consistent adversarial networks. arXiv preprint [arXiv:1711.11293](https://arxiv.org/abs/1711.11293) (2017)
5. Kaneko, T., Kameoka, H., Tanaka, K., Hojo, N.: Cyclegan-VC2: improved CycleGAN-based non-parallel voice conversion. In: Proceedings of IEEE International Conference on Acoustics, Speech, Signal and Processing (ICASSP 2019) (2019)
6. Kelly, G.: Compare the privacy practices of the most popular smart speakers with virtual assistants. <https://www.common sense.org/education/articles/compare-the-privacy-practices-of-the-most-popular-smart-speakers-with-virtual-assistants>
7. Liao, Y., Vitak, J., Kumar, P., Zimmer, M., Kritikos, K.: Understanding the role of privacy and trust in intelligent personal assistant adoption. In: iConference (2019)
8. Medeiros, J.: The most exciting voice gadgets from CES 2019. <https://www.voicesummit.ai/blog/the-most-exciting-voice-gadgets-from-ces-2019> (2019)
9. Niknazar, M., Vempaty, A., Haley, P.: A privacy solution for voice enabled devices connected to the internet (extended abstract). In: Proc. IEEE Global Conf. Signal and Inf. Proc. (GlobalSIP) (2019)
10. TechCrunch: 41% of voice assistant users have concerns about trust and privacy
11. Toda, T., et al.: The voice conversion challenge 2016. In: Interspeech, pp. 1632–1636 (2016)
12. Wiggers, K.: How Amazon, Apple, Google, Microsoft, and Samsung treat your voice data. <https://venturebeat.com/2019/04/15/how-amazon-apple-google-microsoft-and-samsung-treat-your-voice-data/> (2019)
13. Wikipedia: Virtual assistant privacy. https://en.m.wikipedia.org/wiki/Virtual_assistant_privacy



Selfit – An Intelligent Tutoring System for Psychomotor Development

Laurentiu-Marian Neagu^{1,2}, Eric Rigaud¹, Vincent Guarnieri¹, Sébastien Travadel¹,
and Mihai Dascalu²(✉)

¹ Centre of Research on Risks and Crisis Management, MINES ParisTech, PSL University,
1 Rue Claude Daunesse, Sophia Antipolis, France
{eric.rigaud, vincent.guarnieri,
sebastien.travadel}@mines-paristech.fr

² Computer Science Department, University Politehnica of Bucharest, 313 Splaiul
Independentei, 060042 Bucharest, Romania
{laurentiu.neagu, mihai.dascalu}@upb.ro

Abstract. Recent advancements in Machine Learning and software development are extending the applications of Intelligent Tutoring Systems (ITS) into non-cognitive skill domains, while proposing novel design architectures and tutoring strategies. In this paper we present our ongoing work on *Selfit*, an Intelligent Tutoring System for psychomotor development. The system focuses on and was tested for Anatomical Adaptation training, the first phase of training. The tutoring module includes a contextual multi-armed bandit algorithm for online generation of teaching sequences to overcome multiple problems, such as lack of training time, complexity of user characteristics, or management of motivation. First, the system was evaluated in a virtual environment, where populations of trainees follow several personalization strategies in systematic experiments. Second, *Selfit* is currently being tested by a group of users and a preliminary study revealed that most trainees find the system easy to use, modern, and attractive.

Keywords: Intelligent Tutoring System · Psychomotor development · Contextual Multi-Armed Bandits · Personalization

1 Introduction

Psychomotor development is a lifelong process of learning how to move accordingly to a dynamic environment. Essential movements, such as pushing, pulling, or core, are prerequisites for learning specialized, complex psychomotor tasks required by daily life, or leisure activities. Psychomotor development usually starts with the definition of movement competence and the initial evaluation of trainees, whereas the development of physical qualities requires following the super-compensation cycle [1].

A thorough analysis of relevant works published in the ITS community related to psychomotor skills was conducted by Neagu et al. [2]. The study presented 7 relevant papers, mapping several psychomotor domains, such as: acquiring driving skills, military

(U.S. Army – GIFT [3]), training for laparoscopic surgeries (robotic-assisted), postural retraining in health, improving motor learning (TIKL [4]), or ball-passing training.

Following the previously mentioned literature review, the aim of this research is to introduce and perform an initial evaluation of an Intelligent Tutoring System designed for psychomotor skills – *Selfit* – which is focused on amateurs performing sport for general health. The main challenge for our ITS is to identify the optimal sequence that maximizes the average competence level, across all targeted skills [5]. This challenge is driven by three main factors, which were first tackled by Clement et al. [5] when addressing learning sequence personalization for mathematics: a) limited time for practicing activities; b) managing motivation is hard, and c) individual differences between trainees. The results obtained by Clement et al. [5] using multi-armed bandit algorithms are comparable and even surpass, in certain conditions, the sequences created by expert teachers; a similar approach was used in *Selfit*.

2 Intelligent Tutoring Systems for Psychomotor Development

Selfit aims to support students in performing fundamental and specialized movement tasks correctly and safely, by generating learning sessions adapted to their responses to physical stimuli. First, the system defines the general learning objectives by interacting with trainees and deducing the list of movement skills to be acquired. An initial student profile is calibrated through a set of representative sports tasks, called challenges, which evaluate the trainee’s readiness to perform movement skills.

Four conceptual components, or modules, interact with each another: a) Graphical User Interface (including Authentication, Calibration, Dialogue and Training session), b) Domain model, c) Student model (including Monitoring), and d) Tutoring model. Feedback is crucial to perform motor skills well [6]. Before starting a session, *Selfit* asks trainees to self-evaluate their fatigue level, motivation to train, sleep quality, and stress level. During training, *Selfit* asks trainees to self-evaluate at the end of each exercise. After the training session, the system assesses the session difficulty.

The *Selfit Domain model* consists of an ontology [7] whose core consists of the movement skill class, with associated psychomotor profile, movement patterns, and training program modalities [8]. The ontology describes the relationships between body, muscle chains, joints movements, agonist, antagonist, and synergist muscles for strength qualities development. The *Selfit Student model* contains information about the trainee’s psychomotor capacities, especially the ones related to the supercompensation cycle status, as well as usage statistics. The *Monitoring* module accesses information on how trainees are using the system or how they are progressing with their training. The *Selfit Tutoring model* supports the learning process by providing machine learning mechanisms to support the adaptation of the learning program to the trainee’s characteristics. Current work focuses on the first level of adaptation – a *Novice Trainer*. The underlying model relies on templates of training sequences for generating micro-cycles and sessions based on trainee input. A sub-list of templates used for generating micro-cycles in anatomical adaptation are the presented in Table 1.

The *Novice trainer* has to choose the right exercise from the list of available exercises. An efficient online method, namely contextual Multi-Armed Bandits [9] was used to

Table 1. Micro-cycle templates examples for anatomical adaptation.

Micro-cycle template name	# of trainings	Recommended trainee
Push/Pull/Lower/Upper/Lower	5	Men
Hip Dominant/Knee Dominant/Upper/Lower/Upper	5	Women
Upper/Lower/Full/Full/Full	5	Mixed

explore and optimize different exercises and estimate trainee progress. The context of *Selfit* is the trainee shape-of-the-day which is computed using a Borg scale [10], while a Category-Ratio Scale is used to measure different body shape parameters. The reward of Multi-Armed Bandit is represented as the number of Repetitions in Reserve (RiR) [11]. RiR denotes how many more repetitions a trainee could have performed at the end of a set; 0 marks reaching maximum repetitions.

3 Results

3.1 Simulation with Virtual Trainees

The efficiency of the Tutoring module was simulated with different contextual multi-armed bandits' implementations. The experiment was conducted in an environment configured with 1800 exercises, 300 for each movement family, with 10 per each level of difficulty. The following setup was considered for a person: 400 training sessions, 128 exercises per month, with 4 sessions per micro cycle, each of them with 8 exercises.

Four agents with different strategies, were trained: a) random agent, b) multi-armed bandit upper confidence bound (MaB UCB1), c) multi-armed bandit ϵ -Greedy (0.1), and d) Bayesian MaB UCB1. Initial competence levels were configured randomly for each movement type of the simulated trainees. In the experiment, a population of 1000 trainees was generated, each with a specific competence level. Figure 1 introduces the simulation results in which a datapoint on the Ox axis encapsulates the cumulative reward of 10 training sessions, while the Oy axis uses a squared root scale. The algorithm that provides the best cumulative reward during training is the Bayesian Multi-Armed Bandits UCB1: 1175; next was the ϵ -Greedy strategy (975.8), followed by simple MaB UCB (483.3), and random (33.1).

3.2 Preliminary User Testing

Selfit is currently being tested with real users to evaluate whether it fulfills its requirements (quality test) and observe if trainees use it effectively, efficiently, and are satisfied (usability test). This phase is a long-term process that will last at least 12 consecutive weeks. A group of 18 trainees, from France and Romania, both novice and experimented in psychomotor training, started to use *Selfit*. Each trainee was assigned randomly at registration into one of the training strategies: random, MAB UCB1, ϵ -Greedy (0.1), and Bayesian MAB UCB1.

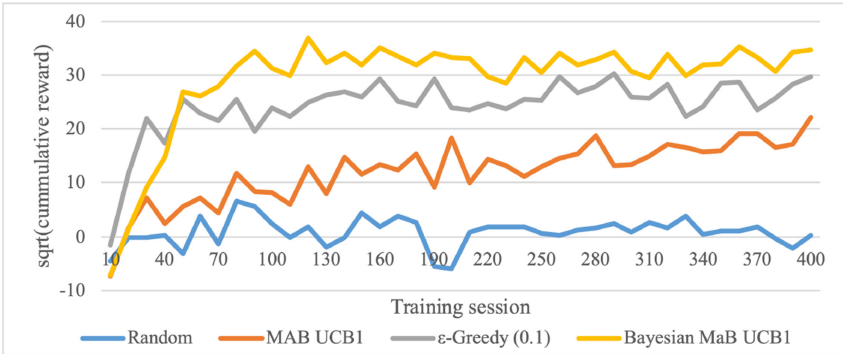


Fig. 1. Training algorithms comparison – 2 years’ timeframe.

A user experience survey was conducted during the testing phase with 18 trainees. The AttrakDiff questionnaire [12] on a 7-point Likert scale was used to assess their perceptions (see Table 2 for summative results). *Selfit* is generally perceived as practical, predictable, simple, connective, and human-oriented when considering its pragmatic qualities. In terms of hedonist qualities, the system is perceived as stylish, motivating, novel, and captivating.

Table 2. *Selfit* user experience feedback based on AttrakDiff questionnaire (Mean and Standard Deviation values corresponding to qualities scored on a 7-point Likert scale).

UX quality	M (SD)	UX quality	M (SD)	UX quality	M (SD)
Pleasant	5.44 (1.11)	Connective	4.61 (1.53)	Human	4.55 (1.25)
Inventive	4.94 (1.80)	Simple	4.50 (1.42)	Professional	4.83 (1.50)
Attractive	5.05 (1.22)	Practical	5.50 (0.89)	Likeable	5.83 (0.95)
Straightforward	5.05 (1.17)	Stylish	5.00 (1.20)	Predictable	4.27 (1.19)
Premium	4.66 (1.29)	Integrating	5.72 (0.80)	Brings people closer	4.72 (1.32)
Novel	5.22 (1.35)	Motivating	5.44 (0.95)	Captivating	5.44 (0.89)

4 Conclusions

Our current work introduces a new Intelligent Tutoring System used for Anatomical Adaptation Psychomotor training – *Selfit* – that integrates ontologies for knowledge representation, modern libraries for user interface design, and recent advancements in teaching strategies for personalizing training content.

Several experiments were conducted in a fully simulated environments, where populations of trainees were generated, with corresponding exercises of different characteristics. Four teaching strategies were tested, from which Bayesian MAB UCB1 exhibited

the best results for a scenario close to a real-world use case. In parallel, a testing phase with real users was started; this phase normally lasts for at least 12 weeks (preferably 6, 12 months for long-term effects). The preliminary user experience survey showed promising results and highlighted the usefulness of the built system. Next, more insights on the efficiency of the personalization training algorithms will be studied, more trainees will be involved in the process, and future developments will be decided after the testing phase.



References

1. Bompa, T., Buzzichelli, C.: *Periodization: Theory and Methodology of Training*, 6th edn. Human Kinetics Publishers (2017)
2. Neagu, L.-M., Rigaud, E., Travadel, S., Dascalu, M., Rughinis, R.-V.: Intelligent tutoring systems for psychomotor training – a systematic literature review. In: Kumar, V., Troussas, C. (eds.) *ITS 2020. LNCS*, vol. 12149, pp. 335–341. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_40
3. Goldberg, B., Amburn, C., Ragusa, C., Chen, D.-W.: Modeling expert behavior in support of an adaptive psychomotor training environment: a marksmanship use case. *Int. J. Artif. Intell. Educ.* **28**(2), 194–224 (2018)
4. Lieberman, J., Breazeal, C.: TIKL: development of a wearable vibrotactile feedback suit for improved human motor learning. *IEEE Trans. Rob.* **23**(5), 919–926 (2007)
5. Clement, B., Roy, D., Oudeyer, P.-Y., Lopes, M.: Multi-armed bandits for intelligent tutoring systems. *J. Educ. Data Mining (JEDM)* **7**, 20–48 (2015)
6. Bilodeau, E.A., Bilodeau, I.M.: Motor-skills learning. *Annu. Rev. Psychol.* **12**(1), 243–280 (1961)
7. Noy, N.F., McGuinness, D.L.: *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory, Stanford (2001)
8. Neagu, L.-M., Guarnieri, V., Rigaud, E., Travadel, S., Dascalu, M., Rughinis, R.-V.: An ontology for motor skill acquisition designed for GIFT. In: *Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8)* (2020)
9. Lu, T., Pal, D., Pal, M.: Contextual multi-armed bandits. In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, vol. 9, pp. 485–492 (2010)
10. Spielholz, P.: Calibrating Borg scale ratings of hand force exertion. *Appl. Ergon.* **37**, 615–618 (2006)
11. Hackett, D.A., Johnson, N.A., Halaki, M., Chow, C.-M.: A novel scale to assess resistance-exercise effort. *J. Sports Sci.* **30**(13), 1405–1413 (2012)
12. Hassenzahl, M., Burmester, M., Koller, F.: AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In: Ziegler, J., Szwillus, G. (eds.) *Mensch & Computer 2003*, pp. 187–196. *Interaktion in Bewegung* (2003)

Assessment



Automated Assessment of Learning Objectives in Programming Assignments

Arthur Rump , Ansgar Fehnker ^(✉) , and Angelika Mader 

University of Twente, Enschede, The Netherlands
hello@arthurrump.com, {ansgar.fehnker,a.h.mader}@utwente.nl

Abstract. Individual feedback is a core ingredient of a personalised learning path. However, it also is time-intensive and, as a teaching form, it is not easily scalable. In order to make individual feedback realisable for larger groups of students, we develop tool support for teaching assistants to use in the process of giving feedback. In this paper, we introduce Apollo, a tool that automatically analyses code uploaded by students with respect to their progression towards the learning objectives of the course. First, typical learning objectives in Computer Science courses are analysed on their suitability for automated assessment. A set of learning objectives is analysed further to get an understanding of what achievement of these objectives looks like in code. Finally, this is implemented in Apollo, a tool that assesses the achievement of learning objectives in Processing projects. Early results suggest an agreement in assessment between Apollo and teaching assistants.

Keywords: Programming education · Automated assessment · Automated feedback

1 Introduction

Learning in the perspective of the 21st-century skills aims, among others, at enthusiasm, deep understanding, the ability to apply, and reflection. For the programming courses of our program of Creative Technology, we address these skills by giving open assignments that allow for individual solutions and creativity, while making students owner of their learning process. The programming assignments let students define their own project, as long as they use the concepts taught in the course and demonstrate mastery of the learning outcomes.

A driving principle in this personalised learning process is individual feedback to get students unstuck in their learning path when needed. However, individual feedback is time-intensive, and, accordingly, does not scale well with an increasing number of students. In order to make individual learning processes also realisable for larger groups of students, we develop tools that support teaching assistants in giving feedback.

We developed an online platform called Atelier to aid communication between students and teaching assistants during programming tutorials. Students can

upload their code and share it with teaching assistants. Teaching assistants can leave comments and see each others' comments, which contributes to consistency in giving feedback. Additionally, an automated code checker [6] identifies standard faults, creating suggestions for comments that the teaching assistant can share and discuss with students. This increases the efficiency of giving feedback.

This paper introduces the extension *Apollo*¹, which analyses programs submitted to Atelier with respect to the desired learning outcomes for the course. The results of Apollo help teaching assistants to identify shortcomings of a program faster, and hence also contributes to an increased efficiency and consistency in giving feedback. Atelier and Apollo were developed in the context of the engineering and design bachelor programme Creative Technology at the University for Twente, where students start programming in Processing².

Section 2 will cover related work on learning outcomes and automated feedback. In Sect. 3 we investigate learning outcomes in programming courses and how mastery of these learning outcomes can be identified by an automated tool is described in Sect. 4. Section 5 uses historical data to calibrate the system and provides early validation of the approach when used in an actual course. The final section closes with discussion and conclusions.

2 Background

This section introduces learning outcomes, including characteristics related to their suitability for automated assessment, and approaches to automated feedback. The combination of both forms the basis for Apollo.

Learning Outcomes. Learning outcomes range from vague aspirations to achieve at the end of a program to very specific objectives to accomplish in a single lecture. Wilson [16] splits learning outcomes into aims, goals and objectives.

- *Aims* give a general direction and are not directly measurable. They are meant to guide an entire program or subject area. An example from our programme is “Graduates understand and can use technology in the domain of software, algorithms and physical interaction.”
- *Goals* are more specific than aims in terms of scope, but they can still relate to an entire program or subject area. They can be formulated as a concrete action, but do not have to be. An example from our program is “Students can create algorithms for solving simple problems.”
- *Objectives* are often written in behavioural terms to describe more specific learning outcomes. They should be observable and measurable, like “Students can implement a divide-and-conquer algorithm for solving a problem.”

We use the term ‘learning outcome’ to mean the intended learning outcome of a course, which may or may not be the actual learning outcome for a student.

¹ Available via <https://github.com/creativeprogrammingatelier/apollo>.

² See <https://processing.org>.

Learning outcomes can also be categorised according to “levels of mastery”, similar to Bloom’s Taxonomy [3,10]. The Computer Science curriculum guidelines by ACM and IEEE [1] use three levels:

- *Familiarity* means the student knows a concept, or the meaning of a concept, but is not able to apply it.
- *Usage* means whether the student can concretely use a concept, for example in a program or when doing analysis.
- *Assessment* means that the student can argue for the selection of a concept to use when solving a problem. This also requires the student to understand available alternatives.

When looking at programming assignments, the related learning outcomes are usually at the ‘usage’ or ‘assessment’ level.

Automated Feedback. A common approach that has been used since the 1960s is the use of automated testing tools to check student submissions for correctness. Douce, Livingstone and Orwell [5] describe several generations of these tools, which all have in common that they require well-defined exercises with supplied test cases to function correctly. In our context where students choose their own projects to work on, these tools are not applicable.

Keuning, Jeuring and Heeren [9] reviewed 101 tools, and found a total of 8 approaches to provide feedback to students, of which automated testing is just one. Four of the other approaches are specific to programming: external tools (such as compilers), static analysis, program transformations, and intention-based diagnosis.

This last approach tries to uncover the student’s strategy by matching code with known ways and code patterns to achieve (sub)goals, following a structured approach to programming [15]. A tool that uses this strategy is PROUST [7,8], which was developed to provide feedback on free-form programming assignments, based on goals the students learned approaches for. This suggests that a similar approach would work well in our context of creative assignments.

An example of a tool that tracks the progress of students is the ACT Programming Tutor (APT) [4], which has a Skill Meter that shows the probability that the student has mastered that skill. APT is also goal-oriented, but works with production rules based on the current assignment: it will not let students take steps that are not on a known path to a correct solution. While the Skill Meter serves a goal similar to what Apollo aims to achieve, the APT approach is not applicable in our context: all assignments need predefined solutions, whereas our course has requirements that allow for a variety of individual solutions.

3 Learning Outcomes

The goal of this section is to identify the learning outcomes that are *assessable* by an automated tool. First, those common in computer science courses are investigated, then those specific to our course.

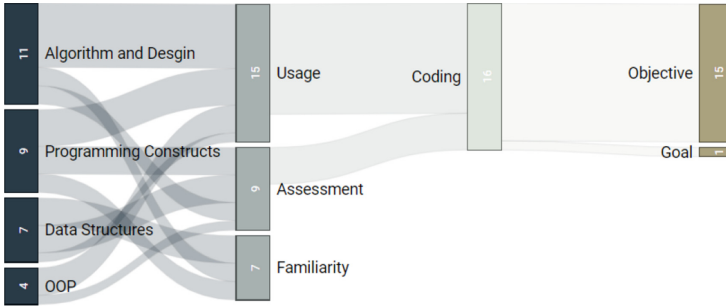


Fig. 1. Learning outcomes from the knowledge areas Software Development Fundamentals (Algorithms and Design, Programming Concepts, and Data Structures) and Programming Languages (OOP) in the ACM and IEEE curriculum guidelines.

Figure 1 illustrates the selection criteria applied to 31 learning outcomes related to programming in the ACM and IEEE Computer Science curriculum guidelines [1], from four topic areas relevant to our course. First, they are distinguished by their level. Only the ‘usage’ and ‘assessment’ level are relevant for our purpose since ‘familiarity’ precludes the ability to use a concept by definition. Next, the learning outcomes need to be directly related to writing code. And finally, the learning outcomes should be concrete and observable, which means that only *objectives* are assessable. Considering these criteria, 15 out of 31 analysed learning outcomes were found suitable for automated assessment.

The focus of our course lies on using programming as a tool for expressing creative ideas [11]. This involves, for example, simulating basic but specific physical systems, rather than learning algorithms in isolation, as would be common in classic computer science programming courses. Consequently, the official learning outcomes have a clear focus on the ‘usage’ level. The specific learning objectives, however, are similar to other introductory programming courses: students have to learn how to write for-loops, they have to understand how variables work and apply basic object-oriented programming concepts.

Based on the curriculum guidelines [1], the actual learning outcomes of our course and the textbook [14], we defined five representative learning objectives:

1. Write a program that uses graphical commands to draw to the screen.
2. Write a program that uses looping constructs for repetition, using the appropriate looping construct.
3. Compose a program using classes, objects and methods to structure the code in an object-oriented way.
4. Implement message passing to enable communication between classes in a complex program, instead of using global variables.
5. Use elementary vector operations to simulate physical forces on an object.

These objectives are selected to give a fair representation of different types of learning objectives and different degrees of freedom in the resulting code.

The five topics are a selection of what students are expected at the end of the course, which also concludes the first year of the Creative Technology programme.

4 Recognising Learning Objectives

Given the learning outcomes suitable for automated analysis, as identified in the previous section, this section focuses on the specific analysis techniques. We first describe the general technology used to detect evidence, and then for each of the five learning objectives what will count as evidence of mastery.

Finding evidence means to recognise relevant usage of programming constructs and programming patterns. Previous research on static analysis of Processing projects [2,6] successfully adapted PMD³. It is also used for Apollo.

For most rules, Apollo uses the following function to translate the count of occurrences to a probability of showing convincing evidence:

$$S_{a,b}(n) = 1 - \frac{1}{\frac{1}{a}n^b + 1} \quad (1)$$

This function defines a family of “S”-shaped curves in the range $[0,1)$, where the slope is determined by the parameters a and b , which can be varied from objective to objective, and from course to course.

The remainder of this section describes which aspects are relevant for the different learning objectives, and how occurrences of relevant structures in the code are counted. Unless mentioned otherwise, the S function is used to translate this count into a probability. Its parameters will be determined in Sect. 5.

1. *Write a program that uses graphical commands to draw to the screen.*

This seems like a relatively simple goal, but full mastery also means that students know different methods, can familiarise themselves with methods that were not explicitly covered in the course and are able to use advanced concepts, such as affine transformations.

Apollo uses a list of all graphical commands in Processing [12], grouped by category. For a given program it creates a list of graphical commands that have been used. The different metrics are then calculated as follows:

- (a) *Use of a variety of different drawing methods covered in the course.*
Count the number of method calls from categories covered in the course.
- (b) *Use of advanced drawing methods, like those in the transform category.*
Count the method calls in the transform category of drawing methods.
- (c) *Use of methods that are not explicitly part of the course material.*

These are the methods that were not part of the count for the first metric. The probability for the entire learning objective is a weighted average of the individual probabilities. The first aspect has weight 3, the second weight 2 and the last weight 1. Students who do not use the covered drawing functions frequently are unlikely to master the drawing commands, even if they do use advanced or non-covered methods, so the first aspect should weigh most.

³ PMD is a tool for detecting code smells in Java, available at <https://pmd.github.io>.

```

1 for(int i = 0; i < ts.length; i++){
2   Thing t = ts[i]; // Get an element
3   s += i * t.getValue(); // Use index directly
4 }

```

Fig. 2. Example of an array iteration that requires the use of an index.

2. *Write a program that uses looping constructs for repetition, using the appropriate looping construct.*

This objective has two aspects: usage of loops, and choosing the appropriate type of loop for a goal. In the context of our course we distinguish the following goals with the related code patterns:

- Repeating some code while a condition holds, using a while-loop.
- Repeating a task n times while increasing a counter, using a while- or for-loop. For-loops are preferred in this case.
- Iterating over all items in an array, using a for-, while-, or foreach-loop; the latter is the preferred option.
- Iterate over all items in an array while using the index independently. In this case, a foreach-loop can not be used, and the for-loop is preferred. See Fig. 2 for an example.

Instead of listing and matching on every possible coding pattern related to using loops, Apollo characterises loops based on their usage. This includes the type of looping condition used, the types of variables used in the body and how the iterator variable is used. Based on this, three metrics are calculated as follows:

- (a) *Use of different types of loops.* The number of different types (either for, while, or foreach) of loops used in the program.
- (b) *Use of loops in a variety of situations, e.g. to iterate over an array, but also for simple repetitions.* Count the number of occurrences of loops with different characterisations.
- (c) *Choose the appropriate looping construct for a given task.* This is calculated as the ratio of loops that are the most appropriate in that situation over all used loops. This value already expresses the chance that a correct looping construct is chosen, and does not need to be converted.

The probabilities resulting from these metrics are averaged to calculate the probability that the program contains convincing evidence that the student achieved this learning objective.

3. *Compose a program using classes, objects and methods to structure the code in an object-oriented way.*

This goal is not just about using the keyword class, but about structuring the program by grouping related data and actions together. The paper [6] proposes an object-oriented structure for interactive Processing applications and also provides static analysis rules for automated detection of so-called design smells in this structure. If a program is structured into multiple classes and none of the common design smells is detected, the student has likely

mastered this objective. Also, it is assumed that any useful class has methods; Apollo uses a minimum of 2.

The metrics are calculated as follows:

- (a) *Use of classes with various methods.* This is simply the number of classes with more than two methods.
- (b) *Relatively few detected design smells in the code patterns.* For this metric, Apollo runs the static analysis rules defined in [6] to detect code smells in the program. Because the chance of a small mistake is bigger in a large program, the amount of smells is divided by the number of classes counted for the first metric. In this case, since a larger number means more problems, and less evidence of mastery, it uses a flipped version of function S .

The probability for the overall objective is the average of these two metrics.

4. *Implement message passing to enable communication between classes in a complex program.*

This objective is related to the previous objective of object-oriented design but is included separately since it receives dedicated attention in the course materials. The goal is to share information between classes. A common, but discouraged, practice is to define a global variable that several objects use. The alternative is method parameter passing, where one object calls a method of another object and passes the information by parameter. The second option is preferred because the information is not shared with other parts of the program that might inadvertently change its value.

To detect if the student can apply message passing, Apollo first finds all global variables. Only variables that are mutated in the program should count for message passing, so global constants are filtered out. Then the number of uses of these global variables across different classes is counted.

The detection of parameter passing happens similarly: Apollo determines all calls to methods from outside the defining class and counts the passed arguments. To get a similar count to the global variable use, only arguments that are locally declared values are counted.

The probability that the program contains convincing evidence that the learning objective is achieved, is then calculated as the ratio of parameter passing instances over the total count of both communication methods.

5. *Use elementary vector operations to simulate physical forces on an object.*

Forces, acceleration, velocity and position are modelled in Processing using the `PVector` class, so physical formulas are commonly translated into operations on these vectors. It is possible to define code patterns for common goals, such as “apply a force to an object with mass”, “calculate the drag force for an object in a medium”, “model gravity using constant downward force” or “calculate friction”. If one or more of these patterns related to known goals are found, it is a good indicator for mastery of this learning objective.

There are, however, many more physical phenomena than specifically covered in the course and known to Apollo, especially since students are free to define their own project. As a weak, but more general, indicator Apollo counts all operations on `PVector` objects. While this could also indicate abstract linear

algebra, an absence of `PVector` usage does indicate that the student is not using `PVectors` for physical modelling.

To keep things manageable, Apollo only considers operations on `PVectors` when comparing code patterns related to physics and simply ignores the calculation of other parts of the solution. Apollo recognises five code patterns for simulating physics, which are defined as a list of method calls on different `PVector` instances.

Two metrics are computed as follows:

- (a) *Use of a known pattern for working with forces.* Count the number of times one of the five specified code patterns for physics matches with the code, as explained before.
- (b) *Use of operations on `PVectors`.* Simply count the number of operations based on the declared `PVectors`.

The overall probability is calculated as the weighted average of both metrics, where the first metric has weight 1, the second 2.5. This means that even when no code patterns are recognised, a chance of 0.7 can still be reached based on the number of `PVector` operations, which is desired because of the mentioned limitations on detection of known physical structures.

5 Calibration and Validation

For an initial validation and determining the parameters for converting counted metrics into probabilities, Apollo was used on an old dataset of final student projects used in the evaluation of [6].

Calibration. To illustrate the method used for calibration of the parameters, we discuss the example of counting calls to drawing methods covered in the course materials. Other metrics were calibrated in a similar fashion.

Considering the drawing methods, we find that all programs in the dataset use between 4 and 17 different drawing methods, with a median of 9. The first quartile is at 8 drawing methods, the third quartile at 11. To determine the correct parameters for the function S , these statistics were mapped to the desired chance. The first quartile is mapped to 50%, the median to 70% and the third quartile to 95%. This means that a chance of understanding above 95% is assigned to the 25% best programs, above 70% to the 50% best programs etc. For the covered drawing methods this means that 8 used drawing methods maps to a 50% chance of being convincing evidence, 9 used maps to 70% and 11 used to 95%. The parameters were then determined with a standard curve fitting algorithm.

Integration. To test Apollo in practice, the tool was integrated with Atelier, our online platform for programming tutorials, where it creates comments with its assessment on every uploaded project. To reduce the chance that teaching assistants would interpret Apollo's assessments as grading, the probabilities calculated by Apollo were translated into words, following a mapping defined in [13].

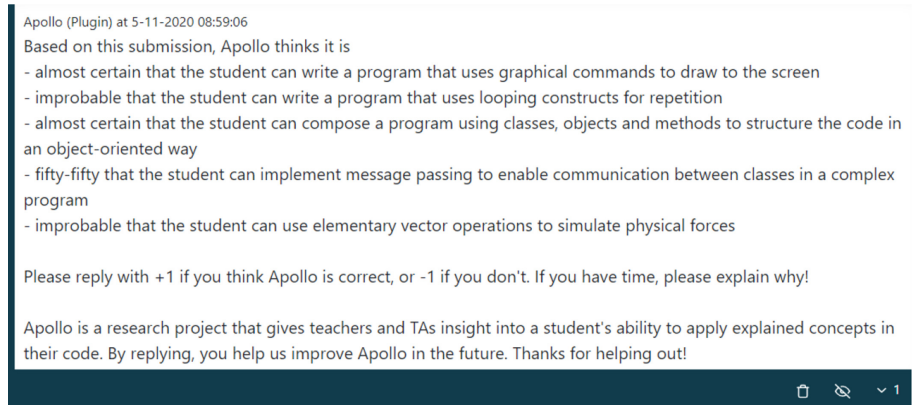


Fig. 3. An example comment by Apollo. The second icon in the lower left corner indicates that this comment is not shared with the student, but only visible for the teaching assistant.

A chance of 20% or less is rendered as ‘improbable’, for example. The intention is that teaching assistants use this information to address when they provide feedback, and not as a substitute for grading. See Fig. 3 for an example.

Apollo ran on 65 student submissions in the final week of the tutorial period where students could ask for feedback on their projects. On 11 of these submissions, a teacher or teaching assistant indicated whether or not they agreed with Apollo’s assessment. Out of these 11 comments, 8 were positive and 3 neither positive nor negative. Message passing and physics were both mentioned three times in these comments. In the case of message passing, always because Apollo indicated a low score on these programs and the commenter agreed with that assessment. For physics this was the case for two comments; the third indicated that there was some physics in the program which was not detected by Apollo.

The low response rate can at least partially be attributed to the fact that Apollo was only deployed in the last week of tutorials when many students seek help on their final projects. Teaching assistants rightly prioritise helping as many students as possible in the limited time they have during a tutorial.

In separate interviews with two teaching assistants, both indicated to agree with the comments made by Apollo most of the time. One of them indicated that they would like to get more information on why Apollo made the assessment, the other mentioned that the overall information presentation in Atelier could be better and that it sometimes feels “rather spammy.” Both did find the provided information useful when looking at a project uploaded by a student.

6 Conclusions

This paper reports on automatic assessment of learning objectives in a first-year programming course, and its implementation in Apollo. Apollo is an extension of

Atelier, our online platform that supports teaching staff in the process of giving feedback to students. To develop the tool, we first selected learning outcomes suitable for automatic analysis of students' code in a systematic way, starting with the Computer Science Curriculum Guidelines, followed by the learning outcomes for programming courses in our program. In general, learning outcomes on the correct usage of e.g. programming elements are suitable for automatic analysis, while more general goals concerning the understanding of an area are not. Based on this, we identified a set of five representative learning objectives to use for the development of Apollo.

For each of these learning objectives, we identified aspects that indicate mastery of that objective and implemented rules in Apollo to recognise these aspects in code. In the end, Apollo draws from both the static analysis and intention-based diagnosis techniques for providing automated feedback. The core of intention-based diagnosis is to uncover the goal a student had in mind while writing their code by identifying common patterns used to achieve these goals. This is most prominent in the learning objective on modelling physics, where concrete patterns were used to represent common goals in that area. For other objectives, we instead tried to characterise the common solutions to achieve a goal to avoid having to list all possible ways in which a loop can be used.

While validation of the tool was limited due to time constraints, feedback from teaching assistants does indicate that Apollo is a useful addition to the Atelier platform. Early signs also tend to indicate that Apollo's assessment is mostly correct, but further validation is required to make a clear statement on this. This will happen as part of an ongoing longer-term study.

There are two main areas in which Apollo could be improved. First, Apollo only assesses individual programs, but it cannot yet accumulate outcomes from a series of programs of one student to an overall chance that a student has achieved a learning objective. Neither can it combine the results of a group of students to create an overview of the overall progress in a course. These additions would extend the use of Apollo beyond giving feedback on single programs into giving insight into the learning paths of students.

Second, instead of a textual presentation, more intuitive visualisations of Apollo's results would be desirable. This becomes especially important when the results are combined into student- and group-level numbers, possibly tracked over time. An insightful presentation of results would also be useful for an evaluation to what extent the tool does improve the efficiency of giving feedback. These two areas will be the subject of further research.



References

1. ACM Computing Curricula Task Force (ed.): Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM, Inc. (2013). <https://doi.org/10.1145/2534860>
2. Blok, T., Fehnker, A.: Automated program analysis for novice programmers. In: Proceedings of the 3rd International Conference on Higher Education Advances. Universitat Politècnica València (2017). <https://doi.org/10.4995/head17.2017.5533>

3. Bloom, B., Englehart, M., Furst, E., Hill, W., Krathwohl, D.: Taxonomy of educational objectives. The classification of educational goals. David McKay Co., Inc, New York (1956)
4. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.* **4**(4), 253–278 (1995). <https://doi.org/10.1007/bf01099821>
5. Douce, C., Livingstone, D., Orwell, J.: Automatic test-based assessment of programming. *J. Educ. Resour. Comput.* **5**(3), 4-es (2005). <https://doi.org/10.1145/1163405.1163409>
6. Fehnker, A., de Man, R.: Detecting and Addressing Design Smells in Novice Processing Programs, pp. 507–531. Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-21151-6_24
7. Johnson, W.L., Soloway, E.: Intention-based diagnosis of programming errors. In: Proceedings of the 5th National Conference on Artificial Intelligence, Austin, TX, pp. 162–168 (1984)
8. Johnson, W., Soloway, E.: Proust: knowledge-based program understanding. *IEEE Trans. Softw. Eng.* **SE-11**(3), 267–275 (1985). <https://doi.org/10.1109/tse.1985.232210>
9. Keuning, H., Jeuring, J., Heeren, B.: A systematic literature review of automated feedback generation for programming exercises. *ACM Trans. Comput. Educ.* **19**(1), 1–43 (2019). <https://doi.org/10.1145/3231711>
10. Krathwohl, D.R.: A revision of Bloom’s taxonomy: an overview. *Theor. Into Pract.* **41**(4), 212–218 (2002). https://doi.org/10.1207/s15430421tip4104_2
11. Mader, A., Fehnker, A., Dertien, E.: Tinkering in informatics as teaching method. In: Proceedings of the 12th International Conference on Computer Supported Education. SCITEPRESS - Science and Technology Publications (2020). <https://doi.org/10.5220/0009467304500457>
12. Processing Foundation: Processing language reference (API) (2020). <https://processing.org/reference/>. Accessed 27 May 2020
13. Renooij, S., Witteman, C.: Talking probabilities: communicating probabilistic information with words and numbers. *Int. J. Approximate Reasoning* **22**(3), 169–194 (1999). [https://doi.org/10.1016/s0888-613x\(99\)00027-4](https://doi.org/10.1016/s0888-613x(99)00027-4)
14. Shiffman, D.: Learning Processing: A Beginner’s Guide to Programming Images, Animation, and Interaction. The Morgan Kaufmann Series in Computer Graphics, Elsevier Science (2015)
15. Soloway, E.: Learning to program = learning to construct mechanisms and explanations. *Commun. ACM* **29**(9), 850–858 (1986). <https://doi.org/10.1145/6592.6594>
16. Wilson, L.O.: The aims, goals and objectives of curriculum - what are the differences? (2014). <https://thesecondprinciple.com/instructional-design/writing-curriculum/>. Accessed 7 May 2020



Ex-Ante and Ex-Post Feature Evaluation of Online Courses Using the Kano Model

Daniel Moritz Marutschke¹  and Yugo Hayashi² 

¹ College of Global Liberal Arts, Ritsumeikan University, 2-150 Iwakura-cho, Osaka, Ibaraki 567-8570, Japan

moritz@fc.ritsumei.ac.jp

² Department of Comprehensive Psychology, Ritsumeikan University, 2-150 Iwakura-cho, Osaka, Ibaraki 567-8570, Japan

y-hayashi@acm.org

Abstract. Evaluating the effectiveness of online courses and what makes or breaks e-learning pose several challenges. Assessing educational data is generally a multivariate problem of high dimensionality. The implementation is often costly, the experimentation setup is complex, and supervision needs technical expertise. This research proposes an ex-ante and ex-post comparison of online course features using the Kano method from customer satisfaction analysis. Undergraduate students were asked to fill out questionnaires before and after taking a fully functional online course to compare their perceived importance of e-learning features. Attitudes towards 12 features, including ease of use, multimedia inclusion, account settings, and other specific features were gathered. The questionnaire also included feedback on overall experience, general positive and negative elements, and a free-form field for comments and suggestions regarding online courses. The results of this experiment suggest a shift in how students perceive the importance of features associated with online courses after successful completion.

Keywords: E-Learning · Kano model · Customer satisfaction

1 Introduction

The advent of the internet has propelled the dissemination of knowledge in an exponential manner. Many areas of education have benefitted from this ease of information availability. The fundamental structure of how education is understood, however, did not change significantly for more than a decade after that. And while the paradigm of an online course was originally not far from its analogue predecessor, recent development in machine learning, statistical analysis, and vigorous research have made progress in understanding educational mechanisms. The cost both in labour and expertise coupled with complex data handling limited the readiness to conduct research.

With online courses gaining popularity, the difficulty in understanding user satisfaction, user motivation, and learning benefits from e-learning systems still

remains. Early approaches were to popularize knowledge with Massive Open Online Courses (MOOCs) without concern for student motivation, others are from a business standpoint to retain students' attention with questionable learning benefits. Recent studies include investigations of student retention and to identify dropout reasons [3, 5, 7, 11].

Research papers span from learning motivation, exploring multiple hypothesis that tie to student satisfaction [15], satisfaction to more mechanical implementation, such as medical software usage [16], and the use of the Kano method to poll students on several e-learning factors [6] and to investigate a hybrid online/face-to-face course using blended learning [17].

How an online course is created and what content is presented to prospective students also depends on their background and subject matter interest. The significance of subject matter difference has been acknowledged since before e-learning, but has not been the focus of online course design [1, 9]. The authors of this present research took into account the target field of psychology students and created the online course with minimal technical focus. This step was also taken to be consistent with future research when online courses are expected to be compared between different study fields.

This research aims to identify learning values for students in higher education. In particular, undergraduate students of the department of comprehensive psychology at the Ritsumeikan university in Japan were surveyed. With this proposal, online course features will be gathered by questionnaires and evaluated using the Kano model (Fig. 1) [8]. An ex-ante (before) and ex-post (after) approach is used to determine a shift in perception of these features. The comparison of perceived importance of features before and after taking a course could provide another dimension to understand the effectiveness of e-learning systems. Previous studies mostly show either one or the other and rarely consider the academic depth of online courses, but rather from extrinsic motivations [4, 10, 14].

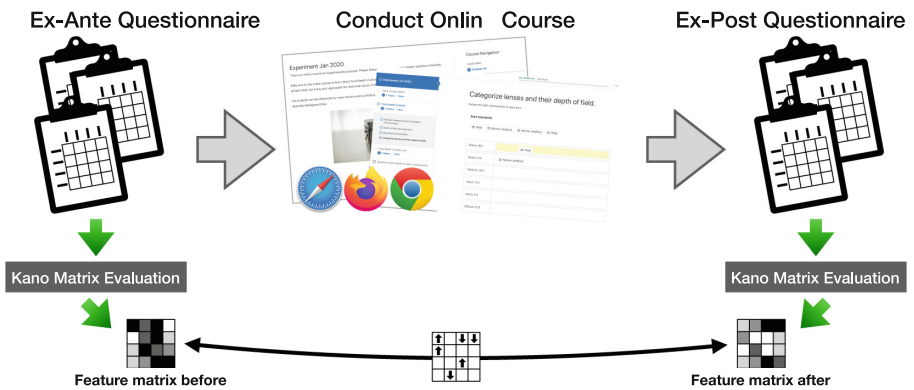


Fig. 1. Overview of the experiment setup.

The results will be analyzed to improve online course learning experiences. The findings are explained in Sect. 3.3 and show how the perceptions of online course features change.

This paper is structured as follows: Sect. 2 outlines the Kano method and its implication. Section 3 is structured into experiment setup describing in detail the online course setup, how the ex-ante and ex-post questionnaires tie into the process (Subsect. 3.2), and finishes with Subsect. 3.3 evaluating and discussing the results. The paper concludes with Sect. 4 summarizing the findings and shaping directions for future works.

2 Methodology

The Kano model was developed for customer satisfaction research and asks users to rate their perceived feeling, when a feature of a product or service is present (*functional* question) and or missing (*dysfunctional* question). Their response is rated from highly satisfied to highly dissatisfied for both question dimensions. For each feature, the combination of the functional and dysfunctional question results in one of the following six:

- B (basic requirement)
- O (one-dimensional requirement)
- I (indifferent requirement)
- Q (questionable requirement)
- A (attractive requirement)
- R (reverse requirement)

The functional and dysfunctional dimensions and how each feature is classified as in the list above is illustrated in Fig. 2. Participants have to rate a feature in five levels—highly satisfied, as expected, neutral, can live with it, highly dissatisfied.

In the case of an e-learning platform or online course, the following illustrates a selection of features. For example a working website, consistent URLs, and everything viewable on PC and mobile would be considered basic requirements (B). They do not increase the satisfaction if properly included, but would reduce the satisfaction if missing. An attractive feature (A) could be considered a live note-taking function or gamification (earning points or badges after completing tasks). Often over time, attractive features become basic requirements (B).

One-dimensional requirements (O) increase the user's satisfaction proportional to the degree of implementation. For example a factor that results in a good User Experience (UX) increases the satisfaction proportionally to the degree of implementation. The opposite is also the case as poor implementation of factors that decrease UX also decreases the satisfaction.

For this research, the Kano model was chosen to gain unique insight into online course design and to include in quantitative user data analysis. The Kano model includes states that cannot be captured with other methods in this field. Reverse requirements (R) have the opposite effect of (A), i.e., they decrease the

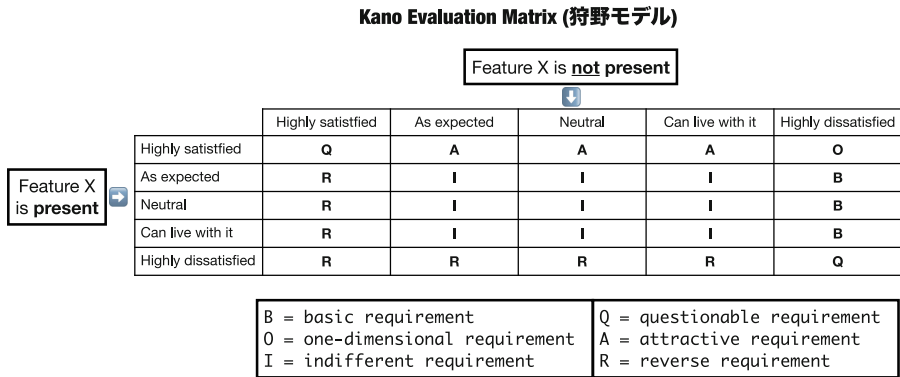


Fig. 2. Overview of the Kano model and how features are classified.

satisfaction when implemented. An example for web-based applications would be pop-up dialogue boxes. This is one of the reasons for choosing this methodology over others commonly used ones [12, 13, 18].

Questionable requirements (Q) are inconsistencies in the questionnaire answers. If a user answers both highly satisfied if a feature is included and is not included, the answer is unusable, in other words questionable (Q).

The above mentioned classifications can qualitatively be visualized on a two dimensional plane with grade of implementation and customer satisfaction (Fig. 3). Ordinary quality, or one-dimensional requirements, grow linear, where as other features show exponential or flattening impact.

3 Results and Discussion

The following section covers the setup of the experiment, how the questionnaire was made to be used with the Kano method (including differences in the ex-ante and ex-post questionnaire), which features were investigated, and an evaluation and discussion of these results.

3.1 Experiment Setup

The experiment was conducted with a total of 16 Japanese students, 5 male and 11 female. The average age was 19.82 ($\sigma = 1.70$), ranging from 18 to 24.

Students can earn extra credit by participating in research experiments. The university has set up this system as means to let students gain additional knowledge and experience. All participants were informed about the content of the experiment, the length of the session, and language requirements, amongst other administrative details.

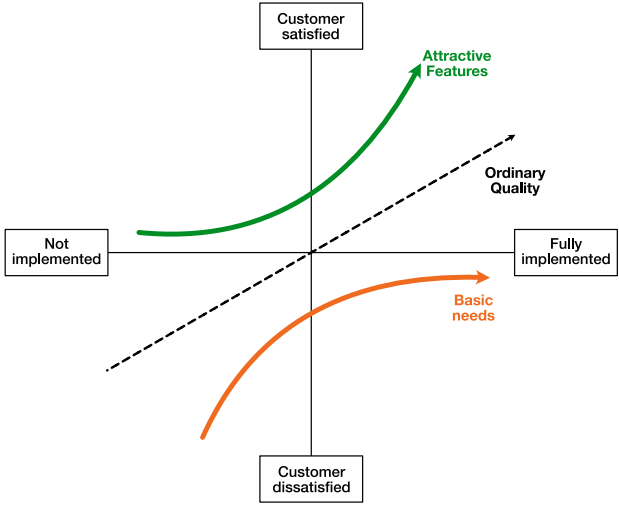


Fig. 3. Kano model requirements illustrated on a plane with grade of implementation and customer satisfaction. Attractive features, basic needs, and the ordinary quality have different velocity of impact.

The format of the experiment was a 90 min session. Introduction, setup explanation, and concluding remarks with describing the research goal and implications were provided at the beginning in Japanese. The rest of the conduct was in English.

The online course was setup with Wordpress and two premium plugins—LeanDash and MemberPress. Wordpress is one of the most popular blogging websites and has a version that can be installed and manages on one’s own server. Although described at the beginning of this paper as costly, time-consuming, and complex, this is an important advantage for propagation of research and reproducibility. Wordpress has evolved into a full-fledged content management system (CMS) and allows the creation of dynamic websites with numerous members at scale. LearnDash allows to create fully customizable online courses with enrollment process, lectures with sub-topics, quizzes, assignments, and statistics on Wordpress sites. The advantage here as well is the control of the system administrator on how to implement the online course in functionality and look. This was kept in mind for future research direction, where findings will be used to improve the online course implementation. The MemberPress plugin allows to manage and fine-tune memberships. MemberPress has since been removed from future experiment setups due to functionality overlap after updating LearnDash and unreliable behavior with the same. Additional plugins were used to protect content and create preset accounts in bulk.

Due to the limited timeframe of the experiment and diverse student background, photography was chosen as a semi-technical topic, in which a multitude of content can be implemented naturally. To narrow down the very broad topic

and tie it to the field of psychology, an introduction to depth of field, its artistic use, and lens design was given in the online course. The course consisted of three lessons, each with at least one sub-topic and a quiz, and a final exam. The content was kept straight-forward with minor technical details about focal length, aperture values, distance from the object, and sensor size. These were explained how they values influence the depth of field. Quizzes were repeated in the final exam to limit strain on the working memory and to allow students to go back and revisit their previous answers. All quizzes were equipped with individual feedback depending on the students' answers.

3.2 Questionnaire for Kano Model

Questionnaires for the Kano model were given before and after the online course was taken. Each participant was given a unique identifier to anonymize the questionnaire, but allow tracking of course content with the survey results. Corresponding accounts on the website were created in advance and students received login information at the beginning of the experiment.

Apart from the questions relevant for the Kano model analysis, additional questions regarding the participants and their overall experience were gathered. These were compiled to reflect demographics to make future comparisons possible. Wording and formulation were conferred with the co-author and separately proofread. From the questionnaire, it was visible that all students had sufficient proficiency to follow the online course. Although smartphone use was prohibited during the experiment, electronic dictionaries (popular in Japan) were allowed.

Ex-Ante Questionnaire (Before). The following questions about the participants were collected before the online course:

- Gender (female, male, other)
- Age
- University grade
- Self-assessed English proficiency (none, basic, moderate, advanced, native level)
- Self-assessed knowledge about photography (none basic, moderate, advanced, professional)
- Open ended question regarding missing features in the provided feature list

According to self-assessed English proficiency, one student indicated *none*¹, 12 indicated *basic*, and four indicated *moderate*.

Ex-Post Questionnaire (After). The following questions were asked after completing the online course:

¹ From several questions asked by this student in the open-ended question section, the authors could infer an English comprehension of *basic* rather than *none*.

- The overall experience of the course (very bad, bad, neutral, good, very good)
- Open ended question for positive feedback about the online course
- Open ended question for negative feedback about the online course
- Open ended question regarding missing features in the provided feature list

Kano Model Features. Functional and dysfunctional questions were asked in both the ex-ante and ex-post questionnaire. The following list of features for online courses were considered:

- User-friendly platform
- Certificate of completion
- Download of course material
- Own profile and account page
- Quizzes and exercises
- Interactive quizzes and exercises
- Comment function
- Personal tutor
- User manual for the platform
- Videos
- Photos/Graphics
- Text

Participants are asked about the same features how they perceive it being implemented (functional) and how they perceive it missing (dysfunctional) (Fig. 2).

3.3 Evaluation and Discussion

To see the shifting in perception of functional and dysfunctional questions from before to after the online course was taken, two box plots were analyzed (Figs. 4 and 5). Each graph includes a before and after comparison. The scale given by the Kano model questionnaire was transformed into numerical values—1: I like it; 2: As expected; 3: Neutral; 4: I can live with it; 5: I dislike it. The results show that is hard to take the two dimensions separately into account and points to the strength of the Kano model. Although some shift can be seen, a statistical evaluation proved difficult. One of the reasons is the limited number of participants in this study and the authors expect to gain additional insight with larger number of students.

Taking both functional and dysfunctional questions and applying the Kano model, each feature is assigned a requirement classification for every participant, as illustrated for one feature in Fig. 2. Taking all evaluations into account, each feature has a customer satisfaction value (CS) and a customer dissatisfaction value (CD) that can be calculated as shown in Eqs. 1 and 2.

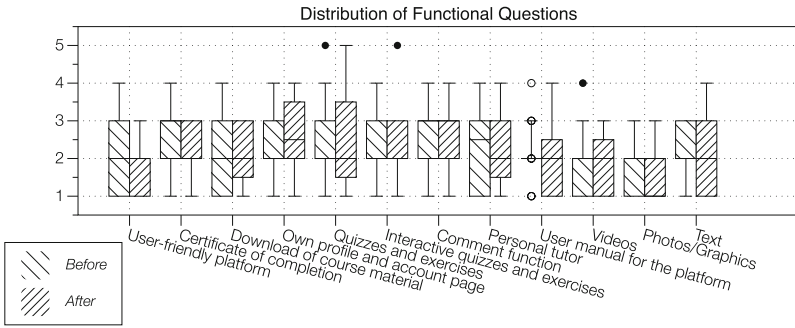


Fig. 4. Box plot of functional questions, comparing before and after taking the online course.

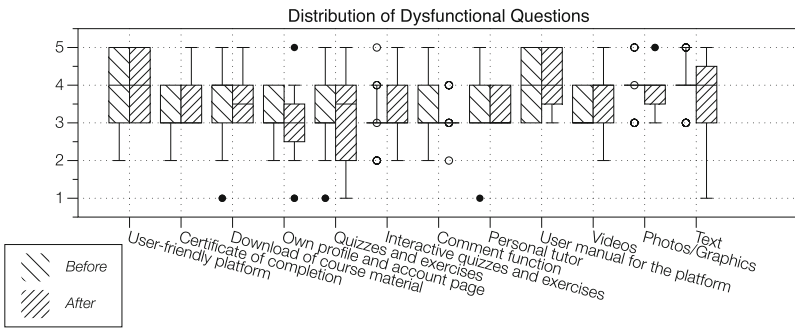


Fig. 5. Box plot of dysfunctional questions, comparing before and after taking the online course.

The two equations indicate at what rate the customer satisfaction is raised when including a feature and how much the customer would be dissatisfied when excluding a feature. This ratio is to understand a cost-benefit tradeoff of gaining satisfaction and preventing dissatisfaction [2].

$$CS = \frac{A + O}{B + O + A + I} \tag{1}$$

$$CD = \frac{B + O}{B + O + A + I} \cdot -1 \tag{2}$$

After calculation of customer satisfaction and dissatisfaction (CS and CD) for all features, a graphical overview is presented in Fig. 6. Their differences can be inspected in detail in Table 1.

Values are labeled in Fig. 6 only for differences that are relevant for the evaluation and discussion in Sect. 3.3.

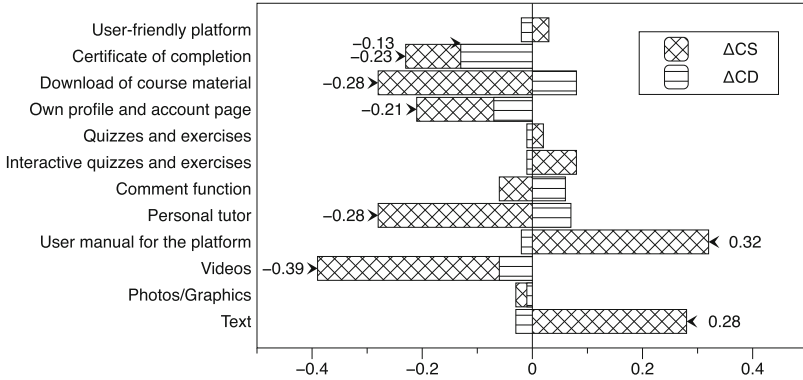


Fig. 6. Change of customer satisfaction and dissatisfaction values (ΔCS and ΔDS) from before to after taking the online course.

Most notably was the drop in importance of videos implemented in the system. The initial value on multimedia, especially on videos, was with 0.82 and 0.59 for videos and photos/graphics, respectively amongst the highest. This suggests the evaluation by subjects without prior experience could be skewed towards expectations that are not practical or desired for an actual users.

The download of course material was also deemed important in the beginning, but less so after completion. The drop in CS here occurred without increase in CD.

A certificate of completion was issued to all participants in PDF format after successful completion of the online course. A threshold was set to issue the certificate only after 80% of the final exam were absolved. The final exam was a mix of quizzes previously absolved to keep the threshold relatively achievable. There was a drop in CS here as well. However, with the added incentive of a sense of achievement could explain the heightened dissatisfaction with failure to implement.

The ability to access and personalize an account page experienced a similar dynamic as the certificate of completion.

As two professors were standing by to help and answer questions, the difference in availability of a personal tutor should be accordingly taken into account.

There was no significant change in the need for a user-friendly platform, which was 0.41 (before) and 0.44 (after). This might tie into the wish for a user manual for the platform, as this increased the most amongst the gathered features. As ΔCD did not dramatically change, the cost-benefit analysis is difficult to make at this point and more investigation is needed to see its impact.

Table 1. Numerical overview of the changes of customer satisfaction and dissatisfaction values (ΔCS and ΔCD) from before to after the online course.

Features	ΔCS	ΔCD
User-friendly platform	0.03	-0.02
Certificate of completion	-0.23	-0.13
Download of course material	-0.28	0.08
Own profile and account page	-0.21	-0.07
Quizzes and exercises	0.02	-0.01
Interactive quizzes and exercises	0.08	-0.01
Comment function	-0.06	0.06
Personal tutor	-0.28	0.07
User manual for the platform	0.32	-0.02
Videos	-0.39	-0.06
Photos/Graphics	-0.03	-0.01
Text	0.28	-0.03

4 Conclusions

This research investigates a set of 12 features for online course implementation using the Kano model from customer satisfaction. As previous research has shown, the model can help identify different requirement categories for such features. Due to the nature of costly implementation and maintenance of online courses, an ex-ante and ex-post comparison of these features were compared to identify changes in perceived importance if these are included or excluded.

The results suggest that the perception of users change after successfully completing an online course. Expectations and post consumption attitude change depending on the online course feature. This knowledge can help in evaluating Kano's model in online course design. This indicates a need for more in-depth view into the use and construction of e-learning systems.

As the experiment had limitations ranging from time constraints in taking the online course to a small sample size, future investigations are needed. Future works include the increase of participants, participants from different fields of study, and multicultural participants.

References

1. Becher, T.: The significance of disciplinary differences. *Stud. High. Educ.* **19**(2), 151–161 (1994). <https://doi.org/10.1080/03075079412331382007>
2. Berger, C., Blauth, R.E., Boger, D.: Kano's methods for understanding customer-defined quality. *Center Qual. Manage. J.* **2**, 3–36 (1993)
3. Chen, C., Sonnert, G., Sadler, P.M., Sasselov, D.D., Fredericks, C., Malan, D.J.: Going over the cliff: MOOC dropout behavior at chapter transition. *Distance Educ.* **41**(1), 6–25 (2020). <https://doi.org/10.1080/01587919.2020.1724772>

4. Chen, L.H., Kuo, Y.F.: Understanding e-learning service quality of a commercial bank by using kano's model. *Total Qual. Manage. Bus. Excellence* **22**(1), 99–116 (2011). <https://doi.org/10.1080/14783363.2010.532345>
5. Chen, Y., Zhang, M.: MOOC student dropout: pattern and prevention. In: *Proceedings of the ACM Turing 50th Celebration Conference - China*. ACM TUR-C 2017, Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3063955.3063959>
6. Dominici, G., Palumbo, F.: How to build an e-learning product: factors for student/customer satisfaction. *Bus. Horiz.* **56**(1), 87–96 (2013)
7. Goopio, J., Cheung, C.: The MOOC dropout phenomenon and retention strategies. *J. Teach. Travel Tourism* 1–21 (2020). <https://doi.org/10.1080/15313220.2020.1809050>
8. Kano, N., Seraku, N., Takahashi, F., Tsuji, S.I.: Attractive quality and must-be quality. *J. Japan. Soc. Qual. Control* **14**(2), 147–156 (1984). <https://doi.org/10.20684/quality.14.2.147>
9. Katai, Z.: Promoting computational thinking of both sciences- and humanities-oriented students: an instructional and motivational design perspective. *Educ. Tech. Res. Dev.* **68**(5), 2239–2261 (2020). <https://doi.org/10.1007/s11423-020-09766-5>
10. Chen, L.H., Lin, H.C.: Integrating Kano's model into e-learning satisfaction. In: *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 297–301 (2007). <https://doi.org/10.1109/IEEM.2007.4419199>
11. Liyanagunawardena, T.R., Parslow, P., Williams, S.: Dropout: MOOC participants' perspective. In: *EMOOCs 2014, the Second MOOC European Stakeholders Summit*, pp. 95–100, February 2014. <http://centaur.reading.ac.uk/36002/>
12. Parasuraman, A., Zeithaml, V.A., Berry, L.: SERVQUAL: a multiple-item scale for measuring consumer perceptions of service quality. *J. Retail.* **64**, 12–40 (1988)
13. Parasuraman, A., Zeithaml, V.A., Malhotra, A.: E-S-Qual: a multiple-item scale for assessing electronic service quality. *J. Serv. Res.* **7**(3), 213–233 (2005). <https://doi.org/10.1177/1094670504271156>
14. Selim, H.M.: Critical success factors for e-learning acceptance: confirmatory factor models. *Comput. Educ.* **49**(2), 396–413 (2007)
15. Sun, P.C., Tsai, R.J., Finger, G., Chen, Y.Y., Yeh, D.: What drives a successful e-learning? An empirical investigation of the critical factors influencing learner satisfaction. *Comput. Educ.* **50**(4), 1183–1202 (2008)
16. Violante, M.G., Vezzetti, E.: Virtual interactive e-learning application: an evaluation of the student satisfaction. *Comput. Appl. Eng. Educ.* **23**(1), 72–91 (2015). <https://doi.org/10.1002/cae.21580>
17. Wang, Y.S., Bauk, S., Šćepanović, S., Kopp, M.: Estimating students' satisfaction with web based learning system in blended learning environment. *Educ. Res. Int.* **2014** (2014). <https://doi.org/10.1155/2014/731720>
18. Wang, Y.S., Wang, H.Y., Shee, D.Y.: Measuring e-learning systems success in an organizational context: scale development and validation. *Comput. Hum. Behav.* **23**(4), 1792–1808 (2007)



Automated Summary Scoring with ReaderBench

Robert-Mihai Botarleanu¹, Mihai Dascalu^{1,2(✉)}, Laura K. Allen³,
Scott Andrew Crossley⁴, and Danielle S. McNamara⁵

¹ University Politehnica of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania
robert.botarleanu@stud.acs.upb.ro, mihai.dascalu@upb.ro

² Academy of Romanian Scientists, Str. Ilfov, Nr. 3, 050044 Bucharest, Romania

³ University of New Hampshire, Durham, Durham, NH 03824, USA
laura.allen@unh.edu

⁴ Department of Applied Linguistics/ESL, Georgia State University, Atlanta, GA 30303, USA
scrossley@gsu.edu

⁵ Department of Psychology, Arizona State University, PO Box 871104,
Tempe, AZ 85287, USA
dsmcnama@asu.edu

Abstract. Text summarization is an effective reading comprehension strategy. However, summary evaluation is complex and must account for various factors including the summary and the reference text. This study examines a corpus of approximately 3,000 summaries based on 87 reference texts, with each summary being manually scored on a 4-point Likert scale. Machine learning models leveraging Natural Language Processing (NLP) techniques were trained to predict the extent to which summaries capture the main idea of the target text. The NLP models combined both domain and language independent textual complexity indices from the ReaderBench framework, as well as state-of-the-art language models and deep learning architectures to provide semantic contextualization. The models achieve low errors – normalized MAE ranging from 0.13–0.17 with corresponding R^2 values of up to 0.46. Our approach consistently outperforms baselines that use TF-IDF vectors and linear models, as well as Transformer-based regression using BERT. These results indicate that NLP algorithms that combine linguistic and semantic indices are accurate and robust, while ensuring generalizability to a wide array of topics.

Keywords: Natural language processing · Text summarization · Automated scoring

1 Introduction

Scoring student writing, which in many cases consists of essays and summaries, is one of the most time-consuming activities teachers have to perform. Yet, it is necessary across the majority of grade levels, academic domains, and in many countries. Teachers must carefully read and evaluate the piece of writing for spelling errors, cohesion and coherence, alignment with the task requirements, plagiarism, and other norms and requirements. Summary evaluation requires even further criteria, such as the faithfulness

of the summary to the reference text, the degree to which the summary abbreviates the original reference text, and the objectivity of the summary. The lack of sufficient time for many teachers (who already have excessive burdens) can thus limit opportunities for students to receive sufficient feedback on their summary writing.

In this work, we propose a method for automatically evaluating student summaries to predict main idea coverage. Our aim is to build an Automated Summary Scoring tool that can be used by both students and teachers. For students, the capability to have their summaries evaluated before handing them in would enable an iterative learning process wherein they could write a draft, have it automatically scored, and then work to improve it before the final submission to their teacher. This would allow learners to improve their summary writing skills through a more consistent and timely feedback loop. For teachers, automated scoring can help lower their workload. Automated Summary Scoring systems can support teachers by affording them more time to focus on rhetorical aspects of students' writing, and in turn provide one-to-one assistance to individual students.

One challenge faced by Automated Summary Scoring systems considers their generalization capabilities across topics and target texts. Thus, we address the following research questions:

1. To what extent do summary scoring models generalize across different reference texts?
2. Does performance expressed as Mean Average Error vary when using neural models relying on textual complexity indices or BERT language models?
3. Can novel insights be gleaned about the underlying summary scoring process from feature importance information extracted from the trained neural models?

To achieve these goals, we explored the use of three types of features to predict main idea coverage in summaries: TF-IDF, hand-crafted linguistic and semantic features, and latent contextualized representations computed with BERT [1]. We also examined the efficacy of three types of machine learning models (Random Forest [2], Lasso [3], Neural Networks including feed-forward networks on top of textual complexity indices and BERT). Once trained, we analyze the most important features used by the two best performing models to identify the most relevant information used for automated scoring. In the remainder of this paper, we provide an overview of related work on the automated evaluation of student writing. We then describe our methodological approach and results of our analyses. We then conclude with a discussion of our findings and suggestions for future work.

2 Related Work

There are two primary means through which student writing is automatically assessed [4]: Automated Writing Evaluation (AWE) systems and Automated Essay Scoring (AES) systems. These two systems are commonly used to assess essays, but not summaries. AWE systems offer targeted, constructive feedback to student users with the purpose of helping them improve their writing, whereas AES systems are primarily focused on the generation of a numerical score of writing quality (i.e., a summative score). Here, we present an approach that falls under the category of AES systems.

Various AES systems have been developed to assess multiple genres of writing. e-Rater [5] was one of the first and relies on a wide range of features that measure grammar, usage, mechanics, style, organization, development, lexical complexity, and prompt-specific vocabulary usage. The initial version of e-Rater offered users a method of manually combining these features using weighted averages in an intuitive and explainable system. The past decade has seen considerable progress in the field of text scoring and numerous approaches have been explored. More recent approaches rely on neural networks to score student writing. For example, SkipFlow [6] uses a mechanism for modeling relationships between hidden representation snapshots generated by Long Short-Term Memory Networks [7]. Hochreiter and Schmidhuber [7] trained a network to predict human scores for a set of essays that were written in response to eight prompts. Their model achieved an average Quadratic Weighted Kappa of 0.764, denoting a high level of agreement with the human scores. Alikaniotis, Yannakoudakis and Rei [8] construct a fully automated framework based on LSTMs trained on the same dataset as SkipFlow, with a reported Spearman rank correlation coefficient of 0.91.

Taghipour and Ng [9] used a combination of a convolutional layer to extract local features from the texts, followed by a Recurrent Neural Network to predict the human scores. Similarly, Jin, He, Hui and Sun [10] introduced a two-stage neural network that aims to increase the performance of AES models in prompt-independent contexts. Their network was trained on human-rated essays with different prompts to detect essays with a level of quality that has high deviation from the average; then, these essays were used as pseudo-training data in the second stage.

Our approach consists of a simpler model, based on domain and language independent indices. We also consider the interpretability of our model and attempt to find the most relevant indices used by the Neural Network for our target evaluation criteria.

3 Method

3.1 Corpus

Our corpus consists of 2,976 summaries of 87 reference texts. Expert human raters provided summary scores on seven different analytic measures, which reflect various qualities of the summary and were manually evaluated on a 1 to 4 Likert scale: main idea coverage (“main point”), amount of key conveyed information (“details”), summary cohesiveness (“cohesion”), use of appropriate paraphrasing (“paraphrasing”), use of lexical and syntactic structures beyond those present in the reference text (“language beyond source text”), objectivity of the language used (“objective language”) and summary length. As a proof-of-concept, the current study focuses only on the prediction of the *main idea coverage* criteria. All expert raters were normed on a set of summaries not included in the main dataset. The raters were considered normed once their inter-rater reliability (IRR) reached Kappa .70. After norming, raters scored each summary independently. IRR after independent rating reported Kappa > .60. After independent rating, raters adjudicated any scores that differed by more than one between the raters.

Given the diversity of the corpus, we opted to perform a selection of the test data based on the statistical distributions of the human scores, with the aim of choosing a subset of reference texts and their corresponding scored summaries that provided a wide range of quality. We first combined the seven target scores into a single measure by summing the values for each. We checked for strong multicollinearity (defined as $r > .899$) and found that none of the variables correlated above that threshold with each other (correlations ranged between .37 and .72). Afterwards, the population variance was measured for each of the 87 reference texts ($M = 16.87$; $SD = 10.10$; $Min = 1.30$; $Max = 44.26$). Sorting the source texts in decreasing order of their population variance, we then select a number of reference texts that amount to at least 10% of the number of summaries in the corpus and that have at least 30 summarizations.

In developing the test set, we ensured that none of the selected summaries had reference texts present during training and that there was a large number of summaries, with a wide variance of target scores. In the end, our test data was based on three reference texts, included ~10% of the data that included the highest population variance (i.e., the widest range of possible values). This selection guaranteed that the test set contains examples that have both well written summaries, as well as poorly written ones, ensuring that it is sufficiently complex in order to properly evaluate the effectiveness of our models.

3.2 Linguistic and Semantic Features

We used the ReaderBench framework [11] to generate over 730 linguistic and semantic textual complexity indices, covering the following categories:

- **Surface.** Indices that measure statistical attributes of the text such as the number of words, punctuation marks, and character entropies.
- **Morphology.** Indices regarding parts of speech (e.g., noun, verb, adverb).
- **Syntax.** Indices using parse trees to define quantifiable information on the syntactical structure of the text. These include the parse tree imbalance, depth, and others.
- **Cohesion.** Indices derived from Cohesion Network Analysis [12] that measure semantic similarities between text elements (i.e., paragraph, sentences, words).
- **Co-reference.** Indices measuring the length of coreference chains and semantic overlap between words and concepts.
- **Lexical.** Various indices related to lexical features (e.g., hypernymy, polysemy counts, word frequency, word familiarity and lexical complexity).
- **N-gram.** Bi-gram and tri-gram frequencies, such as the number of unique and the total number of n-grams found in a text.
- **Subjectivity.** Frequency of subjective and objective words and phrases.

ReaderBench features were augmented with indices reflecting the degree of overlap between the summary and reference texts, such as their cosine similarities, the Jaccard overlap of their n-grams, and the percentage of the summary that constitutes novel or existing vocabulary with regards to the reference text.

In total, 1466 features were initially considered and were normalized using z-scores. For linear regression models, variance inflation factor was used to filter these features into a subset of 67 that did not exhibit multicollinearity. We also considered applying some common-sense filtering to these indices; for example, using only indices potentially related to text cohesion or summary length targets. However, we found that there are non-trivial interactions between the reference and the summary texts. For instance, reference texts that are already fairly compact in details, but lengthy, would also lead to fairly lengthy summaries. In such a case, even though the absolute number of words in the summary may be larger than in other cases, a summary that manages to preserve key information from the reference text, while performing only minimal shortening, would be found more appropriate than a summary with too much information removed. As such, using indices measured only on the summary text would not give the model enough information to accurately predict the human ratings.

3.3 Machine Learning Models

Machine learning regressors were trained on our dataset to predict the *main idea coverage* score. In order to have baseline evaluations for the selected models and features, we selected to use Random Forest [2] and Lasso Regressors [3]. These models were trained using the ReaderBench linguistic and semantic indices [11] and the vectors representing TF-IDF scores [13]. We also utilize Linear Regressors that use ReaderBench indices to predict the main idea coverage score. Since Linear Regression has issues with multicollinearity that the non-linear models (i.e., Neural Networks, Random Forests) do not, we utilize a Variance Inflation Factor cutoff of 10 [14] to select a subset of indices that does not present multicollinearity. For our models, we elect to not discretize the target score into categorical variables because it has a relevant numerical order, and intermediate values (e.g., a predicted score of 3.2) can still be useful for the user, as they can indicate whether a summary is closer to one range of the rounding interval than the other (e.g., summary is closer to 3.5 than 2.5).

The architecture of the Neural Network model is provided in Fig. 1a. The feed-forward network consists of a single hidden layer alongside ReLu activations [15], together with Batch Normalization layers [16] for controlling covariate shift between layers, and Dropout [17] layers with rate p ranging from 0.2 for the input to 0.5 for intermediate layers (0.5 denotes that half of the inputs are zeroed before being used by the successive layer). This helps control the variance of the model and prevent it from overfitting. The target consists of a single continuous variable for the regressors trained on the main idea coverage score. These models are all trained using a One-Cycle Policy [18] for 50 epochs with a batch size of 8. The optimal learning rate for the One-Cycle Policy was searched in a logarithmic space from $1e-5$ to 10 for 70 data points.

We also examined the performance of a BERT [1] model. As shown in in Fig. 1b, the output embeddings were concatenated and then passed through a non-linear layer to perform regression, which was run on both the summary and the reference text. For the BERT model we removed the prediction heads used during pre-training and added a regression head. Since the source texts can exceed the limit of 512 tokens typically

used by BERT, we elect to only use the first 512 tokens in these. We have experimented with running the BERT model on blocks of 512 tokens from the source texts and then concatenating the representations; however, the results were poorer than the simpler alternative that trims texts that are too long.

The BERT model was fine-tuned over 7 epochs, utilizing linear schedule with warmup with a learning rate initialized at 0.0001 and the Adam optimizer. We explored different hyperparameter configurations and varied the width and depth of the regression head that uses the BERT outputs and found the best success with using the standard fine-tuning hyperparameters together with a single fully-connected layer used to combine summary and reference features, before estimating the score. Finally, we assessed two models that combined the ReaderBench and BERT indices. Leveraging the architecture illustrated in Figs. 1a and b, the input comprised a concatenation of the document representations generated by BERT, combined with the ReaderBench indices. The combined model attempts to simultaneously finetune BERT and learn to use the ReaderBench indices to predict the target score. The training setup uses the same configuration as the BERT-based model.

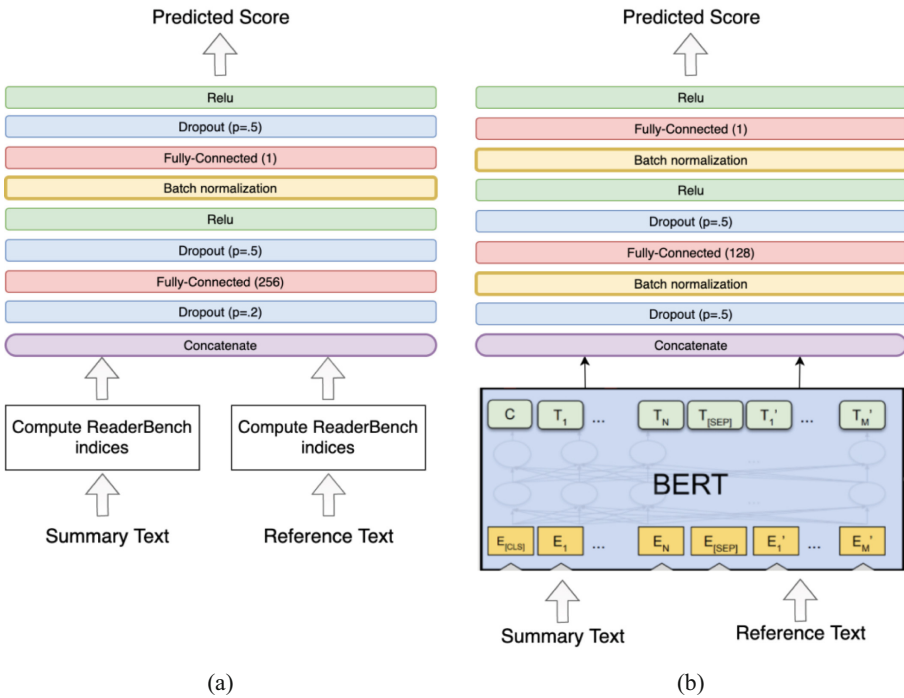


Fig. 1. a. ReaderBench neural network model architecture. b. BERT architecture.

4 Results

4.1 Prediction of Main Idea Coverage Score

The results for predicting main idea coverage scores are presented in Table 1. We can observe that ReaderBench Neural Network model outperforms the BERT and the TF-IDF models. This indicates that the general-purpose language baselines were outperformed, on average, by the networks trained using textual complexity indices. Comparing the three types of machine learning models that used ReaderBench indices, the neural network model tended to yield better results than the other three models.

Table 1. Normalized MAE and R^2 for the “Main Idea Coverage” summarization evaluation criterion.

Models	Normalized MAE	R2
TF-IDF (Lasso)	.17	-.09
TF-IDF (RF)	.17	-.12
ReaderBench: Linear Regression	.16	.15
ReaderBench: Lasso Regression	.17	-.07
ReaderBench: Random Forest	.16	.18
ReaderBench: Neural Network	.13	.46
BERT	.16	.15
Combined model (ReaderBench & BERT)	.14	.39

4.2 Feature Importance

The relevance of features can be measured for the Random Forest Regressors using the Gini importance, with features being assigned importance values, defined as a normalized measurement of the amount of reduced impurity. Linear models can have their feature importance values directly measured through the feature coefficients after training. Because the neural models used are non-linear networks, we selected Integrated Gradients [19], a method of approximating feature importance by using the gradients resulted from the loss function, for a given sample. Starting from an arbitrary baseline, a line integral is computed along the path from the baseline to the sample, with respect to the feature gradients. This is then scaled with the distance between each intermediate sample and the baseline. The equation describing the integrated gradients of a feature i , using a sample x and a baseline x' , is the following:

$$IntegratedGradients_i(x) = (x_i - x'_i) \times \int_{\alpha=0}^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \quad (1)$$

Integrated gradients, by design, only measure the relative importance of a feature with regards to a given sample and baseline. In our case, the baseline x' is a zero vector (i.e.,

no indices are measured). The integrated gradients for each feature were measured on the entirety of the training set in order to obtain dataset-wise results, instead of sample-wise results obtained by averaging the sample values.

We present a selection of 5 features from the 10 most important features by the magnitude of their integrated gradients on the best performing model (see Table 2), accompanied by the top 5 features according to their Gini importance for the Random Forest model. For each index, we also specify whether it was measured on the reference (i.e., reference text), the summary, or if it is an overlap index. In addition, we marked each feature with “±” to highlight whether the corresponding gradients have a positive or negative average, before multiplying this value with the difference between the sample and the intermediate baselines. This gives a sense of the directionality of the features. The reason for choosing this method of determining directionality, instead of more traditional approaches (e.g., Spearman rank correlations), is that many important features marked by the model are not linearly correlated with the target variable. The sign of the gradients, on the other hand, should give an indication as to the directionality of the features.

The interpretability of a neural network using integrated gradients is significantly more limited than what the coefficients of a trained linear model can yield. While feature importance is useful as rough guideline, it does not appropriately express the complexity of non-linear interactions that the model uses to make its prediction. Since the model considers more than a thousand features, results in Table 2 give only a shallow understanding of what the model is doing on average, across the testing data. Another important observation is that integrated gradients are commonly used on a per-sample basis, whereas we attempted to extrapolate a global understanding of the behavior of the model, by aggregating the results on each testing sample. The features were chosen to show an equitable distribution of both positive and negative directionalities, in order to give a better insight into the behavior of the model.

Although there is a significant amount of noise in terms of features that have high importance according to the Integrated Gradients method, others are much more intuitive and three of the five are also reported by the Random Forest model. For instance, the presence of overlap features in the main idea coverage score is expected, since this score is dependent on the nature of the original text and how well the summary manages to capture its reference material. Of these overlap features, the “Source-Summary Similarity” is defined as the cosine similarity between the two texts, the “Existing Vocabulary” reflects the vocabulary overlap between them, and the “Jaccard overlap” index measures the similarity between the n-gram sets of the two texts. “Average parse tree imbalance” and the “average block tree depth” are measures of textual structural complexity, while “character entropy” gives a statistical understanding of a text’s repetitiveness with low entropy texts typically corresponding to low effort writing. Integrated gradients provide a straightforward measurement of feature importance in neural networks; however, it is a post-hoc interpretation that only approximates the most important features, whereas the non-linearity of neural networks cannot be expressed through simple scores assigned to each input. Nevertheless, the use of integrated gradients and other similar approaches is a way of circumventing the black box nature of modern neural network models and can offer insight into what neural models are actually evaluating during inference.

Table 2. ReaderBench indices with high feature importance used by the NN and RF models

	Neural network	Random forest
1	<i>Summary – Character Entropy (-)</i>	<i>Overlap – Source-Summary Similarity</i>
2	<i>Overlap – Source-Summary Similarity (+)</i>	<i>Overlap – Source-Summary Existing Vocabulary</i>
3	Summary – Average Parse Tree Imbalance (-)	Overlap – Source-Summary Jaccard Overlap
4	<i>Overlap – Source-Summary Existing Vocabulary (+)</i>	Summary – Average Block Tree Depth
5	Source – Character Entropy (-)	<i>Summary – Character Entropy</i>

Note: “±” indicates positive or negative gradient values before multiplying this value with the difference between the sample and intermediate baselines. Common indices between the NN and RF models are marked in italics and grayed out cells

5 Conclusions

We performed predictive modelling on a dataset consisting of 2,976 summaries on 87 reference texts to predict main idea coverage. Our results show that, for datasets of this size, the use of hand-crafted features is still very important, with models trained using a variety of textual indices outperforming on average the results of both classic Machine Learning models (such as those based on TF-IDF scores) and state-of-the-art language models (such as BERT). The limitations of BERT, which was designed with larger datasets of shorter texts in mind, made it so that a simpler, fully-connected model, was able to outperform it consistently across different variations and hyper-parameter configurations on our dataset. We relied on a rigorous approach of selecting a testing set such that it precludes any sort of look-ahead bias. In addition, we introduce integrated gradients in the context of using neural networks, together with hand-crafted textual features to better understand what non-linear models are evaluating with regards to the features they learn to use.

Based on our analyses, we found it necessary to use both features that were generated on the reference and the summary text separately, as well as features that were constructed using both texts simultaneously (e.g., the vocabulary overlap). Our feature importance analyses highlighted interesting relationships between the ReaderBench linguistic features and the target variable. Both the Neural Network and the Random Forest models indicate that the semantic similarity between the source text and the summary is an important criterion when scoring the main idea coverage. In addition, the usage of a similar vocabulary to the source text leads to an increase in a summary's score. The evaluations on what the neural model emphasized during inferencing through Integrated Gradients can provide insights into how humans evaluate summaries.

There are several limitations to the proposed method. First of all, the size of the corpus may explain the lower results for the BERT architecture in comparison to the algorithm that uses textual complexity indices because large-scale Deep Learning models, like BERT, benefit from having access to more data during training. Limited datasets, such as the one used in this paper, may often lead to loss of generalization for deep models. Our choice of combining the seven human rating criteria for test set selection offers a proxy towards the holistic view humans develop while evaluating a summary; however, the limited number of data points may have introduced biases. Finally, our method for analyzing dataset-level importance of the different features offers some insight into the mechanisms of the neural network; however, Integrated Gradients is usually used on a sample-by-sample basis. For a certain sample, Integrated Gradients provides an indication as to which sample features are more important, by looking at the gradients that are propagated backward through the network from the loss function. The estimated feature importance is closely tied to the internal mechanisms of the model because the network is updated constantly during training through gradients. Nevertheless, averaging these gradients over the entire dataset can result in certain model behaviors being masked because they are less frequent.

Future avenues of research include the exploration of ways of integrating human domain knowledge to build a model that more closely resembles what humans focus on while evaluating summaries. Our approach considered analyzing the importance of linguistic features after training. The integration of human evaluator preferences into

the system could help increase the confidence that tutors have in such systems. One possibility consists of positively weighting features that human evaluators deem most relevant when evaluating a summary, thus encouraging the model to focus on them, while still ensuring the freedom of finding unexpected feature interactions.

With a normalized mean absolute error of 0.13, our results indicate that the best model is capable of matching human evaluations with an average deviation of only 13%. This error rate appears reasonable, if not exceptional. However, the real issue with automated summary scoring systems is whether they are useful to the end-users, namely to students and their teachers. For this, future studies in real-world settings will be necessary to provide a better assessment of the impact of the system. Our approach of performing a post-training analysis in order to identify the features that the model focuses on can help build confidence in the generated scores, through correlating these with human preconceptions. This process can help identify both possible biases in how the scores are assigned, as well as better inform the development of automated summary scoring systems in general through better feature engineering.

Acknowledgments. The work was funded by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number TE 70 *PN-III-P1-1.1-TE-2019-2209*, ATES – “Automated Text Evaluation and Simplification”. This research was also supported in part by the Institute of Education Sciences (R305A190063) and the Office of Naval Research (N00014-17-1-2300 and N00014-19-1-2424). The opinions expressed are those of the authors and do not represent views of the IES or ONR.

References

1. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018). arXiv preprint: [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
2. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
3. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. Ser. B (Methodol.)* **58**(1), 267–288 (1996)
4. Roscoe, R.D., Varner, L.K., Crossley, S.A., McNamara, D.S.: Developing pedagogically-guided algorithms for intelligent writing feedback. *Int. J. Learn. Technol.* **25**, **8**(4), 362–381 (2013)
5. Attali, Y., Burstein, J.: Automated essay scoring with e-rater V.2.0. In: Annual Meeting of the International Association for Educational Assessment, p. 23. Association for Educational Assessment, Philadelphia (2004)
6. Tay, Y., Phan, M.C., Tuan, L.A., Hui, S.C.: SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring. In: Thirty-Second AAAI Conference on Artificial Intelligence. AAAI, New Orleans (2018)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
8. Alikaniotis, D., Yannakoudakis, H., Rei, M.: Automatic text scoring using neural networks (2016). arXiv preprint: [arXiv:1606.04289](https://arxiv.org/abs/1606.04289)
9. Taghipour, K., Ng, H.T.: A neural approach to automated essay scoring. In: EMLP, pp. 1882–1891. ACL, Austin (2016)
10. Jin, C., He, B., Hui, K., Sun, L.: TDNN: a two-stage deep neural network for prompt-independent automated essay scoring. In: 56th Annual Meeting of the ACL Vol. 1: Long Papers, pp. 1088–1097. ACL, Melbourne (2018)

11. Dascalu, M., Dessus, P., Bianco, M., Trausan-Matu, S., Nardy, A.: Mining texts, learner productions and strategies with ReaderBench. In: Peña-Ayala, A. (ed.) *Educational Data Mining: Applications and Trends*, pp. 345–377. Springer, Cham (2014)
12. Dascalu, M., McNamara, D.S., Trausan-Matu, S., Allen, L.K.: Cohesion network analysis of CSCL participation. *Behav. Res. Methods* **50**(2), 604–619 (2018)
13. Ramos, J.: Using TF-IDF to determine word relevance in document queries. In: *1st Instructional Conference on Machine Learning*, vol. 242, pp. 133–142. ACM, Piscataway (2003)
14. Craney, T.A., Surlles, J.G.: Model-dependent variance inflation factor cutoff values. *Qual. Eng.* **14**(3), 391–403 (2002)
15. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814 (2010)
16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015). arXiv preprint: [arXiv:1502.03167](https://arxiv.org/abs/1502.03167)
17. Dahl, G.E., Sainath, T.N., Hinton, G.E.: Improving deep neural networks for LVCSR using rectified linear units and dropout. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8609–8613. IEEE, Vancouver (2013)
18. Smith, L.N.: A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay (2018). arXiv preprint: [arXiv:1803.09820](https://arxiv.org/abs/1803.09820)
19. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks (2017). arXiv preprint: [arXiv:1703.01365](https://arxiv.org/abs/1703.01365)



Automated Paraphrase Quality Assessment Using Recurrent Neural Networks and Language Models

Bogdan Nicula¹, Mihai Dascalu^{1,2(✉)}, Natalie Newton³, Ellen Orcutt⁴,
and Danielle S. McNamara³

¹ University Politehnica of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania
bogdan.nicula@stud.acs.upb.ro, mihai.dascalu@upb.ro

² Academy of Romanian Scientists, Str. Ilfov, Nr. 3, 050044 Bucharest, Romania

³ Department of Psychology, Arizona State University, P.O. Box 871104,
Tempe, AZ 85287, USA

{nnnewton, dsmcnama}@asu.edu

⁴ Department of Educational Psychology, University of Minnesota, 56 East River Road,
Minneapolis, MN 55455, USA

orcut039@umn.edu

Abstract. The ability to automatically assess the quality of paraphrases can be very useful for facilitating literacy skills and providing timely feedback to learners. Our aim is twofold: a) to automatically evaluate the quality of paraphrases across four dimensions: lexical similarity, syntactic similarity, semantic similarity and paraphrase quality, and b) to assess how well models trained for this task generalize. The task is modeled as a classification problem and three different methods are explored: a) manual feature extraction combined with an Extra Trees model, b) GloVe embeddings and a Siamese neural network, and c) using a pretrained BERT model fine-tuned on our task. Starting from a dataset of 1998 paraphrases from the User Language Paraphrase Corpus (ULPC), we explore how the three models trained on the ULPC dataset generalize when applied on a separate, small paraphrase corpus based on children inputs. The best out-of-the-box generalization performance is obtained by the Extra Trees model with at least 75% average F1-scores for the three similarity dimensions. We also show that the Siamese neural network and BERT models can obtain an improvement of at least 5% after fine-tuning across all dimensions.

Keywords: Paraphrase quality assessment · Natural language processing · Recurrent neural networks · Language models

1 Introduction

A paraphrase is a restatement, generated with different words, of the meaning of a text, generally with the aim of clarifying a sentence or a small group of sentences. Paraphrasing is useful for a number of purposes and applications. For example, in Natural Language

Generation, automated paraphrases are a method to increase diversity of generated text [1] and recognition of queries [2]. By contrast, our focus is on developing algorithms to assess the quality of human-generated paraphrases in order to provide feedback to students who are learning how to paraphrase more effectively and efficiently. Encouraging readers to transform a source text into more familiar words and phrases helps the reader to better understand the text by activating relevant prior knowledge. Learning to paraphrase facilitates both reading comprehension and writing ability, particularly for less skilled readers and writers [3–5]. Thus, paraphrase assessment is used in Intelligent Tutoring Systems aimed at improving reading and writing.

Our overarching objective is to develop feedback for a new version of iSTART (Interactive Strategy Training for Active Reading and Thinking; [6]), called iSTART-Early for young developing readers (ages 9–11). iSTART provides adaptive instruction and practice to use comprehension strategies (e.g., elaboration, bridging), while self-explaining and reading science texts to improve low-knowledge and less skilled readers' comprehension of challenging texts and performance in science courses.

The aim of this work is to assess the generalization capability of these models. First, we analyze the performance obtained by an Extra Trees model, a Siamese neural network model [7], and a BERT-based model [8] when trained on the ULPC dataset and evaluated on a different dataset. Second, we assess the importance of fine-tuning in improving results for the Siamese neural network and BERT models.

2 Related Work

One of the most well-known datasets for paraphrase identification is the Microsoft Research Paraphrase Corpus (MSRP) [9]. Given its relatively small size (5801 sentence pairs out of which 66.5% are positive examples), some of the best results on this dataset were obtained by small models. One example consists of using SWEMs (Simple-Word-Embedding-Models) [10], which rely on aggregating word embeddings via simple pooling operations (e.g., max pooling, average pooling). Another successful approach by Ji and Eisenstein [11] was to use a combination of fine-grained overlap features (e.g., unigram, bigram and dependency relation overlap metrics) and latent sentence-level features extracted using matrix factorization and a term weighting approach based on KL-divergence, called TF-KLD.

At the opposite end of the spectrum is the Quora Question Pair dataset (QQP), which consists of 400,000 question pairs with a binary annotation for paraphrasing. This dataset represents a good fit for data-hungry deep learning NLP models. A significant number of the current top performing models are based on the highly successful Bidirectional Encoder Representations from Transformers (BERT) model. Some approaches focus on reducing the size of the BERT model, while extracting the maximum performance from it [12], while others introduce innovative masking techniques for improving BERT performance [13]. Lastly, there are also models with similarly good performance that do not rely on BERT at all [14] as they create a custom neural network, that is considerably faster and uses much fewer parameters than classic BERT models with GloVe [15] word embeddings.

Despite the differences in style and content quality, both types of datasets (i.e., MSRP vs. QQP) share one shortcoming: they provide very little information regarding the *quality* of the paraphrase, as they solely indicate whether a given pair of sentences constitute paraphrases of one another. To our knowledge, the sole dataset that includes rubric scores regarding quality is the User Language Paraphrase Corpus (ULPC) [16], which scores paraphrases on 10 aspects using a point range from 1 to 6. We leverage this corpus in order to develop and test algorithms to assess paraphrase quality, and then to test the far transfer of these algorithms to paraphrases generated by young developing readers ages 9–11.

3 Method

3.1 Corpus

Two datasets were used as part of this work: the ULPC dataset consisting of 1998 source text – paraphrase pairs, and one smaller dataset, containing 115 paraphrases generated by children aged 9–11. The two datasets will be referred to as ULPC and the children dataset. The ULPC dataset consists of source texts – paraphrase pairs that were extracted from the input that users provided for the iSTART intelligent tutoring system (ITS). The children dataset is composed of paraphrase responses from a group of 13 3rd and 4th grade children participating in a summer school program. Notably, all students participants were English Language Learners. The paraphrase – sentence pairs in both datasets were scored in terms of the following four dimensions: semantic similarity, syntactic similarity, lexical similarity, and paraphrase quality.

For the ULPC dataset, the raters assigned scores ranging between 1 and 6 for each dimension. The four dimensions were then categorized into binary (1.00–3.49 vs 3.5–6.00), or tripartite (1.00–2.66, 2.67–4.33, 4.33–6.00) evaluations. For the children dataset, the four dimensions were originally scored on a binary system, except for paraphrase quality, which was scored on a tripartite scale. In order to have the same approach for both datasets, the problem was modeled as binary classification for semantic, syntactic and lexical similarity, whereas a tripartite classification was used for paraphrase quality.

3.2 Classification Models

Three different classification models were used for these experiments: An Extra Trees model combined with manually engineered features (ET), a Siamese neural (SN) network, and a BERT-based model. Out of the many possible options, these three alternatives were chosen to establish a strong baseline comparing systems relying heavily on manually engineered features versus deep learning systems, as well as lightweight (SN) versus resource intensive (BERT) models.

For the ET model, several types of features based on the sentence-paraphrase pairs were used: a) complexity indices related to surface, lexical, syntactic, and semantic properties of the texts were computed using the ReaderBench framework [17]; b) complexity indices outlining text cohesion were computed on the concatenation of the source text

and the paraphrase, and c) Levenshtein distance [18] at word level between the source and paraphrase, as well as simple overlap indices for both words and part-of-speech (POS) tokens. For the ReaderBench complexity features, the difference between the value of the same index computed for both source and paraphrase was used. The resulting 2368 features were filtered in order to eliminate constant values and features with high intercorrelation. The filtered features were used as input for several ML classifiers from the SciKit Learn library [19] to predict one of the targeted four dimensions. In all four cases, the best results were obtained by the Extra Trees model.

For the Siamese neural network [7], Bidirectional Long Short-Term Memory (BiLSTM; [20]) layers were used with pretrained 300-dimensional GloVe or Word2Vec [21] word embeddings at the entry point in the architecture. Both the source sentence and the paraphrase were converted into an array of indices, each index pointing to an embedding representing the meaning of the corresponding word in the text. This representation was then processed separately (once for the source, once for the paraphrase) by the BiLSTM layers, and after a set of pooling operations, the two processed results were combined, and a prediction was made. The results are reported for the model using GloVe embeddings, as that model obtained a better performance.

For the BERT-based model, a pretrained version of BERT from the Huggingface library [22] was considered. In terms of the architecture, the source and paraphrase texts were passed as a text pair to the pretrained BERT model, delimited by a special BERT separator. The combined input was truncated if longer than a threshold of 75 words, and then converted into embeddings and passed through the BERT pipeline. The output of the BERT model went through a Dropout layer with a conservative $p = 0.2$ dropout rate, and then a fully connected (FC) layer was used to make the final prediction. Different learning rates for the BERT model ($lr_BERT = 1e-5$) and the FC layers ($lr_FC = 2e-2$) were considered to make the fine tuning feasible.

4 Results and Discussions

Our first experiment involved examining accuracy of the models on the children dataset. ET, SN and BERT models were trained on the entire ULPC dataset (training+validation+testing) and tested on the entire children dataset (115 paraphrase pairs). The tripartite split (into low 1–2, mid 3–4, and high 5–6) was used for the Paraphrase Quality dimension, as it was available for both datasets.

The results provided in Tables 1 and 2 indicate that the Extra Trees model obtained the best results in 3 out of 4 cases (in terms of average weighted F1-score). On the three binary dimensions (semantic, syntactic, lexical similarity), the ET model consistently outperformed the SN and BERT models. Overall, the high performance of extra trees is beneficial, given the interpretability of the models relying on linguistic features reflective of writing style and on semantic relatedness between the paraphrase and the source text. The interpretability is beneficial because the features can guide feedback.

The poor overall performance obtained on the Paraphrase Quality task might be caused by the fact that the children paraphrases are more difficult to be split up into three classes, given the simplicity of the text (i.e., most answers are either fair paraphrase attempts or not paraphrases at all, and there is less room to be vague).

Table 1. Performance on ULPC models tested on Children dataset (Semantic similarity, Syntactic similarity and Lexical similarity).

Dimension	Model	Support low	Support high	Low F1	High F1	Avg F1
Semantic similarity	ET	22	93	.706	.916	.875
	SN	22	93	.371	.725	.657
	BERT	22	93	.575	.802	.758
Syntactic similarity	ET	35	80	.688	.776	.749
	SN	35	80	.444	.327	.362
	BERT	35	80	.530	.367	.416
Lexical similarity	ET	31	84	.806	.929	.895
	SN	31	84	.422	.629	.573
	BERT	31	84	.689	.811	.778

Table 2. Performance on ULPC paraphrase quality models tested on children dataset.

Model	Support low	Support mid	Support high	Low F1	Mid F1	High F1	Avg F1
ET	24	60	31	.610	.708	.244	.562
SN	24	60	31	.333	.337	.205	.300
BERT	24	60	31	.454	.712	.000	.466

In the second experiment we evaluated the benefits of fine-tuning for the Siamese Network and BERT-based models. The models trained on the ULPC dataset were trained for a small number of epochs on 67 pairs from the children dataset and tested on the remaining 48 pairs. All examples containing the same source text were added to either the test or the training set, but not both. Because of the nature of the dataset (i.e., for a given source text there are a variable number of paraphrases), the children dataset could not be split into equal halves. In all the cases, the slightly larger half was used for training and the smaller one for validation.

The F1 scores for all the classes, as well as weighted average of the F1 scores, are reported in Table 3 and Table 4. This metric was computed for predictions made by a) the initial pretrained models (e.g., SN and BERT), and b) the pretrained models that were fine-tuned for a short number of epochs.

When looking at the individual F1 scores we tend to see improvements after fine-tuning in most cases. One notable exception is the High class for Paraphrase Quality for which both models have difficulties without pretraining, and BERT does not manage to obtain an F1 of over 0 even after fine-tuning, despite having non-zero scores on the training set.

Table 3. Results obtained after fine-tuning on the children dataset (Semantic similarity, Syntactic similarity and Lexical similarity).

Dimension	Model	Pretrained			Finetuned		
		Low F1	High F1	Weighted Avg	Low F1	High F1	Weighted Avg
Semantic similarity	SN	.303	.635	.572	.348	.795	.711
	BERT	.545	.761	.720	.5	.833	.770
Syntactic similarity	SN	.370	.190	.238	.378	.610	.547
	BERT	.464	.25	.307	.619	.703	.680
Lexical similarity	SN	.457	.689	.631	.435	.822	.725
	BERT	.666	.800	.766	.733	.878	.841

When comparing the weighted F1 scores an improvement of at least .05 can be observed after fine-tuning. In the 2-class setting, the most dramatic improvements were made for the Syntactic similarity class. On this dimension, the distributions for the children dataset were almost inverted versions of the ULPC distribution. This could mean that the model had learned useful features in the initial training phase, but it relied on a bad estimate of the distribution for the classes.

Table 4. Results obtained after fine-tuning on the children dataset (Paraphrase Quality).

Model	Pretrained				Finetuned			
	Low F1	Mid F1	High F1	Weighted Avg	Low F1	Mid F1	High F1	Weighted Avg
SN	.359	.432	.200	.353	.385	.696	.500	.578
BERT	.461	.736	.000	.479	.476	.721	.000	.474

5 Conclusions

The aim of this study was to develop three ML algorithms to assess paraphrase quality leveraging the ULPC dataset and to evaluate how well these models generalize when presented with a new dataset of paraphrases generated by children. When tested on the children dataset, the Extra Trees model obtained the best results. The SN and BERT models also provided improved results after fine-tuning on the children dataset.

In the first generalization task, the Extra Trees model was shown to generalize better. This indicates that the manually extracted features might have a more general character than the ones automatically extracted by Siamese Networks or BERT-base models, making them more robust on new data. For the Semantic, Syntactic and Lexical similarity

dimensions, the Extra Trees model generalized well on the children dataset. However, the results were slightly worse for the Paraphrase Quality dimension. This could indicate that it is more difficult to meaningfully separate poor, satisfactory, and good paraphrases for children, or it could indicate an issue with how the dataset was annotated (e.g., raters had difficulties separating the 3 levels of the dimension). In this case, results could be improved by adding a larger paraphrase dataset with similar characteristics (short source and paraphrase sentences) for initially training the models, followed by a finetuning on the ULPC dataset.

Fine-tuning helped improve results in all cases with differences ranging from 0.05 to 0.20. The BERT model fared better than the SN model in the binary classification tasks, while it underperformed when classifying paraphrase quality. Its poor performance was caused by its difficulties in predicting the High class.

This study provides promising evidence that our approach can generate models that generalize to texts that differ in reading ease and to individuals who vary in age, reading skill, and language abilities. While more evidence is needed to further test this approach and these models, these models provide a strong starting point in iSTART-Early for providing automated feedback on paraphrase quality to young developing readers.

Acknowledgments. The work was funded by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number TE 70 *PN-III-P1-1.1-TE-2019-2209*, ATES – “Automated Text Evaluation and Simplification”. This research was also supported in part by the Institute of Education Sciences (R305A190063 and R305A190050) and the Office of Naval Research (N00014-17-1-2300 and N00014-19-1-2424). The opinions expressed are those of the authors and do not represent views of the IES or ONR.

References


1. Qian, L., Qiu, L., Zhang, W., Jiang, X., Yu, Y.: Exploring diverse expressions for paraphrase generation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3164–3173 (2019)
2. Fader, A., Zettlemoyer, L., Etzioni, O.: Open question answering over curated and extracted knowledge bases. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1156–1165 (2014)
3. McNamara, D.S.: SERT: self-explanation reading training. *Discourse Process.* **38**, 1–30 (2004)
4. McNamara, D.S., Ozuru, Y., Best, R., O’Reilly, T.: The 4-pronged comprehension strategy framework. In: *Reading Comprehension Strategies: Theories, Interventions, and Technologies*, pp. 465–496. Erlbaum, Mahwah (2007)
5. Hawes, K.: *Mastering Academic Writing: Write a Paraphrase Sentence*. University of Memphis, Memphis, TN (2003)
6. Jackson, G.T., McNamara, D.S.: Motivation and performance in a game-based intelligent tutoring system. *J. Educ. Psychol.* **105**(4), 1036 (2013)
7. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences (2014). arXiv, preprint: [arXiv:1404.2188](https://arxiv.org/abs/1404.2188)
8. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding (2018). arXiv preprint: [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)

9. Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Proceedings of the Third International Workshop on Paraphrasing (IWP2005) (2005)
10. Shen, D., et al.: Baseline needs more love: on simple word-embedding-based models and associated pooling mechanisms (2018). arXiv preprint: [arXiv:1805.09843](https://arxiv.org/abs/1805.09843)
11. Ji, Y., Eisenstein, J.: Discriminative improvements to distributional sentence similarity. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 891–896 (2013)
12. Jiao, X., et al.: TinyBERT: Distilling BERT for natural language understanding (2019). arXiv preprint: [arXiv:1909.10351](https://arxiv.org/abs/1909.10351)
13. Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: SpanBERT: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguist.* **8**, 64–77 (2020)
14. Yang, R., Zhang, J., Gao, X., Ji, F., Chen, H.: Simple and effective text matching with richer alignment features (2019). arXiv preprint: [arXiv:1908.00300](https://arxiv.org/abs/1908.00300)
15. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP 2014), vol. 14. ACL, Doha (2014)
16. McCarthy, P.M., McNamara, D.S.: The user-language paraphrase challenge (2008). Accessed 10 Jan 2008
17. Dascalu, M., Crossley, S.A., McNamara, D.S., Dessus, P., Trausan-Matu, S.: Please Reader-Bench this text: a multi-dimensional textual complexity assessment framework. In: Craig, S. (ed.) *Tutoring and Intelligent Tutoring Systems*, pp. 251–271. Nova Science Publishers Inc, Hauppauge (2018)
18. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Phys. Doklady* **10**(8), 707–710 (1965)
19. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representation in vector space. In: Workshop at ICLR, Scottsdale, AZ (2013)
22. Wolf, T., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45 (2020)

Groups, Teams, Social, Crowd and Communities



XGBoost and Deep Neural Network Comparison: The Case of Teams' Performance

Filippos Giannakas^(✉), Christos Troussas, Akrivi Krouska,
Cleo Sgouropoulou, and Ioannis Voyiatzis

Department of Informatics and Computer Engineering, University of West Attica,
Egaleo, P. O. 12243, Athens, Greece

{fgiannakas,ctrouss,akrouska,csgouro,voyageri}@uniwa.gr

Abstract. In the educational setting, working in teams is considered an essential collaborative activity where various biases exist that influence the prediction of teams performance. To tackle this issue, machine learning algorithms can be properly explored and utilized. In this context, the main objective of the current paper is to explore the ability of the eXtreme Gradient Boosting (XGBoost) algorithm and a Deep Neural Network (DNN) with 4 hidden layers to make predictions about the teams' performance. The major finding of the current paper is that shallow machine learning performed better learning and prediction results than the DNN. Specifically, the XGBoost learning accuracy was found to be 100% during teams learning and production phase, while its prediction accuracy was found to be 95.60% and 93.08%, respectively for the same phases. Similarly, the learning accuracy of the DNN was found to be 89.26% and 81.23%, while its prediction accuracy was found to be 80.50% and 77.36%, during the two phases.

Keywords: Deep Neural Network · Machine learning · Comparison · XGBoost · Adamax · Team performance

1 Introduction

Nowadays, Machine Learning (ML) gains applicability in various domains where different applications were developed for addressing e.g., sentiment analysis, image recognition, natural language processing, speech recognition, and others. Especially in the educational setting, various ML applications were developed and incorporated on Intelligent Tutoring Systems (ITS), that among others, support and increase learners' engagement, retention, motivation, towards improving their learning outcomes [12–14].

The learning process is becoming more challenging when learners are engaged in a team-based learning experience. This is because collaboration and communication activities among participants seem to significantly influence students'

learning development [9]. In this context, the knowledge about the team performance is considered an essential task that reflects teams' efficiency, expectations, and learning outcomes of their members [6].

However, the literature shows poor research efforts in predicting team performance in terms of quality [1]. Instead, the research community mainly focused on exploring participants' individual characteristics and analyze the factors that influence their performance, aiming at delivering punctual interventions, avoiding to interpret how the qualities that exist during team-working experience influence teams' performance as a whole. For instance, the work of [4] presents a ubiquitous guide-learning system for tracing and enhancing students' performance by observing and evaluating their active participation. The same situation was found in the work of [8], where learners' communication activity was observed for measuring learners' performance. In the same context, the authors in [2, 15] assessed the interaction activity of Computer-Supported Collaborative Learning (CSCL) environments for concluding about students' learning performance [2, 15].

Exploring the ability of ML to make predictions about the teams' performance, the literature has shown that this is at an initial stage. The only related work found in the literature is that of [11] and [10], where the Random Forest (RF) algorithm for assessing and predicting students' learning effectiveness. Specifically, in the work of [10] the learning accuracy of their ML model was found to be 90%, using little input collaborative data. Later, the same authors in [11] trained the same ML model on a bigger scale of input data and its learning accuracy was measured at 70%. Also, the precision accuracy of their model was also calculated and found to be 0.54% and 0.61% during the learning and production phase of the teams, while the recall metric was 0.76% and 0.82%. Another research work, but for predicting the performance of team's member is that of [16]. In this work a shallow ANN with one hidden layer was enabled to predict students' academic performance in two courses. The model was trained using little input samples and the prediction accuracy was found to be 98.3%, missing other comparison results.

Concluding, while many researchers explored various machine learning algorithms for making predictions, there are missing works for exploring the ability and compare the performance of a ML with a DNN model for making predictions about the teams' performance. In this context, the major contribution of the current paper is to explore the use of the XGBoost supervised gradient boosting decision tree and a gradient descent Deep Neural Network (DNN) with four hidden layers that enables Adamax optimizer model, in terms of learning and accuracy. Summarizing, the current paper intends to answer the following two research questions:

- RQ1: Can a binary classification model being shaped and used by a ML and a DNN model for predicting the teams' performance?
- RQ2: Which is the performance of the XGBoost and DNN model?

The rest of the paper is structured as follows. The next section presents the research design by discussing the structure of the models, the dataset used for

the training and prediction phase, as well as the evaluation of the models. The results and the discussion are summarized in Sect. 3. The last section concludes the paper and provides directions to future work.

2 Research Design

In order to be able to answer RQ1, working in teams and their performance is formulated and explored in terms of a supervised binary classification problem. This formation will further assist the evaluation and comparison of the XGBoost and DNN model as presented below.

2.1 XGBoost Model

Briefly, XGBoost is a gradient boosted decision trees algorithm [3], that enables advanced regularization ($L1$ & $L2$) for assisting model generalization and control overfitting (model learn from noise and inaccurate data), for producing better performance results. In general, gradient boosting machine learning technique produces an ensemble of weak prediction models in the form of typical decision trees. A decision tree model is structured by partitioning a set of features X into a set of T non-overlapping regions (nodes) ($R1$ to RT). Then, a prediction is generated for each region by calculating a simple constant model f as shown in Eq. 1.

$$f(x) = \sum_{j=1}^T w_j * I(x \in R_j) \quad (1)$$

2.2 DNN Model

The proposed DNN structure consists of 84 input neurons, whereas the four hidden layers have 64, 32, 16, and 8 neurons, respectively. The total parameters of the model are 8.193, that all considered trainable. The linear activation function of ReLu (Rectified Linear Unit) was used in the first layers, whereas for the output layer the Sigmoid (Uni-Polar) function was selected. Also, the “Adamax” adaptive learning rate algorithm for weights optimization [7], and the “Binary Cross-Entropy” loss function were enabled. The proposed structure of the DNN model was selected after conducting a test-based evaluation of the model’s performance by varying the layers and/or the nodes of its structure.

2.3 Evaluation Procedure

In order to evaluate both the XGBoost and the DNN models, different metrics that of the “confusion matrix” (row 1: TN-FP, row 2: FN-TP), and its related metrics for assessing accuracy, recall, and precision, were computed.

Additionally, “F1 score” and “AUC-ROC (Area Under The Curve - Receiver Operating Characteristics)” metrics were also calculated in order to compute the preciseness and the robustness of the model.

Table 1. Evaluation results

Model	Data	Learning	Prediction	Precision	Recall	F1 score	ROC-AUC
		Acc. (%)	Acc. (%)	Acc.			
XGBoost	Process	100	95.60	0.9778	0.8800	0.9220	0.9354
XGBoost	Product	100	93.08	0.8906	0.9344	0.9120	0.9315
DNN	Process	89.26	80.50	0.7049	0.7679	0.7350	0.8776
DNN	Product	81.23	77.36	0.7039	0.8451	0.7692	0.8558

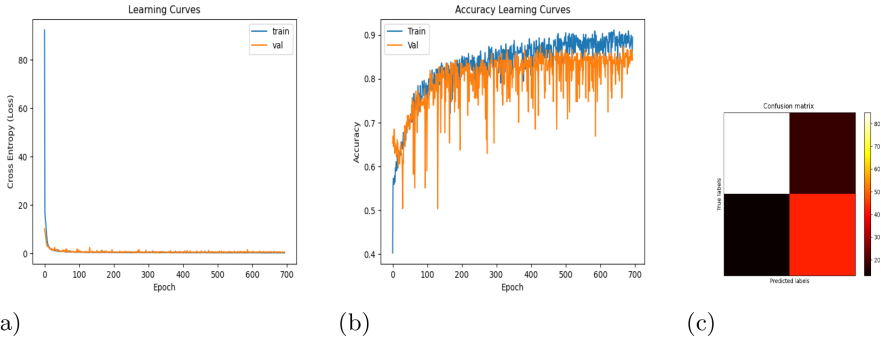


Fig. 1. DNN’s performance curves for the process data: (a) Cross-entropy (loss); (b) Learning accuracy; (c) Confusion matrix.

Both models were trained and evaluated using the same dataset of “Data for Software Engineering Teamwork Assessment in Education Setting Data Set (SETAP)”. The dataset contains records from different teams that attended the same software engineering course at San Francisco State University (USA), Fulda University (Germany), and Florida Atlantic University (USA), from 2012 to 2015 [11].

The SETAP project contains 30000 entries separated into two different phases that of process and product. During the process phase the learners educated how effectively they apply software engineering processes in a teamwork setting. On the contrary, in the product phase teams follow specific IT requirements for developing a software. The last column of the dataset classifies the teams’ performance into two classes [10].

3 Models’ Performance Results and Discussion

In order to answer the RQ2, performance analysis was conducted for both models by computing the relevant metrics, as discussed previously in Subsect. 2.3. Specifically, the evaluation results are summarized in Table 1. Observing the table, XGBoost algorithm succeeded better accuracy results compared to the DNN model. Specifically, it was found that when the XGBoost algorithm is used,

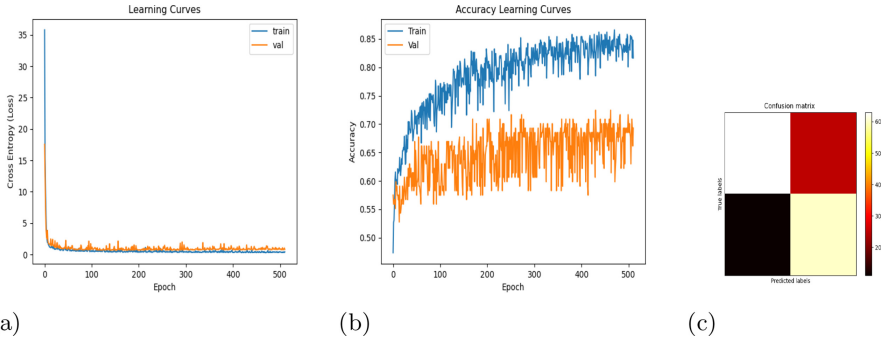


Fig. 2. DNN’s performance curves for the product data: (a) Cross-entropy (loss); (b) Learning accuracy; (c) Confusion matrix.

the learning and prediction accuracy was found to be at 100% and 95.60% during the process phase, whereas during the production phase the results were found to be 100% and 93.08%. Further, the precision, recall, F1 score, and ROC-AUC were found to be 0.9778, 0.8800, 0.9263, and 0.9354 for the process phase, while during the production phase the results for the same metrics were computed at 0.8906, 0.9344, 0.9120, and 0.9315.

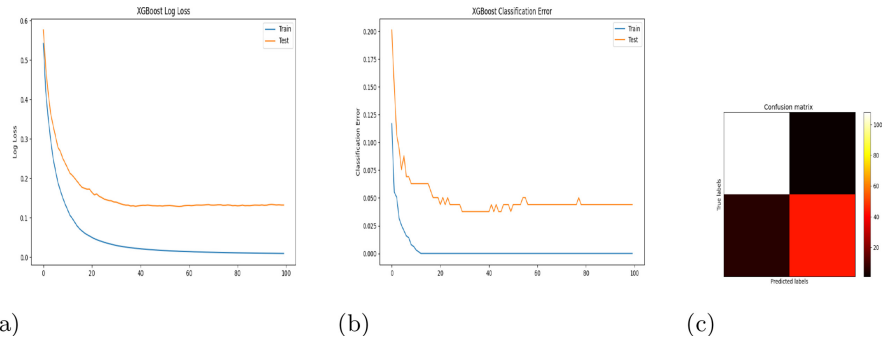


Fig. 3. XGBoost’s performance curves for the process data: (a) Loss; (b) Classification error; (c) Confusion matrix.

Similarly, observing the result for the DNN model, when the data from the process phase were used, its learning and prediction accuracy was found to be at 89.26% and 80.50%, respectively. The rest of the metrics that of precision, recall, F1 score, and ROC-AUC were found to be 0.7049, 0.7679, 0.7350, and 0.8776. Same, for the product data, the DNN model succeeded 81.23% and 77.36% learning and prediction accuracy, whereas the rest of the metrics were computed at 0.7059, 0.8451, 0.7692, and 0.8558.

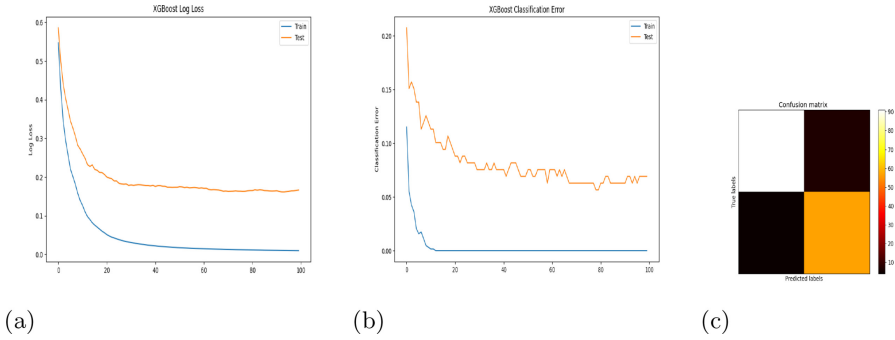


Fig. 4. XGBoost’s performance curves for the product data: (a) Loss; (b) Classification error; (c) Confusion matrix.

Further, XGBoost loss error, classification error, and confusion matrix during the process and product phases are shown in Figs. 3 and 4. Also, DNN model’s cross-entropy(loss), learning accuracy and confusion matrix curves, for the two phases are shown in Figs. 2 and 1.

To sum up, for shaping knowledge about the teams’ performance in terms of binary classification, the shallow machine learning model that enables the XGBoost algorithm performed outstanding performance results compared to the DNN model. However, we can not safely argue that the proposed ML model that enables the XGBoost algorithm is better than the DNN one since the performance of the models maybe also influenced by the quality of the input data [5].

4 Conclusion

The paper explores and compares the performance of the XGBoost and the DNN model that enables Adamax optimizer and Binary Cross-Entropy loss function with four hidden layers were explored and evaluated for predicting the teams’ performance. The results showed that the XGboost model outperformed the DNN by succeeding 100% learning accuracy, during the process and production phase of the teams, while its prediction accuracy was found to be 95.60% and 93.08%, for same phases. The overall learning performance of the DNN model was found to be 89.42% and 81.23%, while the prediction accuracy computed at 80.50% and 77.36%, respectively for the same phases.

However, the low results of the DNN model underline the importance of conducting further research in order to explore if other parameters, such as the amount and the quality of data, can improve its prediction performance. This will assist further the deployment of learning/assisting tools that will encapsulate machine learning, which will help further the development of more intelligent tutoring systems.

References

1. Alshareet, O., Itradat, A., Doush, I.A., Quttoum, A.: Incorporation of ISO 25010 with machine learning to develop a novel quality in use prediction system (QIUPS). *Int. J. Syst. Assur. Eng. Manag.* **9**(2), 344–353 (2018). <https://doi.org/10.1007/s13198-017-0649-x>
2. Aouine, A., Mahdaoui, L., Moccozet, L.: A workflow-based solution to support the assessment of collaborative activities in e-learning. *Int. J. Inf. Learn. Technol.* **36**, 124–156 (2019)
3. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system, pp. 785–794 (2016)
4. Chin, K.Y., Ko-Fong, L., Chen, Y.L.: Effects of a ubiquitous guide-learning system on cultural heritage course students' performance and motivation. *IEEE Trans. Learn. Technol.* **13**, 52–62 (2019)
5. Devan, P., Khare, N.: An efficient XGBoost–DNN-based classification model for network intrusion detection system. *Neural Comput. Appl.* **32**(16), 12499–12514 (2020). <https://doi.org/10.1007/s00521-020-04708-x>
6. Dunnette, M.D., Fleishman, E.A.: *Human Performance and Productivity: Volumes 1, 2, and 3.* Psychology Press, Taylor and Francis (2014)
7. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: *Proceedings of International Conference on Learning Representations* (2015)
8. Mengoni, P., Milani, A., Li, Y.: Clustering students interactions in elearning systems for group elicitation. In: Gervasi, O., et al. (eds.) *ICCSA 2018. LNCS*, vol. 10962, pp. 398–413. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95168-3_27
9. O'Donnell, A.M., Hmelo-Silver, C.E., Erkens, G.: *Collaborative Learning, Reasoning, and Technology.* Routledge, Milton Park (2013)
10. Petkovic, D., et al.: SETAP: software engineering teamwork assessment and prediction using machine learning. In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pp. 1–8. IEEE (2014)
11. Petkovic, D., et al.: Using the random forest classifier to assess and predict student learning of software engineering teamwork. In: *2016 IEEE Frontiers in Education Conference (FIE)*, pp. 1–7. IEEE (2016). <https://archive.ics.uci.edu/ml/datasets/Data+for+Software+Engineering+Teamwork+Assessment+in+Education+Setting>
12. Troussas, C., Giannakas, F., Sgouropoulou, C., Voyiatzis, I.: Collaborative activities recommendation based on students' collaborative learning styles using ANN and WSM. *Interact. Learning Environ.* 1–14. Taylor and Francis
13. Troussas, C., Krouska, A., Giannakas, F., Sgouropoulou, C., Voyiatzis, I.: Automated reasoning of learners' cognitive states using classification analysis, pp. 103–106 (2020)
14. Troussas, C., Krouska, A., Giannakas, F., Sgouropoulou, C., Voyiatzis, I.: Redesigning teaching strategies through an information filtering system, pp. 111–114 (2020)
15. Wang, C., Fang, T., Gu, Y.: Learning performance and behavioral patterns of online collaborative learning: impact of cognitive load and affordances of different multimedia. *Comput. Educ.* **143**, 103683 (2020)
16. Zacharis, N.Z.: Predicting student academic performance in blended learning using artificial neural networks. *Int. J. Artif. Intell. Appl.* **7**(5), 17–29 (2016)



Using Graph Embedding to Monitor Communities of Learners

Fabio Gasparetti¹, Filippo Sciarrone^{1(✉)}, and Marco Temperini²

¹ Department of Engineering, ROMA TRE University, Via della Vasca navale, 79, Rome, Italy

{sciarro,gaspare}@ing.uniroma3.it

² Department of Computer, Control, and Management Engineering, Sapienza University of Rome, Viale Ariosto 25, Rome, Italy
marte@diag.uniroma1.it

Abstract. How to keep track of the learning process of a community of learners is a problem whose resolution requires accurate assessment tools and appropriate teaching and learning strategies. Peer Assessment is a standard didactic strategy which requires students in a course to correct their peers' assignments. Since the representation of a community, even a large one, of students, is based on directed graphs, it is difficult to follow its whole dynamics. In this paper, we investigate the possibility of using two machine learning techniques: Graph Embeddings, and Principal Component Analysis, to represent a students' communities by points in a 2D space, in order to have valuable and understandable information on the dynamics of the group. For this purpose we present a case study based on three real Peer Assessment sessions. The first results are encouraging.

Keywords: Graph Embedding · Communities of learning · Peer Assessment

1 Motivations and Goals

In the knowledge Society era, the individual is pushed to promptly keep updated on a particular area throughout her working life. Traditional learning paths are rarely designed with this goal in mind, so there is often a need of alternative methods to traditional training, which take into account new technologies, in particular when a teacher has to manage a huge number of students. To this aim, Technology Enhanced Learning (TEL) gives the possibility for using new learning/teaching strategies, supported by tools and algorithms that would be impossible to use without the technological and logical advancements of Computer Science. In this work we investigate the use of a Machine Learning (ML) technique, called *Graph Embedding* (GE) [2], to support the representation and analysis of the activities of a learning community, where the members of the

community interact, give hints each other, assess their peers and work in a cooperative way. These interactions, that we might summarize under the term *dynamics*, provide significant advantages to the student's learning flow, with respect to an individual approach, where the interaction takes place only with the teacher and the networked platform. Given that the classic graph based representations of the interactions and relationships among the members of a community are powerful, our research conjecture is that GE may help overcoming some of the limitations imposed by such approach. Indeed graphs, are difficult to be dynamically analyzed, mainly for computational reasons, especially when we compare two different learning states, and dynamics, of a whole large community [7–9]. In this study, we apply the GE to different Peer Assessment (PA) sessions, to generate status changes in the community. The first Research Question (RQ) of our investigation concerns the possibility to represent a community of students using PA, through a vector, which can be represented as a point in a low-dimensional 2D space. The second RQ concerns the possibility to have a fruitful vision of the community dynamics through the compression and representation mechanism offered by GE. In order to verify the RQs, this work proposes a case study based on the analysis of a real classroom of students belonging to a computer science high school and composed by 25 students. Three PA sessions were performed to evaluate Open Answers on specific topics of interest. The paper is structured as follows. Section 2 shows some important related works from the literature. Section 3 briefly introduces the GE technique to provide a background. In Sect. 4 the study of an application case and discuss the research questions. Finally the last section discusses future work and conclusions.

2 Related Work

Several works deal with the use of ML algorithms on learning processes in learner communities, to extract useful information and improve learning [3, 12]. Such approach is supported by availability of large, ever increasing, amounts of data generated by nowadays learning systems, about the interactions among learners. Deep Learning is a set of techniques based on Neural Network architectures which are able to identify and extract complex relationships from data. Some of these complex architectures have also been devised for compressing data into lower dimension spaces (e.g. the curse of dimensionality problem). Our work uses the GE technique, to compress the graph representing a community of learners into a low-dimensional vector and, eventually, to a point in a 2D space, in order to monitor the PA process. In fact, the GE algorithms can be effective in converting high-dimensional sparse graphs into low-dimensional, dense and continuous vector spaces, preserving maximally the graph structure properties. Our work refers to a community of students, even large (e.g. MOOCs), where the learning strategy of PA is used. In this context, the community can be represented as a multigraph where a node V_i represents a student and an edge between two nodes V_i and V_j represents the grade, that the student s_i gave the student s_j . A student can grades the same peer more than once. The

applications of GE to PA sessions, aiming to capture valuable information on the learning progress, seems attractive, and to our knowledge no such applications are studied so far. On the other hand, several researches considered textual relationships and other characteristics of the group of learners. The *TransConv* algorithm uses GE to incorporate textual interactions between pair of users to improve representation of learning of both users and relationships [5]. This algorithm represents a structural embedding approach using relation hyperplanes, where every relationship can be viewed as a translation of users in the embedding space, using textual communications among users as well. In [10], the authors conduct a systematic investigation of the problem of course concept extraction for MOOCs. They propose to learn latent representations for candidate concepts via an embedding-based method and developed a graph-based propagation algorithm to rank the candidate concepts based on the learned representations. In [11], the authors present the *GEval* system, a modular and extensible evaluation framework for graph embedding techniques. This framework is useful to compare different approaches of Graph Embedding both for developers of new embedding techniques and consumers of these techniques in choosing the best approach according to the task(s) the vectors will be used for.

3 Background

Given a directed graph $G = \{V, E\}$, GE is a deep learning technique that is used to transform nodes, edges, and their features into a vector space with a low dimension whilst maximally preserving properties like graph structure and information. Embedding should capture the graph topology, the node-to-node relationships, and other relevant information about graphs, subgraphs, and nodes. Consequently one of the greatest advantages is that vector operations are easier to computer in comparison with operations on graphs. To this aim, the literature proposes two groups of embeddings [1,2]. They are (i) *Node Embeddings*, where each node is encoded with its own vector representation. One could use this embedding when performing visualization or prediction on the node level, e.g. visualization of nodes in a 2D plane, or prediction of new connections based on nodes similarities; (ii) *Graph Embeddings* where the whole graph is represented by a unique vector. These embeddings are used both to make predictions on the graph level and to compare or visualize the whole graphs. In our work we used the GE approach. However, the GE approach needs to satisfy some important requirements, such as: 1) the embeddings ought to encode the relevant properties of the graph i.e., node connections and neighborhood, and in general the graph topology; 2) the prediction or visualization quality should depend on embeddings' quality; 3) network's size should not delay the embedding process: graphs are usually large, and an embedding approach needs to be efficient. An essential challenge is the accurate estimation of the dimension of the embedding space. In fact, longer embeddings preserve more information while they induce higher time and space complexity than shorter ones. Users need to make a trade-off based on the requirements. In our approach we considered a fast algorithm which relies

on spectral features of the graph. In particular, it uses a spectral decomposition of graph *Laplacian* to perform graph classification [6]. Each eigenvalue of the Laplacian is evaluated and physically interpreted as the energy level of a stable configuration of the nodes in the embedding space. The lower the energy, the stabler the configuration. In our experiments, we set the size of the embedding space to 12, which has been tuned accordingly to the dimension of the considered graphs. In particular, we generated several configurations similar to the ones obtained in the experiments, differently perturbed by new nodes and altered edges' values. By the F-measure, the optimal size of the embeddings space was estimated, in order to cluster together very similar graphs. This was done relying solely on the distance measure calculated on two vectors in the embeddings space. In order to examine and interpret the vectors in the embeddings space, we adopted the *Principal Component Analysis* (PCA) [4] technique, which is one of the most popular dimensionality reduction techniques. By setting to 2 the required output dimensions, we were able to plot the embeddings in 2D diagrams.

4 The Case Study

As the experimental sample we used a classroom of 25 students attending the third year of an industrial technical high school of computer science. We run three PA sessions, collected in the set $S = \{s_1, s_2, s_3\}$. In each session s_i , the peers had to accomplish a task t_i , consisting in an open answer question. Consequently, we gathered a set $T = \{t_1, t_2, t_3\}$ of session-related tasks. The session tasks had to be accomplished in a pre-stated time, according to some predefined and explained assessment criteria, with a grading scale set in the range $R = [1, 10]$. The open answers questions were about a microprocessor architecture. For each PA session two kinds of community representations were built. The first representation was comprised of 3 directed graphs, G_1, G_2 and G_3 , corresponding to the PA session. In each G_i graph, each node V_i represents the student st_i while an edge E_{ij} represents the way a student st_i graded student st_j , with edge weight being the grade itself. The second kind of graph is an incremental one, where the graph G_{i+1} is the graph G_i generated by the s_i session, merged with the edges generated by the s_{i+1} session. The resulting graph is a directed multigraph, being more grades from st_i to st_j possible. For each session a further graph with the teacher's grades was built, useful to compare the peers' grades with the teachers' ones, and determine how each student had been good at assessing. Data from the 3 PA sessions are shown in Table 1; for each session, we computed 1) the mean \bar{x}_s and the standard deviation \bar{s}_s of the students grades, 2) the mean \bar{x}_t and the standard deviation \bar{s}_t of the teacher's grades. From this data we can draw the conclusion that the two distributions, assuming two Gaussian distributions, differ on average in two cases by 1 while in another case by 1.5. As might have been expected, students' grades are generally higher than teacher's ones, while the standard deviations are similar.

As explained in the previous section, we embedded all the graphs in order to represent them as a simple vector in a 2D space. In particular we computed

Table 1. The evaluation data for each PA session.

PA session	\bar{x}_s	\bar{s}_s	\bar{x}_t	\bar{s}_t
S_1	6.25	1.77	5.24	1.51
S_2	6.27	1.1	4.96	2.1
S_3	6.09	1.4	5.42	2.4

12 embeddings, one for each generated graph, always maintaining the duality *student graph* (real) Vs. *teacher graph*, as shown in Table 2. The embeddings are shown in Fig. 1(a) and Fig. 1(b). Moreover, the vector representation allows to compute the relative euclidean distance among them. In Fig. 2 are shown the distances in all the two types of graphs.

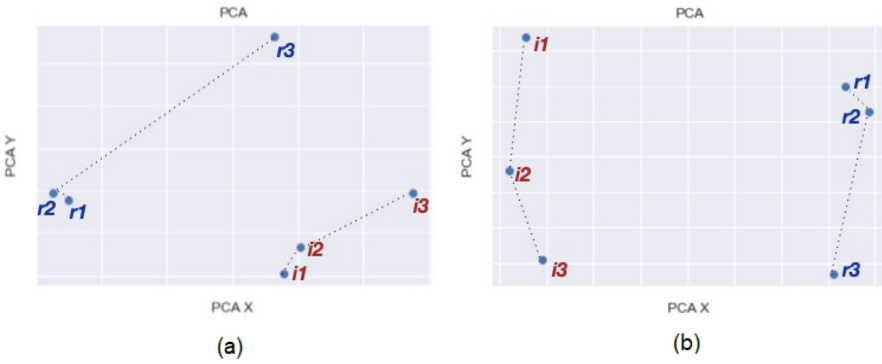


Fig. 1. The 2D representation of the two embeddings. In Figure (a), each point represents a graph, r_i as a real graph Vs. i_i as the teacher’s graph. Figure (b) is the representation of the multigraphs generated merging a graph g_i with the graph g_{i+1}

Now we try to answer the two RQs illustrated in the Sect. 1. Regarding the RQ_1 , the answer is certainly positive: the two types of graphs were both first compressed and subsequently subjected to the PCA dimensionality reduction. Consequently, the 2D representations represent both the students of each PA sessions, which we called *real* graphs, and the graphs with the teacher’s grades, which we have called the *ideal* graphs. Furthermore, we were also able to generate a representation of the multigraphs obtained by merging the different graphs built for representing different sessions. About the RQ_2 , the vector representation of the graphs allows us to have an idea on the evolution of the learning process. For example, the representation of the multigraphs shows that their relative distances, taking into account all the PA sessions, are decreasing, as illustrated in Fig. 2. This can be interpreted as the capability of the system to represent the dynamics of the community during the PA sessions. Another important feature is the capability to follow the dynamics of the community through a point in a 2D

Graphs	Distance
r_1-r_2	0.08
r_2-r_3	0.37
i_4-i_5	0.05
i_5-i_6	0.77

(a)

Graphs	Distance
r_1-r_2	0.08
r_2-r_3	0.03
i_4-i_5	0.05
i_5-i_6	0.12

(b)

Fig. 2. In Figure (a) the distances in the 2D space among the real and ideal graphs for the first kind of graphs. In Figure (b) The distances in the 2D space among the real and teacher graphs for the second kind of graphs.

Table 2. The two types of graphs produced by the PA sessions. By r_i we mean the real graphs, i.e., the PA graphs while i_i graphs are the ideal ones, i.e., with teacher's grades. The last column shows the coverage of the graphs in terms of Edges.

Students' graphs	Teacher's graphs	PA session	Coverage
r_1	i_1	s_1	25%
r_2	i_2	s_2	25%
r_3	i_3	s_3	25%

space. To highlight this feature, in the Fig. 1 the points representing the graphs have been joined with a dashed line. However, it is important to highlight that each point represents all the properties of the community. Alternatively we are compelled to observe the entire graph and its evolution. Obviously, additional aspects are yet to be deeply investigated, such as useful interpretations of the X and Y values in the 2D space.

5 Conclusions and Future Work



In this paper, we have proposed the use of Graph Embeddings to investigate the chance of modeling learning communities through PA sessions, while in the literature, a community of students is usually represented by a directed graph where each node corresponds to a student and each edge represents a relationship between two students. We have presented a case study based on a high school computer science classroom composed by 25 students, over three PA sessions. GE and PCA have been applied in order to compress the graph into a n -dimensional vector and subsequently into a two-dimensional vector. These techniques allowed us to graphically represent all the PA sessions and monitoring the dynamics of the community in a 2D space. The same work done on the original graph would have entailed greater challenges. As a future work, we would like to deepen this approach by studying the possibility of using this technique on very large communities like MOOCs together with other teaching/learning strategies.

References

1. Cai, H., Zheng, V., Chang, K.: A comprehensive survey of graph embedding: problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.* **30**, 1616–1637 (2018)
2. Dai, Y., Wang, S., Xiong, N.N., Guo, W.: A survey on knowledge graph embedding: approaches, applications and benchmarks. *Electronics* **9**(5) (2020)
3. Gasparetti, F., Limongelli, C., Sciarrone, F.: Exploiting wikipedia for discovering prerequisite relationships among learning objects (2015)
4. Jolliffe, I.: Principal component analysis. In: Lovric, M. (ed.) *International Encyclopedia of Statistical Science*, pp. 1094–1096. Springer, Heidelberg (2011)
5. Lai, Y.Y., Neville, J., Goldwasser, D.: Transconv: relationship embedding in social networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33(01), pp. 4130–4138 (2019)
6. Lara, N.D., Pineau, E.: A simple baseline algorithm for graph classification (2018)
7. Limongelli, C., Gasparetti, F., Sciarrone, F.: Wiki course builder: a system for retrieving and sequencing didactic materials from wikipedia (2015)
8. Limongelli, C., Mosiello, G., Panzieri, S., Sciarrone, F.: Virtual industrial training: joining innovative interfaces with plant modeling (2012)
9. Limongelli, C., Sciarrone, F., Starace, P., Temperini, M.: An ontology-driven olap system to help teachers in the analysis of web learning object repositories. *Inf. Syst. Manag.* **27**(3), 198–206 (2010)
10. Pan, L., Wang, X., Li, C., Li, J., Tang, J.: Course concept extraction in MOOCs via embedding-based graph propagation. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 875–884. Asian Federation of Natural Language Processing, Taipei, Taiwan, November 2017
11. Pellegrino, M.A., Altabba, A., Garofalo, M., Ristoski, P., Cochez, M.: GEval: a modular and extensible evaluation framework for graph embedding techniques. In: Harth, A., Kirrane, S., Ngonga Ngomo, A.-C., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M. (eds.) *ESWC 2020. LNCS*, vol. 12123, pp. 565–582. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_33
12. Sciarrone, F., Temperini, M.: K-openanswer: a simulation environment to analyze the dynamics of massive open online courses in smart cities. *Soft. Comput.* **24**(5), 11121–11134 (2020)



Three Common Group Formations in Online Collaborative Learning

Tao Wu¹  and Maiga Chang² 

¹ South China Agricultural University, Guangzhou, China

² School of Computing and Information Systems, Athabasca University, Athabasca, Canada
maigac@athabascau.ca

Abstract. With the widespread use of emerging computer technologies in teaching and learning, computer-supported collaboration as a beneficial teaching and learning strategy is now embraced in campus. Analyzing the influencing factors of individual participation is important to discover a more efficient group forming in terms of accomplishing collaborative learning tasks. Through the social network analysis approach on the collected data, this research has found that personality trait does affect a student's activeness and the group's mode. Three main group modes are discovered: Unipolar, Multi-Center, and Centerless-Flat mode. Multi-Center group mode is more stable and has higher average academic achievement than the other two modes. The research findings can be used to implement an intelligent tutoring system that can make recommendations for teachers on the better options of grouping students based on their personality traits.

Keywords: Social network analysis · Personality traits · Collaborative Learning · Social messaging app · Big Five Inventory (BFI)

1 Introduction

Research on cooperative and collaborative learning has provided empirical support for the cognitive, motivational and social benefits of group work [1]. Various empirical studies show evidences that group work and collaboration enhance students to engage with the learning materials and develop deep disciplinary understanding [2, 3]. Some researchers realized that the group members' emotion influence their collaborative learning interactions in group [4]. Rehm and colleagues (2016) have done empirical study that contributes to the understanding of how the characteristics of group members influence their collaborative learning interactions [7]. They also find that participants who are more active have better learning performances.

This research aims to investigate the factors that may affect the group modes and learning effects while students working on the designed collaborative learning tasks. With the identified influential factors, a better group could be recommended to form so students will establish group awareness and grasp collaboration and communication skills better to solve known or unknown problems [5].

There have been studies proving that individuals whose social behavior is readily predictable from measures of personality traits [6]. Few studies analyze the model of team discussion and engagement in e-learning based on personality trait. The research team assumes that a group's learning outcome will be mainly depending on the communications and collaboration level the group members can achieve. Furthermore, students' personality traits like openness, conscientiousness, extraversion, neuroticism may influence their participation, activeness, and contribution to collaborative learning tasks in a group.

Therefore, this research has the following two hypotheses accordingly:

- Hypothesis H1: Individuals' personality traits affect their social behaviors in online discussion.
- Hypothesis H2: Individual's engagement level will be reduced when working on more complex collaborative tasks.

2 Empirical Study

To verify the hypotheses and further understand the collaborative learning and interactions in the online groups. The experiment was conducted in second term during the academic year 2019–2020 with 7 Sophomore classes (N = 370, 114 male and 256 female students) of Tax Law in South China Agricultural University (SCAU). At the end there were 216 valid responses collected from 56 male and 160 female students. This course was running on the Mosoteach Learning Management System¹ and included seven quizzes conducted at an interval of ten days in two months.

At the end of each chapter, collaborative learning task and group discussion were conducted. Different group modes were investigated and compared based on their group members' average quiz performance. In this study, the task complexity and difficulty levels are divided into three: simple, medium and difficult according to the three dimensions of element interaction, memory participation and logical inference.

Four collaborative tasks that have different difficulty levels are designed for the students to complete in groups within two and half months. Every couple of weeks a task will be given to the student groups. Table 1 lists the four tasks' complexity and purpose. Students will freely form study groups where each group has 6 to 8 members.

Table 1. Four tasks designed for the experiment

Task	Complexity	Duration	Topic
#1	Simple	March 3–March 14	Animation production
#2	Medium	March 17–March 28	The learning & use of mind maps
#3	Medium to Difficult	April 7–April 21	Group discussion on “Value-Added Tax (VAT)”
#4	Difficult	April 28–May 14	Teacher assigned discussion question

¹ <http://www.mosoteach.cn>.

In order to verify the two hypotheses, the participants' interactions within a group while working on the collaborative learning tasks and their personality traits are needed to be collected. The experiment uses the mobile app WeChat for group discussions. WeChat is a popular app in China and it is similar to all other social messaging apps in the world (e.g., Whatsapp, LINE and Telegram). The teacher will not participate in the group discussion so students can feel more comfort while talking to each other. At the end of a collaborative task, the group leader will send the screenshot of the group's discussion process to the instructor.

The research uses social network analysis on the interactions among group members to categorize groups into different modes accordingly. The research also adopts the forty-four 5-point Likert-scale items Big Five personality traits instrument to create an online questionnaire for participants filling out at the end of the experiment [8, 9]. With the understanding of individual group members' personality traits, the research can further investigate the potential correlations between group's personality traits and modes.

3 Data Collection and Analysis

When the teacher received the screenshots of groups' discussion in WeChat from the group leaders, the teacher created Excel files to transcribe and store the interactions in a matrix form for all groups. The interaction matrix was then processed by social network analysis (SNA) approach with the iGraph² package of the R software, to generate a diagram for the interactive pattern. With SNA diagram the group members' status and interaction frequency can be told easily [10].

Three group modes had been found while students doing the collaborative learning tasks, they are: Unipolar mode where the student (either the group leader or another member) is the center and interacts with other group members; Multi-Center mode where two or more people are particularly active participating and become centers in the group; and, the Centerless-Flat mode where everyone participate more or less equally in the group.

For simplifying the analysis and comparison, this paper takes three groups (i.e., Group C, Group V, and Group A) as examples and explains the changes of students' interactions from simple to difficult tasks (i.e., from Tasks #2 to #4). Figure 1 shows the SNA diagrams of a Unipolar mode Group C doing the collaborative learning tasks. The student c_1 (who is also the group leader) constantly asks the group members to answer the questions and do the jobs. The interactions within the group are found to be very negative (based on the recorded discussion on the screenshots). At the end the group members are no longer willing to participate for Task #4.

Figure 2 shows the SNA diagrams of a Multi-Center mode Group V doing the collaborative learning tasks. Starting from Task #2, student v_1 (who is the group leader) and v_2 are actively interacting with each other frequently. While working on the Task #3, student v_7 is also jumping in become another center besides the leader v_1 . In order to solve Task #4, it is necessary to find the answer on the group's own. Among eight group members five of them choose to not participate. At the end v_2 actively played in the group to complete Task #4.

² <https://rpubs.com/crconline/igraphreview>.

Figure 3 shows the SNA diagrams of a Centerless-Flat mode Group A doing the collaborative learning tasks. In Group A, student a_1 is the group leader. In this group there is no outstanding one but the interactions and discussion among the group members are positive and active. Even when one member drops out under the pressure of the difficult task while working on Task #4, the interaction mode is still maintained and the rest of group members are still working on the task actively. Nevertheless, all the groups do suffer from the individuals' reduced engagement in the difficult task, Task #4. This result verifies the Hypothesis H2.

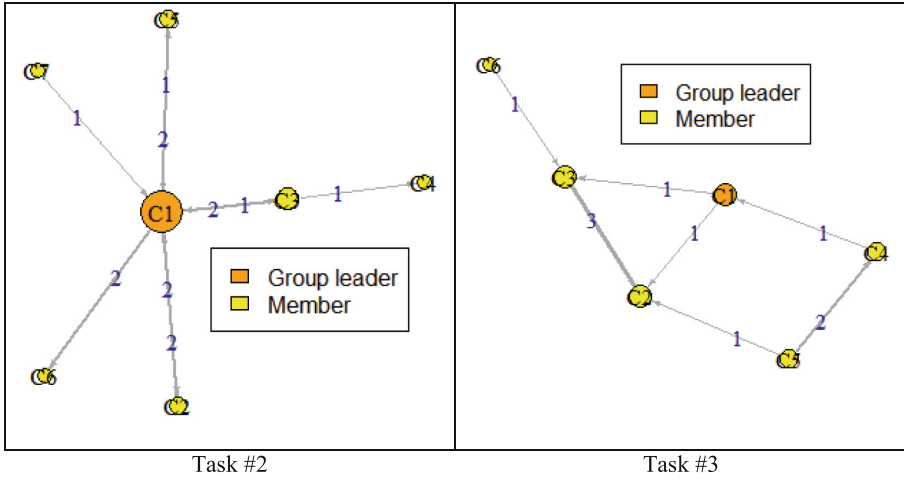


Fig. 1. Group C (Unipolar mode) works for Tasks #2 and #3.

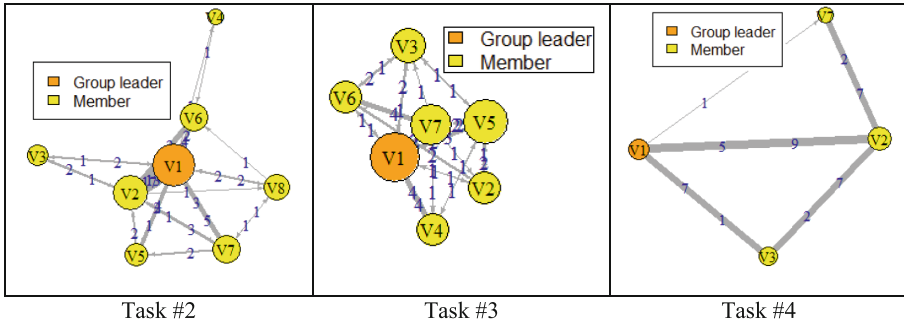


Fig. 2. Group V (Multi-Center mode) works for 3 tasks.

The research team is also interested in understanding the differences of personality trait distributions that different group modes may have. The investigation results could not only enlighten the way of forming better groups for all students learn more efficiently, but also become the basis of implementing an intelligent tutoring system that can make recommendation for teachers on how to group their students to maximize the learning

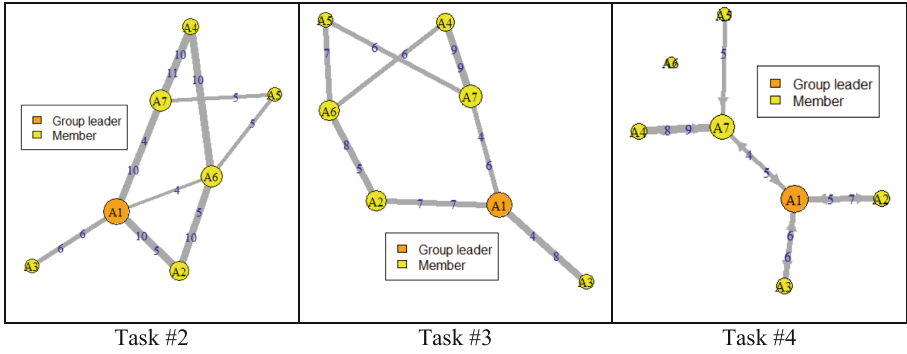


Fig. 3. Group A (Centerless-Flat mode) works for 3 tasks

achievement. The research team compares personality trait distributions among the three group modes based on the average personality traits distributions that (a) top one-third students who have higher performance on the quizzes (i.e., high achievement students), (b) bottom one-third students (i.e., low achievement students), and (c) all students in a group. Figure 4 uses the radar charts to present the differences of students’ average personality traits in the three group modes.

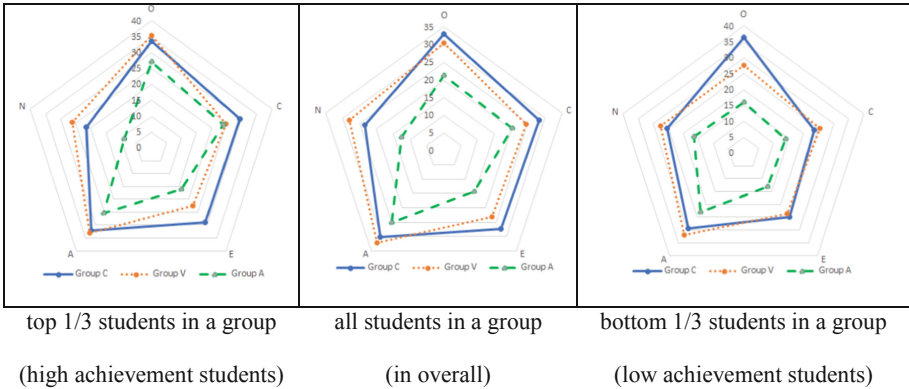


Fig. 4. Group personality traits distributions

From the radar charts shown in Fig. 4 we can tell that the three groups do have different personality traits distributions – Unipolar mode Group C with solid line in blue, Multi-Center mode Group V with dot line in orange, and Centerless-Flat mode Group A with dash line in green. Comparing the three groups’ average personality traits values by taking all students in a group into consideration as well as the high and low achievement students.

The research has found that Centerless-Flat mode Group A not only has lowest average personality traits than the other two groups but also has positive skewness on Agreeableness personality trait than the other four traits. From the group-based average

academic performance on the final exam and the quizzes (see Table 2), the group’s average quiz performance is lowest. Furthermore, doesn’t like other two modes the standard deviations of Centerless-Flat mode group members’ final exam marks and average quiz marks are at similar level, although the group’s average final exam mark is not the lowest.

Table 2. Descriptive analysis of academic performance

Group	Mode	Final exam (mean)	Final exam SD (high/low)	Quiz avg (mean)	Quiz avg SD (high/low)
Group C	Unipolar	43.50	24.71 (84/19)	60.85	12.29 (74.86/43.58)
Group V	Multi-Center	53.88	18.86 (83/29)	60.14	12.19 (77.82/42.00)
Group A	Centerless-Flat	47.33	14.81 (70/30)	54.29	13.35 (72.12/39.36)

On the other hand, the Unipolar mode Group C and the Multi-Center mode Group V do share similar personality traits distributions according to Fig. 4. However, the group personality traits distributions show the Unipolar mode Group C has (1) higher Openness trait value, especially for the low achievement students; (2) lower Neuroticism trait value, especially for the high achievement students; and (3) higher Conscientiousness and Extraversion personality traits, especially for high achievement students. Last but not the least, while the two groups’ students have similar average quiz performance in terms of the mean value and the standard deviation, they do have different performance on the final exam according to Table 2. Therefore, Hypothesis H1 is supported.

4 Conclusion

This research investigates the interactions of doing collaborative learning activities within groups in which each group has 6 to 8 members involved. The research identifies three major group modes: Unipolar, Multi-Center, and Centerless-Flat mode. Multi-Center mode has been proved to have overall better academic achievements in terms of average quiz mark and the group structure and member connections are more stable and tighter even under the pressure of doing the difficult collaborative learning task online together. Personality traits do have influences on the communication types; for instances, a group has very high Openness personality trait value distribution may make the group become a Unipolar mode group in which most of students may not engage in the discussions; and a Centerless-Flat mode group may be formed when the group’s personality trait value distribution shows that the group’s Agreeableness personality trait outstanding to other traits.

References

1. Volet, S., Mansfield, C.: Group work at university: significance of personal goals in the regulation strategies of students with positive and negative appraisals. *Higher Educ. Res. Develop.* **25**(4), 341–356 (2006)
2. Engle, F.R.: Guiding principles for fostering productive disciplinary engagement: explaining an emergent argument in a community of learners classroom. *Cogn. Instr.* **20**(4), 399–483 (2002)
3. Hmelo-Silver, C.E.: Problem-based learning: what and how do students learn? *Educ. Psychol. Rev.* **16**, 235–266 (2004)
4. Järvenoja, H., Järvelä, S.: Emotion control in collaborative learning situations: do students regulate emotions evoked by social challenges. *Br. J. Educ. Psychol.* **79**(3), 463–481 (2009). <https://doi.org/10.1348/000709909X402811>
5. Rehm, M., Mulder, R.H., Gijsselaers, W., Segers, M.: The impact of hierarchical positions on the type of communication within online communities of learning. *Comput. Hum. Behav.* **58**, 158–170 (2016)
6. Goodyear, P., Jones, C., Thompson, K.: Computer-Supported Collaborative Learning: Instructional Approaches, Group Processes and Educational Designs. In: Spector, J.M., Merrill, M.D., Elen, J., Bishop, M.J. (eds.) *Handbook of Research on Educational Communications and Technology*, pp. 439–451. Springer, New York (2014). https://doi.org/10.1007/978-1-4614-3185-5_35
7. Rehm, M., Mulder, R.H., Gijsselaers, W., Segers, M.: The impact of hierarchical positions on the type of communication within online communities of learning. *Comput. Hum. Behav.* **58**, 158–170 (2016)
8. John, O.P., Naumann, L.P., Soto, C.J.: Paradigm shift to the integrative big five trait taxonomy: History, measurement, and conceptual issues. In: *Handbook of Personality: Theory and Research*, 3rd edn. (2008)
9. Ridgell Susan, D., Lounsbury, J.W.: Predicting academic success: general intelligence, “Big Five” personality traits, and work drive. *College Stud. J.* **38**, 507–618 (2004)
10. de Laat, M., Lally, V., Lipponen, L., Simons, R.-J.: Investigating patterns of interaction in networked learning and computer-supported collaborative learning: a role for Social Network Analysis. *Int. J. Comput. Support. Collab. Learn.* **2**(1), 87–103 (2007)



New Horizons on Online Tutoring System Inspired by Teaching Strategies and Learning Styles

Karima Boussaha¹(✉) and Samia Drissi²

¹ Department of Computer Science, Research Laboratory On Computer Science's Complex Systems (ReLa(CS)2), Larbi Ben M'hidi University, Oum El Bouaghi, Algeria

² Department of Computer Science, Chérif Messadia University, Souk Ahras, Algeria

Abstract. Tutoring is a human activity, which has been applied in several fields. In the last decade, this task has become indispensable, especially in higher education institutions. The principal objective attached to the tutor is to assist and support the learners throughout their learning process. Several researchers have studied the impact of collaboration between the learners on their cognitive levels, but few studies have been carried out on the impact of collaboration among the tutors.

In the previous work, we have studied the impact of the collaboration among the learners with a specific collaborative CEHL(Computing Environment for Human Learning) [1]. In this work, we focused on the impact of collaboration among tutors.

Keywords: Tutors collaboration · Computer-supported collaborative coaching · CSCC · Tutor's groups · Coach's group · Coach's profile · Learning style · Teaching strategies

1 Introduction

In CSCL (Computer-Supported Collaborative Learning) research field, researchers are mainly interested in the collaboration between learners as means for supporting collaborative learning, which allows the learner to work with the group to achieve a common goal [3, 4]. But, collaboration among learners is not enough to solve some problems as some learners find it difficult to communicate and share experiences within the group [3, 5]. As result, monitoring functionality is required in these environments. Tutoring is a key element of any distance learning system.

The roles of tutors in distance learning are numerous. Some researchers argue that the tutor is responsible for his intervention to facilitate the learning process and monitor their activities. His role is that of an accompanying person, coach, or resource-person [3]. Other researchers confirm that in most distance courses, the tutor has mainly a psychological and methodological role [3]. Other roles are cited by many researchers [3–5]: guide, evaluator, moderator, etc.

Furthermore, this actor has several names: mentor, coach, facilitator [3, 5], etc. As a conclusion about the tutors' roles, we can say that there is confusion about these roles

from one researcher to another. This confusion is due to the lack of works about the standardization and the instrumentation of the tutors' roles [3, 5]. In other words, we can say that the name of the tutor depends on his role and his field of work or the place of work, in our research we have attributed the name of the coach to the tutor because his ultimate role is to support and assist all the activities of learners to find learning difficulties and problems in higher education level.

This paper is organized as follows. In Sect. 2, we are confronted with a new field of research, which is CSCC “Computer-Supported Collaborative Coaching”, we present a definition of CSCC, its purpose, and its advantages over other types of existing tutoring systems. The architecture of a CSCC system is presented in Sect. 3. In Sect. 4, we explain the scenarios of collaboration in the CSCC system proposed. Finally, we conclude with a conclusion and future works.

2 CSCC: Computer-Supported Collaborative Coaching Field

In higher education institutions, new recruits (teachers/tutors/coaches) who work in individual environments, have encountered difficulties that are often related to their demotivation and isolation, which are in most cases related to lack of experience [3, 5]. Our work is interested in limiting these difficulties by developing autonomy and reducing the coaches' feeling of isolation and helping them to acquire the know-how of their most experienced colleagues, by creating social links of communication and collaboration between the coaches and the rest of the educational community. The main goal of computer-supported collaborative coaching is to support and guide learners to improve their learning level and continuous monitoring. Besides, collaboration among (coaches/tutors) helps them through this important training activity. Figure 1 shows the position of this new research field among the main related research fields.

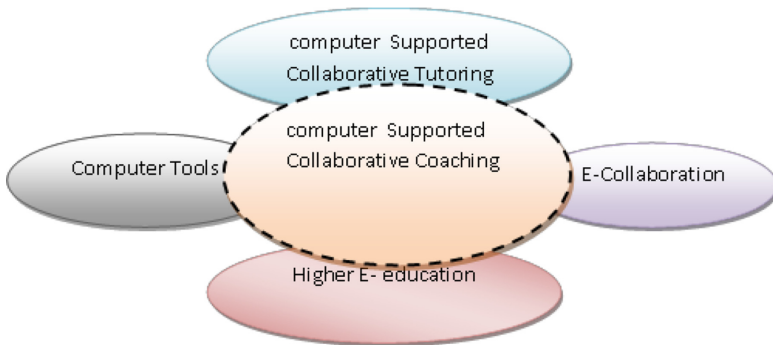


Fig. 1. Description of CSCC domain

As shown in Fig. 1, the CSCC is the intersection of three research fields: computer-supported collaboration, computer-supported collaborative tutoring, and Higher e-education. To better present this new field, we give some definitions of it.

Definition 1: CSCC is a tutoring strategy where the tutors in higher education institutions called **coaches** form tutoring groups (also called communities of practice) whose goal is to support learners in their educational activities (learning, assessment...) using collaboration tools and techniques.

Definition 2: CSCC is a process that takes into account collaboration among coaches (tutors in higher education institutions) using computer tools. Coaches get together in small groups to provide better tracking of learners and to help novice coaches in their mission.

3 General Architecture of a CSCC System Proposed

A computer-supported collaborative coaching system is a collaborative platform that brings together all tools of distance tutoring, but in a specific area which is the higher education institutions. To provide a space for coaches (tutors) for working together to assist learners, and eliminating some problems for example: in the existing system, the learner is assigned to one and only one human coach. The latter replies to his/her learners' queries and tries to solve their problems. Furthermore, when a learner's needs do not belong to the coach's skills, the learner's queries will not be satisfied. The collaboration among coaches to carry out the task of learner monitoring will enable coaches to meet all the needs of learners seeking assistance. Figure 2 shows the various components of a CSCC system, which are:

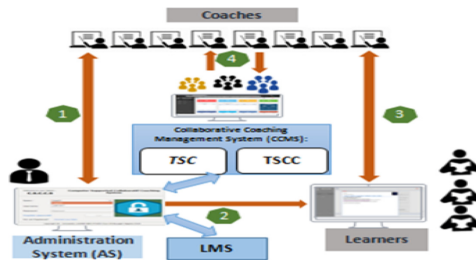


Fig. 2. The various components of a CSCC proposed system.

1. **Administrator System (AS):** this component is responsible for managing all the actors of the system (coaches, learners, and teachers) and the learning paths.
2. **Learning Management System (LMS):** it manages to learn activities and materials.
3. **Collaborative Coaching Management System (CCMS):** It consists of two subsystems:
 - a. **A Tracking System of Coaches (TSC):** that provides the coach with a set of information to track learners (coaching journal, traces, ask helps..) and allows responding to learner's assistance requests. Knowing that the learners involved

in this system are graduating learners (bachelor, master, doctorate) then each of them is assigned to a coach. This tracking system manages the areas and disciplines of the department, teachers, and learners and assigns them a username and password that they will use later to enter in their spaces, then inform them with their email addresses.

- b. **A Tracking System of Collaboration among the Coaches (TSCC):** This provides the coaches with a collaboration space presented as a set of collaboration and communication tools (chat, forum, mail, etc.), and mechanisms to monitor the collaboration. To achieve the main task of our system which is the collaboration between coaches, we have adopted the concept of groups. At the group level, members can discuss collectively: using either intergroup or intragroup discussion.

4 Scenarios of Collaboration in CSCC System Proposed

4.1 The Mechanism for Forming Coaches' Groups

The methodology proposed for creating the coaches' groups is based on the set of coaches' profiles.

In our work, we need to know the ability of each coach to collaborate in the activities of his/her pairs, and his/her preferences about how he/she prefer to learn new notions about the coaching process, that is why we proposed to create two new profiles: The first one called:

4.1.1 Collaborator Profile

The aim of this new profile is to pre-classify coaches according to their level of collaboration in the different activities of their colleagues (very passive, little passive, little active, active, highly active). We must note that we need this profile when a coach has the desire to join a group. We applied in this collaborator profile a new strategy very known in the teaching strategies for determining the learning style of each coach it is a learning strategy.

Learning Strategy: Uses combining and adapting teaching strategies, learning styles, and electronic media According to Felder-Silverman's learning style model [2]. FSLSM contains four dimensions when the learner is characterized by a specific preference for each of these dimensions [2]. Each dimension includes two variables as shown in Fig. 3. Let us note that in our case we adopted this model for the coaches learning.

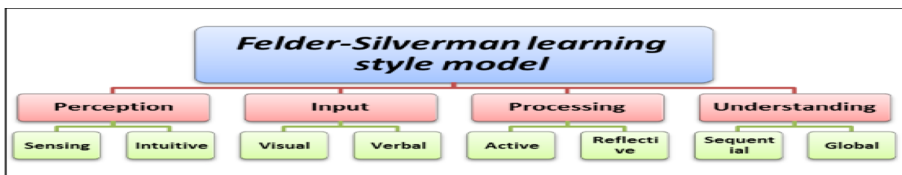


Fig. 3. Felder-Silverman learning style model [2]

4.1.2 Group Profile

the interest domain of the coach is related to his specialty diploma we try to reduce the number of groups to eight (e-learning – multi-agent system, Artificial vision, image processing, artificial intelligence, networks, information system, internet of things).

5 Conclusion and Future Works

In this paper, as the first step of our research project, we have presented the basic principles of the new research field which is computer-supported collaborative coaching, which supports collaboration among human coaches in higher education institutions. We present a definition of CSCC, its purpose, and its advantages over other types of existing tutoring systems. Also, its general architecture. We describe the scenarios of the collaboration between coaches by proposing two new profiles (the coach's collaborator profile, and the coaches' group profile) using for forming coaches' groups.

In future work, we plan to extend the proposed approach by developing a prototype of the Computer-supported collaborative coaching system proposed, and the Classification algorithm for forming coaches' groups, this algorithm used the values of collaborator and group profiles proposed.

References

1. Boussaha, K., Mokhati, F., Zakaria, C.: Architecture of a specific platform for training practical works: integration of learners' assessment component. *Int. J. Technol. Enhanced Learn.* **7**(3), 195 (2015). <https://doi.org/10.1504/IJTEL.2015.072809>
2. Drissi, S., Amirat, A.: Adaptation with four-dimensional personalization criteria based on Felder Silverman model. *Int. J. Distance Educ. Technol.* **15**(4), 1–20 (2017)
3. Lafifi, Y., Hadjeris, M., Seridi, A., Bourbia, R.: Architecture of a collaborative tutoring system. *Procedia Soc. Behav. Sci.* **31**, 459–463 (2012). <https://doi.org/10.1016/j.sbspro.2011.12.086>
4. Tadjer, H., Lafifi, Y., Seridi-Bouchelaghem, H., Gülseçen, S.: Improving soft skills based on students' traces in problem-based learning environments. *Interact. Learn. Environ.* (2020). <https://doi.org/10.1080/10494820.2020.1753215>
5. Zedadra, A., Lafifi, Y., Zedadra, O.: Dynamic group formation based on a natural phenomenon. *Int. J. Distance Educ. Technol.* **14**(4), 13–26 (2016). <https://doi.org/10.4018/IJDET.2016100102>



A Comparative Evaluation of the Effect of Social Comparison, Competition, and Social Learning in Persuasive Technology on Learning

Fidelia A. Orji and Julita Vassileva^(✉)

Department of Computer Science, University of Saskatchewan, Saskatoon, Canada
fidelia.orji@usask.ca, jiv@cs.usask.ca

Abstract. The design of successful online learning systems that attract and sustain students' active engagement with their learning activities remains an open question. With the rapid adoption of online learning systems (including learning management systems) due to the Covid-19 pandemic, there is a need to ensure that the systems' design can encourage and sustain students to actively use them to achieve the required learning objectives. We argue that it is possible to incorporate socially-oriented persuasive strategies in the design of online learning systems to support students' engagement and improve learning outcomes. We integrated three socially-oriented persuasive strategies (upward social comparison, social learning and competition) with a learning management system used in a university course. The strategies were implemented as social visualizations, and we explored their influence on 628 students' access to learning materials. We found that the three persuasive strategies implemented as social visualizations encouraged students to access more frequently the learning materials, and this increase in frequency correlated positively with their academic performance (final grade in a course). Our results can help designers of online learning systems to improve the efficacy of their systems.

Keywords: Persuasive Technology · Learning Management Systems · Social Influence · Social Comparison · Student Engagement · Social Learning · eLearning Systems · Competition

1 Introduction

Computer applications and services datasets play an important role in aiding decision-making based on actual application/service usage. Thus, organizations nowadays explore and analyze their datasets to make informed decisions concerning their progress and success. For example, datasets from various domains such as education, e-commerce, healthcare, and businesses are now providing data-based evidence that informs decision-making at multiple levels. In the education domain, there is growing interest in supporting and improving learning through technological applications (such as eLearning systems, learning management systems, intelligent tutors, and collaborative learning environments). These applications can improve their performance by learning from their usage data.

© Springer Nature Switzerland AG 2021

A. I. Cristea and C. Troussas (Eds.): ITS 2021, LNCS 12677, pp. 369–375, 2021.

https://doi.org/10.1007/978-3-030-80421-3_41

Previous studies analyzed patterns of students' access to various learning materials to answer different research questions. Wang et al. [1] indicated that the frequency of access to online learning materials and personality variables predicted the final course grade. Also, Heffner et al. [2], who explored website access for learning materials, revealed that students who passed the course accessed the course online pages more frequently than those who failed the course. While the previous studies evaluated how the use of LMS in universities affected students' learning and performance, our study explored how incorporating some socially-oriented persuasive strategies (upward social comparison, social learning and competition) implemented as visualizations affect students' learning activities and performance. Only a few studies focused on investigating the effects of the three strategies on students' learning activities.

In previous research [3], our colleagues developed a Student Advice Recommender Agent (SARA) based on a predictive analytics model and integrated it with the Blackboard Learning Management System (LMS) used in our University. SARA detects opportunities to provide personalized advice and direct the students to relevant learning resources based on each student's learning needs. The system has been used in various undergraduate courses to assist and support students by providing personalized learning support. The system has a positive effect on some students' learning experience and outcome, but many of the students do not engage with the system and ignore the advice. To improve engagement with the SARA system, we developed and deployed a persuasive intervention using socially-oriented strategies and social visualizations. Interventions utilizing visualization [4], self-monitoring tools, and persuasive systems [5] have proven effective in improving engagement. Hence, we decided to incorporate persuasive strategies into the SARA system to evaluate whether it can encourage students to engage actively with their learning resources.

This paper seeks to answer the following research questions:

RQ1: Which of the three investigated socially-oriented strategies most effectively increases the frequency of the students' access to the learning materials?

RQ2: How is the frequency of access to learning materials related to academic performance?

RQ3: How do the Social Comparison, Competition, and Social Learning strategies employed in the online learning system affect the students' performance?

The learning system log analysis of 628 student records confirms the effectiveness of the three socially-oriented persuasive strategies in motivating students to access the learning materials more frequently. The results also show a correlation between the frequency of learning material access and academic performance (the final grade in a course). Our findings suggest that online learning systems can employ in their design the socially-oriented strategies explored in this research to improve student engagement with the learning materials and academic performance. The implementation of the strategies is straightforward and domain-independent, and they can be incorporated readily into the learning systems design.

2 Research Method

The dataset used in this work is part of a large research project aimed at using socially-oriented persuasive technology strategies in improving students' engagement in learning activities. The dataset was collected in a study approved by our University's behaviour ethics board. The design and implementation of the socially-oriented strategies in a learning management system have been presented elsewhere [6, 7] along with the pre- and post-survey of students to evaluate the system's persuasiveness [8]. This paper aims to evaluate how incorporating persuasive strategies into a personalized learning system affects students' access to the provided relevant learning resources (lecture materials and slides, video lectures and tutorials related to the topic of each week). All student viewing activities of the various learning resources were recorded in the system log. After cleaning, the logs consist of 628 records of students' learning data distributed across the four experimental conditions (student groups exposed to each of the three persuasive strategies and a control group) as follows: competition – 36 records, social comparison – 258 records, social learning – 190 records, and control – 144 records.

2.1 Data Analysis

We are interested in understanding whether there are changes in the frequency at which students view their learning materials which could be attributed to the incorporation of the three distinct socially-oriented persuasive strategies in the learning management system. This entails examining and comparing the data from the experimental conditions (four different groups of students) in terms of the viewing of learning materials over time (before and during the persuasive intervention). To achieve this, we used a within-subject design to collect data about viewing behaviour before the intervention was introduced (the first half of the semester) and during the intervention (the second half of the semester). The results are shown in Fig. 1. After validating the data for ANOVA assumptions, we performed Repeated Measures Analysis of Variance (RM-ANOVA) using the viewing of learning materials before and during our persuasive intervention as within-subjects factors and experimental conditions (competition, social comparison, social learning, and control groups) as between-subjects factors. The weekly view frequency was calculated by normalizing the total frequency of views made by each student before the intervention and during the intervention for the number of weeks in each

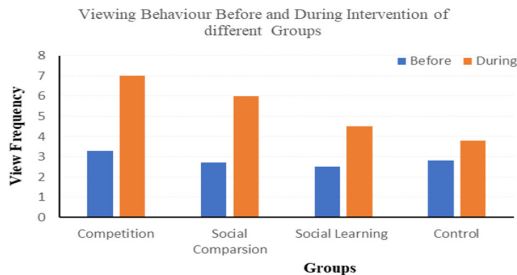


Fig. 1. Summary of viewing behaviour of experimental conditions

period (7 and 6, respectively). This analysis helped us determine if there is a significant difference in the viewing pattern of learning materials among our different groups.

In addition, we examined whether there is a relationship between viewing behaviour across the groups and their academic performance (measured by the final grade in a course) using correlation and ANOVA. We used SPSS for all our analyzes and report results as significant when $p < .05$. Furthermore, we analyzed the students' feedback (provided in their learning system) to understand their perception of the persuasive strategies.

3 Results

The analysis of our data on the students' viewing behaviour revealed some interesting insights and trends.

3.1 Effect of Persuasive Strategies on the Students' Viewing Behaviour

The results of RM-ANOVA with time (before and during intervention) as within-subject factors and four experimental conditions as between-subjects factors show that there is a significant interaction between time (before, during intervention) and experimental conditions in terms of viewing pattern score ($F_{3, 624} = 7.464, p < .001$). Also, there was a significant main effect of experimental conditions ($F_{3, 624} = 4.896, p < .005$) on viewing pattern scores overall. This means that there was a statistically significant difference between the students' groups with respect to the viewing behaviour of their learning materials before and during the intervention. The pairwise comparison results between the experimental conditions show that students in the competition group viewed their learning materials more actively than those in the social learning ($p < .050$) and the control group ($p < .005$). The difference in the viewing pattern between students in the control and the social comparison groups was significant ($p < .005$). The difference in the viewing pattern between students in the competition and social comparison groups was not significant ($p > .100$). Furthermore, the results showed that students in the social comparison group viewed their learning materials more often than those in the social learning group ($p < .050$). However, the difference in the viewing pattern between students in the control and the social learning groups was not significant ($p < .200$). This means that the students in the competition and social comparison groups viewed the learning material more often than those in the social learning and control groups.

3.2 Effect of Students' Viewing Behaviour on Their Performance

The result of Pearson correlation analysis showed that the students' viewing frequency and their academic performance positively correlated, $r(626) = .304, p < .001$. The descriptive statistics of the dataset show that students in the competition group had the highest mean in academic performance (final course grade) ($M = 79.28$), followed by the social comparison ($M = 62.38$), the social learning ($M = 58.02$), and control groups ($M = 57.34$). A one-way between-subjects ANOVA was conducted to compare whether the differences in academic performance across the four groups are statistically significant.

The results show a significant main effect of the experimental condition on academic performance for the four conditions ($F_{3, 624} = 24.499, p < .001$). This means that there is a significant performance difference between the students across the four experimental conditions.

The results of the post hoc comparison indicated that the mean score for the competition condition was significantly different from the social comparison condition ($p < .001$), social learning condition ($p < .001$), and control condition ($p < .001$). Also, the mean score for the social comparison condition was significantly different from the social learning ($p < .020$) and control conditions ($p < .005$). However, the social learning condition did not significantly differ from the control condition ($p > .900$). Specifically, the results suggest that the three socially-oriented strategies affect academic performance since they affected the students' viewing frequency of learning materials overall. The competition and social comparison strategies groups had better performance than the social learning and the control groups.

3.3 Effect of the Persuasive Intervention Based on Students' Feedbacks

We present a summary and a brief discussion on the feedback collected from students in the three experimental groups using versions of the learning system with persuasive visualizations. The students paid attention to the strategies implemented in the system, as shown by their feedback. For example, common comments among students in the competition group were: *"I'm at the top of the class!"*, *"Happy that I did so well but think I could have got even higher"*, and *"Wish my grade was higher"*. This means that the incorporation of the competition principles in their learning system motivated students to perform better. Based on our analysis the competition group performed better than the other three groups in both their access to learning materials and academic performance. For students in the social comparison group, comments such as: *"I did not get over 83"*, *"I thought I did better"*, and *"I am over the class average"* were common. These comments suggest that students compared themselves to those who performed better than them in the course assessments (role models). The students will be motivated to improve their learning behaviour to perform better like their role models. Students in the social learning group made comments such as: *"Surprised I did so good on the midterm yet so bad on the lab exam"*, *"How I got so low grades?"* and *"I thought I was well prepared for the class midterm"* which means that the students acquired (passively) information relating to their progress by observing their peers' performance. Thus, the persuasive strategies provided an opportunity for students to be frequently reminded of their performance goals and their progress to promote more frequent access to the learning materials.

4 Discussion

We examined students' access frequency of learning materials in a university course to answer our research questions concerning the influence of socially-oriented persuasive strategies on learning activities. The students were grouped based on the version of the persuasive visualization they were exposed to (or lack thereof, for the control group).

Using as the baseline the students' viewing behaviour before the persuasive intervention's introduction, the results of our analysis revealed that students' viewing frequency increased during the intervention period. We also found that the viewing frequency differed across the four experimental conditions, with students in the competition group having the highest activity frequency, followed by the social comparison, social learning, and control groups respectively, which answers our first research question, *RQ1*.

Our correlational analysis showed that the access frequency to learning materials is moderately positively correlated with the students' final grades, which answers the second research question, *RQ2*. The results align with previous research, indicating that frequent access to web-based course materials leads to high performance in terms of course grades [2].

To answer *RQ3*, we compared the frequency of access to learning materials and the final grades of the students in the four experimental groups. The results reveal that the students in the competition and social comparison groups demonstrated a significant increase in their views of learning materials compared to that of the social learning and control groups. The access frequency of these groups increased during the intervention compared to the weeks before introducing it. Incorporating the visualizations implementing the three persuasive strategies into the learning system motivated the students to compare and compete with their peers, translating into comparison and competition in their use of the learning resources and leading to a better performance in the final exam. The results are in line with social learning [9], competition, and social comparison theories [10], which suggest that people model their behaviour based on observable information. They seek opportunities to compete with others and engage in upward social comparison with better performing peers (role models) to improve themselves. This finding is consistent with previous research, which indicates that upward social comparison influences students' learning positively [10].

There was no significant difference between the viewing patterns of the competition and social comparison groups, but their academic performance revealed a significant difference. This may be because students have other means of learning outside of the system. The learning management system whose access logs we analyzed was just one of the many tools to aid learning, so the difference in academic performance might not be solely attributed to our intervention. Nevertheless, in comparing the three socially-oriented strategies, our analysis has shown that competition and social comparison most effectively promoted frequent access to learning materials and correlate with higher final grades. This implies that competition and upward social comparison should be preferred in the design of learning systems to improve their efficiency.

5 Conclusion

This study investigated the influence of three socially-oriented persuasive strategies on students' access to learning materials and performance. Understanding how online learning technology affects students' learning behaviour and performance has been the focus of research in computers and education. In this study, we analyzed system logs of 628 students that used a learning management system in accessing learning resources for a biology course in a blended learning system to find out if incorporating persuasive social visualizations affect the students' engagement with the learning materials

in terms of access frequency and their performance in terms of their final grades. The results revealed that the introduction of persuasive visualizations in the learning system increased students' access to the learning resources. Secondly, the increased access to learning resources influenced by the persuasive visualizations implementing the competition and social comparison strategies contributed to improved academic performance. Finally, these results suggest important advantages in supplementing online learning systems with the three persuasive strategies investigated in this research.

Acknowledgement. This research was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Program of J.Vassileva.

References

1. Wang, A.Y., Newlin, M.H.: Characteristics of students who enroll and succeed in psychology web-based classes. *J. Educ. Psychol.* **92**, 137–143 (2000)
2. Heffner, M., Cohen, S.H.: Evaluating student use of web-based course material. *J. Instr. Psychol.* **32**, 74–81 (2005)
3. Greer, J., Frost, S., Banow, R., Thompson, C., Kuleza, S., Wilson, K., Koehn, G.: The student advice recommender agent: SARA. In: *User Modeling and Adaptation Workshops (2015)*
4. Vassileva, J., Sun, L.: Evolving a social visualization design aimed at increasing participation in a class-based online community. *Int. J. Cooperat. Inform. Syst.* **17**, 443–466 (2008)
5. Orji, R., Moffatt, K.: Persuasive technology for health and wellness: state-of-the-art and emerging trends. *Health Inform. J.* **24**, 66–91 (2018)
6. Orji, F., Deters, R., Greer, J., Vassileva, J.: ClassApp: a motivational course-level app. In: *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018*, pp. 49–53. Institute of Electrical and Electronics Engineers Inc. (2019)
7. Orji, F.A., Vassileva, J., Greer, J.: Personalized persuasion for promoting students' engagement and learning. In: *Personalized Persuasive Technology Workshop Proceedings*, pp. 77–87 (2018)
8. Orji, F., Greer, J., Vassileva, J.: Exploring the Effectiveness of Socially-oriented Persuasive Strategies in Education. In: Oinas-Kukkonen, H., Win, K.T., Karapanos, E., Karppinen, P., Kyza, E. (eds.) *PERSUASIVE 2019*. LNCS, vol. 11433, pp. 297–309. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17287-9_24
9. Bandura, A.: *Social Learning Theory*, pp. 1–46. General Learning Corporation (1971)
10. Huguet, P., Dumas, F., Monteil, J.M., Genestoux, N.: Social comparison choices in the classroom: further evidence for students' upward comparison tendency and its beneficial impact on performance. *Eur. J. Soc. Psychol.* **31**, 557–578 (2001)



Sovereignty by Personalization of Information Search: A Collective Wisdom May Influence My Knowledge

Stefano A. Cerri^{1,2,3(✉)} and Philippe Lemoisson^{4,5}

¹ DKTS: Digital Knowledge Technologies Services, Via Ampère 61/A, 20131 Milan, Italy
sacerri@didaelkts.it

² FBK: Fondazione Bruno Kessler, Via Sommarive 18, 38123 Povo, Trento, Italy

³ LIRMM, Univ. Montpellier and CNRS, 161 Rue Ada, 34095 Montpellier, France

⁴ CIRAD, UMR TETIS, 34398 Montpellier, France

philippe.lemoisson@cirad.fr

⁵ TETIS, Univ Montpellier, AgroParisTech, CIRAD, CNRS, IRSTEA, Montpellier, France

Abstract. With the experiment that we outline in this paper, we have had the ambition to pave the way for addressing the problem of *supporting, enhancing and measuring collective AND informal learning, in particular serendipity*. We want to support a new type of free navigation on Web resources (Documents, Topics, Events and Agents – human and artificial -) that is driven by the learner’s current needs and the preferences of the community of trust chosen by the learner, not by external actors. The experiment exploits the ViewpointS Web Application (VWA) prototype, that restructures a private version of a subset of the Web according to personalized choices in order to determine *distances/proximities* among resources. The process allows to enable, empower and measure the *influence of members of the community of trust of the learner*, on the learner’s choices when navigating in search of THE resources corresponding to THE immediate need, goal, strategy, wish. In the following, we will outline: 1. *the rationale of our efforts* and 2. the user’s reactions during the phase of -formal and informal- learning the functions and use of the prototypical software environment VWA, i.e.: *a proof of concept for VWA*.

Keywords: Learning as a side effect of interactions · Collaborative and group learning · Personalized and adaptive learning environments · Recommender systems for learning

1 Introduction

Since a number of years, we work on a model, an approach, a paradigm called ViewpointS [1–6]. Recently, we also developed a system, called VWA (ViewpointS Web Application) than embodies the principles of the ViewpointS paradigm.

After a preliminary study concerned with the collaborative construction of ontologies [7], we have decided that simpler principles may better serve the process of structuring

Information in order to usefully retrieve it when knowledge is needed, in particular through interactions with peers.

We have assumed that the Web consists of four types of resources: Agents (human and artificial, i.e.: event-driven software programs), documents, topics and events. As examples: two authors may be more or less «professionally distant»; but also an author is more proximal to his/her own papers (documents) as to someone else's; to his/her topics of interest; a sub-topic is more proximal to its super-topic as two totally distinguished topics; similarly: a Conference - an event - is more proximal to its topics as to other ones. You may compute proximities/distances in various ways building a kind of «spatial, geographic representation» of the world, governed by distances in a graph.

In our approach, these distances are directly influenced by the community of trust, rather than by other “logical, algorithmic” rules, such as those adopted in numerous previous works typical of various kinds of recommender systems based on the Semantic Web [8–12]. This community of trust is what we consider the origin of collective wisdom, contributing to the personalization of the graph and thus the corpus of resources accessed by the user. We have basically adopted the recommendation [13] “*that the combination of visualization and recommendation techniques to empower users with actionable knowledge to become an active and responsible part-taker in the recommending process, instead of being the typical passive provider of just personal preferences and social connections*” is a necessary, even if perhaps not sufficient, condition for the personalization of informational processes. We have interpreted this vision as an encouraging mandate towards the integration of collective human and artificial intelligence. Further, we have also capitalized (see e.g.: [1, 6]) from the [14] that “*new user-centric directions for evaluating new emerging aspects in recommender systems, such as serendipity of recommendations, are required.*”

For us, this approach may represent a disruptive change of paradigm in many relevant processes of construction and access to Information (and therefore Knowledge) including the most relevant side effect: human informal learning. It is for us a strong assumption that “proximity” is a property known to facilitate learning [15] under the condition that proximity of resources depends on the dynamic behavior of the community of trust chosen by the learner. We are at the same time aware that the challenge we have adopted years ago is not yet demonstrated.

We have started to experiment VWA with a small but significant number of users. In this experiment we may distinguish two aspects: the user's (or learner's) reactions to the “new tool” during an indispensable initial phase of training and the effects of the new tool concerning understanding, discovering, learning using the new tool: informal learning [16] and social learning in a knowledge domain. Social learning consists of a kind of collective intelligence, where “*collective intelligence suggests that in certain settings, a group is better able to solve difficult problems than an individual working alone*” (see, for instance [17] in the crucial domain of medicine, or – even more generally: [18]). Notice that informal learning, even if it has no explicit learning objective, requires anyhow to solve the difficult problem that one should finally learn.

While the generic effects of VWA on informal and collective learning are described in another paper [19], the learner's reactions to the new tool (components, functions and use) are shortly described in this contribution.

An instance of context that seems to us relevant is offered by the compilation of a state of the art on some new, cross disciplinary research domain (e.g.: in translational research): it is evident that each researcher has his/her own preferences for collaborative filtering of too many and differently useful items, and that the best advisors are the human members of the trusted community of peers that (s)he has chosen.

2 Three Degrees of Personalization Supporting Sovereignty

In this section we outline the applicability of our approach to the goal as stated in the title, leaving details of the architecture and the algorithms to other contributions that have been quoted in the introduction.

The ViewpointS model relies on two concepts: *resources* and *viewpoints*. The *resources* are ‘Human Agents’, ‘Documents’, ‘Topics’, and ‘Events’. ‘Topics’ may be keywords or short expressions aimed at describing other resources. Each *viewpoint* is a connection between two *resources* established by a ‘Human Agent’ (or alternatively by an ‘Artificial Agent’). Both *resources* and *viewpoints* can be either extracted from the Web, or directly created by Human Agents. The *viewpoints* can be of five types; in this experiment, we concentrate on the two most important types: ‘factual’ and ‘subjective’. A factual *viewpoint* means that the semantics linking the two *resources* can be checked by others, e.g.: when a ‘Human Agent’ is the author of a ‘Document’ or when a ‘Human Agent’ participates to an ‘Event’. A subjective *viewpoint* means that the link indicates an emotion, an opinion or a belief of the emitter of the viewpoint, e.g.: when a ‘Human Agent’ likes a ‘Document’ or believes a ‘Document’ is relevant with respect to a ‘Topic’. The bipartite graph consisting of *resources* connected by *viewpoints* is called *Knowledge Graph* (KG).

Since this graph is too complex to be interpreted by humans, it is locally transformed in the neighborhood of a target *resource*, whenever a user is searching information, into a *Knowledge Map* (KM). This transformation is automatic and goes through the following process: i) the user chooses a *perspective* by choosing the respective strengths of the ‘factual’ versus the ‘subjective’ *viewpoints* (the rule may be more complex), ii) the *viewpoints* connecting the pairs of *resources* are valued and aggregated into ‘synapses’ reflecting proximities and iii) the labels indicating distances (inverse from the synapses strengths) appear on the KM edges between resources. The KG- > KM transformation is dynamic: whenever a member of the community updates the KG, the various KMs computed for the other members are impacted.

Figure 1 illustrates a KM computed around the neighborhood of the *resource* “topo uno”. The distances labeling the edges of the KM in the right part are SP-distances (shortest path distances). In the central part, the tabular view recapitulates the SP-distances and K-distances (pseudo-distances taking into account the multiplicity of possible paths) between each resource and the target “topo uno”.

In the current experiment, the only Agents that produce resources and viewpoints are the Human Agents. This is a temporary simplification: artificial Agents may fruitfully produce many more useful resources and viewpoints in subsequent applications of VWA by activating softbots instructed to reason on Web resources. This aspect of our model enables us to declare that our approach is synergic and complementary with

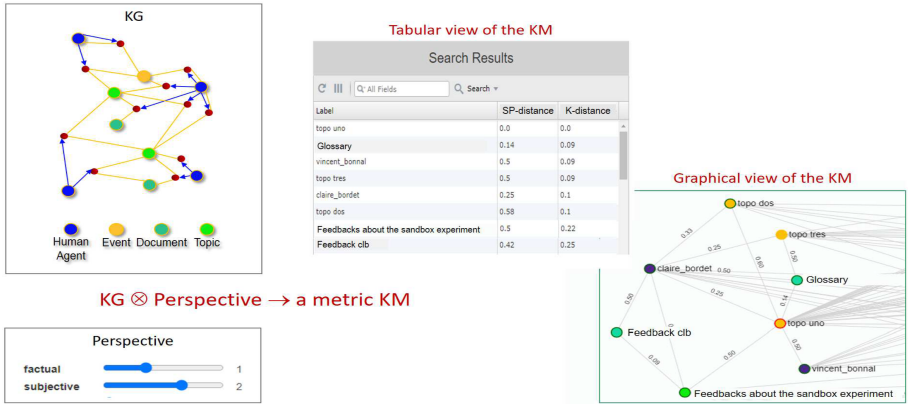


Fig. 1. The ViewpointS paradigm: 1. resources and viewpoints are stored in the KG (upper left), 2. a user chooses a perspective (down left) and searches for a resource (e.g.: “topo uno”), 3. a KM is computed in the neighborhood of “topo uno” and 4. this KM is displayed both in a table view (center) and in graphical view (right).

other ones available in the literature. The main differences with other models of access to Information (e.g.: Google) are:

- i. the whole set of resources available on the Web is exploited in order to build a *subset of “relevant and trusted” resources*, organized in a bipartite graph called **Knowledge Graph (KG)**. Notice that Agents (Human or Artificial) are first class *resources* in ViewpointS, in the same way as Documents or Topics or Events. The process of selection of resources by qualified Agents offers a *first degree of personalization*;
- ii. the User (or Learner) does not navigate on the KG, rather on a transformed graph, called **Knowledge Map (KM)** that is built *dynamically* -by means of a MapReduce transformation- according to a set of *preferences* (called a “perspective”) chosen by the Learner; viewpoints are weighted according to the preferences and then aggregated in binary links called “**synapses**” (adopting the metaphor of the brain [5]). The choice of preferences by Users offers a *second degree of personalization*;
- iii. the Learner may share with a community of trust (a group) the same KG in such a way that *other Agents may contribute* (dynamically) with *new resources and/or new viewpoints*, leading to the *strengthening or weakening of synapses*. The “**collective behavior/wisdom**” offers a *third degree of personalization* which engages the collective rather than the individuals.

3 VWA (ViewpointS Web Application): The SandBox Experiment

The process of learning “how to use” a new tool is not simple, for several reasons. The main problem is that if the tool is really new, it represents functionalities that are previously neither conceived nor acquired or mastered by the learner. Therefore, in order to expose our subjects to the “concepts” of the Information processes envisaged by the

ViewpointS model, we designed and exploited a “SandBox” where we have invited our subjects to follow us in a first introduction on “concepts and essential procedures”.

In the “SandBox” tutorial experiment we have tried to teach Users-Learners to feed Data (e.g.: documents and topics) to VWA and to structure the Information necessary for VWA in order to answer a query (i.e.: to add viewpoints). The challenges were multiple: i) to keep Users within a *pro-active learning process*, ii) to introduce Users *progressively* to the various features and functionalities of the prototype while leaving them discovering it at their own pace, iii) to record and assess their *positive but also their negative* reactions to the learning environment, and particularly to verify their *acceptance and preferences* with respect to the main innovation of VWA.

Any User entering VWA immediately becomes a *resource* of the type ‘Human Agent’ and, as such, will appear both in the tabular view and the graphical view of the KM as a blue node. All the learning resources are hosted by the KG “SandBox”. The three learning modules respectively named “topo uno”, “topo dos” and “topo tres” are *resources* of type ‘Event’; they aggregate *resources* of type ‘Document’: either videos or textual pedagogical documents. The three learning modules which introduce the ViewpointS paradigm are sketched hereafter:

Learning module n°1: discovering the environment. Users are firstly invited to listen to a 4’30” clip presenting the ViewpointS model, then to follow a clip teaching them how to connect themselves to a learning module;

Learning module n°2: the basics for proactivity. Users are taught how to create a new resource, how to connect two resources (a viewpoint), and finally how to connect a preview, i.e.: an image intended to give a hint before opening the resource;

Learning module n°3: understanding the underlying processes. Users are explained the importance of the “perspective”, they go through the notions of “SP-distance” and “K-distance”; they learn how to emit reactive viewpoints and finally discover how “shortcut viewpoints” enhance serendipity [see, e.g.: 1, 6].

The educational paths followed by the Users are hosted by the KG “SandBox” as well. As soon as a User has read (or listened to) the documents linked to a module, (s)he is asked to establish a connection, i.e. to emit a factual viewpoint, between him/her and the corresponding module. This appears in Fig. 2 which is the view of the KG “SandBox” taken at the end of the process: the 36 participants (blue nodes) are connected to the three modules (orange nodes) which aggregate a Glossary and the 9 pedagogical documents (green nodes).

The pedagogy of the VWA SandBox intertwines therefore three learning modes: i) learning through documents, i.e.: “classical” knowledge acquisition through *resources*, ii) learning by doing, i.e.: creating resources and viewpoints and iii) participating to collective learning by reshaping the KMs browsed by the others.

Among the 55 initial volunteers, 36 people actually took the time to go through the three modules, as illustrated in Fig. 2, despite the heavy time schedules and constraints of the autumn 2020. This could be interpreted as a relative success with respect to the challenge of keeping users within a *pro-active learning process*.

Figure 3 illustrates this proactivity; in the middle of the KM, we can see Users (Human Agents) subjectively connected to the ‘Documents’ they have appreciated. Note that the chosen perspective (‘subjective’ only) is orthogonal to the perspective of Fig. 2 (‘factual’

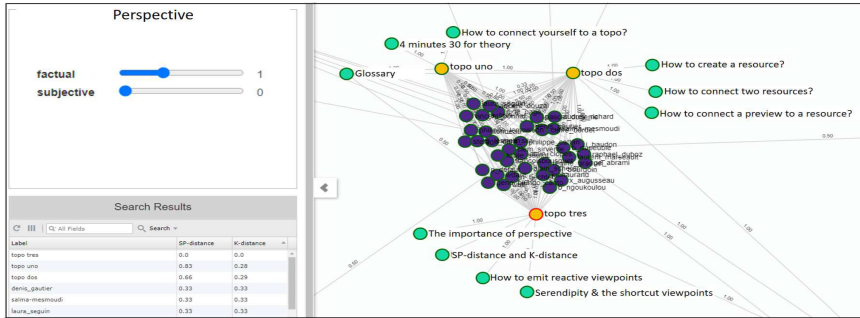


Fig. 2. A view on the three learning modules in the SandBox (resource type = “Event”; colour = orange), the 10 documents linked to them (resource type = “Numeric Document”; colour = green), and the 36 active Users (resource type = “Human Agent”; colour = blue). The chosen perspective selects the factual connections and discards the subjective ones. (Color figure online)

only) so that the map illustrates opinions about the content rather than participation in the modules.

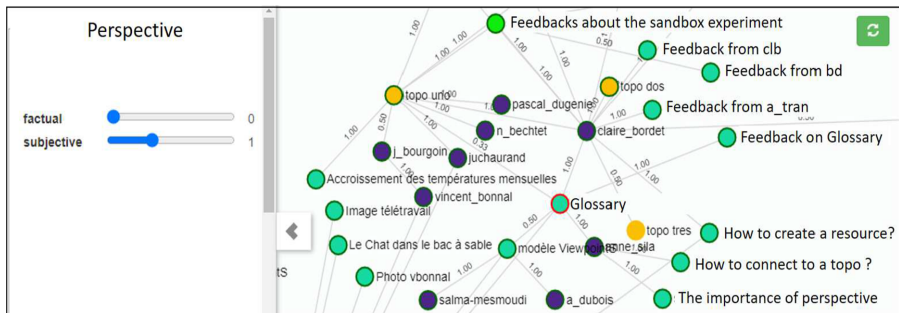


Fig. 3. A view illustrating the proactivity of the participants. The chosen perspective selects the subjective connections and discards the factual ones.

Assessment of the Learning Modules. Almost all participants acknowledged clarity of the pedagogical documents and easiness in the progression.

Assessment of the VWA Environment. The participants had been firstly invited to contribute with free comments to a specific ‘Topic’ named “Feedbacks about the SandBox experiment”. These comments pointed out several points concerning the environment: *a.* some users were disappointed not to be able to *suppress viewpoints* they had created; *b.* most users observed difficulties in exploiting the KM as soon as it became dense; *c.* several users asked for a “global view” of the whole KG, *c.* several users asked for special means to find back the resources created by themselves; *d.* one user asked for a special feature allowing the batch import of documents; *e.* one user asked for a shortcut grouping the actions of creating a new resource AND connecting it to an existing one; *f.*

several users asked for a special feature facilitating contextualized comments on existing resources. In addition to those free comments, the participants were asked to rate from 1 (worst) to 5 (best) the two alternative views on the KM (tabular and graphical). This survey led to the following: tabular view: mean rating = 2,2; standard deviation = 0,80; graphical view: mean rating = 4,1; standard deviation = 0,53.

4 Conclusion

We make reference to [20] in order to qualify our SandBox experiment as a proof of concept of several rather radical changes in the collective construction and retrieval of knowledge. Referring to the goal of this paper indicated by its title, we believe to have proved several concepts: 1. Users exploit the graph representation with relative ease and increasing interest; 2. the proximity introduced by “synapses” in the KM is a useful means for aggregating resources and influencing the Users’ navigation; 3. the three levels of personalization favor not only the trust of Users, but also their protection from external undesired influences (sovereignty); 4. the exploitation of collective wisdom by a trusted community allows to privilege shared values, interests, goals and knowledge; 5. learning the use of VWA in the SandBox has been a relative success, even if many suggested improvements of the current VWA platform will require to engage significant energy in the months to come.

References

1. Cerri, S.A., Lemoisson, P.: Tracing and enhancing serendipitous learning with ViewpointS. In: Frasson, C., Kostopoulos, G. (eds.) *Brain Function Assessment in Learning*. LNCS (LNAI), vol. 10512, pp. 36–47. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-67615-9_3
2. Lemoisson, P., Surroca, G., Jonquet, C., Cerri, S.A.: ViewpointS: when social ranking meets the semantic web. In: *FLAIRS 17 Conference, Proceedings of Florida Artificial Intelligence Research Society Conference, North America* (2017)
3. Lemoisson, P., Rakotondrahaja, C.M.H., Andriamialison, A.S.P., Sankar, H.A., Cerri, S.A.: VWA: ViewpointS web application to assess collective knowledge building. In: Nguyen, N.T., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B. (eds.) *ICCCI 2019*. LNCS (LNAI), vol. 11683, pp. 3–15. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28377-3_1
4. Lemoisson, P., Surroca, G., Jonquet, C., Cerri, S.A.: ViewPointS: capturing formal data and informal contributions into an evolutionary knowledge graph. *Int. J. Knowl. Learn.* **12**(2), 119–145 (2018)
5. Lemoisson, P., Cerri, S.A.: ViewpointS: a collective brain. In: Frasson, C., Bamidis, P., Vlamos, P. (eds.) *BFAL 2020*. LNCS (LNAI), vol. 12462, pp. 34–44. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60735-7_4
6. Cerri, S.A., Lemoisson, P.: Serendipitous learning fostered by brain state assessment and collective wisdom. In: Frasson, C., Bamidis, P., Vlamos, P. (eds.) *BFAL 2020*. LNCS (LNAI), vol. 12462, pp. 125–136. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60735-7_14
7. Lemoisson, P., Cerri, S.A.: Interactive knowledge construction in the collaborative building of an encyclopedia. *Appl. Artif. Intell.* **19**(9–10), 933–966 (2005). (Taylor & Francis, Special issue on Learning Grid Services)

8. Alaa, R., Gawich, M., Fernández-Veiga, M.: Personalized recommendation for online retail applications based on ontology evolution. In: ACM ICCTA 2020, Antalya, Turkey, 14–16 April (2020)
9. Musto, C., Lops, P., deGemmis, M., Semeraro, G.: Context-aware graph-based recommendations exploiting personalized PageRank. *Knowl.-Based Syst.* **216**, 106806 (2021)
10. Obeid, C., Lahoud, I., Khoury, H., Champin, P.-A.: Ontology-based recommender system in higher education. In: *The Web Conference Companion (WWW 2018)*, Lyon, France (2018)
11. Kotkov, D., Wang, S., Veijalainen, J.: A survey of serendipity in recommender systems. *Knowl.-Based Syst.* **111**, 180–192 (2016)
12. Pandey, G., Kotkov, D., Semenov, A.: Recommending serendipitous items using transfer learning. In: *CIKM 2018 Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1771–1774. ACM Press (2018)
13. Verbert, K., et al.: Context-aware recommender systems for learning: a survey and future challenges. *IEEE Trans. Learn. Technol.* **5**(4), 318–335 (2012)
14. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: state of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) *Recommender Systems Handbook*, pp. 73–105. Springer, Bostond (2011). https://doi.org/10.1007/978-0-387-85820-3_3
15. ZPD (Vygotsky). https://en.wikipedia.org/wiki/Zone_of_proximal_development. Accessed 18 Mar 2021
16. Breuker, J., Cerri, S.A.: Learning as a side effect. In: Seel, N.M. (ed.) *Encyclopedia of the Sciences of Learning*. Springer, Boston (2012). https://doi.org/10.1007/978-1-4419-1428-6_59
17. Tucker, J.D., Day, S., Tang, W., Bayus, B.: Crowdsourcing in medical research: concepts and applications. *Peer J.* **7**, e6762 (2019). <https://doi.org/10.7717/peerj.6762>
18. Muthukrishna, M., Henrich, J.: Innovation in the collective brain. *Phil. Trans. R. Soc. B* **371**, 20150192 (2016). <https://doi.org/10.1098/rstb.2015.0192>
19. Lemoisson, P., Cerri, S.A., Douzal, V., Dugénie, P., Tonneau, J.-P.: Collective and informal learning in the viewpoints interactive medium. *Information 2021.* **12**(5), 183 (2021). <https://doi.org/10.3390/info12050183>
20. Academic Careers for Experimental Computer Scientists and Engineers: to be downloaded from <https://www.nap.edu/read/2236/chapter/1>. Accessed 18 Mar 2021

Games and Gamification



Confusion Detection Within a 3D Adventure Game

Mohamed Sahbi Benlamine^(✉) and Claude Frasson

Computer Science and Operational Research Department, University of Montreal, Montreal, Canada

ms.benlamine@umontreal.ca, frasson@iro.umontreal.ca

Abstract. In this study we built a deep-learning model based on EEG data to recognize the confusion of the player. The model was constructed from the EEG data of 20 participants and their confusion measured using a camera-based emotion recognition system (Facereader 7.1) while playing adventure 3D game. We asked the participants to identify their emotions while playing the game using a menu always displayed in the interface. This paper presents a confusion recognition model based on EEG features that can be used in levels of confusion detection. Results show that we can detect the level of confusion with high accuracy (94.8% accuracy for four confusion levels). We discussed about our results and the potential applications of such model (for entertainment or education purposes...).

Keywords: Confusion · Games · Physiological data · EEG · AI models

1 Introduction

In different situations, especially in learning [1], people can experience the emotion of confusion. There is a variety of educational games ranging from immersive, 3D virtual worlds [2] to puzzle games [3] that generate confusion. The learning experience is influenced by the learner's emotions. For example, when a learner is confused, he can make erroneous decisions, bad performance and finally disengage.

In fact, confusion under certain level may favorize engagement, but it may lead to frustration and boredom if there is more confusion and no understanding. The confusion can lead to frustration or boredom [4]. Especially for people with dementia [5], confusion is more observable since the decline of their cognitive abilities. So, the earlier the source of confusion is identified, the more efficient the treat or the help for them. Therefore, in many situations it is important to detect confusion, for example in car driving confusion can lead to accident [6].

Therefore, we conducted an experiment on 20 participants playing an open space 3D game that generates confusion when they get lost. We recorded their facial expressions and the EEG signals. Our hypothesis is that we can create an effective EEG-based model to detect four levels of confusion (from not confused to very confused). To train our model, we used the EEG signals as input and the levels of confusion extracted from facial expressions with FaceReader 7.1 as output. The Facereader program provides

objective results because facial expressions measures are based on the standardized Facial Action System (FACS) [7]. Each expression corresponds to a certain pattern of Action Units (facial muscle movement). Therefore, a trained model on classes that derive from these facial expressions measures provides objective results that are independent from the game environment. So, the generated model will be able to predict the confusion level only from EEG signal and can be used in different environments like e-learning context and VR/AR systems without any further training to the new environment. The generated model is very useful in VR/AR situation where the user's face is hidden with a VR headset and the use of the camera is impossible, so we use the EEG data to detect the confusion.

2 Confusion Recognition

There are different studies about confusion detection using EEG signals, but they deal with binary classification (2 classes: no-confusion/confusion). In the medical field, a study [8] used electroencephalography (EEG) to identify fluctuating confusion with patients with dementia. In the Educational field, an affective model with several emotion categories (boredom, confusion, engagement, and frustration) was build with EEG data from sessions on a modified version of the Wisconsin Card Sorting Test [9]. The accuracy of the confusion classifier was less than 50%. Other study [10] used a one-channel Mind-Set EEG headset to detect confusion in ten adults watching videos of Massive Open Online Courses. The best classifier achieved 57% accuracy. The authors [10] have used this dataset in more recent article [11] with a Bidirectional Long Short-Term Memory (Bi-LSTM) neural model that achieved 75% accuracy. Other studies have been conducted to detect confusion by combining EEG signals with audio-visual sources. For example, the Sedmid model [12] detects confusion by combing EEG with video features with an accuracy of 87.8%.

In this study, we build a model that corresponds EEG signals with facial expressions data to detect confusion precisely and objectively. Facial expressions could be measured by facial EMG (Electromyography). However, this method is more intrusive compared to camera based Facial expression system, because of the use of electrodes placed on the face of the participant to measure his facial muscles activities (EMG). With our EEG-based confusion detection model, we not only predict whether an individual is confused or not but also his level of confusion. We obtained the confusion with multinomial classification (multiclass: no confusion, low, medium and high confusion).

3 Experimental Settings

3.1 Equipment

iMotions

iMotions is a platform for multi-modal studies that ensures equipment synchronization.

Here, we wanted the synchronization of the facial units from the webcam and the EEG signals.

Emotiv Epoc

The Emotiv Epoc headset comprises 16 electrodes (14 channels and 2 references behind the ears). The electrodes are positioned according to the international 10–20 system at AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4. The EEG signals are in μ Volt with a sampling rate of 128 samples/s and frequencies' range between 0.2–45 Hz.

FaceReader 7.1

FaceReader 7.1 program recognizes besides the seven facial expressions of primary emotion, three secondary emotions (confusion, boredom, and interest) using real-time frame-by-frame analysis of the user's face via a webcam-based on face muscles movement. Facereader's resulted file includes the following emotion categories with values between 0 and 1: neutral, happy, anger, sadness, surprise, fear, disgust, arousal, confusion, boredom, and interest. FaceReader also provides the valence, which indicates whether the person is in a negative or positive emotional state. The valence values are between -1 and 1 . In this study, we focused only on confusion data.

3.2 Experimental Protocol

We recruited Twenty undergraduate students (7 women, 13 men) from the computer science Department of the University of Montreal to participate in our experiment. They ranged in age from 21 to 35 years old. Twenty percent of them reported themselves as "gamers". As this research study was approved by the ethical comity, all participants signed a consent form before beginning the experiment. Then the experimenter had the participants sit on a chair and checked the chair so that they maintained a good view on the computer screen. Then the experimenter setup the Emotiv Epoc and started the iMotions software to ensure the synchronization of the EEG with the webcam. Prior to the participants playing, the experimenter checked the contact quality of the EEG electrodes. Once the participant finished playing, he gets compensated \$20 for participation and debriefed at the end. The experiment session took approximately one hour. After each session, we collected the synchronized data (face recordings and EEG signals). We passed the participant's face video to Facereader 7.1 software to extract emotional values.

3.3 Danger Island Game

The environment Danger Island (Fig. 1) is an adventure game where the player encounters a lot of enemies (zombies, wild animals, and machine guns). The player's mission is to find fuel cans and go back to a helicopter to escape the island. Player has to find orientation within the game and may experience confusion and frustration to find their way. The game's difficulty level depends on the player's category (gamer/non-gamer) selected in the game's start menu.



Fig. 1. Danger island environment

During the game session, the user was invited to select his actual emotional state among 12 emotions categories¹ in the upper right corner menu. These choices are recorded in the log file with the current time to be analyzed later.

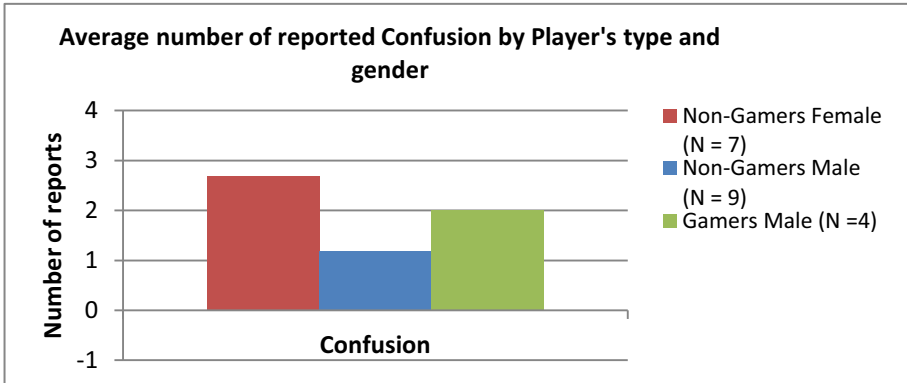


Fig. 2. The average of reported confusion during game session by player's category and gender

The figure above (Fig. 2) illustrates the average self-reported confusion by player's category and gender. We can note that female non-gamers in average experienced more confusion than male gamers and non-gamers.

4 Method

4.1 Building Dataset

The Input EEG Signals

For each participant, we obtained a CSV file from iMotions with a sample rate of 128 samples/s for all 14 EEG electrodes (Fig. 3). Finally, after extracting all the values from the CSV files, the size of the inputs of EEG data was $28057 \times 14 \times 128$.

¹ Self-report's emotion categories: Confusion, Surprise, Frustration, Fear, Boredom, Sadness, Anger, Engagement, Flow, Excitement, Joy and Calm.

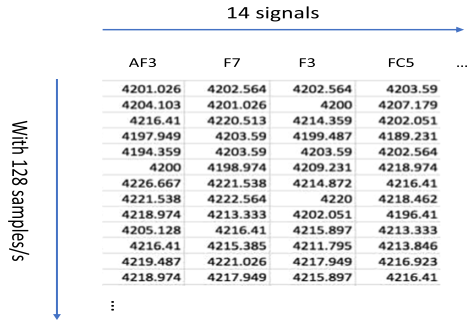


Fig. 3. A sample file containing the EEG signals

The Output Labels Obtained with FaceReader

The second step was to use videos of participants’ faces with FaceReader 7.1 to get confusion values for the whole game sessions. We developed a java program that took our 14 first-in-first-out queues of size 128 as mobile windows of 1 s of EEG data from each electrode (each window contained 128 samples). For each FaceReader frame time (every 1/6 s), the program recorded in a separate CSV file: the content of each EEG windows, the confusion intensity value (between 0 and 1) and the corresponding confusion level (no confusion [0, 0.2], low [0.2, 0.4], medium [0.4, 0.6] and high level of confusion [0.6, 1]). The Fig. 4 illustrates the entire experimental settings designed for the construction of our dataset for EEG-based confusion recognition. At the end we obtained one second of EEG signals (vector of size 14 × 128) associated to one level of confusion.

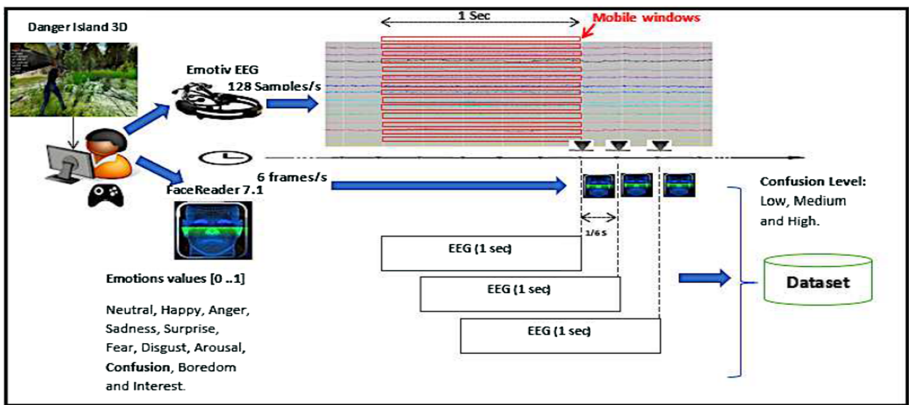


Fig. 4. Experimental settings and data extraction

4.2 Models Training

We trained three models for multinomial classification (multiclass): a support vector memory (SVM), K-nearest neighbors (KNN) and a long short-term memory (LSTM).

We split the data into 3 sets: 60% train set, 20% validation set, and 20% test set. Because we had computational limits, and cross-validation proved to be too time-consuming.

SVM

We first used a support vector machine (SVM) because it is very popular in BCI applications [13] and its robustness against non-linear data. Moreover, it is efficient in high dimensional space. The SVM algorithm uses a technique called kernel trick to transform the data. It then separates the data according to their classes. For data that cannot be separated linearly, the kernel trick is used. We put the data in a space of higher dimensions to make them linearly separable.

As SVM has a high training time complexity, we had to train our models with a reduced number of features. We used `sklearn.decomposition.PCA`² and set the percentage of variance we wanted to keep on our data to 99%.

$$pca = PCA(n_components = 0.99, whiten = True)$$

$$X = pca.fit_transform(X)$$

PCA reduced the features from 1792 (14×128) to 49. We also reduced the size of the dataset to 30% to speed up the process.

We used the `sklearn.svm.SVC` implementation of SVM. To tune the regularization parameter C , the kernel and the kernel coefficient gamma we used the Grid Search. We tested values for C between [0.01, 0.0001], values for the kernel between [poly degree 3, poly degree 4], values for gamma between [1, 100].

KNN

We chose KNN algorithm because it is fast in training and only has a few parameters to tune (number of neighbors). KNN classifies a given data point according to the majority of its k closest neighbors.

We used the `sklearn`'s KNN implementation (`KNeighborsClassifier`). To tune the k hyperparameter, we used the class `sklearn.model_selection.GridSearchCV`. The Grid Search tests a given set of values for each specified hyperparameter and gives the accuracy of the model at each test. We tested values of k between 1 and 25.

LSTM

The long short-term memory (LSTM) neural network is often used with time series [14] because of its memory capacity. Therefore, LSTM looked promising with our EEG signals as it can learn over a sequence of events to predict the next one.

The first step of the LSTM is to decide which information in the previous memory cell is not relevant so that it can be deleted. The next step is to choose among the new incoming data x_t , the relevant ones, and to store them in the memory C at timestamp t . The last step is the selection of the necessary values in the memory according to the desired result. The result is a filtered version of the memory. For example, in the case of language, if the model has just seen a subject, it may want to select the relevant information for a verb.

² <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>.

We used the `tf.keras.layers.LSTM` implementation of LSTM. To tune the number of layers, the number of neurons, the batch size, and the number of epochs, we used the Grid Search. We tested values for the number of hidden units between [50, 500], values for the size of the batch between [128, 3000], values for the number of epochs between [100, 300].

5 Results

In this final section, we compare the results of multiclass classification on validation set with our SVM, KNN, and LSTM. We also compare our results on test set with those of Yang et al. [12] which represent the state of the art for the binary classification of the accuracy of confusion. We achieved all our results with a dataset split of 60/20/20 and subset accuracy metrics.

We first evaluated our SVM model with different configurations of features, dataset sizes, and hyperparameters as described in Table 1. The size of the dataset and the choice of the kernel seem to be the most significant parameters for accuracy with the SVM model. The kernel coefficient *gamma* has also allowed to improve the accuracy in a consequent way. We had the best accuracy of 80.9% on the validation set, using the EEG full dataset with a *polynomial* kernel of *degree* 3, *gamma* = 10 and *C* = 0.001. We were limited by hardware limitations to test other configurations with the full dataset and the EEG signals.

Table 1. Accuracy on validation set of SVM model

Features	Dataset	Kernel	Gamma	C	Accuracy
PCA 49 features	30% of original	poly, degree 3	1	0.001	64.0%
PCA 49 features	30% of original	rbf	10	0.001	38.8%
PCA 49 features	30% of original	poly, degree 3	10	0.001	69.0%
PCA 49 features	30% of original	poly, degree 3	10, 100	0.01, 0.001	68.8%
PCA 49 features	30% of original	poly, degree 3	10	0.0001	69.4%
PCA 49 features	30% of original	poly, degree 3	10	0.00001	68.0%
PCA 49 features	30% of original	poly, degree 4	10	0.0001	69.4%
PCA 49 features	Full dataset	poly, degree 3	10, 15, 100	0,0001, 0.001	78.9%
EEG signals	Full dataset	poly, degree 3	10	0,0001, 0.001	80.9%

We then switched to the KNN model, which had better time complexity allowing us to use all the dataset and the original features. We tested the KNN model with the Euclidean distance and a neighbor number between 1 and 25. As showed in Fig. 5, 1-NN obtained the best accuracy of 96.3% on the validation set. To make sure we did not overfit, we take $k = 5 \approx \log(\text{number of samples})$ as recommended in statistics³,

³ <https://stats.stackexchange.com/questions/384542/how-to-prevent-overfitting-with-knn>.

we obtained 92.08% accuracy. We also did a 5-fold validation with $k = 5$ and obtained 89.27% accuracy. This indicates that the validation data has a high similarity with the training data. The boundaries between the classes are also very explicit.

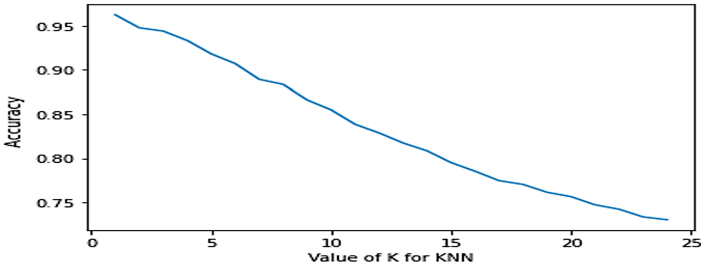


Fig. 5. Accuracy on validation set of KNN model for k between 1 and 25.

Table 2. Accuracy on validation set of LSTM model with EEG signals

Number of neurons	Batch size	Number of epochs	Accuracy
50, 75	128	100	86.2%
75	2000	100	88.4%
100	2000	100	90.3%
200, 300	2000, 3000	100	93.1%
300	3000	100	93.4%
300	2000, 3000	200, 300	94.4%
350	2000	300	94.6%
360	2000	300	94.8%
400	2000	300	94.7%
500	2000	300	94.6%

We then trained an LSTM because the model is known for its performance with the times series [15]. Another advantage is that it learns patterns from data on the contrary of KNN that compute distances. We configured the number of neurons, the batch size, and the number of epochs as showed in Table 2 The number of neurons and the number of epochs seem to be the most significant parameters for accuracy with the LSTM model.

We got the best accuracy of 94.8% on the validation set, using 360 hidden units, a batch of size 2000, and 300 epochs. Figure 6 shows the learning curve.

Finally, we tested our models on the test set and compared them with state-of-the-art Yang et al. [12] (see Table 3). Our LSTM model exceeded state of the art and achieved 94.8% accuracy on the test set. Figure 7 shows its confusion matrix.

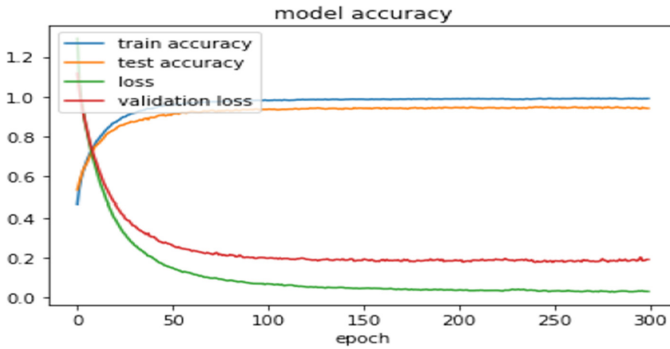


Fig. 6. Learning curve with an LSTM model

Table 3. Overview of best classifiers accuracy on test set

Method	Accuracy
Yang et. al. [12]	87.8%
Our SVM	81.9%
Our KNN	92.08%
Our LSTM	94.8%

<class 'tensorflow.python.keras.engine.sequential.Sequential'>
Accuracy: 0.948

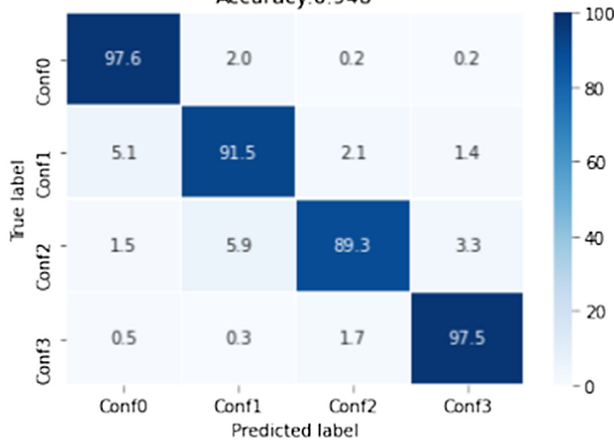


Fig. 7. The LSTM confusion matrix on the test set

6 Discussion and Conclusion

We wanted to see if we could predict four levels of confusion from not confused to highly confused. We used FaceReader 7.1 to get the confusion labels, which, by analyzing facial

expressions, gave accurate values at a high sampling rate. We could not continue tuning our SVM because we had hardware constraints. The KNN and the LSTM algorithms achieved high and close accuracy. The question then arises as to which one to choose.

KNN computes the distances and, therefore, does not find patterns in the data. The only information KNN gives is that the training data form clusters and that the examples of the same class are close in the feature space. KNN, therefore, requires representative training samples because it cannot abstract and learn patterns. If the data is exposed to certain transformations that change distances, KNN can lose its efficiency. Another of its constraints is that it always needs all the data to make a new prediction. On the other hand, neural networks learn patterns on the data. Thus, they can be more appropriate for real-time data that may be isolated from the other clusters. In our case, the LSTM model was faster to train. We believe that it will be an efficient model to detect confusion in real-time afterward.

EEG headsets are becoming more popular [16] because they offer convenient design for real situation, easy interconnexion with mobile devices and more accessible prices. These devices are already used by players to control their games⁴ and enhance their user experience. As these devices are beneficial⁵ in therapeutic and senior mental health applications, they can fit in school to be used by students to monitor their mental performance and give the teacher more insight about the mental and emotional state of his class to adapt his pedagogical strategies and provide maximum understanding.

In conclusion, Confusion recognition is very important, particularly to effectively adapt a system or a care to the user. This innovative study demonstrates the feasibility of multiclass classification of confusion for four levels of intensity. In addition, our LSTM model for classifying levels of confusion reached 94.8% accuracy. The results exceed those of the state of the art. It would be interesting for a future study to analyze whether confusion led to engagement or frustration and find a way to predict if confusion is likely to have a positive or negative outcome. Another interesting aspect may be to develop a model capable of predicting confusion in real-time.

Acknowledgments. We acknowledge NSERC-CRD, BMU and Prompt for funding this research.

References

1. D’Mello, S., Lehman, B., Pekrun, R., Graesser, A.: Confusion can be beneficial for learning. *Learn. Instr.* **29**, 153–170 (2014)
2. Taub, M., Mudrick, N.V., Azevedo, R., Millar, G.C., Rowe, J., Lester, J.: Using multi-channel data with multi-level modeling to assess in-game performance during gameplay with Crystal Island. *Comput. Hum. Behav.* **76**, 641–655 (2017)
3. Shute, V.J., et al.: Modeling how incoming knowledge, persistence, affective states, and in-game progress influence student learning from an educational game. *Comput. Educ.* **86**, 224–235 (2015)

⁴ www.next-mind.com.

⁵ www.neuro.com/.

4. Arguel, A., Lockyer, L., Lipp, O.V., Lodge, J.M., Kennedy, G.: Inside out: detecting learners' confusion to improve interactive digital learning environments. *J. Educ. Comput. Res.* **55**(4), 526–551 (2017)
5. Berry, B.: Minimizing confusion and disorientation: cognitive support work in informal dementia caregiving. *J. Aging Stud.* **30**, 121–130 (2014)
6. Beanland, V., Fitzharris, M., Young, K.L., Lenné, M.G.: Driver inattention and driver distraction in serious casualty crashes: data from the Australian National Crash In-depth Study. *Accid. Anal. Prev.* **54**, 99–107 (2013)
7. Ekman, P., Friesen, W.V.: Constants across cultures in the face and emotion. *J. Pers. Soc. Psychol.* **17**(2), 124 (1971)
8. Walker, M.P., et al.: The clinician assessment of fluctuation and the one day fluctuation assessment scale: two methods to assess fluctuating confusion in dementia. *Br. J. Psychiatr.* **177**(3), 252–256 (2000)
9. Mampusti, E.T., Ng, J.S., Quinto, J.J.I., Teng, G.L., Suarez, M.T.C., Trogo, R.S.: Measuring Academic Affective States of Students Via Brainwave Signals. IEEE, City (2011)
10. Wang, H., Li, Y., Hu, X., Yang, Y., Meng, Z., Chang, K.-m.: Using EEG to Improve Massive Open Online Courses Feedback Interaction. City (2013)
11. Wang, H., Wu, Z., Xing, E.P.: Removing Confounding Factors Associated Weights in Deep Neural Networks Improves the Prediction Accuracy for Healthcare Applications. World Scientific, City (2018)
12. Yang, J., Wang, H., Zhu, J., Xing, E.P.: Sedmid for confusion detection: uncovering mind state from time series brain wave data. arXiv preprint [arXiv:1611.10252](https://arxiv.org/abs/1611.10252) (2016)
13. Garcés, M.A., Orosco, L.L.: Chapter 5 – EEG Signal Processing in Brain–Computer Interface. Academic Press, City (2008)
14. Lipton, Z.C., Kale, D.C., Elkan, C., Wetzel, R.: Learning to diagnose with LSTM recurrent neural networks. arXiv preprint [arXiv:1511.03677](https://arxiv.org/abs/1511.03677) (2015)
15. Soleymani, M., Asghari-Esfeden, S., Fu, Y., Pantic, M.: Analysis of EEG signals and facial expressions for continuous emotion detection. *IEEE Trans. Affect. Comput.* **7**(1), 17–28 (2015)
16. LaRocco, J., Le, M.D., Paeng, D.-G.: A systemic review of available low-cost EEG headsets used for drowsiness detection. *Front. Neuroinform.* **14**, 42 (2020)



Representation of Generalized Human Cognitive Abilities in a Sophisticated Student Leaderboard

Christos Troussas^(✉), Akrivi Krouska, Filippos Giannakas, Cleo Sgouropoulou,
and Ioannis Voyiatzis

University of West Attica, Egaleo, Greece

{ctrouss, akrouska, fgiannakas, csgouro, voyatzeri}@uniwa.gr

Abstract. Gamification is a popular method for enhancing learners' motivation and thereby strengthening learning efficiency. An example of gamification is the leaderboard, namely an approach showing a ranking of students. Although leaderboards are currently implemented in various domains, previous studies reported that they are solely based on one student characteristic, such as grade. This paper presents a sophisticated leaderboard showing a more reliable ranking of students. This leaderboard is available to both instructors and learners; instructors can be adequately informed by this ranking and redesign their teaching strategies, while learners can be motivated by the ranking and try more to advance their knowledge. The sophistication of this leaderboard lies in the employment of the Weighted Sum Model (WSM), which is the best-known multi-criteria decision analysis technique and responsible for evaluating a number of alternatives in terms of a number of decision criteria. The input of WSM is multiple learners' characteristics, including current and previous knowledge, interaction time and frequency of misconceptions, so that a more robust representation of students is achieved. Our presented model was incorporated in an intelligent tutoring system for the computer programming language C#, and the evaluation results show high accuracy in the values of the leaderboard.

Keywords: Gamification · Intelligent Tutoring System · Leaderboard · WSM

1 Introduction

To be adapted to the needs of 21st century and even more to the changes, provoked by the COVID-19 pandemic, the field of education has had to be revamped and modified based on the preferences and needs of learners [1]. As a result of the digital transition, the lack of student participation in the educational software and the lack of motivation to students to study have become a vital issue in contemporary education. It is in these challenges that gamification can help the implementation of educational software [2]. Gamification refers to the inclusion of elements of the game, in a way not relevant to the game itself. The field of gamification embodies features intended to encourage and inspire the accomplishment of the mission of producing entertainment encounters and growing involvement in particular tasks [3], aiming to advancing students' cognitive skills [4].

In addition to badges and points, leaderboard is another example of the most common gamification techniques that can cause various forms of interaction and can theoretically connect to different performance in the development of skills and transfer of information [5]. Since the leaderboard will have many possible targets, the user is encouraged to pursue the target. In general, the leaderboard is a gamification technique that can represent a list of competitors in a competition context. The player list is sorted with respect to a metric, such as the highest to the lowest scores. It needs to be noted that this game system encourages a competitive situation that allows players to equate their success with others regarding a certain challenge or mission. There are at least three reasons why leaderboards in various domains are so common and effective [6]. First of all, players of varying ages are all familiar with the leaderboard idea. Second, the leaderboard makes it transparent to participants that success and loss are exploited. Third, by exposing them to social contrast, it ultimately motivates the player. There is no question, however, that the use of the leaderboard is a traditional technique that can inspire the person to better advance his/her knowledge.

Leaderboards can have great pedagogical potential [2, 3, 5], since they can motivate students to try more to achieve higher grades. In the related literature, there have been several efforts that explore the effect of leaderboards in the context of e-learning [1–12]. The results of these researches show that the students who could view the leaderboard and could attain badges finally achieved higher scores than other students only in some assignments. Also, leaderboards can provide a positive result which was related to the overall time spent working on the assignment. It, also, allows competitions to be established as a means to promote social interaction between users. However, based on the aforementioned literature, the list of players in a leaderboard is ordered regarding to one variable, commonly such as highest to the lowest scores.

In view of the above, this paper presents a sophisticated leaderboard which creates a ranking of students, being available to instructors and learners; instructors can be adequately informed by this ranking and redesign their strategy and learners can be motivated by the ranking and try more to advance their knowledge. The sophistication of this leaderboard lies in the employment of the Weighted Sum Model (WSM), which is the best known multi-criteria decision analysis (MCDA) evaluating a number of alternatives in terms of a number of decision criteria. The input of WSM is multiple learners' characteristics, including current and previous knowledge, interaction time and frequency of misconceptions, so that a more robust representation of students is delivered. Our presented model was incorporated in an intelligent tutoring system for the computer programming language C#, and its evaluation results show the accuracy of the leaderboard.

2 Construction of the Automated Leaderboard

2.1 Learners' Characteristics

For the construction of the automated leaderboard, the system takes into account the following learners' characteristics, which have been reported as important in the related scientific literature [13–15]:

- Grade (G): The grade of the students reflects their current knowledge level. It is the average of the grades that a student has achieved in all the tests s/he has tried. This characteristic can take values from 0 (lower) to 10 (higher).
- Previous knowledge level (PKL): The previous knowledge level reflects the knowledge of the student that s/he has before the first interaction of the system. The previous knowledge level takes values from 0 (lower) to 10 (higher) and is determined by a preliminary test that a student takes prior to his/her interaction with the system.
- Interaction time (InT): This characteristic refers to the time that a student dedicates to use the system. Interacting with the systems means spending time on studying the educational material or taking an assessment. It derives from the log files of the system and can take integer values showing the hours spent using the system per day on average.
- Frequency of mistakes (FM): This characteristic refers to the frequent or repeated errors which are made by a specific student. It derives from the log files of the system and can take integer values showing the number of mistakes per test on average. Since the domain knowledge of the system is the programming language C#, the mistakes can be either syntax or logic.

2.2 WSM-Based Technique

The Weighted Sum Model (WSM) is a multi-criterion decision-making method in which there will be multiple alternatives and the best alternative has to be determined based on multiple criteria.

In general, assume that a particular MCDA problem has m alternatives and n decision criteria. Let us also presume that all of the criteria are benefit criteria, meaning that the if there are higher values, better results can be achieved. Assume that w_j represents the criterion C_j 's relative weight of significance, and that a_{ij} is the performance value of alternative A_i when compared to criterion C_j . Then, the total importance of alternative A_i (i.e., when all the criteria are considered simultaneously), denoted as $A_i^{\text{WSM-score}}$, is defined as follows:

$$A_i^{\text{WSM-score}} = \sum_{j=1}^n w_j a_{ij}, \text{ for } i = 1, 2, \dots, m. \quad (1)$$

For the maximization case, the best alternative is the one that yields the maximum total performance value.

The weights can be dynamic and, in our approach, they have been determined by the instructors. It needs to be noted that they can be altered in the tutoring of other domains.

3 Example of Operation

In this section, an example of operation is provided. We used data from 5 students and we show the process of creating the final leaderboard based on their students' characteristics and the weights given by the instructors. The data concern the instruction of the undergraduate course of the programming language C#. Table 1 consists of the characteristics of 5 students, including their grade, previous knowledge, interaction time and frequency of mistakes.

Table 1. Example of operation.

	Student	G	PKL	InT	FM	Performance	Ranking
		w ₁	w ₂	w ₃	w ₄		
Sample Data set & Deciding the maximum value for a beneficial attribute and minimum value for non-beneficial attribute	Student 1	8 (max)	5	4 (max)	1 (min)		
	Student 2	6	5	2	4		
	Student 3	7	5	2.5	2		
	Student 4	5.5	6	1.5	5		
	Student 5	6.5	7 (max)	3	3		
Normalization & the Weight-Normalized decision matrix	Student 1	1	0.7142	1	1		
	Student 2	0.75	0.7142	0.5	4		
	Student 3	0.875	0.7142	0.625	2		
	Student 4	0.6875	0.8571	0.375	5		
	Student 5	0.8125	1	0.75	3		
Multiplying each parameter with the respective weights	Student 1	0.35	0.1853	0.25	0.15		
	Student 2	0.2625	0.17855	0.125	0.6		
	Student 3	0.30625	0.17855	0.15625	0.3		
	Student 4	0.240625	0.214275	0.09375	0.75		
	Student 5	0.284375	0.25	0.1875	0.45		
Calculation of rank of students for the leaderboard	Student 1	0.35	0.1853	0.25	0.15	0.9353	5
	Student 2	0.2625	0.17855	0.125	0.6	1.16605	3
	Student 3	0.30625	0.17855	0.15625	0.3	0.94105	4
	Student 4	0.240625	0.214275	0.09375	0.75	1.29865	1
	Student 5	0.284375	0.25	0.1875	0.45	1.171875	2

The weights of each characteristic have been assigned by the instructors, based on their experience about the characteristics they consider important for student assessment. In this example of operation, the weights are as follows: $w_{\text{grade}} = 35\%$ (w_1), $w_{\text{previous_knowledge}} = 25\%$ (w_2), $w_{\text{interaction_time}} = 25\%$ (w_3) and $w_{\text{frequency_of_mistakes}} = 15\%$ (w_4).

In Table 2, the beneficial and non-beneficial attributes are shown. Beneficial attribute is one in which maximum values are desired. In our case, the grade, the previous knowledge level and the interaction time are beneficial attributes as instructors expect the students to have more of these attributes.

Non-beneficial attribute is one in which minimum values are desired. In our case, the frequency of mistakes is a non-beneficial attribute. Instructors expect the students to make less mistakes. In order to create the leadership by using Weighted Sum Method, we need to firstly normalize the values of Table 1. For beneficial attributes, there is $X = x/x_{\text{max}}$. For non-beneficial attributes, there is $X = x_{\text{min}}/x$.

Concluding, in the leaderboard the students will be appeared in the following ranking: Student 4, Student 5, Student 2, Student 3, Student 1.

4 Evaluation

In order to maintain the efficiency and usefulness of the software, evaluation is the key. In this case, the aim of the evaluation is to access the quality of the leaderboard by instructors as well as students. To achieve this, the t-test was used in order to measure the accuracy of the leaderboard in terms of the row of students. In our experiment, 20 university lecturers and 80 students participated. The university lecturers are in the field of computer science and specifically computer programming and software and the students are in the first year of their studies in a public university in the capital city of the country. Both of them used the learning technology system for the tutoring of the programming language C# during an academic semester and utilized the leaderboard. The instructors wanted to check the progress of students while the students wanted to check the ranking of the class.

For the experiment, we created two groups of instructors (Group 1 and Group 2) and two groups of students (Group A and Group B). Groups 1 and 2 included 10 instructors each while Groups A and B included 40 students each. Group 1 and Group A used our proposed approach holding the sophisticated leaderboard, while Group 2 and Group B used a leaderboard in which the ranking of students was solely based on their average grade.

At the end of the academic semester, both groups were asked to rate the leaderboard (ratings from lower 0 to higher 10). It needs to be noted that the main objective of the instructors when checking the leaderboard was to gain a better understanding on the progress of their students and see a more indicative and illustrative ranking of them. Also, the main objective of the students when checking the leaderboard was to see their progress in relation to the one of their classmates, so that they can advance their knowledge through competition and noble rivalry. For the experiment, the alpha value was 0.05 and we analyzed the p-values. Based on the results, for the null hypothesis: "There is no difference between the two groups of instructors as well as the two groups

of students” the t-Test rejects both hypotheses for the question “Rate the accuracy of the leaderboard”. That means that there is statistical significance and the proposed automated leaderboard is more qualitative than a simple leaderboard which is based on students’ average grade, according to the point of view of instructors and students.

Table 2. T-Test results.

t-Test: Two-Sample Assuming Equal Variances				
	Question to instructors		Question to students	
	Group 1	Group 2	Group A	Group B
Mean	9	7,2	8,175	7,35
Variance	0,444444	0,177778	0,814744	0,64359
Observations	10	10	40	40
Hypothesized Mean Difference	0		0	
df	15		77	
t Stat	7,216054		4,320714	
P(T <= t) one-tail	1,5E-06		2,3E-05	
t Critical one-tail	1,75305		1,664885	
P(T <= t) two-tail	2,99E-06		4,59E-05	
t Critical two-tail	2,13145		1,991254	

The above results were expected (Table 2). The presented approach, holding the WSM technique for creating a leaderboard in which a ranking based on multiple students’ characteristics is delivered, can have greater pedagogical potential since both instructors and students can gain a better understanding of the progress of the whole class. As such, instructors can have a proper idea of the ranking and progress of their class, while students can see their place in the class ranking and try to advance their knowledge. On the contrary, the leaderboard delivered to Group 2 and Group B lacked sophistication and therefore it was not so qualitative and adequate to present the “real” ranking of students.

5 Conclusions and Future Work

This paper presents a novel sophisticated leaderboard for the users (learners and instructors) of an intelligent tutoring system; the domain to be taught is the computer programming language C#. The leaderboards have significant pedagogical potential for learners to try to advance their knowledge in a competitive environment and for instructors who can gain a better understanding of their students and tailor their strategies to them. However, according to the related scientific literature, the leaderboards have been created based solely on one students’ characteristic, such as grade. In this paper, we introduce a

sophisticated leaderboard which is created using the WSM technique, considering multiple students' characteristics. As such, the leaderboard can better represent the cognitive skills of students. This is also attested by the evaluation results which show that our approach serves for a more accurate leaderboard.

Future steps include a more wide-range evaluation to further assess the accuracy of our approach. Furthermore, it is our future plans to explore the possible incorporation of other intelligent techniques for the development of a more robust leaderboard.






References

1. Troussas, C., Krouska, A., Sgouropoulou, C.: A novel teaching strategy through adaptive learning activities for computer programming. *IEEE Trans. Educ.* (2020). <https://doi.org/10.1109/TE.2020.3012744>
2. Garcia-Iruela, M., Hijón-Neira, R.: What perception do students have about the gamification elements? *IEEE Access* **8**, 134386–134392 (2020)
3. Romero-Rodríguez, L.M., Ramírez-Montoya, M.S., González, J.R.V.: Gamification in MOOCs: engagement application test in energy sustainability courses. *IEEE Access* **7**, 32093–32101 (2019)
4. Krouska, A., Troussas, C., Sgouropoulou, C.: A personalized brain-based quiz game for improving students' cognitive functions. In: Frasson, C., Bamidis, P., Vlamos, P. (eds.) *BFAL 2020*. LNCS (LNAI), vol. 12462, pp. 102–106. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60735-7_11
5. Tejedor-García, C., Escudero-Mancebo, D., Cardeñoso-Payo, V., González-Ferreras, C.: Using challenges to enhance a learning game for pronunciation training of English as a second language. *IEEE Access* **8**, 74250–74266 (2020)
6. Chernbumroong, S., Sureephong, P., Muangmoon, O.: The effect of leaderboard in different goal-setting levels. In: 2017 International Conference on Digital Arts, Media and Technology (ICDAMT), pp. 230–234. IEEE, Chiang Mai (2017)
7. de Pontes, R.G., Medeiros, K.H.M., Guerrero, D.D.S., de Figueiredo, J.C.A.: Analyzing the impact of leaderboards in introductory programming courses' short-length activities. In: *Frontiers in Education Conference (FIE)*, pp. 1–9. IEEE, San Jose, CA, USA (2018)
8. Landers, R.N.: Developing a theory of gamified learning: linking serious games and gamification of learning. *Simul. Gaming* **45**, 752–768 (2015)
9. Sisomboon, W., Phakdee, N., Denwattana, N.: Engaging and motivating developers by adopting scrum utilizing gamification. In: 4th International Conference on Information Technology (InCIT), pp. 223–227. IEEE, Bangkok, Thailand (2019)
10. Ferianda, M.R., Herdiani, A., Sardi, I.L.: Increasing students interaction in distance education using gamification. In: 6th International Conference on Information and Communication Technology (ICoICT), pp. 125–129. IEEE, Bandung (2018)
11. Flores, R., Elvira, G., Guevara, S., Brenda, N.: Work in progress engaging professional competencies through gamification. In: *Global Engineering Education Conference (EDUCON)*, pp. 1159–1163. IEEE, Porto, Portugal (2020)
12. Denden, M., Tlili, A., Essalmi, F., Jemni, M.: Students' learning performance in a gamified and self-determined learning environment. In: *International Multi-Conference on: Organization of Knowledge and Advanced Technologies (OCTA)*, pp. 1–5, IEEE, Tunis, Tunisia (2020)
13. Ortega-Alvarez, J.D., Sanchez, W., Magana, A.J.: Exploring undergraduate students' computational modeling abilities and conceptual understanding of electric circuits. *IEEE Trans. Educ.* **61**(3), 204–213 (2008)

14. Troussas, C., Krouska, A., Sgouropoulou, C.: Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education. *Comput. Educ.* 144, 103698 (2020)
15. Troussas, C., Krouska, A., Sgouropoulou, C., Voyiatzis, I.: Ensemble learning using fuzzy weights to improve learning style identification for adapted instructional routines. *Entropy* **22**(7), 735 (2020)



Learning and Gamification Dashboards: A Mixed-Method Study with Teachers

Kamilla Tenório¹(✉) , Bruno Lemos¹, Pedro Nascimento¹, Rodrigo Santos¹ ,
Alexandre Machado¹, Diego Dermeval² , Ranilson Paiva¹ ,
and Seiji Isotani³ 

¹ Computing Institute, Federal University of Alagoas, Maceió, AL 57072-900, Brazil
{kktas,b11,phbn,rss3,ajbm,ranilsonpaiva}@ic.ufal.br

² School of Medicine, Federal University of Alagoas, Maceió, AL 57072-900, Brazil
diego.matos@famed.ufal.br

³ Institute of Mathematics and Computational Sciences, University of São Paulo,
São Carlos, Brazil
sisotani@icmc.usp.br

Abstract. Previous studies have investigated the provision of students' relevant data, usually available in intuitive dashboards, to assist teachers in pedagogical decision-making in learning systems. Researchers are also interested in investigating how students' interaction data with gamification elements improve teachers' understanding of students' status and helps students with adequate pedagogical recommendations. However, there is a lack of understanding of how teachers perceive data visualizations provided through dashboards recently explored in gamified educational systems. In this paper, the authors investigate teachers' perceptions about three different dashboards with visualizations about 1) students' interaction with learning resources, 2) students' interaction with gamification elements, and 3) students' interaction with learning resources and gamification elements. As such, the researchers conducted a mixed-method study with 47 teachers to evaluate their perceived understanding, perceived usefulness, perceived behavioral change, and perceived decision-making support regarding the three dashboards. The results suggest teachers perceived that the dashboard with visualizations concerning students' interaction with learning resources and gamification elements provide better support to them in the decision-making process. Teachers also perceived the third dashboard has a more significant potential to impact behavioral changes than the other two dashboards.

Keywords: Learning analytics · Gamification analytics · Data visualization · Teachers · Pedagogical decision-making · Adaptive learning systems

1 Introduction

Innovations in technologies-enhanced learning context are revolutionizing education and, hence, transforming teachers' role, requiring them to be more tech-

nologically oriented [1,16]. Teachers' active participation in the technologies-enhanced learning context is essential to educational success [13,27]. However, the lack of teacher's educational technologies' involvement is one of the most significant barriers to access to digital learning in education [13]. Even artificial intelligence-enhanced educational systems that offer content in a personalized way (e.g., intelligent tutoring systems)[10], potentially substituting teachers' primary role of instruction, face barriers as the low rate of adoption and use due to the lack of support for teachers[15,21].

Researchers and practitioners are increasingly concerned in supporting teachers integrating educational technologies into their pedagogy to obtain academic success, putting teachers at the frontline of education instead of replacing them in the intelligent tutoring systems context [5,7]. Previous research suggests that teachers should participate in all intelligent tutoring systems life-cycle stages [6]. For instance, in the pre-instruction, researchers are enabling teachers to design educational technologies, delivering a personalized environment to their students [4]. During instruction, students' interaction data may help the teacher's decision-making process [2,18,22], allowing them to visualize data related to students' performance and progress over time [14]. Therefore, when students' understanding is not progressing as expected, teachers intervene through pedagogical actions [14]. In the post-instruction phase, teachers could re-design curricula, add new learning resources, change existing features in the educational system, etc. [4].

In a novel perspective, previous studies have investigated how learning and gamification analytics could help teachers during the instruction phase in gamified adaptive educational environments [24–26]. These studies explored how the provision of students' interaction with learning resources and gamification elements data may increase teachers' awareness and how this gamification analytics approach could impact students' outcomes. According to these previous studies' empirical results, gamification analytics may help avoid unexpected effects regarding students' engagement, learning, and motivation in the gamified environments [24–26]. However, there is a lack of a more profound understanding of how teachers perceive different dashboards, including learning and gamification dashboards. This type of research could be relevant because it may help design more straightforward approaches to support teachers' decision-making, considering both students' learning and gamification data in the context of gamified adaptive educational systems [19,25].

Therefore, this paper extends these studies to evaluate teachers' perceptions concerning different types of dashboards. This research was conducted with 47 teachers with different backgrounds, in which they evaluated three types of dashboards with students' information retrieved from a gamified adaptive learning system. The first dashboard provides only information regarding students' interaction with learning resources. The second dashboard provides only information regarding students' interaction with gamification elements. The third one includes information regarding both interactions. We aim to investigate how

teachers perceive these dashboards regarding understanding level, usefulness, behavioral changes, and perception about decision-making support.

The remainder of this paper is structured as follows. Section 2 discusses the related works. The authors explain the experiment conduction in Sect. 3. Section 4 describes the quantitative and qualitative results obtained after the experiment's conduction with teachers. Finally, in Sect. 5, the authors discuss this research's main results and depict this work's concluding remarks.

2 Related Works

Learning analytics, defined by the 1st International Conference on Learning Analytics and Knowledge, is “the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs” [11]. Learning analytics is a research field that emerged based on previous related areas such as academic analytics and educational data mining and recently received growing attention from educational researchers and practitioners [9]. Previous studies have investigated processing, analyzing, and displaying students' learning data in a meaningful way to assist teachers in their decision-making process in gamified education systems based on learning analytics research [7, 12, 18]. Recent studies have pointed out the effectiveness of the provision of students' learning data through learning analytics to aid teachers in their decision-making process in gamified learning systems [12, 18].

More recent gamification studies in the education research field have been exploring the use of students' data interaction with gamification elements to increase teachers' awareness about students' status, based on research in gamification analytics [17, 24–26, 28]. Gamification analytics, defined by Heilbrunn, Herzig, and Schill, is the “data-driven processes of monitoring and adapting gamification designs” [8]. In the educational context, the provision of students' data interaction with gamification elements has been pointed out as a promising approach to support teachers' decision-making process during learning systems [17, 24, 26, 28]. However, to the best of our knowledge, there are no studies in the literature investigating the difference between the teachers' perception regarding visualizations that present students' interaction data with learning resources and visualizations that offer students' interaction data with gamification elements.

3 Experiment

This section presents the materials and methods used in our study, highlighting our hypotheses, instrumentation, research questions, and the independent and dependent variables. Moreover, we also describe the data analysis method we are using in this research.

3.1 Materials and Methods

As previously explained, in this work, we evaluate how teachers perceive different types of dashboards in gamified learning systems. The detailed visualizations available in each dashboard are described below and shown in Fig. 1.

Dashboard 1 - Students' Interaction with Learning Resources: The following visualizations are provided for teachers in dashboard 1, as seen in Fig. 1: the number of students registered for the course; the expected period for mastering a specific topic in the course; the total number of students who have mastered a topic; class' progress over time concerning the interaction with learning resources; the number and names of students who have mastered the topic or not; the number and names of students who interacted (with or without success) or did not interact with each learning resource of a topic.

Dashboard 2 - Students' Interaction with Gamification Elements: The following visualizations are provided for teachers in dashboard 2, as seen in Fig. 1: the number of students registered for the course; the average of the students' gamified points in the system; the total number of students who have mastered a topic; class' progress over time concerning the interaction with learning resources and markings in the charts to show when the teacher created missions; the number and names of students who are at each gamification level; the number and names of students who reached, did not reach or did not attempt to perform each mission created by the teacher during the learning process.

Dashboard 3 - Students' Interaction with Learning Resources and Gamification Elements: The following visualizations are provided for teachers in dashboard 3, as seen in Fig. 1: the number of students registered in the course; the expected period for mastering a topic of the course; the total number of students who have mastered a specific topic; class' progress over time concerning the interaction with learning resources and markings in the charts to show when the teacher created missions; the number and names of students who have mastered the topic or not; the number and names of students who interacted (with or without success), or did not interact with the learning resources; the number and names of students who are at each gamification level; the number and names of students who reached, did not reach or did not attempt to accomplish each mission created by the teacher during the learning process. This dashboard unites the visualizations present in the previous dashboards.

The researchers conducted a mixed-method study (quantitative and qualitative research) with an online instrument's support. The authors developed the online tool and interactive dashboards using HTML/CSS, React JS, Highcharts JS, and Node JS technologies. The dashboards were connected with a gamified adaptive learning system (game elements implemented: points, levels, and missions), so the data showed in these dashboards were retrieved from a real environment. Teachers, recruited by e-mail, read the informed consent form, answered the demographic questionnaire, and watched a video tutorial to understand this research. After that, teachers visualized and interacted with the three dashboards in random order – the system randomly presented the dashboard



Fig. 1. Overview of the dashboards evaluated: (1) the dashboard that presents data related to students' interaction with learning resources; (2) the dashboard that shows data related to students' interaction with gamification elements; (3) The dashboard that shows information about both interactions.

sequence to teachers to avoid maturation bias. After viewing each dashboard, teachers answered a questionnaire with nine multiple-choice questions (using a 7-point Likert scale) regarding understanding, perceived usefulness, and behavioral change factors and optionally answered two open-ended questions regarding positive and negative points of the dashboards. This study's design considers understanding, perceived usefulness, and behavioral changes factors to evaluate the dashboards based on the instrument validated by Park and Jo's work [20] to measure dashboard success. Finally, at the end of the survey, the researchers asked teachers to inform which of the evaluated dashboards would best help them in the decision-making process during the learning process in gamified learning systems. Therefore, this paper investigates the following hypotheses:

H1: Teachers' understanding level concerning the three dashboards is identical.
H2: Teachers' perception of usefulness concerning the three dashboards is identical.

H3: Teachers' perceived behavioral changes concerning the three dashboards are identical.

H4: Teachers' perception of the three dashboards' support to help them in the decision-making process in gamified learning systems is identical.

3.2 Data Analysis Procedure

The authors conducted a quantitative analysis to test the four hypotheses previously defined. This research verifies the first, second, and third hypotheses by applying the non-parametric Friedman's ANOVA test to evaluate if the teachers' understanding, perceived usefulness, and behavioral changes concerning the three dashboards are equal. For the fourth hypothesis, the authors conduct a non-parametric Pearson's chi-square goodness-of-fit test to determine if there is a statistically significant difference between the expected frequencies and the observed frequencies in teachers' responses. To perform the analysis, we considered the numbers of teachers that mutually pointed out a dashboard as the most helpful in the decision-making process.

The authors also conducted a qualitative analysis to evaluate teachers' positive and negative opinions about each dashboard. We adopted the open coding scheme [3] – an analytic process through which concepts are identified in data. After that, we grouped teachers' answers into categories to better understand their positive and negative perceptions and opinions regarding each of the three dashboards assessed in the experiment.

4 Results

This section presents this research's experiment results, depicting the sample characteristics, showing each hypothesis's statistical results, and the open questions' results.

4.1 Sample Characteristics

Thirty-three out of 47 teachers are male, and 14 are female. Most of the teachers are from 26 to 50 years old, followed by 19 teachers from 41 to 65 years old, and one teacher whose age is between 18 to 25 years old. Most teachers (25) declared they have a medium technical level, followed by 21 teachers who reported a high technical level and only one teacher reported a low technological level. Moreover, most teachers (38) teach at the secondary level, 22 teach at the post-secondary, seven at the post-baccalaureate, and four at the elementary. Eighteen teachers teach at more than one educational level.

4.2 Hypothesis Tests

Teachers' Understanding Level - H1 assumed that teachers' understanding level concerning the visualizations of the three dashboards evaluated is identical. Therefore, to test this hypothesis, we evaluated firstly if the current data met the normality assumption for the analysis of variance (ANOVA). A Shapiro-Wilk test of normality distribution was performed to examine the distribution, and the test results indicate that the data are not from a normal population (dashboard 1: $W = 0.762$, $p < 0.001$; dashboard 2: $W = 0.728$, $p < 0.001$; dashboard 3: $W = 0.749$, $p < 0.001$). Therefore, we compute the non-parametric Friedman's ANOVA test. Results indicate that there is not a statistically significant difference in teachers' understanding level concerning the visualizations provided by the three dashboards evaluated (dashboard 1: $M = 5.73$, $SD = 1.18$; dashboard 2: $M = 5.68$, $SD = 1.14$; dashboard 3: $M = 5.71$, $SD = 0.97$) $\chi^2(2) = 0.061$, $p > 0.05$, confirming the null hypothesis.

Teachers' Perceived Usefulness - H2 assumed that teachers' perceived usefulness after interacting with each of the three dashboards evaluated is identical. We also run a Shapiro-Wilk test to examine the distribution of the data concerning the factor evaluated. Results indicate that the data regarding teachers' perceived usefulness of the three dashboards are not from a normal distribution (dashboard 1: $W = 0.903$, $p < 0.001$; dashboard 2: $W = 0.875$, $p < 0.001$; dashboard 3: $W = 0.866$, $p < 0.001$). We perform a non-parametric Friedman's ANOVA test to evaluate if the data concerning teachers' perceived usefulness about the three dashboards have identical means. The test's outcome indicates that there is not a statistically significant difference in teachers' perceived usefulness after interacting with each of the three dashboards evaluated (dashboard 1: $M = 5.38$, $SD = 1.15$; dashboard 2: $M = 5.37$, $SD = 1.11$; dashboard 3: $M = 5.55$, $SD = 1.03$) $\chi^2(2) = 4.353$, $p > 0.05$, confirming the null hypothesis.

Teachers' Perceived Behavioral Changes - H3 assumed that the teachers' perceived behavioral changes after the interaction with each of the three dashboards are identical. A Shapiro-Wilk test was also performed to evaluate the distribution of the data, and the test results indicate that the data are not from a normal population (dashboard 1: $W = 0.787$, $p < 0.001$; dashboard

2: $W = 0.761$, $p < 0.001$; dashboard 3: $W = 0.754$, $p < 0.001$). As such, we perform the non-parametric Friedman's ANOVA test. Results show that there is a statistically significant difference in teachers' perceived behavioral changes after interacting with each of the three dashboards evaluated (dashboard 1: $M = 5.58$, $SD = 1.29$; dashboard 2: $M = 5.41$, $SD = 1.26$; dashboard 3: $M = 5.70$, $SD = 1.14$) $\chi^2(2) = 6.523$, $p = 0.038$, rejecting the null hypothesis. Dashboard 3 (provision of data related to students' interaction with learning resources and gamification elements) received a better evaluation from teachers, showing that this dashboard can have a greater potential to affect teachers' behavioral changes than the other two dashboards.

Teachers' Decision Making Process - H4 assumed that teachers' perception regarding the three dashboards' support to help them in the decision-making process in gamified learning systems is identical. To test H4, we performed a Pearson's chi-square goodness-of-fit test [23]. The results of chi-square tests performed on the frequencies of observed teachers' responses for each dashboard indicate that there are significant differences in the teachers' preference for one of the three dashboards for the decision-making process ($\chi^2 = 6.936$; $p = 0.031$), rejecting the null hypothesis. Therefore, we can perceive that there is a teachers' preference for dashboard 3 (dashboard 1: 13 teachers; dashboard 2: 10 teachers; dashboard 3: 24 teachers), which might indicate that a complete dashboard that provides information concerning students' interaction with learning resources and gamification elements better supports the teachers' decision-making process during the teaching-learning process in gamified learning systems, according to teachers' perception.

4.3 Open-Ended Responses Analysis

For each dashboard, teachers answered two optional open-ended questions concerning positive and negative points. Regarding dashboard 1, 18 out of 47 participant teachers (38,29%) pointed out the positive issues classified into three major categories: visual attraction, usefulness, and usability. The following are some of the answers.

Visual Attraction: 1. "This dashboard was simple and intuitive."

Usefulness: 1. "The dashboard gathers essential information that facilitates the teacher's understanding of the class's individual and collective advances." **2.** "Synthetic dashboard with essential information for decision-making."

Usability: 1. "Information about the students' progress was evident."

Regarding dashboard 2, 16 out of 47 participant teachers (34,04%) pointed out the positive points. They were classified into two major categories: usefulness and usability. The following are some of the answers.

Usefulness: 1. "The presented data collaborate for an assessment of the class in an interactive way that allows viewing important points, such as the individual assessment of the student about the rest of the class, which provides a clear way for the possibility of recovering individual learning so that everyone can reach the goal at the end of the period."

Usability: 1. “Quickly map the progress of the class’ learning.”

Concerning dashboard 3, 17 out of 47 participant teachers (36,17%) pointed out the positive points. They were classified into three major categories: visual attraction, usefulness, and usability. The following are some of the answers.

Visual Attraction: 1. “Self-explanatory images, especially the line charts.”

Usefulness: 1. “The tool has enormous potential, easy visualization of data and information. There are many insights for different applications, including decision-making in teaching strategies and learning measurement.”

Usability: 1. “The type of language used is very positive considering that it uses tools related to games.”

Concerning the negative points of dashboard 1, thirteen out of forty-seven participant teachers (27,65%) pointed out negative points. Teachers’ negative opinions about dashboard one were classified into three major categories: visual attraction, usefulness, and usability.

Visual Attraction: 1. “It could have a greater variety of colors and a more pleasant design.”

Usefulness: 1. “Despite little information on this dashboard compared to the others, I missed the possibility of seeing the student individually. It could have better analyzes with the possibility to investigate case by case, especially in the points that were not reached.”

Usability: 1. “Absence of description of the resources used by students. Not all the resources indicated may be known to teachers.”

Concerning dashboard 2, 14 out of 47 participant teachers (29,78%) pointed out negative points. They were classified into two major categories: visual attraction and usability, as follows.

Visual attraction: 1. “I see many types of data presented on just one screen... this information could be categorized and summarized, so it does not become a tiring visualization.”

Usability: 1. “Very technical. I believe that most teachers, who work in multiple classes, would not have time to analyze the data to improve their teaching practice.”

Regarding dashboard 3, 14 out of 47 participant teachers (29,78%) pointed out negative points. They were classified into two major categories: visual attraction, usability, as presented below.

Visual attraction: 1. “It would be interesting to show statistically where students miss or fail the most.” **2.** “A lot of information in few charts.”

Usability: 1. “I thought the informative charts with little usability. They are very technical, and for those who are not in the technology area, it would be confused.”

5 Discussion and Conclusion

This work investigated the perception of 47 teachers regarding three different dashboards: the first showing students’ interaction with learning resources, the second showing students’ interaction with only gamification elements, and the

third one presenting students' interaction with gamification elements and learning resources. This research brings a novel contribution. It investigates how teachers perceive the benefits of relying on students' learning and gamification data through dashboards in gamified adaptive learning systems.

Overall, the understanding, perceived usefulness, and behavioral change factors mean for all three dashboards were high, indicating that the three dashboards caused a satisfactory effect on teachers' understanding level, perceived usefulness, and perceived behavioral changes. The results also suggest that teachers evaluated dashboard 3 (students' interaction with learning resources and gamification elements) as more helpful to them in the decision-making process. Dashboard 3 also received a better evaluation from teachers concerning the behavior changes factor. Moreover, although the results may indicate that dashboard 3 has a similar effect on teachers' perceived usefulness, compared to dashboards 1 and 2, dashboard 3 received a slightly better evaluation from teachers concerning perceived usefulness. These results might suggest that this dashboard can have a more significant potential to affect teachers' perceived usefulness and perceived behavioral changes positively.

Based on the qualitative results, the teachers' perceptions and opinions about the three dashboards were similar. They were more focused on dashboards' visual attraction, usability, and usefulness. Teachers highlighted positive points regarding the visual attraction of dashboards 1 and 3, emphasizing that they present simple, intuitive (for dashboard 1), and self-explanatory (for dashboard 3) visualizations. Teachers pointed out that all three dashboards can assist them in the decision-making process and teaching planning. Teachers also pointed out the three dashboards' usability, reporting that the dashboards effectively fulfill their role, informing teachers about the students' status.

Regarding negative opinions about visual attraction, teachers highlighted the lack of a greater variety of colors, and a more pleasant design (dashboard 1), a provision of a lot of information on a single screen (dashboard 2), and a lack of slightest visual care (dashboard 3). Dashboard 1 received negative opinions regarding its usefulness, where teachers highlighted the lack of possibility of seeing the student individually to help in the teachers' decision-making process. Finally, the three dashboards received negative opinions from teachers regarding usability. The teachers highlighted the absence of a description of the available learning resources (dashboard 1). They pointed out that very technical visualizations can confuse teachers who do not have computational technical skills.

In future work, the authors aim to target the usability issues mentioned by teachers. We aim to evaluate dashboard 3 in real settings, integrated into a gamified adaptive educational system, to effectively evaluate how it helps teachers make decisions, including analyzing the effect of teachers' decisions on the students' learning, motivation, and engagement.

References

1. Amin, N.A.: Redefining the role of teachers in the digital era. *Int. J. Indian Psychol.* **3**(3) (2016), <http://www.doi.org/10.25215/0303.101>

2. Araújo Paiva, R.O., Bittencourt, I.I., Da Silva, A.P., Isotani, S., Jaques, P.: Improving pedagogical recommendations by classifying students according to their interactional behavior in a gamified learning environment. In: Proceedings of the ACM Symposium on Applied Computing, vol. 13–17-April-2015, pp. 233–238. Association for Computing Machinery, New York (2015). <http://www.doi.org/10.1145/2695664.2695874>
3. Corbin, J.M., Strauss, A.L.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications, Los Angeles (2008)
4. Dermeval, D., Bittencourt, I.I.: Co-designing gamified intelligent tutoring systems with teachers. *Braz. J. Comput. Educ. Revista Brasileira de Informática na Educação-RBIE* **28**, 73–91 (2020). <http://www.doi.org/10.5753/RBIE.2020.28.0.73>
5. Dermeval, D., et al.: An ontology-driven software product line architecture for developing gamified intelligent tutoring systems. *Int. J. Knowl. Learn.* **12**(1), 27–48 (2017). <https://doi.org/10.1504/IJKL.2017.10009129>
6. Dermeval, D., Paiva, R., Bittencourt, I.I., Vassileva, J., Borges, D.: Authoring tools for designing intelligent tutoring systems: a systematic review of the literature. *Int. J. Artif. Intell. Educ.* **28**(3), 336–384 (2017). <https://doi.org/10.1007/s40593-017-0157-9>
7. González González, C., Toledo, P., Muñoz, V.: Enhancing the engagement of intelligent tutorial systems through personalization of gamification. *Int. J. Eng. Educ.* **32**(1), 532–541 (2016)
8. Heilbrunn, B., Herzig, P., Schill, A.: Gamification analytics—methods and tools for monitoring and adapting gamification designs. In: Stieglitz, S., Lattemann, C., Robra-Bissantz, S., Zarnekow, R., Brockmann, T. (eds.) *Gamification*. PI, pp. 31–47. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-45557-0_3
9. Hui, Y.K., Kwok, L.F.: A review on learning analytics. *Int. J. Innov. Learn.* **25**(2), 197–222 (2019). <https://doi.org/10.1504/IJIL.2019.097673>
10. Izumi, L., Fathers, F., Clemens, J.: *Technology and Education: A primer*. Fraser Institute, Canada (2013)
11. Lee, L.-K., Cheung, S.K.S., Kwok, L.-F.: Learning analytics: current trends and innovative practices. *J. Comput. Educ.* **7**(1), 1–6 (2020). <https://doi.org/10.1007/s40692-020-00155-8>
12. Llorens-Largo, F., et al.: Chapter 12 - LudifyME: An Adaptive Learning Model Based on Gamification. In: Caballé, S., Clarisó, R. (eds.) *Formative Assessment, Learning Data Analytics and Gamification*, pp. 245–269. Academic Press, Boston (2016). <https://doi.org/10.1016/B978-0-12-803637-2.00012-9>
13. Macleod, H., Sinclair, C.: Digital Learning and the Changing Role of the Teacher. In: Peters, M.A. (eds.) *Encyclopedia of Educational Philosophy and Theory*, pp. 566–571. Springer, Singapore (2017). https://doi.org/10.1007/978-981-287-588-4_126
14. Molenaar, I., Knoop-van Campen, C.: How Teachers Make Dashboard Information Actionable. *IEEE Trans. Learn. Technol.* **1** (2018). <https://doi.org/10.1109/TLT.2018.2851585>
15. Nye, B.D.: Barriers to ITS adoption: a systematic mapping study. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014*. LNCS, vol. 8474, pp. 583–590. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07221-0_74
16. Jethro, O.O., Grace, A.M., Thomas, A.K.: E-Learning and its effects on teaching and learning in a global age. *Int. J. Acad. Res. Bus. Soc. Sci.* **2**(1), 244–252 (2012)

17. Paiva, R., Bittencourt, I.I.: The Authoring of Pedagogical Decisions Informed by Data, on the Perspective of a MOOC. In: Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (CBIE 2017), vol. 1, p. 15. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC) (2017). <https://doi.org/10.5753/cbie.wcbie.2017.15>
18. Paiva, R., Bittencourt, I.I., Tenrio, T., Jaques, P., Isotani, S.: What Do Students Do On-line? Modeling Students' Interactions to Improve Their Learning Experience. *Comput. Hum. Behav.* **64**(C), 769–781 (2016). <https://doi.org/10.1016/j.chb.2016.07.048>
19. Paiva, R., de Holanda, J.F.S., Peixoto, M.D., Vieira, J.P.: Augmenting teachers with data science powers: Joining human and artificial intelligence to assist students. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), vol. 30, p. 1721 (2019). <https://doi.org/10.5753/cbie.sbie.2019.1721>
20. Park, Y., Jo, I.-H.: Factors that affect the success of learning analytics dashboards. *Educ. Technol. Res. Dev.* , 1–25 (2019). <https://doi.org/10.1007/s11423-019-09693-0>
21. Pinkwart, N.: Another 25 years of AIED? challenges and opportunities for intelligent educational technologies of the future. *Int. J. Artif. Intell. Educ.* **26**(2), 771–783 (2016). <https://doi.org/10.1007/s40593-016-0099-7>
22. Prenger, R., Schildkamp, K.: Data-based decision making for teacher and student learning: a psychological perspective on the role of the teacher. *Educ. Psychol.* **38**(6), 734–752 (2018). <https://doi.org/10.1080/01443410.2018.1426834>
23. Sharpe, D.: Chi-square test is statistically significant: now what? *Pract. Assess. Res. Eval.* **20**(1) (2015). <https://doi.org/10.7275/tbfa-x148>
24. Tenório, K., et al.: Helping teachers assist their students in gamified adaptive educational systems: towards a gamification analytics tool. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) AIED 2020. LNCS (LNAI), vol. 12164, pp. 312–317. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52240-7_57
25. Tenório, K., Dermeval, D., Monteiro, M., Peixoto, A., Pedro, A.: Raising teachers empowerment in gamification design of adaptive learning systems: a qualitative research. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) AIED 2020. LNCS (LNAI), vol. 12163, pp. 524–536. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52237-7_42
26. Tenório, K., et al.: An Evaluation of the GamAnalytics Tool: Is the Gamification Analytics Model Ready for Teachers? In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação), pp. 562–571. Sociedade Brasileira de Computação (2020). <https://doi.org/10.5753/cbie.sbie.2020.562>
27. UNICEF: Raising Learning Outcomes: the opportunities and challenges of ICT for learning. Technical Report (2018). <https://www.unicef.org/esa/media/2636/file/UNICEF-AKF-IU-2018-ICT-Education-WCAR-ESAR.pdf>
28. Zarc, N., Gottschlich, M., Roepke, R., Schroeder, U.: Supporting gamification with an interactive gamification analytics tool (IGAT). In: Alario-Hoyos, C., Rodríguez-Triana, M.J., Scheffel, M., Arnedillo-Sánchez, I., Dennerlein, S.M. (eds.) EC-TEL 2020. LNCS, vol. 12315, pp. 461–466. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57717-9_45



Encouraging Teacher-Sourcing of Social Recommendations Through Participatory Gamification Design

Elad Jacobson¹(✉), Armando Toda², Alexandra I. Cristea²,
and Giora Alexandron¹(✉)

¹ Weizmann Institute of Science, 234 Herzl Street, Rehovot, Israel
{elad.jacobson,giora.alexandron}@weizmann.ac.il

² Durham University, Stockton Road, Durham, UK
armando.toda@usp.br, alexandra.i.cristea@durham.ac.uk

Abstract. Teachers and learners who search for learning materials in open educational resources (OER) repositories greatly benefit from feedback and reviews left by peers who have activated these resources in their class. Such feedback can also fuel social-based ranking algorithms and recommendation systems. However, while educational users appreciate the recommendations made by other teachers, they are not highly motivated to provide such feedback by themselves. This situation is common in many consumer applications that rely on users' opinions for personalisation. A possible solution that was successfully applied in several other domains to incentivise active participation is gamification. This paper describes for the first time the application of a comprehensive cutting-edge gamification taxonomy, in a user-centred participatory-design process of an OER system for Physics, PeTeL, used throughout Israel. Physics teachers were first involved in designing gamification features based on their preferences, helping shape the gamification mechanisms likely to enhance their motivation to provide reviews. The results informed directly the implementation of two gamification elements that were implemented in the learning environment, with a second experiment evaluating their actual effect on teachers' behaviour. After a long-term, real-life pilot of two months, teachers' response rate was measured and compared to the prior state. The results showed a statistically significant effect, with a 4X increase in the total amount of recommendations per month, even when taking into account the 'Covid-pandemic effect'.

Keywords: Gamification · Blended learning · Recommendation · Crowd-sourcing

This research is supported by a Making Connections Grant funded by Weizmann UK. The work of GA is also supported by the Willner Family Leadership Institute for the Weizmann Institute of Science.

© Springer Nature Switzerland AG 2021

A. I. Cristea and C. Troussas (Eds.): ITS 2021, LNCS 12677, pp. 418–429, 2021.

https://doi.org/10.1007/978-3-030-80421-3_46

1 Introduction

Personalised learning environments rely on repositories of digital learning materials, and on meta-data that provide semantic information about the digital content [10]. The semantic information is fundamental to the ability of AI agents to make ‘intelligent’ decisions, such as recommending content to learners, assisting teachers in search & discovery of learning resources, and for re-using materials between contexts [2–5, 15]. Recommendations about the learning resources is an important component of the semantic information, since teachers searching for learning materials in blended learning environments value the feedback and review of peers who have previously used these resources [7]. However, a major challenge in mining recommendations from teachers is their low motivation to contribute the time and effort needed to produce such feedback [12].

One possible solution to this challenge is the use of *Gamification*: a term describing the use of game elements (such as points, prizes, progression through levels, time pressure, competition, cognitive challenges, and more) to improve user experience and user engagement in non-game services and applications [9]. The underlying idea of gamification is that by making a task entertaining, it is possible to engage humans to do tasks that do not provide any other tangible reward [8, 16]. Gamification is being used in various domains and types of systems, including social networks, e-commerce, search engines, healthcare systems, and more [6, 8, 13, 19].

One of the most prominent fields in which gamification is used, is that of educational technology [8]. Attempts at applying gamification elements and methods in educational contexts have shown promising results [1, 14, 17, 18]. However, to the best of our knowledge, the potential of gamification to incentivise teachers in teacher-sourcing tasks was not evaluated before.

In this paper, we report on the results of a pilot research aimed at studying the impact of gamification on teachers’ motivation to contribute feedback on the resources that they have used (typically for in-class activities or as homework), and fuel a social-based recommendation system within an OER repository in Physics. Specifically, we seek to answer the following research questions (RQs):

- **RQ1:** What gamification mechanisms do teachers believe will encourage them to provide feedback on the learning resources that they have used?
- **RQ2:** Does implementing these elements actually enhance teachers’ willingness to provide feedback?

This paper makes the following contributions:

1. It is the first real-life design and implementation of a cutting-edge Gamification Taxonomy [21].
2. It presents, for the first time, a participatory design approach for introducing Gamification into a large OER system, used throughout a whole country.
3. The paper provides results on the implementation of the Gamification elements via a long-term pilot study within a real-life OER system for teachers.

4. Results show statistically significant increase in feedback from the teachers, to an unprecedented 4X increase, even when taking into account the ‘Covid-pandemic effect’.

2 The Learning Environment – PeTeL

PeTeL (Personalised Teaching and Learning) is a shared repository of open educational resources (OER), and a Learning Management System (LMS) that also includes social network features and learning analytics tools. It is developed at the Department of Science Teaching at the Weizmann Institute of Science, with the goal of assisting STEM teachers in providing personalised instruction in blended-learning environments.

PeTeL is divided into separate modules for each subject matter: Biology, Chemistry and Physics. It is implemented on top of a Moodle LMS. To assist teachers in searching and discovering learning materials that best suit their students’ needs, PeTeL provides common search filters such as subject matter, level of difficulty, duration, technical requirements (e.g. projector or mobile devices), nature of the activity (e.g. diagnostic questionnaire, interactive task, home assignment, etc.), and in addition, social-based search and discovery features. For example, teachers can follow other teachers within a social network-style collaborative environment (referred to as the ‘peer network’), receive recommendations from them, copy their teaching sequences, and more. Teachers can also search and rank materials based on reviews provided by their peers.

After using an activity in their class, the teachers are presented with a ‘pop-up’ window, requesting them to provide feedback concerning the resource they used. The teachers can either fill the pop-up survey, postpone filling out the form to a later date, or cancel it. This feedback mechanism was initially activated in PeTeL during the 2019–2020 school year. However, teachers’ cooperation was relatively low, and their response rate to the feedback requests during this first year was below 3%. Since the reviews were identified by the teachers as very influential on their decision on which activities to use, and also provide the basis for an automatic ranking algorithm that is currently under design, we marked the issue of increasing the response rate as a major challenge that should be addressed, and decided to examine gamification as a conceptual framework for addressing this challenge.

3 Gamification Taxonomy

Concerning the gamification elements, our conceptual framework relied on the new, cutting-edge Taxonomy of Gamification Elements for Educational Environments (TGEEE) [20,21]. The taxonomy was built based on large-scale data collection on gamification preferences of educational users, and proposes 21 gamification elements suited for educational contexts. These elements are grouped into five major dimensions: Performance, Social, Ecological, Personal, and Fictional.

The Performance dimension includes elements that are related to the environment’s response to student interactions, such as badges and points. The Social dimension refers to elements that deal with interactions between the students in the environment, e.g. cooperation and competition. The Personal dimension is related to the learner using the environment, usually related to meaning and purpose, for example, by setting objectives. The Ecological dimension refers to properties/characteristics provided by the environment, such as economy and chance. Finally, the Fictional dimension deals with the context of the environment, affecting both users (Narrative) and the environment (Storytelling). A graphical representation of the elements, and their grouping into dimensions, is depicted in Fig. 1.

It is important to state that, according to the authors, an environment does not necessarily need to contain all the elements from all dimensions. The selection of elements should be aligned with the objectives of the environment and the users who will interact with it [22,23]. This justified our first experiment, the participatory design with teachers, described next.

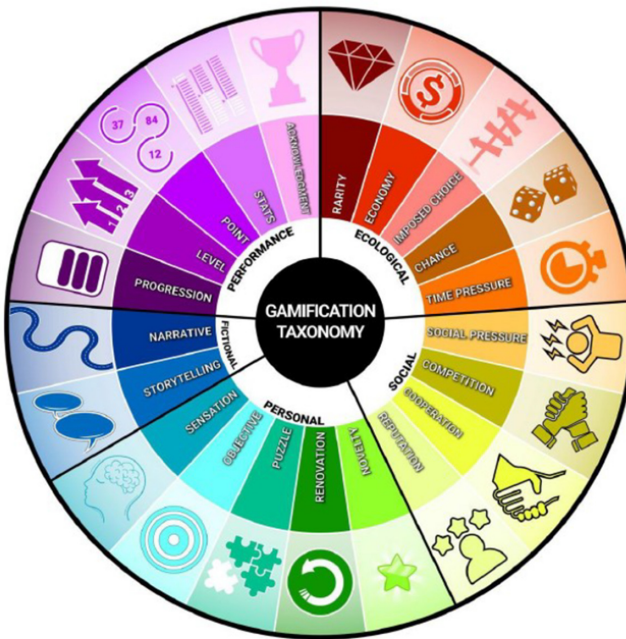


Fig. 1. The TGEEE Gamification Taxonomy from [21]

4 First Experiment: Teacher Preferences

This experiment was designed to answer the first research question: What gamification mechanisms do teachers believe will encourage them to provide feedback on the learning resources that they have used?

4.1 Procedure and Materials

The first experiment, a participatory design experiment, was conducted with seventeen Physics teachers, ten women and seven men, who participated in a one-day yearly training conference on PeTeL during July 2020. This was part of a session on the search and discovery mechanisms that PeTeL offers. A previous iteration of this event, allowing for interviews with teachers, marked the social recommendation as especially valued by teachers. We discussed the low response rate on the pop-up surveys, the potential use of gamification as means to increase it, and presented the taxonomy to the teachers.

Then, the teachers were presented with five mock-ups of different gamification elements, each implementing a certain dimension of the Taxonomy (see below), and were requested to rate how much they believed that the concept underlying this element (e.g., social reward) could enhance teachers' motivation to provide feedback (on a 1–5 Likert scale). In addition to the Likert questionnaire, the teachers were requested to expand their answers as much as they wished, via open-ended questions. Then, a group discussion was held. We note that the mock-ups were visually integrated into the front-end of PeTeL, to provide an authentic user experience.

4.2 The Five Elements Presented to Teachers

Badges: the first element was giving teachers virtual badges (gold, silver or bronze) according to the amount of reviews they gave. We based this element on two different concepts from the taxonomy: first, the “acknowledgement” concept from the “performance” dimension in the taxonomy, which refers to elements in the environment that praise the user's actions. The second was the “reputation” concept from the “social” dimension in the taxonomy, meaning that teachers may value the possibility of being recognised by their peers as contributing members to the entire teacher community.

Leader-Board: the second element was a leader board, presenting the number of points each teacher accumulated by filling in reviews. This element was also based on two different concepts from the taxonomy: the first was the “points” concept taken from the “performance” dimension in the taxonomy, meaning that the notion of receiving credit for their performance could raise teachers' motivation. The second was the “competition” concept from the “social” dimension in the taxonomy, indicating that the presentation of a teacher's ranking in comparison to other teachers can encourage them to participate.

Progress-Bar: the third element was a progress bar, showing the accumulation of required feedbacks on each learning resource. This element was based on the three following concepts: “cooperation” taken from the “social” dimension in the taxonomy, the “progression” concept taken from the “performance” dimension in the taxonomy, and the “objectives” concept taken from the “personal” dimension. The “cooperation” element builds upon the notion that the teachers’ feeling that they are working together towards a common goal, could motivate them. The “progression” concept claims that allowing teachers to view their progression within the environment will foster their willingness to contribute information. Finally, the “objectives” concept states that giving teachers a clear goal will raise their motivation.

Virtual Applause: the fourth element was virtual applause, meaning that each time teachers filled out a feedback form, the learning environment would present them with an animation of fireworks, confetti, and the sound of an audience applauding. This element is based on the “sensation” concept taken from the “personal” dimension in the taxonomy. This means that using the teachers’ senses in the manner of visual or audio stimulation, can affect their motivation.

PeTeL Dollars: the fifth element was PeTeL Dollars, meaning that the teacher would receive virtual currency for giving feedbacks. At the end of the school year, if the teacher has reached a certain amount of virtual dollars, he/she can replace them for a real-life reward such as lab equipment or a field trip with the students. This element is based on the “Economy” concept from the “ecological” dimension in the taxonomy, meaning monetising teachers’ actions in the environment. We note that this element does not fully coincide with the aforementioned definition of gamification (“do not provide any tangible reward..”, see Sect. 1). However, previous attempts at implementing gamification in different contexts used such mechanisms (e.g., the Spanish league for cardiologists¹). Therefore, we decided to include this extended definition in our first experiment.

An example of an item from the questionnaire, presenting a ‘virtual applause’ gamification element, is presented in Fig. 2.

4.3 Analysis and Results

Following the above participatory design phase, teachers’ ratings and responses to the open-ended questions, as well as the transcription of the group discussion were analysed. As can be seen in Table 1, the two elements that received the highest average ratings are the PeTeL-Dollars (3.67) and the progress bar (3.24). The virtual applause received the lowest rating (1.47).

The rating results were triangulated with the open-ended responses and the group discussion. This analysis yielded the following conclusions:

First, teachers want to have clear goals, and to know their status with respect to them. This was contrasted with the previous design, which sent feedback

¹ <https://ligamosclinicos.com>.

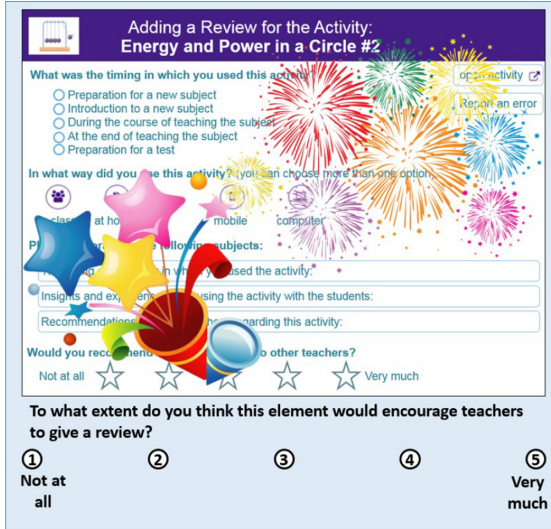


Fig. 2. Example of an item from the gamification questionnaire

request on each resource that was being used, without giving any indication of what is the expected level of contribution.

The second conclusion was that teachers wish to know that their contribution matters, that it is useful for other teachers, and that it helps to improve the environment. This incentive was recognised as much stronger than competition or sensation. This was contrasted with the previous design, in which their feedback was ‘buried somewhere’, and they had no idea whether it was actually being used for anything.

The third, and maybe most surprising finding, was that social recognition matters – we found that for many teachers it was important that their contribution would be seen by the community. This was contrasted with the previous design, in which the individual contribution was not acknowledged. We interpreted this through the prism of the “going green to be seen” [11] phenomenon found among environmentally-aware consumers, who wish to signal a statement about themselves as responsible members of the community (this was used for example to explain the phenomenal success of the Toyota Prius, with its distinctive design, over similarly fuel-efficient cars with conventional design²).

5 Second Experiment: The Effect of Gamification-Driven Design

This experiment was designed to answer the second research question: Does gamification-driven design enhance teachers’ willingness to provide feedback?

² <https://www.theatlantic.com/national/archive/2009/07/prius-effect/21108/>.

5.1 Procedure and Methods

Following the results of the first experiment, two gamification elements were implemented and integrated into PeTeL, which are described below.

Table 1. Teachers' rating of the gamification elements

Teacher	Badges	Points	Progress-bar	Applause	Dollars
1	2	3	3	1	5
2	3	3	4	1	–
3	3	3	4	3	4
4	2	2	3	1	4
5	3	3	5	3	2
6	3	4	5	2	5
7	4	3	3	1	1
8	1	1	1	1	4
9	3	5	3	1	3
10	2	2	4	1	5
11	3	2	2	3	–
12	3	5	2	1	5
13	2	3	4	1	3
14	1	4	3	2	5
15	1	2	1	1	3
16	1	1	3	1	2
17	1	1	5	1	4
Mean	2.24	2.71	3.24	1.47	3.67

Progress Bar. This element addresses the first conclusion – that teachers wish to have a clear goal and know their status with respect to it. A goal of five reviews per year was set (the value was decided by the Physics development team), and a progress bar feature showing for each teacher her progress towards this goal was designed and integrated into PeTeL. It is illustrated in Fig. 3. We note that in the original design presented to the teachers, the progress bar showed the accumulation of information per each **resource** in PeTeL, while the actual progress bar that was implemented showed the number of reviews filled per **teacher**. This change was performed due to a concern that capturing progress by resource will be harder to translate into an evident, global contribution of the individual teacher action to the whole system (which includes many resources). However, the new design still maintained the “progression” and “have a clear goal” dimensions of the original design. In addition, the ‘social’ aspect of the

‘by resource’ progress bar is actually captured by the public bulletin board (see below), while the eventual ‘by teacher’ design addresses the need for having an individual goal and status with respect to it.

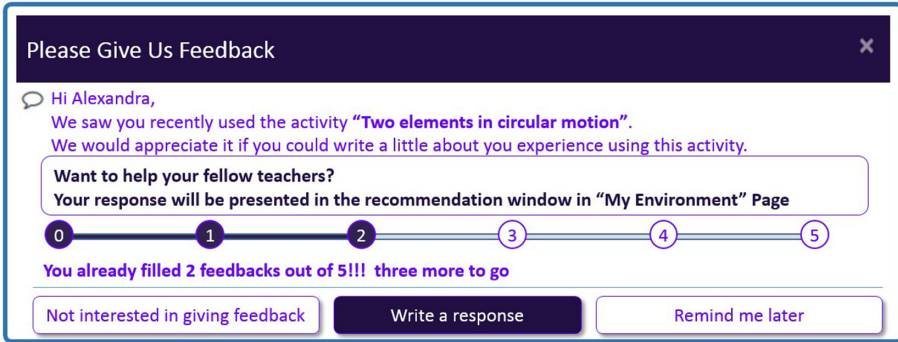


Fig. 3. Request for feedback with progress bar

Bulletin Board. The second element was a ‘bulletin-board’, showing teachers’ feedback on the activities that they have used. It is shown in Fig. 4. The bulletin-board is presented to the teachers in the main page of the learning environment. Each time a teacher reviews an activity, the bulletin-board is updated for all the teachers, with the new review on top and highlighted.

Each input in the bulletin-board contains the name of the teacher who reviewed the learning resource, and the title of the resource that has been reviewed. When hovering with the mouse over the review, a mouseover text showing the details of the review pops-up. The items in the bulletin-board are ‘linkable’, so teachers can easily follow a review, in case that they wish to mark a certain resource for future use in their class.

Although the bulletin-board was not one of the elements presented to the teachers in the first experiment, it addresses two key issues raised by the teachers in the open-ended questions and during the group discussion: First, that they wish to know that their contribution matters. The fact that everyone can see and use their recommendation, and they know that, addresses this. The second is the social recognition, achieved by presenting the name of the teacher who provided the review. We also note that the “PeTeL-Dollars” element was not implemented even though it was ranked highest among the elements, as we decided to avoid tangible rewards and test a model that is sustainable, budget-wise.

5.2 Analysis and Results

We monitored teachers feedback during the first 2 months after these two gamification elements were activated, and compared them with the data we had from

The screenshot shows a web interface for a digital learning platform. At the top, there are navigation tabs: 'My environment', 'Joint repository', and 'Social environment'. The user's name 'Elad Yacobson' is visible in the top right. A sidebar on the left lists subjects like 'mechanics', 'kinematics', 'dynamics', etc. The main area displays '254 Items For Mechanics'. Two resource cards are shown: 'Constant speed – two basic questions' (129 downloads, 1 response) and 'Definitions of Distance and Displacement' (105 downloads, 1 response). Each card includes a diagram, a description, and metadata like 'Level: Easy', 'Device: Mobile/Computer', and 'Duration: 30 Min.'. A sidebar on the right titled 'Teachers Recommending' lists users like Neil Diamond, Paul Young, Joan Baez, and Stevie Wonder, along with their recommended activities.

Fig. 4. Recommendation “bulletin-board”

the previous school year (Sep. 2019 - July 2020). We note that in order to allow for direct comparison, the pop-up review itself was not modified. For our analysis, we additionally took into account the fact that more people turned to online work during the Covid pandemic - what we call the ‘Covid-pandemic effect’.

We accounted for the effect via two metrics: i) The total amount of reviews, normalised by the amount of active teachers; and ii) response rate – the percentage of review requests that are answered.

Total Amount of Reviews. First, we compared the average amount of reviews received each month. Considering the ‘Covid-pandemic effect’, we did not consider only raw numbers, but normalised them by the amount of active teachers. Active teachers are teachers who used at least one learning resource in their class. Comparing the number of active teachers this year and in the previous one yielded that the number of active teachers was very similar (actually somewhat smaller this year, probably due to the shorter time period): 177 active teachers in the previous year (out of which 33 teachers filled reviews = 18.64% of the active teachers), 169 active teachers this year (out of which 34 filled reviews = 20.11% of active teachers). During the previous year, 62 reviews were provided by teachers over a period of 10 months, averaging at 6.2 reviews per month. During the 2 months since the implementation of the gamification elements, we received 56 reviews, an average of 28 reviews per month, more than X4 that of the previous year. Considering however that some of this increase may still be due to active teachers just spending more time online, we continued our analysis.

Response Rate. Next, we measured the difference in the *response rate* before-and-after the implementation of the gamification elements. The response rate is defined as the percentage of feedback requests that are answered by the teachers. Thus, the response rate accounts for other activities that might have increased in the system, such as learning resources usage. Last year, the teachers used a

total amount of 2,372 learning resources, and filled 62 reviews, a response rate of 2.61%. During the 2 months since the implementation of the gamification elements into PeTeL, the teachers used a total amount of 840 learning resources, and filled 56 reviews: A response rate of 6.67%, more than X2.5 increase in comparison to the previous year. A proportion test confirmed that the gamification-driven design generated a significantly higher response rate than the previous design (6.7% versus 2.6%; $z = 5.4$, $p\text{-value} < 0.0001$).

6 Conclusions

This paper describes a pilot research that aims at studying the potential of gamification-driven design as means to incentivise teachers to participate in crowdsourcing activities. Results show that teachers want to have clear goals, to know that their contribution matters, and to be recognised by peers as contributing members of the community. Following these findings, two gamification elements – a progress bar and a bulletin-board presenting teachers' recommendations, were designed and integrated into the learning environment, and their impact on teachers' motivation to provide reviews was measured. Analysing teachers behaviour two months after the new features were aired showed a substantial increase in the amount of reviews provided by the teachers and their response rate, suggesting that the use of gamification can indeed enhance teachers' motivation to take part in crowdsourcing activities, and specifically, in recommending learning resources to other teachers.

References

1. Alamri, A., et al.: An intuitive authoring system for a personalised, social, gamified, visualisation-supporting e-learning system. In: Proceedings of the 2018 the 3rd International Conference on Information and Education Innovations, pp. 57–61 (2018)
2. Anderson, T., Whitelock, D.: The educational semantic web: visioning and practicing the future of education. *J. Interact. Media Educ.* **2004**(1) (2004). <https://jime.open.ac.uk/2004/1>. Special Issue on the Educational Semantic Web. ISSN:1365-893X
3. Aroyo, L., Dicheva, D.: The new challenges for e-learning: the educational semantic web. *J. Educ. Technol. Soc.* **7**(4), 59–69 (2004)
4. Barker, P., Campbell, L.M., Roberts, A., Smythe, C.: *Ims meta-data best practice guide for IEEE 1484.12. 1–2002 standard for learning object metadata* (2006)
5. Bittencourt, I.I., Isotani, S., Costa, E., Mizoguchi, R.: Research directions on semantic web and education. *Interdisc. Stud. Comput. Sci.* **19**(1), 60–67 (2008)
6. Cavusoglu, H., Li, Z., Huang, K.W.: Can gamification motivate voluntary contributions? the case of stackoverflow q&a community. In: Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing, pp. 171–174 (2015)
7. Clements, K.I., Pawlowski, J.M.: User-oriented quality for oer: Understanding teachers' views on re-use, quality, and trust. *J. Comput. Assist. Learn.* **28**(1), 4–14 (2012)

8. Darejeh, A., Salim, S.S.: Gamification solutions to enhance software user engagement-a systematic review. *Int. J. Hum. Comput. Interac.* **32**(8), 613–642 (2016)
9. Deterding, S., Sicart, M., Nacke, L., O’Hara, K., Dixon, D.: Gamification. using game-design elements in non-gaming contexts. In: *CHI 2011 Extended Abstracts on Human Factors in Computing Systems*, pp. 2425–2428 (2011)
10. Downes, S.: Models for sustainable open educational resources. *Interdisc. J. E-Learning Learn. Objects* **3**(1), 29–44 (2007)
11. Griskevicius, V., Tybur, J.M., Van den Bergh, B.: Going green to be seen: status, reputation, and conspicuous conservation. *J. Pers. Soc. Psychol.* **98**(3), 392 (2010)
12. Heffernan, N.T., et al.: The future of adaptive learning: does the crowd hold the key? *Int. J. Artif. Intell. Educ.* **26**(2), 615–644 (2016)
13. Jurado, J.L., Fernandez, A., Collazos, C.A.: Applying gamification in the context of knowledge management. In: *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*, pp. 1–4 (2015)
14. Mayo, M.J.: Games for science and engineering education. *Commun. ACM* **50**(7), 30–35 (2007)
15. Porcello, D., Hsi, S.: Crowdsourcing and curating online education resources. *Science* **341**(6143), 240–241 (2013)
16. Quinn, A.J., Bederson, B.B.: Human computation: a survey and taxonomy of a growing field. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1403–1412 (2011)
17. Shi, L., Cristea, A.I.: Motivational gamification strategies rooted in self-determination theory for social adaptive e-learning. In: Micarelli, A., Stamper, J., Panourgia, K. (eds.) *ITS 2016. LNCS*, vol. 9684, pp. 294–300. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39583-8_32
18. Silva, F., Toda, A., Isotani, S.: Towards a link between instructional approaches and gamification-a case study in a programming course. In: *Anais do Workshop de Informática na Escola*. vol. 24, p. 157 (2018)
19. Thiebes, S., Lins, S., Basten, D.: Gamifying information systems-a synthesis of gamification mechanics and dynamics (2014)
20. Toda, A., et al.: *GamiCSM: Relating Education, Culture and Gamification - a Link between Worlds*. Association for Computing Machinery, New York (2020). <https://doi.org/10.1145/3424953.3426490>
21. Toda, A., et al.: A taxonomy of game elements for gamification in educational contexts: Proposal and evaluation. In: *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, vol. 2161, pp. 84–88. IEEE (2019)
22. Toda, A.M., et al.: Analysing gamification elements in educational environments - using an existing gamification taxonomy. *Smart Learn. Environ.* **6**(1), 16 (2019). DOI: <https://doi.org/10.1186/s40561-019-0106-1>, <https://slejournal.springeropen.com/articles/10.1186/s40561-019-0106-1>
23. Toda, A.M., et al.: How to gamify learning systems? an experience report using the design sprint method and a taxonomy for gamification elements in education. *J. Educ. Technol. Soc.* **22**(3), 47–60 (2019)



Automatic Adaptive Sequencing in a Webgame

Tong Mu¹(✉), Shuhan Wang², Erik Andersen², and Emma Brunskill¹

¹ Stanford University, Stanford, USA
{tongm,ebrun}@cs.stanford.edu
² Cornell University, Ithaca, USA

Abstract. Intelligent tutoring systems can improve student outcomes, but developing such systems typically requires significant expertise or prior data of students using the system. In this work we propose a new approach for automatically adaptively sequencing practice activities for an individual student. Our approach builds on progress for automatically constructing curriculum graphs and advancing a student through a graph using a multi-armed bandit algorithm. These approaches have relatively few hyperparameters and are designed to work well given limited or no prior data. We evaluate our method, which can be applied to a diverse range of domains, in our online game for basic Korean language learning and found promising initial results. Compared to an expert-designed fixed ordering, our adaptive algorithm had a statistically significant positive effect on a learning efficiency metric defined using in game performance.

Keywords: Adaptive sequencing · Automatic curriculum generation · Educational games

1 Introduction

As educational technology continues to grow in popularity, it is desirable to have scalable, robust methods for creating efficient and adaptive learning pathways for new tutoring systems. When no prior data is available, one potential way to ensure that a new adaptive tutoring system will yield strong learning benefits is to heavily involve experts to create a curriculum structure and specify model parameters of an assumed student model. However this type of approach can be time consuming and difficult to scale. To lighten this burden of human expertise, data driven approaches use collected data to create or refine statistical models of student learning (e.g. [6, 9, 15]). Such work has demonstrated significant improvements in student learning outcomes and/or learning efficiency. However this approach still requires either a significant amount of expert input to ensure the initial learning pathways (when only a few students have used the system) are beneficial, or may have significantly suboptimal performance for students early on.

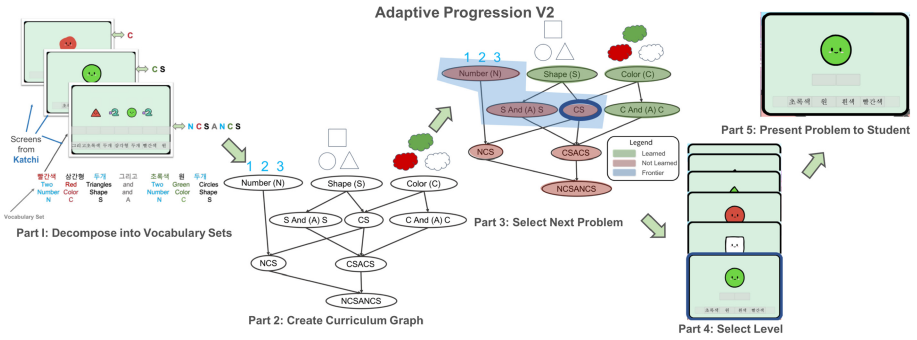


Fig. 1. Full pipeline of our adaptive algorithm applied to Katchi. Part 1 shows the grouping of individual items (such as Red and Green) into concepts (such as Color) and grouping problems into nodes, Part 2 shows organizing these nodes into a curriculum graph, Part 3 shows the adaptive algorithm’s internal belief state of the mastery of each node, Part 4 shows selecting a specific level from within the selected node, and Part 5 shows presenting the selected problem to the student. (Color figure online)

In our work we propose a new method that, given a set of activities, will automatically create adaptive learning pathways for students without requiring prior data. Our approach builds on several developments: methods that take a set of skills labeled with features and automatically constructs a knowledge (prerequisites) graph among this set [2,21], and the work to use multi-armed bandits to progress through such a graph [11]. This bandit based method only has a few hyperparameters, reducing the burden of expert time needed, and is demonstrated to potentially be more robust to variability in the student learning process [10].

Our method can be used in any domain where solving a pedagogical activity can be described by a simple program or by the execution trace of that program, which has been shown to cover diverse domains including basic arithmetic [2], logical proofs [1], and algebra [16]. We experimentally investigate our method in a Korean language learning webgame and perform analysis using evaluation metrics based on in game performance. We found that while there was no significant difference in total learning, learners in the adaptive progression condition had a statistically significant higher learning efficiency compared to students in the expert-designed fixed progression condition. This initial finding highlights the potential of our method to increase learning efficiency and save students time.

Related Work. Our work considers the question of adaptively sequencing educational content. Bayesian Knowledge Tracing (BKT) [12] can be used to adaptively monitor a student’s learning given a decomposition of problems into knowledge components (KCs), but does not specify how to select amongst the unknown skills.

Various recent approaches [5, 13, 15, 19] often found promising results. However our work differs in that these previous works all use prior student data to create an adaptive algorithm. Most recently, Bassen et al. [4] used deep reinforcement learning (RL) with minimal expert input and no prior data to learn to sequence problems efficiently. They demonstrate good performance for learners after the experience from the first 200 learners. In general, even the most efficient deep RL methods will require at least such an amount of experience to achieve good performance. Their method does not discuss a way to enforce a curriculum graph for the initial learners when concepts and activities exhibit strong prerequisite dependencies as they do in our domain. Methods such as ours that can enforce a curriculum can improve the experience for early learners.

Our work builds on the adaptive algorithm presented by Clement et al. that automatically provides personalized student advancement through curriculum graphs using concepts of a ZPD and multi-armed bandits [11]. This work previously assumes an expert specified curriculum graph was given, which we do not. In our work, we automatically generate a curriculum graph from a set of practice activities using developments in automatic curriculum generation using execution traces [2, 20]. The use of the curricula generated by these automatic methods for adaptive sequencing have not been previously investigated. As a result, our work aims to provide, to our knowledge, one of the first evaluations of a system that, using no prior data and very little expert input, automatically creates a curriculum graph and an adaptive progression for students from a set of practice items.

2 Method

Our method is illustrated in Fig. 1 and described in more detail below.

2.1 Domain: Katchi, a Korean Language Learning Webgame

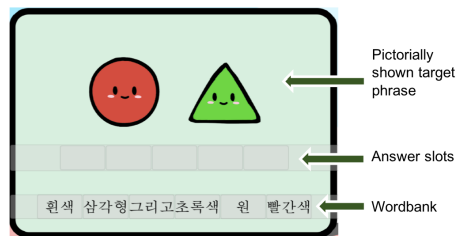


Fig. 2. A Screenshot of the *Red Circle And Green Triangle* activity of Katchi. learners drag and drop words from the wordbank at the bottom of the screen to the answer slots to describe the pictorially shown target phrase. (Color figure online)

We run our experiments on our online Korean learning webgame *Katchi*. The goal of Katchi is for students to master simple Korean phrases of the form

Number Color Shape And Number Color Shape for all possible combinations of the taught vocabulary. To pass an activity (Fig. 2), learners must drag and drop the Korean words from the wordbank into the correct answer slots to describe the pictorially shown target phrase. Activities vary in complexity in terms of the number of words needed, ranging from having 1 to 7 empty slots. Learners are allowed an unlimited amount of attempts on each activity and must answer the activity correctly to advance. An activity bank containing 402 unique activities was created.

2.2 Automated Curriculum Graph Creation

We first build a hierarchical structure to model difficulty dependencies between a set of pedagogical items. We follow Wang et al.'s [2,21] work in automatic curriculum generation for language learning and define:

Definition 1 ([21]). *A phrase s_1 is harder than another phrase s_2 , indicated as $s_1 > s_2$, if s_1 is longer than and covers all the vocabulary words in s_2 . A phrase s_1 is directly harder than s_2 if there does not exist a third phrase such that $s_1 > s_3 > s_2$.*

This definition implies a directed graph where a directed edge represents a “directly harder than” relation between two phrases. We represent each level that involves multiple words as a high level conceptual phrase (for example, a node is *color shape*, and the activities *red triangle* and *green square* will both fall under this node). We represent all single word levels as separate nodes to ensure students will learn each of the basic vocabulary words. The curriculum graph used can be seen in Part 2 of Fig. 1.

While we evaluate our method in language learning, there exist similar curriculum generation methods that use solution execution traces, or the steps taken to solve an educational activity [2]. These methods can be applied to any domain where automatic solution generation is possible, which include a diverse range of domains such as arithmetic and logical proofs.

2.3 Automatic Progression Through Curriculum Graph

Given a curriculum graph \mathcal{G} over a set of nodes representing concepts and with one or more practice materials mapped to each node, our algorithm for selecting the next item to practice consists of combining a model of forgetting [17] and a model of the zone of proximal development (ZPD) [11]. We incorporate a model of forgetting as we consider a language learning setting where forgetting is known to be very important [3,7,17] and spaced repetition has long been a gold standard. Overall our method induces a policy that interleaves review activities with learning activities on which the student is making the fastest learning progress.

Progression Metric: To progress students to items farther in the curriculum, we need a signal of learning. Measuring the true knowledge state of students is a key challenge in the online game setting. As a proxy, we use correctness on the first attempt on an activity from the node to mark the node as learned.

Node Selection: To select a node from a curriculum graph to present to the student, our method tracks three subsets of nodes for that student as shown in Fig. 1 Part 3: the learned set (\mathcal{L}), the not learned set, and the frontier set (\mathcal{ZPD}). The frontier set consists of nodes in the not learned set on the boundary of the learned set. Such items are considered to be the Zone of Proximal Development (ZPD). Following prior work in psychology that hypothesized students will learn best when they are given activities that are at the appropriate level of difficulty [8], items in the ZPD are prime targets for learning. Upon initialization, all nodes with no incoming edges define the frontier and all nodes are in the not learned set. To select a node, the algorithm first checks if there are any review nodes from \mathcal{L} that should be presented (described further below). If not, a learning node from \mathcal{ZPD} is chosen following the bandit based ZPDES algorithm [11] (See original paper for details). A node is moved to \mathcal{L} once student performance on the node satisfies the mastery progression metric and an unlearned node is added to \mathcal{ZPD} if all prerequisites of that node are marked as learned.

Review Nodes: To incorporate review, we track potential forgetting using the MCM model of forgetting [17]. MCM models the memory strength of items using a sum of exponential decaying memory traces left each time an item is reviewed. A node is marked as needing review if its memory strength is lower than a threshold.

Selecting an Item From a Node: Once a node is selected, an activity from the node needs to be selected. Activities need to be selected in a way that ensures all basic components are practiced and students understand the concept generally taught by each node as opposed to only subsets of its instantiation. To give an extreme example of an undesired situation, if a student only sees *Circle* in problems that involve *Shapes* and never experiences *Square*, then a student who started out as a novice would not be able to complete problems involving *Square* even if they could complete problems of that node with *Circle*. It is infeasible to include all 402 unique activities so we instead ensure students see a varying array of vocabulary through time which the expert baseline was carefully designed to do. To ensure this in our adaptive algorithm, we use a second MCM model over each basic vocabulary word and at every timestep we choose the item from the node with the vocabulary word that has been practiced least recently.

3 Preliminary Experiment

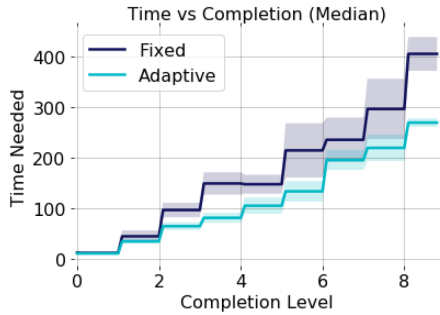


Fig. 3. Results from our experiment. Figure (a) shows completion level with time, with 95% confidence intervals shaded. We see the adaptive progression allows learners to reach completion levels with consistently less time than the fixed condition.

We compared our method, which we will refer to as the **Adaptive** progression, with an expert designed **Fixed** progression. The Fixed progression consists of 43 levels, 34 of which are unique, and was carefully created to achieve the learning goals of Katchi. We posted our game on a popular gaming website, Newgrounds¹ and uniformly at random assigned each learner to a condition. We collected data from 117 and 89 learners assigned to the Adaptive and Fixed conditions respectively.

Evaluation Metric: In this initial experiment, we focus on using within task signals for evaluation. Correctness on the first attempt is a strong signal of learning as for all activities that have more than 1 slot (which is 6 out of the 9 nodes, and 393 of the 402 possible unique activities) the probability of guess is very low. For the simplest multi-slot activity, the *Color Shape* activity, the probability of answering correctly through random guess is $\frac{1}{12}$. For the most complex, such as the activity *Two Green Circles And Three Red Triangles*, the probability of guess is $\frac{1}{5040}$. We use this to define a completion metric which counts a node as **completed** once a learner answers a problem from that node correctly on the first attempt. We define the **completion level** of a learner at a given time as the number of unique nodes they have completed so far. We evaluate the overall learning efficiency in terms of number of completed nodes per minute (**CNPM**), of all the learners before dropout.

Results: We did not find a statistically significant difference in the average amount of completion upon dropout among conditions, meaning learners dropped out at similar stages of material difficulty (the mean completion level at

¹ <https://www.newgrounds.com/>.

dropout for Fixed and Adaptive were 2.6 and 2.4 respectively, a Mann-Whitney-U test results in $p = 0.37$ ($U = 2457$). However we did find a statistically significant difference in learning efficiency. We found the adaptive algorithm overall enabled learners to progress through the same material faster than the fixed progression. On average, learners in the adaptive progression progressed at 1.7 CNPM while learners in the fixed progression progressed at 1.2 CNPM. Figure 3 shows completion through time. Due to differential dropout, there is overlap at different points, but overall the adaptive progression is progressing learners faster. Accounting for multiple comparisons using the Bonferroni Correction, the result of a Mann-Whitney-U test suggested there was a statistically significant difference between the conditions at the $\alpha = 0.01$ level, ($U = 1677$, $p = 0.003$). Increases of learning efficiency are very beneficial as it allows learners to save time and mental energy to put towards other studies. This is especially so as there has been research that shows faster and slower learners that learn a concept to the same level of mastery, irregardless of the amount of practice items needed to reach that level, showed the same strength of learning in terms of rate of forgetting [6, 14, 18]. Therefore our initial findings which suggests our method can potentially increase learning efficiency is promising.

4 Conclusion

Following evidence that creating personalized and adaptive educational systems can lead to improved learning, our work presented a novel system that can take in practice items and descriptions of those items in terms of its underlying skills, and automatically create an adaptive personalized sequence of the material for a student. Key features of our adaptive algorithm is that it only needs minimal expert input and does not require existing student data to set the hyperparameters of adaptivity. Our method can be applied to a wide range of domains and we run a preliminary study in a language learning domain to examine the effectiveness of our method. We found initial evidence our method may able to increase learning efficiency compared to a strong fixed progression.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1657176 and the BIGDATA award. It is also supported by the Stanford Human Centered AI HoffmanYee grant and the Graduate Fellowships for STEM Diversity. Additionally, we thank Brandon Cohen, Nicholas Teo, Evan Adler, Urael Xu, and Nicole Cheung for their work in creating Katchi.

References

1. Ahmed, U.Z., Gulwani, S., Karkare, A.: Automatically generating problems and solutions for natural deduction. In: 23rd International Joint Conference on Artificial Intelligence (2013)
2. Andersen, E., Gulwani, S., Popovic, Z.: A trace-based framework for analyzing and synthesizing educational progressions. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 773–782. ACM (2013)

3. Bahrick, H.P., Bahrick, L.E., Bahrick, A.S., Bahrick, P.E.: Maintenance of foreign language vocabulary and the spacing effect. *Psychol. Sci.* **4**(5), 316–321 (1993)
4. Bassen, J., et al.: Reinforcement learning for the adaptive scheduling of educational activities. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–12 (2020)
5. Brinkhuis, M.J., Savi, A.O., Hofman, A.D., Coomans, F., van der Maas, H.L., Maris, G.: Learning as it happens: a decade of analyzing and shaping a large-scale online learning system. *J. Learn. Anal.* **5**(2), 29–46 (2018)
6. Cen, H., Koedinger, K.R., Junker, B.: Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. *Front. Artif. Intell. Appl.* **158**, 511 (2007)
7. Cepeda, N.J., Pashler, H., Vul, E., Wixted, J.T., Rohrer, D.: Distributed practice in verbal recall tasks: a review and quantitative synthesis. *Psychol. Bull.* **132**(3), 354 (2006)
8. Chaiklin, S.: The zone of proximal development in Vygotsky’s analysis of learning and instruction. In: Vygotsky’s Educational Theory in cultural Context, vol. 1, pp. 39–64 (2003)
9. Chi, M., VanLehn, K., Litman, D., Jordan, P.: Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Model. User Adap. Inter.* **21**(1–2), 137–180 (2011)
10. Clement, B., Oudeyer, P.Y., Lopes, M.: A comparison of automatic teaching strategies for heterogeneous student populations. In: 9th International Conference on Educational Data Mining, EDM 2016 (2016)
11. Clement, B., Roy, D., Oudeyer, P.Y., Lopes, M.: Multi-armed bandits for intelligent tutoring systems. *J. Educ. Data Min. (JEDM)* **7**(2), 20–48 (2015)
12. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User Adap. Inter.* **4**(4), 253–278 (1994)
13. David, Y.B., Segal, A., Gal, Y.K.: Sequencing educational content in classrooms using Bayesian knowledge tracing. In: Proceedings of the 6th International Conference on Learning Analytics & Knowledge, pp. 354–363. ACM (2016)
14. Gentile, J.R., Voelkl, K.E., Pleasant, J.M., Monaco, N.M.: Recall after relearning by fast and slow learners. *J. Exp. Educ.* **63**(3), 185–197 (1995)
15. Mandel, T., Liu, Y.E., Levine, S., Brunskill, E., Popovic, Z.: Offline policy evaluation across representations with applications to educational games. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, pp. 1077–1084. International Foundation for Autonomous Agents and Multiagent Systems (2014)
16. O’Rourke, E., Butler, E., Díaz Tolentino, A., Popović, Z.: Automatic generation of problems and explanations for an intelligent algebra tutor. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) AIED 2019. LNCS (LNAI), vol. 11625, pp. 383–395. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23204-7_32
17. Pashler, H., Cepeda, N., Lindsey, R.V., Vul, E., Mozer, M.C.: Predicting the optimal spacing of study: a multiscale context model of memory. In: Advances in Neural Information Processing Systems, pp. 1321–1329 (2009)
18. Underwood, B.J.: Speed of learning and amount retained: a consideration of methodology. *Psychol. Bull.* **51**(3), 276 (1954)
19. Vainas, O., et al.: E-gotsky: sequencing content using the zone of proximal development. arXiv preprint [arXiv:1904.12268](https://arxiv.org/abs/1904.12268) (2019)

20. Wang, S., Andersen, E.: Grammatical templates: improving text difficulty evaluation for language learners. In: The 26th International Conference on Computational Linguistics: Technical Papers, Proceedings of COLING 2016, pp. 1692–1702 (2016)
21. Wang, S., He, F., Andersen, E.: A unified framework for knowledge assessment and progression analysis and design. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 937–948. ACM (2017)



Towards Smart Edutainment Applications for Young Children. A Proposal

Adriana-Mihaela Guran^(✉), Grigoreta-Sofia Cojocar^{},
and Laura-Silvia Dioşan^{}

Faculty of Mathematics and Computer Science, Babeş-Bolyai University,
Cluj-Napoca, Romania
{adriana,grigo,lauras}@cs.ubbcluj.ro

Abstract. Education, like the large majority of domains, has been impacted by the rapid development of communications and technology. What was perceived before as an ideal, i.e., the enhancement of the learning process through modern techniques, now it has been rapidly transformed into a mandatory requirement due to the current COVID-19 pandemic [13]. Edutainment applications are meant to support the learning activities of young children even in the absence of an in-person teacher. In this paper, we present our proposal for developing smart edutainment applications for young children that allow automatic identification of the child and adaptation of the interaction flow based on the child's emotions.

Keywords: Edutainment · Adaptation · Learning · Emotion recognition · Affective computing

1 Introduction

Education technology is continuously expanding in parallel with the technical progress (projectors, smart boards, etc.) and it enables better interaction between teachers and students in the classroom. Every day, various aspects of education technology are becoming an inherent part of the educational experience for students, teachers, parents, and management. Nowadays, even young children are confronted with the need to study more often using various interactive applications (digital story-telling, edutainment, game-based learning tools), sometimes in the absence of an in-person teacher. That is why we consider that digital educational resources for young children should address now not only the learning content, but they should provide a customized environment for learning and they should recognize and even adapt to different learners emotions [6, 8]. To our knowledge, there aren't any approaches proposed for the development of smart edutainment applications that adapt based on young children's emotions, yet. There are several approaches for adult learners that integrate emotion awareness into learning support tools. For example, Feidakis gives in [3] an overview of the emotion-aware systems developed for e-learning in virtual environments.

Ruiz [12] proposed a method to measure students mood based on a model of twelve positive and negative emotions, making use of self-report and measuring the interactions with the teachers.

In this paper, we propose an approach to enhance edutainment applications for young children (aged 3 to 6 years) with smart capabilities: automatic identification of the child and interaction flow adaptation based on the child's emotions. For the development of these capabilities, we propose to use Artificial Intelligence (AI) techniques. We describe in this paper our preliminary studies in developing such applications and discuss future directions.

2 Smart Edutainment Applications for Young Children

Pekrun [9] has identified the so-called *academic emotions* and found that positive mood supports holistic, creative ways of thinking and negative emotions, such as anger, sadness, fear, boredom, are negatively related to the learning process and learning outcomes. Positive emotions are positively related to the learning process and outcomes, while negative emotions are detrimental to motivation, performance, and learning in many situations [10, 11].

Our proposal is to develop smart edutainment applications for young children in order to keep them in a positive emotional state to ensure an optimal learning environment [2, 7, 9] and to support learning at their own pace. In our vision, a smart edutainment application should contain an *edutainment* part responsible for presenting the learning content and the tasks to support the understanding of the new knowledge and an *emotion recognition* part responsible for the identification of the user's emotional state. The two modules should be executed in parallel and should communicate, as shown in Fig. 1.

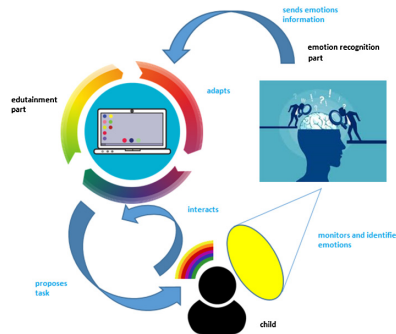


Fig. 1. High-level view of a smart edutainment application.

However, designing and implementing this kind of edutainment applications is not easy as various challenging aspects, some of them due to the young age of the users, must be overcome: how to identify the user (i.e., authentication),

how to detect the emotional state of the user, and when and how to adapt the interaction with the application when changes in the user's emotional state appear.

2.1 Young Children's Authentication

The authentication of the young user is important for personalized interaction, but also for evaluating and monitoring the learner's progress. The classical approach with a username and password is not feasible as children aged 3–6 years old do not have the required reading/writing skills, yet. Another solution that is used as an alternative to the classical one is face authentication. The existing face recognition techniques available today can reach an accuracy greater than 99% for adults. However, the accuracy decreases when the face to be recognized belongs to a young child. We have performed a preliminary study, in which computer science master students have used different AI techniques for automatic young children face recognition. The obtained results have an accuracy between 95% and 99.38% and depend on the face-recognition models used. The models based on mechanisms like Histogram of Oriented Gradients (HOG) [1] have lower accuracy than the models based on Convolutional Neural Networks (CNNs) [4]. The used models also affect the response time. The HOG-based models are somehow less accurate than CNN-based models, but they are simpler and faster, needing less time for identifying the child's face.

2.2 Young Children's Emotion Recognition

In order to adapt the interaction flow of an edutainment application to a child's emotional state, first we have to be able to automatically identify the child's emotions. If the existing approaches for face recognition have reached an accuracy higher than 99% for adult faces, this is not true for automatic emotion recognition approaches. The accuracy decreases even more for children's automatic emotion recognition. We have conducted studies about the accuracy of different emotion recognition approaches from faces in the case of young children [5]. In a more recent study, a team of computer science master students have built a new dataset with face images of young children and they have used a CNN to identify children's emotions. On this new dataset composed of fifty images for each of the six investigated emotions (happiness, sadness, disgust, anger, surprise and neutral), an accuracy of 70% was obtained, that represents an improvement by 2% related to those described in [5]. Analysis of the obtained results show that the children's age is a factor that highly influences the recognition accuracy, possibly due to certain transitions that can be much more pronounced in younger children (such as the cheeks). In addition, there are confusing emotions that have similar effects on their faces.

2.3 Interaction Adaptation

The adaptation decisions of a smart edutainment application are determined by the identification of negative emotions like frustration, anger, or boredom. The

interaction with the edutainment application should start only if the child is in a positive emotional state and should continue only if negative emotions are not detected. Otherwise, the edutainment application should propose activities to change the child's emotional state by providing encouraging messages, friendly messages, or by proposing some entertaining physical activities.

We have conducted a preliminary study to test our idea of modifying an edutainment application based on the results of a facial expression emotion recognizer. A team of computer science master students has developed a prototype application in which for a selected edutainment material (in this study, a video animation), the application adds a filter to the edutainment based on the automatically identified emotions of the viewer. For example, when anger is detected a distorted effect is applied, or when disgust is detected, a blurred effect is applied. For testing purposes, we have used adults viewers due to the improved accuracy of emotion recognition compared to the results obtained by our approach on children images. Some examples of video animation changes based on the viewer's emotions are presented in Fig. 2.

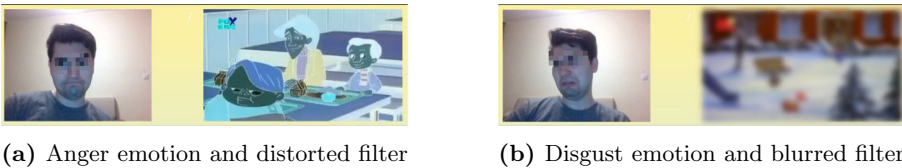


Fig. 2. Emotion-based adaptation of a video

The results of our study show a promising starting point towards adapting the interaction flow of an edutainment application based on a young child's emotions. An important aspect revealed by this study that requires further investigation is the delayed adaptation. It takes 2–3 s until the addition of the filter is visible to the viewer. This is important as it may also impact the time needed to adapt the edutainment application.

In conclusion, our preliminary studies show that the development of smart edutainment applications for young children is possible, but some optimizations for the emotion recognition and interaction adaptation must be considered.

3 Conclusions and Further Work

In this paper, we have presented our vision on developing smart edutainment applications for young children by personalizing and adapting the interaction. In the future we intend to validate our proposal on real case studies, to evaluate the appropriateness of the adaptation approach to real settings, and to extract emotion information from multiple channels (body posture, voice, sensors).


Acknowledgment. The authors would like to thank to the computer science master students involved in the described studies, to the children participating in activities, and to their parents for agreeing it.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, pp. 886–893. IEEE (2005)
2. Denham, S.A., Bassett, H.H., Thayer, S.K., Mincic, M.S., Sirotkin, Y.S., Zinsser, K.: Observing preschoolers' social-emotional behavior: structure, foundations, and prediction of early school success. *J. Genetic Psychol.* **173**, 246–278 (2012)
3. Feidakis, M.: A review of emotion-aware systems for e-learning in virtual environments, Chap. 11. In: Caballé, S., Clarisó, R. (eds.) *Formative Assessment, Learning Data Analytics and Gamification, Intelligent Data-Centric Systems*. Academic Press, pp. 217–242 (2016). ISBN 9780128036372. <https://doi.org/10.1016/B978-0-12-803637-2.00011-7>
4. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press (2016)
5. Guran, A.M., Cojocar, G.S., Diosan, L.: A step towards preschoolers' satisfaction assessment support by facial expression emotions identification. In: *Proceedings of the 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - Procedia Computer Science 2020*, vol. 176, pp. 632–641 (2020)
6. Hyson, M.: *The Emotional Development of Young Children: Building an Emotion-Centered Curriculum*, 2nd edn. Teachers College Press, New York (2004)
7. Ortony, A., Clore, G.L., Collins, A.: *The Cognitive Structure of Emotions*. University Press, UK Cambridge (1988). <https://doi.org/10.1017/CBO9780511571299>
8. O'Connor, E., McCartney, K.: Examining teacher-child relationships and achievement as part of an ecological model of development. *Am. Educ. Res. J.* **44**(2), 340–369 (2007)
9. Pekrun, R.: The impact of emotions on learning and achievement: towards a theory of cognitive/motivational mediators. *Appl. Psychol.* **41**, 359–376 (1992)
10. Pekrun, R., Lichtenfeld, S., Marsh, H.W., Murayama, K., Goetz, T.: Achievement emotions and academic performance: longitudinal models of reciprocal effects. *Child Dev.* **88**, 1653–1670 (2017). <https://doi.org/10.1111/cdev.12704>
11. Rowe, A.D., Fitness, J.: Understanding the role of negative emotions in adult learning and achievement: a social functional perspective. *Behav. Sci. (Basel)* **8**(2), 27 (2018). <https://doi.org/10.3390/bs8020027>
12. Ruiz, S., Urretavizcaya, M., Fernández-Castro, I., López-Gil, J.-M.: Visualizing students' performance in the classroom: towards effective F2F interaction modelling. In: Conole, G., Klobučar, T., Rensing, C., Konert, J., Lavoué, É. (eds.) *EC-TEL 2015. LNCS*, vol. 9307, pp. 630–633. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24258-3_75
13. UNESCO: Covid-19 Crisis: UNESCO Call to Support Learning and Knowledge Sharing through Open Educational Resources. <https://en.unesco.org/news/covid-19-crisis-unesco-call-support-learning-and-knowledge-sharing-through-open-educational>



Do Students Use Semantics When Solving Parsons Puzzles? – A Log-Based Investigation

Amruth N. Kumar^(✉) 

Ramapo College of New Jersey, Mahwah, NJ 07430, USA
amruth@ramapo.edu

Abstract. Parsons puzzles are jigsaw puzzles wherein students are given a program in scrambled order and tasked with reassembling the program in its correct order. Do students use program semantics when solving Parsons puzzles? The answer to this question has implications for the use of Parsons puzzles as a pedagogic tool. In order to answer this question, we considered semantics at the level of statements and control-flow. We analyzed the data collected by a Parsons puzzle tutor over 5 semesters and measured the extent to which students' puzzle-solving behavior conformed with the use of statement-level and control-flow semantics. We found that students used statement-level semantics to assemble up to 73% of the lines in a puzzle and control-flow semantics to assemble up to 47% of the lines. They used statement-level semantics more than control-flow semantics and more on some puzzles than others. Whenever we found a significant difference between C++ and Java students, C++ students used semantics more than Java students. Finally, we did not find an increase in the use of semantics with increased practice.

Keywords: Parsons puzzle · Puzzle-solving strategy · Program semantics

1 Introduction

In a Parsons puzzle [10], first proposed as an engaging way to learn programming, the student is given a program in scrambled order and asked to reassemble it in its correct order. Some studies have found that Parsons puzzles are effective for learning programming [1–3]. Researchers have looked into what helps students solve Parsons puzzles (e.g., [2, 4, 6–9]). But, to date, no research has been done on why the puzzles are effective. For example, do students use program semantics when solving the puzzles? We attempted to answer this question. We considered semantics at two levels:

1. **Statement-level semantics:** Do students reassemble lines that appear next to each other in a program one after the other? For example, do students pick line 21 to assemble after assembling line 20 (or vice versa)? A student who does not use statement-level semantics might assemble the program in seemingly random order: line 35 after line 20, line 8 after line 35, and so on.

2. **Control-flow semantics:** Do students reassemble sections of code in a program in the order in which control flows through the sections when the program is executed? For example, do students assemble input section before output section? The order among the sections of code is a partial order, e.g., both if-clause and else-clause in an if-else statement are conditionally executed before any section that appears after the if-else statement. But the two clauses themselves may be assembled in either order, as long as one clause is assembled completely before the other clause is attempted.

Note that statement-level semantics determines which lines should be assembled together within each section of code (e.g., input section, compute section, or output section). Control-flow semantics builds upon statement-level semantics by superimposing an order among the sections of code (e.g., input section must precede compute section). The hypotheses of our study were:

1. RQ1: Students use statement-level semantics when solving Parsons puzzles;
2. RQ2: Students use control-flow semantics when solving Parsons puzzles;
3. RQ3: The use of statement-level and control-flow semantics increases with practice.

2 Computing the Use of Semantics

We define **action sequence** as the order of the lines in the puzzle to which a student applies permissible actions during the puzzle-solving process. For example, in [4, 1, 4, 3, 2, 3] the student assembles 4th line first, but decides to reorder it after assembling the 1st line. Assuming the student is required to solve the puzzle completely and correctly, **solution sequence** is the order in which the lines in the puzzle are placed in their *final/correct* location. For example, solution sequence corresponding to the above action sequence is [1, 4, 2, 3]. The two sequences differ when a student revises the order before arriving at the correct solution to the puzzle. For our study, we analyzed the solution sequences of students.

We define **semantic sequences** as the possible orders in which a student might solve a puzzle using either statement-level or control-flow semantics. A puzzle may have several semantic sequences. In order to compute a student's use of semantics when solving a puzzle, we compared the student's solution sequence against each semantic sequence for the puzzle as follows:

- Statement-level semantics is the sum of the number of lines in the solution sequence that appear in the same order in the semantic sequence. For example, if the solution sequence is [3,4,1,2,6,5] and a semantic sequence for the puzzle is [1,2,3,4,5,6], the use of statement-level semantics is 4 corresponding to the subsequences [3,4] and [1,2].
- Control-flow semantics is computed by giving credit to only the subsequences in the solution sequence that appear in the same relative order as in a semantic sequence. In the earlier example, the numerical value returned is 2, corresponding to the subsequence [3,4] – the subsequence [1,2] is not credited because it should have appeared before the subsequence [3,4]. If the solution sequence is [3,4,1,2,5,6] instead, the

numerical value returned is 4, corresponding to the subsequences [3,4] and [5,6] which are in correct relative order.

So, computation of statement-level semantics credits all matching subsequences, even if they are out of order, whereas control-flow semantics credits only matching subsequences that are in correct relative order, even if non-contiguous. The degree of use of semantics of a student on a puzzle is the maximum of comparing the student's solution sequence against all the semantic sequences for the puzzle.

3 The Study

For this study, we used epplets (epplets.org), a suite of software tutors on Parsons puzzles available freely online for educational use [5]. The tutors present Parsons puzzles, which students solve using drag-and-drop actions. Students are required to solve each puzzle completely and correctly before going on to the next puzzle. A puzzle containing n lines can be solved with n actions. If a student takes more than 10% redundant actions to solve a puzzle, the tutors schedule additional similar puzzles for the student to solve. The tutors log the sequence of actions taken by students to solve the puzzles.

For this study, we analyzed the data collected by the tutor on if-else statements. The first two puzzles presented by the tutor were on the following programs:

1. A program to read two numbers and print the smaller value among them. This puzzle was on the concept of a single if-else statement in the program.
2. A program to read numerical grade, convert it to letter grade and print it. This puzzle was based on the concept of cascading nested if-else statements, i.e., nesting in either if-clause or else-clause, but not both.

The tutor was used by introductory programming students, both majors and non-majors, as after-class assignments. For this study, we used the data collected by the tutor over five semesters: Fall 2016 – Fall 2018. Data was included in the study from only the students who gave permission for their data to be used for research purposes. When students used the tutor multiple times, data was included only from their first use of the tutor so that results were not affected by practice effect. Both C++ and Java students used the tutor. Where possible, we will separate our analysis by language.

When solving puzzles using the tutor, students were instructed to completely and correctly reassemble each puzzle. They were not instructed to use any particular strategy to solve the puzzles. Students could either solve a puzzle completely and correctly or bail out by clicking on Quit button. If students attempted to submit an incomplete solution, the tutor asked them to complete assembling the unassembled lines in the puzzle. If students attempted to submit a complete but incorrect solution, the tutor highlighted the first misplaced line in the assembled program and suggested whether the line had to be moved earlier or later in the program. Each puzzle contained two distracters. The student was expected to delete them.

Once the puzzle-solving session ended, the tutor logged the click-stream data of the session, including every permissible operation applied by the student to solve each puzzle. This was the data we used to compute the action sequence and solution sequence of

each student for each puzzle. Permissible actions applied to distracters were not included in the solution sequence since distracters were not part of the correctly assembled program.

4 The Results

The first puzzle in if-else tutor contained 16 lines of code and included a single if-else statement. The mean statement-level and control-flow semantics scores for the puzzle, along with 95% confidence interval, separated by programming language, are shown in Table 1.

Table 1. Mean scores on the if-else puzzle no. 1 containing 16 lines of code.

Puzzle 1	N	Statement-level semantics \checkmark	Control-flow semantics \checkmark
C++	98	10.37 \pm 0.41	6.57 \pm 0.51
Java	302	9.72 \pm 0.20	5.79 \pm 0.30

Note that mean statement-level semantics score was around 10 lines out of 16 for both the languages, and mean control-flow semantics score ranged from 5.79 to 6.57 lines. ANOVA analysis of the two scores as dependent variables and programming language as the fixed factor yielded a significant main effect for language on both the scores: statement-level semantics [$F(1,409) = 10.56, p = 0.001$] and control-flow semantics [$F(1,409) = 6.64, p = 0.01$]. In both the cases, the scores as shown in Table 1 were statistically significantly higher for C++ than Java. In the tables, statistically significant differences are marked with \checkmark .

The second puzzle contained 36 lines of code and included nested if-else statements. The scores on this puzzle are listed in Table 2. The difference between the languages was again statistically significant for both the scores. Again, C++ scores were significantly higher than Java scores.

Table 2. Mean scores on the nested if-else puzzle no. 2 containing 36 lines of code.

Puzzle 2	N	Statement-level semantics \checkmark	Control-flow semantics \checkmark
C++	77	18.95 \pm 1.10	9.52 \pm 1.05
Java	167	15.86 \pm 0.70	8.25 \pm 0.67

If a student solved puzzle no. 1 with redundant actions, the tutor presented two more puzzles involving a single if-else statement. These follow-up puzzles nos. 3 and 4 were on the following programs:

- A program to read a number and print whether it is odd or even.

- A program to read sound in decibels, and print whether it is loud (if over 85 decibels) or tolerable.

Table 3 lists the mean scores of students on these follow-up puzzles nos. 3 and 4. The difference between C++ and Java was statistically significant on both the scores for puzzle no. 3, but on neither score for puzzle no. 4.

Table 3. Mean scores on the follow-up if-else puzzles nos. 3 and 4 (13 lines each).

Puzzle	Language	N	Statement-level semantics	Control-flow semantics
3 ✓	C++	31	8.07 ± 0.62	6.00 ± 0.78
	Java	134	7.25 ± 0.31	4.61 ± 0.39
4	C++	30	7.57 ± 0.59	4.97 ± 0.72
	Java	125	7.42 ± 0.30	4.65 ± 0.36

If a student solved puzzle no. 2 with redundant actions, the tutor presented two more puzzles on nested if-else statements. These follow-up puzzles nos. 5 and 6 were on the following programs:

- A program to read the month and print the corresponding season: Winter (January-March), Spring (April-June), Summer (July-September) and Fall otherwise).
- A program to read a location in degrees and print the corresponding quadrant: First (1–90), Second (91–180), Third (181–270) and Fourth otherwise.

Table 4 lists the mean scores of students on these follow-up puzzles nos. 5 and 6. The difference between the languages was statistically significant on statement-level semantics score for puzzle no. 5, and on both the scores for puzzle no. 6. Again, we found that the scores were significantly higher for C++ than Java.

Table 4. Mean scores on the follow-up nested if-else puzzles nos. 5 and 6 (27 lines each).

Puzzle	Language	N	Statement-level semantics	Control-flow semantics
5	C++	30	15.73 ± 1.11	8.83 ± 1.15
	Java	111	13.32 ± 0.58	7.59 ± 0.60
6 ✓	C++	25	16.36 ± 1.18	9.88 ± 1.31
	Java	85	14.38 ± 0.65	7.92 ± 0.72

Did the scores of students increase with practice? We considered only the students who had solved all three puzzles: puzzle no. 1 and the two follow-up puzzles nos. 3 and 4 that were similar to it. Since the first puzzle had 16 lines whereas puzzles 3 and 4 had only 13 lines, we normalized the scores as percentages of the number of lines in the puzzle,

as shown in Table 5. The change in score percentage from one puzzle to the next was not statistically significant except for statement-level semantics, which *decreased* from puzzle 1 to puzzle 3 for Java students. In all the other cases, solving multiple puzzles did not lead to significant change, let alone increase in the score percentage.

We had expected the scores to increase with practice due to practice effect. The tutor itself did not provide any feedback to help students increase their use of local or control-flow semantics while solving puzzles. What we found was that without such explicit support / feedback from the tutor, the puzzle-solving strategies of students did not change over the course of solving three similar puzzles.

Table 5. Normalized scores on the first three if-else puzzles nos. 1, 3 and 4 as percentages.

Language	Semantics	Puzzle 1	Puzzle 3	Puzzle 4
C++ (N = 30)	Statement-level	61.67	61.54	58.21
	Control-flow	39.38	45.90	38.21
Java (N = 120)	Statement-level	58.80	55.13 ✓	57.18
	Control-flow	33.13	35.83	35.96

5 Discussion

We analyzed the data of students solving six if-else puzzles in this study. The measure of statement-level semantics was 55–62% on the three if-else puzzles and 44–61% on the three nested if-else puzzles. This confirmed our hypothesis RQ1 that students used statement-level semantics when solving Parsons puzzles.

The measure of control-flow semantics was 33–46% on if-else puzzles and 23–37% on nested if-else puzzles. This confirmed our hypothesis RQ2 that students used control-flow semantics when solving at least part of the Parsons puzzles. Students used statement-level semantics more than control-flow semantics, and more on some puzzles than others.

We observed that whenever the difference between C++ and Java scores was significant, C++ scores were greater than Java scores. A simple explanation is that this was a comparison between two independent samples – C++ students never used Java puzzles and vice versa. So, the difference may be attributable to differences between the two groups of students, such as in their level of prior preparation. Since we did not collect students' course performance data, we could not verify this hypothesis.

Statement-level and control-flow semantics scores did not increase with practice. This refuted our third hypothesis RQ3. The tutors did not provide any feedback to promote the use of semantics. In the absence of such explicit support, the puzzle-solving strategy of students did not change, much less improve, in spite of the potential of practice effect.

In this study, we considered only complete and correct student solutions. We did not consider partial solutions, which may provide additional insights into the use of

semantics when solving Parsons puzzles. We used solution sequence representation that ignores intermediate deliberations of students. So, our analysis did not capture any trial-and-error puzzle-solving behaviors.

Acknowledgments. Partial support for this work was provided by the National Science Foundation under grants DUE-1432190 and DUE-1502564.

References

1. Denny, P., Luxton-Reilly, A., Simon, B.: Evaluating a new exam question: Parsons problems. In: Proceedings of the Fourth International Workshop on Computing Education Research (ICER 2008), pp. 113–124. ACM, New York, NY, USA (2008). <https://doi.org/10.1145/1404520.1404532>
2. Ericson, B.J., Foley, J.D., Rick, J.: Evaluating the efficiency and effectiveness of adaptive Parsons problems. In: Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER 2018), pp. 60–68. ACM, New York, NY, USA (2018). <https://doi.org/10.1145/3230977.3231000>
3. Ericson, B.J., Margulieux, L.E., Rick, J.: Solving Parsons problems versus fixing and writing code. In: Proceedings of the 17th Koli Calling International Conference on Computing Education Research (Koli Calling 2017), pp. 20–29. ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3141880.3141895>
4. Helminen, J., Ihantola, P., Karavirta, V., Malmi, L.: How do students solve Parsons programming problems?: an analysis of interaction traces. In: Proceedings of the Ninth Annual International Conference on International Computing Education Research (ICER 2012), pp. 119–126. ACM, New York, NY, USA (2012). <https://doi.org/10.1145/2361276.2361300>
5. Kumar, A.N.: Epplets: a tool for solving Parsons puzzles. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE 2018), pp. 527–532. ACM, New York, NY, USA (2018). <https://doi.org/10.1145/3159450.3159576>
6. Kumar, A.N.: Mnemonic variable names in Parsons puzzles. In: Proceedings of the ACM Conference on Global Computing Education (CompEd 2019), pp. 120–126. ACM, New York, NY, USA (2019). <https://doi.org/10.1145/3300115.3309509>
7. Kumar, A.N.: Helping students solve Parsons puzzles better. In: Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2019), pp. 65–70. ACM, New York, NY, USA (2019). <https://doi.org/10.1145/3304221.3319735>
8. Kumar, A.N.: The effect of providing motivational support in Parsons puzzle tutors. In: André, E., Baker, R., Hu, X., Rodrigo, M.M.T., du Boulay, B. (eds.) AIED 2017. LNCS (LNAI), vol. 10331, pp. 528–531. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61425-0_56
9. Morrison, B.B., Margulieux, L.E., Ericson, B., Guzdial, M.: Subgoals help students solve Parsons problems. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE 2016), pp. 42–47. ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2839509.2844617>
10. Parsons, D., Haden, P.: Parson’s programming puzzles: a fun and effective learning tool for first programming courses. In: Tolhurst D., Mann S. (eds.) Proceedings of the 8th Australasian Conference on Computing Education (ACE 2006), vol. 52, pp. 157–163. Australian Computer Society, Inc., Darlinghurst, Australia (2006)

Emotions and Affect



Tutorial Intervention's Affective Model Based on Learner's Error Identification in Intelligent Tutoring Systems

Soelaine Rodrigues Ascari^{1,2}(✉) , Andrey Ricardo Pimentel² ,
and Ernani Gottardo³ 

¹ Federal University of Technology – Paraná (UTFPR), Pato Branco, Brazil
`soelaine@utfpr.edu.br`

² Federal University of Paraná (UFPR), Curitiba, Brazil
`andrey@inf.ufpr.br`

³ Federal Institute of Education, Science and Technology of Rio Grande do Sul - IFRS, Erechim, Brazil
`ernani.gottardo@erechim.ifrs.edu.br`

Abstract. Intelligent Tutoring Systems (ITS) environments have the ability to adapt to each learner's individual needs and thus provide immediate and personalized instructions, both in content and in form. This personalization can consider several aspects, such as the interaction, the level of knowledge, the error, and the affective state of the learner aiming to improve the teaching strategies. One of the strategies is the possibility of presenting tutorial interventions when verifying an error made in solving an exercise, or when detecting that the learner is unmotivated or frustrated. These tutorial interventions can improve teaching methods in order to improve performance and the level of knowledge acquired by the learner. In this sense, this research presents a model that allows the automatic presentation of tutoring interventions based on identification of mathematical error kind committed by the learner, in addition to inferring his affective state. Experiments were carried out in a real learning environment, using the proposed model implemented in a fraction game. In general, the results presented indicate that personalized tutorial interventions favor greater engagement and motivation of learners and improvement in learning outcomes.

Keywords: Intelligent tutoring system · Tutorial intervention · Affective states

1 Introduction

ITS environments can provide individualized assistance, immediate and adapted instructions through data collection, observation of the learners' behavior and actions carried out in the system [26]. The goal is to assist learners and tutors in

the knowledge acquisition process. In the learning and cognitive processes, affective states play an important role, directly influencing aspects such as creativity, attention, decision making, social interaction, perception, and memorization [1, 21, 24]. Furthermore, affective states directly impact, positively or negatively, the learner's motivation and engagement [18, 22]. Learner's errors, whether due to lack of concepts knowledge or even inattention, should not be neglected or seen as something negative [15]. Identifying these errors can help the learner understand his errors making it possible to correct them in the future: that is, it becomes an opportunity for knowledge construction [8, 23]. The error in the mathematical field is considered a natural event, common in the learner's trajectory, regardless of age and/or performance level [14]. Models or theories about classification of mathematical errors are presented in studies such as [14, 19, 23, 25, 28].

Based on errors, ITS have the opportunity to individually identify the learner's difficulties and thus provide a more appropriate tutorial intervention [15]. These interventions help to prevent the learner from exploring paths that are not part of the instructional objectives [4]. When automating the tutorial intervention process, it is necessary to plan and define the time, the type and how the tutorial intervention's content will be presented to the learner. As for the intervention's timing, according to [27], the ITS can implement an external loop and an internal loop. The outer loop is performed once per task, while the inner loop is performed once for each step. The focus of this work is on internal loop interventions.

In this context, this article presents the MAFint - Tutorial Intervention Affective Model to ITS. This model allows the automatic presentation of tutorial interventions from the identification of the type of learner's mathematical error, in addition to inferring his affective state. The classification of mathematical errors developed by [14] is applied to identify the error. The inference of the affective state, performed by capturing the learner's facial expression, uses the representation model of emotions in quadrants developed by [11]. MAFint was implemented in a fraction game designed for this purpose, and experiments were carried out in a real learning environment. The results indicate that the personalized tutorial interventions favor greater engagement and motivation and contribute to an improvement in learning.

2 Tutorial Intervention Affective Model - MAFint

MAFint has three main processes: identifying the type of error, the inference of the affective state, and the tutorial intervention classification. Figure 1 presents the tutorial intervention model that generates interventions based on the identification of learner's error type.

The learner answers the exercise (1st) and the model identifies whether the answer presented is correct or not (2nd). If the answer is correct, the environment loads a new exercise. Otherwise, the environment performs an analysis to identify the learner's error (3a) and simultaneously captures the learner's facial image (3b). Immediately after detecting the type of error (3a) and inferring the affective

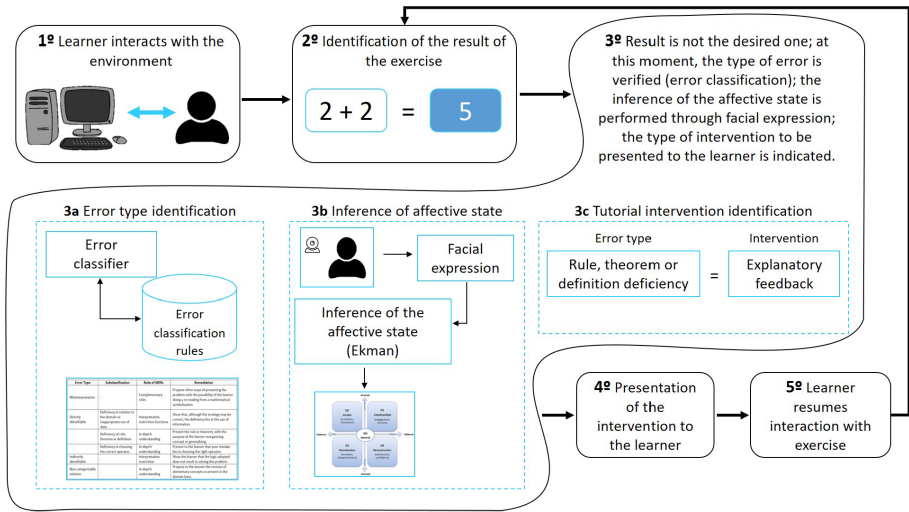


Fig. 1. Tutorial Intervention Affective Model - MAFint. Source: [3]

state (3b), a tutorial intervention is classified (3c) and then displayed to the learner (4th) in the form of a text in natural language, intending to provide help in solving the exercise in question. After receiving the tutorial intervention, the learner can solve the exercise again or end the use of the environment (5th). If it succeeds, the environment loads a new exercise, otherwise, it returns to the third step.

2.1 Error Classification

For the process of identifying the type of error (3a), the classification of mathematical errors developed by [14] was used. At this stage, the possible errors present in certain contents need to be previously identified, with specialists' help, to be subsequently organized into a rule base. Based on these rules, the type of error made by the learner is classified. The types and subtypes of errors contained in the classification of [14] are: a) Misinterpretation: this type of error would alert the learner's difficulty in advancing the understanding of the problem structure; b) Directly identifiable: this type is subclassified in deficiency errors in relation to the domain or inappropriate use of data, deficiency errors in the rule, theorem or definition and deficiency errors in the choice of the correct operator; c) Indirectly identifiable: this type contemplates the error presented by the lack of correct logic; and d) Non-categorizable solution: the error that is not included in any of the aforementioned types will be included in this classification.

2.2 Inference of Affective State

The inference of the affective state (3b) is based on the learner's facial expression captured by a standard camera according to the basic emotions of the Ekman

model [7]. The inference model applied in MAFint uses the representation of emotions in quadrants developed by [11]. The quadrants are formed by the Valence (horizontal axis) and Arousal (vertical axis) dimensions that were named Q1, Q2, Q3, Q4, in addition to a Neutral state (QN - Neutral Quadrant), as shown in Fig. 2.

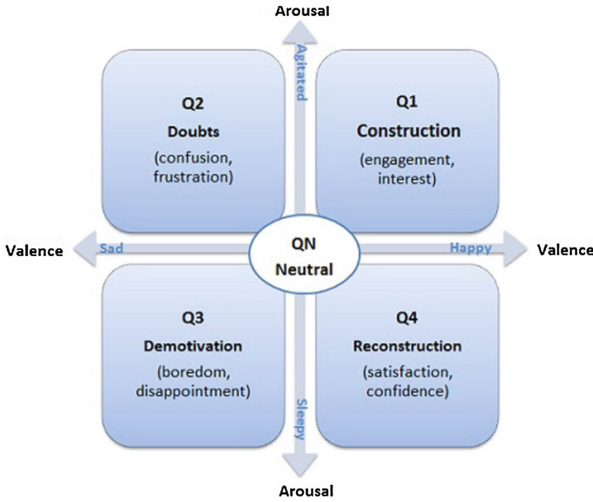


Fig. 2. Model of representation of emotions in quadrants. Source: [10]

The inference of emotions returns scores for each of the emotions considered in the model and then maps the scores of the emotions to the respective quadrants (Q1: joy and surprise; Q2: fear, disgust, anger, surprise, and contempt; Q3: sadness; Q4: joy; QN: neutral). The Q1 and Q2 quadrants with positive activation form the class called agitation, and the Q3 and Q4 quadrants with negative activation form the class drowsiness. These quadrants combine a set of emotions that can potentially impact the learning process [11].

3 Tutorial Interventions

For the development of MAFint, initially, a study on the types of tutorial interventions applied to ITS was carried out. From the works of [6, 9, 12, 13, 16, 17, 20] a classification of types of tutorial interventions was organized. Table 1 presents the types and their respective subtypes of interventions.

The completion of this step allowed viewing and organizing the tutorial interventions based on their characteristics in types and subtypes with the indication of their application. The presentation of this classification aims to assist in choosing the tutorial interventions to be applied in ITS. The definition of the type and subtype of the tutorial intervention displayed to the learner must consider

Table 1. Classification of types and subtypes of tutorial interventions.

Type	Subtype	When to use
Scaffolding [12] and [17]	Conceptual	To guide the learner to focus on central points and concepts that can present various interpretations.
	Procedural	To assist the learner in how to return to a particular location or how to use a particular resource offered by the system.
	Strategic	When it is desired to enable the learner to participate in planning and decision making.
	Metacognitive	To recommend the learner to plan ahead, evaluate their development and progress, and determine needs.
Feedback Based on function [9]	Confirmation	To indicate whether an answer is correct or incorrect.
	Corrective	To complement the incorrect answer presented by the confirmation feedback.
	Explanatory	To present relevant information that helps to identify why an answer is incorrect.
	Diagnosis	To highlight the error and indicate a solution.
	Elaborative	To provide information that can complement or expand the knowledge being evaluated.
	Knowledge of the answer	To indicate whether an answer is correct or incorrect with minimal information.
Feedback - Based on content [6], [16] and [20]	Answer until correct	To inform that the answer is still not correct and that the learner try again.
	Contingent topic	When desired, when verifying the learner's response, develop general feedback on the topic.
	Contingent response	To explain why the correct answer is correct and the incorrect answer is incorrect.
	Related bug	To present the common errors made by the learner.
	Attribute isolation	To present the main characteristics of a concept.
	Performance knowledge	Feedback can be the number of errors made or the percentage of tasks solved correctly.
	Knowledge of the correct answer	To inform the correct answer.
	Elaborate	To provide additional information (suggestions, tips, explanations, guidance questions).
	Condition violation	To guide the learner on the steps that must be followed.
	Goal or objective	To assist the learner in finding the correct answer.
Hints [13]	Combined	When you want to combine the types of condition violation feedback and goal feedback.
	Conveyed Information	When you want to ask the learner to infer or record an answer, or even the next step in a solution.
	Pointing To	Point to the location of the information in a knowledge base, for example, but without transmitting the information.
	Directed Line of Reasoning	When you want to make the learner have to "think" about each of the steps until reaching a solution.

the instructional objectives and need to be adapted to the content that will be worked. Thus, the participation of the specialist in the subject in question becomes relevant.

4 Experiments

4.1 Methodology

For the purpose of checking the feasibility and the results obtained by MAFint, experiments were carried out with learners from the fifth year of elementary school in a real learning environment using a game of mathematical fractions. The experiments also evaluated the hypothesis that the presentation of tutorial interventions considering the learner’s error contributes to the resolution of exercises, to improve their performance and provide greater engagement and motivation. There are three experiments: Pilot, Experiment 1 (E1), and Experiment 2 (E2). The Pilot experiment aimed to test the game and the entire framework, making it possible to analyze, review, and improve the research instruments and procedures. The experiment (E1) had the purpose of collecting data during the interaction of the learners with the game, such as errors, affective states, and tutorial interventions, verifying which interventions most helped them. And the

last experiment (E2) was applied to assess whether there was an improvement in learning outcomes using MAFint. In this experiment (E2), the learners also interacted with the game but were divided into two groups, Control Group (CG) and Experimental Group (EG). This experiment's objective is to verify whether there was a significant difference between the two groups concerning to learning outcomes.

4.2 Fraction Game

A game about fractions was developed to carry out the experiments. The game consists of the presentation of 22 fractions operations that considers the four basic operations: addition, subtraction, multiplication, and division, divided into four levels of difficulty. These operations were evaluated, adjusted and approved by the mathematics teachers of the classes involved with the laboratory experiments. An avatar was created and used to guide and present the interventions to the learners during the game. The game was developed in Python using the Django framework with the SQLite database. For facial expression recognition, the API Microsoft Azures Cognitive Services Emotion¹, a free tool, was used.

The game implemented the types and subtypes of errors according to these identified in a list of printed mathematical fractions exercises performed by the learners. With the help of a mathematics teacher, it was possible to identify and classify the occurrence of the following types and subtypes of errors, considering the classification of [14]: 1) Directly identifiable - errors of rule deficiency, theorem and definition and deficiency errors in choosing the correct operator; 2) Indirectly identifiable; and 3) Non-categorizable solution.

From the set of 24 intervention subtypes presented in Table 1, six were selected and implemented in the game to be presented to learners whenever an answer was not correct. The Feedback type has implemented the Explanatory, Diagnostic and Goal or Objective subtypes. The Hints subtypes were Conveyed Information, Pointing to, and Directed line of reasoning. The subtypes Knowledge of the correct answer, Corrective and Contingent response were also implemented, which are presented after exhausting all attempts by the learner to answer a fraction operation in the game. The displayed intervention is of the Feedback type with the indication of the correct answer and the error made or how it could have solved the question. The choice of subtypes took into account the content (operations on fractions), how it is presented, and the suitability to the game model.

4.3 Experiments

Seventy four learners participated in the experiments. They come from the fifth year of elementary school in two county schools, 27 females and 47 males, aged between 10 and 12 years. Participated in E1 34 learners. The E2 was divided into three stages, named pre-test, game, and post-test. Forty learners from two

¹ <https://azure.microsoft.com/pt-br/services/cognitive-services/emotion/>.

classes participated in these stages, divided into two homogeneous groups, Control Group (CG) and Experimental Group (EG).

Experiment E1. In this experiment, the learners interacted with the fraction game. The Table 2 shows an example of an error made by the learner, indicating the type/subtype of error, type/subtype of intervention, as well as the text of the intervention tutorial presented to the learner.

Table 2. Operation with error type and subtype and tutorial intervention. Source: [2]

Operation	Incorrect answer	Error type	Error subtype	Type of intervention	Intervention subtype
$\frac{3}{5} + \frac{1}{5} =$	$\frac{4}{10}$	Directly identifiable	Rule, theorem or definition deficiency	Feedback	Diagnosis
Intervention text: Incorrect answer! The error is in the denominator value. When adding fractions with equal denominators, just repeat the denominator in the result.					

With each new attempt by the learner to answer an operation, the game made a random choice among all interventions that fit the rules. These rules refer to the set of information collected from the question itself: Operation, Same or different Denominators, and comparing the Numerators and Denominators of the selected answer with these of the correct answer. Through this experiment, the learners errors, the specific interventions presented, and the affective states based on their facial expressions were collected. By analyzing these data, it was possible to observe and identify which tutorial interventions, associated with the types of errors, helped the learners solve the game’s fractions and monitor their changes in affective states. Thus, the game was adjusted for E2.

Experiment E2. The experiment was divided into three stages: pre-test, interaction with the game of fractions, and post-test. The first stage, the pre-test, aimed to create two homogeneous groups (CG and EG) for the stage that uses the fraction game and then for the post-test. For this, from the notes of a list of fractions exercises printed, the learners were ordered and then divided into two groups using a random sampling process. Then, the homogeneity of the groups was verified in the mean (t-test) and the variance (f test). Both tests indicated homogeneity, with a significance level of $\alpha = 0.05$, $t = 0.2550$, and $f = 1.0543$ (variance). With these results, each group’s learners were defined (CG and EG) in each class.

After forming the CG and the EG, the learners participated in the interaction phase with the fraction game. This step was carried out in the school’s computer lab. For CG learners, the game features minimal tutorial interventions with two

categories, correct and incorrect. That is, the game only shows whether the learner got the operation right or wrong. It presents specific tutorial interventions for EG learners, such as Tips or Feedback, considering the model developed. Thus, when the learner selects a wrong answer in EG, the type of error is checked, the affective state is inferred through facial expression, and then the tutorial intervention is performed. As the game features different tutorial interventions for each group, opted to work with the CG and EG learners separately. While one group was in the computer lab, the other remained in the classroom with the teacher doing other activities. Thus, two sessions were held with each class, with an average time of 40 min.

In the last step, the post-test, the learners answered a new list of fractions exercises printed, and just as in the pre-test, the teacher applied, corrected, and assigned a grade. It is important to highlight that the post-tests objective was to compare the pre-test results with these of the post-test to verify whether the EG's specific tutorial interventions helped them obtain a better result in relation to the CG learners.

Two hypotheses were formulated for the experiment. The first hypothesis verifies whether the EG learner's post-test note's average was significantly, at a level of $\alpha = 0,05$, higher than the average of the post-test note of the CG. The second hypothesis formulated verifies whether the application of specific interventions associated with this work provides greater engagement and motivation compared to minimal interventions.

5 Results

5.1 Hypothesis 1

For the first hypothesis, average of the post-test notes, the first step was performed using the Shapiro-Wilk test to verify the normality of the data (notes), which indicated $p\text{-value} = 0,0046$ less than 5% (significance level adopted in the research $\alpha = 0,05$), that is, as $p\text{-value} < 0,05$, then the data normality is confirmed. A t-test was then applied for independent groups, whose results demonstrated, with a 95% degree of confidence, with $p\text{-value} = 0.0348$ and $t = 1.8674$, that there is a significant difference ($\alpha = 0,05$) in terms of the average of the grades between the CG learners and the EG. Thus, like $p < \alpha$, there is evidence that this difference is due to the specific tutorial interventions presented to the EG that helped learners achieve a better learning outcome.

From the grades obtained by the two groups' learners in the pre-test and post-test, it was possible to observe that the CG post-test mean higher than the average of the pre-test. However, this difference is not statistically significant ($p > \alpha$), with a confidence level of 95% ($\alpha = 0,05$), $p\text{-value} = 0,0660$ and $t = 4,0652$. Thus, there is an indication that minimal interventions have not helped learners to acquire concepts. It is also observed that the mean of the EG in the post-test was higher than the average in the pre-test. A significant difference ($p < \alpha$), with a confidence level of 95% ($\alpha = 0,05$), $p\text{-value} = 0,0006$ and $t = 6,2084$. Thus, it can be seen that there is evidence that there was a greater

performance of the learners of the SG in relation to the CG. The paired t-test for dependent samples was used to analyze these data.

5.2 Hypothesis 2

Some information will be presented to answer the second hypothesis. A total of 1361 occurrences of affective states were recorded during the game. Of these, 1244 (92,1%) are identified occurrences. It was possible to infer the learner’s emotion through facial expression and map in the corresponding quadrant, according to the approach of representing emotions by quadrant. The number of unidentified occurrences was 117 (8,6%). The main reasons for not identifying the face were the learner’s position in relation to the camera, the hand in front of the mouth or face. The average number of affective states per learner was 31,1, with a standard deviation of 6,4. The quadrant with the highest number of occurrences of affective states was QN, in both groups, with 94,4% for the CG and 84,8% for the EG. The predominance of this affective state was expected for this activity and is in lined up with the results of other works in this area [5,11]. In the sequence, the Q1 quadrant presented 2,5% of occurrences for the CG and 9,7% for the EG. This means that EG learners obtained more occurrences in the Q1 quadrant in relation to the CG. This quadrant identifies affective states’ occurrence such as joy, engagement, and motivation, both of which are desired to improve the experience and learning results. The Q2 quadrant registered 1,6% for the CG and 2,8% for the EG, and the Q4 quadrant registered 1,6% for the CG and 2,6% for the EG. No occurrences were recorded for Q3.

Table 3 shows the number of occurrences of interventions for each group. These interventions were presented to EG’s learners whenever the response selected for an operation was incorrect. The EG obtained a percentage value of 35,5% of correct answers using specific interventions in relation to the total occurrences of the group and 64,5% of errors. The CG obtained 24,2% of correct answers using minimal interventions and 75,8% errors in relation to the total number of occurrences. It is observed that the learners of the EG had more success than the learners of the CG. This indicates that the interventions received by EG’s learners helped to solve the fractions operations of the game in relation to the other group that did not receive the same interventions.

Table 3. Total interventions - E2. Source: [3]

	Number of occurrences of interventions	Number of hits	Number of errors	% of hits per group	% of errors per group
CG	264	64	200	24,2%	75,8%
EG	217	77	140	35,5%	64,5%
Total	481	141	340		
Average	241 (σ=33,2)	71 (σ=9,2)	170 (σ=42,4)		

Table 4 shows the quadrant changes by type of intervention in each group. It is also possible to observe that EG’s specific interventions obtained a higher percentage (36,7%) in relation to the minimum intervention of the CG (11,4%). This information indicates that specific interventions, such as Hints and Feedback, presented to the EG in this context helped the learners remain more motivated and engaged than the CG learners who received minimal interventions.

Table 4. Change of quadrant x intervention. Source: [3]

	CG		EG	
	Minimal intervention	% of total changes	Specific intervention	% of total changes
Changed state to Q1	5	11,4%	18	36,7%
Changed state to Q2	3	6,8%	5	10,2%
Changed state to Q3	0	0,0%	0	0,0%
Changed state to Q4	6	13,6%	4	8,2%
Changed state to QN	30	68,2%	22	44,9%
Total	44	100,0%	49	100,0%

The most significant change was from the Q1, Q2, or Q4 quadrants to the QN quadrant. The specific interventions of the type “Hints” and “Feedback” of the EG generated 18 changes from some quadrant to Q1. In addition to maintaining 12 times the quadrant Q1 after the intervention. There were six occurrences of changes in different quadrants to Q1 from interventions of the type “Feedback” and nine occurrences of interventions where the learner remained in Q1. In EG, there were 12 occurrences of changes in different quadrants for Q1 from interventions of the type “Hints” and three occurrences of interventions in which the learner remained in Q1. As for the CG, there were only five changes from one quadrant to Q1 from a minimal intervention (right/wrong), and there was no occurrence in which the Q1 quadrant remained. From this result, it is possible to conclude that the second hypothesis was met.

6 Conclusions and Future Work

In this article, an experiment was presented with learners from the fifth year of elementary school applying a game of mathematical fractions, in which, from the identification of the error type in the fractions operation, a specific tutorial intervention was presented to the learner: “Hints” or “Feedback.” The affective state was inferred for each response of the learner to verify if these interventions contribute to a greater engagement in relation to a minimum intervention.

From the results presented, it is possible to observe that the EG learners obtained a higher percentage of correct answers in relation to the CG learners. This indicates that the interventions presented to EG’s learners helped solve the fractions operations of the game in relation to the CG that received minimal

interventions. The results also indicate that specific interventions contributed to a more significant occurrence of change and permanence of learners in quadrant Q1 represented by emotions such as joy, motivation, interest, and engagement, which favor and have the potential to impact the learning process positively. The results also indicate an improvement in learning outcomes that may be due to a better motivational state.

An indication of future work is to map the game's events to allow the inference of the learner's cognitive affective states. Another recommendation is to integrate to MAFint that the affective state inferred by the representation of quadrants should be considered for the tutorial intervention's indication. In this way, the intervention would be presented based on the error and the learner's affective state.

References

1. Arguedas, M., Xhafa, F., Casillas, L., Daradoumis, T., Peña, A., Caballé, S.: A model for providing emotion awareness and feedback using fuzzy logic in online learning. *Soft. Comput.* **22**(3), 963–977 (2016). <https://doi.org/10.1007/s00500-016-2399-0>
2. Ascari, S.R., Gottardo, E., Pimentel, A.R.: Identificação de Intervenções Tutoriais para Ambientes Virtuais de Aprendizagem. In: Brazilian Symposium on Computers in Education, pp. 842–851. *Anais do XXXI Simpósio Brasileiro de Informática na Educação - SBIE* (2020). <https://doi.org/10.5753/cbie.sbie.2020.842>
3. Ascari, S.R., Gottardo, E., Pimentel, A.R.: MAFint: modelo afetivo de intervenção tutorial para Ambientes de Virtuais de Aprendizagem. In: Brazilian Symposium on Computers in Education, pp. 832–841. *Anais do XXXI Simpósio Brasileiro de Informática na Educação - SBIE* (2020). <https://doi.org/10.5753/cbie.sbie.2020.832>
4. Burns, H.L., Capps, C.G.: Foundations of intelligent tutoring systems: an introduction. In: Polson, M.C., Richardson, J.J. (eds.) *Foundations of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates, Hillsdale (1988)
5. D'Mello, S., Picard, R.W., Graesser, A.: Toward an affect-sensitive autotutor. In: *IEEE Intelligent Systems*, vol. 22, pp. 53–61. IEEE (2007). <https://doi.org/10.1109/MIS.2007.79>
6. Economides, A.A.: Adaptive feedback evaluation. In: *Proceedings of the 5th WSEAS international Conference on Distance Learning and Web Engineering*, pp. 134–139 (2005)
7. Ekman, P.: An argument for basic emotions. *Cogn. Emot.* **6**, 169–200 (1992). <https://doi.org/10.1080/02699939208411068>
8. Fiori, C., Zuccheri, L.: An experimental research on error patterns in written subtraction. *Educ. Stud. Math.* **60**, 323–331 (2005). <https://doi.org/10.1007/s10649-005-7530-6>
9. Fleming, M.L., Levi, W.H.: *Instructional message design: principles from the behavioral and cognitive sciences*. In: Educational Technology Publications, Englewood Cliffs, NJ (1993)
10. Gottardo, E., Ricardo Pimentel, A.: Improving inference of learning related emotion by combining cognitive and physical information. In: Nkambou, R., Azevedo, R., Vassileva, J. (eds.) *ITS 2018. LNCS*, vol. 10858, pp. 313–318. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91464-0_33

11. Gottardo, E., Pimentel, A.R.: Reconhecimento e adaptação à dinâmica de estados afetivos relacionados à aprendizagem. In: Brazilian Computer Society - SBC, vol. 29, pp. 1223–1232. Anais do XXIX Simpósio Brasileiro de Informática na Educação (2018). <https://doi.org/10.5753/cbie.sbie.2018.1223>
12. Hannafin, M., Land, S., Oliver, K.: Open learning environments: foundations, methods, and models. In: Instructional-Design Theories and Models: A New Paradigm of Instructional Theory, vol. 2, pp. 115–140 (1999)
13. Hume, G., Michael, J., Rovick, A., Evens, M.: Hinting as a tactic in one-on-one tutoring. *J. Learn. Sci.* **5**, 23–47 (1996). https://doi.org/10.1207/s15327809jls0501_2
14. Leite, M.D., Pimentel, A.R., Pietruchinski, M.H.: Remediação de erros baseada em múltiplas representações externas e classificação de erros aplicada a objetos de aprendizagem inteligentes. In: Brazilian Symposium on Computers in Education - SBIE, vol. 23, pp. 1–10. Anais do 23º Simpósio Brasileiro de Informática na Educação (2012). <https://doi.org/10.5753/cbie.sbie.2012.25p>
15. Marczal, D., Direne, A., Pimentel, A.R., Krynski, E.M.: Farma: Uma Ferramenta de Autoria para Objetos de Aprendizagem de Conceitos Matemáticos. In: Brazilian Computer Society - SBC, vol. 4, pp. 23–32. Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação - CBIE (2015). <https://doi.org/10.5753/cbie.wcbie.2015.23>
16. McKendree, J.: Effective feedback content for tutoring complex skills. *Hum. Comput. Interact.* **5**, 381–413 (1990). https://doi.org/10.1207/s15327051hci0504_2
17. McLoughlin, C.: Achieving excellence in teaching through scaffolding learner competence. In: Seeking Educational Excellence (2004)
18. Morais, F., da Silva, J., Reis, H., Isotani, S., Jaques, P.: Computação Afetiva aplicada à Educação: uma revisão sistemática das pesquisas publicadas no Brasil. In: Brazilian Computer Society - SBC. Brazilian Symposium on Computers in Education - SBIE, vol. 28, pp. 163–172. (2017). <https://doi.org/10.5753/cbie.sbie.2017.163>
19. Movshovitz-Hadar, N., Zaslavsky, O., Inbar, S.: An empirical classification model for errors in high school mathematics. *J. Res. Math. Educ.* **18**, 3–14 (1987). National Council of Teachers of Mathematics. <https://doi.org/10.2307/749532>
20. Narciss, S.: Designing and evaluating tutoring feedback strategies for digital learning environments on the basis of the interactive tutoring feedback model. In: Digital Education Review, vol. 23, pp. 7–26. Digital Education Observatory (OED) (2013). <https://www.learntechlib.org/p/131614>
21. Pekrun, R.: Emotions as drivers of learning and cognitive development. In: Calvo, R., D’Mello, S. (eds.) *New Perspectives on Affect and Learning Technologies. Explorations in the Learning Sciences, Instructional Systems and Performance Technologies*, vol. 3. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-9625-1_3
22. Pekrun, R., Goetz, T., Titz, W., Perry, R.P.: Positive emotions in education. In: Frydenberg, E. (ed.) *Beyond Coping: Meeting Goals, Visions, and Challenges*, pp. 149–173. Oxford University Press (2002). <https://nbn-resolving.org/html/urn:nbn:de:bsz:352--139080>
23. Peng, A., Luo, Z.: A framework for examining mathematics teacher knowledge as used in error analysis. *Learn. Math.* **29**, 22–25 (2009)
24. Picard, R.: *Affective Computing*. MIT Press, Cambridge (1997)
25. Radatz, H.: Error analysis in mathematics education. *J. Res. Math. Educ.* **10**, 163–172 (1979). <https://doi.org/10.2307/748804>

26. dos Santos, D.C.V., Falcão, T.P.: Acompanhamento de alunos em ambientes virtuais de aprendizagem baseado em sistemas tutores inteligentes. In: Brazilian Symposium on Computers in Education - SBIE, vol. 28, pp. 1267–1276 (2017). <https://doi.org/10.5753/cbie.sbie.2017.1267>
27. Vanlehn, K.: The behavior of tutoring systems. *Int. J. Artif. Intell. Educ.* **16**, 227–265 (2006)
28. Vergnaud, G.: A classification of cognitive tasks and operations of thought involved in addition and subtraction problems. In: *Addition and Subtraction: A Cognitive Perspective*, pp. 39–59. Lawrence Erlbaum Associate (1982)



A Recommender System Based on Effort: Towards Minimising Negative Affects and Maximising Achievement in CS1 Learning

Filipe D. Pereira¹(✉), Hermino B. F. Junior¹, Luiz Rodriguez⁴,
Armando Toda⁴, Elaine H. T. Oliveira², Alexandra I. Cristea³,
David B. F. Oliveira², Leandro S. G. Carvalho², Samuel C. Fonseca²,
Ahmed Alamri³, and Seiji Isotani⁴

¹ Department of Computer Science, Federal University of Roraima, Boa Vista, Brazil
filipe.dwan@ufrr.br

² Institute of Computing, Federal University of Amazonas, Manaus, Brazil

³ Department of Computer Science, Durham University, Durham, UK

⁴ ICMC, University of Sao Paulo, Sao Carlos, Brazil

Abstract. Programming online judges (POJs) are autograders that have been increasingly used in introductory programming courses (also known as CS1) since these systems provide instantaneous and accurate feedback for learners' codes solutions and reduce instructors' workload in evaluating the assignments. Nonetheless, learners typically struggle to find problems in POJs that are adequate for their programming skills. A potential reason is that POJs present problems with varied categories and difficulty levels, which may cause a cognitive overload, due to the large amount of information (and choice) presented to the student. Thus, students can often feel less capable, which may result in undesirable affective states, such as frustration and demotivation, decreasing their performance and potentially leading to increasing dropout rates. Recently, new research emerged on systems to recommend problems in POJs; however, the data collection for these approaches was not fine-grained; importantly, they did not take into consideration the students' previous effort and achievement. Thus, this study proposes for the first time a prescriptive analytics solution for students' programming behaviour by constructing and evaluating an automatic recommender module based on students' effort, to personalise the problems presented to the learner in POJs. The aim is to improve the learners achievement, whilst minimising negative affective states in CS1 courses. Results in a within-subject double-blind controlled experiment showed that our method significantly improved positive affective states, whilst minimising the negatives ones. Moreover, our recommender significantly increased students' achievement (correct solutions) and reduced dropout and failure in problem-solving.

Keywords: Online judge · Data-driven analysis · Recommender System

1 Introduction

Programming Online Judges (POJs) are automatic code correction environments that are typically used by students to improve their programming skills and/or train for programming competitions [26, 40, 41, 46, 47]. The adoption of these environments by instructors and institutions has increased in the last few years in introductory computing (so-called 'CS1') classes [36, 41]. Typically, in educational scenarios, students code in an integrated development environment (IDE) tied to a POJ [7, 36]. Students then design their algorithms in the IDE and submit them to be evaluated by the POJ system, which provides them real-time feedback based on a case test analysis [7, 34, 39, 41, 44].

Alongside the growing popularity of POJs, data within these systems are gaining attention [6, 7, 13, 14, 28, 29, 31–33, 39, 41, 46]. Despite the notorious benefits of POJs in education, these systems are not able to recommend the appropriate problems for the students, which may impact on affective perception, leading, over time, to affective states such as frustration [8, 24, 38, 44, 45]. Frustration has been shown to be directly related to the amount of effort a student needs to spend to solve a problem and may even lead to dropout [1, 22, 25, 30, 38]. This happens due to effort being intrinsically related to the students' confidence, competence and consequently affecting their motivation [19]. According to [5], the learners' effort can be measured by the amount of energy and time they expend to meet the academic requirements. [19] explain it is necessary to measure effort to assess students' motivation and satisfaction. Moreover, a good balance of effort required to solve tasks is related to increased achievement [12].

To adapt the programming problems to the students' effort, Recommender Systems (RS) appear as a viable solution [39]. RS are environments used to identify and provide content based on rules designed from user data. These systems have been widely used in educational scenarios [2, 39, 44]; however, few studies have tackled ways to provide recommendations based on a deep analysis of user behaviours. Specifically in the scope of programming learning, there are only a few studies available in the literature proposing methods to recommend problems in POJs, and such studies typically make the recommendations only based on students' attempts and results from the submissions to the POJ [39, 44]. Notice that a deep behavioural analysis of fine grained data is crucial to make appropriate recommendations [20].

As such, in this work, besides the variables previously used in the literature (attempts and results from submissions), we also track how students solve problems in the embedded IDE of a POJ and construct a holistic set of fine-grained features to represent the effort expected to solve a given problem. Using these features, we make a recommendation based on the following hypothesis: if a student s solves a given problem p (which we call a target problem), our method recommends a problem p' that requires an effort to be solved similar to that of p , assuming the student s would be able to solve the problem p' . Through exploring this hypothesis, we believe that the recommendations will minimise students' negative affective states, whilst maximising the positive ones, as the problems recommended will not require a disproportionate effort from the learners.

In addition, as aforementioned, effort has been related to students' achievement in other fields beyond POJ [12,34]. Hence, our second hypothesis is that our recommendation based on expected effort will increase the student achievement and decrease dropout and failure rate in problem-solving. Thus, this work aims at solving the following research question: *Does personalised recommendation based on effort influence the students' affect and achievement in online judges?*

2 Related Work

Programming is learned by doing, that is, students need to solve many problems to improve their skills and a POJ is a suitable tool for practising [34,41]. However, given the huge amount of problems available in this system, frustration, confusion and other negative affective states might be triggered when learners are searching for problems or solving inadequate questions [39,44]. Thus, in this section, we analyse studies that propose methods for the automatic recommendation of problems or pedagogical material in automatic assessment systems.

In this sense, [9] conducted a study mapping code submitted by a group of students to create a code profile that provided personalised instructions, which was based on previous recommendations made by humans. Through these instructions, the authors suggested that it was possible to reduce difficulties faced by students. They used a multi-label k-nearest neighbour technique to recommend a topic of programming for the student. For instance, after solving a given problem using the "if-then-else" structure, the recommender could suggest problems using "loops". However, this recommender presents a generic list of problems based on the topic. In this sense, students were still responsible for finding problems that they thought were closest to their programming skills. Here, using our hypothesis, our novel behavioural-based model recommends directly the problem to the student, not the programming topic. Moreover, POJ questions are typically not annotated with the topic and, hence, a human endeavour would be needed to apply manual annotations of topics [15].

Following, [8,17] proposed to extract information from codes to choose the suggestions for students learning programming. [17] created an RS that provided hints during the solving process, using techniques such as *term frequency-inverse document frequency* to represent similarities. [8] created an RS that suggested learning materials to help teachers in designing courses. Different from [17] and [8] who focused on tips about learning materials, we used data-driven behaviour analysis to infer the knowledge and effort of the students based on the data logs generated through real-time execution.

[44] and [39] used learner behavioural data for automatic recommendation of problems in POJs. [44] proposed an RS that recommends problems based on a collaborative approach. They used a binary matrix as a basis for the recommender. [39] proposed a learning path recommendation system based on learner's submission history in an online judge. However, these authors [39,44] considered only the number of attempts and results from the submissions as features. In this work, we extend the features that were used in previous works and others

that were extracted from the codes submitted and fine-grained log data, including self-devised features. Moreover, different from all of them, we evaluated our method with real learners and checked their affective states and achievement rates when solving the problems.

In brief, none of the previous studies performed an analysis of effort considering such fine-grained set of features to design a behavioural recommender model, as a way to personalise the recommendations in POJs. Moreover, for the first time, to the best of our knowledge, our RS provides problems adapted to the students' skills, and we measured the resulting achievement rate and affective states when solving our recommended problems.

3 Materials and Methods

In this section, we present the methods and tools used in this study, describe the data collection process, feature extraction and architecture of the RS, as well as the evaluation method for this study. Following, we present the instrument and data collection process:

- **Instrument:** We used a POJ called CodeBench¹, a home-made POJ system designed by the Institute of Computing team from Federal University of Amazonas, Brazil. Codebench allows teachers, instructors and lecturers to provide assignments for students to develop their programming skills. Moreover, students can perform self-direct learning. Once an answer for the problem is submitted, the system provides a real-time feedback. This system also includes an online IDE, where students can write and execute their codes. Some other features are management of classes, social interactions between students and lecturers, and learning materials sharing. All these characteristics are common in other POJs such as URI online judge [4], UVA online judge [37], and others.
- **Data collection:** We collected data from CS1 courses that were offered during 6 semesters (from 2016 to 2018). Students had to solve seven sets of exercises using Python. Each set of exercises is related to one of the following topics: (1) variables; (2) conditionals; (3) nested conditionals; (4) while loops; (5) vectors; (6) for loops; and (7) matrices. Furthermore, learners could solve other problems as wanted (self-direct learning). All the students solved the problems using the IDE provided in the POJ CodeBench. We collected data from 2,058 students that generated 535,619 code submissions to solve these exercises. Students' keystrokes were recorded in their logs on the server side. These log files are the sources of our extracted fine-grained features related to the students' behaviour to be used as input into our data-driven approach. With the data from the student's logs, we will represent the expected effort for the student to solve the programming problems.

¹ codebench.icomp.ufam.edu.br.

3.1 Behaviour-Based Recommender System Based on Effort

Students' effort is the amount of energy and time learners expend to meet the academic requirements [5]. In this work, we represent the student effort expected to solve a given problem as the aggregation of students' procedural and intellectual effort [5] combined with consolidated code metrics [27]. The procedural effort is related to how students succeeded (e.g. proportion of solved questions) in the assignment and the intellectual effort is related to how much energy the students used to solve the problems (e.g. number of attempts, time spent). The code metrics (e.g. number of loops) come from the software engineering field and are used to measure the learners' effort in building their code solutions.

Notice that effort is a psychological construct and, therefore, there is no standard way to measure it. In other words, there is no standardised scale that will measure how students efforts are used to solve problems [16]. In these cases, there is a need of features that may be used to indirectly measure one's effort. In addition, we need to define some observable, recordable measures that reflect the construct, called the *operational definition of the construct*. As such, we have established an operational definition to compute the expected effort to solve a given problem, using features that have already proved to be efficient in the literature [7, 18, 27, 34, 42, 44] and code metrics established in software engineering to measure the effort of programmers in the development process.

We extracted these features from the students' logs and codes and further processed them, e.g., extracting the average number of students' attempts for each problem, average number of lines of codes for each problem, etc. Thus, our observable, recordable measure to represent the effort is represented via the following features: *noAttempts* - proportion of users who didn't try to solve a given programming problem [34, 43]; *unsucNoRes* - proportion of users who failed to solve a given programming problem with a few attempts [34, 43]; *unsucRes* - proportion of users who tried hard, but failed to solve the problem [34, 43]; *sucNoRes* - proportion of users who solved the problem with a few attempts [34, 43]; *sucRes* - proportion of users who tried hard and managed to solve a given problem [34]; *attempts* - average of students' attempts to solve a given problem; *IDEUsage* - average resolution time of student on the Online IDE to solve a given problem [13, 34]; *contCicle* - average number of loops from submitted students' codes for a given problem [27]; *contCondition* - average number of conditional structures from submitted students' codes for a given problem [27]; *cyclomaticComplexity* - average cyclomatic Complexity from submitted students' codes for a given problem, where cyclomatic Complexity represent the source code as a control flow graph, corresponding to the number of independent paths of this graph [27]; *events* - average log lines of problems solved [34]; *nDistinctOperands* - average distinct arithmetic operands in the source codes; *nDistinctOperators* - average distinct arithmetic operators in the source codes; *quotientError* - average of repeated mistakes made by the students [18]; *quotientWatson* - attribution average penalties for mistakes made in a short period of time [42]; *sloc* - average lines of code sent in the online code edit [27]; *sucessAverage* - average problem submissions assessed as correct [13]; *test* - average number of times the

student tested the source code [13]; *totalOperands* - average operands in the source codes; *totalOperators* - average operators present in the source codes; *variables* - average number of variables in the source code [34].

We use such a large set of features to measure effort following [5], who explains that effort should be measured by a wide range of variables and expectations. Using this set of features forms the input data-driven behaviour for our Behaviour-based Recommender System (BRS) based on students' effort. The similarity between the recommended problem and the target problem is computed through nearest neighbour analysis, using cosine similarity as distance metric. We use this technique to support our first hypothesis that considers that a student s is able to solve a (recommended) problem p' of a same or similar level to a previously solved one p (target problem). As such, the nearest neighbour analysis is playing the role of matching the target and recommended problem by analysing the problems' similarities.

4 Evaluation of the Recommender Model

4.1 Participants

For the evaluation of our BRS, we recruited students who had already done introductory programming, from the Federal University of Roraima (UFRR), Brazil, due to convenience sampling, and since these students had already experience of learning with POJs, and could much easier and faster understand the purpose of the study. We have sent a message to computer science students from UFRR, explaining our research goals, asking for volunteers to participate in a 10-min phone call, scheduling calls for all who replied. Before the evaluation, we have explained the study to the learners and obtained their consent to participate. In total, 15 students agreed to participate of the experiment.

4.2 Measures

For each recommended problem, we asked the participants to make a comment about the effort required to solve the target problem and the recommended problem. Thus, we could evaluate manually the affective states within the comments based on the most frequent affective states when solving problems [10], which are boredom, confusion, engagement, neutral, and frustration. Besides that, in our context it is also crucial to evaluate when the learner is satisfied with the recommendation. Therefore, we included happiness, as [11] explain that happiness is a typical affective state presented when students are satisfied when solving problems. In Table 1, we summarise and describe all affective states used in our analysis.

Table 1. Affective states used to evaluate learner’s comments

Affective state	Description
<i>Boredom</i>	Uninterested in the current recommended problem
<i>Confusion</i>	Poor comprehension of the problem, attempts to resolve erroneous belief
<i>Engagement</i>	Student motivated to solve the current problem recommended
<i>Neutral</i>	No visible affect, at a state of homeostasis
<i>Frustration</i>	Problem recommended was not as expected, that is, more difficult or easier
<i>Happiness</i>	Satisfaction with the recommendation, feelings of pleasure about the problem

We further use a data-driven approach to evaluate the recommendations in terms of achievement using the following metrics: i) achievement rate, which is the proportion of correct submissions over all submissions; ii) failure rate, which is the proportion of incorrect over all submissions; and iii) dropout rate, which is the proportion of recommended problems that were not attempted by the students over all problems. It is worth mentioning that students were free to execute and submit solutions for the recommended problems as many times as they wished, i.e., there was no limit of attempts set for recommended problems.

4.3 Experimental Manipulation

To evaluate the BRS itself (experimental treatment) we compared the personalised recommendation with a random recommender (control treatment). We called the second one Random Recommender System (RRS) because the input is known but not the output, which means that given a target problem, the RRS recommends the next problem(s) by performing a random selection of questions from pre-determined lists of problems selected by instructors. Instructors created these lists for students on CS1 courses, so they are real-life recommendations. The comparison between the BRS and RRS was conducted through a within-subject double-blind controlled experiment, where neither students nor authors knew which treatment (BRS or RRS) they were receiving.

Thus to conduct our experiment, we created two personalised lists of recommendations using each recommender method (BRS and RRS). Each personalised list comprises 8 problems, totalling 16 (2×8) problems per student, containing 12 recommendations and 4 target problems, totalling 180 recommendations (i.e., 12×15) to be evaluated. Each student had their own personalised list split into 2 groups, with each group containing 4 problems. The first group was composed of easy problems, whilst the second one of intermediate problems. The first question of each group was a target problem (TP_1 and TP_2) that was selected by the authors of this study in collaboration with lecturers and professors of programming. These target problems act as a starting point to balance the RS towards

generating the recommendations, as a way to deal with the cold-start problem [21]. After the target problems, we have sequenced 3 automatic recommendations based on each target problem. Thus, we constructed the personalised list of recommended problems for the participants as follows: $TP_1 \rightarrow R_1, R_2, R_3$ and $TP_2 \rightarrow R_4, R_5, R_6$.

Doubtlessly, target problems were not included as part of the recommendation. For the easy target problems, we chose sequential and conditional problems (if...then...else), whereas for the intermediate problems we selected problems that use repetition structures (loops), vectors, strings and matrices. To personalise the recommendation for each student, we selected five different target problems for TP_1 and TP_2 . Moreover, we calculated the 10 nearest neighbours for each target problem, so that we had 10 different recommendations for each target problem. After that, we randomly assigned 3 out of the 10 nearest neighbours of a given target problem to compose the above recommendations.

5 Results and Discussion

We performed a qualitative analysis of the students' comments² over the recommended problems to identify their affective states. To perform that analysis, two authors independently classified each comment based on the affective states presented in Table 1. Subsequently, we performed a Kappa Cohen Test to check the agreement level and, as a result, we achieved 0.83, which is considered a high level of agreement [3]. For the cases of disagreement, another author acted as the third judge. Using this classification, Fig. 1 (left) shows the affective states present in the comments about the recommendations for each method. Comparing the methods, we can see a clear difference in terms of happiness and frustration (as affective measures - see Sect. 4.2). Indeed, the difference is statistically significant ($p - value < 0.05$, χ_2 - even after Bonferroni correction), which reveals that our method maximises the positive affective state (happiness, related to satisfaction), whilst minimising frustration.

Analysing each affective state in isolation, we observe only few neutral comments, which makes sense, since the students' comments about the recommendations tend to be pragmatic, that is, they usually stated that they were satisfied with the recommendation (happiness) or that the recommendations did not require the same effort as the target problems (frustration). Moreover, we can observe that boredom and engagement were not assigned for any comment. A possible reason is that the students may not have experienced an aversive state to the activity nor have felt sufficiently engaged as they treated the recommendations as an experiment and not as an usual learning activity.

Another affective state that occurred with a relatively low frequency ($N = 21$) was confusion. In total, there were 11 cases of confusion in the RRS and 10 cases in the BRS, which reveals a balance in relation to this state. This affective state occurred when students did not understand the problem statement, or the way in which the outputs of their codes should be presented in order to pass,

² www.dropbox.com/s/uxkvhohvuo1itmqq/comments.english.xlsx?dl=0.

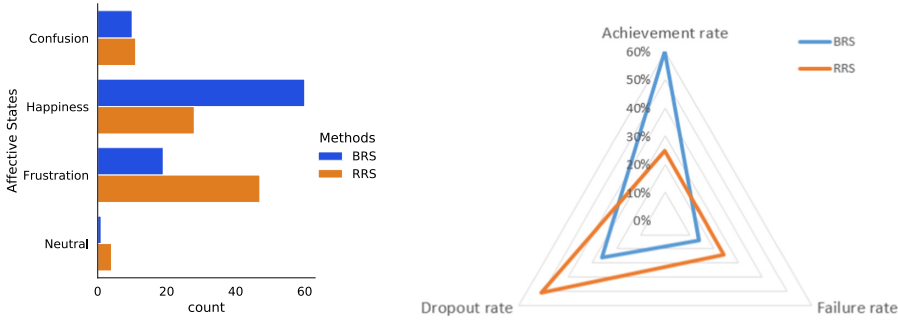


Fig. 1. Affective states classified based on learners' comments (left) and analysis of the recommendations submitted to the POJ (right).

for each test case. To illustrate, in some comments, students reported that their codes were correct, but the POJ did not judge them right because, apparently, the test cases were pointing out an error that they could not find. [41] state that this phenomenon can happen, as test cases are analysed by comparing strings, so if the student forgot a line break in a *print* command, then the problem can be assessed as wrong, even with the logic being right. Nonetheless, notice that this is a limitation of the way in which POJs assess exercises and not a limitation of our recommender method. Similarly, a poorly designed question (which can cause confusion) is out of the scope of our method. Still, these cases of confusion may have slightly influenced the failure rate and dropout in both methods. However, as there are almost the same number of confusion cases for both methods, their influence potentially weighed equally.

For happiness, there are 28 cases in our baseline, whereas 60 in our method, more than double. In addition, there were 47 cases of frustration in the RRS, whilst 19 in our method. This is a first evidence that our method is mitigating frustration, whilst maximising the students' satisfaction (i.e. happiness), supporting our first hypothesis that the recommendations will minimise students' negative affective states, whilst maximising the positive ones, as the problems recommended will not require a disproportionate effort from the learners.

After this qualitative analysis of the student affective states, we analysed the achievement of learners when solving the recommended problems. Figure 1 (right) shows the results for each method in terms of three rates: *achievement*, *failure* and *dropout*. When the students solved the problems recommended by the BRS, 60% of their submissions were assessed as correct (achievement rate), whereas using the RRS, the achievement rate was of only 25%. In terms of failure rate, only 14% of students' solutions were not accepted within the BRS, against 24% within the RRS. Indeed, these differences are statistically significant ($p - value < 0.05$ - χ^2 test, even after Bonferroni correction). Thus, these results indicate that for the RRS, the effort required to solve these problems is much higher than that used for the target problems. Furthermore, this evidence suggests the importance of recommending problems that are more appropriate

to the students' efforts, so as not to lead them to a low achievement rate and, hence, to frustration. Additionally, something worth noting is that for the RRS, students had a high rate of untried problems (51% of dropout rate), whereas for the BRS this was only 26%. These findings support our second hypothesis that recommendation based on effort expected will increase the student achievement and decrease dropout and failure rate.

About the difference in terms of dropout and failure rate, we can state that this is another confirmation that the problems recommended by the RRS either required more effort from learners or were more complex to the point where the students did not even try to solve them. Such high dropout rate from the RRS is a clear evidence of students' frustration in trying to solve problems not adequate to the effort expected, here, the one applied to the target problem. Other reasons that may have led the student not to try may be either the lack of understanding of the problem (confusion) or the lack of skills. Nonetheless, the target problems for each method are, respectively, an easy and intermediate problem. As such, if the RS works well, the first group of recommendations should comprise only easy problems, and the second group of recommendations should contain only intermediate problems. Consequently, the lack of skills to solve the problem should not have been present, as all the students who solved the recommended problems (via both methods) had already done introductory programming and were able to solve easy and intermediate problems. So, what likely happened was some bad recommendations in both systems, proportional to the students' dropout rate and failure rate. Notice that BRS was statistically superior in terms of achievement rate, failure rate and dropout rate, which likely means that the recommendation of the BRS were more suitable to students effort.

6 Pedagogical Implications

In summary, it is worth noting that our BRS shows potential to support programming classes for learners and for instructors who use POJs.

For the student, our recommender mitigates the burden of searching for problems that are adequate to their knowledge level and skills, capable of enabling self-directed learning in POJs. Moreover, our results showed that students felt less frustration and more happiness when completing assignments recommended by the BRS. Mitigating and improving frustration and happiness, respectively, is important to improve learning outcomes. Thus, this finding implies the usefulness of our proposed recommendation approach shows its potential to enhance learning experiences in solving programming assignments. Additionally, our finding about the students' achievement implies the stringent need to provide adequate recommendations for programming students to practice.

Finally, instructors typically need to create variations of programming assignments lists for different classes, in order to avoid plagiarism, for example. Using our method, considering each problem in a list of exercises already created by an instructor as a target problem. By generating N recommendations for each of these problems, we can automatically compose N new lists of exercises that

require effort and knowledge similar to those required to solve the original. Thus, *the instructor's workload to design new programming assignment lists is significantly reduced.*

7 Conclusions, Limitations and Future Works

The evidence we found in our qualitative analysis is aligned with the achievement rate analysis, supporting our hypotheses that, in general, new recommendations require a similar level of effort to the target problem. Indeed, the higher level of frustration in our baseline is potentially the driving factor that lead to such a high dropout and failure rate in problem-solving, whereas the higher rate of happiness might be related to the high achievement rate in our method. Thus, supporting our second hypothesis that the affective states influences achievement, defined here in terms of lower failure and dropout and higher number of problems solved.

Notice that human responses may be subject to bias [5], as it is difficult to control human attitudes and behaviour, even in a controlled experiment. Thus, the way we evaluated our recommendation method was designed to reduce potential biases. That is, besides the comments analysis, we also evaluated the students' interaction with the POJ and the problem solving process. In future works we envision to evaluate the effect of our methods for teaching and learning introductory programming by employing our method in real CS1 classes.

Finally, effort required to solve a given problem depends on the previous knowledge acquired by the learner about the topic of that problem. To illustrate, if the student already knew how to manipulate vectors using Python, it is easier for them to code a vector sum, as the *numpy* module allows summing up vectors as scalars. However, for a student who had no prior knowledge of vector manipulation with *numpy*, the effort to learn would be greater. The way effort was modelled in our BRS does not take into account this prior knowledge that the student would have about the topic. Thus, a potential limitation of our BRS is recommending a vector sum problem for a student who is learning how to sum scalars. As a way to solve that problem, in future works we envision to take into consideration topics of problems, and merge this study with other work we have on the topic of automatic detection [35]. Additionally, according to our first hypothesis, the problems recommended by our BRS tend to require similar effort of the target problem. However, students would not progress if the effort required to solve the next recommended problems does not increase. To deal with this, we also intend to merge this study with our work about detecting the difficult level of programming problems [23], so that we can create a mechanism to progressively increase the effort required to solve problems when making recommendations.

Acknowledgements. This research, carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 48 of Decree n° 6.008/2006 (SUFRAMA), was partially funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law n° 8.387/1991, through agreements 001/2020 and 003/2019, signed with Federal University of Amazonas and FAEPI, Brazil.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq grant 308513/2020-7).

References

1. Alamri, A., et al.: Predicting MOOCs dropout using only two easily obtainable features from the first week's activities. In: Coy, A., Hayashi, Y., Chang, M. (eds.) ITS 2019. LNCS, vol. 11528, pp. 163–173. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22244-4_20
2. Aljohani, T., Pereira, F.D., Cristea, A.I., Oliveira, E.: Prediction of users' professional profile in MOOCs only by utilising learners' written texts. In: Kumar, V., Troussas, C. (eds.) ITS 2020. LNCS, vol. 12149, pp. 163–173. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_20
3. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Educ. Psychol. Meas.* **20**(1), 37–46 (2008)
4. Bez, J.L., Tonin, N.A., Rodegheri, P.R.: Uri online judge academic: a tool for algorithms and programming classes. In: 2014 9th International Conference on Computer Science & Education, pp. 149–152. IEEE (2014)
5. Carbonaro, W.: Tracking, students' effort, and academic achievement. *Sociol. Educ.* **78**(1), 27–49 (2005)
6. Caro-Martinez, M., Jimenez-Diaz, G.: Similar users or similar items? comparing similarity-based approaches for recommender systems in online judges. In: Aha, D.W., Lieber, J. (eds.) ICCBR 2017. LNCS (LNAI), vol. 10339, pp. 92–107. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61030-6_7
7. Carter, A., Hundhausen, C., Olivares, D.: Leveraging the idea for learning analytics. In: Fincher, S.A., Robins, A.V. (eds.) *The Cambridge Handbook of Computing Education Research*, pp. 679–705. Cambridge University Press, Cambridge (2019)
8. Chau, H., Barria-Pineda, J., Brusilovsky, P.: Content wizard: concept-based recommender system for instructors of programming courses. In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pp. 135–140 (2017)
9. De Oliveira, M.G., Ciarelli, P.M., Oliveira, E.: Recommendation of programming activities by multi-label classification for a formative assessment of students. *Expert Syst. Appl.* **40**(16), 6641–6651 (2013)
10. D'Mello, S., Calvo, R.A.: Beyond the basic emotions: what should affective computing compute? In: *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pp. 2287–2294 (2013)
11. D'Mello, S.K., Lehman, B., Person, N.: Monitoring affect states during effortful problem solving activities. *Int. J. Artif. Intell. Educ.* **20**(4), 361–389 (2010)
12. Duckworth, A.L., Eichstaedt, J.C., Ungar, L.H.: The mechanics of human achievement. *Social Pers. Psychol. Compass* **9**(7), 359–369 (2015)
13. Dwan, F., Oliveira, E., Fernandes, D.: Predição de zona de aprendizagem de alunos de introdução à programação em ambientes de correção automática de código. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, vol. 28, p. 1507 (2017)
14. Fonseca, S., Oliveira, E., Pereira, F., Fernandes, D., de Carvalho, L.S.G.: Adaptação de um método preditivo para inferir o desempenho de alunos de programação. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, vol. 30, p. 1651 (2019)

15. Fonseca, S.C., Pereira, F.D., Oliveira, E.H., Oliveira, D.B., Carvalho, L.S., Cristea, A.I.: Automatic subject-based contextualisation of programming assignment lists. In: EDM (2020)
16. Haden, P.: Descriptive statistics. In: Fincher, S.A., Robins, A.V. (eds.) *The Cambridge Handbook of Computing Education Research*, pp. 102–131. Cambridge University Press, Cambridge (2019)
17. Hosseini, R., Brusilovsky, P.: A study of concept-based similarity approaches for recommending program examples. *New Rev. Hypermedia Multimedia* **23**(3), 161–188 (2017)
18. Jadud, M.C.: Methods and tools for exploring novice compilation behaviour. In: *Proceedings of the Second International Workshop on Computing Education Research*, pp. 73–84. ACM (2006)
19. Keller, J.M.: *Motivational Design for Learning and Performance: The ARCS Model Approach*. Springer, Heidelberg (2009). <https://doi.org/10.1007/978-1-4419-1250-3>
20. Kulkarni, P.V., Rai, S., Kale, R.: Recommender system in elearning: a survey. In: Bhalla, S., Kwan, P., Bedekar, M., Phalnikar, R., Sirsikar, S. (eds.) *Proceeding of International Conference on Computational Science and Applications*. AIS, pp. 119–126. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-0790-8_13
21. Lam, X.N., Vu, T., Le, T.D., Duong, A.D.: Addressing cold-start problem in recommendation systems. In: *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, pp. 208–211 (2008)
22. Lee, D.M.C., Rodrigo, M.M.T., Baker, R.S.J., Sugay, J.O., Coronel, A.: Exploring the relationship between novice programmer confusion and achievement. In: D’Mello, S., Graesser, A., Schuller, B., Martin, J.-C. (eds.) *ACII 2011*. LNCS, vol. 6974, pp. 175–184. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24600-5_21
23. Lima, M., de Carvalho, L.S.G., de Oliveira, E.H.T., Oliveira, D.B.F., Pereira, F.D.: Classificação de dificuldade de questões de programação com base em métricas de código. In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pp. 1323–1332. SBC (2020)
24. Luxton-Reilly, A., et al.: Introductory programming: a systematic literature review. In: *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pp. 55–106 (2018)
25. Ngai, G., Lau, W.W., Chan, S.C., Leong, H.V.: On the implementation of self-assessment in an introductory programming course. *ACM SIGCSE Bull.* **41**(4), 85–89 (2010)
26. de Oliveira, J., Salem, F., de Oliveira, E.H.T., Oliveira, D.B.F., de Carvalho, L.S.G., Pereira, F.D.: Os estudantes leem as mensagens de feedback estendido exibidas em juízes online? In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pp. 1723–1732. SBC (2020)
27. Otero, J., Junco, L., Suarez, R., Palacios, A., Couso, I., Sanchez, L.: Finding informative code metrics under uncertainty for predicting the pass rate of online courses. *Inf. Sci.* **373**, 42–56 (2016)
28. Pereira, F.D., Oliveira, E.H., Fernandes, D., Cristea, A.: Early performance prediction for cs1 course students using a combination of machine learning and an evolutionary algorithm. In: *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, vol. 2161, pp. 183–184. IEEE (2019)

29. Pereira, F., Oliveira, E., Fernandes, D., de Carvalho, L.S.G.C., Junior, H.: Otimização e automação da predição precoce do desempenho de alunos que utilizam juízes online: uma abordagem com algoritmo genético. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), vol. 30, p. 1451 (2019)
30. Pereira, F.D., et al.: Early dropout prediction for programming courses supported by online judges. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) AIED 2019. LNCS (LNAI), vol. 11626, pp. 67–72. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23207-8_13
31. Pereira, F.D., et al.: Can we use gamification to predict students' performance? a case study supported by an online judge. In: Kumar, V., Troussas, C. (eds.) ITS 2020. LNCS, vol. 12149, pp. 259–269. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49663-0_30
32. Pereira, F.D., Fonseca, S.C., Oliveira, E.H., Oliveira, D.B., Cristea, A.I., Carvalho, L.S.: Deep learning for early performance prediction of introductory programming students: a comparative and explanatory study. *Braz. J. Comput. Educ.* **28**, 723–749 (2020)
33. Pereira, F.D., Oliveira, E.H.T., Oliveira, D.F.B.: Uso de um método preditivo para inferir a zona de aprendizagem de alunos de programação em um ambiente de correção automática de código. Universidade Federal do Amazonas, Manaus, Mestrado em informática (2018)
34. Pereira, F.D., et al.: Using learning analytics in the amazonas: understanding students' behaviour in cs1. *Brit. J. Educ. Technol.* **51**, 955–972 (2020)
35. Pereira, F.D., et al.: Towards a human-ai hybrid system for categorising programming problems. In: SIGCSE 2021. Association for Computing Machinery, New York (2021). <https://doi.org/10.1145/3408877.3432422>
36. Pereira, F.D., de Souza, L.M., de Oliveira, E.H.T., de Oliveira, D.B.F., de Carvalho, L.S.G.: Predição de desempenho em ambientes computacionais para turmas de programação: um mapeamento sistemático da literatura. In: Anais do XXXI Simpósio Brasileiro de Informática na Educação, pp. 1673–1682. SBC (2020)
37. Revilla, M.A., Manzoor, S., Liu, R.: Competitive learning in informatics: the UVA online judge experience. *Olymp. Inf.* **2**(10), 131–148 (2008)
38. Rodrigo, M.M.T., Baker, R.S.: Coarse-grained detection of student frustration in an introductory programming course. In: Proceedings of the Fifth International Workshop on Computing Education Research Workshop, ICER 2009., pp. 75–80 Association for Computing Machinery, New York (2009)
39. Saito, T., Watanobe, Y.: Learning path recommendation system for programming education based on neural networks. *Int. J. Dist. Educ. Technol. (IJDET)* **18**(1), 36–64 (2020)
40. dos Santos, I.L., Oliveira, D.B.F., de Carvalho, L.S.G., Pereira, F.D., de Oliveira, E.H.T.: Tempos de transição em estados de corretude e erro como indicadores de desempenho em juízes online. In: Anais do XXXI Simpósio Brasileiro de Informática na Educação, pp. 1283–1292. SBC (2020)
41. Wasik, S., Antczak, M., Badura, J., Laskowski, A., Sternal, T.: A survey on online judge systems and their applications. *ACM Comput. Surv. (CSUR)* **51**(1), 1–34 (2018)
42. Watson, C., Li, F.W., Godwin, J.L.: Predicting performance in an introductory programming course by logging and analyzing student programming behavior. In: 2013 IEEE 13th International Conference on Advanced Learning Technologies, pp. 319–323. IEEE (2013)

43. Yera, R., Martínez, L.: A recommendation approach for programming online judges supported by data preprocessing techniques. *Appl. Intell.* **47**(2), 277–290 (2017)
44. Yera Toledo, R., Caballero Mota, Y., Martínez, L.: A recommender system for programming online judges using fuzzy information modeling. In: *Informatics*, vol. 5, p. 17. Multidisciplinary Digital Publishing Institute (2018)
45. Yu, R., et al.: The research of the recommendation algorithm in online learning. *Int. J. Multimedia Ubiq. Eng.* **10**(4), 71–80 (2015)
46. Zhao, W.X., Zhang, W., He, Y., Xie, X., Wen, J.R.: Automatically learning topics and difficulty levels of problems in online judge systems. *ACM Trans. Inf. Syst. (TOIS)* **36**(3), 27 (2018)
47. Zordan Filho, D.L., de Oliveira, E.H.T., de Carvalho, L.S.G., Pessoa, M., Pereira, F.D., de Oliveira, D.B.F.: Uma análise orientada a dados para avaliar o impacto da gamificação de um juiz on-line no desempenho de estudantes. In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pp. 491–500. SBC (2020)



Evaluation Test Generator Using a List of Keywords

Doru Anastasiu Popescu^{1(✉)}, Gabriel Ciprian Stanciu², and Daniel Nijloveanu³

¹ Department of Mathematics and Computer Science, University of Pitesti, Pitesti, Romania

² Delft University of Technology, Delft, Netherlands

³ Sciences and Veterinary Medicine Bucharest, Faculty of Management and Rural Development, University of Agronomical, Slatina Branch, Romania

Abstract. Assessment is one of the most important issues in terms of learning. A new challenge in the field of assessment came in the long periods of learning on online platforms. In this paper we propose a model for generating evaluation tests consisting of questions that have statements, keywords and answers. The genetic algorithm that generates the tests will use the keywords to rank the tests through the fitness function. Once the test that maximizes the number of keywords in the initially given list is generated, it can be used for evaluation by the teacher through various mobile applications, web, etc. The implementation of the algorithm in the paper was done using the Java language. The visual application at the end of the paper highlights the results of the presented algorithm.

Keywords: Genetic algorithm · E-learning · Chromosome · Education · Assessment

1 Introduction

Assessment is an important component in the learning process. Especially, with the pandemic situation in 2019–2021, the usefulness of using online platforms-based learning systems has proved essential. Testing the knowledge acquired in the courses is a intensely debated topic at present, studies such as those in [1, 3] and [10] present the main aspects covered by them. In [13] and [14] the mechanisms used in the evaluation of learning are presented. A possible method to verify the knowledge of students in certain courses is the use of tests consisting of questions, which involves selecting one option from a group of possible answers. Learning platforms have implemented various ways of creating tests, but these are generally based on the teacher building the tests as presented in [3, 4] and [12].

This paper aims to provide a way to create a test consisting of a fixed number of questions selected from a database using keywords that are associated with the questions. Because the number of questions used in tests can be in the hundreds or even thousands, we cannot use exponential complexity generation algorithms (for example those that use the backtracking method) and that is why we used genetic algorithms, even if the solution is approximate (the optimal condition is not checked in all cases, but converges to it).

These algorithms are used in many areas where we have to select subsets of elements that roughly fulfill an optimal property from a set with many elements. In [2] and [11] models for generating such tests based on genetic algorithms are presented. Depending on the objective pursued by the evaluating teacher or on the hardware and software resources available to teachers and students, once created these tests can be used through online access platforms, web and mobile services, aspects that are presented in detail in [4] and [9].

We started from the premise that the teacher uses a great amount of questions to create a test, the questions naturally being created in parallel with the lectures on the sections of the course. These questions are characterized by: ID (natural identification number), statement (text), answer options (we used 3 options in implementing the algorithm), the correct variant and keywords associated with the question. Compared to existing models based on genetic algorithms, described in articles such as [5, 8] or [6] - the novelty of the model presented in this article is related to the fact that we associate all questions with a list of keywords (generally small number: 1–3 keywords most of the time) and test a general list of keywords. The test is generated by the application presented in this article in order to contain (in the keywords of the questions) as many keywords as possible from the general list associated with the test (entered by the teacher in the application). The idea presented can be useful in other types of learning such as those based on blocks in interactive visual programming environments (Mindstorms, Scratch, etc.), aspects described in the article [7]. In addition, with small changes the algorithm can provide more tests with the property of maximizing the number of keywords in the overall list.

2 The Elements of a Test and the Conditions It Must This

During a course, the teacher identifies the most important notions (referred to as keywords) and uses them to create questions with answer options (one or more of them being true). These will be entered into a database. The information for a question will be:

- ID (unique natural questionidentificationnumber)
- Statement (text withthe content of thequestion)
- Keyword list
- Answeroptions (in theimplementation of ouralgorithmwewilluse 3 variants)
- Correctansweroption

For example, if we are in the *Data Structures* course a possible question could be defined by the information:

- 23
- If initially the stack S is empty, after the operations: Insert 10, Insert 304, Delete, Insert 11, Insert 13, Insert 10, Delete, what are its elements (in the correct order)?
- Stack, Delete, Insert
- a) 10, 11, 13 b) 13, 11, 10 c) 10, 13, 11, 10
- b)

Thus, at some point we can use the questions created to build a test based on a list of keywords. An example of a keyword list associated with a test for the *Data Structures* course could be:

Stack, Queue, Set, Binary Trees, Insert, Delete.

To make it easier to use the questions when creating a test in the algorithm, the ID will be used because it is a unique number in the database that contains the questions. Thus, if we have 100 questions with IDs 1, 2, ..., 100 a test with 7 questions could be defined by the sequence of IDs: 13, 89, 4, 99, 45, 8, 11. We will take care to generate a test that maximizes the number of keywords that appear in the list associated with the test (chosen by the teacher before generation). A simplified diagram of a system model for generating such a test is shown in Fig. 1.

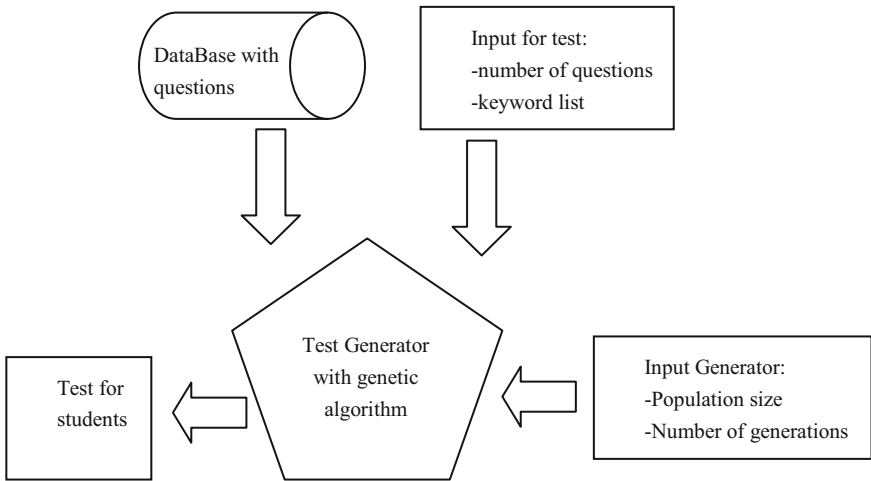


Fig. 1. Model for generating a test system

3 Algorithm for Generating a Test

For the Genetic Algorithm used in this section we use the form described in the book [15] or papers [5, 8]. Firstly, we will define the main notions that will be used in the test generation algorithm.

Definition 1. A question q ($id; st; kl; as; ca$) is an object composed of the next components:

- the identification number of the question id ;
- the statement st ;
- keyword list kl ;
- answers list as ;
- correct answer ca .

Observations 1

- Question identification number id ;
- The test that the algorithm will generate will contain nt questions
- klt is the keyword list that will be used for the test
- A test $T(S, nkl)$ is a set of questions $q_i, i = 1, |S|$, where S is the set of questions that forms the test and nkl is the number of keywords in klt that appear in the keyword list of a question in S :

$$nkl = |\{kw \in klt \mid \exists q \in S : kw \in kl \text{ from } q\}| \quad (1)$$

Definition 2. Given the database question list Q , the keyword list klt and nt a natural number less than or equal to $|Q|$, a chromosome C is an list with nt components. The components of the list are question numbers in the Q list, called genes (numbers from the set $\{1, 2, \dots, nt\}$).

Observations 2

- a) In a test a question can not appear several times and therefore the components (genes) of a chromosome are distinct.
- b) The genes of a chromosome uniquely define a question and so any test is defined by a chromosome.
- c) Using the specifications in the definition from a C chromosome we obtain the test $T(S, nkl)$, where S is the set of questions from the list Q , which have indices equal to the genes in C , and nkl is the number defined by the relation (1).
- d) In order to obtain a "good" test, nkl must be as close as possible to the number of elements of the klt list. If $nkl = klt$, then the test obtained is "verygood".

Definition 3. For a C chromosome, the value of the fitness function will be the number nkl , given by formula (1) representing the number of keywords in the list given by the teacher for the test, which are found in the test obtained from chromosome C .

Observation 3

The *fitness function* that measures our chromosome quality can be supplemented with other conditions when comparing them. Thus, if two chromosomes have the same *fitness value function*, then a new condition can be added, such as the number of keywords in the test associated with the "considered better" chromosome that are not on the klt teacher's list to be lower.

Definition 4. Given a chromosome C_i ($i = 1, |NCl$) and random positions a and b ($a, b = 1, |S|$), the mutation operation is defined as the shift of the genes found on the positions a and b .

Definition 5. Given two chromosomes C_i and C_j and a random position p , the crossover operation is defined as a succession of steps as follows:

The two chromosomes are split at the position p .

The first part of the chromosome C_i is combined with the second part of the chromosome C_j and the first part of the chromosome C_j is combined with the second part of the chromosome C_i .

Two new chromosomes C'_i and C'_j are obtained, as follows:

$$C'_i = (g_{i_1}, g_{i_2}, \dots, g_{i_{p-1}}, g_{j_p}, \dots, g_{j_s}) \tag{2}$$

$$C'_j = (g_{j_1}, g_{j_2}, \dots, g_{j_{p-1}}, g_{i_p}, \dots, g_{i_s}) \tag{3}$$

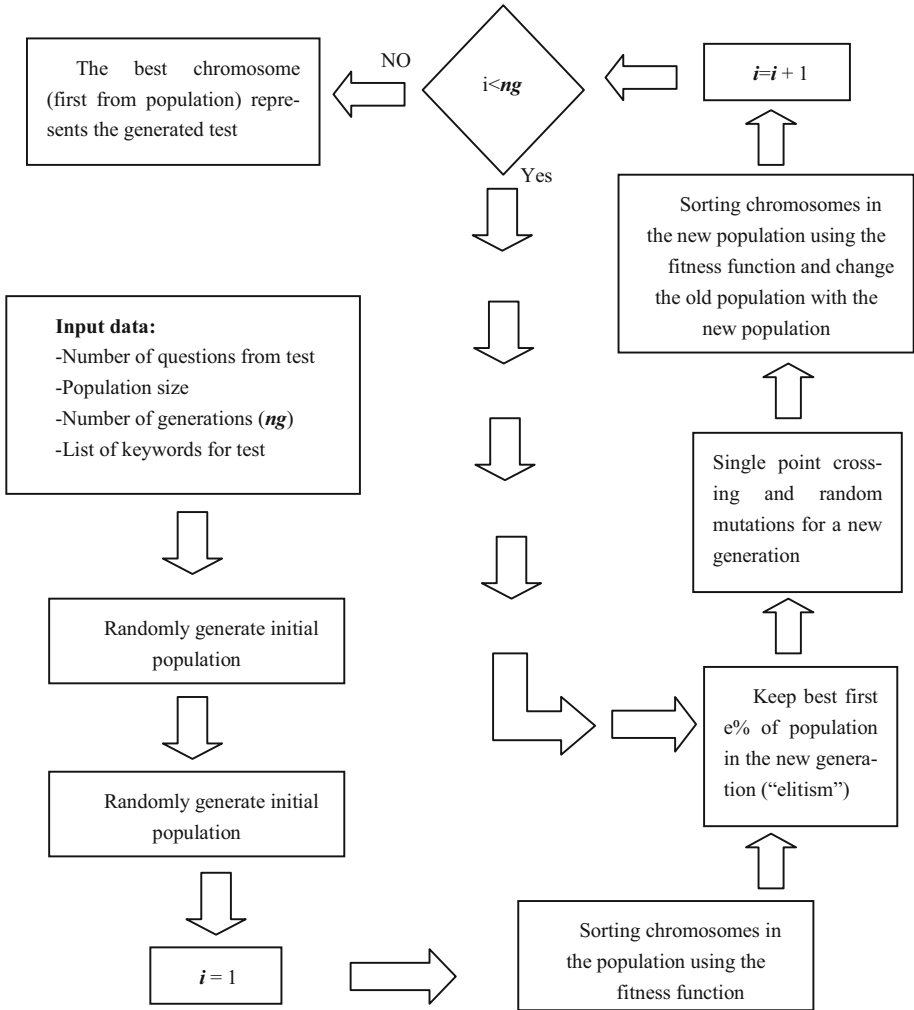


Fig. 2. Genetic algorithm for generating a test system

If, the two new chromosomes C_i' or C_j' chromosomes defined by (2) and (3) respectively contain equal genes, then one of them will be replaced by a randomly generated number that does not exist in the chromosome and is less than or equal to $|Q|$.

Observation 4

The mutation and crossover has as result the generation of a new chromosome. A population consists of a list of chromosomes, their number being entered into the algorithm by the user. The initial population (Q) will be randomly generated.

Within the algorithm, the order of the operations is:

- Generation of the initial population (Q)
- Sort of chromosomes based on fitness
- Mutation of chromosomes for $e\%$ from population (“elitist” chromosomes)
- Crossover of chromosomes

Operations b), c) and d) are repeated for a previously-set number of generations. In the implementation of the algorithm the best results were obtained with $e = 20$.

The steps of the algorithm is presented in Fig. 2.

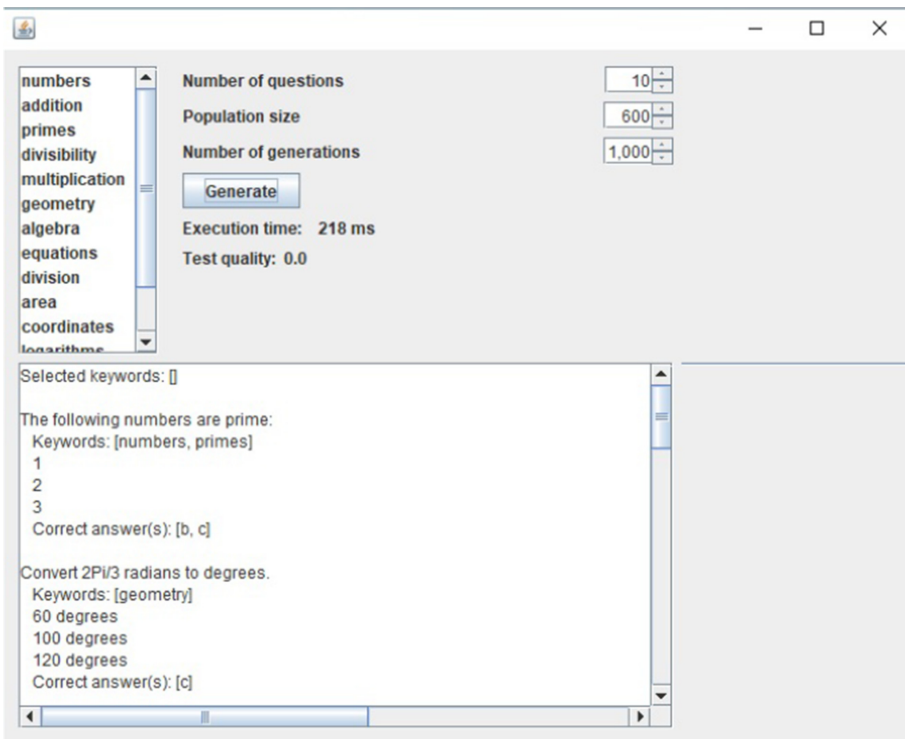


Fig. 3. Interface for generator

The final result is a list of tests from which we store a finite number of tests which have the highest value of the fitness. The first test in the list (population) generated by the algorithm after the last repetition of operations b), c), d) will be the sought solution.

4 Implementation

For the implementation of the algorithm in Sect. 4 we used the Java language. The interface of the application is shown in Fig. 3.

The Java application interface allows us to enter the following data to generate a test:

- Keyword list (top left of window)
- Number of questions
- The number of chromosomes used for the population
- Number of generations

Pressing the Generate button will provide the test questions. After deleting the correct answers the test can be used.

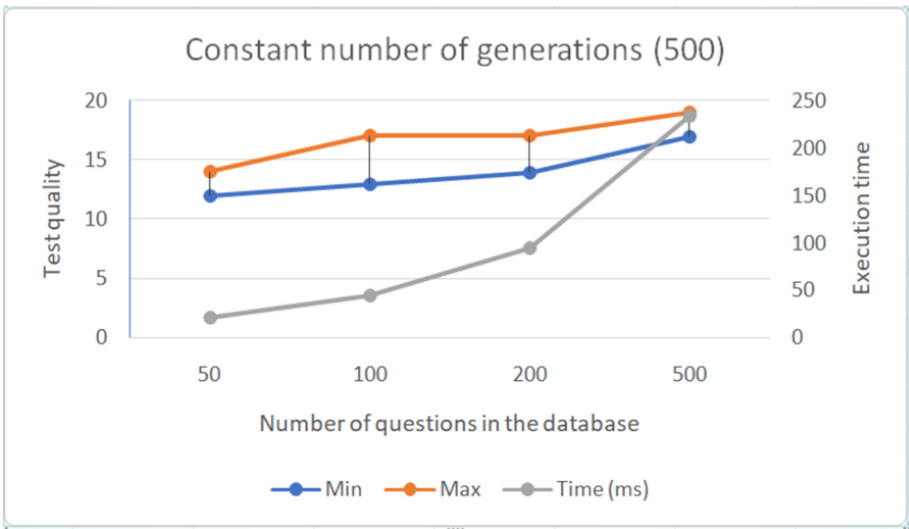


Fig. 4. Visual representation of the result with generator test

Table 1. Result with generator test

Number of questions in database	50	100	200	500
Min fitness function (nkl)	12	13	14	17
Max fitness function (nkl)	14	17	17	19
Execution time (ms)	21	44	94	234

Using the test generator several times for 500 generations, 20 questions, 20 keywords and various databases we obtained the results from Table 1 and Fig. 4.

Using the test generator several times for 200 generations, 20 questions, 20 keywords and various databases we obtained the results from Table 2 and Fig. 5.

Table 2. Result with generator test

Number of questions in database	100	200	500	1000
Min fitness function (<i>nkl</i>)	13	14	14	13
Max fitness function (<i>nkl</i>)	15	16	17	16
Execution time (ms)	9	18	44	90

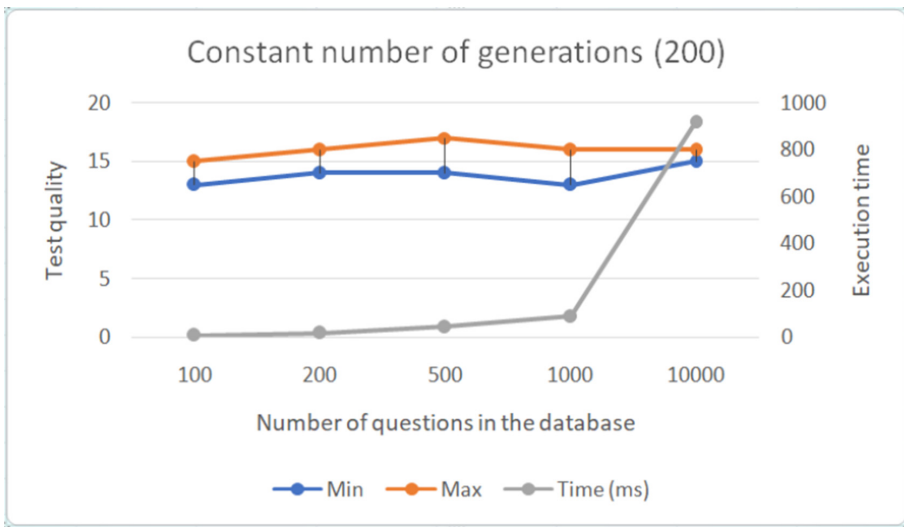


Fig. 5. Visual representation of the result with generator test

5 Conclusions

The automation of the evaluation activity is inevitable, this activity being in a continuous development and that is why we consider that our approach through this article to be a constructive one. The novelty element is the use of keywords for each question in the database associated with a course. The fact that in the test generator a genetic algorithm was used has the advantage of obtaining different tests at repetitive executions of the application. The article completes the results obtained in this education field, some of them presented in papers [4–6] and [11].

The generator model presented in this article can be used on e-learning platforms and can be completed with various web or mobile applications for distributing the students’ tests and capturing the results and centralizing them.

References

1. Becheru, A., Popescu, E.: Design of a conceptual knowledge extraction framework for a social learning environment based on social network analysis methods. In: Proceedings ICC 2017, pp. 177–182 (2017)
2. Alharbi, S., Venkat, I.: A genetic algorithm based approach for solving the minimum dominating set of queens problem. *J. Optim.* **2017**, 1–8 (2017)
3. Boopathiraj, C., Chellamani, K.: Analysis of test items on difficulty level and discrimination index in the test for research in education. *Int. J. Social Sci. Interdiscipl. Res.* **2**(2), 189–193 (2013)
4. Yang, B.W., Adam, J.R., Persky, M.: Using Testing as a Learning Tool. *Am. J. Pharm. Educ.* **83**(9), 7274 (2019)
5. Colorni, A., Dorigo, M., Maniezzo, V.: A Genetic Algorithm To Solve The Timetable Problem (1994)
6. Popescu, D.A., Bold, N., Domsa, O.: A generator of sequences of hierarchical tests which contain specified keywords. In: 11th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2016. Timisoara, Romania, 12–14 May (2016)
7. Popescu, D.A., Nijloveanu, D., Bold, N.: Generator of Tests for Learning Check in Case of Courses that Use Learning Blocks. In: Di Mascio, T., Vittorini, P., Gennari, R., De la Prieta, F., Rodríguez, S., Temperini, M., Silveira, R.A., Popescu, E., Lancia, L. (eds.) *Methodologies and Intelligent Systems for Technology Enhanced Learning*, 8th International Conference, pp. 239–244. Springer International Publishing, Cham (2019)
8. Popescu, D.A., Bold, N., Domsa, O.: Generating assessment tests with restrictions using genetic algorithms. In: 12th IEEE International Conference on Control and Automation, ICCA 2016. Kathmandu, Nepal, 1–3 June (2016)
9. Popescu, E., Stefan, C., Ilie, S., Ivanović, M.: EduNotes – A mobile learning application for collaborative note-taking in lecture settings. In: Chiu, D.K.W., Marenzi, I., Nanni, U., Spaniol, M., Temperini, M. (eds.) *Advances in Web-Based Learning – ICWL 2016*, pp. 131–140. Springer International Publishing, Cham (2016)
10. Guang, C., et al.: A implementation of an automatic examination paper generation system. *Math. Comput. Model.* **51**(11–12), 1339–1342 (2010)
11. Li, Y., Li, S., Li, X.: Test paper generating method based on genetic algorithm. *AASRI Procedia* **1**, 549–553 (2012)
12. Liu, D., Wang, J., Zheng, L.: Automatic test paper generation based on ant colony algorithm. *J. Softw.* **8**, 2600–2606 (2013)
13. Alruwais, N., Wills, G., Wald, M.: Advantages and challenges of using e-assessment. *Int. J. Inform. Educ. Technol.* **8**(1), 34–37 (2018)
14. Thessen, A.E., Cui, H., Mozzherin, D.: Applications of natural language processing in biodiversity science. *Adv. Bioinform.* **2012**, 1–17 (2012)
15. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA (1996)



Voice Emotion Recognition in Real Time Applications

Mahsa Aghajani^(✉), Hamdi Ben Abdesslem, and Claude Frasson

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal
H3C 3J7, Canada
{mahsa.aghajani, hamdi.ben.abdessalem}@umontreal.ca,
frasson@iro.umontreal.ca

Abstract. This paper reports the results of voice emotion recognition in real time using machine learning models. The models are trained with some commonly used and well-known audio emotion datasets together with a custom dataset. This custom dataset was recorded from non-actor and non-expert people who were trying to imagine themselves in scenarios leading to arise of the related emotion. The reason for considering this important dataset is to make the model proficient in recognizing emotions in people who are not perfect in reflecting their emotions in their voices. The results from several machine learning classifiers while recognizing five emotions like anger, happiness, sadness, neutrality and surprise are compared. Models were evaluated with and without considering the custom data set to show the effect of employing an imperfect dataset. Our experiments showed that without using our custom dataset, the ensemble machine learning models such as gradient boosting, bagging and random forest reach validation accuracies 89.82%, 88.58% and 84.83% respectively, which are higher than other evaluated models. After considering our custom dataset, again these ensemble methods obtained better accuracies of 87.34%, 86.71% and 82.98% respectively. This shows that although considering our custom dataset lowers the overall accuracy but empowers the model for predicting the emotions in everyday scenarios.

Keywords: Voice emotion recognition · Machine learning · Brain computer interface · Affective computing

1 Introduction

In today's applications, recognizing emotions in users' voices leads to a better user experience and more user-friendly applications. In applications where the user's satisfaction is an important factor or the competitive advantage of that application, detecting the emotion of the user through interacting with the application becomes of great value. Examples of these applications are customer support applications, artificially intelligent virtual assistants, etc. [1, 2].

Along with these mentioned applications, many other applications can also benefit from voice emotion detection hugely but have some specific requirements too. One of

these specific requirements for this set of applications is that the emotion detection should be done in real time. Instances of such applications are educational applications, video games, driving assistant applications, etc. The need for real time emotion detection in these applications originates from enabling the application to behave differently based on the current emotion of the user. For example, if the student is stressed, angry or sad, the hardness level of following materials can be reduced; On the other hand, getting a positive feedback from the student's emotion, can allow us to increase the hardness of the upcoming material. For Alzheimer patients detecting negative emotions would be important to apply for instance relaxing techniques.

For detecting the emotion from voice, we have used several machine learning models. For training these models, there are a few rich datasets. In these datasets, usually the speakers are actors, who are most of the time experts in showing and reflecting their emotions in their voices. Although this characteristic of these commonly used datasets, makes them ideal for training and validating machine learning models, but debilitates the models in predicting the emotions in non-actor and non-expert people's voices in everyday scenarios. Therefore, in addition to these available datasets, we need other datasets, recorded from non-actor people for training and validating the models.

Detecting the emotion using the speaker's voice in real time applications has several important challenges; First the nature of this detection is a completed task, even for humans. In different scenarios, based on the person's ability to reflect his emotions in his voice, this recognition task can get even harder. For example, when the speaker suffers from the Alzheimer disease or when the speaker has autism. Even with people who do not have these conditions, detecting several emotions from each other in their voice, such as happiness and surprise, or neutrality and sadness requires high proficiency. In all these cases, the predictor model should already be trained with similar data; Second, the feature set for machine learning models should be selected carefully. The third challenge is related to the real time emotion detection requirement of the target applications. Considering this specific need, the emotion detection should be done in an acceptable amount of time.

The outline of this paper is as follows: Sect. 2 reviews related work in voice emotion recognition. Section 3 introduces the datasets we have used. In Sect. 4 we describe our methodology and machine learning models. Section 5 is about explaining the experiments and reporting their results. Section 6 concludes our work and discusses our future works that may lead to improvements.

2 Related Work

Voice emotion recognition has been around for decades [3–5]. In most of these works, the data collection process was done in a studio environment [6], leading to using clear audio files for training the models. In [6] it is aimed to use a corpus that has background noise and is close to the real world. The authors used the Multimodal Emotion Challenge (MEC) 2017 corpus. This corpus contains clips from films and TV programs and the speakers are professional actors; therefore, the problem of having models that are trained with emotion reflective voice is still remaining.

H. Chen et al. in [7] used CASIA Chinese Emotional Speech Corpus in training several models. The authors report SVM as the best model with the highest accuracy of

81.11%. A perfect dataset, from reflecting the emotional point of view, is also considered in this work.

S. Yacoub et al. in [8] have focused on extracting features from short utterances which are commonly used in Interactive Voice Response (IVR) applications. Authors have recognized the anger and neutral emotions from each other with an accuracy of 90% with models trained over the Linguistic Data Consortium at University of Pennsylvania. In our work, we obtained the accuracy of 90% and 87% (without and with our custom dataset respectively) while predicting five emotions at the same time.

In [9] the relation between age and gender and the emotion recognition accuracy is studied. The authors have obtained an accuracy of 74% by using a hierarchal model, trained and validated over the RAVDESS [10] dataset, which contains speakers of different ages.

Our work's novelty is in gathering and using non-expert and non-actor speaker voices, trying to involve them in scenarios leading to emerging the related emotion. We also evaluated several machine learning models, trying to predict 5 emotions at the same time. Due to the covid-19 restrictions, while conducting our experiments, we were not able to extend our validation techniques with EEG experiments able to directly provide emotions assessments. After returning to the normal situation, we will resume our validation procedures with EEG experiments.

3 Datasets

In all machine learning applications, selecting the proper dataset is extremely important. There are many different datasets for voice emotion recognition [11]. In our work, Toronto Emotional Speech Set (TESS) [12], Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), Surrey Audio-Visual Expressed Emotion (SAVEE) Database [13] and a Custom Database these datasets are used for training and validating the models. In our custom dataset, audio files were recorded from non-actor speakers whom were asked to say different sentences while trying to imagine themselves in a situation causing them to have the related emotion. Table 1 shows the distribution of audio files related to each emotion in the final dataset which is the result of appending all the above datasets:

Table 1. Distribution of audio files related to each emotion in the whole dataset

Emotion	Final dataset	
	Count	Proportion
Angry	677	0.202
Happy	677	0.202
Neutral	641	0.191
Surprised	677	0.202
Sad	677	0.202

4 Methodology

After gathering a rich dataset from mentioned emotional databases, we extracted appropriate features from the audio files. The extracted feature set was fed into several machine learning classifiers separately. For validating the performance of each classifier, the input data set was split into training and validation sets under two different approaches. In the first approach, 25% of the input data set was randomly chosen as the validation set. In the second approach, 25% of the input data set was selected as the validation set in an actor-based approach; that is there was no common actor between the training and validation sets. The goal of this approach was to validate the performance of the classifiers dealing not only with unseen data but also when the input voice is completely new for the classifier. For feature extraction, we have used the feature set provided in INTERSPEECH 2010 paralinguistic challenge [14], which is a common selected feature set among related works. These features are fed into several classifiers such as SVM, random forests, bagging, gradient boosting and RNN.

5 Results and Discussion

For validating the models, we considered two datasets; The first dataset contained TESS, RAVDESS and SAVEE datasets, which are all common speech emotion recognition datasets, recorded by actor speakers. In the second case, we added our custom dataset to the previous datasets.

For validating the prediction results of classifiers trained and tested with these two datasets, we have used 5-folds cross fold validation technique with the same distribution of classes in each round of validation, using StratifiedKFold class from Sklearn library.

The predicted results of classifiers trained and tested with these two datasets are explained in the following parts.

5.1 Considering TESS, RAVDESS and SAVEE Datasets

In this case, classifiers were trained with TESS, RAVDESS and SAVEE dataset. Table 2 contains the average performance measures of each classifier on the test set. The results for each performance measure are the averages over the results of that classifier over all the 5 repetitions of 5-fold cross validations. As it can be seen, gradient boosting, bagging and random forest have the best average accuracies of 89.82%, 88.58% and 84.83% respectively. These classifiers have also the overall best performance measures among all the classifiers which shows that we can rely on ensemble methods for speech emotion prediction applications.

5.2 Considering TESS, RAVDESS, SAVEE and Custom Datasets

In the second case, the dataset included our custom dataset together with three previously mentioned datasets. In this approach, the total number of audio files reached 3349 files. Again, we validated our models using 5-fold cross validation.

Table 2. Average performance measures of each classifier on the test set

Classifier	Accuracy	F1-score	Precision	Recall
SVM	0.7795	0.7794	0.7811	0.7795
Random forest	0.8483	0.8482	0.8506	0.8483
Bagging	0.8858	0.8855	0.8862	0.8858
Gradient boosting	0.8982	0.8981	0.8995	0.8982
RNN	0.7444	0.7454	0.7744	0.7193

Table 3. Average performance measures of each classifier on the test set in the second approach

Classifier	Accuracy	F1-score	Precision	Recall
SVM	0.7647	0.7639	0.7669	0.7647
Random forest	0.8298	0.8296	0.8328	0.8298
Bagging	0.8671	0.8669	0.8681	0.8671
Gradient boosting	0.8734	0.8732	0.8748	0.8734
RNN	0.7082	0.7077	0.7093	0.7061

Since the speakers in the recorded audio files in our custom dataset were not professional actors, we anticipated a decrease in the accuracy of the classifiers, which is also shown in Table 3.

Figure 1 shows another comparison between the accuracies of the classifiers on the test set after adding our custom dataset. In this case, the gradient boosting and bagging classifiers had the best prediction results with average test accuracies of 87.34% and 86.71% respectively.

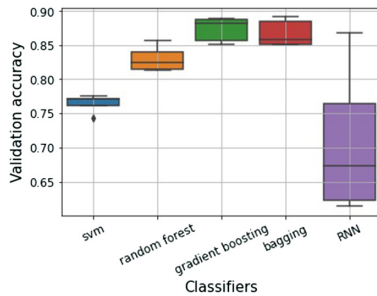


Fig. 1. Validation accuracies of classifiers in the second approach

Considering Fig. 2, which is the confusion matrix for gradient boosting as our best classifier, it can be observed that predicting neutrality has the best accuracy of 93.0% among other emotions. On the other hand, predicting happiness has the lowest accuracy

of 84.0%. It should also be noted that among the audio files that are predicted as angry ones, 4.7% of them are truly happy audio files. This shows that anger and happiness are the most confusing emotions when predicting anger. The same scenario holds when the emotion is predicted to be neutrality, that is 4.1% of the files recognized to have neutrality as their most dominant emotion, are in fact sad. With the same explanation, the most confusing emotion while predicting surprise is happiness.

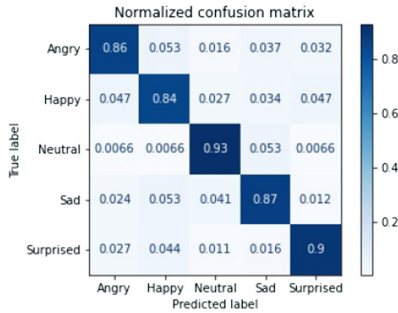


Fig. 2. Confusion matrix of gradient boosting classifier with 5 emotions

6 Conclusion and Future Work

In this study, we reported that by using audio features provided in INTERSPEECH 2010 paralinguistic challenge, we could get acceptable prediction accuracies from several ensemble classifiers. We also explained why considering datasets recorded from non-actor speakers is necessary while training the machine learning models. We tried to tackle this challenge by gathering a custom data set. We also discussed the most confusing emotion pairs in our experiments.

Since in real case scenarios, emotions may not always be reflected in the speaker's voice noticeably, getting help from spoken words may help in predicting the true emotion. Regarding this, we are going to use speech to text techniques and add the recognized words to our feature set. We are also going to increase the size of our custom dataset to evaluate its effect in real case emotion detection scenarios. In this work we recognized 5 emotions, considering the limitations in gathering the custom dataset; We are going to expand our considered emotion set to 7 emotions of angry, happy, sad, neutral, disgust, fear and surprised. With all of this, we are going to extend our testing procedures using EEG experiments able to measure emotions and compare them with our current and modified classifiers.

Acknowledgment. We acknowledge NSERC-CRD (National Science and Engineering Research Council Cooperative Research Development), Prompt, and BMU (Beam Me Up) for funding this work.

References

1. Petrushin, V.: Emotion in speech: Recognition and application to call centers. In: Proceedings of Artificial Neural Networks in Engineering, vol. 710, p. 22 (1999)
2. Petrushin, V.A.: Emotion recognition in speech signal: experimental study, development, and application (2000)
3. Busso, C., et al.: Analysis of emotion recognition using facial expressions, speech and multimodal information. In: Proceedings of the 6th International Conference on Multimodal Interfaces, pp. 205–211 (2004)
4. Lee, C.M., et al.: Emotion recognition based on phoneme classes (2004)
5. Deng, J., Xinzhou, X., Zhang, Z., Frühholz, S., Grandjean, D., Schuller, B.: Fisher kernels on phase-based features for speech emotion recognition. In: Jokinen, K., Wilcock, G. (eds.) Dialogues with social robots, pp. 195–203. Springer Singapore, Singapore (2017). https://doi.org/10.1007/978-981-10-2585-3_15
6. Tao, F., Liu, G., Zhao, Q.: An ensemble framework of voice-based emotion recognition system for films and TV programs. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6209–6213 (2018)
7. Chen, H., Liu, Z., Kang, X., Nishide, S., Ren, F.: Investigating voice features for Speech emotion recognition based on four kinds of machine learning methods. In: 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), pp. 195–199 (2019)
8. Yacoub, S., Simske, S., Lin, X., Burns, J., Recognition of emotions in interactive voice response systems (2003)
9. Shaqra, F.A., Duwairi, R., Al-Ayyoub, M.: Recognizing emotion from speech based on age and gender using hierarchical models. *Proc. Comput. Sci.* **151**, 37–44 (2019)
10. Livingstone, S.R., Russo, F.A.: The Ryerson Audio-visual database of emotional speech and song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE* **13**(5), e0196391 (2018)
11. El Ayadi, M., Kamel, M.S., Karray, F.: Survey on speech emotion recognition: features, classification schemes, and databases. *Pattern Recognit.* **44**(3), 572–587 (2011)
12. Pichora-Fuller, M.K., Dupuis, K.: Toronto emotional speech set (TESS). *Scholars Portal Dataverse* (2020)
13. Surrey Audio-Visual Expressed Emotion (SAVEE) Database. <http://kahlan.eps.surrey.ac.uk/savee/>. Accessed 5 Jan 2021
14. Schuller, B., et al.: The INTERSPEECH 2010 paralinguistic challenge (2010)
15. Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the 18th ACM international conference on Multimedia, pp. 1459–1462 (2010)
16. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning, pp. 1310–1318 (2013)



Affect-Aware Conversational Agent for Intelligent Tutoring of Students in Nursing Subjects

Moh'd Abuazizeh^(✉), Kristina Yordanova, and Thomas Kirste

University of Rostock, Albert-Einstein-Street 22, 18059 Rostock, Germany
{mohd.abuazizeh,kristina.yordanova,thomas.kirste}@uni-rostock.de

Abstract. In many social professions employees require skills in affect- and situation-aware social interaction. One option for teaching and training such social interaction skills by computer-based training methodology is the use of dialogue simulations. Here, a student interacts with a simulated dialogue partner and the dialogue flow explores specific interaction situations and affectual settings. Conversational agents provide a basic technology for creating such dialogue simulations. However, they usually lack a means for managing affect-related dialogue state. In this paper we propose an approach to integrate affective reasoning into a conversational agent for intelligent tutoring applications in order to improve the agent's ability to recognise dialogue intents, generate emotionally aligned responses, and provide a metric for evaluating student performance.

Keywords: Affective computing · ACT · Probabilistic models · Emotional reasoning · Conversational agents

1 Introduction

In many social professions employees need skills in affect- and situation-aware social interaction. One option for teaching and training such social interaction skills by computer-based training methodology is the use of dialogue simulations. Here, a student interacts with a simulated dialogue partner and the dialogue flow explores specific interaction situations and affectual settings.

Conversational agents have been gaining considerable interest over the course of the last few years. The ability of conversational agents to understand and respond to human language allows many areas such as customer support, tutoring, personal assistance and medical assistance to benefit from these agents [1, 3, 9]. A conversational agent is assessed depending on its ability to simulate a human-like understanding and response. In human interaction, affective state as well as cognitive state play an important role in determining which action to take at a certain point of time. The purpose of our research is to develop an intelligent agent that is able to reason on an affective level to improve the learning experience of a nursing student. An affect-aware conversational agent has

the advantage of simulating believable responses that are emotionally aligned with the conversation taking place. It also improves the ability of the agent to recognise the user's intended actions.

2 State of the Art

Affect aware conversational agents for dialogue simulation are a specific kind of affective tutoring system (ATSs). In general, constructing ATSs is challenging, as it requires cross-disciplinary knowledge domains that involve education, psychology, and computer science [10]. Research on ATS so far has focused on adapting the tutoring strategy to the emotional state of the student. Examples for this are Autotutor [7], which, based on textual and video data, uses dialogue features, body language, and facial features to map the student's emotional state to one of five categories. Another example is Eve [15], an ATS for primary school students, which maps facial expressions to eight emotional categories. Research on Eve confirmed that an affect-aware tutoring system is able to improve student performance in learning.

While in Autotutor and Eve affect is used to adapt the tutoring strategy, our objective is to use an affect aware conversational agent for creating dialogue simulations. Specifically, the agent will have the task of simulating a patient that responds in an affectual way to the dialogue actions of a nursing student.

Affective computational models can be classified in two main categories; discrete emotion theories and continuous theories. Discrete emotion theories propose that emotions can be mapped to a limited set of basic emotions. Continuous theories assume that emotions can be represented as points in a coordinate system with suitable dimensions (such as "arousal" and "valence"). Most conversational agents adopt discrete emotion models, such as Ekman's six basic emotions model [8]. Continuous models provide the means to represent affective state in a continuous dimensional-space. Such representation enables quantitative analysis to compare affective states. In our research, we adopt Affect Control Theory (ACT) [11], a sociological continuous emotion model with three dimensions ("Evaluation", "Potency", and "Activity"). Provided with the textual cues which can be considered as a viable channel to infer student affect [6], ACT delivers an emotional model that is able to capture affective state dynamics in a social interaction.

3 Basic System Design

The goal of our project is to provide nursing students with a digitalised learning platform to help them learn nursing topics. Moreover, the intended intelligent agent should be able to help students learn how to practically cope with different nursing scenarios in a case-based manner. The system should also be able to evaluate the performance of a student and provide the student with feedback. Our dialogue system is subdivided into two main components: Natural Language Processing (NLP) model and the reasoning model. In this paper we focus on the

affective part of the reasoning model and do not cover the details of the NLP model. The probabilistic part of the reasoning model is described in more details in our preliminary study [4].

3.1 Dialogue Representation

Interaction situations in dementia care typically revolve around the caregiver achieving a certain goal, such as making sure a patient is dressed in time to reach an appointment. A given dialogue can be considered a turn-taking sequence of dialogue acts between caregiver and patient. In order to achieve a flexible representation of dialogues that are not fixed to limited set of predefined paths, we employ concepts from the symbolic planning domain. Dialogue state is represented as state variables (“wears left shoe”, “not(wears right shoe)”), the goal is a specific state (“wears(left shoe)”, “wears(right shoe)”), dialogue actions can be represented as precondition-effect pairs (“suggest-put-on-shoe”: precondition = “not(wears left-shoe)”, effect = hint-active(put-on left-shoe)). The formal language we intend to use for representing dialogues based on this paradigm is a dialect of the planning domain definition language (PDDL) [14]. PDDL has been developed to describe transition models in sequential state estimation for dynamic systems with complex discrete state spaces [13]. The resulting models can be used for both disambiguating a student’s textual replies using recursive Bayesian state estimation functionality and for simulating responses.

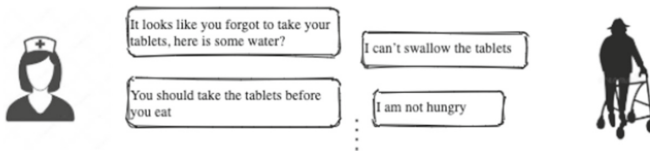


Fig. 1. Dialogue example

Dialogue models are developed based on input from domain experts. The domain experts, cooperation partners from dementia care and nursing sciences, provide example dialogue structures for different training situations (see Fig. 1 for an example), which are then generalised and translated into the PDDL formalism.

3.2 Dialogue Execution

The typical execution cycle is as follows: given a certain dialogue state, the system will expect input from the student (initially written text; natural language input will be considered later). The text will be matched to the nearest student action in the dialogue model by the NLP engine. Execution of the student’s action will then change the simulation state. The resulting intermediate state is the basis for a nondeterministic selection of a patient’s reply action (from those

actions, whose preconditions are met by the intermediate state). The execution of the selected patient response will change the intermediate state to the result state, which forms the basis for the next interaction cycle.

3.3 Observation Model

One central technical challenge is the NLP engine, which needs to map a student's textual input to possible dialogue actions of the PDDL dialogue model. This is ongoing research; currently we are considering to use the RASA chatbot environment [5] for creating this mapping. Specifically, we consider mapping PDDL actions to RASA intents or to slots of RASA forms. The latter method would provide a mechanism for gracefully handling student input that specifies several actions simultaneously.

4 Affective Model

ACT is a sociological theory that assumes humans act to maintain an affective consistency. Moreover, humans seek actions that result in a *transient* affective sentiment which confirms their culturally shared, or *fundamental* affective sentiment. Transient and fundamental sentiments are represented in an evaluation, potency and activity (EPA) dimensional space. The squared euclidean distance between the transient and fundamental sentiments is referred to as *deflection*. This provides a quantitative affect model, i.e. maintaining affective consistency through seeking actions that minimises the deflection. BayesAct [12] is a generalisation of ACT which presents a probabilistic formulation of ACT with the help of partially observable Markov decision process (POMDP). ACT and BayesAct provide a probabilistic quantitative model as well as EPA-datasets [2] which enable the modelling of an affect-aware conversational agent. An affective model will allow the agent to generate emotionally aligned responses and map user input to its corresponding action more accurately by comparing affective values of the input and intended action. Furthermore, ACT provides a metric for student performance evaluation. The conversational agent model provides multiple paths for the student to reach his goal state. The sequence of actions (*policy*) chosen by the student defines the path taken to reach their goal. Student performance can then be rated by evaluating the chosen path. In our research, we adopt the ACT key concept that people act to maintain affective consistency. In this sense the affective state helps in evaluating the student's performance, where for instance action sequences that have smaller deflection values might be considered better than action sequences with high deflection values.

5 Research and Future Plans

Simple case scenarios are modelled with the help of PDDL and will be extended to ensure expert knowledge is modelled accurately in our System. Preliminary

results from our probabilistic model are published in [4]. With the help of our project partners, we are optimising the knowledge representation model to be more similar to real use cases. After developing the baseline model we plan to integrate affective reasoning. Later, the resulting affect-aware agent will be compared to the baseline agent. Then we will evaluate the significance of integrating affective reasoning for our conversational agent.

Acknowledgments. This project is funded by the European Social Fund (ESF) through the Excellence Initiative of the State Mecklenburg-Vorpommern (grant number: ESF/14-BM-A55-0020/19). We thank our domain experts from HS Neubrandenburg and DZNE for their invaluable contribution in providing real world data for developing the dialogue models.

References

1. Amazon lex. <http://aws.amazon.com/lex/>. Accessed 30 Mar 2021
2. Datasets affect control theory. <http://affectcontroltheory.org/resources-for-researchers/data-sets-for-simulation/>. Accessed 30 Mar 2021
3. U report. <https://ureport.in/>. Accessed 30 Mar 2021
4. Abuazizeh, M., Kirste, T., Yordanova, K.: Computational state space model for intelligent tutoring of students in nursing subjects. In: Proceedings of the 13th ACM International Conference on Pervasive Technologies Related to Assistive Environments. PETRA 2020, Association for Computing Machinery, New York (2020). <https://doi.org/10.1145/3389189.3397979>
5. Bocklisch, T., Faulkner, J., Pawlowski, N., Nichol, A.: Rasa: Open source language understanding and dialogue management. CoRR abs/1712.05181 (2017). <http://arxiv.org/abs/1712.05181>
6. D’Mello, S.K., Graesser, A.: Language and discourse are powerful signals of student emotions during tutoring. *IEEE Trans. Learn. Technol.* **5**(4), 304–317 (2012). <https://doi.org/10.1109/TLT.2012.10>
7. D’Mello, S., et al.: Autotutor detects and responds to learners affective and cognitive states. In: Workshop on Emotional and Cognitive Issues at the International Conference Intelligent Tutoring Systems. Montreal, Canada (01 2008)
8. Ekman, P.: An argument for basic emotions. *Cogn. Emotion* **6**(3–4), 169–200 (1992). <https://doi.org/10.1080/02699939208411068>
9. Google: Google assistant. <https://assistant.google.com/>
10. Hasan, M.A., Noor, N.F.M., Rahman, S.S.B.A., Rahman, M.M.: The transition from intelligent to affective tutoring system: a review and open issues. *IEEE Access* **8**, 204612–204638 (2020). <https://doi.org/10.1109/ACCESS.2020.3036990>
11. Heise, D.R.: Understanding events: affect and the construction of social action. Cambridge University Press, Cambridge, New York (1979). <http://www.loc.gov/catdir/enhancements/fy0909/78024177-t.html>
12. Hoey, J., Schröder, T., Alhothali, A.: Bayesian affect control theory, pp. 166–172 (09 2013). <https://doi.org/10.1109/ACII.2013.34>
13. Krüger, F., Nyolt, M., Yordanova, K., Hein, A., Kirste, T.: Computational state space models for activity and intention recognition. Feasibility Study. *PLOS ONE* **9**(11), 1–24 (2014). <https://doi.org/10.1371/journal.pone.0109381>

14. McDermott, D., et al.: PDDL-the planning domain definition language, technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control (1998)
15. Sarrafzadeh, A., Fan, C., Dadgostar, F., Alexander, S., Messom, C.: Frown gives game away: affect sensitive systems for elementary mathematics. In: 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), vol. 1, pp. 13–18 (2004). <https://doi.org/10.1109/ICSMC.2004.1398265>

Extended Reality



ARDNA: A Mobile App Based on Augmented Reality for Supporting Knowledge Exploration in Learning Scenarios

Alessia Genovese^(✉), Federica Marino, Francesco Orciuoli,
and Gennaro Zanfardino

DISA-MIS, Università Degli Studi di Salerno, Via Giovanni Paolo II 132,
84084 Fisciano, SA, Italy
{a.genovese28,f.marino43,g.zanfardino3}@studenti.unisa.it,
forcuoli@unisa.it
<https://www.disa.unisa.it/>

Abstract. Nowadays, Augmented Reality is broadly integrated across multiple environments, but the smart learning area is still polarized on visualizing articulated 3D models. Our proposal consists of a new approach to collect and organize notes, that takes full advantage of the AR platform providing an useful learning tool for students. Augmented Reality Data Navigation App (ARDNA) manages to display on screen notes, images and 3D graphs floating alongside traditional study material to support students during their learning acquisition process. The app has been implemented through the use of Unity and Vuforia Augmented Reality SDK. A minimum viable product has been handed to a small group of students and their experiences have been monitored and evaluated; the results obtained support the founding statement of the app with measurable improvement for smarter studying.

Keywords: Augmented reality · Technology enhanced learning · Mobile computing

1 Introduction and Motivations

Given the widespread use of mobile technologies in diverse fields, it is increasingly common, especially for students, to have a great number of notes hard to merge, given that they are divided between physical media and digital contents.

Augmented Reality (AR) is widely used in the educational field being an efficient and engaging support for students, helping them to better remember newly learned information [1]. The main advantage of AR in the context of AR Data Navigation App (ARDNA) is its ability to complement reality [2], enhancing the learning process on paper by adding digital information floating on the study material.

ARDNA provides a platform capable of merging paper and digital contents, enabling 3D navigation and visualization of a knowledge base. A student may start by handling just some notes regarding certain topics, in a way that feels more compact and intuitive to navigate than physical media. The goal of ARDNA is to improve the quality of student learning in their study sessions, avoiding dispersion of contents and encouraging focus.

The remaining part of the work is structured as follows: Sect. 2 presents a brief overview of Related Work. Section 3 describes the ARDNA conceptual Model and its main functionalities. Section 4 explains the app design process and tools used in the implementation phase. Section 5 describes the experiment carried out to test the application and the obtained outcomes. Section 6 provides final remarks and insights on possible future improvements.

2 Related Work

In the last decade the use of AR technology significantly spread in every area of our society and strongly impacted teaching and learning methodologies [3]. Learning has received benefits from AR through many software solutions in various academic fields (e.g. human anatomy, chemistry, astronomy, etc.) [4]. For example, Chemistry students can benefit from Chemist which uses 3D models to carry out chemical experiences and observe the reactions, exercising several tools and reagents. In astronomy, Spacecraft 3D may help students explore the solar system and interact with spacecraft models. In medical disciplines, AR is broadly used for interns' learning and training, avoiding dangerous consequences if any mistake occurs [5]. Students can take advantage of apps like Human Anatomy Atlas, to visualize 3D models exploring the human body and learning how it works, while training with AccuVein helps medical interns to locate patients' veins for injections.

This brief analysis exhibits that these AR software solutions are subject specific, while our app instead aims to display and organize notes meaningfully, in order to help students in acquiring and fixing concepts in any learning field. Moreover, taking into account existing applications whose purpose is to leave notes on real-world objects, like StickyNotesAR and Google's AR 'Save button', one main difference with the proposed solution is that none of them exploited more than some elementary AR advantages.

3 The ARDNA Model

Nowadays, students study from various sources (e.g. books, notes, slides, PDFs). This fragmentation leads to a higher cognitive load, because students have to search for information, integrate it with other media, re-read concepts and so on.

ARDNA tries to provide a solution to this issue, by supporting the learning process without adding complex cognitive processes. The app can collect all user's contents, enriching the physical media with digital data to create a neater learning environment and facilitate the knowledge acquisition process. The app

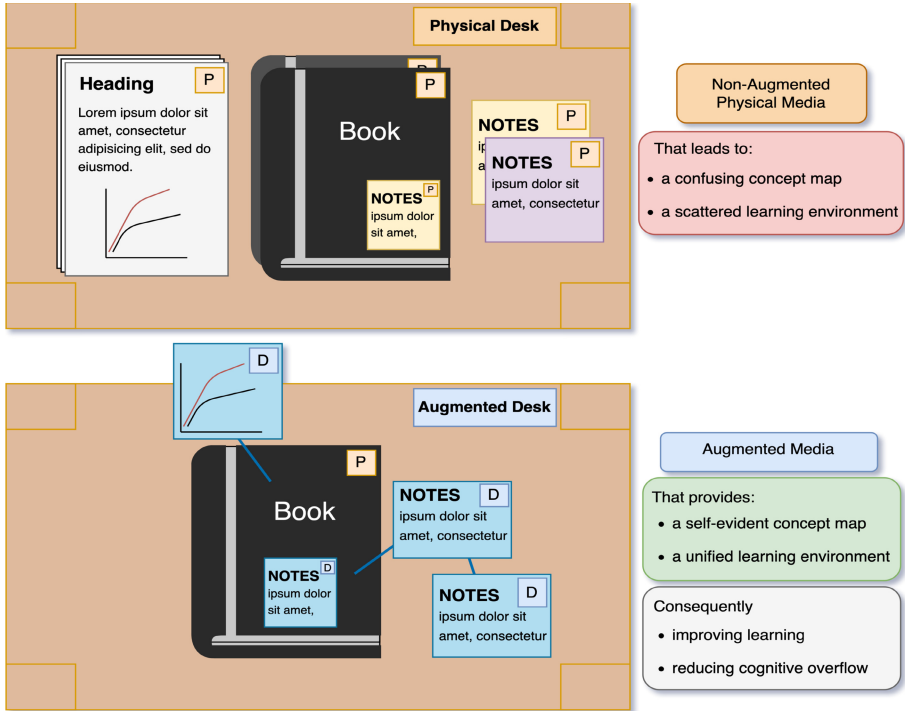


Fig. 1. Diagram representing problem (top half) and solution (bottom half) scenarios alongside main differences.

has been designed with usability in mind and addresses both tech-savvy users and lesser ones from different education levels.

Exploiting the AR advantages, the app uses books and notebooks' covers as markers, which are visual cues able to trigger the display of the virtual information (Fig. 1). In this way, the user can store information about a specific topic, update their progress when a book chapter is finished, save important definitions, and retrieve contents in a faster and easier way.

The core functionalities of the app and the interactions the user may perform using it are explained in the following.

Books and Notebooks Cover Recognition. The user can create subjects to separate notes either by typing the title manually or by enabling the camera and opening the OCR (Optical Character Reader) Module. Then, it will be possible to attach the subject to a cover.

Notes Uploading and Opening. The user can add a new note (text, 2D&3D graphs or images), attaching it to subjects in order to incrementally enhance chunks of their study material. Notes can be nested and the app will show them using predefined layouts to organize contents in an effective way, displaying first

recent and relevant notes. Moreover, it is possible to fix notes on screen for better consultation.

3D Data Visualization and Interaction. An important feature of ARDNA is the 3D data visualization of a plot. The graphs are placed next to the physical media surrounded by panels that outline their features and meaning. Furthermore, the student can interact with the data representation through buttons to toggle decisional attributes (Fig. 2). This will allow the user to better understand the data distribution, and to find out information about it that may be hard to find in a 2D representation [7].

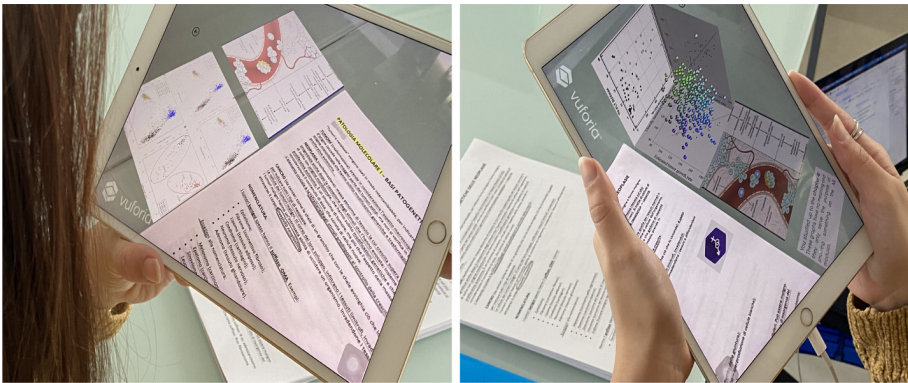


Fig. 2. Student navigating notes.

These functions allow the user to have a complete and tidy collection of notes, to free physical space on their desk and on their notebooks. The students can earn both in terms of time and neater study materials, with the possibility to access them wherever and whenever they are.

4 Implementation

During the discovery and design phase of the app, Personas expressing the categories of potential users have been defined (i.e. high school and undergraduate students), but in this paper only one will be presented [8, 9]. This helped to better outline problem and solution scenarios that the app could address and to turn the needs elicitation into a suitable requirements definition. Using mobile AR technologies, students are more interested in their study, therefore able to concentrate and perform better [10]. AR technology is useful to represent complex concepts difficult to grasp. In fact, the application guides the students through their study materials promptly displaying data, through explanations and visual aid collected from the teacher or third-party material. ARDNA combines the advantages of the AR to enhance learning and the students' need of having a

neat collection of notes, graphs and data, easily integrating contents from books and notebooks.

ARDNA has been developed using Unity and Vuforia. The former provides tools to develop experiences in both 2D and 3D, and the supports C# scripting. The latter is a cross-platform augmented reality SDK, that enables the creation of Augmented Reality applications (e.g. Virtual Buttons with effects on 3D Graphs - Fig. 3). Given its cross compatibility with both MacOS and Windows, the app can be built for both iOS and Android Devices, making it a viable technology adoption choice among heterogeneous academic groups. The storing and management of knowledge in ARDNA has been handled through MongoDB, a cross-platform NoSQL document-oriented database program that fits the unstructured nature of the data the student may collect.

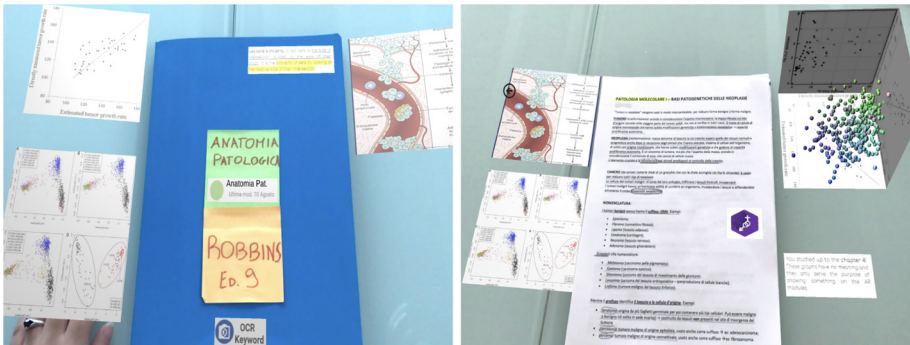


Fig. 3. Subject and notes attached to a cover, on the right image the sex virtual button to toggle a decisional attribute.

5 Experimentation and Evaluation

By considering the scenarios described in Sect. 3, the early app prototype has been provided to a small sample of academic students (from the fields of Medicine, Economics, and Computer Science) due to time constraints. This first experimentation phase is aimed to evaluate possible improvements in students' engagement and time required for data retrieval during study sessions.

Students were asked to answer a questionnaire to evaluate the user experience (UX) and, if found, enhancements while consulting study material through the app. The main questions were about (i) the speed of note retrieval, (ii) the app usability for content navigation, (iii) the ability to concentrate and comprehend during learning sessions. A small sample of the questionnaire and the corresponding results, is reported in Table 1.

This brief experimentation has led us to assume that the app may be actually a useful tool for supporting students in their learning, reducing cognitive overload

Table 1. Questionnaire sample

Question	Answer
Was it faster to find the needed study material?	Yes: 70%
Was it easy to become familiar with the use of the app?	Yes: 75%
Were you able to stay more focused while using the app?	Yes: 80%

[6] and favouring engagement. Nevertheless, the activity of uploading all contents together is quite time-consuming. Thus, for an optimal app functioning, students should be incentivised to study steadily, to keep their notes up-to-date in the system.

6 Final Remarks

In this paper we proposed an Augmented Reality application to support learning in the academic environment, encouraged by scientifically valid and diverse evidence that AR fully enhances learning. The purpose of the app is to manage the great amount of notes and study materials acquired over time. After the experimentation phase, we noticed improvements in terms of notes collection and organization, time saving, and an enhancement of students' performance.

Moreover, future improvements will concern: (i) the implementation of a networking module, to facilitate collaborative learning in hybrid learning environments [11]; (ii) the app could be adapted to take full advantage of the accessibility tools provided by the mobile operating systems on top of which the app is built (e.g. voice over for better reading, and alternative pointers for easier use from motion impaired users [12]); (iii) the use of the app through Microsoft HoloLens, to make movements more agile while studying; (iv) integrating the latest Vuforia release equipped with LiDAR sensors support for enhanced target scanning features. These technical improvements may help towards building a better and seamless experience for users, avoid discouragements produced by limits of their devices.

References

1. Garzón, J., Kinshuk, Baldiris, S., Gutiérrez, J., Pavón, J.: How do pedagogical approaches affect the impact of augmented reality on education? A meta-analysis and research synthesis. *Educ. Res. Rev.* **31**, 100334 (2020) ISSN 1747–938X
2. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. *IEEE Comput. Graph. Appl.* 21(6), 34–47 (2001). <https://doi.org/10.1109/38.963459>
3. Wu, H.K., Lee, S.W.Y., Chang, H.Y., Liang, J.C.: Current status, opportunities and challenges of augmented reality in education. *Comput. Educ.* **62**, 41–49 (2013) ISSN 0360–1315

4. Lee, K.: Augmented Reality in Education and Training. *Tech Trends*. **56**, 13–21 (2012). <https://doi.org/10.1007/s11528-012-0559-3>
5. Kamphuis, C., Barsom, E., Schijven, M., Christoph, N.: Augmented reality in medical education? *Perspect. Med. Educ.* **3**(4), 300–311 (2014). <https://doi.org/10.1007/s40037-013-0107-7>
6. Akçayır, M., Akçayır, G.: Advantages and challenges associated with augmented reality for education: a systematic review of the literature. *Educ. Res. Rev.* **20**, 1–11 (2017) ISSN 1747–938X
7. Hirve, S.A., Kunjir, A., Shaikh, B., Shah, K.: An approach towards data visualization based on AR principles. In: 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, pp. 128–133 (2017) <https://doi.org/10.1109/ICBDACI.2017.8070822>
8. Marsden, N., Pröbster, M.: personas and identity: looking at multiple identities to inform the construction of personas. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI 2019), Association for Computing Machinery, New York, Paper 335, pp. 1–14 (2019)
9. Manthey, R., et al.: Visual system examination using synthetic scenarios. In: Karwowski, W., Ahram, T. (eds.) *IHSI 2019. AISC*, vol. 903, pp. 418–422. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11051-2_63
10. Chiang, T., Yang, S., Hwang, G.-J.: An augmented reality-based mobile learning system to improve students' learning achievements and motivations in natural science inquiry activities. *Educ. Technol. Soc.* **17**, 352–365 (2014)
11. Dunleavy, M., Dede, C., Mitchell, R.: Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning. *J. Sci. Educ. Technol.* **18**, 7–22 (2009)
12. Quintero, J., Baldiris, S., Rubira, R., Cerón, J., Velez, G.: Augmented Reality in Educational Inclusion. a Systematic Review on the Last Decade. *Front. Psychol.* **10**, 1835 (2019) <https://doi.org/10.3389/fpsyg.2019.01835>



Extraction of 3D Pose in Video for Building Virtual Learning Avatars

Kodjine Dare, Hamdi Ben Abdessalem^(✉), and Claude Frasson

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal
H3C 3J7, Canada
{kodjine.dare, hamdi.ben.abdessalem}@umontreal.ca,
frasson@iro.umontreal.ca

Abstract. From an image of a person, we can easily guess the 3D coordinates of the body parts. This is because we have acquired a 3D mental model from observing humans and interacting with them. This capacity easily achievable for humans is not systematic when it comes to computers. In this paper, we describe an approach that aims at estimating poses from video with the objective of reproducing the observed movements by a virtual avatar. We propose the fragmentation of submitted videos into series of RGB frames to process individually. We aim two main objectives in our work. First, we achieve the extraction of initial 2D joints coordinates using a method that predicts joint locations by part affinities (PAFs). Then we infer 3D joints coordinates based on a human full 3D mesh reconstruction approach supplemented by the previously estimated 2D coordinates. Secondly, we explore the reconstruction of a virtual avatar using the extracted 3D coordinates with the prospect to transfer human movements towards the animated avatar. This would allow to extract the behavioral dynamics of a human, allowing to detect some health problems, for instance in Alzheimer. Our approach consists of multiple subsequent stages that show better results in the estimation and extraction than similar solution due to this supplement of 2D coordinates. With the final extracted coordinates, we apply a transfer of the positions (per frame) to the skeleton of a virtual avatar in order to reproduce the movements extracted from the video.

Keywords: Machine behavior · Behavior detection · Visualization · 3D pose estimation · Virtual avatar

1 Introduction

Considering a video that displays a person performing a task or even just a movement (walking, running, dancing), human can easily identify the different body parts location and orientation of the person also known as pose in the video. The analysis of this simple human process introduces the interrogation regarding the capacity of computers to automatically detect human body pose.

It is in this same vein that the purpose of this article revolves around two focal points. Firstly, we articulate our work around this task known as Human pose estimation that aims to automatically locate the human body parts from images or videos in order to

extract information such as human poses. This human pose estimation is also known as **keypoints detection**. The extraction of these poses provides a collection of data that will be relevant for the next stage. Then we address the second objective of our work that consists in using the extracted body part location for the reproduction of the behavior with an animated avatar. We define the animation of an avatar as a sequence of poses extracted in a video. This objective aims to allow to transfer human movements towards an animated avatar to highlight the behavioral dynamics of the human from the keypoints that have been extracted.

The achievement of these two objectives can lead to multiple implications in both applied and theoretical research. In the context of this research, we focus on the application for Alzheimer behavior. Alzheimer is an irreversible, progressive brain disorder that slowly destroys memory and thinking skills and affects behavior. Alzheimer patients can have sometimes specific behavior (walking, equilibrium,...) which could be observed by video camera at different times of the day, extracted, and rebuilt in a virtual avatar. This avatar would serve as a training model to educate the medical staff to recognize an episode of Alzheimer patients.

Throughout this paper, we put special emphasis on the extraction of human pose estimation. We use in the rest of this paper the term pose and keypoints interchangeably. The remaining of this paper is organized as follows. In Sect. 2 we present related works. In Sect. 3 we present our approach and discuss the results in Sect. 4.

2 Related Work

To address human pose estimation, several approaches have been proposed, we focus our discussion on relevant 2D and 3D human pose estimation.

Human 2D Pose Estimation: There are single person pose estimation methods and multi-person pose estimation methods. The single person methods are divided between the heatmap approach that chooses the locations with the highest heat values as the keypoints [1] and the direct regression approach which utilize the output feature maps to regress keypoints directly [2]. Multi-person methods host two categories: top-down approaches which consist of applying a person detector and then running a pose estimation algorithm per every detected person [3, 4] and bottom-up approaches which first step is to locate all the keypoints in an image and the second step is to group them according to the person they belong to [6].

3D Pose Estimation: Agarwal and Triggs [7] rely on silhouette feature while Zhou et al. [8] proceed by manual interaction from user. With deep learning, direct regression approaches integrate the SMPL [9] to train models to directly infer the SMPL parameters. These methods infer the 3D pose and shape based on: RGB image as suggested by Kanazawa et al. [10], RGB image and 2D keypoints [11], keypoints and silhouettes [12], or keypoints and body part segmentations [13]. Some methods extend their work to estimate 3D pose from **video**. Among these methods, the vast majority rely on elaborate environment which capture sequences on multiple angles. Due to the perspective of our work, we focus on the approaches that deal with video captured by regular cameras. Some methods obtain accurate shapes and textures of clothing by pre-capturing the actors

and making use of silhouettes [14]. While these approaches obtain satisfying shape, reliance on the pre-scan and silhouettes restricts these approaches to videos obtained in an interactive and controlled environment. Therefore, we propose an approach that rely on RGB image supplemented by 2D keypoints.

3 Proposed Approach

The approach that we propose estimates 3D keypoints through a 2-stages estimation of frames of the input video and a transfer of results towards reconstruction. We propose the fragmentation of the video into series of RGB frames (see Fig. 1).

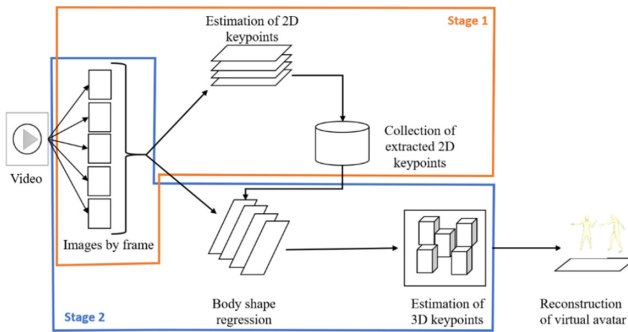


Fig. 1. Architecture of the proposed approach

First Stage: It insures the 2D pose estimation using the Realtime Multi-Person 2D Pose Estimation [6] to estimate the human body 2D keypoints in the submitted frames.

This method presents a bottom-up approach for estimation of multi-person pose in RGB image and produces, as output, the 2D locations of anatomical keypoints for each person in the image, without using person detector. The model defines a network architecture that iteratively predicts affinity fields that encode part-to-part association and detection confidence maps. The network is split into two main sections: a first one that predicts the confidence maps, and the second predicts the affinity fields. Each section is an iterative prediction architecture, that finetunes the predictions throughout multiple stages, with intermediate supervision at each stage.

Once the estimation completed, we proceed to the extraction of the estimated keypoints. The result of such operation is a collection of keypoints per frame.

Second Stage: We use End-to-end Recovery of Human Shape and Pose [10] to iterate the feature of every frame to predict the human body. The method infers a full 3D mesh of a human body directly from an RGB image of a human. With that approach, the 3D mesh of a human body is encoded using SMPL which generates human bodies into shape with regards to the variation in height, weight and body proportion, and the deformation of the surface due to the movements.

While this 3D estimation method presents a great approach to infer the human body shape from an RGB image, this method is introduced to work on input images of a particular scale and quality. To address that limitation, we strengthen the prediction by using the output of the first stage. This helps to estimate the final 3D keypoints that fits as much as possible the size of individuals in every frame. This process results in the extraction and storage of 3D keypoints that will be further used in a virtual environment to reconstruct an avatar which keypoints would correspond to the extracted keypoints. The overall idea behind the presented pipeline is to take a video as input and produce a virtual avatar that replicates the movement observed in the submitted video.

4 Results and Discussion

4.1 2D Estimation

We conducted the experiment with the objective of estimating accurately human 2D pose from single RGB image regardless of the complexity of pose described by the image. In order to determine how adequate the method was for the purpose of our work, we have analyzed the results obtained and compared with the performance of similar methods. The methods compared were AlphaPose [4], PersonLab [15], Mask R-CNN [3], Deepcut [5], Stacked Hourglass Network [1]. While DeepCut achieved average performance, the qualitative results were by far the poorest. Stacked Hourglass Network and Mask R-CNN presents improved results on the qualitative and quantitative level compared to the previous one but did not deliver optimal performance. AlphaPose, PersonLab and our adopted approach gave the most consistent results and high amelioration.

In Fig. 2 we can observe the estimation difference between some of the methods we have tested and evaluated. The first column represents the estimation of the method we opted for, the second column the estimation of AlphaPose, the third one is the performance of PersonLab and the last represent the performance on Mask R-CNN.

The performance of AlphaPose and the adopted approach are not so far from each other, but we observe a rapid decay of the estimation when there is an occlusion of the human pixels with the background pixels.

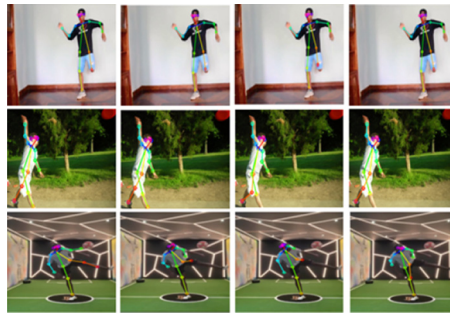


Fig. 2. Estimation across set of methods

4.2 3D Estimation

The proposed approach was evaluated on different format of images. The objective that drove the assessment of the method was to find out the scope of the improvement of the results in comparison to the initial approach. Hence, we first measured the achievement of the model as suggested by Kanazawa et al. [10] and the model based of 2D keypoints the same set of images to see if our approach gives better results. We have not noticed a significant variation in the mesh inference, hence the 3D keypoints coordinates. This insignificant variation could be explained by the fact that regardless of the presence of 2Dkeypoint given that the initial approach performed best on images of with such scale, providing a base 2D keypoints to help determine the location of individual does not change much the outcome.

We have also evaluated the results of our approach on images of different scale to compare the mesh reconstruction. We represent the inference without the context provided by prior 2D estimation by a mesh in magenta, and in blue the inference with prior 2D keypoints.

In Fig. 3, we can observe the marge of variation performance for slight improvement to obvious differences. The first row shows how the most difference, without the initial context provided by the 2D keypoints, the model is not able to infer the 3D keypoints. The last row shows how well the model performs when it comes to reconstruct a limb that is out of vision range.



Fig. 3. Performance on images of different scale and bounding box

This finalized approach will be used to reproduce movement by avatar. These avatars will be leveraged in the context of education systems. The idea behind these systems will be to allow learners to interact with avatars that display some behaviors and visualize the response of their interactions.

5 Conclusion

In this paper, we proposed an approach to animate a virtual avatar based on 3D keypoints estimated from a video. Our proposed approach divides the video into series of RGB images, and for each image we suggested to perform a 2-stages estimation. During the

first stage, we have estimated 2D keypoints using a 2D pose estimation and we have proceeded to the extraction of the estimated keypoints. In the second stage, we have used a 3D mesh reconstruction method to infer 3D keypoints using the output of the first stage. This process results in the extraction and storage of a sequence of 3D keypoints. We use this sequence to reproduce the movement from the video on a virtual avatar. With the proposed solution, we were successfully able to reproduce the video behavior on an avatar in a virtual environment. This approach could have multiple applications, but in the context of this research, we focus on the application for Alzheimer's disease. In fact, such solution is devoted to help in the creation of a system by reproducing Alzheimer's disease patient's behavior on a virtual avatar to educate medical staff in the interaction with them.

Acknowledgment. We acknowledge NSERC-CRD (National Science and Engineering Research Council Cooperative Research Development), Prompt, and BMU (Beam Me Up) for funding this work.

References

1. Newell, A., Yang, K., Deng, J.: Stacked Hourglass Networks for Human Pose Estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
2. Toshev, A., Szegedy, C.: DeepPose: human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on CVPR, pp. 1653–1660. IEEE Computer Society, Las Vegas-USA (2014)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV, pp. 2980–2988. IEEE Computer Society, Venice-Italy (2017)
4. Fang, H., Xie, S., Tai, Y.-W., Lu, C.: RMPE: regional multi-person pose estimation. In: ICCV, pp. 2353–2362. Venice-Italy (2017)
5. Pishchulin, L., et al.: Deepcut: Joint subset partition and labeling for multi person pose estimation. In: IEEE Conference on CVPR, pp. 4929–4937. USA (2016)
6. Cao, Z., Simon, T., Wei, S., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on CVPR, pp. 7291–7299. Hawaii (2017)
7. Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. TPAMI **28**(1), 44–58 (2006)
8. Zhou, S., Fu, H., Liu, L., Cohen-Or, D., Han, X.: Parametric reshaping of human bodies in images. In: SIGGRAPH '10, pp. 1–10. Association for Computing Machinery, USA (2010)
9. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: a skinned multi-person linear model. ACM Trans. Graphics **34**(6), 1–16 (2015)
10. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: Proceedings of the IEEE Conference on CVPR, pp. 7122–7131. IEEE Computer Society, Salt Lake City-USA (2018)
11. Tung, H.-Y., Tung, H.-W., Yumer, E., Fragkiadaki, K.: Self-supervised learning of motion capture. In: Proceedings of the 31st International Conference on NIPS, pp. 5242–5252. Curran Associates Inc, Long Beach-USA (2017)
12. Pavlakos, G., Zhu, L., Zhou, X., Daniilidis, K.: Learning to estimate 3D human pose and shape from a single-color image. In: Proceedings of the IEEE Conference on CVPR, pp. 459–468. IEEE Computer Society, Salt Lake City-USA (2018)

13. Omran, M., Lassner, C., Pons-Moll, G., Gehler, P., Schiele, B.: Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In: International Conference on 3DV, pp. 484–494. IEEE Computer Society, Verona-Italy (2018)
14. Alldieck, T., Magnor, M., Xu, W., Theobalt, C., Pons-Moll, G.: Video based reconstruction of 3d people model. In: Proceedings of the IEEE Conference on CVPR, pp. 8387–8397. IEEE Computer Society, Salt Lake City-USA (2018)
15. Papandreou, G., Zhu, T., Chen, L.-C., Gidaris, S., Tompson, J., Murphy, K.: PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 282–299. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_17



A Non-immersive Virtual Reality Application for Children with Autism Spectrum Disorder

Muhamad Irfan Rosli^(✉) , Zarina Che Embi , and Junaidi Abdullah 

Multimedia University, Cyberjaya, Selangor, Malaysia

Abstract. Children with autism spectrum disorder (ASD) are progressively acquainted with digital technologies in their training and diagnosis. Due to their growing accessibility and practicality, the potential should not be overlooked. Virtual Reality (VR) has become a noticeable tool to help children with ASD in their social training due to its competitive potential that provides extensive interaction, economical cost, safe environment and an enjoyable experience. The implementation of VR is often associated with heavy usage of head-mounted display (HMD) where through it, the immersive experience can be achieved. However this may not be applicable to children with ASD. Side effects such as motion sickness and claustrophobia may affect their training and performance. Therefore, this paper aims to present the development of non-immersive VR application with a serious game concept with inclusion of analytics, where progresses can be calculated, compared and observed. Upon post evaluation by experts, this game is improved and holds a larger potential to achieve its objectives.

Keywords: ASD · Non-immersive · Virtual reality · Serious game · Analytics

1 Introduction

Virtual Reality (VR) is a three-dimensional (3D), computer simulated and generated environment, which includes exploration and interaction by a user. There are two common types of VR; immersive and non-immersive. Immersive VR (IVR) is achievable with the application of head-mounted display (HMD) and input readers, whereas non-immersive VR (NIVR) is experienced from a desktop display. Non-immersive environments allow users to interact with contents by using multiple devices (mouse, keyboard or microphone). It has lower immersiveness in order to increase comfort for users with less tolerant towards immersive VR. Both type of VR provides a realistic environment that can be controlled, manipulated and interacted with. This allows the subjects, who have developmental challenges, to receive support in sharpening their everyday skills.

Autism Spectrum Disorder (ASD) is a developmental challenge that involves difficulties in social interaction, speech and nonverbal communication and restricted or repetitive behaviours (Copeland 2018). The term spectrum is being used due to variety of type and severity of symptoms in each autistic person regardless of their ethnic, economic status, intelligence level. Children who are diagnosed with ASD require support in order to build their social intelligence. Research conducted by National Research

Council (2001) documents that early interventions and diagnosis for autism shows significant long-term positive effects on symptoms and later skills of children. Therefore, early interventions are essential for treatment in providing a better life quality for children with ASD. As an alternative that provides realistic virtual context and experiences for learning and assessment, VR holds the potential to assist children with ASD in their social training.

The purpose of this paper is to exhibit the development of the NIVR intervention educational game for children with ASD. Section 2 provides the background studies of the initial game design. Section 3 reports on the methodology in enquiring the experts' feedbacks to improve the game. Section 4 explains the proposed solution and lastly Sect. 5 presents a conclusion of this paper.

2 Background Studies

Evidence shows that the implementation of VR has the potential to help ASD children. Schwarze et al. (2019) report that with virtual setting, people with ASD shall be able to recognise emotion better. They concluded that in order to create a solid immersive experience for children with ASD, there are four important factors that should be considered. Firstly is to have detailed design of the VR environment which brings comfort. Secondly, VR learning environment should be gamified to create a motivation to learn specific skills. Thirdly, children with ASD prefer to be in virtual environment when learning emotion recognition. Lastly, VR environment should have an enhanced immersion to allow escape from reality. Halabi et al. (2017) further expand the immersive element by adding verbal or nonverbal interactive system, which is proven to be efficient in improving communication skills of children with ASD.

Another application of IVR is the implementation of real-life situation as a simulation. Dixon et al. (2019) suggest an IVR safety training can teach children with ASD necessary pedestrian skill. Dixon et al. (2019) concluded that, the usage of HMD allows larger vision field which enhances immersiveness. Bradley and Newbutt (2018) also report the potential of IVR emphasising on the HMD. However, they discovered almost 50% of the studies mention cybersickness as the main negative effect of using the HMD. This may be due to the heightened senses of people with ASD and their sensitivity to inputs.

Many researchers focus on the level of immersion in order to produce better outcome (Parsons 2016). IVR is implemented without considering the suitability of ASD children towards it (Newbutt et al. 2020). Motion sickness, cyber sickness or claustrophobia should not be overlooked. Due to more focuses are given to IVR, there is a big gap in the research of NIVR for ASD children. There is no data analytics integrated within a system or framework; it is done externally. In studies with serious game, the in-game analytics is limited to the implementation of the scoring system. Many analytics implement collecting feedbacks solely from educators, excluding the ASD participants. By combining elements aforementioned, a VR application framework is developed. Table 1 shows the main features of the proposed learning game.

Table 1. Main features of proposed VR learning game

Features	Description
Gamified social skill training	Serious game (educational game) based on everyday task
Emotion recognition	Help ASD children to recognise common emotions
Verbal and non-verbal interactive system	Add speech recognition in the game to promote two-way communication between user and other characters
Real-life scenario	Real life inspired environment for ASD children. Free roaming allows user to exercise creativity
NIVR environment	To avoid motion sickness and discomfort
Analytics	Scoring system and time taken to finish the mission

3 Methodology

A prototype was developed from the inputs acquired from literature reviews. An evaluation research is conducted to further improve the game. Four experts that consist of two therapists, a special need public school teacher and a teacher from private autism centre were interviewed and played the learning game. Semi structured interviews were conducted, where the objective of the interview is to understand the current situation in ASD treatment, a brief explanation of ASD interventions and feedbacks on the learning game.

The interview contains 31 questions divided into three sections according to the objectives mentioned earlier and the inputs are collected qualitatively. In section A, experts were asked about their background and experiences with ASD children. This section is to identify the local demographic of children with ASD such as their age range, common autism level and also diagnosis method used. In section B, intervention programs that are being implemented in their centre were inquired. In the last part of the interview, all four experts were given a chance to play the game as a part of validation process. During the gameplay, the purpose of every level, the tasks of each non-playable character (NPC) and the technical aspect that is running in the background were explained. The experts gave their constructive feedbacks focusing on the game suitability that includes level of difficulty, playability for correct age group and the relevance towards ASD children. The data obtained are analysed to further improve the game design. Based on the experts' feedbacks, four main features in Table 2 have been added for improvement of the learning game experiences.

Table 2. Features derived from experts’ feedback

Features	Comments
Background music	To allow adjustment at the beginning of the game for acoustic comfort
Level difficulty	Lower NPC difficulty
Encourage verbal communication	Add two-way interactions between player and NPC
Analytics	Parameter such as travel distance is good data that can be observed and analysed to see progress of ASD children

4 Proposed Solution

A learning game prototype is developed using Unity 2017 engine based on literature reviews. The three important themes for the game development are to improve social skills, enhance emotional understanding and train the concentration level of ASD children. From here, the themes are being presented in three levels. The game also implemented four proposed characteristics of VR framework and these characteristics are Feedback, Non-linear gameplay, Goal directed learning and Increasing difficulty level.

Engaging elements such as visual and naturalistic conversation are the key components in providing more immersive feedback to player. Hence, speech recognition and text-to-speech (TTS) interactions are implemented in the game. Additionally, to promote motivation and instilling a sense of control, player is allowed to free roam in the virtual environment. A non-linear gameplay is implemented to allow player to be in control of their actions and a goal directed learning is implemented to help player focus and motivate them to finish the game. Lastly, the learning game has an increasing difficulty level to create challenges to help player in scaffolding their skills.

The prototype is further improved with the feedback data acquired from the experts’ interviews. As mentioned before, the game has three main levels. Player will be in the main menu level when the game starts and in here player is allowed to familiarise themselves with the game environment and adjust the volume settings. A single building is placed as an attraction point and at the entry; player needs to say “Start” to proceed to the next level, which is the Shopping Level (Fig. 1).



(a) Shopping building



(b) Exit gate



(c) Settings Box

Fig. 1. Main menu level

The Shopping Level is an implementation of a real life scenario, which involves in purchasing items in a supermarket. In this level, the player needs to find three items namely a bowl, a book and a toaster and buy them on behalf of the mother NPC. The player may free roam inside the supermarket, interacts with 3D NPCs verbally and receives feedback from them. Interactions with two specific NPC in the supermarket will trigger mini levels; concentration game and emotion game. Each mini level has two stages which has an increased difficulty or different variance of execution when moving to the next stage. The emotion game has two stages which are differentiated by execution. The first stage is completed by having player to speak out the emotions according to the situation that is being verbally explained by an NPC. The second stage requires user to carry the emotion box, and passes it to the correct emotion NPC (Fig. 2).

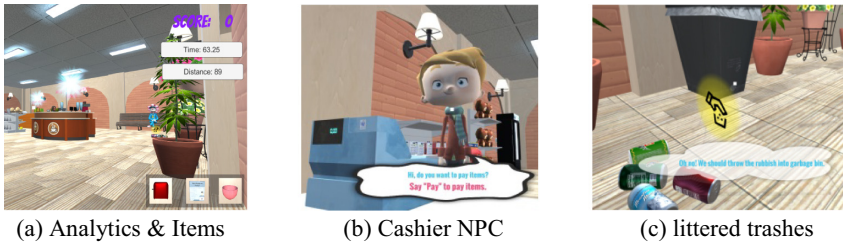


Fig. 2. The screenshots of Shopping level

The concentration game has an increased in difficulty for its stages. The first stage requires player to follow the correct object for 3 seconds. The second stage uses the same mechanisms, except the speed of the NPC will be slightly faster than previous stage. After completing the mini levels, player may continue the mission in the Shopping Level where when completed, player is required to go to the cashier NPC and purchase the items by selecting the correct amount of money. The cashier NPC will provide instruction to the player regarding the payment. The mother NPC will be highlighted after the player had finished the payment process and the training will be completed after the player interacted with the mother NPC. There is also a task to promote social responsibility, where the player can pick up littered trashes on the floor and put them in the trash bin (Fig. 3).



Fig. 3. The screenshots show the concentration level in (a) and (b) and the emotion level in (c) and (d)

Another important feature of this game is analytics, which allows player data to be collected and be analysed. Three types of data will be recorded and shall be integrated in the game. These data are accumulated score points, time taken for task completion and distance traveled. The Scoring system is a reward system and an analysis of individual performance in the game. By completing a task, the player can accumulate score points while completing the mission. Interacting with NPCs will also give players additional score points. Second data is time taken to complete a mission. This data records the reaction of ASD children when executing a task in order to see their efficiency in handling a situation. Each level has its own timer set in the background. The timer is included in all levels separately. Lastly, the third data is the distance traveled during gameplay. Distance traveled is a key component to analyse the behaviour of children with ASD. From here, data taken can reflect how active an ASD child move during the training session, which may provide more substantial information on their behaviour. Recently, the immersiveness in VR for ASD is further explored with a comparison of real versus virtual environment. Simões et al. (2020) report that interpersonal distance (IPD) regulation of ASD individuals in real world gave the same outcome in IVR. The interaction of ASD individuals in IVR mirrors a similar interaction in the real world as mentioned before by Parsons (2016).

In Unity engine, there is no simple method to count character steps as the game movement is generated from the coordinates of the first person camera view. To acquire an approximation, the value of the 3D planes; plane-x, plane-y and plane-z are calculated to generate an integer number that will be used as number of steps taken by the player. This is not an accurate representation of steps taken, but it has ample information to provide a numerical change in plane value when the player moves. The player is moving forward and backward along the plane-z and moves to the sides along the plane-x. The movement is relative with the direction the player is facing. As there are changes in plane values, distance increases by one point. This is possible to execute as plane-y is a constant variable as the main character has a fixed vertical standing position during the free roaming in the shopping level. However the calculation might be affected when the player is jumping around. But this data will be accepted as this can be also justified as the player movement.

5 Conclusion

VR is recognised and considered as one of the potential solutions to help children with ASD in their social training. However, there are arguments and concerns in creating good immersive experiences for children with ASD. Motion sickness is often dismissed in many studies and is often generalised that every child with ASD is comfortable wearing a HMD. We conclude that a learning game that fulfils the characteristic of NIVR which is near-reality, 3D environments that a person can interact and explore in the virtual environment should be developed. We hope that the game shall provide a good early intervention for children with ASD based on the implementation of real-life scenario and also safe environment for social training. We also plan to evaluate the game using Kirkpatrick Evaluation Model to validate its effectiveness in the future.

The implementation of data analytics can be used to see the trend of ASD children in a group. Data comparison can be constructed to observe the level of progress of the group members when the game is played simultaneously on different devices, this opens the potential of the application to be used as a diagnostic tool in the future.



Acknowledgement. This study is funded by Fundamental Research Grant Scheme (FRGS) of Ministry of Higher Education, Malaysia (FRGS/1/2019/ICT04/MMU/03/12).

References

- Bradley, R., Newbutt, N.: Autism and virtual reality head-mounted displays: a state of the art systematic review. *J. Enabling Technol.* **12**, 101–113 (2018)
- Copeland, J.N.: What is autism spectrum disorder? *Am. Psychiatr. Assoc.* <https://www.psychiatry.org/patients-families/autism/what-is-autism-spectrum-disorder> (2018)
- Dixon, D.R., Miyake, C.J., Nohelty, K., Novack, M.N., Granpeesheh, D.: Evaluation of an immersive virtual reality safety training used to teach pedestrian skills to children with autism spectrum disorder. *Behav. Anal. Pract.* **13**, 631–640 (2019)
- Halabi, O., El-Seoud, S.A., Alja'am, J., Alpona, H., Al-Hemadi, M., Al-Hassan, D.: Design of immersive virtual reality system to improve communication skills in individuals with autism. *Int. J. Emerg. Technol. Learn. (iJET)* **12**(05), 50–64 (2017)
- National Research Council: *Educating Children with Autism*. The National Academies Press, Washington, DC (2001)
- Newbutt, N., Bradley, R., Conley, I.: Using virtual reality head-mounted displays in schools with autistic children: views, experiences, and future directions. *Cyberpsychol. Behav. Soc. Netw.* **23**(1), 23–33 (2020)
- Parsons, S.: Authenticity in virtual reality for assessment and intervention in autism: A conceptual review. *Educ. Res. Rev.* **19**, 138–157 (2016)
- Schwarze, A., Freude, H., Niehaves, B.: Advantages and propositions of learning emotion recognition in virtual reality for people with autism. In: *Proceedings of the 27th European Conference on Information Systems (ECIS)*. AIS, Stockholm & Uppsala, Sweden (2019)
- Simões, M., Mouga, S., Pereira, A.C., de Carvalho, P., Oliveira, G., Castelo-Branco, M.: Virtual reality immersion rescales regulation of interpersonal distance in controls but not in autism spectrum disorder. *J. Autism Dev. Disord.* **50**, 4317–4328 (2020). <https://doi.org/10.1007/s10803-020-04484-6>



Using Augmented Reality in Computing Higher Education

Sarah Alshamrani Alshaikhi^{1,2}(✉)  and Mike Joy² 

¹ Department of Computer Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia

Saalshamrani@pnu.edu.sa

² Department of Computer Science, The University of Warwick, Coventry, UK
{Sarah.Alshamrani, M.S.Joy}@warwick.ac.uk

Abstract. In the past, the only way to receive educational content was through traditional methods, which included engaging the learners in full interaction with non-interactive books. Nowadays, we can generate three-dimensional virtual vision by using high-performance computer graphics, which is called Augmented Reality (AR).

This research focuses on investigating a new approach to emerging and integrating computing education with AR technology in Saudi Arabia.

The research will follow the design science methodology which is a mixed method approach for data collection and analysis.

As design science research is considered to be problem focused research, its main tasks are the illustration of the design problems and evaluation of design solutions.

Keywords: Augmented reality · Computer science · Augmented reality in education

1 Introduction

Augmented Reality is defined as using augmented video by covering an image with generated data in order to achieve a high performance three-dimensional image [1]. It will give three-dimensional virtual vision by using high-performance computer graphics [2]. Furthermore, it augments virtual media by modelling the real world with the user's complete control on both view and interaction. It will provide a multi view of the real objects covered with computer generated virtual objects.

This interactive simulation of the real world is done by engaging regular space, place and things that are partly unmediated [3]. It allows the real-world objects to combine with virtual objects or information. Thus, virtual objects seem to synchronize within the same area of the real-world objects. Using AR enhances user interaction and perception with the real world, and the augmented virtual object can offer information which cannot easily be detected with the user's own senses. The information transmitted by the virtual objects can give the opportunity to the user to examine actual real world tasks [4].

In recent years, a large number of research studies have evaluated the impact of applying AR to learning. These research studies can provide valuable information for both educators and AR designers who are enriching the new generation of minds through novel technologies. The benefit of harnessing results of research studies, supported with practical testing, can give practical and theoretical guidance to current and future educators who are interested in AR.

Although much research has investigated the impact of using AR in education as a simulation for practical education, with positive results, the possibility of integrating AR applications in Saudi Arabia has not been fully investigated, which is the main contribution in this research.

AR can be applied using wearable devices which have been used in several areas such as therapy of movement disorders and administration of drugs. Google Glass, which is an example of a wearable technology, has recently been used through medical training role-play tasks that can provide observation recording. Observation recording gives helpful information to be used through reflective learning and group debriefing, and a recording includes: patient attention, patient times spent on focusing several information sources [5]. One of the main approaches for using AR in education is the educational laboratory, which gives the student the opportunity to do an experiment virtually, as using the required equipment is more expensive and limited access [6]. The integration of AR technology components, such as animations, video, and images into a real lab environment has enhanced the student's science Learning capabilities as well as the student's laboratory skills. Moreover, AR can give the student the opportunity to observe some events which are impossible to be seen by real laboratory settings, for example molecules movement. Studies have reported that the usage of AR components on labs has improved the student's laboratory performance. In addition, the usage of AR has a significant impact on the increase of student's interaction which definitely affects their learning outcome [7].

AR has been widely introduced in the education area, and highly positive impact has been recorded in various courses. However, the Saudi Arabia education system has not fully deployed AR in its courses to improve the courses outcome, which is the focus of this research.

The overall motivation for this research is applying an AR system to computer hardware labs in Saudi Arabia where it has never been tested as an educational tool.

2 Problem Definition and Proposed Solution

Computer science is generally regarded as a hard subject to learn and teach, as the nature of implementing both the practical and scientific approach is challenging [8]. Computer hardware is a field where it is desirable for the labs to offer the full hardware equipment in order for the student to be engaged in understanding the lesson. Without such equipment having been provided the students will find it difficult to understand the main concepts of the lesson, and may not be able to work with hardware. On the other hand, the lecturer will spend extra time in teaching and illustrating to help the students to understand the hardware. Therefore, the learning efficiency will drop, which will affect the university outcomes for both students and lecturers.

This research addresses the following problems related teaching and learning hardware in computer science:

- The current knowledge approach using the traditional “unplugged” style of teaching hardware, which results in incomplete and difficult to understand material and may not provide students with the best educational experience;
- The lack of hardware material equipment provided to student;
- The difficulty of teaching computer hardware in theoretical way for the lecturer;
- AR tools simulating equipment in hardware computer labs.

1. Proposed Solution

Computer hardware labs are difficult to use and are poorly accessible for the student most of the time. Therefore, we propose to apply an AR simulation tool as a potential solution to give the student the familiarity with the equipment to reduce the damage of the hardware and offer the time needed for the student to be more engaged with the material and the augmented hardware.

3 Methodology

The research will be mixed methods using both quantitative and qualitative data. It will follow the design science approach as a methodology.

The reason behind following the design science research is that design science is technology oriented and is focused in producing a valuable technology product [9]. It is considered to be a problem focused research approach which analyses the problem and develops a suitable solution for it [10].

This research aims to investigate new ways of interactive learning by immersing AR technology into computer science labs. Applying AR technology to computer hardware labs will allow students to view internal operations as an interactive simulation. This approach, supported by pedagogy, is hypothesized to enhance the student learning experience.

To achieve this aim, the research will be divided into three major phases rounded with a student experience theory as illustrated in Fig. 1:

- Problem diagnosis: An investigative/exploratory study has been carried out to understand the current situation and student needs;
- Technology design: the development of an AR tool based on the investigative study data with the respect to pedagogy and student experience theory;
- Technology evaluation: assessing the augmented reality tools in terms of learning experience and technology acceptance.

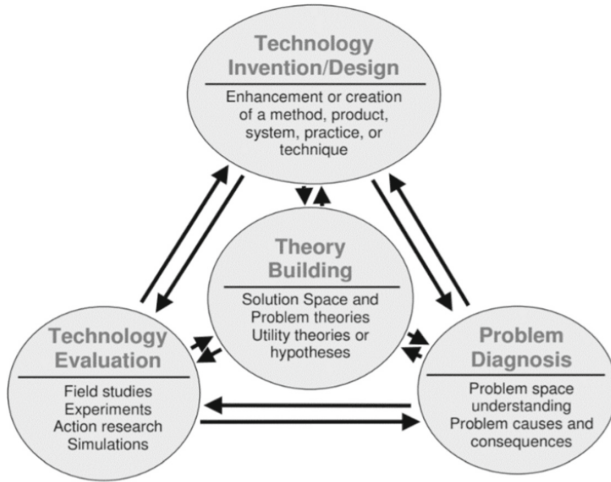


Fig. 1. An activity framework for design science research [11]

4 Results and Future Work

The exploratory study which covers the first phase has been conducted and the data have been collected. The data analysis is in process.

A preliminary analysis of both the qualitative and quantitative data confirms our initial hypothesis that there is a lack of hardware equipment in computing labs and that accessibility is difficult. Data further support acceptance of students to use new education tools, and that AR might be effective.

The future design science steps and phases will be:

- Making a full analysis the data;
- Developing and testing the tools;
- Deploying and evaluating usage of the tool.

References

1. Koll-Schretzenmayr, M., Casaulta-Meyer, S.: Augmented reality. *disP Plann. Rev.* **52**(3), 2–5 (2016)
2. Datcu, D., Lukosch, S., Brazier, F.: On the usability and effectiveness of different interaction types in augmented reality. *Int. J. Hum. Comput. Interact.* **31**(3), 193 (2015)
3. Kesim, M., Ozarslan, Y.: Augmented reality in education: current technologies and the potential for education. *Procedia Soc. Behav. Sci.* **47**, 297–302 (2012)
4. Azuma, R.: A survey of augmented reality. *Presence Teleoper. Virtual Environ.* **6**(4), 355–385 (1997)
5. Bower, M., Sturman, D.: What are the educational affordances of wearable technologies? *Comput. Educ.* **88**, 343–353 (2015)

6. Pasc, M.I., Tarca, R.-C., Vesselenyi, T., Popentiu-Vladicescu, F., Nagy, R.-B.: Remote educational system using virtual and augmented reality. In: The International Scientific Conference eLearning and Software for Education, pp. 221–228. National Defence University, Bucharest (2015)
7. Akçayır, M., Akçayır, G., Pektaş, H.M., Ocak, M.A.: Augmented reality in science laboratories: the effects of augmented reality on university students' laboratory skills and attitudes toward science laboratories. *Comput. Hum. Behav.* **57**, 334–342 (2016)
8. Webb, M., et al.: Computer science in the school curriculum: issues and challenges. In: Tatnall, A., Webb, M. (eds.) WCCE 2017. IAICT, vol. 515, pp. 421–431. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-74310-3_43
9. March, S., Smith, G.: Design and natural science research on information technology. *Decis. Support Syst.* **15**, 251–266 (1995)
10. March, S.T., Storey, V.C.: Design science in the information systems discipline: an introduction to the special issue on design science research. *MIS Q.* **32**(4), 725–730 (2008)
11. John, V.: The role of theory and theorising in design science research. In: First International Conference on Design Science Research in Information Systems and Technology, p. 17 (2006)

Author Index

- Abdessalem, Hamdi Ben 68
Abdullah, Junaidi 519
Abuazizeh, Moh'd 497
Aghajani, Mahsa 490
Alamri, Ahmed 119, 148, 173, 466
Alexandron, Giora 418
Alharbi, Khulood 211
Aljohani, Tahani 136
Allen, Laura K. 321
Alotaibi, Mona 131
Alrajhi, Laila 78, 148
Alshamrani Alshaikhi, Sarah 526
Alshehri, Mohammad 173
An, Sungeun 189
Andersen, Erik 430
Anikin, Anton 52
Anton, Chukhnov 259
Ascari, Soelaine Rodrigues 453
- Ben Abdessalem, Hamdi 490, 512
Benlamine, Mohamed Sahbi 387
Botarleanu, Robert-Mihai 321
Boussaha, Karima 364
Brisson, Janie 60
Broniec, William 189
Brown, Chris 211
Brunskill, Emma 430
- Carvalho, Leandro S. G. 466
Cerri, Stefano A. 376
Chang, Maiga 232, 357
Che Embi, Zarina 519
Christofaki, Maria 3
Christos, Michalakelis 22
Cojocar, Grigoreta-Sofia 439
Contractor, Maheen Riaz 247
Cristea, Alexandra 8
Cristea, Alexandra I. 28, 78, 119, 136, 148, 173, 211, 418, 466
Crossley, Scott Andrew 321
- Dare, Kodjine 512
Dascalu, Mihai 291, 321, 333
Denisov, Mikhail 52
- Dermeval, Diego 406
Dioşan, Laura-Silvia 439
Drissi, Samia 364
Drousiotis, Efthymoulos 161
Dufresne, Aude 239
- Escudeiro, Nuno 3
- Fehnker, Ansgar 299
Fonseca, Samuel C. 466
Frasson, Claude 68, 387, 490, 512
- Gasparetti, Fabio 350
Genovese, Alessia 505
Getseva, Vanesa 73
Giannakas, Filippos 343, 398
Goel, Ashok 189
Gottardo, Ernani 453
Graham, John 8
Guarnieri, Vincent 291
Guran, Adriana-Mihaela 439
- Hammock, Jennifer 189
Harit, Anoushka 28, 78
Hayashi, Yugo 99, 224, 310
Hodgson, Ryan 8
Hosseini, Hadi 279
- Isotani, Seiji 406, 466
- Joy, Mike 131, 526
Jung, Heeseok 41
Junior, Hermino B. F. 466
- Karampidis, Konstantinos 3
Kim, Hyeoncheol 41, 267
Kim, Seounghun 41, 267
Kim, Woojin 41, 267
Kirste, Thomas 497
Kokku, Ravi 286
Krahn, Ted 232
Krouska, Akrivi 343, 398
Kumar, Amruth N. 73, 444
Kuo, Rita 232

- Lajoie, Susanne P. 201
 Lemoisson, Philippe 376
 Lemos, Bruno 406
 Limongelli, Carla 112

 Machado, Alexandre 406
 Mader, Angelika 299
 Margiotta, Carmine 112
 Marin, Victor J. 247, 279
 Marina, Aivazidi 22
 Marino, Federica 505
 Marutschke, Daniel Moritz 310
 Maskell, Simon 161
 McNamara, Danielle S. 321, 333
 Morita, Junya 99, 224
 Mu, Tong 430

 Nascimento, Pedro 406
 Neagu, Laurentiu-Marian 291
 Newton, Natalie 333
 Nicula, Bogdan 333
 Nijloveanu, Daniel 481
 Niknazar, Mohammad 286
 Nkambou, Roger 60, 239

 Ogata, Hiroaki 107
 Ohmoto, Yoshimasa 99, 224
 Oliveira, David B. F. 466
 Oliveira, Elaine H. T. 466
 Orciuoli, Francesco 505
 Orcutt, Ellen 333
 Orji, Fidelia A. 369
 Orthlieb, Téó 68

 Paiva, Ranilson 406
 Papadourakis, Giorgos 3
 Penskoj, Nikita 52
 Pereira, Filipe Dwan 119, 148, 466
 Pimentel, Andrey Ricardo 453
 Popescu, Doru Anastasiu 481
 Posov, Ilya 259
 Pozdniakov, Sergei 259
 Prokudin, Artem 52

 Rigaud, Eric 291
 Rivero, Carlos R. 247, 279

 Robert, Serge 60
 Rodriguez, Luiz 466
 Rosli, Muhamad Irfan 519
 Rugaber, Spencer 189
 Ruiz-Segura, Alejandra 201
 Rump, Arthur 299

 Santos, Rodrigo 406
 Sciarrone, Filippo 350
 Sgouropoulou, Cleo 343, 398
 Shi, Lei 8, 28, 78, 161, 211
 Shimojo, Shigen 224
 Sodoké, Komi 239
 Stanciu, Gabriel Ciprian 481
 Stewart, Craig 119
 Sun, Zhongtian 28, 78, 119
 Sychev, Oleg 52, 93

 Taibi, Davide 112
 Tanoubi, Issam 239
 Tato, Ange 60
 Temperini, Marco 350
 Tenório, Kamilla 406
 Toda, Armando 418, 466
 Travadel, Sébastien 291
 Trigoni, Athina 3
 Troussas, Christos 343, 398
 Tymms, Peter 211

 Uglev, Viktor 93

 Vassileva, Julita 369
 Vempaty, Aditya 286
 Voyiatzis, Ioannis 343, 398

 Wang, Jingyun 107
 Wang, Shuhan 430
 Weigel, Emily 189
 Wu, Tao 357

 Yacobson, Elad 418
 Yordanova, Kristina 497
 Yu, Jialin 28, 78

 Zanfardino, Gennaro 505