

1 Towards Digitalisation in Examination and Grading: Best Practice and Challenges

Autor:
Deebul
Nair and
Tim
Metzler
and Mo-
hammad
Wasil and
Minh
Nguyen
and
Anastassia
Kuesten-
macher
and Paul
G. Plöger¹

Abstract

Although digital technologies are widely-used in teaching and learning, to the best of our knowledge besides multiple choice there are no well established solutions for electronic exams and/or grading assistance. In this paper we summarize our experiences which we gathered during several semesters of conducting electronic exams in combination to electronically assisted grading and our research to further automate this process.

Since 2012 in the department of computer science in Hochschule Bonn-Rhein-Sieg electronic exams using various digital format were carried out. In 2018, the first grading-assisted digital exam for the bachelor course "Grundlagen der Wahrscheinlichkeitstheorie und Statistik" (WuS) with 120 students was conducted.

Based on the literature study and practical experiences, we identify requirements for conducting an electronic exam and how we selected the respective implementation of a suitable format, for which we used Jupyter notebook². We introduce its advantages and limitations as well as discuss the types of assignments which can be utilized in Jupyter notebooks.

Despite multiple benefits from the plain use of a digital format in examinations, the major reduction of the teacher's workload is reached by using grading assistance. We have studied various auto grading techniques which assist an examiner during the grading process, applied some of them for WuS assignments and show the significant time reduction achieved for grading students' exam using only partial auto grading.

Introduction

Digitalisation in education is not a novel topic. Personal computers, mobile devices, various software applications, Internet, have already widely used to educate students. Digital technologies simplify communication between an educator and students, introduce remote teaching and class attendance, offer on-line easily available teaching material as well as allow digital services for homework assignments and feedback that have the most convenient format and time for both student and teacher.

¹Bonn-Rhein-Sieg University, Sankt Augustin, Germany
deebul.nair@h-brs.de, tim.metzler@h-brs.de, mwasil.wasil@smail.inf.h-brs.de, minh.nguyen@smail.inf.h-brs.de, anastassia.kuestenmacher@h-brs.de, and paul.ploeger@h-brs.de

²<https://jupyter.org>

Despite of massive usage of digital technologies in teaching and learning, examination is frequently take place without utilisation of digital tools completely i.e. written and oral type of exams or partially scan-exam as well as grading process is also completely manual. However electronic examination have multiple assignments in comparison to written one, starting from avoid having to decipher bad handwriting and finishing with introducing a new type of tasks which are not possible to present in the written form. In addition grading written examinations are time consuming procedure especially for the large group of students.

Learning is an individual process, if a student knows in which topics (s)he has deficits in. The role of teacher is to introduce material and identify deficits of each student. If a teacher has a course with 100 or more students, (s)he has to spend a lot of time on grading assignments and less to focus on giving high quality feedback. This invested time can be reduced by using a digital representation of the tasks as well as grading assistance. In this paper we present our work, experiences and results in processes of (i) digitalisation of homework assignments, (ii) digitalisation of written examinations and (iii) grading assistance.

In this paper, we introduce our practical experiences in digitalisation of written examinations in the department of computer science at Hochschule Bonn-Rhein-Sieg, based on which we can highlight the challenges which occur during digitalisation, present solutions to some of these challenges as well as discuss possible ways to deal with others. Moreover important assumptions and activities which should be taking into consideration are identified. In addition, we also propose the firsts steps towards automated grading assistance. Finally, the statistics and results from conducted examinations visualise the advantages of digital examinations with partially auto-gradable solutions versus paper examinations.

Related Work

Learning Management Systems: The first step in the digitization of education for universities is to look into different Learning Management Systems(LMS). The most popular open source LMS are Moodle³, Blackboard⁴, Canvas⁵, and ILIAS⁶. All these LMS provides different modules for digital exams and grading. For example, ILIAS provides the following question types: Multiple choice questions, Cloze questions, Ordering questions, Matching questions, Imagemap questions, Java applet questions, Essay questions, Numeric questions and Text subset questions. The major drawback of the LMS systems is the lack of coding based questions. All the learning management systems come with their format.

Jupyter Notebooks: Jupyter notebooks⁷ come as a good alternative for doing code based questions for assignments and exams. Jupyter notebooks provide a format that allows for interactive visualisation of data and code and combines code, description and visualisation in one single document. This makes it suitable for teaching and learning. While basic Jupyter notebooks only provide core functionality, there exist three ways of adapting notebooks: cus-

³<https://moodle.de>

⁴<https://www.blackboard.com/index.html>

⁵<https://www.canvaslms.com/open-source-lms>

⁶<https://www.ilias.de>

⁷<https://jupyter.org>

tom frontend extensions that change the look and behavior of the notebook, custom server extensions that allow for adding backend services and Jupyter widgets which can be used to include HTML and JavaScript elements into a cell. Another advantage is that cells in Jupyter notebooks have a metadata field which can be used to store arbitrary data in cells.

This makes Jupyter notebooks a popular medium for doing assignments and exams for large scale courses. CoCalc is an online service which provides online compute power for Jupyter notebooks. Gryd⁸ is another online service for Jupyter notebooks. HubHero⁹ provides professionally configured JupyterHub servers for teachers. The following provide services for running Jupyter notebooks codio¹⁰, Microsoft Azure notebooks¹¹ Amazon Sagemaker¹² Gradient by Paperspace¹³ Google Colaboratory (<https://colab.research.google.com/>) AI based Grading tools: Gradescope¹⁴.com is a system for online assessment of paper based, digital and code based homework assignments and exams. Gradescope uses AI based techniques like OCR to scan written exams and grade them in assisted manner. Crowdmark is another paper-scan software for converting paper exams to digital and grade them. Gradecam¹⁵ also provides an online portal for easy creation and grading of paper based exam. Codio¹⁶ provides an interface for grading coding questions. OpenEdx is an open platform for MOOC has a plugin for grading jupyter notebook assignment. Nbgrader is a jupyter notebook assignment creation and grading software. It automates grading of code based questions.

- multiple choice is fully automated.
- only result checking == multiple choice.

Digital Examinations

Digital or electronic examinations differ from classical written examinations mainly by using an electronic format. From this format two main requirements follow: the examination has to be conducted on an electronic device and an architecture for fetching and submitting the examination needs to be provided. To conduct digital exams one does not only have to consider technical requirements but also legal issues, for example electronic signature or verifying the integrity of the submission.

An electronic exam can be described as the following process: students enter a class-room where they have access to electronic devices provided by the university. A special network and corresponding software ensures that each device in the room only has access to the exam server from where they fetch the exam. After completing the exam, students submit their solutions, once this is done the students receive a hash code. It is used to verify that the students' solutions will not be manipulated after submission. After all students have submitted, examiners can start to grade these submissions. The grading is also done digitally on a grading

⁸<https://gryd.us>

⁹<https://hubhero.net>

¹⁰<https://codio.com/features/ide>

¹¹<https://notebooks.azure.com>

¹²<https://docs.aws.amazon.com/sagemaker/latest/dg/ex1-prepare.html>

¹³<https://www.paperspace.com/gradient>

¹⁴<https://www.gradescope>

¹⁵<https://gradecam.com/>

¹⁶<https://gradecam.com/gradecam-go>

server, which provides tools to guide the grading process. These tools include: automated tests, custom views as well as support for simultaneous grading by multiple examiners.

Based on our practical experiences, the students have to be familiar with the exam format and procedure. This can be achieved by giving assignments and trial exam in a similar format throughout the semester. In our case we use Jupyter Notebook as a digital format.

There are some key differences between the exam mode and assignment mode when using Jupyter Notebook. By default Jupyter notebooks are highly customisable, allowing students to add and delete cells, execute shell commands and JavaScript in the cells. While this is favorable in assignments that are run locally, in exam mode the functionality of Jupyter notebooks needs to be reduced. For this we use several frontend extensions which change the interface by removing buttons for creating, moving and removing cells and to disable keyboard shortcuts with the same functionality. Additionally the execution of code cells can be automated to initialize certain cells when the Jupyter notebook is loaded or to prevent the student from executing certain cells at all.

We plan to further customize Jupyter notebooks to increase security and functionality. On the security side we plan to filter out potentially dangerous shell and JavaScript commands. We also plan to implement new cell types that allow for multiple choice questions or provide an editor for graphs and diagrams such as UML or class diagrams.

- Scalability (number of users simultaneously)
- Extensibility (additional tools, services)
- Grading process: task view, statistics, two graders
- Task digitalisation: Which kind of assignments exist in general? How to digitalise various kind of assignments?
- digitalisation can already speed up the process by (i) task-view (other GUI advantages?) (ii) reachability (do not need to carry 300 exams, can correct during traveling), (iii) centralised tool setup.

Digital Grading Assistance

One of the challenges in modern classrooms is a large number of assignments that need to be evaluated, often pitted against a limited number of graders. Even if the graders are able to cope with the large workload, inconsistent scores among graders is also a serious concern [MBM11]. Digital grading assistance aims to tackle these problems via automating portions of the grading process. Traditionally, such automation can be done via creating answers in simplified formats such as multiple choice, fill-in-the-blanks or single value answers. For coding assignments automated test cases can be constructed. Naturally, these approaches come with limitations on what type of questions and answers can be assigned to students. It may be beneficial, for example, to be able to automate grading short text answers.

Ordinal Peer Grading with Client-side Component Model

Recently, *peer-grading* has become a popular grading scheme which work with a great variety of answer formats [RJ14, LL18, CKV15]. The scheme requests students themselves to evaluate each other, then combine these peer evaluations to calculate the final grades . The student can either provide direct evaluations for each of her assigned responses (cardinal peer grading, CPG), or an ordering of the response subset indicating their relative quality (ordinal peer grading, OPG) [RJ14]. While Raman & Joachims [RJ14] demonstrate that OPG and CPG perform similarly accurate in term of grade estimation, Lin & Lu [LL18] point out that students' lack of grading expertise and personal incentives may result in inaccuracies if they are to give cardinal grades.

In this context, we implement a platform for ordinal peer grading using the Client-Side Component Model (CCM) framework ¹⁷, another open source project from Bonn-Rhein-Sieg University. CCM provides a service for embedding CCM components inside web-based contents, and one for managing data used by such components. Our peer grading platform contains a CCM component for instructors to create and modify questions, a component for students to answer these questions, a component which sample random entries from each question's answer pool for students to rank, and a component which display the score for each answer calculated from their rankings. The implementation of this platform is open source and can be found on GitHub ¹⁸.1

- Which kind of assignments exist and how they can be automated graded: task hierarchy: from multiple choice to case-studies.
- ML in grading assistance:
 - Where to take dataset: (i) Q&A session, students evaluate themselves CCM; (ii) Central repository, inside of university between programs and departments as well as between universities.
 - Existing approaches: depending on the type of the assignment various methods could be used (i) Asserts/docstring (ii) NLP, (iii) Active learning (this approach is not really for a particular assignments)
 - Evaluation of auto-graded tasks: - Ask human/tester to grade the tasks in parallel and compare the grades.
- Output of ML: grade, statistics for teacher/student, recommendations of material for self-study.
- Law issues. Can we only use auto-grading?

Current Results

The Autonomous Systems Group in the Computer Science Department of Hochschule Bonn-Rhein-Sieg were the early adapters of the Jupyter Notebook based examination. The group

¹⁷<https://github.com/ccmjs/ccm>

¹⁸https://github.com/digiklausur/ccm_components

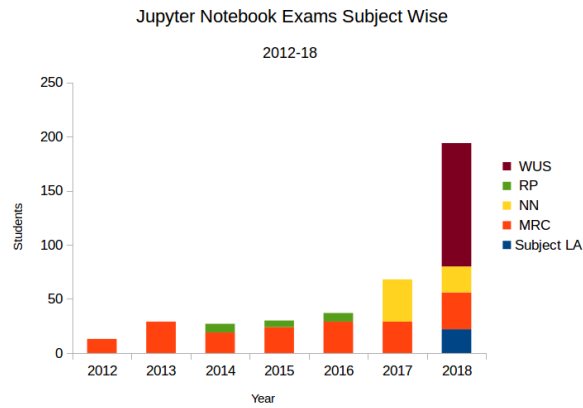


Figure 1.1: Jupyter Notebook Exams 2012-2018.

has conducted a total of 20 exams from 2012 till now, spread across 5 subjects. A total of 398 students have taken the exams digitally.

- How many e-examinations have been conducted so far.
- Time comparison. Manual grading vs. e-grading
- Which percentage of auto-gradable assignments do we have so far (see WuS exam) How to evaluate auto-grading?

Conclusions

- Contributions
- Assumptions
- Limitations
- Future work

Bibliography

- [CKV15] CARAGIANNIS, Ioannis ; KRIMPAS, George A. ; VOUDOURIS, Alexandros A.: Aggregating Partial Rankings with Applications to Peer Grading in Massive On-line Open Courses. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. Richland, SC : International Foundation for Autonomous Agents and Multiagent Systems, 2015 (AAMAS '15). – ISBN 978-1-4503-3413-6, 675-683
- [LL18] LIN, Chuang-Chieh ; LU, Chi-Jen: Efficient Mechanisms for Peer Grading and Dueling Bandits. In: ZHU, Jun (Hrsg.) ; TAKEUCHI, Ichiro (Hrsg.): Proceedings of The 10th Asian Conference on Machine Learning Bd. 95, PMLR, 14-16 Nov 2018 (Proceedings of Machine Learning Research), 740-755
- [MBM11] MOHLER, Michael ; BUNESCU, Razvan ; MIHALCEA, Rada: Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, USA : Association for Computational Linguistics, 2011. – ISBN 978-1-932432-87-9, 752-762
- [OTXX] ONE, Author ; TWO, Author: Test 1. Version: 06.November XXXX. <https://www.froscon.de/> – note: visited 2018-08-22
- [RJ14] RAMAN, Karthik ; JOACHIMS, Thorsten: Methods for Ordinal Peer Grading. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA : ACM, 2014 (KDD '14). – ISBN 978-1-4503-2956-9, 1037-1046