SwaGrader: A Honest Effort Extracting, Modular Peer-Grading Tool

Somu Prajapati, Ayushi Gupta, Shubham Kumar Nigam, Swaprava Nath
Indian Institute of Technology Kanpur
{somupra,gayushi,sknigam,swaprava}@iitk.ac.in

ABSTRACT

Massive open online courses pose a massive challenge for grading the answer scripts at a high accuracy. Peer grading is often viewed as a scalable solution to this challenge, which largely depends on the altruism of the peer graders. In this paper, we propose to demonstrate a tool designed for strategic peer-grading with the help of a structured and typical grading workflow. SwaGrader, a modular, secure and customizable (to any grading workflow) peer-grading tool enables the instructor to handle large courses (MOOCs and offline) with limited participation by teaching staff via a web-based application (extensible to any front-end framework based application) and a mechanism called TRUPEQA[1]. TRUPEQA (a) uses a constant number of instructor-graded answer-scripts to quantitatively measure the accuracies of the peer graders and corrects the scores accordingly, and (b) penalizes deliberate under-performing. We show that this mechanism is unique in its class to satisfy certain properties. Our human subject experiments show that TRUPEQA improves the grading quality over the mechanisms currently used in standard MOOCs. Our mechanism outperforms several standard peer grading techniques used in practice, even at times when the graders are non-manipulative.

KEYWORDS

Education, Peer-grading, Mechanism design, Tool development

ACM Reference Format:
Somu Prajapati, Ayushi Gupta, Shubham Kumar Nigam, Swaprava Nath. 2020. SwaGrader: A Honest Effort Extracting, Modular Peer-Grading Tool. In *7th ACM IKDD CoDS and 25th COMAD (CoDS COMAD 2020), January 5–7, 2020, Hyderabad, India.* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Traditional approaches in the peer-grading literature largely depend on the altruism of the peer graders. Some peer-grading approaches treat it as a *best-effort* service of the graders, and statistically correct their inaccuracies before awarding the final scores. Approaches that incentivize *non-strategic behavior* of the peer graders do not make use of certain possible additional information, e.g., that the true grade can eventually be observed at the additional cost of the teaching staff time if an affected student raises a *regrading request*. In this paper, we consider a mechanism TRUPEQA that uses this additional information and demonstrate a tool that is based on it. Properties and uniqueness of the algorithm are described later in the paper.

CoDS COMAD 2020, January 5–7, 2020, Hyderabad, India

ACM ISBN 978-1-4503-7738-6/20/01.

2 FLOWCHART OF THE PROPOSED TOOL

Through this demonstration, we propose the tool *SwaGrader*, which can be used to handle massive online and offline courses and is highly customizable. Frontend is based on webapp framework for this demo. Backend of the tool exposes a service-oriented, largely RESTful API, allowing several frontends - desktop, mobile apps to be independently developed and allows for a lightweight backend which means speedy responses, low maintenance and high scalability (by dockerization of backend). Components are connected by layers of security/authentication and load balancing. Object level permissions are given to the user based on his role in the course, thus, preserving sensitive information from anarchy attacks, in which attacker gets increased control over the application once the account of some user is compromised. A demonstration is made

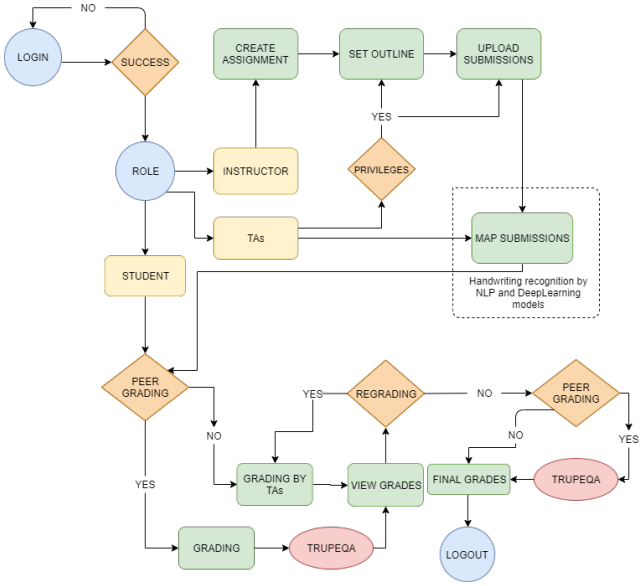


Figure 1: A typical grading workflow with TRUPEQA implemented

on a typical course workflow, each level can be customized on its own, peer-grading component is totally modular and is exposed as RESTful, service oriented, lightweight API. Figure 1 shows the grading workflow which:

(1) Allows instructor to post assignments and set outline.
(2) After getting all the submissions, allows instructor, or teaching assistants to grade some of the copies.

(3) Maps submissions to respective graders and their student owners.[1]
(4) Collects the grades given by the peers and calculates the final TRUPEQA score for each of the students.
(5) Allows students to raise regrading requests which is addressed by the teaching staff.

SwaGrader thus implements TRUPEQA and minimizes efforts of the teaching staff.

3 DETAILS OF THE IMPLEMENTATION

The solution is a Web-based application with a modular RESTful Backend (thus, extensible to mobile platforms) with orchestral layer of authentication/security and load balancing.
The mechanism **TRU**thful **P**eer **E**valuation with **Q**uality **A**ssurance (TRUPEQA) used behind the scenes is an intelligent, scalable, efficient, and robust mechanism devised to incentivize honest efforts of the peer-graders. TRUPEQA estimates the accuracy of the graders using the probe papers. It incentivizes the graders to grade at their highest level of reliability and is insensitive to any grader biases.[2]

3.1 Staging

Before starting the peer-grading, users registered for a course must be given roles. Since the whole design of our tool is customizable, a user can have different permissions based on his/her role (they can be set by the instructor on a single click). Handwritten submissions are stacked in a file and are uploaded. The backend mechanism, on the basis of the given outline, automatically partitions the file to small files which are then mapped to respective students.

Mapping the author of the submitted files is currently done manually by the teaching staff, but we aim in future to use NLP and deep learning models to automate this. Once the staging is done, and all the submissions are mapped, instructor initiates the peer-grading. Alternatively, she may opt to grade the copies herself, and then the submissions will be directly checked by the teaching assistants and the mappings will be done section-wise.

3.2 Distribution of Papers

Before distributing the papers, teaching staff grades a fixed number of copies (say p), which we call the *probe* papers. Every grader $i \in N$ is assigned K (even) papers to grade (we assume that the number K is a design choice and is decided apriori), $K/2$ of which are probe papers (these are randomly picked from the p probe papers) and rest are non-probe, in such a way that every non-probe paper is assigned to exactly $K/2$ graders. Before assigning the probe papers, they are categorized in $K/2$ zones based on their *performance scale*.[3] The following cases may arise:

(1) Performance of students are known previously, in this case, a sorted input vector V of length = N (Strength of the class), is given. And, $K/2$ zones (partitions) are made. File descriptors are pointed to files of each zone ensuring that each copy in $K/2$ probes belong to exactly one performance zone.
(2) If no performance vector can be justified, a random partition in $K/2$ partitions is made, and current assignment grades are fed to next assignment performance vector for later use.

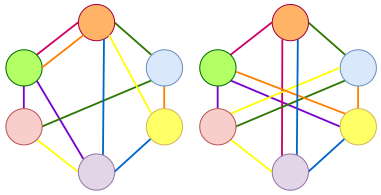


Figure 2: An incorrect distribution vs a correct distribution for $K = 6$, nodes represent students while edges represent copies

Thus, the assignment of papers to graders ensures that the grader does not get her own paper assigned to her (consider Figure 2 as an example, each node should have exactly 4 edges, of which 3 are of different color, and each edge should be repeated twice), and gets probe paper of $K/2$ performance zones. Figure 3 shows the peers' view for $K = 8$, out of these K copies assigned to him, he has exactly 4 probes of 4 different levels. The accuracy matrix $q_i, i \in N$ is estimated by applying e_i on the P_i probe papers as discussed later. This matrix has column vectors as the bias and reliability of graders respectively.

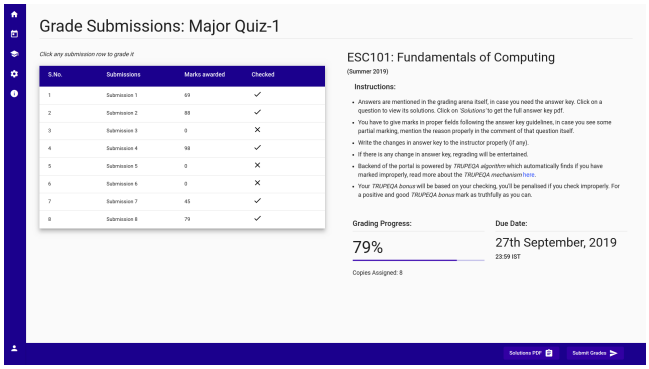


Figure 3: A peer's view with $K = 8$ copies assigned to him for grading

3.3 TRUPEQA model and mechanism

After the copies are distributed, now peers grade the copy and submit their scores. So, marks given to each question can be mapped in a 3D vector space. So, The score given by the i^{th} grader, in the j^{th} copy's k^{th} question would be $(\tilde{y}_{j\setminus k})$ for $i \in G(j)$ where G is the set of graders for j^{th} copy. Database schema for the input model is shown below by Figure 4, only the students' metadata should be passed at the endpoint, output will be the TRUPEQA scores and TRUPEQA bonus which will be saved directly to the respective relational database tables:

[1]Development of NLP models are in progress to do this task automatically with minimum intervention by the teaching staff.
[2]For the detailed definition of these properties, see [1].
[3]Performance scale is based on how a grader perceives a copy even before checking it thoroughly, it is likely possible that a *'good paper'* has a neat handwriting or representation, with *'bad logic'*. But, the zones can be formed on the basis of the overall marks of the students, assuming that a *'good copy'* has higher marks.

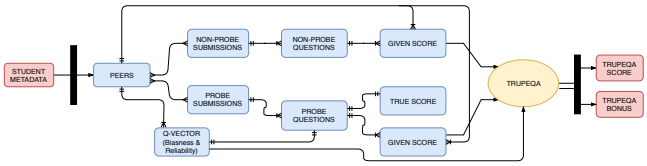


Figure 4: Database schema for the peer-grading module

3.3.1 EPBI and EPRM:. A peer grading mechanism is **E**x-**P**ost **B**ias **I**nsensitive (EPBI) if for every grader i, the utility u_i does not change with her bias b_i irrespective of the biases and reliabilities chosen by other graders, the true scores and the scores reported by different graders.

A peer-grading mechanism is **E**x-**P**ost **R**eliability **M**onotone (EPRM) if for every grader i, the utility u_i is monotonically non-decreasing with her reliability irrespective of the biases and reliabilities chosen by other graders, the true scores and the scores reported by different graders. We can show that[4]

TRUPEQA is both EPBI and EPRM.

3.3.2 Bias and Reliability: For the bias and reliability of each grader we use their maximum likelihood estimates from the given scores and true scores of the grader on the questions of probe papers. The bias, b_i and reliability, τ_i are $n \times 1$ vectors (if the papers have n questions each). Hence for grader i and question k the estimates $e_{i\backslash k}(\tilde{y}^{(i)}_{P_{i\backslash k}}, y_{P_{i\backslash k}}) = (\hat{b}_{i\backslash k}, \hat{\tau}_{i\backslash k})$ are calculated as follows:

$$\hat{b}_{i\backslash k} = \frac{\sum_{j \in P_i} (\tilde{y}^{(i)}_{j\backslash k} - y_{j\backslash k})}{|P_i|}$$

$$\hat{\tau}_{i\backslash k} = \frac{|P_i|}{\sum_{j \in P_i} (\tilde{y}^{(i)}_{j\backslash k} - (y_{j\backslash k} + \hat{b}_{i\backslash k}))^2}$$

where, P_i is the set of probe papers assigned to the ith grader and $\tilde{y}_{j\backslash k}$ for $i \in G(j)$ is the marks given by the i^{th} grader to j^{th} paper's k^{th} question. For the generation of the true scores and the error model of the peer-graders, we use the $\mathbf{PG}_1$ model of grader bias and reliability as described in [2], which is a widely used model for continuous scores (used by Coursera).

3.3.3 Calculation of TRUPEQA scores: The operating principle of this mechanism is to assign equal number of papers to the graders and pick the score of a paper to be a weighted sample mean (with appropriately chosen weights), which 'almost' minimizes the expected cost. Finally, the transfer (peer-grading bonus score) is the marginal contribution of the grader towards minimizing this cost. Following are the steps of TRUPEQA:

(1) *Inputs:* The parameters μ and γ of the priors on $y_j, \forall j \in N$ and the observed scores of paper j by grader i, given by $\tilde{y}^{(i)}_j$, for all $i, j \in N$, are taken as inputs.The model parameters μ, γ are chosen appropriately to reflect a realistic peer grading scenario.

In traditional examinations, the scores typically lie between 0 and 100. The parameters of the above model are chosen so as to compress this score spread within a width of 1. The value of $\mu = 1$ and $\gamma = 16$ implies that the true score comes from the prior $N(1, 1/16)$ which ensures that 95% of the score values lie within $[0.5, 1.5]$ with mean 1.

These are the TRUPEQA priors which should be set manually, setting them according to previous results is a scope in future. Instructor can change these parameters, thus, has a total TRUPEQA control over the peer-grading process.[5]

(2) *TRUPEQA Score* r^*_j*:* The score computing function $r_k = (r_{j\backslash k} : j \in N \backslash P)$ of a mechanism M is **inverse standard-deviation weighted mean (ISWM)** if $r_{j\backslash k}$ for every question k of a paper j is given by:

$$r^*_{j\backslash k} = \frac{\sqrt{\gamma}\mu + \sum_{i \in G(j)} \sqrt{\hat{\tau}_{i\backslash k}} (\tilde{y}^{(i)}_j - \hat{b}_{i\backslash k})}{\sqrt{\gamma} + \sum_{i \in G(j)} \sqrt{\hat{\tau}_{i\backslash k}}}$$

Hence, the total score r^*_j of a paper j with n questions is given by:

$$r^*_j = \sum_{k=1}^{n} r^*_{j\backslash k}$$

These are the scores calculated by TRUPEQA, and will be considered as true score for a paper until it is submitted for regrading. Figure 5 shows the stage when the TRUPEQA scores are released alongwith the bonus, student may opt to submit his paper again for regrading.

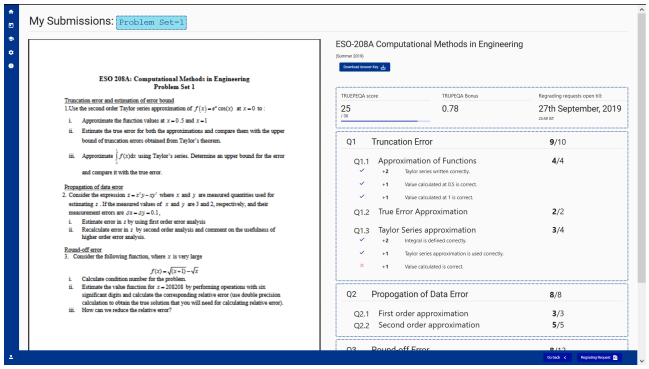


Figure 5: Students' view of peergraded sheet, Regrading requests kept open by the instructor.

(3) *Regrading:* Students can ask for regrading if they are not satisfied with the given grade $r^*_{j\backslash k}$. In such a case, a teaching staff looks at the paper (not only question $j\backslash k$) and reassigns the marks for each question. These newly given marks are considered the true score $y_{j\backslash k}$ and are replaced in the expressions of social welfare and transfer as follows.

(4) *Social Welfare:* The **social welfare** at a score $r^*_{j\backslash k}$ for k^{th} question of paper j when the true score is $y_{j\backslash k}$ is denoted by:

$$W^*_{j\backslash k} = R(r^*_{j\backslash k}, y_{j\backslash k})$$

where R is the Reward function, given by:

$$R(x_i, y_i) = -|x_i - y_i|$$

[4]The formal definitions of the properties and the proofs are available in [1].

[5]For a more rigorous analysis, please refer to [1].

The **social welfare** at a score r_j^* for k^{th} question of paper j **without grader** i when the true score is $y_{j\backslash k}$ is denoted by:

$$W_{j\backslash k}^{(-i)*} = R(r_{j\backslash k}^{(-i)*}, y_{j\backslash k}), \qquad \text{where,}$$

$$r_{j\backslash k}^{(-i)*} = \frac{\sqrt{\gamma}\mu + \sum_{m\in G(j)\backslash\{i\}} \sqrt{\hat{\tau}_{m\backslash k}}(\tilde{y}_{j\backslash k}^{(m)} - \hat{b}_{m\backslash k})}{\sqrt{\gamma} + \sum_{m\in G(j)\backslash\{i\}} \sqrt{\hat{\tau}_{m\backslash k}}}.$$

This is basically the influence grader i has on the class, for a large negative social welfare for a question (paper) without the grader, his TRUPEQA bonus will be larger (as shown in the next section), this is basically the reward for truthful grading relative to others.

(5) *Transfer Score* t_i*:* Transfer score represents the bonus given to a grader for checking paper j truthfully. The transfer to grader i for grading k^{th} question of paper $j \in NP_i$ is given by $t_i^{j\backslash k} = W_{j\backslash k}^* - W_{j\backslash k}^{(-i)*}$. Hence the total transfer score for a paper j consisting of n questions is given by$t_i^j = \sum_{k=1}^n t_i^{j\backslash k}$. Therefore, the total transfer score given to grader i for checking all the non-probe papers assigned to him is given by $t_i = \sum_{j\in NP_i} t_i^j$.

The TRUPEQA transfer is the *marginal contribution* of a grader in the grading of a paper. It implies that the grader whose participation increases the accuracy of the given grade significantly from that in absence of him, will be paid higher than the grader for whom this does not happen. A repeated interaction with this mechanism makes a rational grader realize the penalties for grading without effort. Hence we can informally say:

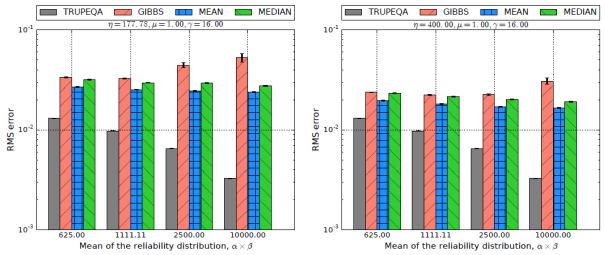
> Once the graders know how to play the game, it will stabilize to a game where every peer-grader tries to make their grading as accurate as possible.

4 COMPARISON OF TRUPEQA WITH OTHER PEER-GRADING MECHANISMS

Previous literature uses some peer-grading tools such as Gibbs [2], mean and median (as used by Coursera), but their mechanisms assume that the peer-graders reveal the scores truthfully, while TRUPEQA incentivizes the graders to do this. On a synthetic data, even when graders are non-strategic, interestingly, it yields statistically significant lower RMS (Root Mean Squared) error than the Gibbs sampling mechanism, and also the mean and median mechanisms. It also receives less fraction of regrading requests compared to those mechanisms. Quite naturally, it performs significantly better when the graders are strategic.

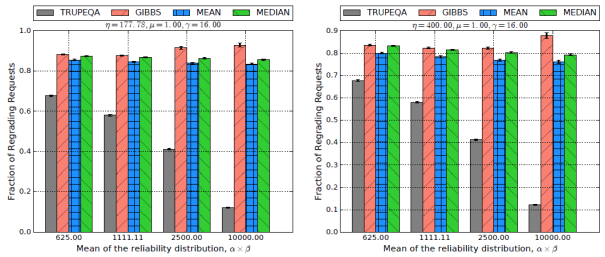
Both TRUPEQA and Gibbs need the knowledge of priors of the scores. If the prior used by the mechanism is different from the true prior, the performance of these two mechanisms are affected. However, Gibbs turns out to be too sensitive to it and the error increases with increasing reliability (as evident from the results in the text[1]), while TRUPEQA continues to perform better as reliability increases.

Experiments with human subjects show that TRUPEQA gives a much more accurate score to the papers compared to a popular mechanism used often in MOOCs.



(a) $\eta = 177.78$ (b) $\eta = 400.00$

Figure 6: RMS errors for different mechanisms-truthful graders.



(a) $\eta = 177.78$ (b) $\eta = 400.00$

Figure 7: Fraction of regrading requests for different mechanisms-truthful graders

5 DEMONSTRATION

The application is currently available only within IIT Kanpur firewall. Some orchestral components are still under development and NLP model of mapping is under progress. However some screenshots of the application are provided for the demonstration purposes. These are available at: http://bit.ly/SwaGrader.

6 SUMMARY

The SwaGrader peer-grading tool is able to handle large courses (offline and online), following a typical but flexible and customizable grading workflow, and uses peer-grading to self-enforce the graders for a better effort. For deployment, dockers and containers would be used to support horizontal as well as vertical scaling of the product. RESTful API backend for TRUPEQA lets independent development of frontend with full template customization freedom (and then plugging to the abstract endpoints of the tool), keeping the algorithm backend pluggable and untouched. Object level security measures are taken for the prevention of some common but major security threats like SQL injection, XSS attacks, clickjacking, CSRF attacks etc. Roles are highly customizable (instructor may choose to give various permissions, thus setting roles, of the teaching staff), making the tool flexible in its own. Further enhancements include an NLP model to recognize handwritten names to map the submissions automatically in the database, and optimization of TRUPEQA priors at each iteration of peer-grading.

ACKNOWLEDGMENTS

This work is supported by the IIT Kanpur Grant Number 2017198.

REFERENCES

[1] Anujit Chakraborty, Jatin Jindal, and Swaprava Nath. 2019. Ensuring Honest Effort in Peer Grading. *arXiv preprint arXiv:1807.11657* (2019).

[2] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. 2013. Tuned models of peer assessment in MOOCs. *arXiv preprint arXiv:1307.2579* (2013).