Employing Peer Review to Evaluate the Quality of Student Generated Content at Scale: A Trust Propagation Approach

Ali Darvishi
The University of Queensland
Brisbane, QLD, Australia
a.darvishi@uq.edu.au

Hassan Khosravi
The University of Queensland
Brisbane, QLD, Australia
h.khosravi@uq.edu.au

Shazia Sadiq
The University of Queensland
Brisbane, QLD, Australia
shazia@itee.uq.edu.au

ABSTRACT

Engaging students in the creation of learning resources has been demonstrated to have pedagogical benefits and lead to the creation of large repositories of learning resources which can be used to complement student learning in different ways. However, to effectively utilise a learnersourced repository of content, a selection process is needed to separate high-quality from low-quality resources as some of the resources created by students can be ineffective, inappropriate, or incorrect. A common and scalable approach to evaluating the quality of learnersourced content is to use a peer review process where students are asked to assess the quality of resources authored by their peers. However, this method poses the problem of "truth inference" since the judgements of students as experts-in-training cannot wholly be trusted. This paper presents a graph-based approach to propagate the reliability and trust using data from peer and instructor evaluations in order to simultaneously infer the quality of the learnersourced content and the reliability and trustworthiness of users in a live setting. We use empirical data from a learnersourcing system called RiPPLE to evaluate our approach. Results demonstrate that the proposed approach can propagate reliability and utilise the limited availability of instructors in spot-checking to improve the accuracy of the model compared to baseline models and the current model used in the system.

Author Keywords

Learnersourcing; Crowdsourcing in Education; Learning Analytics; Peer Review; Consensus Approaches; Trust Propagation.

CCS Concepts

•**Applied computing** → **Computer-assisted instruction; Interactive learning environments; Collaborative learning;** •**Information systems** → *Crowdsourcing;* •**Human-centered computing** → Collaborative and social computing systems and tools;

L@S '21, June 22–25, 2021, Virtual Event, Germany.

ACM ISBN 978-1-4503-8215-1/21/06 ...$15.00.
http://dx.doi.org/10.1145/3430895.3460129

INTRODUCTION

The concept of engaging learners as contributors to novel content, also referred to as learnersourcing [25], is emerging as a viable learner-centred and pedagogically supported approach to engaging students in higher-order learning and authentic learning experiences at scale [21]. Learnersourcing has strong roots in the learning sciences and is aligned with established and contemporary learner-centered approaches [33] such as inquiry-based learning [19], contributing student pedagogy [16] and students as partners [28]. A side benefit and an increasingly recognised application of learnersourcing have been to use students' contributions within adaptive engines to support the personification of education [18, 20, 23, 44].

Although learnersourcing offers various benefits for both students and instructors, there are several concerns associated with the use of learnersourcing in the educational systems. For example, Heffernan et al. [18] stress that the main risk inherent to the use of learnersourcing is to control the quality of the content created by students. They also suggest that anonymous quality ratings and peer-feedback would enable identifying reliable and high-quality learning contents by students themselves. Previous work has demonstrated that the quality of students' contribution in creating learning resources is rather diverse, with some developed resources meeting rigorous judgmental criteria while other resources are ineffective, inappropriate, or incorrect [13, 8, 1, 35, 38].

Employing peer-evaluation approaches, where multiple students' decisions are integrated towards inferring a resource's quality, is a viable candidate for inferring the quality of student-generated resources. Engaging students as evaluators of learning resources encourage them to think critically and analytically about the learning resources, reflect on their own created content, and help them develop evaluative judgement, which has been recognised as an important aspect of the learning process [36]. However, this method poses the problem of "truth inference" since the judgements of students as experts-in-training cannot wholly be trusted. While some prior work has reported on students' ability to effectively evaluate resources [6, 13, 35, 43], the consensus approaches by which learnersourced evaluations can be meaningfully integrated towards inferring the quality of a resource is under-developed.

The aim of this paper is to propose and evaluate a graph-based reliability propagation approach that uses data from peer and instructor evaluations to simultaneously infer the quality of the learnersourced content and the reliability of users. Results

demonstrate the out-performance of the reliability propagation approach compared to baseline models and the current model used in RiPPLE in evaluating the quality of learnersourced content. In what follows, Section 2 first presents an educational system called RiPPLE that relies on learnersourcing for creating and evaluating learning content. We discuss the current consensus approach employed in RiPPLE, the rationale for why this approach was chosen and a data-driven reflection of its limitations. Section 3 then presents related work on learnersourcing, current peer review consensus approaches, and trust propagation methods. Section 4 provides notation and a formal definition of the problem under investigation and Section 5 presents our proposed approach. Section 6 uses empirical data from the adoption of RiPPLE in five courses to evaluate our proposed approach. Finally, Section 7 discusses the implications and potential benefits and shortcoming of integrating our presented algorithms into an educational system that supports student creation and evaluation of the content. We also present several interesting directions to pursue in future work to overcome current limitations.

BACKGROUND

This section presents an overview of the moderation process in the adaptive educational system used in this study – the RiPPLE platform. It provides an individualised learning experience by adapting the type or the difficulty level of instruction tailored to students' needs or preferences. To do this, it needs a large repository of learning resources. Relying only on domain experts or instructors to create such a repository would be expensive in terms of cost and time. As such, students are engaged to collaborate with instructors in the creation of content and also the evaluation of their peer's work quality. Figure 1 provides an overview of the moderation process in RiPPLE.

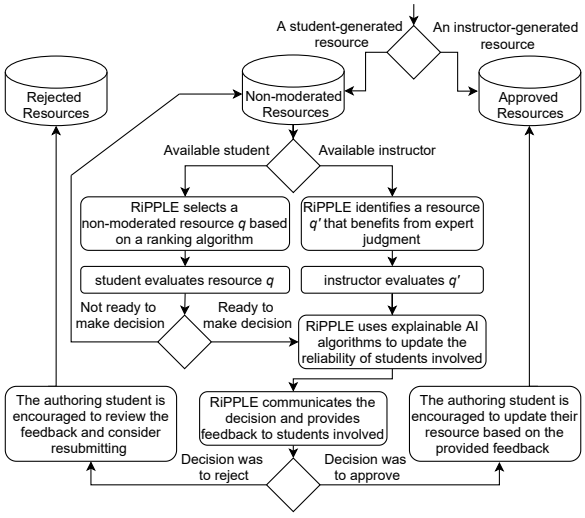

Figure 1: Overview of the creation and moderation process.

An instructor-generated resource is directly approved, whereas a student-generated resource must pass the moderation process. Moderators use an interface, as shown in Figure 2, to evaluate a resource quality, which guides students to consider a rubric of four items – alignment, correctness, difficulty, and critical thinking level of the resource.

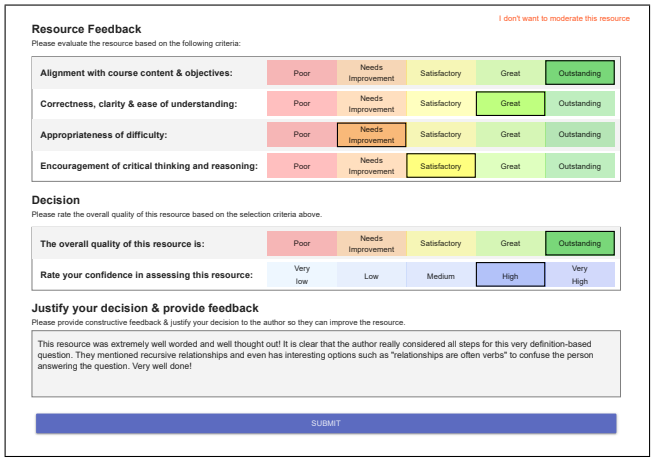

Figure 2: Evaluation rubric for moderation

Then, the system determines whether or not it is ready to make a decision on the quality of the resource under moderation based on a number of criteria, such as the number of received moderations, the level of agreement between the moderators, and their reliability. If it is not ready to make a decision, then the resource remains in the non-moderated resources repository for further moderations. Otherwise, RiPPLE uses an explainable consensus algorithm to simultaneously infer the reliability of student moderators and the quality of resources. The current consensus model relies on the expectation maximisation (EM) algorithm [29]. As described in [6] and Section 6, it begins with initialising all students' reliability to an equal value. Then, in the expectation step, it uses a weighted aggregation of students' ratings to infer a resource's quality. Finally, in the maximisation step, it updates students' reliabilities based on the goodness of their ratings compared to the inferred quality. The authoring student is then encouraged to revise their submission based on the provided feedback to remove the approved resource's minor issues or to review and consider resubmitting if rejected. Figure 3 shows an example of a moderation outcome and feedback.

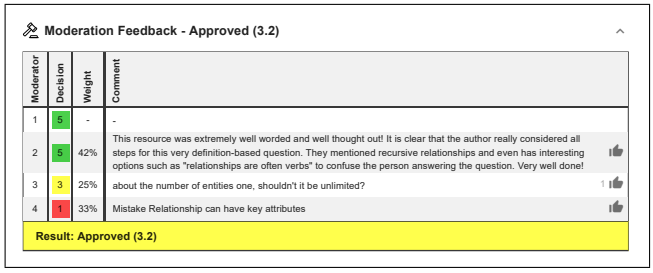

Figure 3: An example of moderation outcome and provided feedback.

The current model was selected as: (1) it has low computational complexity to be implemented in a live setting, (2) the inferred ratings and reliability updates are easily explainable, and (3) our previous work shows that the system seemed fair to students and instructors [21]. Still, it holds a flaw in its

consensus approach. Like most crowdsourcing systems, the current algorithm mainly depends on the majority's decision and often diminishes judgments from a minority of wise moderators. In the collected data from 5 courses (cf. Section 6), a total number of 77,297 moderations have been submitted on 12,803 resources from 2,141 students. Figure 4 provides a preliminary analysis based on the 4,918 moderations on 694 resources that had also received an instructor moderation.

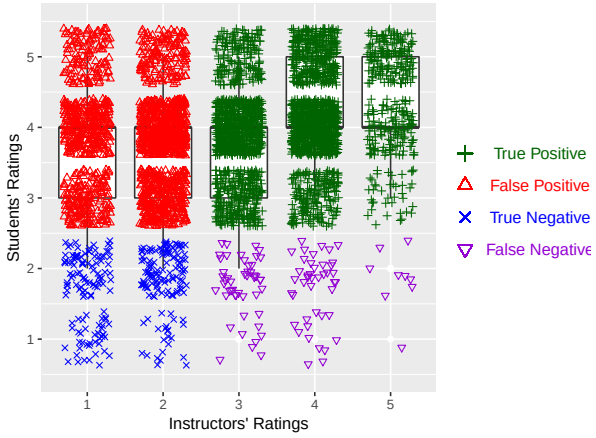

Figure 4: Student moderations performed on resources that also received an instructor moderation.

This figure demonstrates that in cases where instructors have approved a resource (i.e., provided a rating of 3, 4 or 5), the probability of receiving a true positive (TP) where students have also approved that resource (i.e., provided a rating of 3, 4 or 5) is much higher $(96.6\%(2868) \gg 3.4\%(101))$ than receiving a false negative (FN) where students have rejected a high-quality resource (i.e., provided a rating of 1 or 2). In contrast, it also shows that in cases where instructors have rejected a resource (i.e., provided a rating of 1 or 2), the probability of receiving a true negative (TN) where students have also rejected that resource (i.e., provided a rating of 1 or 2) is much lower $(13.5\%(264) \ll 86.5\%(1685))$ than receiving a false positive (FP) where students have approved a low-quality resource (i.e., provided a rating of 3, 4 or 5). This pre-analysis illustrates that consensus approaches depended on the majority's judgements may not accurately discern low-quality resources. Another limitation is that the current model does not optimally utilise the instructors spot-checking in students' reliability inference.

RELATED WORK

In what follows, we provide examples of learnersourcing systems that incorporate students to create learning resources and those that use it for content evaluation. Then, we discuss a number of consensus approaches in group decision making and peer review. Finally, we present a brief review of commonly used trust propagation approaches in social networks.

Learnersourcing

A growing number of platforms primary use learnersourcing for students' potential to generate novel learning content. For example, Crowdy is a platform in which students collaboratively develop subgoal labels on a set of instructional videos by answering reflective surveys [42]; AXIS uses students to generate, revise, and evaluate explanations for problem-solving tasks [44]; PeerWise is an online platform that empowers students to author, answer, and evaluate multiple-choice questions [10]; CodeWrite offers practice support and peer review for Java programming using exercises created by students [9]; StudySieve enables students to create free-response to questions designed by their peers [27]; UpGrade sources student open-ended solutions to create scalable learning opportunities [41]. We rely on RiPPLE in this study which learnersources generation of learning activities that are used as part of an adaptive educational system [22]. The student-generated content in these systems ranges from multiple-choice and open-ended questions to instructional video annotations and solutions to specific problems.

Beyond creating novel content, several educational systems mainly benefit from learnersourcing for evaluating students works or providing feedback alongside the content creation to ensure the quality of the student-generated resources. For example, Mechanical TA, an automated peer review system, aims to advance review quality by involving teaching assistants who evaluate reviews of novices and spot check that of experienced students [46]; Dear Beta and Dear Gamma, web applications, learnersource personalized hints involving students in hint creation on their own work and on that of their peers [14]; Aropä, an online system, facilitates peer review activity based on a rubric provided by academics to enable students to upload assignments, write reviews on peer submissions and view the feedback given on their own works [32]; PeerScholar, an automated online tool for writing and critical thinking assessments, is designed to assist instructors in managing student assignments [31]; CrowdGrader enables students to submit, review and grade homework, and receive feedback on the quality of their assignment and reviews [7]; edX, a MOOC platform, pairs students randomly to review their submissions in a peer assessment system to facilitate education in tasks such as writing and design, which are challenging to assess automatically [34]; Peergrade, a web-based peer assessment tool, attempts to improve the feedback quality by an intelligent allocation of reviewers and automatic flagging for instructor moderation [45]. Peer evaluation is also used alongside the content creation in the learnersourcing platform used in our study (RiPPLE) to control the quality of the student-generated resources. By and large, existing peer-evaluation systems commonly utilise summary statistics methods like mean aggregation to integrate student decisions on their peer's work [31, 46, 32, 45]. While these methods benefit ease and explainability, as demonstrated in previous work (e.g., [6, 11]) and the results in this paper, they perform poorly to evaluate learnersourced content.

Consensus approaches in peer reviews

A challenging task in learnersourcing is to infer a final decision on a student-generated resource's quality and reaching a consensus from multiple peer reviews. It is challenging because students have diverse labelling behaviours, not known at inference time. In crowdsourcing literature, the problem of optimal integration of the crowdsourced decisions in the absence of a ground truth has been studied under the general terms of truth

inference or consensus approaches [47, 17]. Some of these approaches are established on Bayes' theorem for opinion aggregation, combining estimation, and truth inference [12, 24]. These approaches are mostly used for consensus over categorical labels in crowdsourcing. Various techniques are utilised to aggregate estimated ordinal labels such as ratings. For example, a technique presented in [4] uses the posterior distribution to identify the quality of sampled labels by optimising the separating width among classes. An ordinal labelling model proposed by [48] aggregates noisy ordinal labels from a crowd by representing worker ability and item difficulty.

One of the well-adopted weighted aggregation methods in consensus approaches is Expectation-Maximization (EM) that estimates the quality of responses to infer the reliability [29]. EM has been widely utilised in different tasks in diverse fields, from text classification and data clustering to chemical systems and medicine [30, 3, 2, 5]. Whitehill et al. [43] developed an aggregation model using EM that combines subjective ratings from a set of students into an aggregate quality score for each resource. They also posited that the estimated quality rating was a better predictor of student average learning gains than the test scores in their study. EM commonly iterates over the entire dataset until convergence of estimated parameters in an offline manner. In its current version, RiPPLE has aimed to adapt EM for an online setting but has not worked well. This study has explored the possibility of employing a graph-based trust propagation approach to use the instructors' contribution optimally in the inference model.

Trust Propagation

The success and growth of crowdsourcing platforms such as Wikipedia, Stackoverflow, and Amazon Mechanical Turk, and social networks such as Facebook and Twitter develop the immense scale interactions between users and information overload. The potential of anonymity inherent in these large scale networks allows malicious behaviour such as spamming and providing false or misleading information, which raises the need for trust evaluation. Trust between users promotes their interaction in the networks with different purposes, including recommendation, group decision making, or assessment in diverse applications such as e-health, e-commerce, and e-learning [37]. This trust can also propagate through the users' connections, offering them a perception of the given information's quality based on their previous interactions. The well-known balance theory can explain the trust propagation concept, where people are more likely to interact with friends of friends than unfamiliar individuals and regulate their preference and beliefs on other objects (e.g., service or products) based on the relationships with their peers [49]. A review of the literature on trust propagation and opinion dynamics in social networks and group decision-making frameworks is provided in [37].

Most trust propagation works aim to detect spammers or untrustworthy users in a social network such as e-commerce platforms. For example, a review graph model is introduced by [39] to identify untrustworthy online-store-reviewers using an iterative approach in an offline manner. We adapt their approach to estimating users' reliability in a live system. Guha et al. [15] introduce a trust propagation and also include distrust in their framework. Following their approach, we consider both agreement and disagreement between users to compute the similarity and propagate users' reliability. Besides, in most existing algorithms, a constant value of reliability is usually inferred for an individual and used throughout the system to weigh all contributions. However, we believe students' reliability and or competency would gradually change throughout the training and experience. In an educational context, the aim is to estimate students' reliability and then identify those who need help or guidance to improve their skill in a particular field or topic.

PROBLEM DEFINITION

Notations and a formal definition of the problem under investigation are presented in this section. Table 1 gives a summary of the notations and their definitions that are used in the computations of the proposed model.

	Inputs
U_N	A set of users $\{u_1 \dots u_N\}$ who are enrolled in a course, where u_i is an arbitrary user.
Q_M	A repository of learning resources $\{q_1 \dots q_M\}$ available within the system, where q_j is an arbitrary resource.
$R_{N\times M}$	A two dimensional array in which $1 \leq r_{ij} \leq 5$ shows the decision rating given by user u_i to resource q_j.
Γ	A threshold of quality for moderated resources.
ρ	The initial value of the score for all users.
	Function 1: Decision Making
W_N	A set of decision weights $\{w_1 \dots w_N\}$ in which $0 \leq w_i \leq 1$ infers the decision weight of a user u_i.
Ω_M	A set of cumulative decision weights $\{\omega_1 \dots \omega_M\}$ in which ω_j shows the cumulative decision weights of trustworthy users who moderated resource q_j.
P_N	A set of a total number of positive ratings $\{p_1 \dots p_N\}$ in which p_i shows the number of resources that u_i approved and contributed to decision making.
N_N	A set of a total number of negative ratings $\{n_1 \dots n_N\}$ in which n_i shows the number of resources that u_i has rejected and contributed to decision making.
	Function 2: Updating Scores
$F^G_{N\times M}$	A function where f^G_{ij} determines the goodness of the rating provided by u_i for q_j.
F^D_M	A function where f^D_j determines the "discrimination" value of resource q_j.
$A_{N\times N}$	A two dimensional array in which $a_{ik} \geq 0$ shows the number of agreements between user u_i and user u_k.
$D_{N\times N}$	A two dimensional array in which $d_{ik} \geq 0$ shows the number of disagreements between user u_i and user u_k.
$S_{N\times N}$	A two dimensional array in which $-1 \leq s_{ik} \leq +1$ shows the similarity value between user u_i and user u_k.
Φ_N	A set of users' score $\{\phi_1 \dots \phi_N\}$ in which ϕ_i shows the score of user u_i.
	Function 3: Reliability Propagation
Λ_N	A set of users' reliability $\{\lambda_1 \dots \lambda_N\}$ in which $-1 \leq \lambda_i \leq +1$ infers the reliability of user u_i.
T_N	A set of users' trustiness value $\{t_1 \dots t_N\}$ in which t_i infers the value of trustiness on user u_i.
$F_{N\times N}$	A two dimensional array in which f_{ik} shows the belief value of user u_i on user u_k.
	Output
$\hat{R}_M$	A set of M ratings $\{\hat{r}_1 \dots \hat{r}_M\}$ where each rating $1 \leq \hat{r}_j \leq 5$ shows the quality of resource q_j.

Table 1: Descriptions of the notations used in the problem definition and the presented approaches.

Let $U_N = \{u_1 \dots u_N\}$ denote a set of users who are enrolled in a course, where $u_1 \dots u_{N'}$ represent the students and $u_{N'+1} \dots u_N$ represent the instructors of the course $(N' \leq N)$. Let $Q_M =$

$\{q_1 \dots q_M\}$ denote a set of learning resources, where q_j refers to an arbitrary resource. Let $R_{N\times M}$ capture users' decision ratings on evaluation of learning resources where $1 \leq r_{ij} \leq 5$ shows the decision rating given by user u_i to resource q_j.

Given $R_{N\times M}$, our aim is to infer a set of quality ratings $\hat{R}_M = \{\hat{r}_1 \dots \hat{r}_M\}$ in which $1 \leq \hat{r}_j \leq 5$ refers to the inferred quality of q_j.

MODERATION GRAPH MODEL

Figure 5 illustrates how the problem can be formulated in the form of a moderation graph, which consists of four kinds of node – students, decision ratings, resources, and instructors.

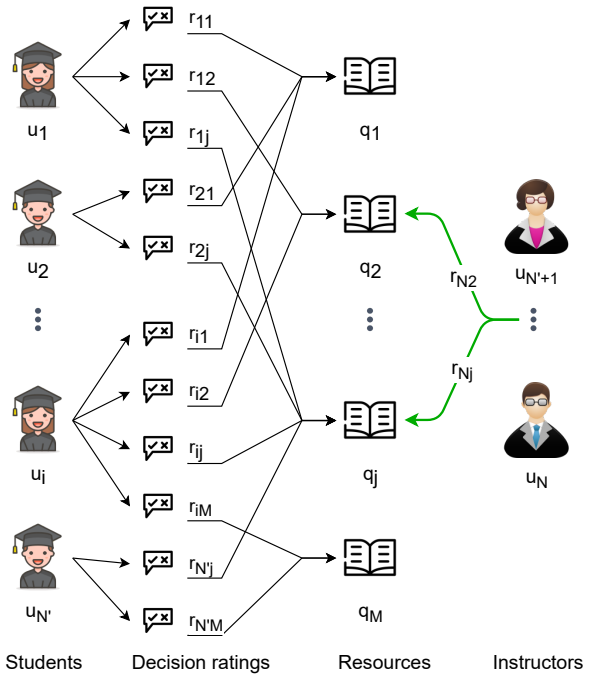

Figure 5: Moderation graph.

In our approach for inferring a student u_i's decision weight w_i, and a resource q_j's quality $\hat{r}_j$, we make use of the following assumptions and definitions.

- We define a two dimensional array $A_{N\times N}$ where $a_{ik} \geq 0$ shows the number of agreements between user u_i and user u_k. Similarly, we define a two dimensional array $D_{N\times N}$, in which $d_{ik} \geq 0$ shows the number of disagreements between user u_i and user u_k. Information on agreements $A_{N\times N}$ and disagreements $D_{N\times N}$ between users is used to define a two dimensional array $S_{N\times N}$ in which $-1 \leq s_{ik} \leq +1$ shows the similarity value between user u_i and user u_k.

- We define a one dimensional array Λ_N where $-1 \leq \lambda_i \leq +1$ represent the reliability of user u_i. Instructors are considered to be extremely reliable and their reliability set to maximum (i.e., $\forall i > N' : \lambda_i = 1$), and students would enter into the system with an initial reliability value (i.e., $\forall i \leq N' : \lambda_i \simeq 0.1$).

- We define a one dimensional array T_N, where t_i infers the trustiness of a u_i. The trustiness of user u_i is computed as $t_i = \sum_{k=1}^{N} \lambda_k \times s_{ik}$, which depends on the amount of similarity to other reliable and non-reliable user.

- Only decisions made by trustworthy users are contributed to inferring the quality of resources. Here, we consider users trustworthy if their reliability, trustiness, and instructor belief are above a threshold.

- A resource q_j with an inferred quality of $\hat{r}_j$ is approved and used within the system if $\hat{r}_j \geq \Gamma$ and rejected otherwise.

Algorithm 1 presents the high-level pseudo-code of our proposed peer review approach using a graph-based trust propagation approach[1]. This algorithm consists of three main stages– Decision Making, Updating Scores, and Reliability Propagation. Figure 6 shows the main stages in the propagation network model used in this study. First, Given a non-moderated resource q_j and decision ratings $(r_{1j}, \dots, r_{kj})$ from students $(u_1, \dots, u_k)$, system starts the decision making stage when enough reliable moderators have evaluated the given resource. Then, $\hat{r}_j$, the inferred quality of q_j, is used to calculate the gained score of the contributed student-moderators. Finally, all other students connected to $(u_1, \dots, u_k)$ and have less reliability value than them would receive an updated score. For example, u_l, who is connected to u_1, u_4, *and* u_k, would receive a score of φ_l from the user u_k who has a bigger value of reliability than the others in this group.

Algorithm 1: Main procedure of the peer review process

Input : U_N, Q_M, $R_{N\times M}$, Γ, ρ, α, κ
Output : $\hat{R}_M$

```
initialization();
while nonModeratedResource do
    if newModeration then
        receivedModeration(u_i, q_j)
    end
end
Function receivedModeration(u_i, q_j)
    U_{q_j} ← ∀u_k ∈ r_{kj} ≠ 0 ;        /* Set of users moderated q_j */
    (ω_j, r̂_j) ← computeQuality(u_i, q_j, U_{q_j}) ;  /* see Function 1 */
    if ω_j < τ_Ω then
        return ∅ ;          /* not ready for decision making */
    else
        updateScore(U_{q_j}) ;                  /* see Function 2 */
        propagateReliability(U_{q_j}) ;         /* see Function 3 */
        return r̂_j;                        /* Inferred quality */
    end
end
```

In Algorithm 1, a set of variables is first defined in initialization to be used in the moderation process. We devote an initial amount of score (i.e., $\phi_N \leftarrow \rho$) to all students. This set of scores is transformed into an initial set of users' reliability Λ_N. The Similarity matrix $S_{N\times N}$ would be initialized with an identity matrix (i.e.,$\forall i, j \in U_N : s_{ij} = 1 \; if \; i = j$; $s_{ij} = 0 \; if \; i \neq j$) as an indicator of self similarity. Then, the Trustiness vector T_N, as a product of the similarity matrix and reliability

[1]The source code (in R) is available at: https://github.com/ali-darvishi/Quality-via-Trust

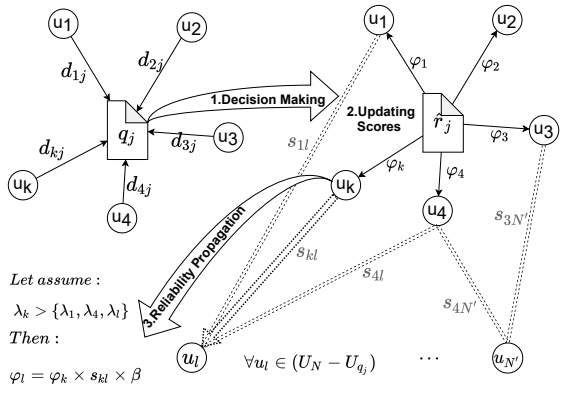

Figure 6: Propagation network.

vector (i.e., $T_N = S_{N\times N} \times \Lambda_N$), would be then initialized with the initial values of reliability. After initialization, when a moderation on a newly generated resource receives, the system starts the process of inferring the quality of the moderated resource and the reliability of users using the following stages:

1. Decision Making

This process depends on both the cumulative decision weights of the resource ω_j and the trustworthiness of the users who moderated the resource q_j. As shown in Function 1, in the set of users who moderated the current resource (i.e., U_{q_j}), only those with positive trustworthiness can contribute to the decision making process. To have a positive trustworthiness, a student-moderator should satisfy three criteria: (1) Reliability (i.e., $\lambda_k > \tau_\Lambda$), (2) Trustiness (i.e., $t_k > \tau_T$), and (3) Instructors' belief (i.e., $f_{kN} \geq \tau_B$). We discuss each of these variables later in the Reliability Propagation stage. Then, the decision weight ($0 \leq w_i \leq 1$) of trustworthy users is computed using Function *computeDecisionWeight* (Line 21-Function 1). This function considers the frequency of approving or rejecting a resource and the user reliability to adjust the decision weight of a user. To calculate and keep track of the number of times a user has approved or denied a resource and contributed to decision making, p_i or n_i are first updated based on the quality threshold Γ. Driven by credits, students tend to game the system by overrating the resources as the most prevalent answer (e.g. a high rating of 4 or 5) to gain more scores in the system. Therefore, ζ degrades the decision weight of those users who only approve resources. On the other hand, ζ would escalate the decision weight of trustworthy users who deny a resource after several reliable moderations. At the end of the decision making process, Function *computeQuality* returns both the inferred quality of the resource ($\hat{r}_j$) and the cumulative decision weights (ω_j) from the trustworthy users. If ω_j is less than a threshold τ_Ω, it means the resource has not yet received enough moderations from trustworthy users, and it is not ready for making a decision. Therefore, the system waits to receive moderation from other users. Otherwise, if we have enough trustworthy users in the moderation (i.e, $\omega_j \geq \tau_\Omega$), Function *receivedModeration* updates scores and propagates reliability, as discussed below, and then returns the inferred quality ($\hat{r}_j$).

Function 1: Decision Making

```
1  Function computeQuality(u_i, q_j, U_{q_j})
2      if (i > N') then                                  /* u_i is an instructor */
3          r̂_j ← re_ij;                                  /* instructor decision on q_j */
4          ω_j ← τ_Ω;   /* cumulative decision weight = threshold */
5      else                                              /* u_i is a student */
6          if (r_ij ≥ Γ) then
7              p_i ← p_i + 1 ;          /* Increment #positive ratings */
8          else
9              n_i ← n_i + 1 ;          /* Increment #negative ratings */
10         end
11         ω_j ← 0; r̂_j ← 0;
12         for ∀u_k ∈ U_{q_j} do
13             if ((λ_k > τ_Λ)&(t_k > τ_T)&(F[k,N] ≥ τ_B)) then
                                                         /* u_k is trustworthy */
                   w_k ← computeDecisionWeight(u_k, q_j);
14                 r̂_j ← r̂_j + w_k × r_kj;  ω_j ← ω_j + w_k;
15             end
16         end
17         r̂_j ← r̂_j / ω_j;
18     end
19     return (ω_j, r̂_j)
20 end
21 Function computeDecisionWeight(u_k, q_j)
22     if (r_kj ≥ Γ) then
23         θ ← (p_k + n_k)/p_k;
24     else
25         θ ← (p_k + n_k)/n_k;
26     end
27     ζ ← 0.1/(0.1 + e^{−θ/2});  w_k ← λ_k × ζ;
28     return w_k
29 end
```

2. Updating Scores

Function 2 shows how we update the students' scores. The score ϕ_i of user u_i would change based on the "goodness" of the user's decision rating r_{ij} and the "discrimination" value of resource q_j. The amount of this change $\varphi_{ij} = f_{ij}^G \times f_j^D$ is computed using two functions – $F_{N\times M}^G$ and F_M^D as shown in Equations (1) and (2). f_{ij}^G estimates the "goodness" of u_i's rating on q_j using the threshold Γ and Gaussian functions at value $(r_{ij} - \hat{r}_j)$ with centre (μ) 0 and variance (σ) 1 as:

$$f_{ij}^G = \begin{cases} \frac{\kappa\Upsilon_j e^{-(r_{ij}-\hat{r}_j)^2/2}}{\sqrt{2\pi}} & r_{ij}, \hat{r}_j \geq \Gamma \vee r_{ij}, \hat{r}_j < \Gamma, \\ \frac{\kappa\Upsilon_j e^{-(r_{ij}-\hat{r}_j)^2/2} - \kappa\Upsilon_j}{\sqrt{2\pi}} & Otherwise. \end{cases} \quad (1)$$

Where $1 \leq \kappa \leq \rho$ adjusts the maximum value of reward/punishment that a user can achieve per each moderation, and $0 \leq \Upsilon_j \leq 1$ is the maximum value of the reliability among the users who moderated resource q_j. In other words, $F_{N\times M}^G$ will grant a large positive value (reward) to users if their rating r_{ij} and the final inferred resource quality $\hat{r}_j$ have the same polarity based on the threshold Γ (in terms of approved or denied) and r_{ij} is close to $\hat{r}_j$ and will be a large negative value (punishment), otherwise. Figure 7a shows the reward/punish function used in this study ($\kappa = 100$). f_j^D determines the "discrimination" value of resource q_j using another Gaussian function at value $\hat{r}_j$.

$$f_j^D = \frac{1 - e^{-(\hat{r}_j - \mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}} + \varepsilon \quad (2)$$

where $\mu = \Gamma = 3$ in our 5-scale rating system, ε is a small value that determines the minimum discrimination for the resources with inferred quality $\hat{r}_j = 3$. Resources with a $\hat{r}_j$ close to Γ are considered "close calls" where the system has less confidence in the decision. Informally, F_M^D determines how much we can rely on a resource to distinguish a good moderation from the bad one. Grounded on common sense, we assume that with an inferred quality close to $\Gamma = 3$ (i.e., minimum satisfactory quality value), it is hard to judge the quality of moderation, and the reward/punishment value will be diminished by F_M^D. On the other hand, if an inferred quality is close to 5 (i.e., outstanding quality) or close to 1 (i.e., poor quality), F_M^D will consider a higher impact for that resource to update the scores. The variance of this Gaussian function is selected as $\sigma = 0.455$ to have the maximum discrimination value $Max(f_j^D) \simeq 1$ for $\hat{r}_j = 1$ or 5 and ε is selected as 0.1 to have the minimum discrimination value (i.e., $Min(f_j^D) = 0.1$) for $\hat{r}_j = 3$. Figure 7b illustrates the final resource discrimination function used in this study.

Also, to calculate similarities between users, the numbers of their direct agreements and disagreements based on the threshold of resource quality (Γ) are counted. Two users (e.g., u_l and u_k), who contributed on the moderation of resource q_j, agree if their decision ratings are both less than the threshold or greater than or equal to the threshold (i.e., $r_{lj}, r_{kj} \geq \Gamma \vee r_{lj}, r_{kj} < \Gamma$), and disagree otherwise. A two dimensional array $A_{N\times N}$ records the number of agreements between all users, and $D_{N\times N}$ records the number of disagreements. Here, following the approach by [39], a logistic function is used to translate the agreement and disagreement matrices to a similarity matrix ($S_{N\times N}$ between users as shown in Equations (3):

$$s_{lk} = \frac{2}{1 + e^{-(a_{lk} - d_{lk})}} - 1 \quad (3)$$

where $-1 \leq s_{lk} \leq +1$, shows the similarity value between user u_l and user u_k. A, D, and S matrices are symmetric (e.g., $A^T = A : a_{lk} = a_{kl}$) and sparse.

3. Reliability Propagation

Here, we aim to propagate the reliability from those who achieved scores by contributing to the moderation of a resource to other users connected to them in the system network. In this scenario, users' score would be changed based on the quality of their own collaborations and also their peers who are directly connected to them as a result of their collaboration in the previous moderations. Function 3 shows the main steps for the propagation process. Two main criteria are needed to be satisfied by a user (e.g., u_l) other than the current moderators to achieve a score,: (1) u_l should directly connected to one or more users from the set of current moderators (i.e., $\sum_{u_k \in U_{q_j}} |s_{lk}| \neq 0$), and (2) u_l only receives a score from the most reliable moderator (e.g., u_k) who must have a greater reliability than the receiver (i.e., $\lambda_k > \lambda_l$). The amount of updating score φ_l is calculated by the similarity value between two users and a discount factor for propagation steps ($\beta = 0.5$).

Function 2: Updating Scores

```
Function updateScore(U_{q_j})
    Υ ← Max_{u_k ∈ U_{q_j}} (λ_k) ;        /* Max reliability of users in U_{q_j} */
    for ∀u_k ∈ U_{q_j} do
        φ_k ← computeReward (r_kj, r̂_j, κ, Υ);   using Equation (1),(2)
        ϕ_k ← ϕ_k + φ_k;
        for ∀u_l ∈ (U_{q_j} − u_k) do
            if ((r_lj ≥ Γ) & (r_kj ≥ Γ)) ‖ ((r_lj < Γ) & (r_kj < Γ)) then
                a_lk = a_lk + 1 ;          /* Increment #agreement */
            else
                d_lk = d_lk + 1 ;          /* Increment #disagreement */
            end
        end
    end
    S_{N×N} ← updateSim (A_{N×N}, D_{N×N});   using Equation (3)
end
```

Function 3: Reliability Propagation

```
Function propagateReliability(U_{q_j})
    for ∀u_l ∈ (U_N − U_{q_j}) do
        max_Λ ← λ_l;
        φ_l ← 0;
        for ∀u_k ∈ U_{q_j} do
            if s_lk ≠ 0 ;          /* u_k is directly connected to u_l */
            then
                if (λ_k > max_Λ) then
                    max_Λ ← λ_k;
                    φ_l ← β × φ_k × s_lk;
                end
            end
        end
        ϕ_l ← ϕ_l + φ_l
    end
    Λ_N ← updateReliability(Φ_N, α);   using Equation 4
    T_N ← S_{N×N} × Λ_N ;                  /* Update trustiness */
    F_{N×N} ← S_{N×N} × S_{N×N} ;          /* Update belief */
end
```

The user score can turn into a very high positive point or even a very low negative one. Therefore, to have a more meaningful value in user decision weight calculation, this score is then transformed to a reliability value λ_k for user u_k in the range between $[-1, +1]$ using a logistic function as shown in Equation (4):

$$\lambda_k = \frac{2}{1 + e^{-\alpha\phi_k}} - 1 \quad (4)$$

Where $0 < \alpha \leq 1$ and is selected based on the range of score Φ_N. Here, we set $\alpha = 0.001$ so that the reliability score would saturated for very high score (e.g., > 4000) as shown in Figure 7d. Then, users' trustiness would be then updated using the users' reliability vector (Λ_N) and similarity matrix ($S_{N\times N}$). The trustiness of u_k is the aggregation of all users' similarity

Data	INFS1			INFS2			NEUR1			NEUR2			COMP		
	Total	Selected	Spot-Checks	Total	Selected	Spot-Checks	Total	Selected	Spot-Checks	Total	Selected	Spot-Checks	Total	Selected	Spot-Checks
# Resources	2,095	911	112	1,835	921	41	4,875	4,851	145	2,803	2,757	303	1,195	926	93
# Students	389	378	250	385	372	127	532	526	304	535	527	483	300	295	191
# Moderations	6,991	4,321	508	6,182	4,158	165	28,152	28,071	728	30,642	30,512	3,131	5,330	4,882	386

Table 2: Data sets details.

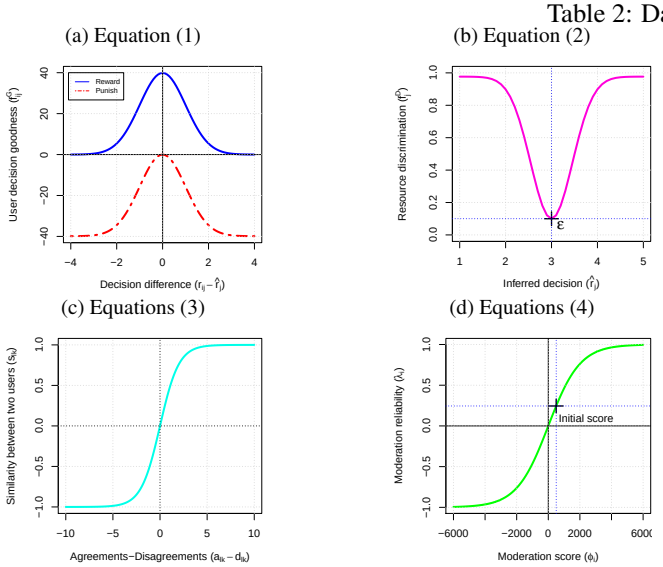

Figure 7: Scoring functions: (a) goodness of user u_i's decision on resource q_j, (b) discrimination of resource q_j, (c) similarity between user u_l and user u_k, and (d) reliability of user u_i.

against u_k multiple by their reliabilities (i.e., $t_k = \sum_{i=1}^{N} s_{ik} \times \lambda_i$). Notice that both similarity and reliability can be either positive or negative. Hence, if a user's similarity towards an unreliable user is negative, the trustiness value increases.

Finally, we introduce a belief matrix with one step propagation of the similarity matrix, following the work by [15]. The similarity matrix is relatively sparse as students only interact with a small group of their peers during the moderation process. By propagating the similarity values, we can have an estimation for unseen values. Consequently, using these estimated values, we are able to predict the instructors' " belief" in all students, including those who had no history of interactions on the spot-checked resources. The belief of instructors on user u_k is essentially the aggregation of all users' similarity against u_k multiple by their similarity with the instructor (i.e., $f_{kN} = \sum_{i=1}^{N} s_{ik} \times s_{iN}$). These three values of λ_k, t_k, and f_{kN} are used to determine the trustworthiness and to determine whether user u_k's rating r_{kj} can participate in quality inference of resource q_j as discussed earlier in Decision Making stage. Due to this network's dynamic nature and regular update of users' scores in the live system, the thresholds – τ_Λ, τ_T, and τ_B – are also set dynamically as the current first quantile of users' reliability (λ_N), users' trustiness (T_N), and instructor's belief vector ($\{f_{1N} \dots f_{N'N}\}$), respectively.

Instructors' participation in the moderation process allows us to have ground truth on the spot-checked resources' inferred quality. We use this opportunity to readjust students' reliability scores by comparing their decisions to the expert's decision in the "Updating Scores" and "Reliability Propagation" procedures. We set the instructors' reliability score as the maximum to escalate the impact of the instructor decision in the calculation of score and in the propagation of the reliability. Therefore, for $\Upsilon \leftarrow Max(\lambda_i)$ in Function 2 that is set as the maximum value of the reliability among the users who moderated resource q_j, we can have $\Upsilon \leftarrow 1$ as the reliability value of the instructor and also the discount factor would set $\beta = 1$ which provides a more significant change in scores than cases with student-moderators only where we set $\beta = 0.5$.

EVALUATION

Data Sets. The data sets used in this study are obtained from piloting RiPPLE in five course offerings at The University of Queensland. Two of the offerings (Introduction to Information Systems (code: INFS1) and The Brain and Behavioural Sciences (code: NEUR1)) ran in Semester 1, and three of the offerings Introduction to Information Systems (code: INFS2), The Brain and Behavioural Sciences (code: NEUR2), and Artificial intelligence (code: COMP) ran in Semester 2. Overall information about the collected data from these offerings is presented in Table 2. RiPPLE collects and records all submitted moderations on student-generated resources. However, we only use resources with at least 4 moderations for evaluation as recommended by [26]. They suggest that the number of annotations per instance should not be less than 3 annotations per instance to conduct a fair comparison between proposed methods and baselines.

RiPPLE flags and prioritises a limited number of resources to be inspected by instructors based on several criteria such as disagreements between student-moderators, questionable distractors, reported resources, and low effective resource based on student answers. The spot-checked resources are then used to evaluate the presented methods' performance in the quality inference according to the instructors' decisions.

Models for comparison

Baselines. We take a set of well-known aggregation approaches for reaching consensus as baselines including *Majority Vote*, *Mean*, *Median*, and *Debiased Mean*. These approaches only use the student provided numerical ratings (i.e., $R_{N\times M}$) and assume an equal weight for all moderators (i.e., $W_N \leftarrow 1$). Majority vote, which is viewed as a fair approach in categorical applications, takes the decision given by the majority as the final result. Mean is the most simple aggregate statistics that uses the average of all provided ratings as the outcome (i.e., $\hat{r}_j = \sum_{i=1}^{k} r_{ij}/k$ where k is the number of moderators). Median is another common simple aggregation method that also considers all moderators equally and infers the quality of a resource as $\hat{r}_j = \text{Median}(r_{1j}, \dots r_{kj})$. Debiased Mean remove the bias of users in decision making. First, the average

Model	INFS1				INFS2				NEUR1				NEUR2				COMP			
	TPR	TNR	AUC	ACC	TPR	TNR	AUC	ACC	TPR	TNR	AUC	ACC	TPR	TNR	AUC	ACC	TPR	TNR	AUC	ACC
Majority Vote	0.96	0.15	0.56	0.68	**0.94**	0.04	0.49	0.44	0.99	0.06	0.53	0.68	**1.00**	0.02	0.51	0.63	0.98	0.02	0.50	0.52
Mean	0.97	0.21	0.59	0.71	**0.94**	0.09	0.52	0.46	**1.00**	0.08	0.54	0.70	**1.00**	0.03	0.52	0.63	**1.00**	0.02	0.51	0.53
Median	**0.99**	0.15	0.57	0.70	**0.94**	0.04	0.49	0.44	**1.00**	0.08	0.54	0.70	**1.00**	0.00	0.50	0.62	**1.00**	0.00	0.50	0.52
Debiased Mean	0.95	0.15	0.55	0.67	**0.94**	0.13	0.54	0.49	**1.00**	0.08	0.54	0.70	**1.00**	0.03	0.51	0.63	0.98	0.07	0.52	0.54
Current Model	0.95	0.33	0.64	0.73	**0.94**	0.17	0.56	0.51	**1.00**	0.10	0.55	0.70	0.99	0.03	0.51	0.63	0.96	0.13	0.55	0.56
Graph Model	0.81	**0.62**	**0.71**	**0.74**	**0.94**	**0.57**	**0.75**	**0.73**	0.88	**0.52**	**0.70**	**0.76**	0.94	**0.43**	**0.68**	**0.75**	0.90	**0.22**	**0.56**	**0.57**

Table 3: Comparing the inferred quality using baselines and graph models with the instructors' spot-checks evaluated with True Positive Rate (TPR), True Negative Rate (TNR), Area Under the Curve (AUC), and Accuracy (ACC) for moderation decisions.

decision rating of user u_i is computed as $\bar{r}_i$. Next, the average decision rating of all users is computed as $\bar{r} = \sum_{i=1}^{N} \bar{r}_i / N$. Then, the bias of user u_i is computed as $b_i = \bar{r}_i - \bar{r}$. Finally, the quality of a resource in debiased mean is inferred by removing moderators' bias as $\hat{r}_j = \sum_{i=1}^{k} (r_{ij} - b_i)/k$.

Current model. A weighted aggregation approach based on the well-adopted Expectation-Maximisation (EM) technique is currently applied in our educational system. The resources' quality and the users' reliability are highly dependant on each other where knowing the true value of one set can be used to estimate the other one. However, in the absence of ground truth, the current model uses the following steps to infer the quality of the resource and adjust the reliability of the users based on the EM method. First, the scores of all students set to an initial value such as ρ. Next, in the expectation step, the quality of a resource $\hat{r}_j$ is inferred based on the provided ratings $r_{1j}, \ldots r_{kj}$ on q_j and the current values of students' score $\phi_1, \ldots \phi_k$ as their decision weights (i.e., $\hat{r}_j = \sum_{i=1}^{k} \phi_i \times r_{ij} / \sum_{i=1}^{k} \phi_i$). Then, in the maximisation step, the users' scores $\phi_1, \ldots \phi_k$ are updated based on the "goodness" of their rating r_{ij} in comparison with the inferred quality $\hat{r}_j$ using the height of a Gaussian function with centre 0 as $\varphi_i = (2\kappa e^{-(r_{ij}-\hat{r}_j)^2/(2\sigma^2)} - \kappa)/(2\sigma\sqrt{2\pi})$, $\phi_i = \phi_i + \varphi_i$, where $1 \leq \kappa \leq \rho$ and the variance $0 \leq \sigma \leq 1$ adjust the value of reward/punishment that a user can achieve as a score (here, we set $\rho = 1000$, $\kappa = 100$ and $\sigma = 0.7$ across all courses).

Results and Analysis

Table 3 shows the performance of the presented consensus inference models by comparing the inferred to the expert ratings. We report True Positive Rate (TPR), True Negative Rate (TNR), Area Under the Curve (AUC), and Accuracy (ACC) for the binary classification task where a resource is automatically moderated as acceptable or unacceptable to be shown to students, which is one of the main goals of this work. We compared baseline models, the current implemented model in RiPPLE based on EM, and proposed network-based models. In both current and graph models, we first calculate the current decision weights of students based on their gained scores from previous moderations to infer the quality of the resource under moderation. This inferred quality is separately recorded for evaluation and comparison with instructors' spot-checks. Then, if an instructor also moderated the resource (e.g., q_j), the inferred quality would be set to the instructor's decision (i.e., $\hat{r}_j = r_{Nj}$) and students' scores (Φ_N) would be updated based on this decision.

The baseline approaches (Majority Voting, Mean, Median, and Debiased Mean) assume an equal contribution to all student provided numerical ratings for moderation. These approaches are commonly used in various applications; however, based on our analysis and the reported results, they do not work well for reaching a robust consensus on peer-reviews of student-generated content. Results in Table 3 indicate that Baselines have the highest value of TPRs (INFS1: 0.99, INFS2: 0.94, and NEUR1-2&COMP: 1.00) and the lowest value of TNRs (INFS1: 0.15, INFS2: 0.04, NEUR1: 0.06, and NEUR2 & COMP: 0.00) across all courses. Taken together, these results suggest that most students tend to overrate, resulting in approval of low-quality content in the baseline models mentioned above.

The results obtained from the currently implemented consensus approach in RiPPLE indicate some improvements in the TNR values in almost all courses (INFS1: 0.33, INFS2: 0.17, NEUR1: 0.10, and COMP: 0.13) except for (NEUR2 0.03). This method reweights students' contribution using the EM method based on how well their rating aligns with the inferred rating. However, with a slight decline in TPR values, no significant and consistent improvements were observed in most cases in terms of AUC values compared to the best in baselines (INFS1: 0.59 ↗ 0.64, INFS2: 0.54 ↗ 0.56, NEUR1: 0.54 ↗ 0.55, NEUR2: 0.52 ↘ 0.51, and COMP: 0.52 ↗ 0.55). Therefore, these findings indicate that the current model is still biased toward the majority who overrate.

The use of the graph model substantially improves the TNRs (INFS1: 0.62, INFS2: 0.57, NEUR1: 0.52, NEUR2: 0.43, and COMP: 0.22) across all courses. Updating scores and propagating reliabilities using instructors' decision in the graph model provides additional improvements in the results in terms of AUC scores compared to the current model (INFS1: 0.64 ↗ 0.71, INFS2: 0.56 ↗ 0.75, NEUR1: 0.55 ↗ 0.70, NEUR2: 0.51 ↗ 0.68, and COMP: 0.55 ↗ 0.56). This finding is promising, suggesting that the graph-based trust propagation model works well by identifying the reliable and trustworthy student-moderators with a little help of instructors' intervention and supervision.

Table 3 also demonstrates that commonly used metrics such as accuracy (ACC) or error rate are not sensitive to long-tail distributions. For example, the median model has TNR values of 0.08 and 0.00 in NEUR1 and NEUR2, respectively, which means it fails to identify poor-quality resources. However, the accuracy metric (ACC) shows a performance of 0.70 and 0.62 for the median model in NEUR1 and NEUR2, respectively, which results from a higher proportion of the positive class (approved resources) and might be misleading. In contrast, AUC shows the performance of 0.54 and 0.50 for the me-

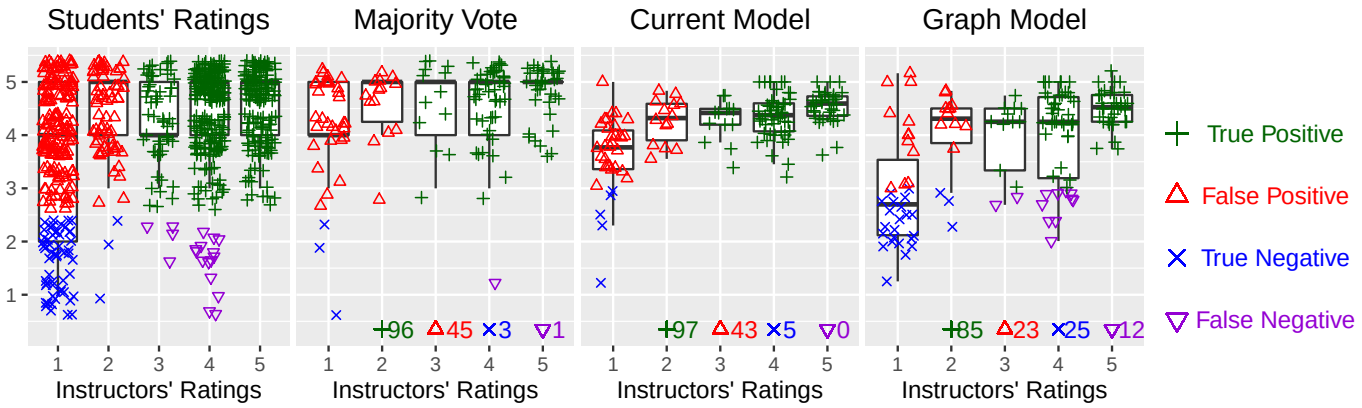

Figure 8: Outcomes of a baseline model (Majority Vote), the current model (EM) and the graph model in NEUR1

dian model in NEUR1 and NEUR2, respectively, which better shows the median model's failure in identifying the poor- from high-quality resources. Despite the insensitivity of the ACC in our case as skewed data, the graph model also outperforms all other presented models in terms of accuracy.

Figure 8 offers further insights on applying different models on NEUR1. From the left, the first plot illustrates students' ratings on resources that have been moderated by an instructor. From 145 spot-checks in this course, instructors have approved 97 resources and have rejected the remaining 48. These resources have also received 728 moderations from students. For the approved resources by the instructor, 446 students have submitted moderation where $427 \simeq 96\%$ have also approved (i.e., TP). For the rejected resources by the instructors, 282 students have submitted moderation where only $57 \simeq 20\%$ have rejected (i.e., TN). Therefore, it can be concluded that while the majority of learners usually provides high ratings, a minority (i.e., reliable student moderators) correctly identified the poor-quality resources. The second plot demonstrates the outcome at the resource level for the majority vote model. It illustrates that this model has approved more than 97% of the resources in this evaluation set. This outcome suggests that judgments from a minority of wise moderators are often overturned by decisions from a majority of less-wise or careless moderators in the baseline models using basic aggregation approaches. The current model has moderately improved the outcomes. It has increased the decision weights for the reliable students and has reduced the value of inferred quality for the rejected resources compared to the majority vote. However, no significant improvement has been achieved, and the current model still fails to discern the poor-quality resources.

In contrast, the graph model, as shown on the right side of Figure 8, was able to better identify those minority of wise moderators by using reliability propagation. This model provides a higher agreement between instructors' ratings and the inferred quality from students' ratings. The TN cases have been significantly increased from 5 in the current model to 25 in the graph model at the expense of moderately increasing FNs in a few borderline cases. Employing the trust propagation approach has addressed the issue of reliability inference in the presence of a prevalent answer (high positive rating here) which causes long-tail distributed data and biased users. This approach proved to be able to discern high- and low-quality moderations well, as results indicate. While the results have improved, we have 23 FP cases that have not been identified as poor-quality resources. Interestingly, in 19 of these cases, all student-moderators had approved a resource that had been rejected by instructors. Given that no student-moderator had rejected these resources, any consensus approach would have failed to classify these resources correctly. This result, aligned with previous studies' findings [40, 13], suggests an evident need for instructors to provide oversight during the moderation process.

CONCLUSION AND FUTURE WORK

Employing peer review has been demonstrated to be a viable approach for evaluating the quality of student generated content. However, it poses the problem of "truth inference" as students evaluations may be inaccurate. Reflecting on peer evaluations conducted in a system called RiPPLE, we showed that identifying low quality resources based on peer review is a challenging task as many of the students tend to be easy graders. This paper presented a graph-based trust propagation approach that simultaneously infers the quality of the student-generated content and estimates the reliability of moderators at scale. Results shows that updating users' reliability using the proposed trust propagation approach improves the performance over baseline models and the current model used in the system. A close inspection of the results demonstrate that the model was particularly successful in identifying low quality resources.

There are several interesting directions to pursue in future work. In the current study, the evaluation was conducted based on existing offline data sets. One potential future direction is to conduct the experiment in a live setting where we can investigate the impact of sharing students' reliability scores with them on their behaviour. Another potential future direction is to use trust propagation towards recommending resources for spot-checking. The current spot-checking mechanism is mostly concerned about the quality of resources rather than the reliability of users. We can update the spot checking recommendation algorithm to consider recommending resources that would propagate the maximum amounts of reliability within the network.

REFERENCES

[1] Simon P Bates, Ross K Galloway, Jonathan Riise, and Danny Homer. 2014. Assessing the quality of a student-generated question repository. *Physical Review Special Topics-Physics Education Research* 10, 2 (2014), 020105.

[2] Giuseppe Boccignone, Paolo Napoletano, Vittorio Caggiano, and Mario Ferraro. 2007. A multiresolution diffused expectation–maximization algorithm for medical image segmentation. *Computers in Biology and Medicine* 37, 1 (2007), 83–96.

[3] Paul S Bradley, UM Fayyad, and CA Reina. 2000. Clustering very large databases using EM mixture models. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Vol. 2. IEEE, 76–80.

[4] Guangyong Chen, Shengyu Zhang, Di Lin, Hui Huang, and Pheng Ann Heng. 2017. Learning to aggregate ordinal labels by maximizing separating width. In *International Conference on Machine Learning*. 787–796.

[5] Atefeh Daemi, Yousef Alipouri, and Biao Huang. 2019. Identification of robust Gaussian Process Regression with noisy input using EM algorithm. *Chemometrics and Intelligent Laboratory Systems* 191 (2019), 1–11.

[6] Ali Darvishi, Hassan Khosravi, and Shazia Sadiq. 2020. Utilising Learnersourcing to Inform Design Loop Adaptivity. In *European Conference on Technology Enhanced Learning*. Springer, 332–346.

[7] Luca De Alfaro and Michael Shavlovsky. 2014. CrowdGrader: A tool for crowdsourcing the evaluation of homework assignments. In *Proceedings of the 45th ACM technical symposium on Computer science education*. 415–420.

[8] Paul Denny, Andrew Luxton-Reilly, and Beth Simon. 2009. Quality of student contributed questions using PeerWise. In *Proceedings of the Eleventh Australasian Conference on Computing Education-Volume 95*. 55–63.

[9] Paul Denny, Andrew Luxton-Reilly, Ewan Tempero, and Jacob Hendrickx. 2011. CodeWrite: supporting student-driven practice of java. In *Proceedings of the 42nd ACM technical symposium on Computer science education*. 471–476.

[10] Denis Duret, Rob Christley, Paul Denny, and Avril Senior. 2018. Collaborative learning with PeerWise. *Research in Learning Technology* 26 (2018).

[11] Kinda El Maarry, Ulrich Güntzer, and Wolf-Tilo Balke. 2015. A majority of wrongs doesn't make it right-On crowdsourcing quality for skewed domain tasks. In *International Conference on Web Information Systems Engineering*. Springer, 293–308.

[12] Tiago M Fragoso, Wesley Bertoli, and Francisco Louzada. 2018. Bayesian model averaging: A systematic review and conceptual classification. *International Statistical Review* 86, 1 (2018), 1–28.

[13] Kyle W Galloway and Simon Burns. 2015. Doing it for themselves: students creating a high quality peer-learning environment. *Chemistry Education Research and Practice* 16, 1 (2015), 82–92.

[14] Elena L Glassman, Aaron Lin, Carrie J Cai, and Robert C Miller. 2016. Learnersourcing personalized hints. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. 1626–1636.

[15] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*. 403–412.

[16] John Hamer, Quintin Cutts, Jana Jackova, Andrew Luxton-Reilly, Robert McCartney, Helen Purchase, Charles Riedesel, Mara Saeli, Kate Sanders, and Judithe Sheard. 2008. Contributing student pedagogy. *ACM SIGCSE Bulletin* 40, 4 (2008), 194–212.

[17] Bertrand K Hassani. 2016. The consensus approach. In *Scenario Analysis in Risk Management*. Springer, 39–50.

[18] Neil T Heffernan, Korinn S Ostrow, Kim Kelly, Douglas Selent, Eric G Van Inwegen, Xiaolu Xiong, and Joseph Jay Williams. 2016. The future of adaptive learning: Does the crowd hold the key? *International Journal of Artificial Intelligence in Education* 26, 2 (2016), 615–644.

[19] Peter Kahn and Karen O'Rourke. 2005. Understanding enquiry-based learning. *Handbook of Enquiry & Problem Based Learning* (2005), 1–12.

[20] Evgeny Karataev and Vladimir Zadorozhny. 2016. Adaptive social learning based on crowdsourcing. *IEEE Transactions on Learning Technologies* 10, 2 (2016), 128–139.

[21] Hassan Khosravi, Gianluca Demartini, Shazia Sadiq, and Dragan Gasevic. 2021. Charting the Design and Analytics Agenda of Learnersourcing Systems. In *LAK21: 11th International Learning Analytics and Knowledge Conference*. 32–42.

[22] Hassan Khosravi, Kirsty Kitto, and Williams Joseph. 2019. RiPPLE: A Crowdsourced Adaptive Platform for Recommendation of Learning Activities. *Journal of Learning Analytics* 6, 3 (2019), 91–105.

[23] Hassan Khosravi, Shazia Sadiq, and Dragan Gasevic. 2020. Development and adoption of an adaptive learning system: Reflections and lessons learned. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 58–64.

[24] Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics*. 619–627.

[25] Juho Kim and others. 2015. *Learnersourcing: improving learning with collective learner activity*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[26] Yuan Li, Benjamin IP Rubinstein, and Trevor Cohn. 2019. Truth inference at scale: A bayesian model for adjudicating highly redundant crowd annotations. In *The World Wide Web Conference*. 1028–1038.

[27] Andrew Luxton-Reilly, Beryl Plimmer, and Robert Sheehan. 2010. StudySieve: a tool that supports constructive evaluation for free-response questions. In *Proceedings of the 11th International Conference of the NZ Chapter of the ACM Special Interest Group on Human-Computer Interaction*. 65–68.

[28] Kelly E Matthews. 2017. Five propositions for genuine students as partners practice. *International Journal for Students as Partners* 1, 2 (2017).

[29] T. Moon. 1996. The expectation-maximization algorithm. *IEEE Signal Processing Magazine* 13 (1996), 47–60.

[30] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine learning* 39, 2-3 (2000), 103–134.

[31] Dwayne E Paré and Steve Joordens. 2008. Peering into large lectures: examining peer and expert mark agreement using peerScholar, an online peer assessment tool. *Journal of Computer Assisted Learning* 24, 6 (2008), 526–540.

[32] Helen Purchase and John Hamer. 2018. Peer-review in practice: eight years of Aropä. *Assessment & Evaluation in Higher Education* 43, 7 (2018), 1146–1165.

[33] Charles M Reigeluth, Brian J Beatty, and Rodney D Myers. 2016. *Instructional-design theories and models, Volume IV: The learner-centered paradigm of education*. Routledge.

[34] Victor Shnayder and David C Parkes. 2016. Practical peer prediction for peer assessment. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

[35] Sean Tackett, Mark Raymond, Rishi Desai, Steven A Haist, Amy Morales, Shiv Gaglani, and Stephen G Clyman. 2018. Crowdsourcing for assessment items to support adaptive learning. *Medical teacher* 40, 8 (2018), 838–841.

[36] Joanna Tai, Rola Ajjawi, David Boud, Phillip Dawson, and Ernesto Panadero. 2018. Developing evaluative judgement: enabling students to make decisions about the quality of work. *Higher Education* 76, 3 (2018), 467–481.

[37] Raquel Urena, Gang Kou, Yucheng Dong, Francisco Chiclana, and Enrique Herrera-Viedma. 2019. A review on trust propagation and opinion dynamics in social networks and group decision making frameworks. *Information Sciences* 478 (2019), 461–475.

[38] Jason L Walsh, Benjamin HL Harris, Paul Denny, and Phil Smith. 2018. Formative student-authored question bank: perceptions, question quality and association with summative performance. *Postgraduate medical journal* 94, 1108 (2018), 97–103.

[39] Guan Wang, Sihong Xie, Bing Liu, and S Yu Philip. 2011. Review graph based online store review spammer detection. In *2011 IEEE 11th international conference on data mining*. IEEE, 1242–1247.

[40] Wanyuan Wang, Bo An, and Yichuan Jiang. 2018. Optimal spot-checking for improving evaluation accuracy of peer grading systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[41] Xu Wang, Srinivasa Teja Talluri, Carolyn Rose, and Kenneth Koedinger. 2019. UpGrade: Sourcing Student Open-Ended Solutions to Create Scalable Learning Opportunities. In *Proceedings of the Sixth ACM Conference on Learning@ Scale*. 1–10.

[42] Sarah Weir, Juho Kim, Krzysztof Z Gajos, and Robert C Miller. 2015. Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. 405–416.

[43] Jacob Whitehill, Cecilia Aguerrebere, and Benjamin Hylak. 2019. Do Learners Know What's Good for Them? Crowdsourcing Subjective Ratings of OERs to Predict Learning Gains. *International Educational Data Mining Society* (2019).

[44] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. 2016. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third ACM Conference on Learning@ Scale*. 379–388.

[45] David Kofoed Wind, Rasmus Malthe Jørgensen, and Simon Lind Hansen. 2018. Peer Feedback with Peergrade. In *ICEL 2018 13th International Conference on e-Learning*. Academic Conferences and publishing limited, 184.

[46] James R Wright, Chris Thornton, and Kevin Leyton-Brown. 2015. Mechanical TA: Partially automated high-stakes peer grading. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 96–101.

[47] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.

[48] Dengyong Zhou, Qiang Liu, John Platt, and Christopher Meek. 2014. Aggregating ordinal labels from crowds by minimax conditional entropy. In *International conference on machine learning*. 262–270.

[49] Cai-Nicolas Ziegler and Georg Lausen. 2005. Propagation models for trust and distrust in social networks. *Information Systems Frontiers* 7, 4-5 (2005), 337–358.