# Exploring Models & Techniuqes For AI-Driven Assessment

**Ayse Kok Arslan**
Oxford Alumni, Northern California, USA

***Abstract*— Artificial intelligence (AI) has an increasingly important role for personalized training systems. This paper explores the issues with the standard assessment paradigm and the challenges associated with AI and assessment. It highlights the need for development of actionable and personalized explanations by incorporating human-centric design into the development process of learning tools towards an ultimate advancement of trustworthy training systems. The suggested platform architecture makes use of an open-source Python API to specify the workflow of an experiment (the "API"), and a Platform-as-a-Service (PaaS), which is a running instance of back-end infrastructure integrated with computational resources and a large training dataset. To evaluate the model, predictive modeling experiments is suggested by following a standard end-to-end supervised learning workflow. Last, but not least, by exploring different techniques, this study aims to synthesize an agenda for future research on AI-driven assessment techniques.**

***Keywords—AI, big data, assessment, ML***

## I. INTRODUCTION

Problem-based learning (PBL) is an instructional approach that exemplifies authentic learning and emphasizes problem-solving within richly contextualized settings. In PBL, users assume primary responsibility for their own development while trainers provide facilitation. Use of additional tools or technologies such as a dashboard can serve as a separate medium allowing trainers to view, just-in-time, how their users are doing overall.

SA (sequence analysis), specifically sequence clustering and comparison, is a promising avenue to the study of individual problem-solving, informing more effective personalized learning. The problem, however, is that many SA techniques cannot be interpreted, so that it can be difficult for a stakeholder to understand the data or how to act on the results.

This study presents a human-in-the-loop approach through visualization of sequences to analyze the sequences and develop a user-level understanding of the data. After a brief review of existing studies, the study explores how SA can be beneficial to learning environments in general, especially in the context of online and hybrid learning, where it allows stakeholders to understand individual learning in detail. By doing so, it aims to synthesize an agenda for future research on AI-driven assessment techniques.

## II. REVIEW OF EXISTING WORK

Problem-based learning (PBL) is "an instructional (and curricular) user-centric approach that empowers learners to conduct research, integrate theory and practice, and apply knowledge and skills to develop a viable solution to a defined problem" (Savery, 2006, p. 12).

PBL requires a detailed understanding of the learner's problem-solving processes, often obtained through the granular analysis of their actions within a training environment (Shute et al., 2010; Min et al., 2016; Kinnebrew &Biswas, 2012; Baker et al., 2006).

Despite its effectiveness, complex user-centric learning environments such as PBL require appropriate scaffolds that support novices' learning and problem- solving processes (Pellegrino, 2004; Simons & Klein, 2007). The drawback of these approaches, however, is that they are overly focused on the outcomes of a user's problem solving, rather than the process.

SA (sequence analysis), specifically sequence clustering and comparison, is a promising avenue to for individual problem-solving.

This approach demonstrates value as an effective method for identifying individual differences within a group of learners and patterns across a community (Kinnebrew &Biswas, 2012; Kinnebrew et al., 2013); yet without an understanding of the data, it can be difficult for a stakeholder to infer how an algorithm or statistical method is understanding the data or why a statistical technique resulted in what it did.

A human-in-the-loop method is defined as one where a human stakeholder can interpret the output of the model and then provide input to the model that impacts how it analyzes, compares, or clusters the sequences. A human-in-the-loop approach to sequence analysis (SA) can produce more interpretable results by allowing stakeholders to understand and correct the model and its outputs.

A great proportion of problem-solving tasks require logical reasoning. To that extent, SA also encompass hidden Markov models(HMMs), which identify meaningful interaction patterns and infer user

problem-solving strategies or predict future actions(Jeong et al., 2008; Balakrishnan & Coetzee, 2013; Boumi & Vela, 2019; Geigle & Zhai, 2017; Doleck et al., 2015).

To give a specific example, transformers used in deep learning architectures find clever ways to learn statistical features that inherently exist in the reasoning problems rather than learning to emulate reasoning functions. These researchers tested a popular transformer architecture, on a confined problem space which could accurately respond to reasoning problems on in-distribution examples in the training space yet couldn't generalize to examples drawn from other distributions based on the same problem space.

For most NLP tasks, one of the major goal for a neural model is to learn statistical patterns, yet, for logical reasoning, even though numerous statistical features inherently exist, models should not be utilizing them to make predictions. The rules of logic never rely on statistical patterns to conduct reasoning. Given the challenge of developing a logical reasoning dataset without any statistical features, learning to reason from data is difficult.

Reinforcement Learning (RL) is a family of ML techniques that may be applied to find increasingly optimal solutions through an automated iterative exploration and training process. Heuristics are algorithms that, empirically, produce reasonably optimal results for hard problems, within pragmatic constraints (e.g. "reasonably fast"). In a real-world setting, two main benefits are expected from ML techniques:

- Heuristics are human-trained based on a human-manageable set of benchmarks and regression cases. ML can easily scale to large corpora of training examples.

- Second, heuristics are human-written code that needs to be maintained. This places a downward pressure on the number of program properties ("features") and the combinations between them that can be practically leveraged. Using more features and feature combinations could result in better optimization decisions.

One reason why RL serves a suitable tool for replacing optimization heuristics is that one can efficiently explore different strategies, and improve strategies from those experiences. The absence of examples ("labels") means one cannot use supervised learning. In contrast, RL is an area of machine learning that learns from trial and error instead of given labels. In RL, an agent (i.e., the compiler) learns by repeatedly interacting with the environment (i.e., compiling) and gradually improves its policy (i.e., decision rules).

Within the context of RL, the Transformer architecture (Vaswani et al., 2017) has enabled large-scale language models (LMs) trained on a huge amount of data (Radford et al., 2019; Dai et al., 2019b; Radford et al., 2018b) to greatly improve the state-of-the-art on natural language processing tasks. Current methods for controlled text generation involve either fine-tuning existing models with RL (Ziegler et al., 2019), training Generative Adversarial Networks (Yu et al., 2017), or training conditional generative models (Kikuchi et al., 2016; Ficler & Goldberg, 2017). These models are used to extract contextualized word embeddings for transfer learning purposes (Devlin et al., 2019) and as natural language generators.

Yu et al. (2016), and more recently Yu et al. (2019); Yee et al. (2019); Ng et al. (2019), leveraged the Shannon Noisy Channel Theory (Shannon, 1948) for improving sequence-to-sequence modeling. Holtzman et al. (2018); Ghazvininejad et al. (2017) consider controlled language generation – the former with discriminators, and the latter with a bag of words – where the decoding procedure is modified to consider the scoring function used for decoding.

The progress in language model (LM) pretraining (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019; Yang et al., 2019; Liu et al., 2019a; Brown et al., 2020; Liu et al., 2020a; Lewis et al., 2020; Raffel et al., 2020; Gao et al., 2020a) has led to the possibility of conducting few-shot learning, that is, learning a new task using a small number of examples without any further training or gradient computation.

Learning to calibrate the few-shot results is essential to reduce the model's performance variance (Zhao et al., 2021), and the selection criteria in choosing the prompts are also important (Perez et al., 2021). Shin et al. (2020); Li and Liang (2021) proposed an automated method to create prompts for a diverse set of tasks by gradient-based tuning instead of manually searching for a good prompt. Using such a method, may allow one to find an optimal prompt easier given the difficulty to discover the optimal prompts for complicated natural language processing tasks, such as semantic parsing (Liu et al., 2021b).

One widely used AI technique for generating RL models is *latent knowledge estimation* (Corbett & Anderson, 1994). The reason this is referred to as *latent* lies in the fact that knowledge cannot be directly observed. What can be observed is whether a user can apply a knowledge component in some context.

Another one is Bayesian knowledge tracing (BKT) which is the best-known technique for latent knowledge estimation (Corbett & Anderson, 1994). The technique uses four parameters to estimate whether a user can apply a knowledge component, including;

(a) probability that the user already masters a knowledge component,
(b) probability of learning a knowledge component after a learning opportunity,
(c) probability of correctly applying a knowledge component even when the user has not mastered it (guess), and
(d) probability of incorrectly applying a knowledge component although they know it (slip).

The architecture of deep learning models has been shown to have a crucial effect on both the training speed and generalization (dAscoli et al., 2019; Neyshabur, 2020). Existing research suggests that having AI systems explain their inner workings to their end users can help foster transparency, interpretability, and trust. A range of analyses provide conceptual frameworks for understanding the challenges of these AI models.

In some models, multiple large pre-trained models may be composed through language (via prompting) without requiring training, to perform new downstream multimodal tasks. This offers an alternative method for composing pre-trained models that directly uses language as the intermediate representation by which the modules exchange information with each other.

As human-beings don't write in the same way that they speak due to the controlled and deliberate nature of written language, transcripts of spontaneous speech (like interviews) are hard to read. Researchers came up recently with some ML techniques on how to "clean up" transcripts of spoken text and to create more readable transcripts and captions of human speech by finding and removing disfluencies in people's speech. Using labeled data, ML algorithms can identify disfluencies in human speech and remove the extra words to make transcripts more readable.

To investigate whether the disfluency detection model is effective in streaming applications, the utterances in the training set is split into prefix segments, where only the first *N* tokens of the utterance were provided at training time, for all values of *N* up to the full length of the utterance. In essence, the model is being asked to "wait" for one or two more tokens of evidence before making a decision.

Language models have also demonstrated remarkable performance on a variety of natural language tasks such as quantitative reasoning. Solving mathematical and scientific questions requires a combination of skills, including correctly parsing a question with natural language and mathematical notation, recalling relevant formulas and constants, and generating step-by-step solutions involving numerical calculations and symbolic manipulation.

Whether it is qualitative or quantitative, when it comes to utilizing these AI models, one of the critical areas supported by human-centered AI is the process of assessment design used to elicit evidence to support claims about learning. While automated question generation can be a powerful tool for making assessment design more feasible for educators, it is not without its limitations. Large-scale datasets are needed to train the models that generate the questions.

Peer assessment has been recognized as a sustainable and developmental assessment method. Peer assessment can formally be defined as "an arrangement for learners to consider and specify the level, value, or quality of a product or performance of other equal-status learners" (Topping, 2009, p. 20). A simple approach would be to use summary statistics such as mean or median. However, summary statistics suffer from the assumption that all users have a similar judgmental ability, which has proven incorrect (Abdi et al., 2021).

By thoughtfully defining parameters, identifying its impact on users and assessing current capabilities, AI tools fitting assessment needs can be chosen. Figure 1 provides a graphical summary of peer assessment processes along with problems and proposed approaches and results.

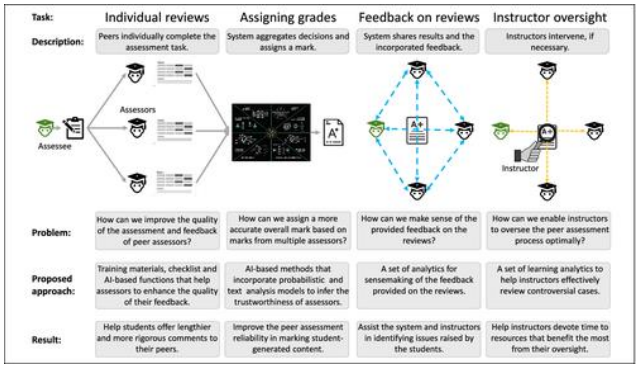



**Figure 1.** A graphical summary of peer assessment processes

The data generated from users' engagement with the peer assessment process may be utilized by learning analytics tools and learning analytics dashboards (Matcha et al., 2019) to enable instructors to gain insights into users learning process.

The next sections explore the model development in more detail.

## III. MODEL DEVELOPMENT

The platform architecture consists of two main components: an open-source Python API for specifying the workflow of an experiment (the "API"), and a Platform-as-a-Service (PaaS), which is a running instance of back-end infrastructure coupled

with computational resources and a large training dataset.

*Controller Scripts:* First, a user creates and submits a configuration file, either using an HTTP request or using the easy_submit() API function. This configuration file contains job metadata, including a pointer to an executable Docker image which encapsulates all code, software, and operating system dependencies for the users' experiment. The configuration file also points to a Python controller script that specifies the high-level experimental workflow, such as how model training and testing should occur and whether cross-validation or a holdout set should be used in a predictive modeling experiment. The use of controller scripts is a best practice for reproducible computational research [18], as it provides a single script to fully reproduce an experiment.

An additional advantage is that controller scripts are human-readable, providing a high-level overview of an experiment. One can use the controller script to manage low-level data platform tasks, including (i) data wrangling (retrieving and archiving necessary data at each step of the experiment); (ii) Docker image setup and execution; and (iii) parallelization.

The controller script provides sufficient information about how one can execute parallelization, which can lead to speed-ups of 1-2 orders of magnitude when CPUs are occupied with a separate task (e.g. training models on each of the different courses available).

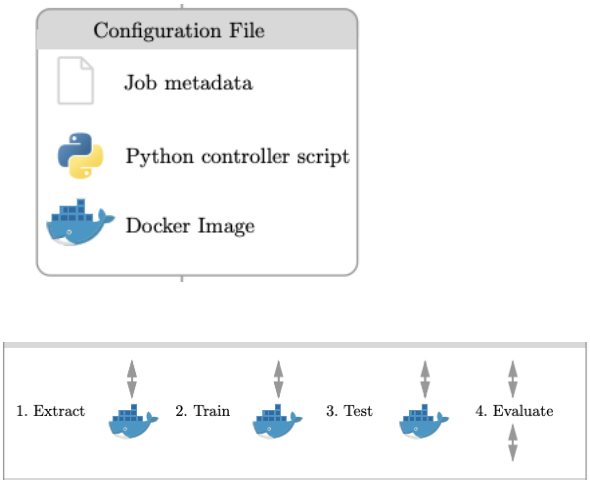



Fig 3. Overview of Docker container

*Docker containers:* These are frequently used in both industrial software applications as well as computational and computer systems research [20], [30], [31]. Their use in data science applications is increasing, but the execution, publication, and sharing of pre-built Docker images as part of a research workflow is rare. Containerization is hidden from the user which might limit users' ability to develop complex, customized environments.

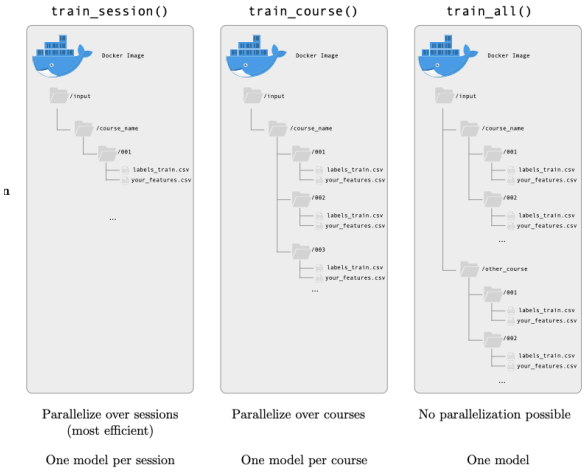
As seen in Figure 3., when submitting a job for execution to the platform, a user generates a Docker image containing the code, and operating system dependencies required to execute their experiment, and uploads the image to a public location (files located locally, HTTP, or in Amazon S3 are supported). The user provides the image's URL the configuration file submitted to the platform, and the image is fetched, checked, and executed according to the controller script. When an experiment completes error-free execution, the platform uploads the image to a public image repository on Docker Hub using a unique identifier. This makes implementations of every experiment immediately and publicly available for verification, extension, citation, or re-use.

A major advantage of Docker over simple code-sharing is that Docker containers fully reproduce the entire execution environment of the experiment, including code, software dependencies, and operating system libraries, exactly as this environment is configured at the time of an experiment. These containers are much more lightweight than a full virtual machine, but achieve the same level of reproducibility [29], [31].

As seen in Figure 4, the Python API allows users to provide a simple execution "recipe" for the platform to execute their experiment specifying the complete end-to-end pipeline from raw data to model evaluation: extract features from raw data; train and test machine learning models (predictive modeling experiments only), and evaluate the experimental results. For example, after extracting the desired features, a predictive modeling experiment could train individual models for every session of a course by using train_session() in their controller script; train one model per course using the data from all sessions by using train_course(); or train a single monolithic model using all data from every session of every course by using train_all().

```
extract_session()
extract_holdout_session()
train_course(label_type = 'dropout')
test_course(label_type = 'dropout')
evaluate_course(label_type = 'dropout')
```

Figure 4. Sample API script for Docker




## RECOMMENDATIONS

A critical aspect is the construction and analysis of predictive models of individual success [8]. To evaluate the model, predictive modeling experiments can be done by following a standard end-to-end supervised learning workflow:

- feature extraction from raw data;
- model training; model testing; and
- model evaluation (whereby performance is analyzed or, optionally, evaluated using statistical tests).

In addition to jointly addressing several challenges to reproducible and replication research within the field of learning sciences, the architecture, workflow, and initial research results have implications for the broader big data community. This includes;

- experimental reproducibility as big data research in many fields uses increasingly complex computational models;
- methodological and inferential reproducibility as big data research enables problematic statistical practices such as massively multiple testing via testing thousands or millions of hypotheses in a single experiment; and
- data reproducibility as available data become massively multimodal (many different formats), measure increasingly private or restricted aspects of users' behavior and identity, and cannot be easily anonymized.

Such a framework would be domain-agnostic, and can support generic workflows for supervised learning and production rule analyses in any domain which works with complex, multiformat data which cannot be easily anonymized (e.g. sensitive medical data, copyrighted media, computational nuclear physics).

## CONCLUSION

This paper brings both the issues with the standard assessment paradigm and the challenges associated with AI and assessment into a deeper conversation that will ultimately improve assessment practices more generally.

The paper has two important implications for learning analytics and AI in education**:**

- *First, this paper gives researchers and practitioners a novel systematic approach to incorporating advances in AI-driven assessment by having a strong grounding in a theoretical model of relevant learning processes. Specifically, the paper demonstrated how a theoretical model can be used to structure the program of research, development, deployment, and evaluation by addressing a problem that may emerge in practice.*

- *Second, the studies reported in the paper provide fresh empirical insights that can inform the development of future AI-driven assessment that seek to enhance trustworthiness of peer-review.*

The comprehensive discussion of learner-sourced adaptive systems, open-ended learning environments, writing analytics tools, team-based learning to support knowledge transfer allows for a detailed understanding of current state-of-art and open challenges. By doing so, this study will hopefully help to synthesize an agenda for future research for AI-driven assessment techniques.

## REFERENCES

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[2] Abdi, S., Khosravi, H., & Sadiq, S. (2020). Modelling learners in crowdsourcing educational systems. In International Conference on Artificial Intelligence in Education (pp. 3–9). Springer.

[3] Abdi, S., Khosravi, H., Sadiq, S., & Darvishi, A. (2021). Open learner models for multi-activity educational systems. Artificial Intelligence in Education, 11–17. https://doi.org/10.1007/978-3-030-78270-2_2

[4] Ahmad, N., & Bull, S. (2008). Do users trust their open learner models? In International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 255–258). Springer.

[5] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S.,

Gil-López, S., Molina, D., Benjamins, R., & Chatila, R. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115.

[6] Ashenafi, M. M. (2017). Peer-assessment in higher education-twenty-first century practices, challenges and the way forward. Assessment & Evaluation in Higher Education, 42, 226–251.

[7] Carless, D., & Boud, D. (2018). The development of user feedback literacy: Enabling uptake of feedback. Assessment & Evaluation in Higher Education, 43, 1315–1325.

[8] Cho, K., & MacArthur, C. (2011). Learning by reviewing. Journal of Educational Psychology, 103, 73–84.

[9] Darvishi, A., Khosravi, H., & Sadiq, S. (2020). Utilising learner sourcing to inform design loop adaptivity. In European Conference on Technology Enhanced Learning (pp. 332–346). Springer.

[10] Darvishi, A., Khosravi, H., & Sadiq, S. (2021). Employing peer review to evaluate the quality of user generated content at scale: A trust propagation approach. In Proceedings of the Eighth ACM Conference on Learning@ Scale (pp. 139–150). Association for Computing Machinery

[11] Gardner, J., Brooks, C., & Baker, R. (2019). Evaluating the fairness of predictive user models through slicing analysis. In Proceedings of the 9th international conference on learning analytics & knowledge (pp. 225–234). Association for Computing Machinery.

[12] Gašević, D., Kovanović, V., & Joksimović, S. (2017). Piecing the learning analytics puzzle: A consolidated model of a field of research and practice. Learning: Research and Practice, 3, 63–78.

[13] Gyamfi, G., Hanna, B. E., & Khosravi, H. (2021). The effects of rubrics on evaluative judgement: A randomised controlled experiment. Assessment & Evaluation in Higher Education, 47(1), 126–143. https://doi. org/10.1080/02602938.2021.1887081

[14] Hadwin, A., Järvelä, S., & Miller, M. (2017). Self-regulation, co-regulation, and shared regulation in collaborative learning environments. In Handbook of self-regulation of learning and performance (pp. 83–106). Routledge.

[15] Han, Y., Wu, W., Yan, Y., & Zhang, L. (2020). Human-machine hybrid peer grading in SPOCs. IEEE Access, 8, 220922–220934.

[16] Hassan, T. (2019). Trust and trustworthiness in social recommender systems. In Companion Proceedings of The 2019 World Wide Web Conference (pp. 529–532). Association for Computing Machinery.

[17] Henderson, M., Phillips, M., Ryan, T., Boud, D., Dawson, P., Molloy, E., & Mahoney, P. (2019). Conditions that enable effective feedback. Higher Education Research & Development, 38, 1401–1416.

[18] Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and anovas. Frontiers in Psychology, 4, 863.

[19] Lee, W., Huang, C. H., Chang, C. W., Wu, M. K. D., Chuang, K. T., Yang, P. A. and Hsieh, C. C. (2018) Effective quality assurance for data labels through crowdsourcing and domain expert collaboration. In 21st International Conference on Extending Database Technology, EDBT 2018 (pp. 646–649). OpenProceedings.org.

[20] Levy, H., & Robinson, M. (2006). Stochastic dominance: Investment decision making under uncertainty (Vol. 34). Springer

[21] Matcha, W., Gašević, D., & Pardo, A. (2019). A systematic review of empirical studies on learning analytics dashboards: A self-regulated learning perspective. IEEE Transactions on Learning Technologies, 13(2), 226–245.

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

[23] Napoles, C., Sakaguchi, K., Post, M., & Tetreault, J. (2015). Ground truth for grammatical error correction metrics. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (pp. 588–593). Association for Computational Linguistics (ACL).

[24] Negi, S., Asooja, K., Mehrotra, S., & Buitelaar, P. (2016). A study of suggestions in opinionated texts and their automatic detection. In Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (pp. 170–178). Association for Computational Linguistics

[25] Purchase, H., & Hamer, J. (2018). Peer-review in practice: Eight years of Aropä. Assessment & Evaluation in Higher Education, 43, 1146–1165.

[26] Ramachandran, L., Gehringer, E. F., & Yadav, R. K. (2017). Automated assessment of the quality of peer reviews using natural language processing techniques. International Journal of Artificial Intelligence in Education, 27, 534–581.

[27] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3982–3992). Association for Computational Linguistics

[28] Topping, K. J. (2010). Peers as a source of formative assessment. In Handbook of formative assessment (pp. 73–86). Routledge.

[29] Urena, R., Kou, G., Dong, Y., Chiclana, F., & Herrera-Viedma, E. (2019). A review on trust propagation and opinion dynamics in social networks and group decision making frameworks. Information Sciences, 478, 461–475.

[30] Wang, W., An, B. and Jiang, Y. (2018) Optimal spot-checking for improving evaluation accuracy of peer grading systems. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1). AAAI Press.

[31] Wang, W., An, B., & Jiang, Y. (2020). Optimal spot-checking for improving the evaluation quality of crowdsourcing: Application to peer grading systems. IEEE Transactions on Computational Social Systems, 7, 940–955.

[32] Wind, D. K., Jørgensen, R. M., & Hansen, S. L. (2018). Peer feedback with peergrade. In ICEL 2018 13th International Conference on e-Learning (p. 184). Academic Conferences and Publishing Limited.

[33] Wright, J. R., Thornton, C., & Leyton-Brown, K. (2015). Mechanical TA: Partially automated high-stakes peer grading. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 96–101). Association for Computing Machinery.

[34] Xiong, W., & Litman, D. (2011). Automatically predicting peer-review helpfulness. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (pp. 502–507). Association for Computational Linguistics.

[35] Yang, M., Tai, M., & Lim, C. P. (2016). The role of e-portfolios in supporting productive learning. British Journal of Educational Technology, 47, 1276–1286.

[36] Yang, T.-Y., Baker, R. S., Studer, C., Heffernan, N., & Lan, A. S. (2019). Active learning for user affect detection. In Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019. International Educational Data Mining Society (IEDMS) 2019 (pp. 208–217). Université du Québec; Polytechnique Montréal.

[37] Yeager, D. S., Purdie-Vaughns, V., Garcia, J., Apfel, N., Brzustoski, P., Master, A., Hessert, W. T., Williams, M. E., & Cohen, G. L. (2014). Breaking the cycle of mistrust: Wise interventions to provide critical feedback across the racial divide. Journal of Experimental Psychology: General, 143, 804–824.

[38] Yu, F.-Y., & Wu, C.-P. (2011). Different identity revelation modes in an online peer-assessment learning environment: Effects on perceptions toward assessors, classroom climate and learning activities. Computers & Education, 57, 2167–2177.

[39] Zheng, L., & Huang, R. (2016). The effects of sentiments and co-regulation on group performance in computer supported collaborative learning. The Internet and Higher Education, 28, 59–67.

[40] Zhu, Q., & Carless, D. (2018). Dialogue within peer feedback processes: Clarification and negotiation of meaning. Higher Education Research & Development, 37, 883–897.

[41] Zimmerman, B. J., Bonner, S., & Kovach, R. (1996). Developing self-regulated learners: Beyond achievement to self-efficacy. American Psychological Association.

[42] Zong, Z., Schunn, C. D., & Wang, Y. (2021). What aspects of online peer feedback robustly predict growth in users' task performance? Computers in Human Behavior, 124, 106924.