

ABSTRACT

RASHID, M PARVEZ. Towards Quality Assessment in the Peer-review Process Using AI and Learning Analytics. (Under the direction of Dr. Edward F. Gehringer).

Peer assessment has been proven to be highly effective for both learning and assessment in on-campus and distance-learning environments. Its utility is particularly evident in massive open online courses (MOOCs), where the potentially overwhelming number of students can challenge the assessment capacity of limited instructional staff. In many MOOCs, instructor involvement in assessment is minimal or absent, making peer assessment a crucial alternative. This process benefits students by providing timely feedback and fostering meta-cognitive thinking. Previous studies have shown that peer assessors can achieve accuracy comparable to that of instructors. However, despite its advantages, peer assessment faces scrutiny regarding the quality and accuracy of the review and the grades provided. Factors such as the helpfulness and congruence of review comments, as well as the accuracy of ratings, are critical to its success. It is important to acknowledge that not every peer assessor is a competent reviewer. The success of peer assessment relies on the quality of feedback provided by the assessors. Low-quality feedback can create confusion for the students, leading them to question the credibility of the review process and the competence of their reviewers.

This dissertation aims to enhance the quality of formative feedback and the accuracy of summative feedback by leveraging artificial intelligence (AI) and learning analytics. Specifically, it focuses on improving the three main tasks of the peer assessment process: (1) developing assessment guidelines, (2) obtaining feedback from peers, and (3) assigning final grades.

In Study 1, we identify the key components of peer review comments that contribute to more helpful feedback for students. In Study 2, we introduce a natural language processing (NLP) approach to analyze peer assessors' rubrics, assisting reviewers in providing high-quality formative feedback. Disagreements or contradictory reviews from peer assessors can confuse the reviewers and lead them to question the assessment quality. Study 3 proposes an AI-driven method to identify and quantify disagreements in formative feedback. Finally, in Study 4, we infer the effort and reliability of peer assessors by employing Bayesian inference and text analysis to calculate fair and accurate student grades.

© Copyright 2024 by M Parvez Rashid

All Rights Reserved

Towards Quality Assessment in the Peer-review Process Using AI and Learning Analytics

by
M Parvez Rashid

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina
2024

APPROVED BY:

Dr. Tiffany Barnes

Dr. Noboru Matsuda

Dr. Meagan Kittle Autry

Dr. Edward F. Gehringer
Chair of Advisory Committee

DEDICATION

To my parents, for their unconditional love and sacrifices.

BIOGRAPHY

M. Parvez Rashid is a passionate researcher and educator committed to transforming learning into a more accessible, efficient, and reliable experience. He launched his teaching career in 2009 at his alma mater following the completion of his BSc in Computer Science. To pursue higher education and gain research experience, he moved to the USA in 2017 and completed his MS in Computer Science at the University of Nebraska-Lincoln. In 2019, he began his PhD in Computer Science at North Carolina State University. His PhD research focuses on improving and analyzing the efficiency and accuracy of peer assessments in open-ended responses using AI and Natural Language Processing (NLP), combined with qualitative and quantitative analysis.

Parvez's research interests are broad and impactful, encompassing the use of AI and data science for advanced learning and addressing pressing social issues while maintaining a strong commitment to ethical considerations, including transparency, fairness, and social responsibility.

During his PhD tenure, Parvez published and presented numerous research papers at esteemed conferences, including Educational Data Mining, Learning Analytics and Knowledge, and the International Conference on IT in Higher Education and Training. His research was recognized with a Best Paper award and a nomination. In recognition of his excellence as a PhD student, he received the Carla Savage Award. His exceptional support as a teaching assistant earned him the Outstanding TA Award. Additionally, Parvez served as a Graduate Student Ambassador, representing NC State University at Graduate Education Day at the NC General Assembly.

ACKNOWLEDGEMENTS

Reflecting on my PhD journey, my heart overflows with gratitude for the incredible individuals and institutions who have supported and encouraged me along the way.

I owe my deepest thanks to my parents, whose countless sacrifices—both seen and unseen—have made this journey possible. My mom, Pyaree Begum, has always been my rock and has always believed in me. My dad, Harun Ar Rashid, has been a constant source of support and encouragement. My brother's unwavering inspiration and my sister's boundless love have been sources of strength and comfort. My heart is full of gratitude for the endless support and love from all my family members.

To my advisor, Dr. Edward Gehringer, I am profoundly grateful for his steadfast support and guidance. This dissertation would not have been possible without his invaluable mentorship. I also want to express my heartfelt thanks to my PhD committee members, Dr. Tiffany Barnes, Dr. Noboru Matsuda, and Dr. Meagan Kittle Autry, whose expert advice and insightful feedback have significantly enriched this work. Special thanks to Dr. Gregg Rothermel for his dedication to supporting all the PhD students and staff in the Computer Science Department.

I am deeply thankful to my co-authors Divyang Doshi, Mitchell Young, Yunkai Xiao, Chengyuan Liu, Qinjin Jia, Samhita Pal, and Dr. Hassan Khosravi. Collaborating with you has been a tremendous honor and privilege.

My heartfelt thanks go to Mariana Sanchez for her constant kindness and support, and to all my friends, Dawn Batten and Dr. Rayhanur Rahman for their unwavering encouragement.

Lastly, I extend my profound gratitude to North Carolina State University for providing the resources and opportunities that have shaped me into a better person and a proud PhD graduate. NC State have cultivated a growth mindset, fostering a love for learning that will endure a lifetime.

TABLE OF CONTENTS

List of Tables	viii
List of Figures	x
Chapter 1 INTRODUCTION	1
1.1 Motivation	1
1.2 Goal	3
1.3 Contribution	5
1.3.1 Identifying helpful feedback from students’ perspective	5
1.3.2 Analyzing rubrics	5
1.3.3 Automated meta-reviewing for locating disagreement in peer assessors’ formative feedback	5
1.3.4 Reliability-based aggregated grading	6
Chapter 2 Related work	8
2.1 Helpful feedback from Students’ perspective	9
2.2 Peer assessment rubric analyzer	10
2.2.1 Analyzing rubrics	11
2.2.2 Analyzing review comments	11
2.3 Automated meta-review for locating (dis)agreement	12
2.3.1 Identifying (dis)agreement	12
2.3.2 Feature Extraction and clustering	13
2.4 Reliability-based aggregated grading	14
Chapter 3 Data	16
3.1 Data Source: Expertiza	16
3.1.1 Data Annotations	17
3.2 Data Privacy	18
Chapter 4 Analyzing Helpful Feedback from the Student’s Perspective	19
4.1 Introduction	19
4.2 Related Work	20
4.3 Data	21
4.4 Method	22
4.4.1 Classical Machine-Learning Models	23
4.4.2 Neural-Network Models	24
4.5 Results	25
4.6 Discussion and Conclusion	29
Chapter 5 Peer Assessment Rubric Analyzer: An NLP approach to analyzing rubric items for better peer-review	30
5.1 Introduction	30
5.2 Related Work	32

5.3	Data	33
5.3.1	Collecting Rubrics and Review Comments	33
5.3.2	Collecting Labeled Review Comments	34
5.3.3	Annotating Review Comments	35
5.3.4	Constructing the Dataset	35
5.4	Methods	36
5.4.1	Classical Machine-Learning Models	36
5.4.2	Neural-Network Models	38
5.5	Experiment	39
5.5.1	Selecting a Model for Review Text Annotation	39
5.5.2	Classifying the rubric items	40
5.5.3	Evaluation Metrics	40
5.6	Results and Discussion	42
5.7	Conclusion	47
Chapter 6	Automated meta-reviewing for locating disagreement in peer assessors' formative feedback	50
6.1	Introduction	50
6.2	Background	54
6.3	Approach: Automated Detection of Peer Feedback Discrepancies	55
6.3.1	Notation and Problem Statement	55
6.3.2	Proposed Approach	55
6.4	Methods	56
6.4.1	Data collection and preparation	56
6.4.2	Text Embedding	57
6.4.3	Active Learning	58
6.4.4	Clustering Algorithm	59
6.5	Experiment	60
6.5.1	Dataset Preparation	60
6.5.2	Algorithm to choose the cosine similarity cut-off	61
6.5.3	Selecting the text feature extraction model	63
6.5.4	Fine-tuning the model	63
6.5.5	Evaluation Metrics	64
6.6	Results and discussion	64
6.6.1	RQ1	64
6.6.2	RQ2	66
6.6.3	RQ3	66
6.7	Conclusion and Implications	70
Chapter 7	Reliability-based Weighted Ratings for better peer grading	74
7.1	Introduction	74
7.2	Background	75
7.3	Approach:	76
7.3.1	Notation and Problem Statement	77
7.3.2	Proposed Approach	77

7.3.3	Extracting review quality likelihood	78
7.3.4	Inference Model	78
7.4	Experimental Setting	81
7.4.1	Research Tool	81
7.4.2	Data collection and Context	82
7.4.3	Dataset Preparation	82
7.5	Results	84
7.6	Conclusion	88
Chapter 8	Conclusion And Future Work	89
References	92

LIST OF TABLES

Table 4.1	Sample review comment and annotations done by students ('1' indicates 'yes' and '0' indicates 'no')	21
Table 4.2	Hyperparameters of Models	25
Table 4.3	Table shows sample comments from helpfulness-detection dataset and corresponding annotations. Note that "Is Helpful" annotations are done by humans (students), while "Detects Problem" and "Gives Suggestion" are annotated by the BERT model. "Contains Problem and Suggestion" is from and ing the "Detects Problem" and "Gives Suggestion" columns.	26
Table 5.1	Percentage of samples for each class labels	35
Table 5.2	Sample review comment and labels for the classes	35
Table 5.3	Hyperparameters of ML Models	40
Table 5.4	Hyperparameters of Neural Network Models	41
Table 5.5	Performance comparison between BERT and Bi-LSTM at different likelihood cut-off points using PRAUC scores. The "Difference between number of samples in the 2 classes" is the difference between number of comments classed as 1 (effective rubric item) and 0 (less effective rubric item). Note that BERT outperforms Bi-LSTM at all the cut-offs points.	45
Table 5.6	Table shows predictions by the BERT model on 13 rubric items at different cut-off points. "Review count" is the number of reviews in our dataset that included the indicated criterion. A "1" in the cut-off point columns indicates the corresponding rubric item is predicted to induce peer-reviewers to write a quality review and "0" means it is not. Note that at higher cut-offs, the model classifies fewer rubric criteria as likely to induce effective comments.	46
Table 6.1	Table shows four peer-reviewers' comments on a piece of work following a rubric item. Three of the reviewers are in agreement, and one reviewer disagrees.	51
Table 6.2	Table shows a sample dataset with paired comments, labeled for agreement (Label "1") and disagreement (Label "0") in two comments. Each pair of comments relates to the same piece of work following the same rubric item	61
Table 6.3	This table shows the angle between the two-dimensional feature vectors of the review comments pair in the feature vector space. The angle between the vectors for comments-pair with disagreement is higher for fine-tuned SBERT than baseline SBERT	68
Table 6.4	Table shows the angle between the two-dimensional feature vectors of the review comments pair in the feature vector space. The angle between the vectors for comments-pair with an agreement is lower for fine-tuned SBERT than baseline SBERT	70

Table 6.5	Table shows the clustering outcomes on review comments using feature extraction by baseline SBERT and Fine-tuned SBERT. In the "Comments" column, each row presents comments of different reviewers on an artifact in response to an assessment criterion. Disagreement measure σ_2 (used Fine-tuned SBERT embedding) is more accurate than σ_1 (used baseline SBERT embedding)	71
Table 7.1	Sample review comment and annotations done by students ('1' indicates 'yes' and '0' indicates 'no')	84
Table 7.2	Correlation between instructor and peer assessors ratings. The ratings from the peer assessors are derived using mean, median, and weighted average	87

LIST OF FIGURES

Figure 1.1	Summary of the three core phases in peer assessment and the research problems we address in this dissertation.	3
Figure 3.1	Flow diagram of peer review and feedback annotation process	18
Figure 4.1	Annotation process of the helpfulness-dataset for mentioned problem and suggestions in the comments using models. The models were trained for detecting problems or suggestions mentioned in the review text. The training datasets were annotated by human (students).	23
Figure 4.2	F1-score comparison to measure performance on classifying review text on problem detection and suggestion detection using classical ML and neural-network models. In overall F1-score comparison, the BERT model shows the best performance.	26
Figure 4.3	Venn diagram of helpful feedback annotated for mentioned suggestion and/or problem	27
Figure 4.4	Top 10 positive and negative coefficient values of words from problem-detection and suggestion-detection dataset	28
Figure 5.1	F1-score comparison to measure performance to classify review text on problem-detection, suggestion-detection and localization datasets using baseline ML models and neural network models. In overall F1-score comparison BERT model shows better performance.	42
Figure 5.2	Change of "quality comments" as the likelihood cut-off point increases. A label of 1 indicates rubric items that are deemed to induce a quality review comments. A label of 0 indicates a less effective rubric item. With a cut-off of 0.1, about 60% of the comments are considered "quality comments," whereas at a cut-off of 0.6, only about 30% are.	44
Figure 5.3	Graphical view of data from Table 5.5. The right horizontal axis "Difference between Binary class Samples" and the bars indicate the difference between number of comments classed as 1 (effective rubric item) and 0 (less effective rubric item). The two lines show the PRAUC score at different cutoff points for the BiLSTM and BERT models. Note that BERT outperforms BiLSTM at all the cut-off points	45
Figure 5.4	Correlation between average length of review comments (<i>y</i> -axis) and the number of rubric items in rubrics (<i>x</i> -axis).	48
Figure 5.5	Correlation between average length of review comments (<i>y</i> -axis) and position of rubric items in the rubrics (<i>x</i> -axis).	48
Figure 6.1	Proposed approach to quantify and locate disagreements in peer assessors' review comments.	53
Figure 6.2	BERT Cross-encoder Architecture for Sentence Similarity	58
Figure 6.3	SBERT Training and Inference Phase architecture	59
Figure 6.4	The most uncertain samples lie near the decision line	59

Figure 6.5	Fine-tuning the SBERT model using active learning with an expert in the loop.	64
Figure 6.6	Comparison of Sentence Embedding Approaches using Accuracy Score on the Test Dataset	65
Figure 6.7	Fine-tuning of SBERT increased accuracy for identifying sentence similarity or difference after each iteration	66
Figure 6.8	Overall angle comparison of feedback texts pairs in agreement or disagreement.	67
Figure 6.9	Agglomerative clustering algorithm's performance using feature vectors from Baseline-SBERT and Fine-tuned SBERT	69
Figure 7.1	Comparison of quality reviewers' ratings with instructor's ratings.	76
Figure 7.2	Inference model of Peer assessors' effort (π) in writing quality comments in Round 1 and ideal Score (β) in Round 2	78
Figure 7.3	Formative (Round 1)and summative (Round 2) assessment rounds in four phases.	83
Figure 7.4	F1-score comparison to measure performance on classifying review text on problem detection and suggestion detection using classical ML and neural-network models. In the overall F1-score comparison, the BERT model shows the best performance.	85
Figure 7.5	Comparison of quality reviewers' mean review scores with estimated effort.	86
Figure 7.6	Distribution of peer assessors' estimated effort (π)	86
Figure 7.7	Gibbs convergence for estimating reviewers effort in the assessment Round 1 for a few randomly picked peer assessors.	87

CHAPTER

1

INTRODUCTION

1.1 Motivation

Peer assessment in classroom environments involves students evaluating each other's work, offering an effective and scalable method of assessment. Peer assessment is beneficial to everyone involved in the assessment process, including reviewers, reviewees, and instructors (Kao (2013)). Studies have shown that students learn more from giving feedback than from receiving it (Rada et al. (1994); Çevik (2015); Li and Grion (2019); Graner (1987); Patchan and Schunn (2015); Glance et al. (2013)), are more engaged to study, learn actively, and think metacognitively Gehringer (2014); Veenman (2013); Prins (2002) during the assessment process. As each reviewer can put more time into providing feedback, reviewees benefit from more detailed, accurate, and timely feedback (Kulkarni et al. (2015)). Moreover, peer assessment alleviates instructors' workloads, allowing them to focus more on students who need additional support (Joyner (2017)). In both on-campus and online learning environments, peer assessment is proven to be a reliable and scalable process to assess students' work (Topping (2010); Yu and Wu (2011)). It is particularly useful in massive open online courses (MOOCs) where the number of students for a course has no bounds. For example, in 2018, 1,385 courses in Coursera used peer assessment, where over 4,000 students made submissions every day for assessment. All these submissions receive timely feedback from the peer assessors (Cambre et al. (2018)).

In peer assessment, reviewers generally provide two kinds of feedback: formative and summative (Evans (2013)). For formative feedback, peer assessors write a review comment analyzing the work. On the other hand, for summative feedback, reviewers provide ratings or point values. While summative feedback helps assign a grade for an artifact, formative feedback offers a greater understanding of the work and areas to improve Prins et al. (2005). However, the learning benefits of peer assessment heavily rely on the quality and accuracy of reviewers' feedback. Despite many advantages of peer assessment, this assessment system is often scrutinized for its quality and accuracy (Wang et al. (2018); Liu and Carless (2006); Darvishi et al. (2020)). Ensuring consistent and reliable evaluations remains challenging, necessitating ongoing improvement of peer assessment methodologies. However, Both formative and summative feedback present significant challenges regarding quality and precision (Wang et al. (2018); Darvishi et al. (2020)).

For students' learning gain, it is crucial that reviewees receive comprehensible and actionable formative feedback from the peer assessors (Evans (2013)). Previous studies have focused on identifying the components of quality formative feedback (Xiao et al. (2020); Zingle et al. (2019); Jia et al. (2021)). However, it is critical to identify the qualities of helpful feedback from students' perspectives. Studies also identified that feedback quality relies on assessment guidelines or rubrics (du Toit (2019); Reinholz (2016)). There is a dearth of literature on analyzing rubrics designed for peer assessors. In peer assessment, disagreements or contradicting feedback are common phenomena and undermine students' learning experience (Prins et al. (2005)) when multiple reviewers review a single work. There is no study to identify and quantify the contradictions in peer assessors' formative feedback.

Extracting accurate ratings from summative feedback in peer assessment remains a significant challenge when multiple peer assessors provide ratings on an artifact. Not all peer assessors possess the necessary competence for quality assessment (Carless and Boud (2018)). Traditional peer assessment systems often rely on aggregated mean and median ratings, but these summary statistics alone do not suffice to provide precise grades for artifacts (Piech et al. (2013); Darvishi et al. (2020); Zarkoob et al. (2023)). Several studies have attempted to address this issue by calculating the reliability of peer assessors (Darvishi et al. (2020); Piech et al. (2013); Zarkoob et al. (2023); Song et al. (2015); Hamer et al. (2005); Lauw et al. (2007)). However, these studies relied entirely on summative feedback and a single round of assessment to infer the reliability of peer assessors.

To ensure that peer assessment is accurate, efficient, and trustworthy, it is imperative to address the identified research gaps.

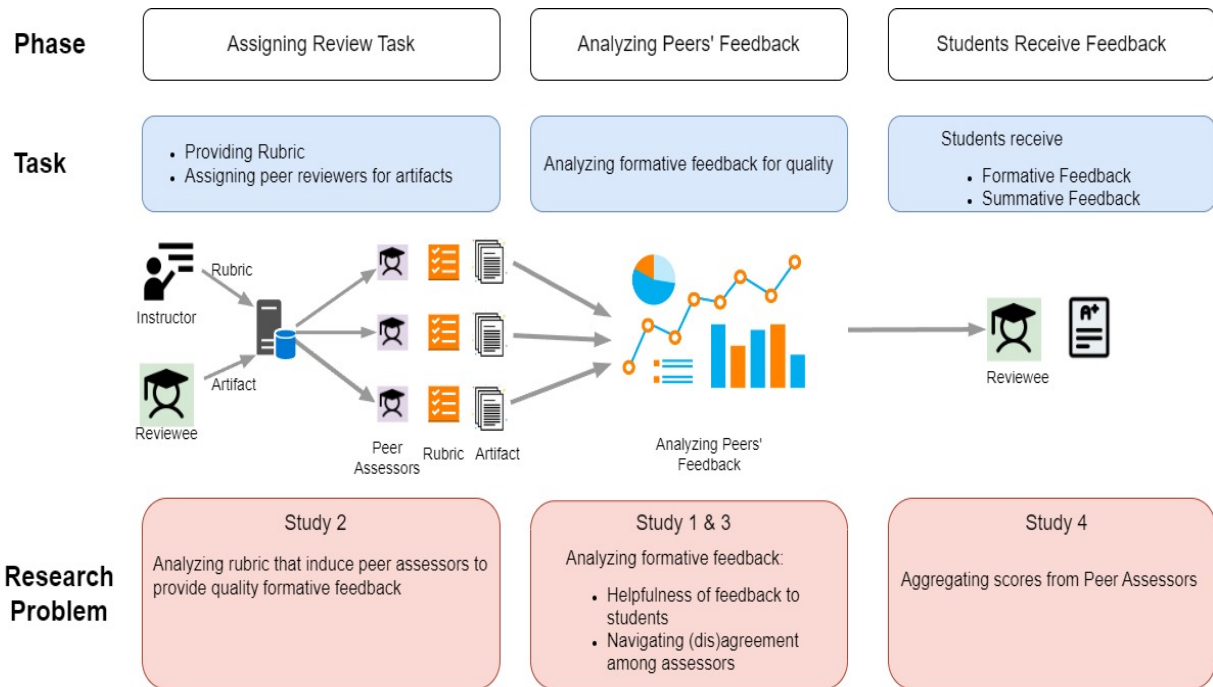


Figure 1.1: Summary of the three core phases in peer assessment and the research problems we address in this dissertation.

1.2 Goal

This dissertation uses AI and learning analytics to make peer assessment effective, accurate, and trustworthy for students. For our studies we collected data from real classrooms utilizing Expertiza (Gehringer (2010)) peer assessment system. In Figure 1.1, we have presented three core phases in the assessment process and the research problem associated with these phases. More specifically, we will address: (1) analyzing quality rubric for peer assessors, (2a) identifying the quality of helpful feedback, (2b) locating disagreement in formative feedback, and (3) reliability-based aggregated ratings for peers. To make the assessment more effective and accurate, we must address research problems associated with the three core phases of peer assessment (Figure: 1.1). We harnessed cutting-edge AI methods and learning analytics to address our research questions. Figure 1.1 shows a summary of the research problems. Later chapters discuss the problem and solutions in detail, with evident results. The research questions we addressed in this thesis are as follows:

1. Problem-1: Analyzing rubrics for quality formative feedback

- RQ1: Which model performs best for classifying quality review text?
- RQ2: Can neural network models detect rubric items that enable peer reviewers to

write quality reviews?

- RQ3: Is there any relation between rubric length or rubric-item position and length of the comments written by students?

2. Problem-2: Analyzing peers' feedback

(a) Sub-problem-1: Analyzing helpful feedback from students' perspective.

- RQ1: Can we build a model to accurately detect comments containing suggestions or detecting problems?
- RQ2: Are "quality comments"—those containing suggestions, detecting problems, or both—actually helpful from the student's point of view?
- RQ3: Can an automated process effectively identify helpful feedback?

(b) Sub-problem-2: Automated meta-reviewing for locating disagreement in peer assessors' formative feedback.

- RQ1: Which pre-trained feature extraction methods for context and semantically related sentences work best?
- RQ2: Can we fine-tune and improve the pre-trained SBERT model's sentence-feature extraction for our context-dependent review text using a semi-supervised approach?
- RQ3: Does improving the sentence feature extraction method improve clustering performance?

3. Problem-3: Reliability-based weighted ratings

- RQ1: Can we estimate peer assessors' effort in writing formative feedback?
- RQ2: Can we quantify the reliability of assessors from the estimated effort and ratings they provide?

Chapter 4 will address the research questions (RQ) of Sub-problem-1 under Problem-2 to identify the qualities of helpful feedback from students' perspectives. Chapter 5 will address RQs under Problem-1. we use NLP approach to analyze rubrics to find which rubric items induce peer assessors to provide quality comments to the reviewer. In Chapter 6, we describe an AI approach to identify disagreement among the peer assessors. The RQs addresses in this chapter are under Sub-problem- 2 in Problem-2. In Chapter 7, to addressed RQs under Problem-3 we implemented a reliability-based grading system using a probabilistic method that considers both formative and summative feedback to quantify the trustworthiness of a peer reviewer and provide more accurate grades for the reviewees.

1.3 Contribution

1.3.1 Identifying helpful feedback from students' perspective

Chapter 4 of the dissertation identifies the properties of peer review comments that make them helpful to students. A few previous studies identified helpful review comments that detected (i) problems in the assessment artifact and (ii) suggestions to resolve the issues (Xiao et al. (2020); Zingle et al. (2019); Nguyen et al. (2017)). This study analyzes helpful feedback in two steps. First, various advanced NLP methods are used to identify if any issues were mentioned and/or a solution was suggested in the review comments. Second, it compared whether mentioning the problem and/or providing solution suggestions was more helpful to the students. The results from the experiment suggest that students find review comments more helpful when an issue in the artifact is mentioned, along with a solution to the issue is suggested.

1.3.2 Analyzing rubrics

Formative feedback provides students with insights into the issues in their artifacts and how to improve the work. The quality of the peers' formative feedback in an assessment heavily relies on understanding the provided assessment criteria (Topping (2009)). A rubric to provide formative feedback has to be worded carefully. It should provide clear instructions to write quality formative feedback by the peer assessors (Reinholz (2016)). In Chapter 5, we discussed an NLP approach to analyze rubric text to identify whether rubric criteria will induce peer-reviewers to write quality reviews. We have analyzed 408,104 formative feedback comments based on 3,164 rubric criteria using natural language processing techniques with advanced neural network methods. To our knowledge, this is the first attempt to analyze rubric prompts using a supervised machine learning approach to improve review comments for the peer review environment.

1.3.3 Automated meta-reviewing for locating disagreement in peer assessors' formative feedback

Disagreement among the peer assessors is common while reviewing the same artifact. However, contradicting formative feedback can confuse the students, leading them to question the credibility of the review process and the competency of their reviewers Prins et al. (2005). Peer assessment has proven to be a valuable tool in the classroom, providing students with high-quality feedback. The effectiveness of formative feedback in peer assessment depends on clear communication with students, enabling them to take action on the feedback. Meta-

reviewing the peers’ feedback to identify disagreement is labor intensive for the instructor and delays feedback to the students. There is a dearth of research on identifying disagreements in peer assessment. In Chapter 6, we propose an automated meta-review approach to identify and quantify disagreements in formative feedback. Our approach will assist instructors to automatically identify cases that require their attention and reduce their burden of meta-review. We collected three distinct datasets with formative feedback from a software development course that utilized peer assessment. Recent natural language processing (NLP) models are trained on massive corpora of text to generate high-quality vector representations of texts. We compared existing NLP models and fine-tuned them to produce high-quality feature vectors from feedback full of technical jargon and inconsistent English. Our results demonstrate that quality feature representations of text improve the performance of the clustering algorithm in identifying and quantifying disagreements in formative feedback.

1.3.4 Reliability-based aggregated grading

Peer assessment is a long-established method to rate students’ work in educational environments. As each peer can devote more time to assess each artifact than the instructor can, the assessment can be of higher quality (Kulkarni et al. (2015)). However, not all the peer assessors are competent (Carless and Boud (2018)). Engaging the instructor to identify the peer assessors’ competency imposes more work on the instructor than simply rating the artifacts him/herself. To make the peer assessment more reliable, more than one reviewer is typically assigned to assess an artifact (Zarkoob et al. (2023)). Although deploying multiple reviewers increases the credibility of the assessment, it also introduces a new challenge of aggregating ratings from multiple peer assessors. As peer assessors rate each artifact independently, the rating scores may differ.

Many peer assessment systems rely on summary statistics to provide aggregated ratings to the reviewees. One issue with summary statistics, such as the mean or median, is that they incorporate scores from unreliable peer reviewers (Darvishi et al. (2020)). A better approach is to calculate an aggregated score by weighting peer assessors’ ratings based on reliability (Darvishi et al. (2020); Piech et al. (2013); Zarkoob et al. (2023); Song et al. (2015); Hamer et al. (2005); Lauw et al. (2007)). Using Equation 1.1 we can calculate the weighted score (S_j) of a peer assessor (j), where w_i indicates the reliability score of a peer assessor.

$$S_j = \frac{\sum_{i=1}^n w_i \times s_{ij}}{\sum_{i=1}^n w_i} \quad (1.1)$$

Competency-based score aggregation is a more reliable method to assess students (Piech

et al. (2013); Darvishi et al. (2020); Zarkoob et al. (2023)). However, this approach requires quantifying the reliability of peer assessors. A generic approach is to quantify competency by considering the grading history of peer assessors' grading quality. The current reputation systems consider only the summative ratings from the peer assessors to calculate a reputation score. It is important to include both summative and formative feedback quality while calculating the reputation score of a peer assessor.

In Chapter 7, we propose a probabilistic method based on the Bayes inference model to quantify the reliability of peer assessors that incorporates the textual feedback quality as well as the accuracy of ratings provided by the assessors.

CHAPTER

2

RELATED WORK

Peer assessment plays a significant role in both brick-and-mortar classrooms and MOOCs to conduct assessment. MOOCs generally have a large number of participants, as a result, the manual grading of assignments and exams is impractical for the relatively small teaching teams. To evaluate participants' performance, instructors depend on tools that facilitate automated grading (Gamage et al. (2021)). Large classrooms commonly use multiple-choice questions (MCQs) to make the assessment convenient (Staubitz et al. (2020)). However, multiple-choice quizzes are inadequate for evaluating higher-level learning that surpasses the lower tiers of Bloom's taxonomy (remember, understand, apply) (Churches (2008)). Open-ended assessment is beneficial for learning, but it puts more of a burden on the instructor to grade the students. With the recent enhancement of AI tools, open-ended questions can be automatically assessed. Despite a steady increase in research publications in this area, the results of automated assessments are often mixed, and their validity may be questionable (Huawei and Aryadoust (2023); Sánchez-Prieto et al. (2020)). Peer assessment is especially significant because it is the only method that allows course instructors to incorporate open-ended and creative assignments into their courses (Gamage et al. (2021)).

While peer assessment plays a crucial role in enhancing learning with open-ended and creative assignments, it frequently faces scrutiny regarding assessment quality and accuracy. This thesis tackles research challenges in the three core phases (Figure: 1.1) of peer assessment.

In this chapter, we will review the relevant literature that has inspired our work. In Section 2.1, we discuss background studies on the components of helpful feedback in peer assessment. In Section 2.2, we discuss the literature on peer-assessment rubrics and their qualities. In Section 2.3, we present literature on finding incoherent review comments. In Section 2.4 we discuss literature on enhancing the accuracy of summative feedback in peer assessment.

2.1 Helpful feedback from Students' perspective

In peer assessment, assessors provide feedback in two main forms: textual feedback, which is qualitative and typically used for formative purposes, and numerical scores on a Likert scale or point-based, used for calculating summative grades. Most online peer-assessment platforms utilize both types of feedback. However, numerous studies indicate that students benefit more from formative feedback as it provides insights into their work (Rada et al. (1994); Çevik (2015); Li and Grion (2019); Graner (1987)), and engages them in active learning and promotes metacognitive thinking (Gehring (2014)). On the other hand, du Toit (2019) showed that with poor reviews, students were confused about the quality of their work, sometimes feeling a false sense of accomplishment. Effective peer learning hinges on the quality of reviews offered by student peers. This section discusses related work that guided us in identifying the properties of quality formative feedback that students find helpful and methods to identify those qualities in peers' reviews.

According to Hattie and Timperley (2007), model feedback must answer three major questions. First, Where am I going? (What are the goals?). Second, How am I going? (What progress is being made toward the goal?) and third, Where to next? (What activities need to be undertaken to make better progress?). This implies that quality feedback in the education domain should include the specification of a problem that a student should work on as his/her goal and a suggestion to make progress on how to reach the goal.

Nelson and Schunn (2009) examined five features of feedback (summarization, specificity, explanations, scope, and effective language) that constitute good-quality reviews, and the correlations among them. Their study divided the features of feedback into cognitive and affective components. According to their findings, summarization, specificity, explanations, and scope are cognitive in nature. Cognitive features of a review are expected to most strongly affect understanding. This explanation helped us to identify suggestions and problem detection as a property of quality feedback.

McGrath and Taylor studied students' perception of helpful feedback for writing performance (McGrath et al. (2011)). Their study defined quality feedback ("developed feedback") as clear, specific, and explanatory in nature. They measured students' perception of developed

feedback by having them rate the feedback on a Likert scale. The results showed that students rated developed feedback highly for helpfulness.

A survey of 44 students done by Weaver (2006) showed that, in order to use the feedback, students needed advice (suggestions). The analysis of the feedback content and students' responses uncovered that vague feedback (e.g., "Good job") is unhelpful, lacking in guidance (void of suggestions), or focused on the negative (mentioning only problems), or was unrelated to assessment criteria.

Ramachandran et al. (2017), developed an automated system to evaluate reviews and show how they compared to other reviews for the same assignment. They extracted attributes like relevance to the submission, content, coverage, tone, and volume of feedback to identify a good-quality review. They constructed word-order graphs to compare the reviews with submission text and extract features from the reviews.

To identify localization (where the review pinpoints the place where a revision should be made) and make suggestions to improve the review, Nguyen et al. (2017) applied natural language processing techniques. They provided real-time formative feedback to reviewers on how to localize their review comments.

Zingle et al. (2019) used neural-network approaches to find suggestions in the review text, and compared them against rule-based NLP approaches. In a similar work Xiao et al. (2020) used NLP techniques with several ML and neural-network approaches to identify problem statements in review text. Our work takes this a step further and asks whether it is enough for a review to detect problems, or whether reviews that also make suggestions are more helpful.

According to the discussed studies, the core features of quality reviews are mentioning issues and providing suggestions to resolve the issues. However, these studies primarily focused on the qualitative study of the review quality. In our study, we implemented methods to automatically identify these features in the peers' review comments. Combining a qualitative and quantitative analysis approach, we analyzed the review quality that students found more helpful.

2.2 Peer assessment rubric analyzer

An approach to improve review quality is to provide the reviewer with a rubric defining the characteristics of a quality review. Jaco du Toit (2019) conducted a study to identify the impact of peer review on essay assignments. The study showed that giving students a rubric describing the characteristics of a good essay can provide them with the insight to produce better quality assessments than they would otherwise produce.

In this section, we discuss some studies we were encouraged by and helped us find direction for our study on analyzing rubric quality. The related works can be separated into two sections:

1) Analyzing rubrics 2) Analyzing review comments. In Section 2.2.1, we discuss studies on the impact of quality rubrics on encouraging peer assessors to write quality feedback. In Section 2.2.2, we present the studies that analyzed review comments to identify the quality of the review.

2.2.1 Analyzing rubrics

Ashton and Davies (2015) acknowledged that providing guidance to peer assessors can enhance evaluative outcomes. Specifically, providing a guided rubric improved students' ability to distinguish novice from advanced performances for certain rubric items, though not all. Students without guidance struggled with subjective or technically complex aspects, often making errors. The study suggests that effective peer assessment relies on quality rubrics, and it aligns student feedback more closely with expert evaluations.

Jonsson and Svingby (2007) surveyed 75 studies on the use of rubrics regarding validity, reliability, and educational consequences. Their work provided a comprehensive list of studies that show, in the context of peer and self-assessment, that providing rubric criteria helps students give more extensive feedback. The authors concluded, "rubrics support learning and instruction by making expectations and criteria explicit, which also facilitates feedback and self-assessment."

Murillo-Zamorano and Montanero (2018) divided 96 students into two groups for an academic presentation test. One group received feedback from the course teacher and the other group obtained peer feedback as a part of peer assessment where the peers followed a rubric to provide formative feedback. Their study showed student group who received peer feedback improved by 10% after the intervention, and the group with teacher's feedback improved by 5%. The rubric was designed to provide feedback on the presentations as well as suggestions for improvement. A study by Jaco du Toit showed that poor-quality feedback caused a poor understanding of the quality of students' work and might even have felt a false sense of accomplishment. Giving peer assessors a rubric that referred explicitly to the features of a good essay gave students the insight to produce effective reviews.

2.2.2 Analyzing review comments

In this sub-section, we discuss past studies that analyzed different features in the review texts and their methods.

Ramachandran et al. (2017) developed an automated system that evaluates reviews and provides suggestions on writing better reviews. They extracted attributes like relevance to the

submission, content, coverage, tone, volume of feedback provided and plagiarism from the review text to identify a good-quality review.

To identify localized review comments and make suggestions to improve reviews, Nguyen et al. (Nguyen et al. (2017)) applied natural language processing techniques. They provide formative feedback in real time to reviewers to make localized comments in their reviews.

Zingle et al. (2019) achieved good results using neural-network-based approaches to find suggestions in the review text and compared them against traditional rule-based natural-language processing and machine learning-based approaches for finding suggestions.

Xiao et al. (2020) identified problem statements in peer assessments. This study used several classical machine learning and deep learning approaches to identify problem statements. However, simply giving a problem statement may not show authors how to improve their work.

Jia et al. (2021) applied multitask learning techniques to detect suggestions, problem detection, and positive tone in the peer-review text. In this study, they compared single-task and multi-task learning using BERT and Distil-BERT models.

These studies indicated that rubrics are essential to provide guidelines to peer assessors. It helps the assessors write good-quality reviews. The mentioned studies also indicated the qualities of good reviews and methods to analyze reviews automatically. However, there is no study on automatically analyzing the quality of a rubric. More specifically, an automated rubric analyzer that can identify whether a rubric will induce peer assessors to write quality comments.

2.3 Automated meta-review for locating (dis)agreement

In peer assessment, misaligned or conflicting feedback can confuse students and complicate grading for instructors. It is beneficial to identify and resolve these discrepancies before providing feedback to the reviewees. While summative assessments with assigned scores make it easier to spot inconsistencies, formative feedback presents a more complex challenge in identifying agreement or disagreement ((dis)agreement). There is a dearth of literature addressing this issue. We present a few studies that tackled similar challenges in different domains.

2.3.1 Identifying (dis)agreement

Rosenthal and McKeown (2015) identified (dis)agreement in conversation by supervised learning. They demonstrated the impact of classification (dis)agreement with different feature extraction methods based on meta-thread of conversation, lexical features, sentiment, sentence similarity, and accommodation. Their study showed that meta-thread and accommodation

are captured in the *semantics* of the text, which helps most to identify (dis)agreement.

Galley et al. (2004) implemented a statistical approach to identify agreement and disagreement in conversation. They argue that adding *contextual* information to account for classification tasks improves accuracy. They capture contextual information by using adjacent words and maximum entropy ranking based on a set of lexical and structural features of words in bi-direction.

Hillard et al. (2003) implemented a supervised (decision tree) and unsupervised (clustering) classifier to identify (dis)agreement in automatically transcribed meetings. They used hand-labeled spurts for feature extraction and clustering algorithms to identify (dis)agreement in the ICSI Janin et al. (2003) corpus.

Hiray and Duppada (2017) argued that Siamese-inspired architecture encodes the online discussions better than hand-crafted feature encoding to identify (dis)agreement. They classified the discussions into three categories (agreement, disagreement, or none) and compared their result on three corpora (ABCD, IAC, and AWTP).

Our work differs from these previous studies in three major categories. First, our definition of (dis)agreement differs from the mentioned works. In the scope of our study, if two peer assessors find different issues in an artifact, we count it as disagreement. Second, most of the previous studies are based on online discussions or meeting transcription. We are using peer assessors' feedback comments from a technical course, where the assessors are providing independent reviews. Third, the previous studies used open-sourced annotated data. They compared their results with accuracy scores.

2.3.2 Feature Extraction and clustering

Reviews of products or short texts on social media are more readily available than feedback comments in the educational environment. Though peer reviews are more specific to a domain, the length of the reviews can be of similar length to product reviews. Understanding methods to analyze short texts will allow us to have ideas about analyzing feedback texts.

Guan et al. (2020) found that bag-of-word-based feature vectors are high in dimension but sparse, which makes it difficult for the clustering models to identify the similarities based on semantic similarity. They proposed a deep feature-based text clustering framework that uses a deep text encoder to process words and provides a semantic representation of texts. They compared their results on AG news, DBpedia, Yahoo! Answers, R2, and R5 corpora using accuracy scores.

Jinarat et al. (2018) identified that a major characteristic of short texts (e.g., Facebook comments and posts, tweeter text, news headlines, product reviews, etc.) is they lack context

information and contain jargon. These affect the performance of traditional text clustering algorithms. They proposed a new clustering technique using a word semantic graph where each word is a node and consists of a vector. Each word vector is obtained using the Word2Vec word embedding model by Mikolov et al. (2013).

Sun et al. (2019) analyzed reviews from Airbnb using K-means clustering to verify the Superhosts in Airbnb. They used Word2Vec embedding-based with K-means clustering to identify the services that reviewers expressed their opinion on.

Abbasi-Moud et al. (2021) proposed a system to provide personalized recommendations to tourists. They extracted tourists' reviews from tourism social networks. They clustered the comments using semantic similarities and analyzed the sentiment of the tourists to understand their preferences. The semantic similarity of the reviews was inferred by constructing a noun-similarity matrix. In our problem, a noun-similarity matrix will not be an ideal way to cluster the review similarities. The comments from the peer reviewers may not contain the same noun, but they may express a similar idea.

2.4 Reliability-based aggregated grading

Formative feedback provides the students with insights into improving their work. However, peers' summative feedback helps to provide grades to the students. Each artifact is generally graded by more than one reviewer. A common approach to calculating students' grades from multiple assessors is to utilize aggregate-based methods like mean and median. However, not all assessors are competent in providing accurate grading; as a result, simple aggregation methods are not ideal for summative feedback Darvishi et al. (2020). Sridharan et al. (2019) found that students can accurately and consistently assess the performance of their peers, particularly in the formative evaluation. However, the results also indicated significant inaccuracies in peer grading when summative feedback was included in final grades. Several studies have proposed methods to aggregate ratings from multiple peer assessors based on their credibility.

Calibrated Peer Review (CPR) is a common approach to produce a reliability index of peer assessors. In CPR, students assess a benchmark sample. The discrepancy between the benchmark and the peer assessor's ratings is used to calculate a ranking of the assessor's accuracy or reliability. Carlson and Berry (2003); Balfour (2013). However, CPR incorporates an additional step for the peer assessors. Additionally, students may not be consistent over the time of multiple assessments, which CPR cannot take into account Capuano et al. (2016).

Hamer et al. (2005) proposed a score calibration approach for peer assessors' grades where their algorithm generated a grade for a student's essay and a weights for each peer assessor that graded the essay. The weight of the peer assessors indicated the quality of the assessment

by that reviewer. This approach initially takes the mean of the peer assessors' ratings as the ideal score and iteratively calculates the weights for the assessors and the weighted average until the algorithm coverage. The weight of the assessors relies on the distance from the mean of the ratings.

Goldin (2012) applied Bayesian inference to quantify the bias of peer graders on each rubric criterion. Quantifying the bias enabled the provision of tuned grades to the peer assessor. A similar approach by Piech et al. (2013) introduces three probabilistic models for adjusting students' grades. These models analyze the grading performance on ground truth ratings, which are provided by the instructor. Their models estimate the reliability and bias of each grader, reflecting their tendency to inflate or deflate grades. Subsequently, the reliability and bias of each grader are utilized to calibrate future grades. In their approach, they also implemented models that consider the assessors' history of quality assessment and their scores in the assignments. Zarkoob et al. (2023) implemented a probabilistic peer grading model that estimated peer assessors' reliability, bias, and effort in the assessment process. In our study, we implemented a probabilistic model using Bayes inference. However, our approach uniquely utilizes formative and summative assessments to infer the reliability of peer assessors.

CHAPTER

3

DATA

This chapter is dedicated to describing and explaining the data we used in our studies. In Section 3.1, we describe the data source, collection, and annotation process. In Section 3.2, we described how we secured the privacy of personal information of students and the integrity of the data.

3.1 Data Source: Expertiza

The data we acquired for the studies in this dissertation is from Expertiza system (Gehring (2010)). Expertiza is an NSF-funded peer-assessment platform where students submit their work for assessment, and peer reviewers assess the work based on a set of criteria found in a rubric created by the instructor. For our studies, we utilized the Expertiza peer assessment system in an object-oriented design and development course (CSC517) taught by Dr. Edward Gehring at NC State University. Before beginning the review process, reviewers are shown examples of good-quality review comments. They are encouraged to write a review that explains any issues with the work or why they liked or disliked it. The system allows for multiple rounds of review, and peer review is typically followed by instructor grading. For example, the "Program 2" assignment of the mentioned course assesses students' performance in developing a Ruby on Rails web application. This assignment incorporates two rounds of peer assessment followed

by a meta-review process:

1. **Formative feedback:** After students submit their assignments, the assignment enters a review phase, where students review their peers, usually in double-blind fashion. The reviewers follow the provided rubric to evaluate and provide textual feedback.
2. **Summative feedback:** After students have had an opportunity to revise their work based on the feedback received from their peers, another rubric is presented to the reviewers, where they are asked to score the work on various criteria, along with comments to explain their decisions.
3. **Meta-review:** After both rounds of peer-assessment are over course instructor and teaching assistants evaluates the review provided by the peer assessors.

The summative score provided by student reviewers may be used as the grade for the assignment, but this is unusual. Normally, instructors assign the final grade, taking into account the issues noticed by peer reviewers.

3.1.1 Data Annotations

For our study, we used several datasets to train machine learning models to identify particular features in the review text. Chapters 4, 5, 6, and 7 include a more in-depth discussion of the text features. Generally, a small number of people cannot annotate a large dataset. It is better to have a large number of people each undertake a small number of annotation tasks; this lessens the chance that an annotator will become fatigued and assign inaccurate labels. We engaged students in the labeling task by offering a small amount of extra credit. After receiving peer feedback, students were asked to label the feedback for features like the problem mentioned, suggestions provided, helpfulness, localization, etc. In different assignments, students were asked to label the feedback for different characteristics; the same comments were not necessarily labeled for all the mentioned characteristics. After labeling was complete, the course instructor and TAs spot-checked the data that each student labeled. If any labels were found to be incorrect, the data labeled by that student was excluded from the dataset.

Since the reviews were done on team projects, and labeling was done individually, two to four students had the opportunity to label (or “tag”) the same review comments. If multiple students did tag the same comment, inter-rater reliability (IRR) could be calculated. We chose Krippendorff’s α Krippendorff (2018) as the metric for IRR. We chose this metric because it is not impacted by missing ratings, which were common since not all students availed themselves of the extra-credit opportunity. In an effort to use only the most reliable labeling, we included

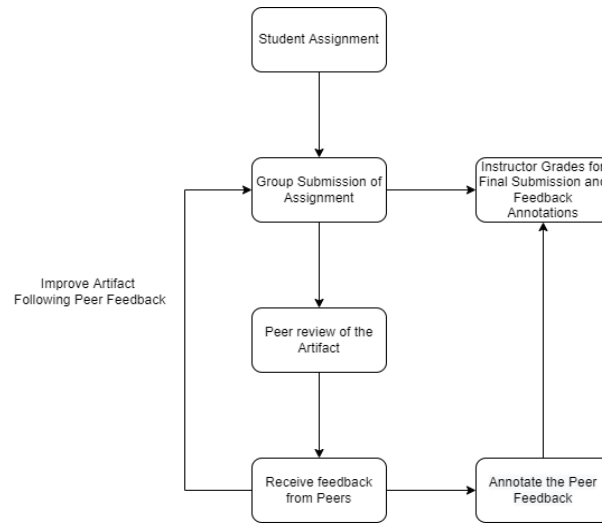


Figure 3.1: Flow diagram of peer review and feedback annotation process

only labels that were assigned (or not assigned) by *all* the students in the team that was reviewed. Figure 3.1 shows the peer-review and annotation process.

3.2 Data Privacy

The data we used for our studies were collected from real classrooms. To preserve students' data privacy, we took maximum care. The data procurement, annotation, and analysis were carried out following a five-step privacy-preserving process. First, the data for the studies were collected from a separate database that contains the ratings and comments submitted by the students, but not the identity of the students who submitted it. Second, we used regular-expression techniques to remove identifiable information like email, ID, or names. Third, any URLs that revealed any student account information were removed. Fourth, university-provided cloud storage was used to store the data. Fifth, data sharing was limited to only the involved researchers with IRB approval and following FERPA compliance.

CHAPTER

4

ANALYZING HELPFUL FEEDBACK FROM THE STUDENT'S PERSPECTIVE

4.1 Introduction

Peer assessment can be a more effective pedagogical method when reviewers provide quality feedback. But what makes feedback helpful to reviewees? Previous studies (Hattie and Timperley (2007); Nelson and Schunn (2009); Weaver (2006); Xiao et al. (2020)) have identified quality feedback as focusing on detecting problems, providing suggestions, or pointing out where changes need to be made. However, it is important to seek students' perspectives on what makes a review helpful to a reviewee. This study explores the helpfulness of feedback from students' perspectives when the feedback contained suggestions or mentioned problems or both. We implemented natural language processing techniques to identify suggestions and problems mentioned in peer reviews. We also analyzed important text features that are associated with suggestions or problems detected by peer feedback. The result showed that students are likely to find a review helpful if a suggestion is provided along with the problem mentioned in the feedback rather than simply identifying the problem.

The learning experience in a peer review environment depends on the quality of the reviews provided by the assessors. Each assessor is expected to provide a review detailing their assess-

ment with suggestions to improve an artifact. However, not all assessors provide constructive feedback due to lack of knowledge on the topic or simply carelessness. Instructors typically need to meta-review to ensure the quality of the review comments. This consumes a good portion of the time that would be saved by having students provide quality feedback.

Previous studies (Zingle et al. (2019); Xiao et al. (2020)) have suggested methods to reduce the meta-review burden of instructors by automatically detecting the characteristics of a quality review. That raises the question of what defines a quality review. According to Nelson and Schunn (2009); Carless and Boud (2018); Ramachandran et al. (2017); Nguyen et al. (2017), high-quality feedback consists of (i) identifying a problem and (ii) suggesting a solution. However, their finding was based on students' performance and not from their (students') perspective. It is important to identify whether "quality feedback" is actually helpful to the reviewees, based on students' opinion of which feedback is helpful.

In this dissertation, we propose a method using natural language processing (NLP) and neural networks to automate the process of analyzing and classifying reviews to discover whether they contain suggestions and/or problems. We have analyzed the words used to include suggestions or problems in the feedback. If we consider that a feedback comment containing suggestions, problems, or both is a "quality comment," we can analyze whether quality comments are helpful from the student's point of view. The related previous studies are discussed in Chapter-2 section 2.1.

4.2 Related Work

In peer assessment, assessors provide both qualitative textual feedback for formative purposes and numerical scores for summative grades. Studies indicate that students benefit more from formative feedback, which provides insights into their work and engages them in active learning and metacognitive thinking (Graner (1987); Rada et al. (1994); Çevik (2015); Li and Grion (2019); Gehringer (2014)). Conversely, poor reviews can confuse students about the quality of their work, sometimes giving them a false sense of accomplishment (du Toit (2019)). Effective peer learning depends on the quality of reviews, which should answer three key questions: (i) What are the goals? (ii) What progress is being made toward the goals? (iii) What activities are needed to make better progress? (Hattie and Timperley (2007)).

Quality feedback should identify problems and suggest ways to reach goals. Key features of good-quality reviews include summarization, specificity, explanations, and scope, which are cognitive in nature and strongly affect understanding (Nelson and Schunn (2009)).

Studies show that clear, specific, and explanatory feedback is perceived as more helpful (McGrath et al. (2011)), and that vague feedback is unhelpful and lacks guidance (Weaver

Table 4.1: Sample review comment and annotations done by students ('1' indicates 'yes' and '0' indicates 'no')

Review Comment	Detects Problem
The Travis CI Build is Failing as of now. No conflicts as per the GitHub report.	1
Yes, the explanation is elaborative and complete.	0
Since the build failed, I would not recommend adding it to the production server yet.	1

Review Comment	Gives Suggestion
Test Plan is too verbose. Trivial areas can be trimmed off.	1
The team needs to look into Travis CI log & 1	1
Many test cases in terms of controllers, but none for models.	0

Review Comment	Is Helpful
The build is failing due to 4 failures in the model specs.	1
The writeup is clear.	0
Since the build failed, I would not recommend adding it to the production server yet.	1

(2006)). Automated systems have been developed to evaluate reviews, extracting attributes such as relevance, content, coverage, tone, and volume (Ramachandran et al. (2017)). Natural language processing techniques have been applied to provide real-time formative feedback to reviewers and to identify suggestions and problem statements in reviews (Nguyen et al. (2017); Zingle et al. (2019); Xiao et al. (2020)).

The core features of quality reviews are mentioning issues and providing suggestions to resolve them. Our study extends this by combining qualitative and quantitative methods to automatically identify these features in peer review comments, analyzing the review quality that students find more helpful.

4.3 Data

Machine learning and neural network-based models can perform as well or as badly as the data they are given. However, obtaining good labeled data is expensive. For the purpose of our experiment, we have collected 3 different labeled datasets with rigorous quality control:

- Problem-detection: A review comment is labeled yes or no according to whether it detects a problem.
- Suggestion-detection: A review comment is labeled yes or no according to whether it contains a suggestion.

- Helpfulness-detection: A review comment is labeled yes or no depending on whether the reviewee found it helpful.

We acquired this labeled peer-review data from the Expertiza system in a systematic manner. Expertiza is a system to support different kinds of communications that are involved in the peer-assessment process. It supports double-blind communications between authors and reviewers, assessment of teammate contributions, and evaluations by course staff. For this study, we collected peers review comments from the Object Oriented Design and Development course at North Carolina State University for about three years. This course implemented peer assessment using the Expertiza system to provide feedback to the students. In each semester, this course typically assigns three peer-reviewed assignments to students, who work in teams consisting of two to four members. Even though the assignments are done in a group setting, the submissions are reviewed by individual students from other groups. After receiving the reviews from peers, teams revise their work and resubmit it for grading. The second round of the assessment is generally summative, where, along with textual comments, the peer-reviewers assign scores to the submission. However, the instructor assigned the final grades.

Following the described annotation process in Chapter 3 section 3.1.1, we accumulated 18,392 annotations for problem-detection, 7,416 for suggestion-detection, and 3,970 for helpfulness-detection datasets. All the three datasets have an equal ratio of the binary class labels (i.e., they are balanced). Sample comments from the three datasets are shown in Table 7.1

4.4 Method

Our goal in this study is to analyze students' perspectives on helpful comments that mentioned problems and/or suggestions. To conduct the study, we had students annotate comments on the basis of whether they found them helpful. We need an automated process to identify those review comments that contain suggestions and/or problem statements. We first train a model (the problem-detection model) to classify reviews that contain a problem statement by training and testing with the problem-detection dataset. We build a second model (the suggestion-detection model) to classify the presence of suggestions in a review comment by using the suggestion-detection dataset. As model performance matters, we applied several ML and neural-network models to pick the most accurate models for annotating the helpfulness-detection dataset. Figure 4.1 shows the annotation process of the helpfulness-dataset using the models.

When approaching a classification problem by any type of machine-learning (ML) or neural network models, there are many different approaches to choose from. No one model is best for

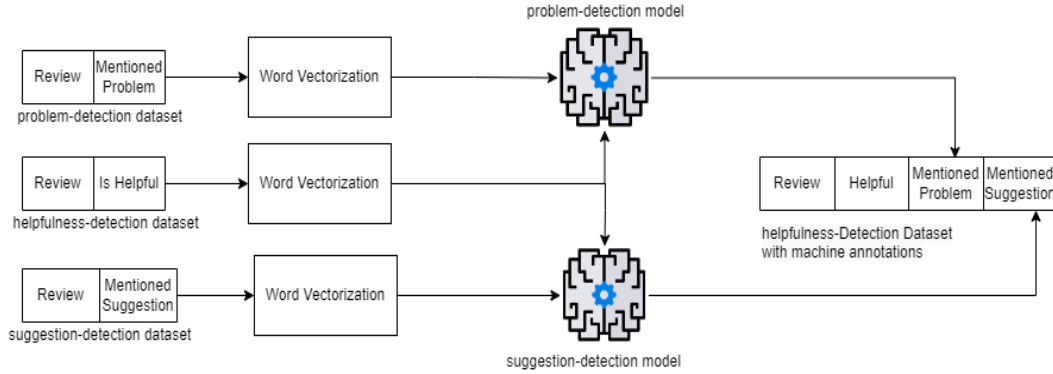


Figure 4.1: Annotation process of the helpfulness-dataset for mentioned problem and suggestions in the comments using models. The models were trained for detecting problems or suggestions mentioned in the review text. The training datasets were annotated by human (students).

all problems. In our study, we have chosen Support Vector Machine (SVM), Random forest (RF), classical ML models and compared their performance with Bi Directional Long Short-term Memory (Bi-LSTM), and Bidirectional Encoder Representations from Transformers (BERT) models. We use our problem-detection dataset and suggestion-detection dataset on these model with 80:10:10 ratio for training, testing and validation.

4.4.1 Classical Machine-Learning Models

Input Embedding with TF-IDF

Machine-learning models are suitable for capturing complex relationships between the input data. But they require numeric input. The review data that we have in our dataset is textual. We have to convert them to numbers and also allow the model to capture the important features of the text. One way to do that is term frequency-inverse document frequency (TF-IDF). TF-IDF measures the importance of a word in a document using statistical calculation. If a word appears more times in a document the importance of the word in the document increases proportionally. We used scikit-learn Pedregosa et al. (2011) library to implement TF-IDF and vectorize the words in the feedback.

Support Vector Machine (SVM)

SVM is very popular for high accuracy and low computational cost. For a classification problem between two classes, SVM maximizes the margin of the separation plane between the two classes. We provided the feature vector of the reviews converted by TF-IDF to the SVM model

to classify the review for having a particular property (contains problem or suggestion in the comment). We applied a grid search to find the best inverse regularization parameter C .

RF

We used Random Forest for its popularity to make more accurate classification with a simple approach. RF makes an ensemble decision from a forest consisting of multiple uncorrelated decision trees. The general idea of the RF is that the decision from individual decision trees increase the accuracy of overall result. We varied the number of decision trees and depth of the trees to get the best result. We used TF-IDF for making feature vectors from the review text.

4.4.2 Neural-Network Models

Input Embedding

Neural network models are popular for text classification tasks. However, to improve the performance of the neural-network models on the text data, it is necessary to represent the data that is suitable for the model to work with, and without losing the underlying latent relations among the features of the data. For our experiment we have used Global Vectors for Word Representation (GloVE) with Bi-LSTM. GloVE not only measures the statistical significance of words, it also considers the statistical co-occurrence and semantic relation of the words.

Bi-LSTM

Bi-Long Short-Term Memory is in general used for sequential data classification tasks. It is a good fit for peer-review texts. Review comments are sequential data, and the words of the text have latent semantic and contextual relations with each other. As Bi-LSTM model takes input from both right and left direction of the text, it can capture the relationship between the words in texts occurring in any order.

BERT

BERT is based on Transformer model and uses attention mechanism to learn the contextual relations of the words in a sentence. Being a bi-directional input reader, BERT learns the context of the word in a sentence by considering words occurring before and after.

Table 4.2: Hyperparameters of Models

Model	Hyperparameter
SVM	c=1
RF	tree = 100 max depth = 4
Bi-LSTM	maximum text length = 300 Embedding = 300d Hidden layer activation = ReLu dropout = 0.4 optimizer = Adam Output layer activation = Sigmoid Epoch=20
BERT	optimizer = AdamW Learning rate = 2e-5 Epoch=4

4.5 Results

In this study, our first step is to construct two separate models where one identifies whether feedback contains a problem statement and another identifies whether feedback contains a suggestion. To identify the best-performing models, we trained and tested the performance of several classical ML models and neural-network models and compared their performance.

Figure 7.4 reports the comparison of the F1-score values of the classical machine-learning (ML) models and neural-network models on the problem-detection dataset and suggestion-detection dataset. To compare the performance of the models we use the F1 score, as this represents the harmonic mean of precision and recall.

- **On the problem-detection dataset:** Among the classical ML models SVM made the highest F1 score 0.90 and among the neural-network models on the problem-detection dataset, the BERT model has the highest F1 score, 0.92.
- **On the suggestion-detection dataset:** BERT achieved the highest F1 score, 0.91. Among the classical ML models, SVM achieved the highest F1 score, 0.87.

As BERT outperformed all other models on both problem and suggestion datasets, we trained two separate BERT models to annotate the feedback comments contained in the helpfulness-detection dataset. The BERT-created annotations recorded whether each comment in the helpfulness-detection dataset detected a problem or offered a suggestion. The models annotated each comment with either “1” or “0”, indicating having the property or not. We perform an **and**-operation using the BERT-created annotations. If both the problem and suggestion were

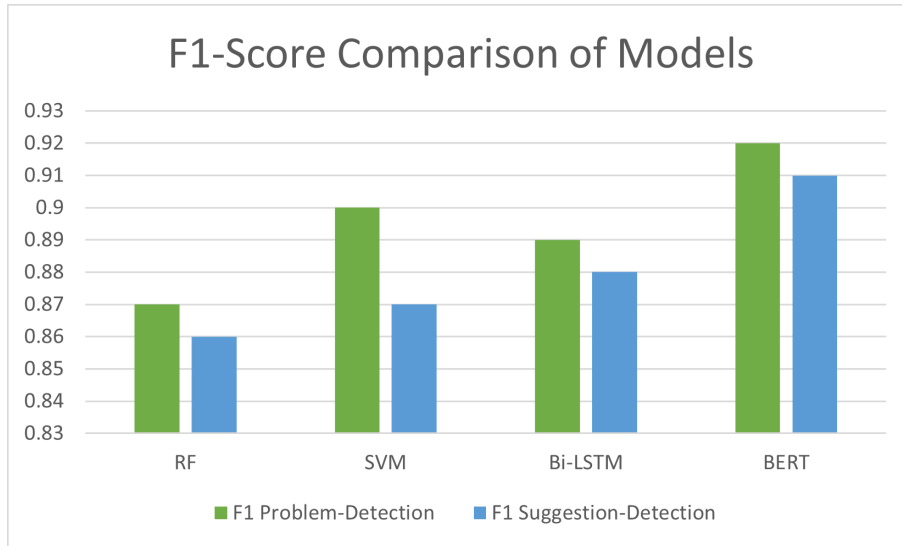


Figure 4.2: F1-score comparison to measure performance on classifying review text on problem detection and suggestion detection using classical ML and neural-network models. In overall F1-score comparison, the BERT model shows the best performance.

Table 4.3: Table shows sample comments from helpfulness-detection dataset and corresponding annotations. Note that “Is Helpful” annotations are done by humans (students), while “Detects Problem” and “Gives Suggestion” are annotated by the BERT model. “Contains Problem and Suggestion” is from **and**ing the “Detects Problem” and “Gives Suggestion” columns.

Review Comment	Is Helpful (human-annotated)	Detects Problem (machine-annotated)	Gives Suggestion (machine-annotated)	Contains Problem and Suggestion (and-operation)
The build is failing due to 4 failures in the model specs.	1	1	0	0
The writeup is clear.	0	0	0	0
Since the build failed, I would not recommend adding it to the production server yet.	1	1	1	1
I would recommend adding more code for helping following their changes.	1	0	1	0

mentioned in a comment the **and**-operation yields 1, otherwise 0. The resulting helpfulness-detection dataset is shown in Table 4.3.

After we computed the annotations for problem detection and suggestions, we did a Venn diagram analysis on the updated helpfulness-detection dataset. The diagrams illustrate the overlap of comments that both detect a problem and offer a suggestion. Figure 4.3(a) shows that 1,985 comments in the helpfulness-detection dataset were annotated by students as being helpful. Among the helpful comments, 1,417 were annotated for having problems and/or suggestions mentioned in the feedback. Out of these 1,417 helpful comments, 912 of them were machine-annotated as containing both problem detection and suggestions. A total of 568 helpful comments did not have any problem or suggestion mentioned, based on machine-annotation.

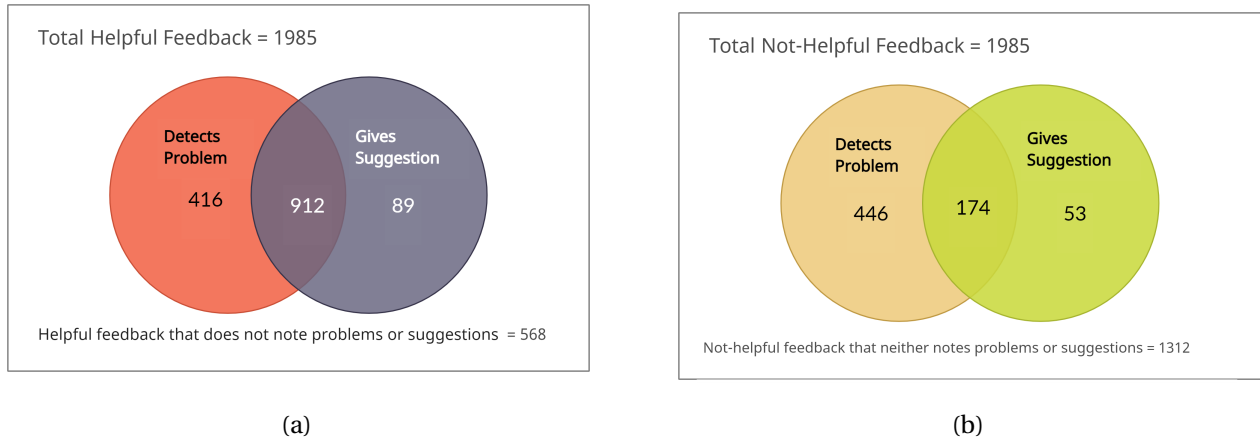
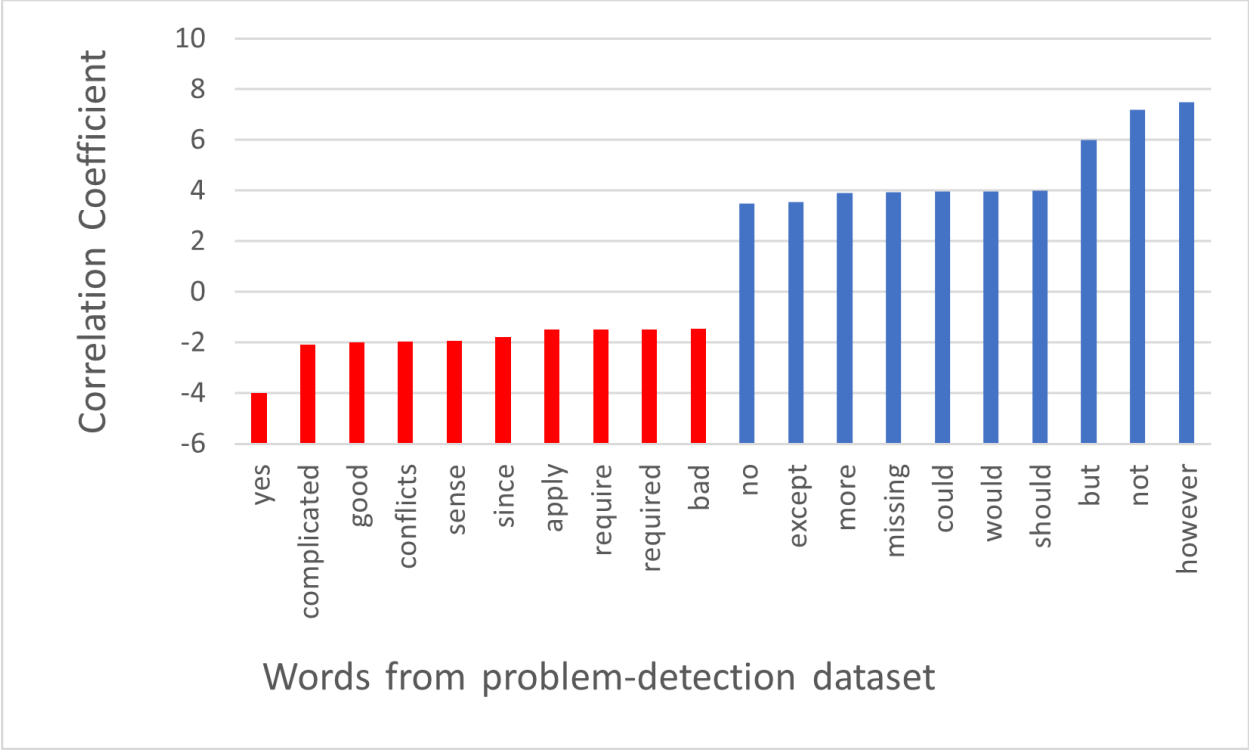


Figure 4.3: Venn diagram of helpful feedback annotated for mentioned suggestion and/or problem

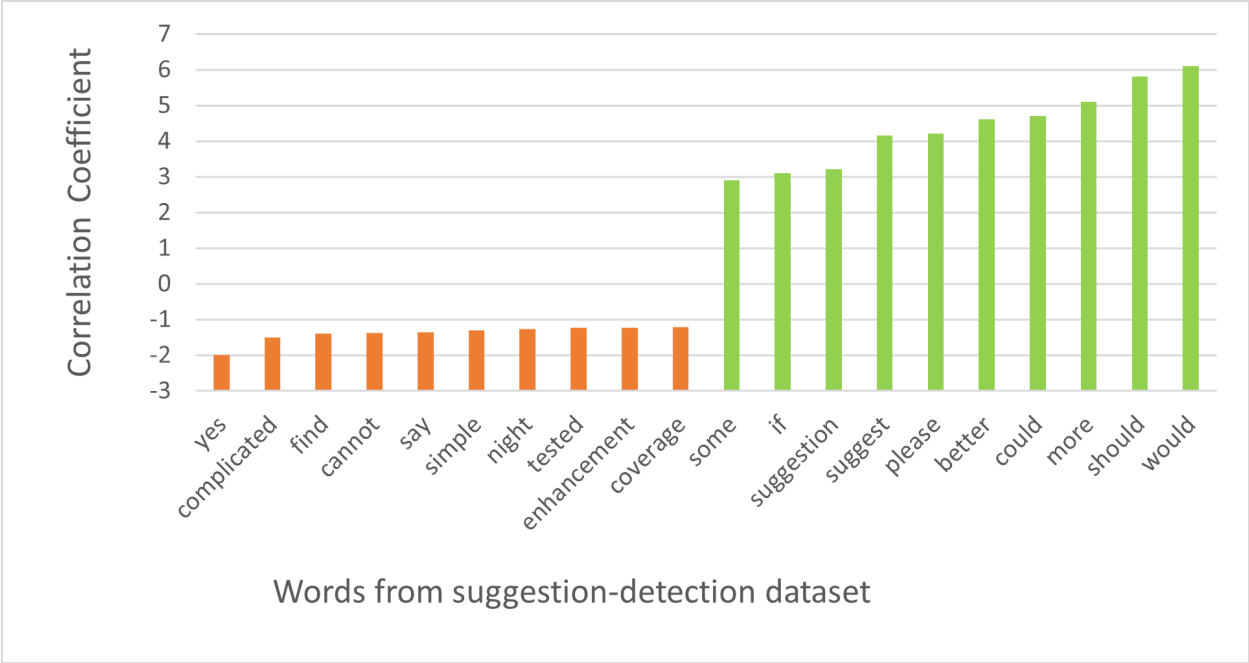
On the other hand, out of the 1,985 comments that were human-annotated as not helpful [Figure 4.3(b)], 673 comments were annotated as having either a problem and/or suggestion mentioned. Among those 673 comments, only 174 were annotated as having both suggestion and problem mentioned. A total of 1,312 comments that did not have any problem or suggestion mentioned were annotated as not helpful by the students.

To summarize the Venn diagram analysis, comments that the students found helpful mostly detected problems and/or contained suggestions. However, among those comments noting suggestions and/or problems, students annotated as helpful mostly comments that *both* pointed out problems and gave suggestions. This indicates that peer feedback is more helpful to the students when a suggestion is given in a comment that detects a problem. On the flip side, Figure 4.3(b) suggests that students rarely find comments helpful when they do not mention any problem or contain a suggestion.

To gain a deeper insight into the words that are highly correlated with text where a problem mention or suggestion is mentioned, we analyzed the top 10 positive and negative correlation coefficient values calculated by the SVM model. Figure 4.4(a) shows the coefficient values that the problem-detection model has calculated for various words. Note that it has positive coefficient values for words such as “however”, “but”, and “not”. In the English language these words are more likely to be used when stating problems. Similarly words like “yes”, “completed” and “good” are not likely to occur in a problem statement. Figure 4.4(b) shows that the suggestion-detection model has positive coefficient values for words like “should”, “would”, “more”, “suggest”. These words are likely to be used in suggestions. On the other hand, words “yes”, “completed”, and “cannot” are more likely not to be used to express suggestions; thus



(a)



(b)

Figure 4.4: Top 10 positive and negative coefficient values of words from problem-detection and suggestion-detection dataset

they have negative coefficient values.

4.6 Discussion and Conclusion

This study constitutes the first analysis of the helpfulness of peer-assessment feedback from student’s perspective. Feedback that mentions problems or includes suggested changes was considered to be quality feedback. We used natural language processing (NLP) techniques in conjunction with several ML and neural networks to identify quality peer feedback. We systematically collected and scrutinized 18,392 comments mentioning problems, 7,416 comments containing suggestions, and 3,970 comments that were annotated by humans (students) as being helpful.

Using the annotated dataset, we trained our ML and neural-network models to identify quality feedback. For identifying suggestions and problems mentioned in the review text, the BERT model outperformed the other models. As the BERT model focuses on the important features of the text, it was best at identifying suggestions and problems in the feedback. We used the BERT model to automatically annotate comments as mentioning problems or making suggestions, and compared these annotations with comments that students had manually annotated as being helpful. We also analyzed important words that are frequently present in comments mentioned a problem or suggestion.

A key question is whether comments automatically annotated as “quality” (meaning that they both identified a problem and gave a suggestion) were the same comments that the students considered helpful (that they manually labeled as helpful). Among the comments that the students considered helpful, 64% of them both mentioned a problem and gave a suggestion. Conversely, of the comments that the students labeled as not helpful, 66% of them neither mentioned a problem nor contained a suggestion.

The results indicate that the automated annotation performed by the BERT model can be very effective in predicting which comments students will consider helpful. While it can’t deliver an actual count of helpful comments in a particular review, that is not important. It *can* determine whether the feedback provided by the reviewer contains a substantial number of quality comments. That is what is needed to automatically detect helpful reviews.

CHAPTER

5

PEER ASSESSMENT RUBRIC ANALYZER: AN NLP APPROACH TO ANALYZING RUBRIC ITEMS FOR BETTER PEER-REVIEW

5.1 Introduction

Rubrics have long been used to provide a grading process that is fair and adherent to standards. Just as rubrics can help instructors assess a piece of work, they can also help students do a more effective job of peer evaluation. In a peer-review environment, reviewers provide formative feedback following the rubric criteria. High-quality feedback can greatly enhance the learning process. Rubric criteria need to be worded carefully to provide clear instruction and effective guidance. Heretofore, little research has been performed on how rubric text affects rubric feedback. This study focuses on analyzing rubric text to identify whether rubric criteria will induce peer reviewers to write quality reviews. We have analyzed 408,104 formative feedback comments based on 3,164 rubric criteria using natural language processing techniques with advanced neural network methods. To our knowledge, this is the first attempt to analyze rubric text to improve review comments for the peer-review environment. Our approach will assist in identifying suitable rubrics for peer reviewers that will induce peer assessors to write quality

reviews.

Rubrics were used for peer assessment in the physical classroom environment much earlier than in online courses. Students would typically exchange their artifacts with fellow students to review each other's work based on criteria provided by the instructor (Gehring (2014)). In the contemporary online learning environment, including massive open online courses (MOOCs), this approach is even more useful, as MOOCs courses can have an overwhelming number of students (Shah and Pickard (2017)) assessed by limited staff. In online peer assessment system, each artifact is generally anonymized and reviewed by more than one peer assessor and they can invest more time in assessment than an instructor, which allows them to be as accurate as assessment by an instructor (Sadler and Good (2006)), and more timely (Cambre et al. (2018)). Not only do students learn from receiving feedback, studies show that, students learn more from giving feedback than receiving (Rada et al. (1994)), are more engaged to study, learn actively (Topping (1998)), and think metacognitively (Gehring (2014)) during the peer assessment process.

However, the success of the peer assessment process heavily depends on peers' understanding of the provided rubric criteria to assess students' work (Topping (2010); Ashton and Davies (2015)). Rubric-based feedback can take two forms. With summative feedback, students often use a Likert scale to give a numerical value in a score range assigned for the rubric item. With formative feedback, they provide textual comments on the work they are evaluating. In most peer assessment environments, both methods are used. While the Likert value is useful in calculating the grade, the review helps the students most to get an insight into the issues with their work and make revisions to improve it. Low-quality formative feedback hampers students' understanding of the quality of their work and in turn diminishes the learning process. Black and Wiliam identified quality formative feedback as critical for recovering the learning gain (Black and Wiliam (1998)). Reinholz found that the quality of feedback is deeply dependent on the quality of the assessment rubrics and if the rubric is more focused on grades it may generate less supportive feedback (Reinholz (2016)). Despite the importance of the wording of rubric criteria, there is a dearth of research on analyzing the rubric text. As rubrics are written in natural language, they can be analyzed with modern natural language processing (NLP) approaches. Therefore, in this paper, we propose an automated rubric analysis approach that identifies whether a rubric item will induce peer reviewers to write quality feedback.

Before we analyze rubrics, we need to identify what constitutes quality feedback. According to Hattie and Timperley (Hattie and Timperley (2007)), model feedback must answer three major questions. First, Where am I going? (What are the goals?). Second, How am I going? (What progress is being made toward the goal?) and third, Where to next? (What activities need to be undertaken to make better progress?). This implies that quality feedback in the

education domain should include the specification of a problem that a student should work on as his/her goal and a suggestion to make progress as to how to reach the goal. (Nelson and Schunn (2009)) emphasized that feedback with specificity (localization) is highly correlated with the implementation of the feedback. Considering these features of quality feedback, in this study we focus on three properties of feedback to identify quality review comments:

- Detects a problem: If a problem is mentioned, the author's attention is directed to an issue that needs attention.
- Gives a suggestion: If a suggestion is mentioned in the review comment, the author can use it to improve the work.
- Is localized: If the comment identifies the exact location of the issue, the author can more easily address it.

In this study, we first take classical machine-learning approaches as baseline models and compare them with neural network methods for detecting features of quality reviews. We use NLP techniques for extracting features to classify review text. We calculate the likelihood that a response to a particular rubric item has each of the three characteristics of a quality review. Finally, we use advance neural network models to learn the features of the rubric items and classify them. We also report the correlation between (i) the length of the review text and the number of items in the rubric and (ii) the length of the review text and the position of the rubric item in the rubric. While applying the above-mentioned approaches we will address three research questions in this study:

- **RQ1: Which model performs best for classifying quality review text?**
- **RQ2: Can neural network models detect rubric items that enable peer reviewers to write quality reviews?**
- **RQ3: Is there any relation between rubric length or rubric-item position and length of the comments written by students?**

5.2 Related Work

Providing peer assessors with a rubric that defines the characteristics of a quality review can significantly improve review quality. Jaco du Toit (2019) demonstrated that giving students a rubric helps them produce higher quality assessments for essay assignments. Similarly, Ashton and Davies (2015) found that providing guidance through a rubric enhances evaluative

outcomes, particularly in distinguishing novice from advanced performances, though not for all rubric items. A survey by Jonsson and Svingby (2007) of 75 studies confirmed that rubrics make expectations explicit, facilitating better feedback and self-assessment. Another study by Murillo-Zamorano and Montanero (2018) showed that students who received peer feedback via a rubric improved by 10%, compared to 5% improvement with teacher feedback.

Studies indicate that rubrics are essential for guiding peer assessors to write quality reviews, as they clarify expectations and criteria. However, no study has yet focused on automatically analyzing the quality of a rubric itself, particularly in terms of its ability to induce high-quality peer comments.

5.3 Data

To allow a neural network model to predict whether a rubric item will induce peer-reviewers to write a review with the three quality features (detects a problem, gives a suggestions, is localized), we need to train the model by annotating sufficient numbers of rubric responses. There is no such ready dataset available to our knowledge. So we embarked on creating such a dataset following the below mentioned processes:

5.3.1 Collecting Rubrics and Review Comments

The data we acquired for this study is from Expertiza system (Gehring (2010)). Expertiza is an NSF-funded peer-assessment platform where students submit their work for assessment and peer-reviewers assess it based on a set of criteria found in a rubric created by the instructor. The system allows for multiple rounds of review, and peer review is typically followed by instructor grading. For example, if two rounds of review are used, they may proceed like this:

1. **Formative feedback** After students submit their assignments, the assignment enters a review phase, where students review their peers, usually in double-blind fashion. The reviewers follow the provided rubric to evaluate and provide textual feedback.
2. **Summative feedback** After students have had an opportunity to revise their work based on the feedback received from their peers, another rubric is presented to the reviewers, where they are asked to score the work on various criteria, along with comments to explain their decisions.

The summative score provided by student reviewers may be used as the grade for the assignment, but this is unusual. Normally instructors assign the final grade, taking into account the issues noticed by peer reviewers.

In certain courses, students have been given the opportunity to earn extra credit by *labeling* the review feedback they have received. They might be asked to label each review comment indicating whether or not it detects a problem, makes a suggestion, has positive tone, or is localized to a particular place in the reviewed work.

For this study, we collected 408,104 formative feedback comments from the Expertiza system. These review comments were written by peer-reviewers in response to 3,164 different rubric criteria. These rubric criteria made up 561 different rubrics created by 33 instructors from various courses. All of the comments were de-identified before being used for analysis, so that neither the writer of the comment nor the author of the reviewed work could be determined.

5.3.2 Collecting Labeled Review Comments

For the purpose of our study, we needed annotated review comments labeled as having, or not having, each of the following characteristics:

- Detects a problem: A review comment is labeled yes or no according to whether it detects a problem.
- Gives a suggestion: A review comment is labeled yes or no according to whether it contains a suggestion.
- Is localized: A review comment is labeled yes or no according to whether it points out an issue that is localized to a particular portion of the reviewed work.

As mentioned above, we engaged students for the labeling task with a small amount of extra credit. As the same students who received the reviews annotated these review comments, they would have a more accurate understanding of whether the comments contained the above-mentioned characteristics.

The annotation process of the data is mentioned in Chapter 3 section 3.1.1. After collecting the labeled data from the individual students, we take each labeled data for a consensus validation check. Since labeling was done individually, each feedback was labeled by more than one student. This allows us to calculate the inter-rater reliability (IRR). We chose Krippendorff's α (Krippendorff (2018)) as the metric for IRR. Krippendorff's α is not impacted by missing ratings which is often available in peer-review data as not all students are meticulous in their work. Table 5.1 shows the percentage of samples for the binary class labels. In Table 5.2 we showed some sample review comments and corresponding class labels.

Table 5.1: Percentage of samples for each class labels

Label	class	%of samples
Detects a Problem	0	50%
	1	50%
Is Localized	0	50%
	1	50%
Gives Suggestion	0	50%
	1	50%

Table 5.2: Sample review comment and labels for the classes

Review Comment	Detects Problem
The Travis CI Build is Failing as of now. No conflicts as per the GitHub report.	1
Yes, the explanation is elaborative and complete.	0
Since the build failed, I would not recommend adding it to the production server yet.	1

Review Comment	Gives Suggestion
Test Plan is too verbose. Trivial areas can be trimmed off.	1
The team needs to look into Travis CI log & 1	1
Many test cases in terms of controllers, but none for models.	0

Review Comment	Is Localized
The admin is still not able to sign in. The team has implemented book search functionality, However they have not updated it on heroku.	1
Yes they have included most of it as far as I know and the content is also quite intuitive.	0
can not find the admin account information	1

5.3.3 Annotating Review Comments

So far, labeling of review comments has been limited to the students. Though tens of thousands of comments have been annotated, it is still a small fraction of the 408,104 comments in our dataset. To determine whether the other comments had any of the characteristics under study, we turned to machine learning (ML) and deep learning models to infer labels for the remaining review comments. We trained three separate models to predict each of the three labels. With these annotations, we constructed a single dataset where each comment is annotated for the three mentioned features. We will call this dataset *Annotated_Review_Comment*.

5.3.4 Constructing the Dataset

The same rubric item can be used for a large number of reviews. All students in the course use the same rubrics for the reviews of each assignment. Frequently, students do several reviews per assignment, and instructors may reuse rubrics from semester to semester. As a result we have multiple review comment for each rubric item.

In our final dataset, we need a binary label (0 or 1) for each rubric item to indicate whether the rubric item is associated with the each of the three characteristics of a quality review (label 1) or not (label 0). We will use this dataset to train and test the classifier models.

As our model annotated all the review comments collected from Expertiza for three properties, this allows us to calculate a probability value that indicates how likely reviewers are to write a review comment with a particular property for that rubric item. For example, If a rubric item R_i has been used for a total of n comments ($C = C_1, \dots, C_n$), and m of these comments have the label 1 to indicate a problem mentioned in the comment, we calculate the likelihood (l_p) by

$$l_p = m/n \tag{5.1}$$

l_p indicates how likely a peer-reviewer is to write a review identifying a problem in a piece of submitted work using this rubric item R_i .

In similar way we calculate likelihood for suggestions (l_s) and localization (l_l) for each rubric item.

To label a rubric item "yes" (label 1) or "no" (label 0) for each of the mentioned properties we need to fix a likelihood cut-off point. Likelihood values above of the cut-off point we label "1" and below "0".

As it is a reviewer's choice to write a comment mentioning a problem, suggestion, or localization, it may occur that many of the reviewers of a rubric item are not engaged enough to write such a comment, regardless of the quality of the rubric item. So it is difficult to fix a hard boundary for the likelihood that a rubric item will induce a reviewer to write a comment with a particular property.

We will show our results with several cut-off points, to allow us to observe the performance of our models. After we select a likelihood value, all the values more than the likelihood we label as "1", and the values below we label as "0".

5.4 Methods

In this section, we discuss several machine-learning (ML) and deep-learning models that we used for classification and annotation tasks. We used the ML models as a baseline and compared them with the advanced neural-network models.

5.4.1 Classical Machine-Learning Models

When approaching a classification problem by any type of ML model, no single ML model is best for all problems. In our study we have chosen Support Vector Machine, Random Forest and Gradient Boosting as baseline models.

Input Embedding with TF-IDF

The review data in our dataset is textual and not suitable for our ML model. We have to convert them to vectors of numbers so the model can take the input and capture the important features of the text. For the ML models, we used the input-embedding method called *term frequency-inverse document-frequency* (TF-IDF). TF-IDF calculates the weights of each word in a document by a statistical measure of how important the word is to a document in a corpus. As the word appears more frequently, the importance of the word in the document increases proportionally, but the increment is also offset by the frequency of the word appearing in the corpus. We have used the TF-IDF vectorizer function in scikit-learn (Pedregosa et al. (2011)) for this process. TF-IDF is not able to capture contextual and semantic relation between words.

Support vector machines (SVM)

SVM is very popular ML model for classification tasks. SVM is preferred for its high accuracy and low computational cost. The objective of SVM is to draw a divider hyperplane in the N -dimensional input space to classify the data. The dimensions of the space vary by the number of features provided in the dataset. For a binary classification problem, SVM tries to maximize the margin of the separator plane between two classes. In our problem we are providing the SVM model the feedback from the reviewers and corresponding labels from the dataset. As the feedback comments are textual, we convert them to a vector using the TF-IDF technique. Each value of the vector acts as an input in the feature space for our SVM classifier. In the training phase it learns the features of our input data and tries to predict the separation plane or the decision boundary. SVM tries with different decision boundaries and chooses the one that works best for our dataset. We used a grid search to find the inverse regularization parameter C .

Random Forest

The Random Forest (RF) classifier is very popular for its accuracy in classification. This classifier consists of a number of individual decision trees that operate as an ensemble method. The approach is very intuitive, as each decision tree takes part in prediction, and the class with the most votes from each decision tree is the final prediction of the RF model. The number of decision trees and the depth of each tree are the hyperparameters for the RF model. We did a grid search to find the tree number and tree depth that performs best for our dataset.

Gradient Boosting

Gradient Boosting (GB) utilizes a number of weak models and ensembles the decision. GB uses gradients in the loss function. The loss function indicates how the model is learning the coefficients that represent the data well. For better performance we did grid search to find optimized tree numbers.

5.4.2 Neural-Network Models

Neural-network models are popular for analyzing text. We used three such models to classify the review texts and compared them with baseline models. However, it is important to use a vector representation for text data. In our study we used Glove embedding for text representation for Bi-LSTM, LSTM, and Convolutional Neural Network with Long Short-Term Memory models.

Glove Input Embedding

GloVE Pennington et al. (2014) is a word-embedding model that converts words into a multidimensional vector representation. An advantage of GloVE is that it is an unsupervised model, and the representation of the words are close to the semantic similarity. As a result, relations between words can be represented by this model, which is perfect for our experiment, as we want to convert the text to a vector representation that considers the semantic relationship between the words.

Long Short Term Memory (LSTM)

LSTM networks are “go to” models for a text-classification task. They are a type of recurrent neural network (RNN). RNNs have an issue of a vanishing gradient. As more layers use the activation function, at some point the gradient of the loss function approaches 0. That makes the model perform poorly and makes it difficult to train. LSTM solves this problem by using a gate that allows important information to be remembered for a time and unrelated information to be forgotten. We used an LSTM of size 100 (100 hidden units). We used dropout as a regularization mechanism.

Bi-LSTM

Simple LSTM models take only sequential data in a fixed direction. One disadvantage to that is that the model can only access words that precede the current input. As a result, the model cannot analyze the relationships between the current word and the words after it. But later words may give valuable information about the current word. Bi-LSTM models are

extensions of LSTM that can take input from both directions of a text, which allows this model to capture valuable information for analysis. For word embedding, we used the pre-trained GloVe embeddings that converted each word into a 300-dimensional vector embedding. The Bi-LSTM has 150 hidden units. For regularization we have used dropout and sigmoid for classification.

Convolutional Neural Network with Long Short-Term Memory

Convolutional Neural Networks (CNNs) are popular for reducing the dimensions of the features. LSTM can process the sequential data. One issue with LSTM is that it takes a long time to train the model. On the other hand, CNNs can be trained faster. Combining these two models will give the LSTM reduced feature data to train but obtain valuable information at the same time. We place a CNN before the LSTM to make a combined model. We have used a 1D CNN followed by a LSTM with 100 hidden units.

Bidirectional Encoder Representations from Transformers (BERT)

BERT is a multi-layer bidirectional transformer encoder and the current state-of-the-art language model. BERT uses a model named Transformer, which is an attention mechanism that learns the contextual relations between words in a sentence. An advantage of BERT is that, while most directional models read text input sequentially in a direction, the Transformer architecture of BERT reads the entire sequence of words at once. This feature of BERT allows it to learn the context of a word based on all of its surroundings. The BERT model we used was pre-trained on Wikipedia and BooksCorpus data. While inputting sentences to the pre-trained BERT model and extracted embedding with outputs from the second-last layer in the pretrained network. The Bert-Base-Uncased model (Devlin et al. (2018)) consists of 12 encoder blocks, 12 attention heads and 110 million parameters.

5.5 Experiment

In this section, we discuss the experimental setup for different models that we mentioned in the Methods section.

5.5.1 Selecting a Model for Review Text Annotation

Our initial objective is to make a data-driven automated process to label the 408,104 review comments for three separate properties (detects a problem, gives a suggestion, and is localized). As mentioned earlier in the Data section, we have three datasets that are annotated by

Table 5.3: Hyperparameters of ML Models

Classification Model	Hyperparameter
SVM	c=1
RF	tree = 100 max depth = 4
GB	estimator = 100 max depth = 1

the students for the required properties and the quality of the labeling process was checked meticulously. We have experimented with two different ratios (60:20:20 and 70:20:10) of the train, validation, and test split in each dataset. We report results for the 60:20:20 ratio, as it makes more test samples available. We used these datasets to train and test several classical machine learning (ML) models as baseline models and advanced neural network models. For different models, we used model-specific and generic hyperparameters for fine-tuning and better performance. Table 5.3 shows the different hyperparameters that we used for the ML models. Table 5.4 shows the hyperparameters of neural network models. We compared their results using evaluation metrics to choose a model that performs best to classify the annotated review texts.

After we fix the best-performing model on our labeled datasets, we enter the annotation phase where we use the selected model to annotate the unlabeled review texts for the mentioned three properties.

5.5.2 Classifying the rubric items

After the annotation of each review comment for three properties (detects problem, gives suggestion, and is localized) we calculate (as described in Section III.D) how likely it is that a rubric item will induce a peer reviewer to write a quality comment. Then varying the likelihood cut-off point, we annotate the rubric items. Each annotated dataset we split according to a 80:10:10 ratio for training, validation and testing, and classify them using the Bi-LSTM and BERT models.

5.5.3 Evaluation Metrics

For comparing the performance measure of the baseline ML models and the neural network models we used the F1-score, since the classes of the dataset are not imbalanced and the F1-score represents the harmonic mean of precision and recall.

To compare the performance of the BERT and Bi-LSTM models in classifying the rubric items,

Table 5.4: Hyperparameters of Neural Network Models

Classification Model	Hyperparameter
LSTM	maximum text length = 300 Embedding = 300d Hidden layer activation = ReLu dropout = 0.5 optimizer = Adam Output layer activation = Sigmoid Epoch=20
CNN+LSTM	maximum text length = 300 Embedding = 300d Hidden layer activation = ReLu dropout = 0.4 optimizer = Adam Output layer activation = Sigmoid Epoch=20
Bi-LSTM	maximum text length = 300 Embedding = 300d Hidden layer activation = ReLu dropout = 0.4 optimizer = Adam Output layer activation = Sigmoid Epoch=20
BERT	optimizer = AdamW Learning rate = 2e-5 Epoch=4

we are using the Precision-Recall Area Under Curve (PRAUC) scores. Precision and recall assess the performance of a classifier minding the minority class (He and Ma (2013)). The sample count of the binary classes in our dataset on rubric items varies based on the likelihood cut-off point (Figure: 5.2). To measure models' performance on skewed dataset precision-recall curve is recommended (Branco et al. (2016)). We used scikit-learn function to calculate the PRAUC score from the Precesion-Recall curve.

5.6 Results and Discussion

In this section we discuss our findings from analyzing the review comments and corresponding rubric criteria.

RQ1: Which model performs best for classifying quality review text?

This study considers a review to be of good quality if either a problem is mentioned, a suggestion is given, or the review comment is localized to a particular portion of the reviewed work. We have shown results comparing the F1 score of baseline ML models and the neural network models on three different datasets (problem-detection, suggestions, localization) in the Figure 5.1. In classifying review comments using the problem-detection annotation dataset, the BERT model performed best with an F1 score of 0.91. The second best F1-Score 0.89 was achieved by the Bi-LSTM and SVM models. Although Bi-LSTM and SVM show equal F1 scores for this dataset, we prefer Bi-LSTM, since the TF-IDF embedding used with the SVM model does not consider semantic and context information.

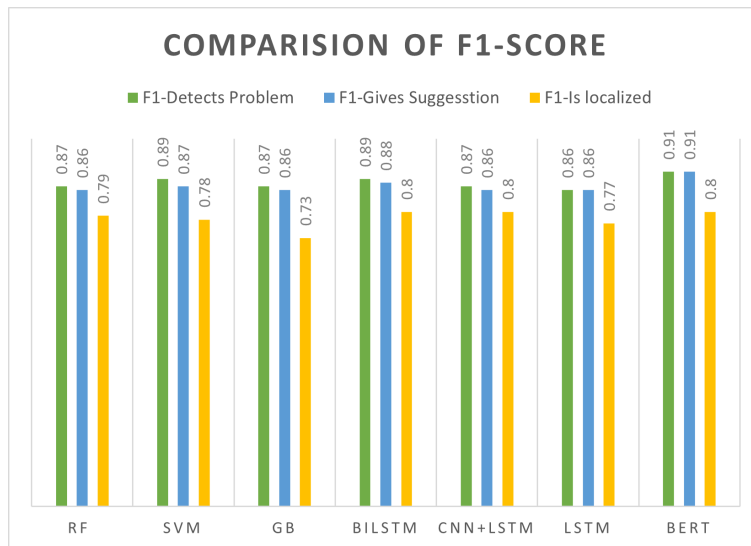


Figure 5.1: F1-score comparison to measure performance to classify review text on problem-detection, suggestion-detection and localization datasets using baseline ML models and neural network models. In overall F1-score comparison BERT model shows better performance.

- For suggestions, the BERT classification achieved a 0.91 F1 Score, and its closest competitors was Bi-LSTM with an F1 score of 0.88. The F1 score of SVM was 0.87. F1-score of RF, GB, CNN+LSTM and LSTM were 0.86.
- For identifying the localization property, the best F1-score of 0.8 came from BERT, CNN+LSTM, Bi-LSTM, and the random forest models.
- Considering the performance on the F1-score of models on the classification task based on all three datasets, the BERT model performed best.

RQ2: Can neural-network models detect rubric items that enable peer reviewers to write quality reviews?

Based on the performance of the various models on classifying review text, we chose the BERT model to annotate the 408,104 review comments collected from the Expertiza system. The trained BERT model performed a binary class annotation for each review comment to indicate presence or absence of the three features (detects problem, gives suggestion, and is localized) that we are considering for quality review.

Each rubric item has many corresponding review comments. As a result it is not straightforward to conclude that a rubric item will or will not induce a reviewer to write a quality comment. We can only determine the likelihood that a reviewer writes a quality comment in response to the rubric item. So in this section, we calculated these likelihoods for the mentioned three features for each rubric item.

After calculating the likelihoods, we need to decide on a likelihood cut-off point (Section III.D). Increasing the likelihood cut-off point "raises the bar" for a comment to be considered a "quality comment." It does this by decreasing the number of review comments that are determined to contain any of the three characteristics (problem detection, suggestion, localization). Figure 5.2 shows how the number of responses deemed to be "quality comments" declines as the cut-off value increases.

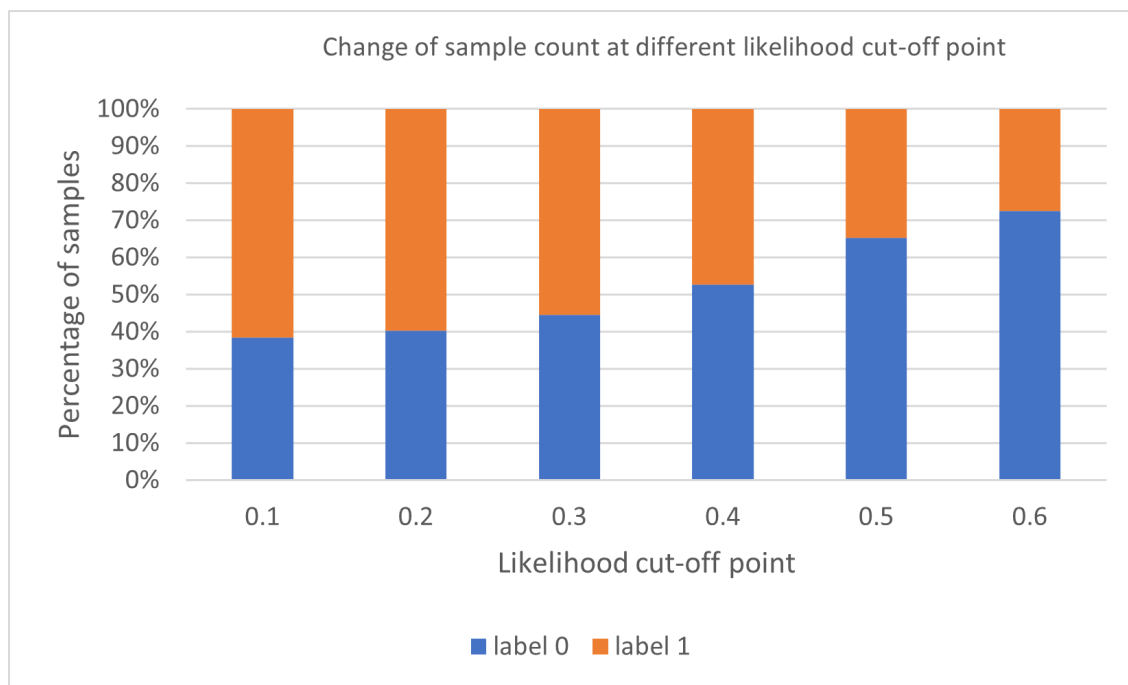


Figure 5.2: Change of "quality comments" as the likelihood cut-off point increases. A label of 1 indicates rubric items that are deemed to induce a quality review comments. A label of 0 indicates a less effective rubric item. With a cut-off of 0.1, about 60% of the comments are considered "quality comments," whereas at a cut-off of 0.6, only about 30% are.

To classify the rubric items we used BERT and Bi-LSTM models. Changing the likelihood cut-off point changes how many samples fall into each of the binary classes. We strive to make the dataset as balanced as possible without discarding any of the labeled data. This suggests that we use 0.4 as a cut-off point, since the difference between the number of 1 and 0 labels is a minimum with cut-off = 0.4 (Fig. 5.2). As the dataset becomes imbalanced, the performance of the models is prone to be better for the class that contains the larger number of samples. F1 score is a good measure of performance when the dataset is balanced. Having an imbalanced dataset we used the PRAUC Score as a performance measure.

For our results, we actually chose cut-off = 0.3 because it yields a fairly balanced dataset for the binary classes, and 0.3 gives a higher PRAUC score than 0.4.

Table 5.5 and Figure 5.3 show how the PRAUC scores of both BERT and Bi-LSTM models change as the cut-off point increases from 0.1 to 0.6. Among the two models, BERT showed the best result at all the cut-off points than Bi-LSTM. BERT achieved the best PRAUC score 0.86 at cut-off point 0.2. Both the models showed the tendency to perform poor with the more skewed dataset.

Table 5.6 shows thirteen rubric items drawn from a diverse set of academic fields, along with the number of reviews each was used in. Each cut-off point column shows the prediction

Table 5.5: Performance comparison between BERT and Bi-LSTM at different likelihood cut-off points using PRAUC scores. The "Difference between number of samples in the 2 classes" is the difference between number of comments classed as 1 (effective rubric item) and 0 (less effective rubric item). Note that BERT outperforms Bi-LSTM at all the cut-offs points.

cutoff	PRAUC BERT	PRAUC Bi-LSTM	Difference between number of samples in the 2 classes
0.1	0.82	0.78	732
0.2	0.86	0.75	618
0.3	0.83	0.70	352
0.4	0.79	0.67	170
0.5	0.63	0.54	972
0.6	0.53	0.45	1420

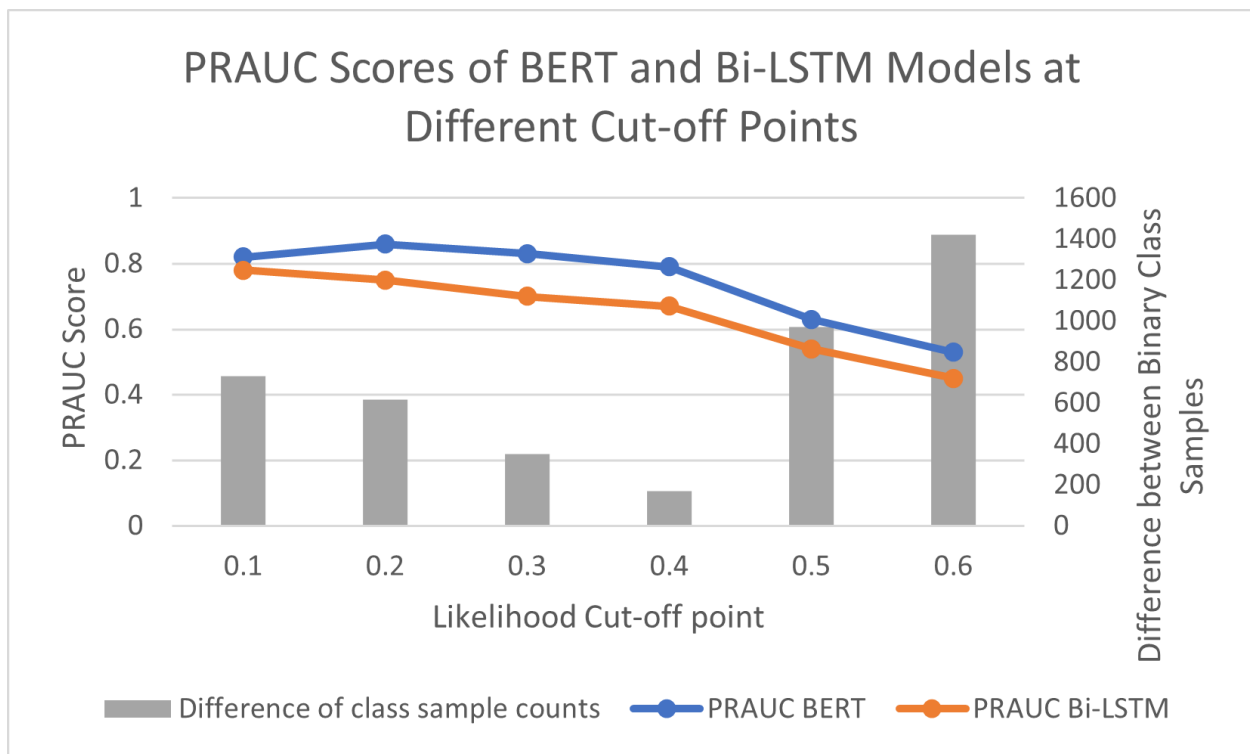


Figure 5.3: Graphical view of data from Table 5.5. The right horizontal axis "Difference between Binary class Samples" and the bars indicate the difference between number of comments classed as 1 (effective rubric item) and 0 (less effective rubric item). The two lines show the PRAUC score at different cutoff points for the BiLSTM and BERT models. Note that BERT outperforms BiLSTM at all the cut-off points

Table 5.6: Table shows predictions by the BERT model on 13 rubric items at different cut-off points. "Review count" is the number of reviews in our dataset that included the indicated criterion. A "1" in the cut-off point columns indicates the corresponding rubric item is predicted to induce peer-reviewers to write a quality review and "0" means it is not. Note that at higher cut-offs, the model classifies fewer rubric criteria as likely to induce effective comments.

Rubric Item		Review Count	Likelihood cut-off points					
			0.1	0.2	0.3	0.4	0.5	0.6
1	Please make a comment about your rating. Provide suggestions for how the author can improve their learning targets:	651	1	1	1	1	1	1
2	Are there any methods that are doing more than 1 task? One method should only handle one task, if there are multiple tasks, there should be function calls. E.g., if a customer is checking out his cart, is it one function that handles the whole flow or there are multiple? Mention examples of where this is not the case.	153	1	1	1	1	1	1
3	The author has appropriately applied knowledge of the metal ion geometry in consideration of delta. The relationship between geometry state and delta is described with elegance. If multiple relevant geometries exist, the author has clearly described each case and its effect on delta. (Please comment on assigned score)	45	1	1	1	1	1	1
4	Did the authors revise their work in accordance with your suggestions?	1030	1	1	1	1	1	0
5	Use "suggesting" mode and the commenting feature in Google Docs to make corrections and suggestions directly in your peer's essay.	109	1	1	1	1	1	0
6	Is the required functionality present in this code? Or, if this is a refactoring project, does the code represent a clear improvement over the previous code? Or, if this is a merge project, can the features be used in an intuitive way, without raising error conditions?	26	1	1	1	1	1	0
7	How well did the author address contingency leadership?	23	1	1	1	1	0	0
8	Is the UI intuitive and are functionalities easy to find?	2945	1	1	1	0	0	0
9	Does the problem appear to be original? (Please Google and try to find it.)	719	1	1	1	0	0	0
10	Does the essay maintain a sharp focus?	23	1	1	0	0	0	0
11	How often are iterations of Ember released?	8	1	0	0	0	0	0
12	This teammate was on time and attended or arranged to contribute to all team meetings.	3765	0	0	0	0	0	0
13	The admin should have more privileges than other. users. Check the functionalities for admin below and see if they all work: Can a student return a book?	3012	0	0	0	0	0	0

by the BERT model for a rubric item using that cut-off point. For example, rubric item indexed 1 in Table 5.6 was predicted as an effective rubric item for cut-off points 0.1 to 0.6. On the other hand rubric item indexed 8 was predicted as an effective one for cut-off points 0.1, 0.2, and 0.3. As we increased the cut-off point to 0.4 the BERT model predicted it as a not-an-effective ("0") rubric item.

We found that the BERT model was most effective in classifying rubric criteria as "effective" or "less effective." The tipping point ("cut-off") between effective and less effective was chosen to balance the dataset as nearly as possible between "effective" and "less effective" comments. Comments rated "effective" tended to have words that explicitly called for feedback, such as "Provide suggestions", "Please comment", and "Mention examples."

RQ3: Is there any relation between rubric length or rubric-item position and length of the comments written by students?

We did a statistical analysis of the rubric items in order to find whether there is any correlation between the amount of text entered in response to the items and (1) the number of items in the rubric, or (2) the position of the rubric item within the rubric. Our hypothesis is that an individual rubric item would tend to attract a larger volume of feedback if it is in a short rubric, or if it is near the beginning of the rubric.

We calculated the average length of the review comments (in characters) and the average length of text in each rubric itself (the sum of the lengths of the rubric items). The Pearson correlation shows that the average length of the review comments is inversely correlated (-0.26) with the number of items in the rubric. This indicates that a rubric containing more items tends to attract shorter responses. Figure 5.4 shows the 2D scatter plot with a regression line for the average length of review comments and number of rubric items.

The average rubric item length is also inversely correlated (Pearson correlation of -0.55) with the position of the rubric item. Thus, a rubric item that is near the end of the rubric tends to garner comments that are shorter. Figure 5.5 shows the 2D scatter plot of the average length of comments and the position of rubric items in rubrics. The regression is showing the negative correlation.

5.7 Conclusion

In this study we collected 3,164 rubric criteria that have been used to write 408,104 peer reviews. The comments were annotated manually at first to train machine learning models and then annotated using the models for the presence or absence of three properties (detects a problem, gives a suggestion, and is localized). We did a comparison of baseline machine-learning models

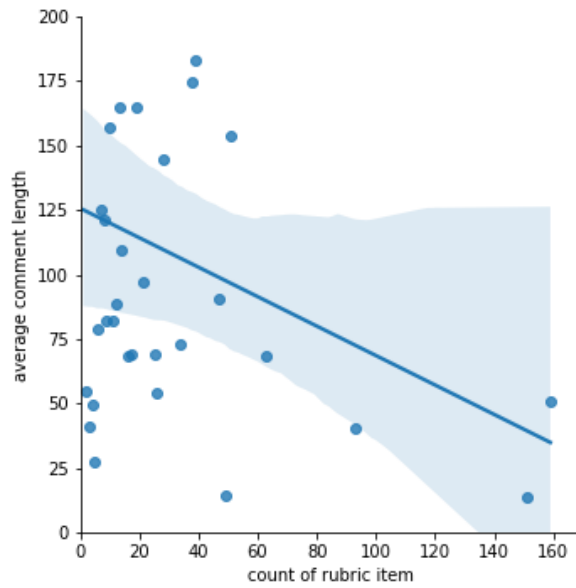


Figure 5.4: Correlation between average length of review comments (y -axis) and the number of rubric items in rubrics (x -axis).

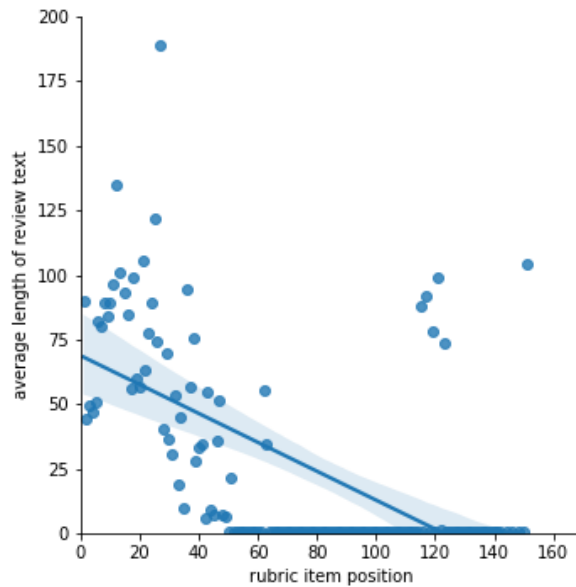


Figure 5.5: Correlation between average length of review comments (y -axis) and position of rubric items in the rubrics (x -axis).

and neural network models to select the best-performing model for annotating the review comments. Among the models, BERT performed best. A comment that was found to contain at least one of these three properties was deemed an "effective" rubric item. We compared the BERT and Bi-LSTM models to classify rubric items. Results showed that both models performed better when trained with a balanced dataset. Rubric criteria identified as "effective" by the model tended to have words that explicitly called for feedback. We measured the correlation between review length and (1) rubric-item position within the rubric and (2) length of the rubric, in terms of number of items.

We found that shortening the rubric was very effective in inducing more text to be written for each rubric item (125 words on average in the shortest rubric, vs. about 30 words in the longest rubric). Placing a rubric item early in the rubric was also associated with more textual feedback, though the impact of placement was not very great, except for very long rubrics.

The results from the study provide a way of using NLP and neural networks to predict what kind of rubric items will induce students to provide useful textual feedback to their peers. As the amount of labeled data increases, the models should become more accurate. In the future, we would like to determine what rubric length and item orderings are most effective in producing actionable feedback from peer assessors.

CHAPTER

6

AUTOMATED META-REVIEWING FOR LOCATING DISAGREEMENT IN PEER ASSESSORS' FORMATIVE FEEDBACK

6.1 Introduction

Disagreement among peer assessors' formative feedback can create confusion for the students, leading them to question the credibility of the review process and the competency of their reviewers. Peer assessment has proven to be a valuable tool in the classroom, providing students with high-quality feedback. It is common that peer assessors may have incoherent review comments while reviewing the same artifact. However, the effectiveness of formative feedback in peer assessment depends on clear communication with students, enabling them to take action on the feedback. Disjointed formative feedback baffles the students (Prins et al. (2005)). Meta-reviewing the peers' feedback to identify disagreement is labor intensive for the instructor and impacts offering timely feedback to the students. Despite disagreement among peer assessors' formative feedback being an important issue, there is a dearth of research addressing this problem. In this study, we propose an automated meta-review approach to identify disagreements in formative feedback. We collected three distinct datasets with for-

formative feedback from a software development course that utilized peer assessment. Recent natural language processing (NLP) models are trained on massive corpora of texts to generate high-quality vector representations of texts. In our research, we compared existing NLP models and fine-tuned them to produce high-quality feature vectors from feedback full of technical jargon and inconsistent English. Our results demonstrate that quality feature representations of text improve the performance of the clustering algorithm in identifying disagreements in formative feedback.

Table 6.1: Table shows four peer-reviewers’ comments on a piece of work following a rubric item. Three of the reviewers are in agreement, and one reviewer disagrees.

Rubric Item	Student	Reviewers	Review Comments
Is the UI of the application neat and logical?	Student_1	Rev_1	UI seems awesome.
		Rev_2	Yes, Nav bar is very clearly implemented.
		Rev_3	The UI is not particularly neat and logical.
		Rev_4	UI is neat and I particularly liked the navigation bar.

Peer assessment is widely used in all disciplines and at all levels of education. It is essential for massive open online courses (MOOCs), where the number of students can be overwhelming for instructors. Grading them all can be daunting for instructors and can impact their teaching performance. The peer-assessment process benefits students by providing them with timely feedback and encouraging meta-cognitive thinking. Multiple studies have shown that peer assessment can be as accurate as instructors’ assessments, as the artifact is assessed by multiple peer assessors who can devote more time than instructors (Sadler and Good (2006); Luo et al. (2014); Panadero and Alqassab (2019)). Most peer assessment processes are double-blind to minimize bias and ensure independence Gehringer (2014). However, disagreement among peer assessors’ feedback can leave the assessee confused about how to act on the feedback and questioning the review process. Identifying disagreement is easier in summative feedback but more effort-intensive in formative feedback. In Table 6.1, we show review comments on an artifact where reviewers had incoherent opinions. Meta-reviewing by the instructor will help render the review comments coherent, but it will place more of a burden on the instructor than simply assessing the artifacts, as the number of reviews on all the artifacts can be overwhelming. For example, r reviewers review s students on c items, making $r \times s \times c$ feedback comments to look into. An automated way to locate the disagreement in formative feedback would reduce the burden on the instructor and provide students with coherent feedback in a timely manner. Despite the importance of locating disagreement automatically, to the best of our knowledge, there are no studies addressing this issue in any academic peer assessment process.

In the scope of this study, We denote “*agreement and disagreement*” by “*(dis)agreement.*” By the term “*Disagreement*” in peer assessors’ feedback, we mean,

1. comments that contradict each other or
2. comments that express different opinions.

However, comments that partially concur with another reviewer’s opinion will not be counted as disagreement.

Formative feedback is written in natural language. With the current progress in the NLP field, this allows us to try several NLP approaches to locate disagreement. A few such approaches are sentiment analysis Yin et al. (2012), argument mining Moens et al. (2007); Palau and Moens (2009), and stance detection Mohammad et al. (2017). However, these methods typically identify the author’s standpoint toward a proposition by classifying texts into fewer categories (positive, negative, or neutral), ignoring the nuances of the feedback comments. Peer assessors may express the same sentiment but address different issues within an artifact. A highly effective approach for detecting (dis)agreement is to utilize advanced NLP techniques to cluster review comments based on their context and semantic similarities Hiray and Duppada (2017). Given that peer evaluators are appraising the same artifact using the same evaluation criteria, their feedback is likely to share similarities in both context and semantics. The number of groups formed from the feedback similarities provides a metric to measure the agreements and disagreements among any number of assessors.

Nevertheless, grouping texts based on context and semantic similarity is a complex task, and achieving precise clustering can be challenging. Grouping or clustering algorithms, in general, identify similar texts based on the distance between text-feature vectors in the text embedding space. The performance of a clustering algorithm depends heavily on the quality of the feature vectors Mikolov et al. (2013), i.e., similar texts’ feature vectors should be closer to each other than they are to other vectors in the text embedding space. In figure 6.1, we show the clustering workflow of review comments. It is important to note that most pre-trained NLP models are trained on common English sentences to construct feature vectors from text. However, comments written in a technical course like software engineering contain a lot of jargon, which is not common in the English language. Besides, when English is a second language to many peer reviewers, sentence structure can be nonstandard, and spelling mistakes are frequent. We found 30% feedback in our collected data contained spelling and/or grammatical mistakes.

Another challenge is that feedback can express opposite ideas to highly semantically similar sentences. Empirically, we observed that comments expressing opposite ideas (disagreement) might contain similar words and structure, or conversely, similar ideas may be expressed with completely different words. For example, in response to an assessment criterion, “If there

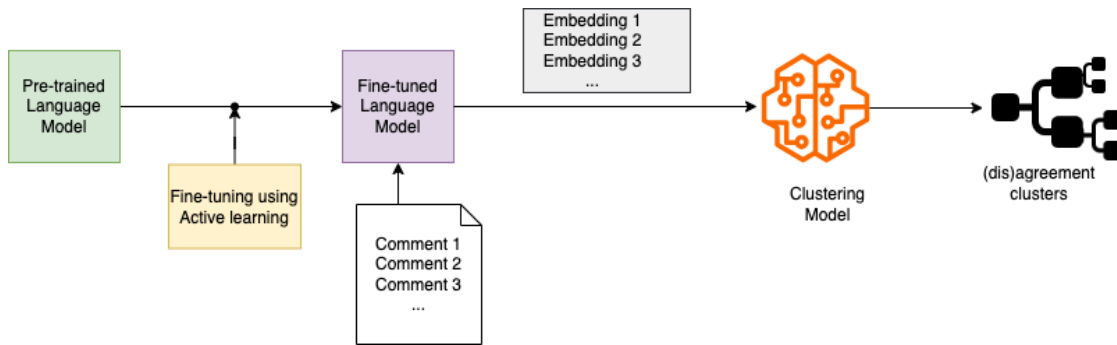


Figure 6.1: Proposed approach to quantify and locate disagreements in peer assessors' review comments.

are functions in the agent controller, are they handling one and only one functionality?" two peer reviewers' comments on the same piece of work are, "All functions are handling one to one functionality" and "They can handle multiple functionalities" These two comments are semantically very similar, while, the reviewers are in disagreement. It makes a difficult case for a state-of-the-art language model to distinguish the difference.

The aim of this study is to identify disagreements in formative feedback from peer assessment using a clustering algorithm. In our approach, the number of clusters formed from the feedback quantifies the disagreements among the peer assessors. Conversely, formative feedback in the same cluster indicates agreement among the reviewers. We hypothesize that a clustering algorithm can identify disagreements in the review comments by analyzing high-quality feature vector representations of comments' texts. In our study, a high-quality feature vector representation indicates that text embeddings of feedback that exhibit agreement are grouped closely together and far from other embeddings in the text-embedding space. This enhances the performance of a clustering algorithm to identify disagreement by making the grouping more obvious.

To test our hypothesis, we will compare various existing pre-trained text feature extraction models, selecting the most effective one as a baseline. Subsequently, we will fine-tune the baseline model using a semi-supervised approach to improve its ability to handle technical jargon and inconsistent English in the feedback text. Finally, we will produce feature vectors using the fine-tuned model and cluster them using a hierarchical clustering algorithm.

Formally, our research endeavor revolves around the following research questions:

- **RQ1:** Which existing pre-trained model is best suited to generate feature vectors that closely resemble those of context and semantically similar texts?
- **RQ2:** Can the performance of our baseline feature-extraction model be fine-tuned and

improved through semi-supervised training meant specifically for jargon-heavy feedback texts?

- **RQ3:** Can fine-tuning the feature-vector generating model enhance the clustering algorithm's ability to distinguish disagreements in the feedback?

Our study answered the research questions by using peer assessors' feedback from a software-development assignment in a real classroom. We compare baseline models by using an annotated dataset and accuracy score. We then fine-tuned a baseline model by a semi-supervised approach and using limited annotated data. We provided the visualization of the resulting improvement of feature vectors in the feature embedding space. Finally, we measured the clustering model's performance by comparing both intra-cluster and inter-cluster distance.

Overall, our findings suggest that utilizing automated meta-reviews in peer assessment can be an effective solution for identifying disagreement in peer assessors' formative feedback, particularly in technical classes with diverse linguistic backgrounds.

6.2 Background

Misaligned or conflicting feedback from peer assessors can confuse students and complicate grading for instructors. Identifying and resolving these discrepancies before providing feedback adds more work for instructors. Identifying agreement or disagreement ((dis)agreement) in formative feedback is particularly challenging. While there is a lack of literature specifically addressing this issue in peer assessment, some studies have tackled similar challenges in different domains.

Rosenthal and McKeown (2015) identified (dis)agreement in conversation using supervised learning, demonstrating the impact of classification with different feature extraction methods based on meta-thread of conversation, lexical features, sentiment, sentence similarity, and accommodation. They found that meta-thread and accommodation, captured in the semantics of the text, were most effective in identifying (dis)agreement. Galley et al. (2004) implemented a statistical approach to identify agreement and disagreement in conversation, arguing that adding contextual information improves classification accuracy. They used adjacent words and maximum entropy ranking based on lexical and structural features. Hillard et al. (2003) used supervised (decision tree) and unsupervised (clustering) classifiers to identify (dis)agreement in automatically transcribed meetings, employing hand-labeled spurts for feature extraction and clustering algorithms on the ICSI Janin et al. (2003) corpus. Hiray and Duppada (2017) argued that Siamese-inspired architecture better encodes online discussions for identifying

(dis)agreement compared to hand-crafted feature encoding, classifying discussions into three categories (agreement, disagreement, or none).

Our work differs in three major ways: First, our definition of (dis)agreement considers different issues found by two peer assessors as disagreement. Second, previous studies focus on online discussions or meeting transcriptions, whereas we analyze independent peer reviewers' feedback comments from a technical course. Third, previous studies used open-sourced annotated data and compared results with accuracy scores.

6.3 Approach: Automated Detection of Peer Feedback Discrepancies

6.3.1 Notation and Problem Statement

Assume D is a dataset that contains a set of reviews $R = \{r_1, \dots, r_K\}$ for a set of artifacts $A = \{a_1, \dots, a_N\}$ on criteria $C = \{c_1, \dots, c_M\}$. Let's assume that d_{nmk} represents the review comment for an artifact n for criterion m by reviewer k , and d_{nm} represents the set of review comments for that artifact n in response to criterion m . Note that d_{nm} is sparse where $|d_{nm}| < |R|$ as few reviewers are assigned to review an artifact.

$G_{nm} = \{g_1, \dots, g_L\}$ represents a collection of disjoint sets of review comments where any g_l contains a group of review comments that are in agreement with only reviews that are inside the group. The disjoint set of review comments in G_{nm} are for assignment a_n on criterion c_m and $(|G_{nm}| \leq |d_{nm}|)$

We define σ as a metric to measure disagreement. σ_{nm} indicates the percentage of disagreement among the reviewers in d_{nm} .

Based on the notations, we can formally define the problem as follows.

Formal problem statement: we identify $G = \{G_1, \dots, G_i\}$ we provide $\Sigma = \{\sigma_1, \dots, \sigma_i\}$ (percentage of disagreement) for A on each criterion in C .

6.3.2 Proposed Approach

We define our proposed approach to identifying discrepancies in the review comments as a grouping problem where the coherent review comments will be in one group and separated from incoherent comments. We opt to implement a clustering algorithm for the grouping task. We decided on this method because (1) clustering algorithms are known for grouping tasks based on similarity, (2) we hypothesize that review comments that are in agreement are semantically and contextually more similar than the comments in disagreement, (3) reviewers'

comments are independent of each other. We do not know the number of groups to form from reviews that are in (dis)agreement. It is not essential for a clustering algorithm to define the number of clusters beforehand. (4) We can measure the performance of a clustering algorithm without an annotated dataset. We implement our approach in the following steps:

Extracting text embedding First, we convert each review text r_c as a fixed-length vector v_c of numbers. These vectors are also called feature vectors and are used as direct input to clustering models. During generating text feature embedding, it considers the context-and-semantic-based similarity of the words in the dataset.

Generating Clusters Second, we provide the feature vectors of all the review comments on an artifact a_n on criterion c_m , which we denoted as d_{nm} to the clustering algorithm. The algorithm generates G_{nm} , where the grouping is based on the (dis)agreements among the comments.

Generating disagreement score

$$\sigma_{nm} = \begin{cases} \frac{|G_{nm}|}{|d_{nm}|} \times 100 \\ 0 \end{cases} \quad \text{for } |G_{nm}| = 1 \quad (6.1)$$

Finally, from the cluster count in the G_{nm} , the disagreement score (σ_{nm}) is calculated using equation 6.1. When $|G_{nm}| = 1$ we conclude there is no contradicting review (disagreement) in d_{nm} . Upon calculating all the disagreement scores for dataset D we return Σ .

6.4 Methods

6.4.1 Data collection and preparation

The study collected data from an assignment in a software development course at North Carolina State University over the course of three semesters. The data was obtained using a peer-assessment system called Expertiza that was funded by NSF. The peer-assessment process at Expertiza is double-blind, meaning that reviewers and reviewees are not aware of each other's identities. Prior to beginning the review process, reviewers are shown examples of good-quality review comments and are encouraged to write a review that explains any issues with the work or why they liked or disliked it. Reviewers were provided with an assessment guideline called a rubric, which consists of several items, as a grading criterion. More than one reviewer was assigned to review each assignment, and each reviewer wrote their response independently

based on the rubric criteria. An identity anonymizer was used to collect data from the X system to protect the anonymity of the students. Table 6.1 shows a sample of the data collected from the Expertiza system.

6.4.2 Text Embedding

Text embedding is a way of representing texts as vectors of numbers. These vectors are also called feature vectors and can be used as direct input to various machine-learning models. Traditionally, for text representation, statistical methods like *TF-IDF* or *one-hot vector* (presence or absence of a word in a text) were prevalent. However, these methods produce very sparse vectors and ignore the context of the text and semantic relations between the words. *GloVE* and *Transformer Attention-based* models produce fixed-length feature vectors for texts while considering the context and semantic relationship of words. In this study, we compared three pre-trained feature vector-generating models as the baseline.

- **Global Vectors (GloVE):** *GloVe* is a word embedding method that represents words in texts to a fixed-length vector. This method does not rely only on local statistics (local contextual meaning) but also includes global statistics (co-occurring words) to produce vector representations. One issue with this method is that the words always have the same vector representation, even though a word may express different meanings in different contexts. For this study, we used *GloVE.6B*, which is trained on *Wikipedia* and *Gigaword5*.
- **Bidirectional Encoder Representations from Transformers (BERT):** *BERT* is a Transformer Attention architecture-based model that has shown great success in sentence classification and many other NLP tasks. It considers the context and semantic similarity of the words in a corpus Devlin et al. (2018). BERT is a large language model, and the training of BERT is hardware intensive. The pre-trained BERT model is trained using a combination of masked language modeling objective and next-sentence prediction on the *Toronto Book Corpus* and *Wikipedia*. It is fine-tunable on different datasets to meet specific requirements. We can get sentence embeddings from BERT either by using a *mean-pooling* method that averages the feature vectors of each word Devlin (2018) or by the *[CLS]* token available at the first position of the BERT sentence embedding output Devlin et al. (2018). The *[CLS]* token is output from training with the next sentence prediction task. To use the BERT for the text-similarity calculation task, we used BERT with a cross-encoder architecture (Figure: 6.2). We passed two texts to BERT, and added an extra classification head at the last layer of BERT, and calculated the cosine similarity score.

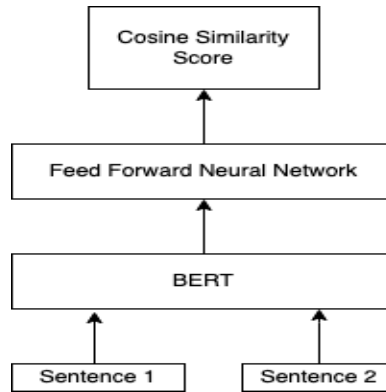


Figure 6.2: BERT Cross-encoder Architecture for Sentence Similarity

- **Sentence-BERT (SBERT)** Reimers and Gurevych Reimers and Gurevych (2019) proposed *SBERT*, a modification of the pre-trained BERT network. This model produces text embeddings in a way that the semantically similar sentences have a very high cosine similarity. Sentence-BERT (SBERT) is placed on top of the *BERT* model with an added *Siamese Neural Network* architecture. The Siamese architecture consists of two identical subnetworks that have the same parameters. During the parameter updating phase, the update is mirrored in both sub-networks. The feature vectors from this model have a very high cosine similarity for semantically similar sentences. SBERT consists of two phases:

1. In the training phase of SBERT, two sentences are given to identical BERT models for generating sentence embeddings. As the BERT model provides feature vectors for each word in the sentence, a pooling (mean pooling) layer is used to convert it to sentence embedding. As shown in Figure 6.3 training phase, for *Sentence A*, we get *embeddings* u , and *embeddings* v for *Sentence B*. Then the two sentence embeddings u and v are concatenated with the element-wise difference $|u - v|$ and multiplied by a trainable weight matrix W_t for the objective function $\text{softmax}(W_t(u, v, |u - v|))$.
2. At the inference stage (Figure 6.3 Inference Phase), the *cosine similarity score* $[-1, \dots, 1]$ of the sentence embeddings (u and v) are calculated for both the input sentences. The higher the *cosine similarity*, the more the sentences are semantically similar.

6.4.3 Active Learning

We used active learning for fine-tuning the pre-trained model on our dataset to enable the model to capture the context and semantic relation of words in the formative feedback and express it in the feature vectors. The underlying principle of active learning is that if a machine-

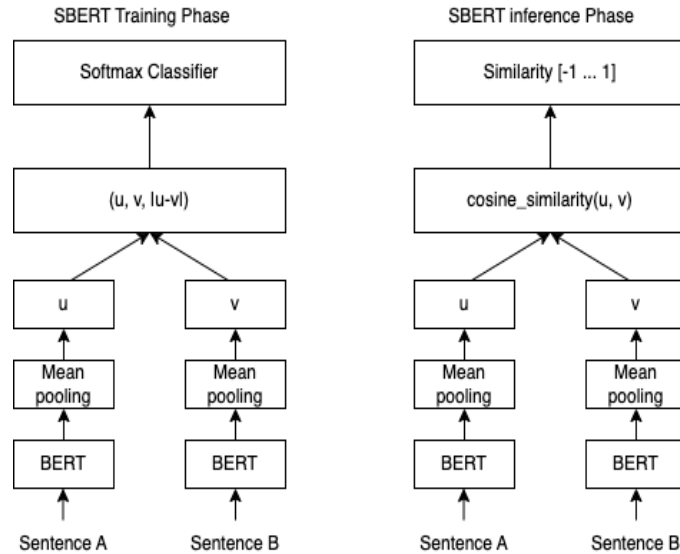


Figure 6.3: SBERT Training and Inference Phase architecture

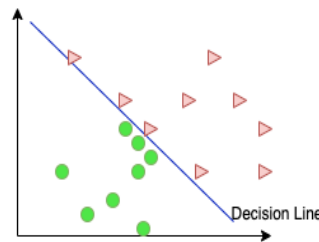


Figure 6.4: The most uncertain samples lie near the decision line

learning model can choose the data from which the model learns most, it can learn faster and with fewer data Settles (2012). To explain it further with an example, in Figure 6.4, if a model draws a line to separate two classes, the predictions that are close to the decision boundary are the more uncertain ones than the ones that are far. Active learning is an iterative process whereby an oracle annotates only the samples that the model is most uncertain about. Subsequently, the model is trained again with these annotated samples. Over iterations, this approach can significantly enhance the model's performance with less annotated data required.

6.4.4 Clustering Algorithm

For our study, we opted to use the *Agglomeration hierarchical clustering* algorithm for the grouping task. One advantage of this algorithm is that it doesn't require us to decide the number of clusters beforehand. This is particularly useful given that the feedback we receive from our peer reviewers is independent, and their opinions may differ. To begin the clustering, each

text (embedding vectors) is treated as its own cluster, and pairs of clusters are merged until all clusters merge into a single cluster and form an agglomerative tree. The agglomerative clustering algorithm follows the following step for clustering:

1. At first, make each data-point a cluster on its own.
2. Join the two closest clusters together to create a single cluster.
3. Continue with step 2 repeatedly until all data-points are under a single cluster.

This clustering algorithm produces clusters and their *cophenetic distance*, which is the height of a dendrogram where clusters merge.

6.5 Experiment

6.5.1 Dataset Preparation

For this study, we prepared three separate datasets from the collected data from the X system. In these datasets, review comments are paired or grouped together based on the artifact and rubric criterion. Each feedback is also exclusive to its corresponding datasets. The three datasets are as follows:

1. **Feature-vector-test dataset:** The performance of text clustering or classification models relies heavily on the vector representation of the texts Hiray and Duppada (2017). The *Feature-vector-test* dataset has a twofold use. First, we use this dataset to compare baseline models that generate quality feature vectors, considering the context and semantic similarity of the review comments. Second, after selecting the best-performing baseline model, we use this dataset to measure the performance of the model after fine-tuning. We carefully created this dataset with 3000 unique pairs of formative feedback. The comment pair is annotated as “1” if the two review comments express a similar idea (agreement) and “0” if they express a different idea (disagreement). This dataset was annotated by five students who completed this assignment and were familiar with the grading criteria. TAs spot-checked the annotation done by each student. We calculated the Krippendorff’s α Krippendorff (2018) to measure inter-rater reliability. In order to ensure the reliability of the dataset, we only included labels agreed upon by all taggers, which raised the α value from .69 to 1. Table 6.2 shows an example of the dataset.
2. **Fine-tuning dataset:** This dataset consists of 11,000 review comment pairs. Initially, this dataset contains no annotation for the review comments pair being in agreement or

Table 6.2: Table shows a sample dataset with paired comments, labeled for agreement (Label “1”) and disagreement (Label “0”) in two comments. Each pair of comments relates to the same piece of work following the same rubric item

	Comment1	Comment2	Label
1	Three attributes, DoB, Major and Phone number, are absent.	Major, date of birth missing from student	1
2	Any required attributes can be null in property class.	There is validation check for all necessary attributes	0
3	New property creation throws some application error, cannot test.	Could not apply to a property, showing a crashing application.	1
4	Some of fields can be null	Non-nulls have been enforced in both admin as well as instructor views	0

disagreement. In the fine-tuning phase of the model to generate quality feature vectors, we annotated a total of 1,600 pairs in four iterations.

3. **Clustering-test dataset:** The previous two datasets are used for testing and fine-tuning the text embedding quality of NLP models. Finally, the clustering-test dataset is used to compare the clustering performance of the algorithm using the feature embeddings of the review texts generated by the baseline and fine-tuned model. This dataset consists of 1,500 formative feedback comments where feedback comments are grouped together if they are on the same piece of work following the same rubric criterion. It is important to notice that this dataset is not annotated. Table 6.1 is an example of the dataset.

6.5.2 Algorithm to choose the cosine similarity cut-off

Our dataset, known as the Feature-vector-test, contains labeled pairs of feedback that either agree or disagree. Our goal is to evaluate the performance of text-embedding models on this dataset by measuring their accuracy score. Essentially, we expect these models to produce embedding vectors in such a way that the feature vectors for feedback pairs in agreement lie close to each other (due to semantic and contextual similarity) in the text embedding space, and vectors for dissimilar sentences will be far apart. Once the model produces these feature vectors, we calculate the cosine similarity for each pair of feature vectors. We then establish a similarity score that identifies feedback pairs below this score as disagreement. This score may vary for each model, so we must choose a similarity cut-off score for each one. In our study, we followed the method introduced by the writer of the SBERT paper (Devlin et al. (2018)), which selects the cut-off point in a greedy way. This algorithm picks the threshold that makes the most accurate prediction for classifying both similar and dissimilar sentence pairs in the test dataset. We have provided the pseudo-code for this algorithm (Algorithm 1).

Algorithm 1 Algorithm for cosine similarity cut-off

Input : $S \leftarrow$ cosine-similarity scores $L \leftarrow$ true labels $H \leftarrow$ 0 or 1

▷ Does high score means more similar

Output : ma

▷ maximum accuracy

 bt

▷ best threshold

function FIND_BEST_ACC_THRESHOLD(S, L, H) $rows \leftarrow$ [score, label]

▷ All score and label in S and L

if H **then** $rows \leftarrow$ Sort($rows, order=descending$)**else if** $rows \leftarrow$ Sort($rows, order=ascending$) **then****end if** $ma \leftarrow 0$ $bt \leftarrow -1$ $positive_so_far \leftarrow 0$ $remaining_negative \leftarrow$ Count of label 0 in L **for each row in rows do** $Score \leftarrow row[score]$ $Next_row_Score \leftarrow row[score + 1]$

▷ score in the next row

 $Label \leftarrow row[label]$ **if** label is 1 **then** $positive_so_far += 1$ **else** $remaining_negative -= 1$ **end if** $acc \leftarrow (positive_so_far + remaining_negatives) / length(L)$ **if** $acc > ma$ **then** $ma \leftarrow acc$ $bt \leftarrow (Score + Next_row_Score) / 2$ **end if****end for**return ma, bt **end function**

6.5.3 Selecting the text feature extraction model

The effectiveness of the clustering algorithm is dependent on the quality of the feature embeddings of the input texts. To improve the performance of our clustering algorithm, we require a text-embedding method that produces feature vectors of equal length for review comments and places context- and semantically similar sentences closer to each other in the feature vector space.

We conducted a performance comparison of the pre-trained GloVe and BERT models with SBERT (without fine-tuning) using our feature-vector-test dataset. We generated the feature vectors of each feedback in the feedback pairs of the dataset using a model. The cosine similarity of the feature vectors for the feedback pairs was then calculated. After getting the cosine similarity value for all the text pairs in the dataset, we established a cut-off score for cosine similarity using the algorithm mentioned in Section 6.5.2 and labeled pairs with a similarity score below the cut-off point as “0” and “1” otherwise. Subsequently, we compare the accuracy of the models with the annotations of the dataset. sec

6.5.4 Fine-tuning the model

We selected the pre-trained baseline model for fine-tuning, which achieved the highest accuracy score on the feature-vector-test dataset. To fine-tune the model, we employed the active learning approach, with the help of an expert (human), to annotate uncertain instances in an iterative manner. Initially, we used a baseline model for inference on the Fine-tuning dataset, which provided a cosine similarity value for each pair of review comments. After each training iteration, we determined the decision boundary (cosine similarity cut-off score) using the algorithm outlined in *section 6.5.2* on the annotated feature-vector-test dataset. Next, we calculated the distance between the cut-off similarity score and inferred cosine score for each pair of review comments in the fine-tuning dataset. Subsequently, we sorted the distance values in ascending order to identify the samples the model was most uncertain of, which indicated how close the inferred cosine-similarity score was to the cut-off score. From this list, we selected the first or most “uncertain” 400 pairs of comments for the expert’s annotation and saved the current model for the next iteration. In the subsequent rounds, we trained the saved model with the annotated instances from the previous round and inferred the cosine distance on the remaining fine-tuning dataset. We repeated this process for four rounds. Figure 6.5 illustrates the iterative process of fine-tuning the SBERT model with active learning.

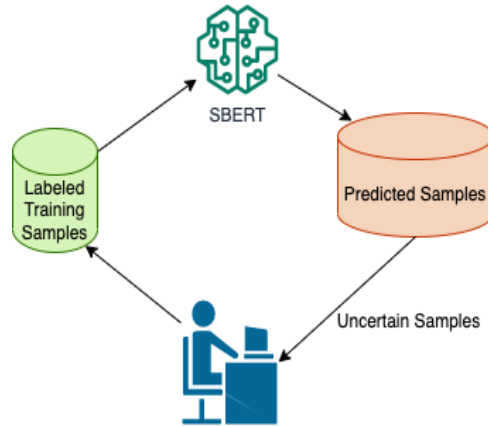


Figure 6.5: Fine-tuning the SBERT model using active learning with an expert in the loop.

6.5.5 Evaluation Metrics

In order to evaluate the text embedding generator NLP models on the feature-vector-test dataset, we utilized an accuracy score for comparison. Additionally, we employed the silhouette score to measure the clustering performance. This score offers a quantitative means of assessing the appropriateness of the clustering. The score ranges from -1 to 1 , with 1 indicating well-separated and distinct clusters, 0 signifying insignificant differences in distance between clusters, and -1 indicating misassigned clusters.

$$(b - a) / \max(a, b) \quad (6.2)$$

Equation 6.2 is used to determine the Silhouette Coefficient where we take into account both the average distance within each cluster (a) and the average distance to the nearest neighboring cluster (b) for every data-point.

6.6 Results and discussion

6.6.1 RQ1

RQ1: Which existing pre-trained model is best suited to generate feature vectors that closely resemble those of context and semantically similar texts?

We compared the performance of feature vector extraction models on the annotated feature-vector-test dataset using the accuracy score.

- Based on our cosine-similarity score comparison from the models on the annotated

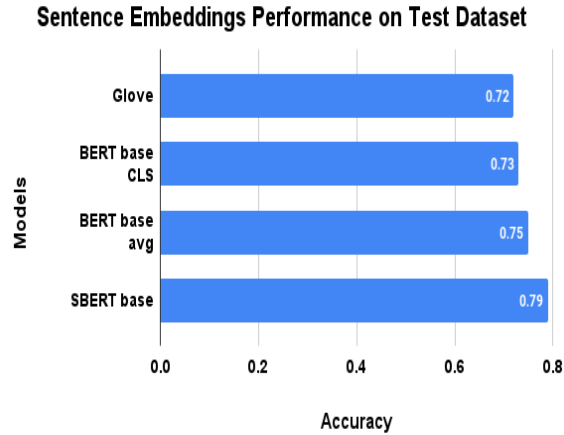


Figure 6.6: Comparison of Sentence Embedding Approaches using Accuracy Score on the Test Dataset

feature-vector-test dataset, we found that the GloVe model accurately identified 72% of the feedback pairs as agreeing or disagreeing. The BERT model, achieved an accuracy score of 73% and 75% using the feature vectors extracted from the [CLS] token and mean-pooling, respectively. The base SBERT model outperformed the other models and achieved a 79% accuracy. (Figure 6.6)

- Based on the analysis of the accuracy scores, it seems that the base SBERT feature extraction is the best choice in terms of providing feature vectors with high cosine similarity when sentences are in agreement and low cosine similarity when there is disagreement.

Though GloVe is known for considering the global statistics of the words in the corpus to generate text embeddings, it produces vectors in a one-to-one relation with the words, which makes it a static model for feature vector generation. On the other hand, BERT's average pooling over the word tokens method is often inefficient for using text embedding and sometimes worse than GloVe embeddings as it loses valuable information in the pooling phase Reimers and Gurevych (2019). Although BERT produces high-quality performance in various NLP tasks, it works poorly in finding the semantic similarity between sentences in the case of using feature vectors directly Li et al. (2020). The Siamese network architecture in SBERT takes two input sentences and encodes them into sentence vectors in the same embedding space. If the two input sentences are similar in context and semantics, they are close to each other in the embedding space, which makes it ideal for producing text embeddings Hiray and Duppada (2017).

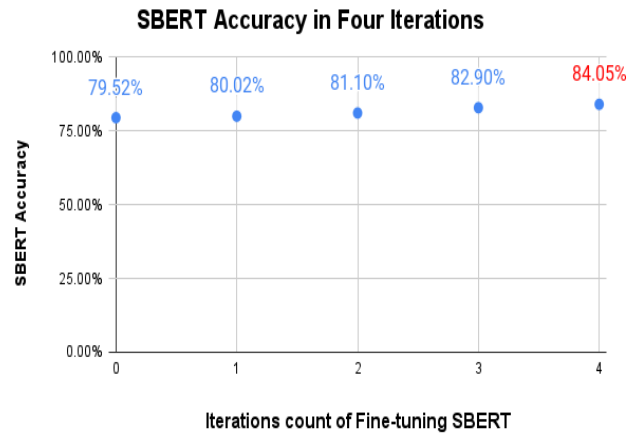


Figure 6.7: Fine-tuning of SBERT increased accuracy for identifying sentence similarity or difference after each iteration

6.6.2 RQ2

RQ2: Can the performance of our baseline feature-extraction model be fine-tuned and improved through semi-supervised training meant specifically for jargon-heavy feedback texts?

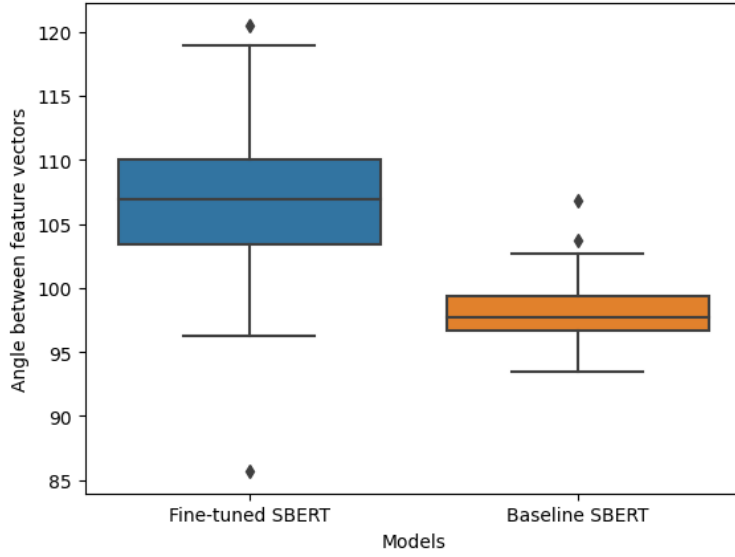
After comparing the accuracy scores of various models mentioned in RQ1, we decided to use the baseline SBERT model for further fine-tuning using the active learning approach and the Fine-tuning dataset. During the fine-tuning phase, we tested the model's feature vector quality on the feature-vector test dataset after every iteration, and we observed that the fine-tuning improved the SBERT model's accuracy in identifying feedback-pair as agreement or disagreement (Figure 6.7).

Moving on to the semi-supervised training phase using the active-learning approach, we noticed that in each iteration, the model learned from the samples that were close to the decision boundary. This approach helped us to fine-tune the baseline model on a dataset consisting of feedback text with jargon from a technical class. Overall, the results were quite promising, and we are confident that our approach can be applied to feedback text in other domains as well.

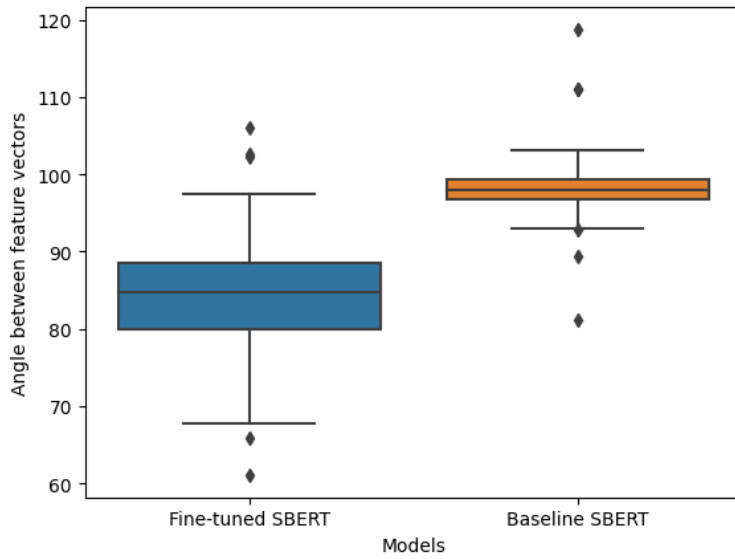
6.6.3 RQ3

RQ3: Can fine-tuning the feature-vector generating model enhance the clustering algorithm's ability to distinguish disagreements in the feedback?

In this study, we hypothesized that fine-tuning a model for domain-specific feedback texts



(a) Feedback text pairs labeled for having disagreement



(b) Feedback text pairs labeled for having agreement

Figure 6.8: Overall angle comparison of feedback texts pairs in agreement or disagreement.

Table 6.3: This table shows the angle between the two-dimensional feature vectors of the review comments pair in the feature vector space. The angle between the vectors for comments-pair with disagreement is higher for fine-tuned SBERT than baseline SBERT

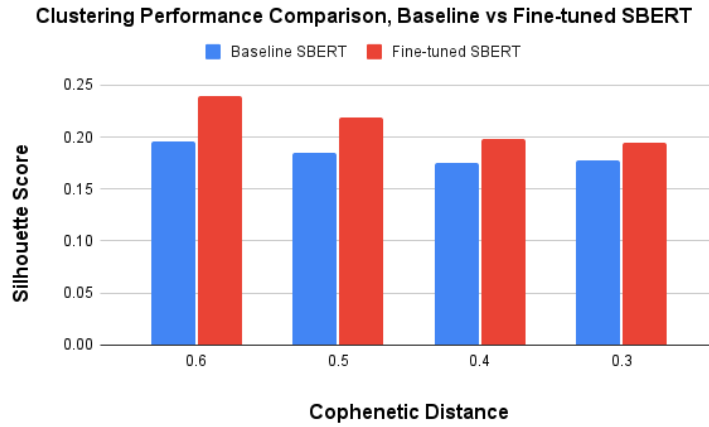
Comment 1	Comment 2	Label	Angle between vectors of baseline SBERT	Angle between vectors of fine-tuned SBERT	Angle Difference
Edit profile for agent is missing.	No issues over how agent is working	0	99.73	118.64	18.91
The UI needs to be improved. The required format is not given.	UI is okay.	0	102.68	120.5	18.18
A student can be put in waitlist and enrolled at the same time by instructor.	Yes. The instructor class has the required attributes	0	96.69	112.93	16.24

produces quality feature vectors. Here, quality feature vectors indicate that vectors of feedback that are in agreement will be close to each other in the embedding space as they are more similar in semantics and context. Conversely, review comments that are in disagreement will be separated in the embedding space. With this fine-tuning, the clustering algorithm will be able to group texts more effectively.

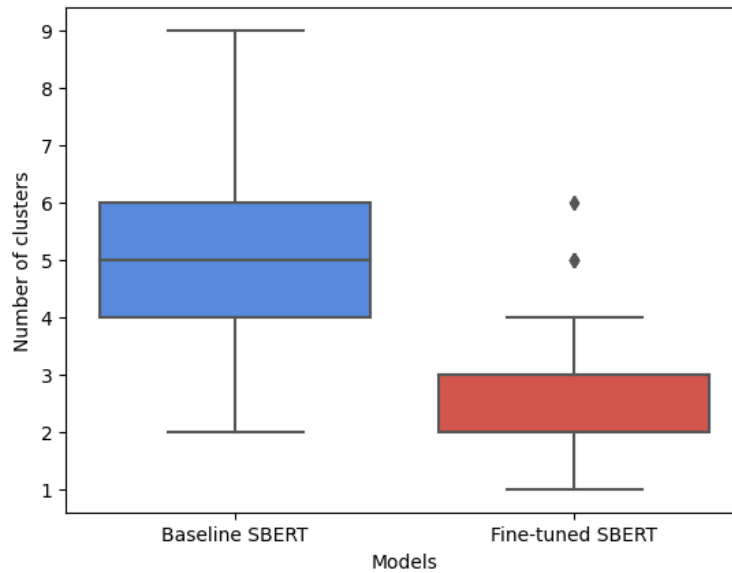
In order to test our hypothesis about the closeness of feature vectors in the embedding space, we calculated and compared the angle between the feature vectors for each feedback pair in our feature-vector-test dataset.

- **Angle Calculation:** We generated feature vectors from a review text using a model and then used principal component analysis (PCA) to reduce the vector dimension to 2-dimension. Next, we projected the vectors of a feedback pair from the origin of the feature-embedding space and calculated the angle between the vectors.
- **Angle Comparison:** We compared the angle between each feedback pair vector produced by the baseline SBERT and the fine-tuned SBERT model. Tables 6.3 and 6.4 display a few samples of comment pairs and their angle comparison. In Figure 6.8, we presented an overall impact of the fine-tuning process on the baseline SBERT model. We observed that the overall angle between feedback text pairs labeled as agreement (“1”) had a smaller angle for fine-tuned SBERT compared to baseline SBERT, while sentence pairs labeled as disagreement (“0”) had a larger angle for fine-tuned SBERT. This indicates feature vectors using the fine-tuned model for the comments pair that are agreeing will be closer to each other in the embedding space, and disagreeing pairs’ vectors will be far from each other.

To test clustering performance, we used the clustering-test dataset. Since the dataset is not labeled, we do not know the number of clusters for each artifact. Before we implemented the clustering algorithm to separate (dis)agreements, we used the baseline and fine-tuned SBERT model to generate text embedding for each review comment. To measure the quality



(a) The clustering performance comparison using Silhouette Score with different thresholds for Agglomerative Hierarchical Clustering shows that Fine-tunes SBERT using Active Learning outperformed at every Cophenetic distance.



(b) Comparison of overall cluster count for each artifact using text embeddings produced by baseline and fine-tuned SBERT. The Agglomerative algorithm produced less number of clusters for every artifact using the sentence embedding from Fine-tuned SBERT.

Figure 6.9: Agglomerative clustering algorithm's performance using feature vectors from Baseline-SBERT and Fine-tuned SBERT

Table 6.4: Table shows the angle between the two-dimensional feature vectors of the review comments pair in the feature vector space. The angle between the vectors for comments-pair with an agreement is lower for fine-tuned SBERT than baseline SBERT

Comment 1	Comment 2	Label	Angle between vectors of baseline SBERT	Angle between vectors of fine-tuned SBERT	Angle Difference
check in and out are missing	check in and check out for applicant is missing for the admin pages	1	99.52	61.10	-38.42
Cannot delete leasing agents	Delete not working	1	99.43	76.15	-23.28
Leasing Agent Name and phone number can be null	The name and phone number can be null. Email has no email validation.	1	96.80	76.62	-20.18

of the clusters formed by the algorithm using the text embeddings, we used the silhouette score. In Figure ??, we presented the Silhouette score for different cophenetic distances. Our observations indicate that the fine-tuned SBERT model had higher Silhouette scores for every cophenetic distance compared to the baseline SBERT model. The silhouette score is calculated based on the intra-cluster and inter-cluster distance, and the silhouette score comparison concludes that the fine-tuned SBERT produces better-quality feature vectors that assist in improving the clustering algorithm’s performance. In Figure 6.9, we showed a comparison of the overall cluster count for each artifact using text embeddings produced by baseline and fine-tuned SBERT. When peer reviewers assess the same artifact using the same assessing criteria, they are expected to provide similar feedback on the same piece of work. We observed when the clustering algorithm used fine-tuned SBERT sentence embeddings; it merged most of the clusters that were separated in the baseline SBERT embeddings model.

To have some more insights in Table 6.5, we are presenting a few examples of clusters formed by the clustering algorithm before and after fine-tuning the SBERT model.

6.7 Conclusion and Implications

In this study, we aimed to measure disagreements in peer-assessors’ formative feedback by implementing a clustering algorithm. Our hypothesis is that formative feedback comments that are in coherence are more contextually and semantically similar than the feedback comments that are incoherent. Capturing this notion of feedback texts into feature vectors will enable a clustering algorithm to separate the disagreement from peer assessors’ formative feedback. To test our hypothesis, we first translated the reviewers’ natural language into feature vectors. After testing various NLP models, we found that the pre-trained SBERT text embedding was the most effective. Subsequently, we fine-tuned the SBERT model by implementing uncertainty sampling and expert annotation. Our study showed that the fine-tuned SBERT text-embedding

Artifact	Assessment criterion	Disagreement (σ_1)	Clusters using Baseline SBERT Embedding	Comments	Clusters using Fine-tuned SBERT Embedding	Disagreement (σ_2)
A1	Are variable names indicative of what the variables are storing/handling? Mention examples of where this is not the case.	85.71%	Cluster 1	Yes, variable names indicative of what the variables are storing/handling. I do not see any variable names that do not indicate what they are storing so thats good.	Cluster 1	28.57%
			Cluster 2	Yes, variable nomenclature is overall good.		
			Cluster 3	Yes, all used variables define their usecase extremely well. made_reservations is a good example. snake_case followed		
			Cluster 4	Proper variable names observed.		
			Cluster 5	All of the variable naming conventions are easily understandable and conform to good ruby style.		
			Cluster 6	It seems the repo is private. When I clicked on the repo link got '404 this is not webpage you looking for'. Couldn't check the code and code related points	Cluster 2	
A2	Are the users allowed to delete a flight reservation?	83.33%	Cluster 1	Yes, a user can delete a flight reservation. User can delete reservations.	Cluster 1	33.33%
			Cluster 2	Yes working		
			Cluster 3	yes		
			Cluster 4	Yes, the user is able to delete any flight reservation made earlier.		
			Cluster 5	I know it's a little thing, but I like that you made the button say 'CancelReservation' rather than sticking with the default 'Destroy'.	Cluster 2	
A3	Is the user able to edit his/her own profile?	75%	Cluster 1	Yes, the user is able to edit his/her own profile.	Cluster 1	0%
			Cluster 2	Yes, the user is able to edit his/her profile. Works great.		
			Cluster 3	Yes user is able to edit all details entered earlier		

Table 6.5: Table shows the clustering outcomes on review comments using feature extraction by baseline SBERT and Fine-tuned SBERT. In the "Comments" column, each row presents comments of different reviewers on an artifact in response to an assessment criterion. Disagreement measure σ_2 (used Fine-tuned SBERT embedding) is more accurate than σ_1 (used baseline SBERT embedding)

model outperformed the baseline SBERT model on our test dataset. We observed that in the text embedding space, the fine-tuned SBERT model's feature vectors are closer for agreeing feedback and farther for disagreeing feedback. Finally, we used the base-case model and the fine-tuned model's text embedding with the agglomerative clustering algorithm. We experimented with different cophenetic distances and compared our results using silhouette scores. The silhouette score and empirical study of the clusters formed by the fine-tuned model's text embedding show that using clustering algorithm we can quantify disagreements in peer reviewers' formative feedback.

The key findings of this study are that the base SBERT model outperforms other feature-extraction methods like Glove and BERT on the task of finding semantic similarities among texts. Also, we show that fine-tuning SBERT on a peer-review dataset containing a high amount of technical jargon and inconsistent English further improves the model's accuracy. We also show that fine-tuning the SBERT model to generate quality text embeddings improves the clustering done on the peer-review data to find disagreement in the review comments.

Since disagreement among reviewers can make it difficult for the instructor to rely on peer assessments when evaluating the work, pinpointing these disagreements can direct attention to the reviews that the instructor needs to reassess before assigning a final score to the work. The current approach could be enhanced by integrating explainable methods (Khosravi et al. (2022)) that allow both instructors and students to clearly understand the rationale behind any flagged disagreements.

Implications. Deploying algorithms for uncovering disagreement in peer feedback bears various implications across different dimensions, interfacing technological, educational, ethical, and psychological spheres. *Educationally*, this approach could elevate the quality and consistency of feedback and help teachers optimally use their time in overseeing the peer review process. *Technologically*, the incorporation of such an algorithm necessitates robust structures to ensure data privacy, security, and ethical use, safeguarding all stakeholders in the educational process. *Ethically*, attention must be devoted to ensuring that the algorithm operates transparently and without bias, providing equitable support across diverse student populations and respecting the integrity of educational interactions. On a *psychological* plane, while the minimization of conflicting feedback can foster a clearer, more constructive learning environment, the role and perceived authority of algorithmic interventions in feedback processes must be handled with sensitivity to preserve trust and engagement within the learning community. *Socially*, attention must be given to ensuring that the algorithm supports, rather than undermines, the collaborative and dialogic nature of learning. While algorithms can sieve through discordant feedback to create cohesion, they must be implemented in a way that continues to value and foster interpersonal communication and collaborative learning among

students. Thus, the overarching implication is the need for a balanced, ethical, and strategically aligned incorporation of algorithms, which enhances rather than eclipses the human-centric ethos that underpins educational environments.

CHAPTER

7

RELIABILITY-BASED WEIGHTED RATINGS FOR BETTER PEER GRADING

7.1 Introduction

Assessment is a fundamental component of the learning process. It provides students with critical insights to improve their understanding of a subject. It is crucial to make the assessment effective and accurate. Peer assessment is an effective evaluation tool. However, the validity of peer assessment is often scrutinized due to the variability in assessors' proficiency and willingness (Zarkoob et al. (2023)). A significant challenge lies in measuring the reliability of peer assessors to ensure the quality of assessment (Piech et al. (2013); Zarkoob et al. (2023); Hamer et al. (2005)).

Typically, peer assessors provide two types of feedback: 1) formative and 2) summative (Evans (2013)). While formative feedback offers students deeper insights, summative feedback is essential for grading. In peer assessment, typically, multiple assessors provide summative feedback (Kulkarni et al. (2015)). While grading from multiple assessors increases the validity of the assessment (Hamer et al. (2005)), it poses a challenge to aggregating scores from multiple peer assessors. Most peer assessment systems utilize aggregated methods such as mean and median to determine students' grades based on the scores provided by assessors (Denny et al.

(2008); Paré and Joordens (2008); Wind et al. (2018)). These methods incorporate the judgments of each reviewer equally. However, prior research has indicated that the statistical mean and median do not always accurately represent the true grade Darvishi et al. (2020). To enhance grading accuracy, some studies have developed methods to estimate students' reliability, bias, and effort (Piech et al. (2013); Zarkoob et al. (2023); Chakraborty et al. (2024)). Nonetheless, these studies have predominantly focused on summative feedback and employed a single round of peer assessment.

In this study, we propose a probabilistic approach to quantify the reliability of peer assessors. Our approach uniquely integrates formative feedback and summative assessment in two assessment rounds. In the first round, designated as Round 1, peer assessors provide formative feedback, allowing students to refine their work and enhance their understanding. In Round 2, students submit their revised work based on the feedback received in Round 1, and the same peer assessors provide summative feedback and grade the artifacts.

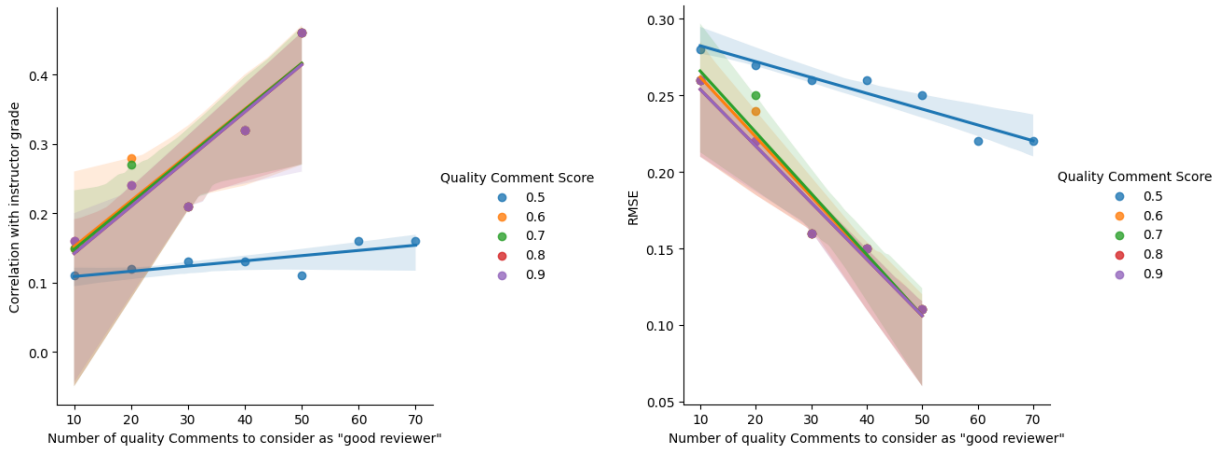
Our findings indicate that peer assessors who invest significant effort in writing high-quality formative feedback also tend to provide summative scores that closely align with those of the instructor. As illustrated in Figure 7.1, peer assessors who frequently write quality feedback tend to give scores that correlate more strongly with the instructor's assessments. Additionally, assessors who consistently write quality comments exhibit lower root-mean-square error relative to instructor grades.

Based on these observations, we developed a probabilistic method that (1) estimates the effort of peer assessors in writing quality comments during Round 1 and (2) evaluates the reliability of assessors by incorporating both their effort in Round 1 and the summative feedback provided in Round 2. We hypothesize that quality comments are indicative of the assessors' effort in the assessment process. In our study, we define quality comments as those that include an identified issue and/or a provided suggestion. For this study, we analyzed 5,596 formative and 2,634 summative feedback from an object-oriented design and development course over four semesters. A total of 106 students participated in the peer assessment.

Our approach demonstrated an 80% increase in the correlation between peer-assigned grades and instructor grades compared to the traditional aggregation-based mean and median grades. To our knowledge, this is the first reliability-based grading system that uniquely utilizes the quality of formative feedback, summative feedback, and multiple assessment rounds.

7.2 Background

In the peer assessment process, summative feedback is typically used to grade students, with multiple reviewers evaluating each artifact. To calculate final grades, aggregate-based meth-



(a) Correlation between good reviewers' and instructor ratings

(b) RMSE with instructor ratings

Figure 7.1: Comparison of quality reviewers' ratings with instructor's ratings.

ods such as mean and median are commonly used. However, simple aggregation may not account for inaccuracies in peer assessments Darvishi et al. (2020). Research shows that while students can assess peers accurately in formative contexts, peer grading can be inconsistent in summative contexts Sridharan et al. (2019).

To address this issue, methods like Calibrated Peer Review (CPR) evaluate the reliability of peer reviewers through benchmark assessments. However, CPR can be cumbersome and does not account for inconsistencies over time Carlson and Berry (2003); Balfour (2013). Alternative approaches include score calibration algorithms that weight peer assessments based on their accuracy Hamer et al. (2005), and Bayesian methods that adjust grades based on detected biases (Goldin (2012); Piech et al. (2013)). More recent models also estimate peer assessors' reliability, bias, and effort (Zarkoob et al. (2023)). Our study employs a probabilistic Bayesian model that incorporates both formative and summative assessments to evaluate peer assessors' reliability.

7.3 Approach:

In this section, we present our methodology for inferring the reliability of peer assessors based on formative and summative feedback across two distinct rounds. In the first round (Round 1), assessors provide formative feedback, from which we estimate each peer assessor's effort in writing reviews. The second round (Round 2) involves summative feedback, where we incorporate the effort inferred from Round 1 and the grades assigned by reviewers in Round 2

to estimate the true latent scores of the artifacts. With these true latent scores, we then calculate the reliability scores for each reviewer. These reliability scores are subsequently used to derive a weighted average score for the artifacts, serving as the final grade.

Notice that in this two-round peer assessment approach, students improve their artifacts based on the comments received in Round 1 and then resubmit them for the second round of assessment, where they receive ratings from peer assessors.

7.3.1 Notation and Problem Statement

Assume D is a dataset that contains a set of reviewer $R = \{r_1, \dots, r_j\}$, for a set of artifacts $A = \{a_1, \dots, a_i\}$ using criterion $F = \{f_1, \dots, f_k\}$ and $C = \{c_1, \dots, c_l\}$ where F is the set of rubric items for formative feedback (used in assessment Round 1) and $|F| = M$. Similarly, C is the set of rubric items for summative feedback (used in assessment Round 2). Let's assume that d_{ijk} represents the review comment for an artifact i for criterion k by reviewer j and d_j is the set of artifacts reviewed by r_j , where $|d_j| = N$. We define $P = \{\pi_1, \dots, \pi_j\}$, where π_j is the effort of r_j for providing review using F . Similarly, in Round 2, A has a set of underlying true score $B = \{\beta_{1l}, \dots, \beta_{il}\}$ in response to C . We define $W = \{w_1, \dots, w_j\}$, where w_j is the reliability of r_j . Based on these notations, we formally define this problem as follows:

Formal Problem statement: Given data set D we infer P , B and W .

We used the following notations to indicate observed variables:

- **Likelihood of a comment being quality:** We define the likelihood of a comment being quality as X_{ijk} of a peer assessor r_j on an artifact a_i , using the criterion f_k .
- **Peer assessor's summative feedback:** The rating given by a peer assessor r_j to an artifact a_i using criterion c_l is Y_{ijl} .

7.3.2 Proposed Approach

We define our proposed approach as an inference problem, aiming to infer the latent variables representing peer assessors' effort (P) in writing quality comments and the latent true scores of artifacts in Round 2 (B). By inferring P and B , we seek to quantify the reliability of assessors (W) in providing accurate ratings. These reliability scores (W) are then employed as weights to compute the weighted average, which is used to assign final grades to students.

To infer the latent values, we frame the problem as inferring the posterior distribution over the observed review quality and the scores given by peer assessors. Formally, our inference approach can be expressed through probabilistic modeling as $P(\pi_j, \beta_{il} | X_{ijk}, Y_{ijl})$.

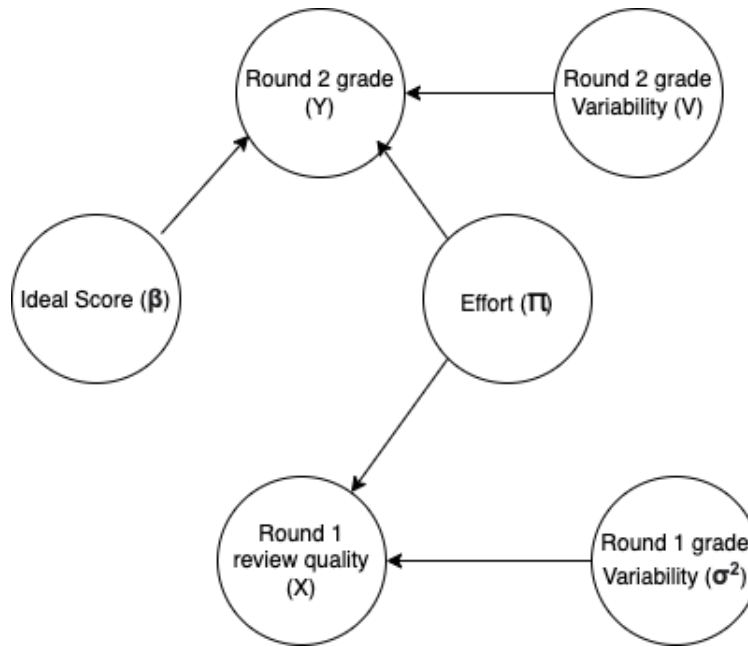


Figure 7.2: Inference model of Peer assessors' effort (π) in writing quality comments in Round 1 and ideal Score (β) in Round 2

7.3.3 Extracting review quality likelihood

In this study, the assessment of review comment quality hinges on the presence of problem identification and/or suggestion provision. Participants were instructed to articulate identified issues and propose resolutions. Machine learning techniques were employed to automatically detect these attributes within review texts. A comparative evaluation encompassing various machine learning and deep learning models was conducted. Supervised training of these models utilized meticulously curated datasets delineated in the Data section. Two distinct models were developed to categorize review comments based on the presence of mentioned problems and provided suggestions. Utilizing the optimal models identified for each task, probabilities were computed for the presence of a problem and suggestion in review comments. The composite measure of review quality (X_{ijk}) was derived as the mean of these probabilities. In this process, higher scores were assigned to comments that contained both a problem and provided a suggestion, compared to those that only identified an issue or offered a suggestion alone.

7.3.4 Inference Model

The Bayesian approach to assessors' reliability updating depends on information that we collect over two rounds as described earlier. We model a peer assessor's comment quality while writing

a formative feedback as a Normal distribution with the mean being a function of the assessor's effort. The probit transformation $\Phi^{-1} : (0, 1) \mapsto \mathbb{R}$ is employed simply to convert the data to the real line, denoted by \mathbb{R} . For the transformed effort parameter, we use a non-informative conjugate Normal prior with mean 0 and variance 1. We also model the assessor's variation in writing comments, denoted by the hyperparameter σ^2 , using a non-informative Inverse Gamma prior distribution. Owing to conjugacy, the conditional posterior distributions for the parameters in the model will belong to the same distributional family as the prior. Consequently, the conditional posterior of the transformed effort parameter will be normally distributed. We will input this posterior distribution of $\Phi^{-1}(\pi_j)$ from Round 1 as the prior distribution in Round 2, thereby making sure that information from the first round is carried forward to the second.

Moving on to Round 2, we collect data in the form of summative scores which we denote by Y_{ijl} corresponding to the i -th student, l -th rubric item rated by reviewer j . Again, transforming the data to the real line using the probit function, we model it using the normal distribution. The model parameters that we use in this round are not only innovative, but captures the nuances of this complex data structure very succinctly. The mean of the Normal prior is considered to be the ideal score (β_{il}) that the i -th student should have received for the l -th rubric item. The variance, on the other hand, is defined as $V(1 - \pi_j)$, where V is the assessor's variability in grading and $(1 - \pi_j)$ can be looked upon as a weight. An assessor who put more effort in Round 1 in writing quality comments should have higher π_j values and hence is expected to be less variable in rating a student in Round 2. Similarly, a less reliable reviewer with low π_j values will lead to higher variability in their summative feedback. For the priors of the model parameters β_{il} and V , we respectively use a non-informative $N(0, 1)$ distribution and a non-informative Inverse Gamma distribution. On the other hand, for the π_j parameters, we use informative priors based on the posterior distributions of π_j 's from Round 1, obtained in (7.1).

Once the prior distributions have been specified, the conditional posterior computations exploiting the conjugacy property are relatively easy. We not only have closed-form expressions for all the conditional posteriors, they are known distributions, and hence, posterior sampling is fast and efficient. We obtain 10,000 posterior samples using the Gibbs sampling technique and employ a burn-in of 1,000 samples to ensure that the samples are independently observed from the respective posterior distributions. Figure 7.2 illustrates the graphical model of our approach, depicting the inference of peer assessors' effort in writing quality comments in Round 1 and the latent true scores of artifacts in Round 2. The inference model and the closed-form solution are shown below.

Peer assessors' effort estimation:

(Assessor's effort) $\Phi^{-1}(\pi_j) \sim \mathcal{N}(0, 1) \quad \forall j = 1, 2, \dots, n$

(Assessor's effort variability) $\sigma^2 \sim \text{InvGamma}(\text{Non Informative}) \propto \frac{1}{\sigma^2}$

(Assessor's comment quality) $\Phi^{-1}(x_{ijk}) \sim \mathcal{N}(\Phi^{-1}(\pi_j), \sigma^2) \quad \forall i, j, k$

Round 1 Posterior computation:

$$z_{ijk} = \Phi^{-1}(x_{ijk}), \quad \xi_j = \Phi^{-1}(\pi_j)$$

$$z_{ijk} | \xi_j \sim \mathcal{N}(\xi_j, \sigma^2)$$

$$\xi_j \sim \mathcal{N}(0, 1)$$

$$p(\sigma^2) \propto \frac{1}{\sigma^2}$$

Conditional posterior of ξ_j :

$$\begin{aligned} P(\xi_j | z_j, \sigma^2) &\propto \left(\prod_{(i,k)} e^{-\frac{1}{2\sigma^2}(z_{ijk} - \xi_j)^2} \right) e^{-\frac{1}{2}\xi_j^2} \\ &= \prod_{(i,k)} e^{-\frac{1}{2\sigma^2}(z_{ijk}^2 - 2\xi_j z_{ijk} + \xi_j^2) - \frac{1}{2}\xi_j^2} \\ &= e^{-\frac{1}{2\sigma^2} \sum_{(i,k)} z_{ijk}^2} \cdot e^{\frac{1}{\sigma^2} \xi_j \sum_{(i,k)} z_{ijk}} e^{-\frac{1}{2}\xi_j^2 (1 + \frac{MN}{\sigma^2})} \\ &\propto e^{-\frac{1}{2\left(\frac{\sigma^2}{\sigma^2 + MN}\right)} \left\{ \xi_j^2 - \frac{2\xi_j}{\sigma^2 + MN} \sum_{(i,k)} z_{ijk} \right\}} \\ &\propto e^{-\frac{1}{2\left(\frac{\sigma^2}{\sigma^2 + MN}\right)} \left\{ \xi_j^2 - 2 \cdot \xi_j \frac{1}{\sigma^2 + MN} \sum_{(i,k)} z_{ijk} + \left(\frac{\sum_{(i,k)} z_{ijk}}{\sigma^2 + MN} \right)^2 \right\}} \\ &= e^{-\frac{1}{2\left(\frac{\sigma^2}{\sigma^2 + MN}\right)} \left(\xi_j - \sum_{(i,k)} \frac{z_{ijk}}{\sigma^2 + MN} \right)^2} \end{aligned}$$

$$\text{Thus, } P(\xi_j | z_j, \sigma^2) \sim \mathcal{N}\left(\frac{\sum_{(i,k)} z_{ijk}}{\sigma^2 + MN}, \frac{\sigma^2}{\sigma^2 + MN} \right). \quad (7.1)$$

Peer assessors' reliability estimation:

(Ideal score) $\beta_{il} \sim \mathcal{N}(0, 1)$

(Ideal score Variability) $V \sim \text{InvGamma}(\text{Non-informative})$

(Assessor's given score) $\Phi^{-1}(Y_{ijl}) \sim \mathcal{N}(\beta_{il}, V(1 - \pi_j))$

where, $\Phi^{-1}(\pi_j) \sim \mathcal{N}\left(\sum_{(i,k)} \frac{\Phi^{-1}(x_{ijk})}{\sigma^2+MN}, \frac{\sigma^2}{\sigma^2+MN}\right)$ is the posterior from Round 1 and used as prior in Round 2.

Posterior computation

$$\omega = \Phi^{-1}(Y_{ijl})$$

$$P(\beta_{il} | \omega, \pi_j, V) \propto \left(\prod_j e^{-\frac{1}{2V(1-\pi_j)}(\omega-\beta_{il})^2} \right) e^{-\frac{1}{2}\beta_{il}^2}$$

Where, j is the set of all reviewers who graded i on rubric item l .

Let $s_j^2 = V(1-\pi_j)$. Then

$$\begin{aligned} P(\beta_{il} | \omega, s_j^2) &\propto e^{-\frac{1}{2}\sum_j \frac{\omega^2 - 2\beta_{il}\omega + \beta_{il}^2}{s_j^2}} e^{-\frac{1}{2}\beta_{il}^2} \\ &\propto e^{-\frac{1}{2}\beta_{il}^2 \sum_j \frac{1}{s_j^2} + \beta_{il} \sum_j \frac{\omega}{s_j^2} - \frac{1}{2}\beta_{il}^2} \\ &= e^{-\frac{1}{2}\beta_{il}^2 \left(1 + \sum_j \frac{1}{s_j^2}\right) + \beta_{il} \sum_j \frac{\omega}{s_j^2}} \\ &= e^{-\frac{1}{2\left(1 + \sum_j \frac{1}{s_j^2}\right)} \left\{ \beta_{il}^2 - 2\beta_{il} \frac{\sum_j \frac{\omega}{s_j^2}}{1 + \sum_j \frac{1}{s_j^2}} \right\}} \end{aligned}$$

and by completing the square, we have

$$\beta_{il} | \omega, s_j^2 \sim \mathcal{N}\left(\frac{\sum_j \frac{\omega}{s_j^2}}{1 + \sum_j \frac{1}{s_j^2}}, \frac{1}{1 + \sum_j \frac{1}{s_j^2}}\right)$$

Once we get posterior samples of the ideal score β_{il} that the i -th student should have ideally received for their l -th rubric item, we define the weight for the j -th reviewer as -

$$w_j = \frac{\pi_j}{\sum_{(i,l)} (\Phi^{-1}(Y_{ijl}) - \beta_{il})^2}$$

7.4 Experimental Setting

7.4.1 Research Tool

Our study used the Expertiza system to implement the peer assessment process from an object-oriented design and development course (CSC 517) taught by Dr. Edward Gehringer at NC

State University. A total of 106 students participated in the peer assessment process. Typically, the course includes three peer-reviewed assignments each semester, with students working in teams of two to four members. A total of 40 teams were formed to work on the assignment. Although the assignments are completed in a group setting, individual students from other groups conduct the reviews.

7.4.2 Data collection and Context

The assignment used for this study is called “Program 2”. In this assignment, students are asked to make a web application using the Ruby on Rails framework. The assignments are completed in 5 phases. In phase 0, students submit their work for peer assessment. In Phase 1, the Expertiza system assigns 2–4 peer assessors for each artifact. In this phase, the assessors provide a round of assessment (Round 1). In this round, peer assessors are asked to provide detailed formative feedback, specifically mentioning any issue they found in the work and to provide suggestions to improve the work. Following peer feedback, in Phase 2, student teams revise their work and resubmit it for grading. Phase 3 is the second round of assessment (Round 2), which is summative, where peer reviewers provide numerical scores indicating the quality of an artifact. However, in Phase 4, instructors and/or TAs provide the final score based on the submissions in Phase 2 submissions. For this study, we have experts’ (instructors and TAs) grades for each submission and grading criteria. The experts’ grades have been collected separately and did not affect the peer grading process. Figure: 7.3 depicts the different phases involved in the assignment. The assessment was double-blinded. For Round 1, peer assessors were shown examples of quality formative feedback and asked to mention any issue they found in the artifact and provide suggestions to mitigate them. Peer assessors followed a rubric consisting of 31 items as review criteria to provide their comments. We collected 5,596 comments from Round 1. For review Round 2, assessors followed a rubric of 12 criteria and provided 2,634 summative feedback.

7.4.3 Dataset Preparation

The performance of supervised machine learning models relies on the quality of the training data. However, acquiring high-quality labeled data can be costly. For our experiment, we have obtained two distinct labeled datasets with strict quality control measures.

- Problem-detection: A review comment is labeled yes or no according to whether it detects a problem.

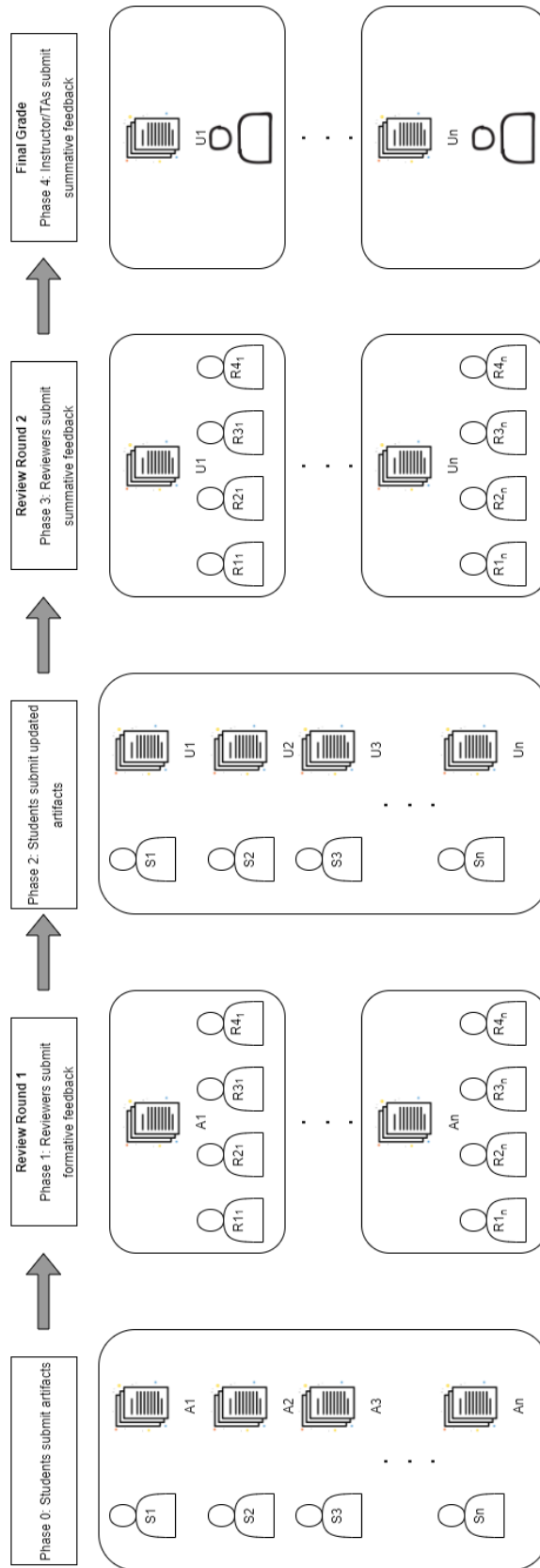


Figure 7.3: Formative (Round 1) and summative (Round 2) assessment rounds in four phases.

Table 7.1: Sample review comment and annotations done by students ('1' indicates 'yes' and '0' indicates 'no')

Review Comment	Detects Problem
The Travis CI Build is Failing as of now. No conflicts as per the GitHub report.	1
Yes, the explanation is elaborative and complete.	0
Since the build failed, I would not recommend adding it to the production server yet.	1

Review Comment	Gives Suggestion
Test Plan is too verbose. Trivial areas can be trimmed off.	1
The team needs to look into Travis CI log & 1	1
Many test cases in terms of controllers, but none for models.	0

- Suggestion-detection: A review comment is labeled yes or no according to whether it contains a suggestion.

In the Chapter 3 section 3.1.1, we described our data annotation process. Using the described process, we collected 18,392 annotations for problem-detection, 7,416 for suggestion-detection. Both datasets have an equal ratio of binary class labels. Table 7.1 displays sample comments and annotations from the datasets. We utilized supervised machine learning models trained using the annotated dataset to quantify the quality of the peer assessors' review comments in Round 1 of the assessment.

7.5 Results

This study defines quality feedback as any comment that addresses problems and/or offers suggestions. Our initial approach involves constructing two distinct models: one to detect the presence of problem statements in feedback and another to identify suggestions. To ascertain the most effective models, we trained and evaluated the performance of various classical machine learning models alongside neural network models, subsequently comparing their f1-score.

Within the problem-detection dataset, the classical machine learning model, Support Vector Machine (SVM), attained the highest F1-score of 0.90. Conversely, among the neural network models, BERT achieved the highest F1-score of 0.92. In the suggestion-detection dataset, BERT also demonstrated superior performance with an F1-score of 0.91, whereas the SVM model achieved an F1-score of 0.87 (Figure: 7.4).

We inferred each peer assessor's effort in writing review comments in Round 1 of the assessment using Bayes inference. As the model considers the quality of the reviews written

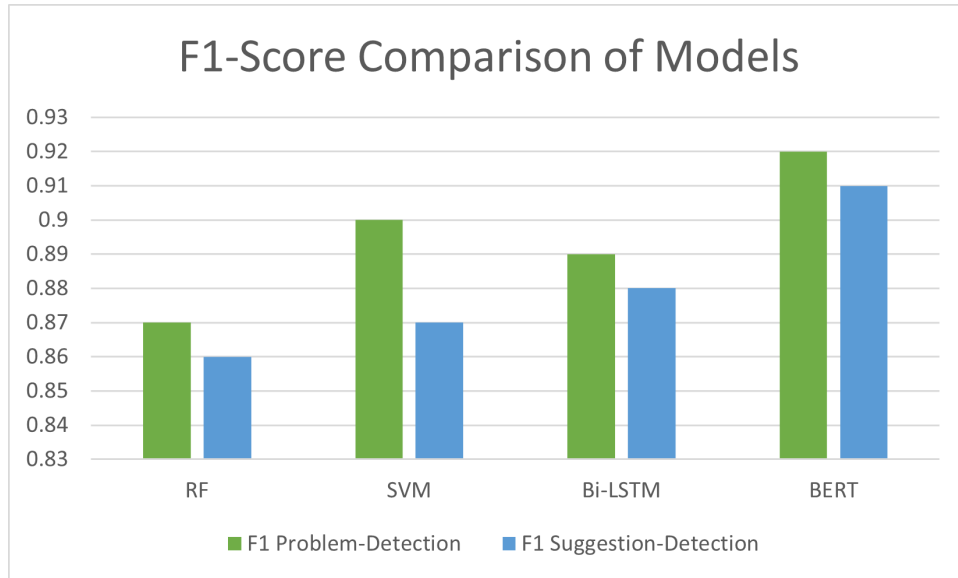


Figure 7.4: F1-score comparison to measure performance on classifying review text on problem detection and suggestion detection using classical ML and neural-network models. In the overall F1-score comparison, the BERT model shows the best performance.

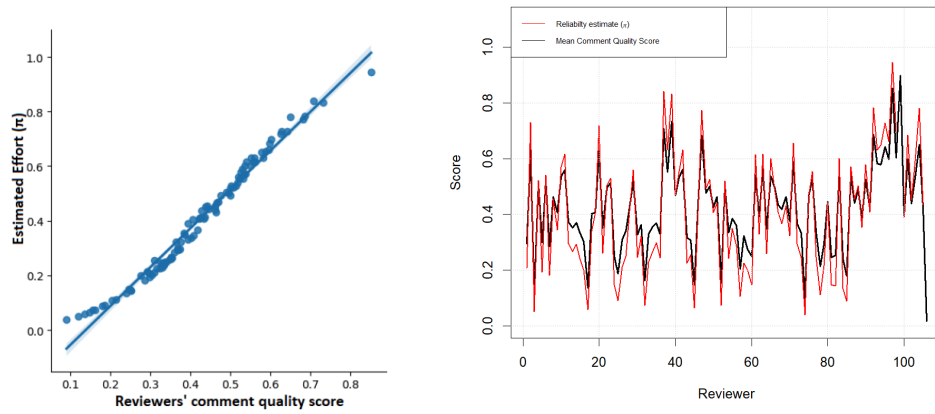
. As the BERT model performed the best in classifying the review text, we used BERT to produce the probability of the review containing a problem or suggestion.

by the peer assessors, we found that the reliability scores inferred from Round 1 are highly correlated (r -value .98 and p -value .001) with the mean of the review quality of each peer assessor (Figure: 7.5).

In Figure 7.6 we are showing some features of the posterior distribution of the reviewers' effort (π). We sort the reviewers from the lowest effort to the highest and plot the posterior mean for all reviewers in Round 1. We also show the sampling variability in the form of the 95% credible intervals for each reviewer plotted in dotted lines in Figure 7.6. We observe a slight increase in the length of the credible intervals towards the higher end. This, however, does not affect the convergence of the Gibbs sampler in any way, as shown by MCMC trace plots in Figure 7.7.

In Round 2, students submitted their revised artifacts following the feedback they received in Round 1. Peer reviewers were given a different rubric in Round 2 to rate the artifacts. These ratings from the assessors were counted towards the final grade. For Round 2, we implemented Bayes inference to estimate the reliability scores for each reviewer from the ratings they provided. The final reliability scores were estimated considering the inferred effort from Round 1 and observed ratings provided by the reviewers in Round 2.

In Table 7.2, we have shown the correlation between the instructor ratings and peer assessors' ratings where the peer assessors' ratings are availed using reliability-based weighted



(a) Correlation between peer assessors' mean review score and estimated effort. (b) Scatter plot of estimated effort score and assessors' mean review score.

Figure 7.5: Comparison of quality reviewers' mean review scores with estimated effort.

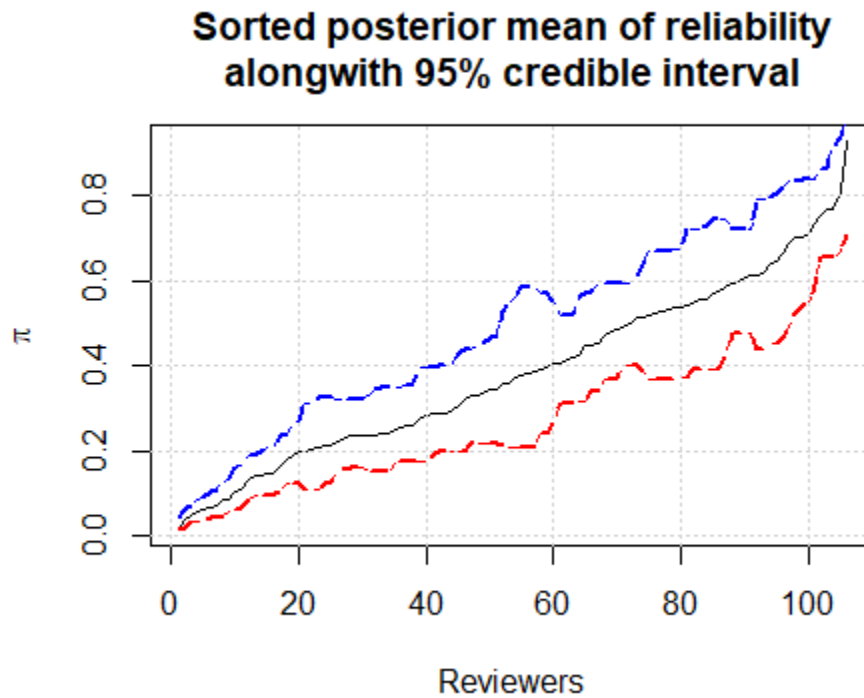


Figure 7.6: Distribution of peer assessors' estimated effort (π)

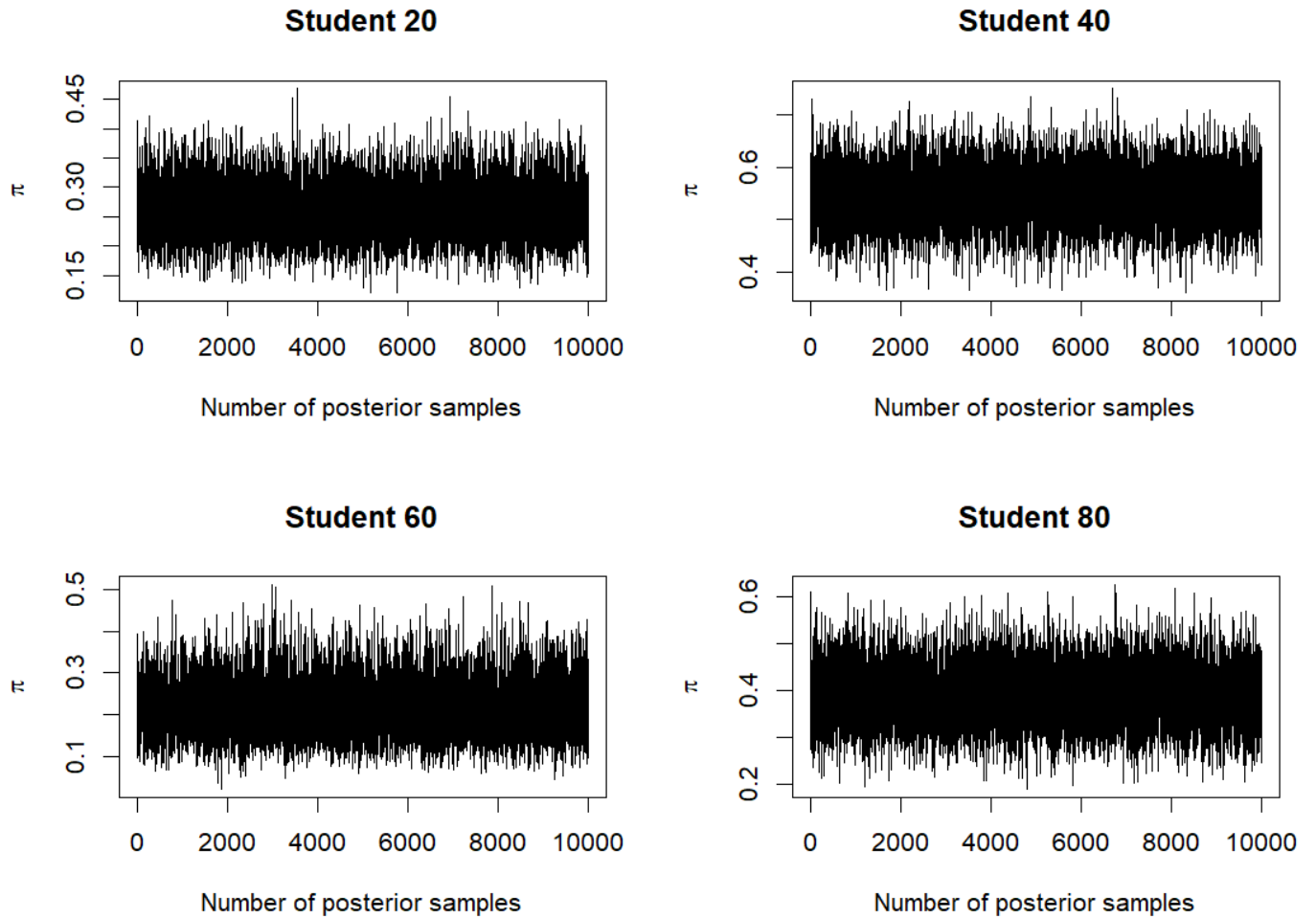


Figure 7.7: Gibbs convergence for estimating reviewers effort in the assessment Round 1 for a few randomly picked peer assessors.

	Method	Correlation	
		<i>r</i> -value	<i>p</i> -value
Aggregation-based Methods	Mean	0.20	$p < 0.001$
	Median	0.23	$p < 0.001$
Reliability-based Method	Weighted Mean	0.36	$p < 0.001$

Table 7.2: Correlation between instructor and peer assessors ratings. The ratings from the peer assessors are derived using mean, median, and weighted average

average and aggregation-based methods, such as mean and median. We observed the correlation using the reliability-based weighted average is 0.36. Conversely, aggregation-based mean, mode and median have correlation scores respectively 0.20, 0.23. In our study, we achieved an 80% increase in correlation using the reliability-based weighted mean than the aggregation-based mean.

7.6 Conclusion

We implemented a probabilistic method based on Bayes inference to quantify the reliability of peer assessors in a real classroom. The unique contribution of this paper is that it infers reliability from formative and summative feedback and can be implemented with assessments with different rounds. Our approach is scalable to any classroom size, both online and brick-and-mortar classrooms.

We collected formative and summative assessments from an Object Oriented Design and Development course. We analyzed 5,596 formative feedback and 2,634 summative feedback in two separate rounds, where 106 students participated in peer assessment. The quality of the formative feedback was determined based on the criteria that the reviewer mentioned issues and/or provided suggestions.

We identified that students who frequently write quality reviews also provide more accurate grades. Based on this finding, we hypothesize that if we put more weight on the grades of the reviewers who put more effort into writing high-quality reviews, it will provide grades that are more correlated with the instructor. Using Bayes inference, we were able to estimate the reliability of the peer assessors, and the reliability score was highly correlated with the review quality. Finally, we used the reliability scores to calculate the weighted average of the students' grade on each rubric item. We achieved 80% improvement in correlation using our probabilistic method.

While our paper achieved a great improvement in the correlation with the instructor's grade, there remains room to improve. Our study identified that inferring the instructor's score is extremely difficult in our scenario. In our classroom, students resubmitted a finer copy of the artifact for peer assessors' ratings after improving their initial work following the feedback from Round 1 of the assessment. As a result, it contained minimum issues. In our dataset, 2,034 student ratings out of 2,634, that is, 78% of the student ratings are the same as the instructor. 88% of grades are within 20% of the instructor ratings. All the rubric items were rated out of 5. A mere 1 point distance from the instructor put the grader 20% points far from the instructor's rating.

CHAPTER

8

CONCLUSION AND FUTURE WORK

There is a growing number of studies acknowledging the impact of peer assessment on students' learning gain and reducing assessment load for instructors. However, there is a gap in research on enhancing assessment quality, accuracy, and trustworthiness. My research involves addressing improving the quality of formative feedback and the accuracy of summative feedback to enhance the effectiveness and reliability of peer assessment.

Chapter 4 of this thesis focuses on identifying the characteristics of effective peer review comments from students' perspectives. My research examined helpful feedback through a two-step process. First, advanced NLP methods are employed to automatically determine if the review comments identify problems and/or propose solutions. Second, my study compared the helpfulness of comments that mention problems with those that also suggest solutions. Among comments annotated helpful by students, 64% identified a problem and gave a suggestion. Conversely, 66% of comments labeled as not helpful by students neither mentioned a problem nor provided a suggestion. The experimental results indicate that students find review comments most beneficial when they both identify an issue in the artifact and suggest a solution.

Peer assessment involves two types of feedback: (i) summative, which is used for grading, and (ii) formative, which provides constructive comments for improvement. The quality of formative feedback heavily relies on the reviewers' understanding of the assessment criteria.

Thus, rubrics must be well-crafted to guide peer assessors effectively. In Chapter 5, I explored a natural language processing (NLP) approach to analyze rubric text to determine if it prompts high-quality reviews. I analyzed 408,104 formative feedback comments based on 3,164 rubric criteria using advanced neural network methods, marking the first attempt to enhance peer-review comments through rubric prompt analysis.

Disagreements among peer assessors' formative feedback can confuse students and lead to doubts about the review process and reviewers' competency. While peer assessment is valuable for providing high-quality feedback, contradictory comments undermine its effectiveness. Clear communication is essential for actionable feedback, and conflicting feedback can hinder student understanding. Meta-reviewing to identify disagreements is labor-intensive for instructors and delays feedback. In Chapter 6, my research introduced a semi-supervised training approach for transformer-based models to generate high-quality vector representations of text. This approach creates feature vectors of peer reviews, positioning comments in agreement close to each other in the vector space and placing comments in disagreement farther apart. Our results demonstrate that these quality text feature representations enhance the clustering algorithm's ability to identify and quantify disagreements in formative feedback. This study enables instructors to pinpoint peer assessments that require meta-reviewing.

Peer assessors can dedicate more time to assessments than instructors, potentially leading to higher-quality evaluations. However, not all peer assessors are equally competent, and involving instructors to identify assessor competency can be more burdensome than conducting the assessments themselves. To enhance reliability, multiple reviewers typically assess each artifact, but this introduces the challenge of aggregating ratings. Independent ratings by peer assessors can vary, and summary statistics may incorporate unreliable scores and impact the accuracy of the ratings. In Chapter 7, I implemented a probabilistic method based on Bayes inference to quantify peer assessors' reliability, applicable in real classrooms. The unique contribution of my study is inferring reliability from both formative and summative feedback, which is usable across different assessment rounds. From the formative round, we infer reviewers' effort in providing quality feedback. Incorporating the effort in the summative feedback round we calculated the reliabilities of the peer reviewers. We then use weighted mean, with reliability scores as weights, to provide ratings to the reviewees. This scalable approach can be applied in both online and traditional classrooms. Our method improved the correlation with instructor ratings by 80% compared to the statistical mean and median-based ratings.

In my thesis, Chapters 4–6 are dedicated to improving the quality of formative feedback. Chapter 7 is dedicated for accurate and reliable summative feedback. The findings of my studies shed light on methods to increase the effectiveness and trustworthiness of peer assessment. However, there are important limitations to address in the future work. One such

limitation is the interpretation of formative feedback quality. While mentioning problems and providing suggestions, identify review as helpful to the students; however, other factors such as encouragement, localization of issues, and tone of the reviewer are among the many other qualities that can enhance formative feedback quality and effectiveness. With the recent improvement in Large Language Models, it is worth studying if the LLMs can pinpoint the issues that reviewers are mentioning in the review. It will enable the reviewers to provide hits to update their review based on the issue they are disregarding and ultimately provide coherent and actionable effective feedback for the reviewee. While identifying the reliability of the peer assessors increases the correlation with the instructor grade, there is room to improve the accuracy of aggregated ratings as close as instructors' ratings.

REFERENCES

- Abbasi-Moud, Z., Vahdat-Nejad, H., and Sadri, J. (2021). Tourism recommendation system based on semantic clustering and sentiment analysis. *Expert Systems with Applications*, 167:114324.
- Ashton, S. and Davies, R. S. (2015). Using scaffolded rubrics to improve peer assessment in a mooc writing course. *Distance education*, 36(3):312–334.
- Balfour, S. P. (2013). Assessing writing in moocs: Automated essay scoring and calibrated peer review™. *Research & Practice in Assessment*, 8:40–48.
- Black, P. and Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: principles, policy & practice*, 5(1):7–74.
- Branco, P., Torgo, L., and Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM computing surveys (CSUR)*, 49(2):1–50.
- Cambre, J., Klemmer, S., and Kulkarni, C. (2018). Juxtapeer: Comparative peer review yields higher quality feedback and promotes deeper reflection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13.
- Capuano, N., Caballé, S., and Miguel, J. (2016). Improving peer grading reliability with graph mining techniques. *International Journal of Emerging Technologies in Learning (Online)*, 11(7):24.
- Carless, D. and Boud, D. (2018). The development of student feedback literacy: enabling uptake of feedback. *Assessment & Evaluation in Higher Education*, 43(8):1315–1325.
- Carlson, P. A. and Berry, F. C. (2003). Calibrated peer review and assessing learning outcomes. In *Frontiers in education conference*, volume 2, pages F3E–1. STIPES.
- Çevik, Y. D. (2015). Assessor or assessee? investigating the differential effects of online peer assessment roles in the development of students' problem-solving skills. *Computers in Human Behavior*, 52:250–258.
- Chakraborty, A., Jindal, J., and Nath, S. (2024). Removing bias and incentivizing precision in peer-grading. *Journal of Artificial Intelligence Research*, 79:1001–1046.
- Churches, A. (2008). Bloom's taxonomy blooms digitally. *Tech & Learning*, 1:1–6.
- Darvishi, A., Khosravi, H., and Sadiq, S. (2020). Utilising learnersourcing to inform design loop adaptivity. In *European Conference on Technology Enhanced Learning*, pages 332–346. Springer.
- Denny, P., Hamer, J., Luxton-Reilly, A., and Purchase, H. (2008). Peerwise: students sharing their multiple choice questions. In *Proceedings of the fourth international workshop on computing education research*, pages 51–58.

- Devlin, J. (2018). *BERT sentence vector*. <https://github.com/google-research/bert/issues/164>.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- du Toit, J. (2019). Enhancing the quality of essays through a student peer-review process. In *International Conference on Innovative Technologies and Learning*, pages 459–468. Springer.
- Evans, C. (2013). Making sense of assessment feedback in higher education. *Review of educational research*, 83(1):70–120.
- Galley, M., McKeown, K., Hirschberg, J. B., and Shriberg, E. (2004). Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies.
- Gamage, D., Staubitz, T., and Whiting, M. (2021). Peer assessment in moocs: Systematic literature review. *Distance Education*, 42(2):268–289.
- Gehring, E. F. (2010). Expertiza: Managing feedback in collaborative learning. In *Monitoring and assessment in online collaborative environments: Emergent computational technologies for e-learning support*, pages 75–96. IGI global.
- Gehring, E. F. (2014). A survey of methods for improving review quality. In *International Conference on Web-Based Learning*, pages 92–97. Springer.
- Glance, D. G., Forsey, M., and Riley, M. (2013). The pedagogical foundations of massive open online courses. *First monday*.
- Goldin, I. M. (2012). Accounting for peer reviewer bias with bayesian models. In *Proceedings of the workshop on intelligent support for learning groups at the 11th international conference on intelligent tutoring systems*. Citeseer.
- Graner, M. H. (1987). Revision workshops: An alternative to peer editing groups. *The English Journal*, 76(3):40–45.
- Guan, R., Zhang, H., Liang, Y., Giunchiglia, F., Huang, L., and Feng, X. (2020). Deep feature-based text clustering and its explanation. *IEEE Transactions on Knowledge and Data Engineering*.
- Hamer, J., Ma, K. T., and Kwong, H. H. (2005). A method of automatic grade calibration in peer assessment. In *Proceedings of the 7th Australasian conference on Computing education-Volume 42*, pages 67–72.
- Hattie, J. and Timperley, H. (2007). The power of feedback. *Review of educational research*, 77(1):81–112.
- He, H. and Ma, Y. (2013). Imbalanced learning: foundations, algorithms, and applications.
- Hillard, D., Ostendorf, M., and Shriberg, E. (2003). Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Companion Volume of the Proceedings of HLT-NAACL 2003-Short Papers*, pages 34–36.

- Hiray, S. and Duppada, V. (2017). Agree to disagree: Improving disagreement detection with dual grus. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 147–152. IEEE.
- Huawei, S. and Aryadoust, V. (2023). A systematic review of automated writing evaluation systems. *Education and Information Technologies*, 28(1):771–795.
- Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., et al. (2003). The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, volume 1, pages I–I. IEEE.
- Jia, Q., Cui, J., Xiao, Y., Liu, C., Rashid, P., Gehringer, E., et al. (2021). All-in-one: Multi-task learning bert models for evaluating peer assessments. arxiv.
- Jinarat, S., Manaskasemsak, B., and Rungsawang, A. (2018). Short text clustering based on word semantic graph with word embedding model. In *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, pages 1427–1432. IEEE.
- Jonsson, A. and Svingby, G. (2007). The use of scoring rubrics: Reliability, validity and educational consequences. *Educational research review*, 2(2):130–144.
- Joyner, D. A. (2017). Scaling expert feedback: Two case studies. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 71–80.
- Kao, G. Y.-M. (2013). Enhancing the quality of peer review by reducing student “free riding”: Peer assessment with positive interdependence. *British Journal of Educational Technology*, 44(1):112–124.
- Khosravi, H., Shum, S. B., Chen, G., Conati, C., Tsai, Y.-S., Kay, J., Knight, S., Martinez-Maldonado, R., Sadiq, S., and Gašević, D. (2022). Explainable artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 3:100074.
- Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage publications.
- Kulkarni, C. E., Bernstein, M. S., and Klemmer, S. R. (2015). Peerstudio: rapid peer feedback emphasizes revision and improves performance. In *Proceedings of the second (2015) ACM conference on learning@ scale*, pages 75–84.
- Lauw, H. W., Lim, E.-P., and Wang, K. (2007). Summarizing review scores of “unequal” reviewers. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 539–544. SIAM.
- Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. (2020). On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.
- Li, L. and Grion, V. (2019). The power of giving feedback and receiving feedback in peer assessment. *All Ireland Journal of Higher Education*, 11(2).

- Liu, N.-F. and Carless, D. (2006). Peer feedback: the learning element of peer assessment. *Teaching in Higher education*, 11(3):279–290.
- Luo, H., Robinson, A., and Park, J.-Y. (2014). Peer grading in a mooc: Reliability, validity, and perceived effects. *Online Learning Journal*, 18(2).
- McGrath, A. L., Taylor, A., and Pychyl, T. A. (2011). Writing helpful feedback: The influence of feedback type on students' perceptions and writing performance. *Canadian Journal for the Scholarship of Teaching and Learning*, 2(2):5.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Moens, M.-F., Boiy, E., Palau, R. M., and Reed, C. (2007). Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230.
- Mohammad, S. M., Sobhani, P., and Kiritchenko, S. (2017). Stance and sentiment in tweets. *ACM Transactions on Internet Technology (TOIT)*, 17(3):1–23.
- Murillo-Zamorano, L. R. and Montanero, M. (2018). Oral presentations in higher education: a comparison of the impact of peer and teacher feedback. *Assessment & Evaluation in Higher Education*, 43(1):138–150.
- Nelson, M. M. and Schunn, C. D. (2009). The nature of feedback: How different types of peer feedback affect writing performance. *Instructional Science*, 37(4):375–401.
- Nguyen, H., Xiong, W., and Litman, D. (2017). Iterative design and classroom evaluation of automated formative feedback for improving peer feedback localization. *International Journal of Artificial Intelligence in Education*, 27(3):582–622.
- Palau, R. M. and Moens, M.-F. (2009). Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107.
- Panadero, E. and Alqassab, M. (2019). An empirical review of anonymity effects in peer assessment, peer feedback, peer review, peer evaluation and peer grading. *Assessment & Evaluation in Higher Education*, 44(8):1253–1278.
- Paré, D. E. and Joordens, S. (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):526–540.
- Patchan, M. M. and Schunn, C. D. (2015). Understanding the benefits of providing peer feedback: how students respond to peers' texts of varying quality. *Instructional Science*, 43:591–614.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., and Koller, D. (2013). Tuned models of peer assessment in moocs. *arXiv preprint arXiv:1307.2579*.
- Prins, F. (2002). *Search & see: the roles of metacognitive skillfulness and intellectual ability during novice inductive learning in a complex computer-simulated environment*. Citeseer.
- Prins, F. J., Sluijsmans, D. M., Kirschner, P. A., and Strijbos, J.-W. (2005). Formative peer assessment in a cscl environment: A case study. *Assessment & Evaluation in Higher Education*, 30(4):417–444.
- Rada, R. et al. (1994). Collaborative hypermedia in a classroom setting. *Journal of Educational Multimedia and Hypermedia*, 3(1):21–36.
- Ramachandran, L., Gehringer, E. F., and 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(3):534–581.
- Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Reinholz, D. (2016). The assessment cycle: A model for learning through peer assessment. *Assessment & Evaluation in Higher Education*, 41(2):301–315.
- Rosenthal, S. and McKeown, K. (2015). I couldn't agree more: The role of conversational structure in agreement and disagreement detection in online discussions. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 168–177.
- Sadler, P. M. and Good, E. (2006). The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1):1–31.
- Sánchez-Prieto, J. C., Gamazo, A., Cruz-Benito, J., Therón, R., and García-Peñalvo, F. J. (2020). Ai-driven assessment of students: Current uses and research trends. In *International Conference on Human-Computer Interaction*, pages 292–302. Springer.
- Settles, B. (2012). Active learning. *Synthesis lectures on artificial intelligence and machine learning*, 6(1):1–114.
- Shah, D. and Pickard, L. (2017). Massive list of mooc providers around the world. *Where to find MOOCs*.
- Song, Y., Hu, Z., and Gehringer, E. F. (2015). Pluggable reputation systems for peer review: A web-service approach. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE.
- Sridharan, B., Tai, J., and Boud, D. (2019). Does the use of summative peer assessment in collaborative group work inhibit good judgement? *Higher Education*, 77:853–870.

- Staubitz, T., Traifeh, H., Chujfi, S., and Meinel, C. (2020). Have your tickets ready! impede free riding in large scale team assignments. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 349–352.
- Sun, N., Liu, D., Zhu, A., Chen, Y., and Yuan, Y. (2019). Do airbnb’s “superhosts” deserve the badge? an empirical study from china. *Asia Pacific Journal of Tourism Research*, 24(4):296–313.
- Topping, K. (1998). Peer assessment between students in colleges and universities. *Review of educational Research*, 68(3):249–276.
- Topping, K. J. (2009). Peer assessment. *Theory into practice*, 48(1):20–27.
- Topping, K. J. (2010). Peers as a source of formative assessment. *Handbook of formative assessment*, pages 61–74.
- Veenman, M. V. (2013). Assessing metacognitive skills in computerized learning environments. In *International handbook of metacognition and learning technologies*, pages 157–168. Springer.
- 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*, volume 32.
- Weaver, M. R. (2006). Do students value feedback? student perceptions of tutors’ written responses. *Assessment & Evaluation in Higher Education*, 31(3):379–394.
- Wind, D. K., Jørgensen, R. M., and Hansen, S. L. (2018). Peer feedback with peergrade. In *ICEL 2018 13th International Conference on e-Learning*, page 184. Academic Conferences and publishing limited.
- Xiao, Y., Zingle, G., Jia, Q., Shah, H. R., Zhang, Y., Li, T., Karovaliya, M., Zhao, W., Song, Y., Ji, J., et al. (2020). Detecting problem statements in peer assessments. In *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, pages 704–709.
- Yin, J., Narang, N., Thomas, P., and Paris, C. (2012). Unifying local and global agreement and disagreement classification in online debates. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 61–69.
- Yu, F.-Y. and 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(3):2167–2177.
- Zarkoob, H., d’Eon, G., Podina, L., and Leyton-Brown, K. (2023). Better peer grading through bayesian inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6137–6144.
- Zingle, G., Radhakrishnan, B., Xiao, Y., Gehringer, E., Xiao, Z., Pramudianto, F., Khurana, G., and Arnav, A. (2019). Detecting suggestions in peer assessments. *International Educational Data Mining Society*.