"Can we reach agreement?": A context- and semantic-based clustering approach with semi-supervised text-feature extraction for finding disagreement in peer-assessment formative feedback. *

M Parvez Rashid, Divyang Doshi, Sai Venkata Vinay, Qinjin Jia, Edward F. Gehringer
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
{mrashid4, ddoshi2, ssamudr, qjia3, efg}@ncsu.edu

ABSTRACT

In the process of review for assessing a piece of work, agreement or consensus among reviewers is vital to review quality. As classroom peer assessments are undertaken by naïve peers, disagreement among peer assessors can confuse the assessees and lead them to question the review process. Although there are methods like inter-rater reliability (IRR) to measure disagreement in summative feedback, in the authors' knowledge, there is no method for finding disagreements within formative feedback. It may take more time and effort for the instructor to review the feedback to find disagreements than it would to simply perform an expert review without involving peer assessors. An automated method can help locate disagreements among reviewers. In this work, we used a clustering algorithm and NLP techniques to find disagreement in formative feedback. As the review comments are related by context and semantics, we implemented a semi-supervised approach to fine-tune the SentenceTransformer model to capture the context and semantics-based relation among the review texts, which in turn improved the comment clustering performance.

Keywords

Peer-review, disagreement, NLP, SentenceTransformer, fine-tuning, clustering

1. INTRODUCTION

Peer review has long been an effective component of students' learning experience [15]. Previous studies showed that assessment by student peers could be as accurate as assessment by the instructor [14]. Not only do students learn from reviews they receive, but they learn even more from providing feedback [11, 2, 9, 5, 13]. To make the peer-assessment process more accurate and unbiased, each artifact is generally anonymized and reviewed separately by multiple reviewers [4]. Since peer reviewers assess fewer artifacts than the instructor, they can afford to spend more time on each [1]. However, peer reviewers do not always agree with each other's reviews. In Table 1, we have shown review comments on a piece of work where reviewers had incoherent opinions.

Though it is important for reviews to be coherent, to our knowledge, no classroom peer review process implements a meta-review round to find disagreements among the reviewers. One reason is that in the peer-review process, the number of reviews can be overwhelming for an instructor to meta-review, causing far more trouble than simply reviewing the artifacts themselves. For example, r reviewers review s students for c items, makes $r \times s \times c$ reviews to meta-review. An efficient way to identify disagreements is by implementing cutting-edge NLP methods to group the reviews expressing similar opinions together and locating the disagreements using a clustering algorithm. However, grouping the peers' comments using a clustering algorithm is not a straightforward task. Peer reviewers are often given a rubric [12] and in ideal cases, reviewers are expected to find the same issues in a piece of work, which makes the review texts semantically similar. Empirically we observed comments expressing disagreement might contain similar words and structure, or conversely, similar ideas may be expressed with completely different words. For example, in response to a rubric item, "If there 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 but clearly, the reviewers are in disagreement. It makes a difficult case for a state-of-the-art language model to distinguish the difference. The accuracy of a text clustering model depends on the feature vectors of the texts, i.e., similar texts should be represented as similar feature vectors [10]. SBERT is a current state-of-the-art sentence feature embedding model that is designed to be fine-tuneable for a downstream dataset.

*(Does NOT produce the permission block, copyright information nor page numbering). For use with edm_article.cls.

M. P. Rashid, D. Doshi, S. V. Vinay, Q. Jia, and E. F. Gehringer. "can we reach agreement?": A context- and semantic-based clustering approach with semi-supervised text-feature extraction for finding disagreement in peer-assessment formative feedback. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 497–501, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115725

Table 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, but lacks functionality and features.
		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.

In this study, we first compare the performance of four sentence-feature-embedding methods and pick the best method to fine-tune the model. We finally compare the clustering result using feature vectors of the pre-tarined and fine-tuned sentence embedding models. While implementing these approaches, we will address three research questions:

- **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?**

In this study, by "Disagreement" in peer assessors' feedback comments, we mean i) Comments that are opposing each other and ii) Comments that relate to disparate issues. Comments that agree partially with another comment are not considered in to be in disagreement.

2. RELATED WORK

Hiray et al. [7] showed that neural network models can be used to identify disagreement in online discussions. Instead of hand-crafted feature extraction they implemented a Siamese inspired neural network architecture to generate feature embedding of the texts. Guan et al. [6] discussed different text clustering approaches and found that clustering algorithms' performance depends on the quality of the feature vectors. Peer-review data in the educational environment is less available than product reviews or social-media text. The length of peer assessments is often similar in length to product reviews. Studying methods used to analyze short texts will give us some idea about analyzing peer-review texts. Jinarat et al. [8] identified that a major characteristics of short texts (e.g. facebook comments and post, tweeter text, news headline, product review etc.) is lack of context information and the presence of much jargon and abbreviations. These affect the performance of traditional text-clustering algorithms.

3. METHOD

This section describes the data collection process, dataset construction, and methods we used for the study.

3.1 Collecting Formative Feedback

We acquired data for this study from the Object Oriented Design and Development course at North Carolina State University for a period of three semesters (Spring 2021, Fall 2021, and Spring 2022). Before the review process started, students were shown examples of how to write quality feedback. The assignments were submitted and reviewed using the Expertiza system. We collected formative feedback comments from the assignment named "Program 2". The peer reviewers wrote the review comments in response to 201 different rubric items. All the reviewers' and reviewees' identities were anonymized before the feedback comments were collected for analysis, so the author of the assignment or the review comment could not be determined.

3.2 Creating the Datasets:

We prepared three datasets from the review comments we collected over the three semesters. The three datasets are as follows:

1. **Sentence-Embedding-Test Dataset:** This dataset consists of 3,000 annotated pairs of review comments. The comment pair is annotated with "1" if the two review comments express a similar idea (agreement) and "0 " if they express a different idea (disagreement). These annotations were done by five experts who are familiar with the Program 2 assignment, including its rubric and review comments. Table 2 shows an example of the dataset.
2. **Fine-Tuning Dataset:** This dataset consists of 11,000 pairs of review comments. They are not initially annotated. We used this dataset for the semi-supervised approach to train the model. We annotated 1,600 pairs during the fine-tuning phase of the sentence-embedding generator model.
3. **Clustering-Test Dataset:** This dataset consists of 1,000 review comments for measuring clustering-algorithm performance quality. In this dataset we grouped all the review comments following the same rubric item on the same piece of work.

3.3 Sentence Embedding

Sentence embeddings are a way of representing different-length sentences with fixed-size vectors of numbers. In this study, we compared the performance of four sentence-embedding methods using accuracy on the Sentence-Embedding-Test Dataset.

Global Vectors (GloVE) is a word vectorization technique used to convert natural language text to feature vectors that are suitable for machine-learning models to process. GloVe incorporates the local statistics of a word in a sentence as well as the global occurrence of the word in the document.

Pre-trained Bidirectional Encoder Representations from Transformers (BERT) model produces word embeddings that has

Table 2: Table shows a sample of Sentence-Embedding-Test dataset with paired comments, labeled for agreement (Label "1") and disagreement (Label "0") in two comments. Each pair of comments is on the same piece of work following the same rubric item

	Comment1	Comment2	Label
1	Application to properties should be when reviewing a property, and the application deployment is crashing hence not able to actibely test	All the required fields of student are enforced to be non-null.	0
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	Yes, validations seem to be enforced	All fields were appropriately validated.	1

shown great success in finding contextual and semantic relations among words in a sentence. It is a multi-layer bidirectional model based on the encoder mechanism of the transformer model. BERT learns the contextual relation of each word by considering the other words in both directions in a sentence. We can get embeddings from BERT by using a mean-pooling method that averages the feature vectors of each word or by the [CLS] [3] token available at the first position of the BERT sentence embedding output.

Sentence-BERT (SBERT) utilizes a Siamese Neural Network, where the neural network consists of two identical subnetworks. The identical subnetworks have the same parameters and weights. The parameter updating is also mirrored in both sub-networks. This model produces sentence embedding in a way that the semantically similar sentences have a very high cosine similarity. Unlike BERT, the Siamese network does not require every possible pair combination to find semantic similarity in sentences. As a result, the computation time is reduced from $O(n^2)$ to $O(n)$.

3.4 Active Learning

For this study, we used a semi-supervised approach known as active learning to fine-tune the SBERT model. The key idea for an active-learning algorithm is that a machine-learning model can run faster and with less labeled data if it can choose the data from which the model needs to learn. During the iterative process of training models, an expert need to annotate only the samples the model is uncertain of, and the model can be trained with these annotations. Continuing this approach iteratively helps the model learn faster with few annotated samples.

3.5 Choosing the cosine similarity cut-off

We used the approach implemented by the SBERT authors for choosing a cosine-similarity threshold [3]. This algorithm picks the threshold that makes the most accurate prediction for classifying both similar sentences and dissimilar sentence pairs on the test dataset.

3.6 Clustering Algorithm

We chose the agglomerative clustering algorithm for our study for the grouping task, as it does not require deciding the number of clusters beforehand. This algorithm initially assigns each sentence (embedding vector) to its cluster and afterward repeatedly merges pairs of clusters until all the clusters merge into a single cluster and form an agglomerative tree.

3.7 Evaluation Metrics

For comparing the sentence-embedding generator model's performance on the Sentence-Embedding-Test dataset, we used accuracy as a metric. For measuring the clustering performance, we used the silhouette coefficient.

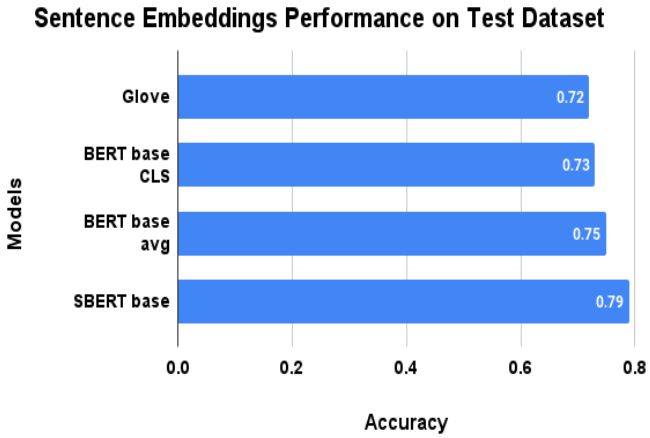

Figure 1: Comparison of Sentence Embedding Approaches using Accuracy Score on the Sentence-Embedding-Test Dataset

4. RESULTS AND DISCUSSION

This section presents the experimental results and discusses the findings of the research questions mentioned in section 1.

RQ1: Which pre-trained feature extraction methods for context and semantically related sentences work best?
We used accuracy as a metric to compare the performance of different sentence feature extraction methods on the Sentence-Embedding-Test dataset.

- Sentence pairs were identified as agreement or disagreement from GloVe feature embeddings with an accuracy score of 0.72. The classification accuracy score was 0.73 using feature embeddings from the BERT model's [CLS] token and 0.75 using the mean-pooling (average) method of BERT word embeddings. The base SBERT model had an accuracy score of 0.79. (Figure: 1)
- Considering the accuracy scores, base SBERT feature extraction performed 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?

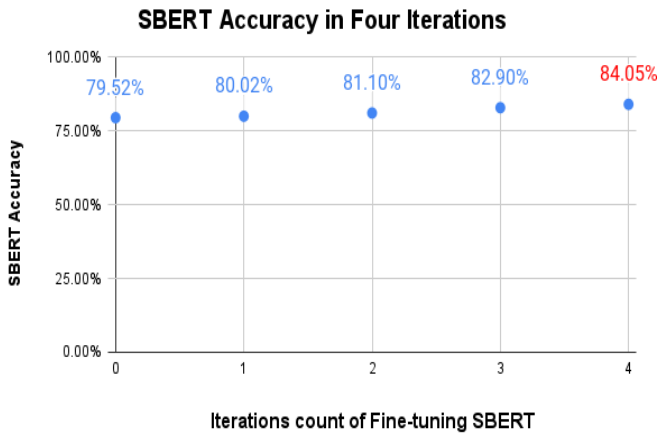

Figure 2: Fine-tuning of SBERT increased accuracy for identifying sentence similarity or difference after each iteration

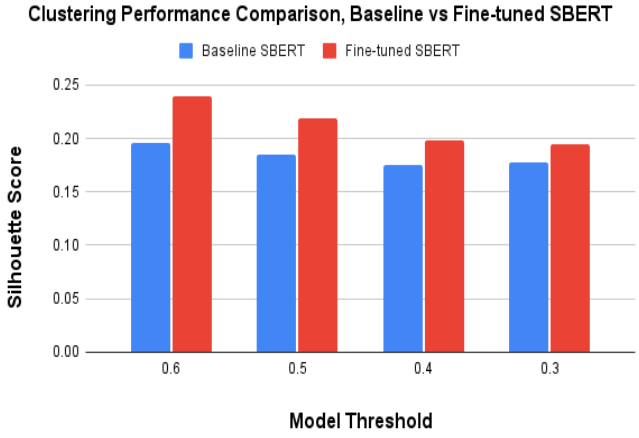

Figure 3: The clustering performance comparison using Silhouette Score with different thresholds for Agglomerative Hierarchical Clustering shows that Fine-tuned SBERT using Active Learning outperformed at every threshold point

Based on the accuracy scores for classifying the review-comment pair as agreeing or disagreeing, we picked the baseline Pre-trained SBERT model for further fine-tuning. We used the Fine-Tuning dataset and active-learning approach to fine-tune the SBERT model. We compared the model's performance using accuracy scores on the test Sentence-Embedding-Test dataset. Fine-tuning using active learning is an iterative method, so we continued the fine-tuning for four iterations. The result showed that the fine-tuned SBERT model improved accuracy after every iteration (Figure 2).

RQ3: Does improving the sentence feature extraction method improve clustering performance?
To test clustering performance, we used the Clustering-Test dataset. For each rubric item, this dataset has 2–5 review comments for each piece of work. We measure the clustering performance of both the baseline SBERT and fine-tuned SBERT using the silhouette score. For every clustering threshold we experimented with, the silhouette score for fine-tuned SBERT was higher than for the baseline SBERT model (Figure: 3).

5. CONCLUSION

In this study, we aim to identify disagreements in peer assessors' formative feedback by implementing a clustering algorithm. Our hypothesis is that reviews expressing similar feedback on a piece of work will be contextually and semantically similar, and that a clustering algorithm will be able to identify the similarity and put the similar feedback in a single group or cluster. On the other hand, feedback that expresses different opinions will be identified by the clustering algorithm and should be separated from other feedback. We showed that the performance of the clustering algorithm depends on the quality of the feature vectors that express the reviewers' natural language as machine-readable numbers. We have experimented with several baseline feature-vector extraction methods and fine-tuned SBERT sentence-embedding methods to compare quality. We carefully constructed the datasets for our experiments from reviews in a course that implemented the peer-assessment process. For fine-tuning the SBERT model, we implemented a semi-supervised active learning approach using uncertainty sampling and expert annotation. Our study showed that the fine-tuned SBERT sentence-embedding model outperformed the baseline SBERT model on our test dataset. Finally, we used the base-case model and the fine-tuned model's sentence embedding with the agglomerative clustering algorithm. We experimented with different thresholds and compared our results using silhouette scores. The silhouette score and empirical study of the clusters formed by the fine-tuned model show that the clustering algorithm can identify disagreements in peer-reviewers' formative feedback.

The key findings of this study are that base SBERT model outperforms other feature-extraction methods like Glove and BERT on the task of finding semantic review similarities on a peer-review dataset containing a high amount of software jargon. Also, we show that fine-tuning SBERT on this context-specific data further improves the model accuracy. We also show that fine-tuning improves the clustering done on the peer-review data to find disagreement in the review comments.

Since disagreement among reviewers can confuse students and lead them to question the review process, finding disagreements can help resolve the confusion by engaging reviewers in discussion and suggesting that the instructor intervene. In the future, we intend to extend this work to implement a recommendation system for reviewers to consider revising their feedback based on key points that other reviewers have identified.

6. REFERENCES

[1] J. Cambre, S. Klemmer, and C. Kulkarni. 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, 2018.

[2] Y. D. Çevik. 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, 2015.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] E. F. Gehringer. A survey of methods for improving review quality. In *International Conference on Web-Based Learning*, pages 92–97. Springer, 2014.

[5] M. H. Graner. Revision workshops: An alternative to peer editing groups. *The English Journal*, 76(3):40–45, 1987.

[6] R. Guan, H. Zhang, Y. Liang, F. Giunchiglia, L. Huang, and X. Feng. Deep feature-based text clustering and its explanation. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[7] S. Hiray and V. Duppada. 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, 2017.

[8] S. Jinarat, B. Manaskasemsak, and A. Rungsawang. 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, 2018.

[9] L. Li and V. Grion. The power of giving feedback and receiving feedback in peer assessment. *All Ireland Journal of Higher Education*, 11(2), 2019.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[11] R. Rada et al. Collaborative hypermedia in a classroom setting. *Journal of Educational Multimedia and Hypermedia*, 3(1):21–36, 1994.

[12] M. P. Rashid, E. F. Gehringer, M. Young, D. Doshi, Q. Jia, and Y. Xiao. Peer assessment rubric analyzer: An nlp approach to analyzing rubric items for better peer-review. In *2021 19th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–9. IEEE, 2021.

[13] M. P. Rashid, Y. Xiao, and E. F. Gehringer. Going beyond" good job": Analyzing helpful feedback from the student's perspective. *International Educational Data Mining Society*, 2022.

[14] P. M. Sadler and E. Good. The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1):1–31, 2006.

[15] K. Topping. Peer assessment between students in colleges and universities. *Review of educational Research*, 68(3):249–276, 1998.