Improving problem detection in peer assessment through pseudo-labeling using semi-supervised learning

Chengyuan Liu
North Carolina State University
cliu32@ncsu.edu

Jialin Cui
North Carolina State University
jcui9@ncsu.edu

Ruixuan Shang
North Carolina State University
rshang@ncsu.edu

Yunkai Xiao
North Carolina State University
yxiao28@ncsu.edu

Qinjin Jia
North Carolina State University
qjia3@ncsu.edu

Edward Gehringer
North Carolina State University
efg@ncsu.edu

ABSTRACT

An online peer-assessment system typically allows students to give textual feedback to their peers, with the goal of helping the peers improve their work. The amount of help that students receive is highly dependent on the quality of the reviews. Previous studies have investigated using machine learning to detect characteristics of reviews (e.g., Do they mention a problem, make a suggestion, or tell the student where to make a change?). Machine-learning approaches to peer-assessment evaluation are heavily reliant on labeled data to learn how to identify review characteristics. However, attaining reliable labels for those characteristics is always time-consuming and labor-intensive. In this study, we propose to apply pseudo-labeling, a semi-supervised learning-based strategy, to improve the recognition of reviews that detect problems in the reviewed work. This is done by utilizing a small, reliably labeled dataset along with a large unlabeled dataset to train a text classifier. The ultimate goal of this research is to show that for peer assessment evaluation, we can utilize both unlabeled and labeled datasets to obtain a robust auto-labeling system and thereby save much effort in labeling the data.

Keywords

Peer assessment, Problem detection, Natural language processing, Semi-supervised learning, Pseudo labeling

1. INTRODUCTION

Peer assessment has long been used as a pedagogical technique in project-based courses [10, 11, 13, 14]. An online system typically allows students to provide numerical scores and give textual feedback on other teams' work. It has been shown to be remarkably effective in improving students' learning and teaming skills [10]. Peer assessment can also help instructors evaluate student work and assign grades to it. Double et al. [3] presented a meta-analysis suggests that peer assessment improves academic performance even more than teacher assessment. However, the reliability and validity of peer assessment are completely determined by peer-review quality [15]. High-quality reviews can help authors precisely identify issues with their work and make corresponding revisions. Low-quality reviews could be unhelpful or even detrimental to students' learning.

Hence, there is a growing interest in evaluating the review quality in peer assessment research [12]. However, having the instructors or TAs evaluate or grade all peer-review comments would be extremely time-consuming. Consequently, several studies have investigated machine-based automated review evaluation with the help of natural language processing techniques as well as machine learning, Nelson et al. [12] carried out a pioneering study on identifying high-quality reviews by investigating the features in the review text and determining what type of comments are most helpful and why. Xiao et al. proposed a machine-learning NLP-based approach for finding problem statements [18] (e.g., Do they mention any problems in the work that required revisions) and suggestions (e.g., Do they provide any suggested solutions on how to revise the work) [[23]] in the peer review comments.

As with most AI tasks, the biggest challenge for applying machine learning and deep learning algorithms to peer assessment is collecting labeled data [18]. Identifying whether review comments contain problem statements and suggestions is sometimes subjective, so the same review will be labeled differently by different students. This creates a major obstacle in collecting precise and reliable labels for the text analysis. Researchers have suggested approaches to tackle this problem of unreliable and insufficient labeling. Jia et al. [8] proposed an annotation process by two graduate students and measured the inter-annotator agreement between them to improve labeling validity. Xiao et al. [18] proposed to apply transfer learning and active learning to tackle the insufficient-labeling problem by using knowledge from a related task that has already been learned. All previous approaches required intervention from either human effort or out-of-domain knowledge; there is not a single study on peer assessment evaluation that has examined how to train a robust classifier on the data alone, and in particular how to

C. Liu, J. Cui, R. Shang, Y. Xiao, Q. Jia, and E. Gehringer. Improving problem detection in peer assessment through pseudo-labeling using semi-supervised learning. In A. Mitrovic and N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 391–397, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.6853127

make good use of unlabeled dataset, which is much easier to collect. Semi-supervised learning has proven to be a effective approach to address these issues [6].

Semi-supervised learning is a learning paradigm that stands between unsupervised and supervised learning [6]. The goal of semi-supervised classification is to train a classifier that uses both a small labeled dataset and a large unlabeled dataset to outperform the traditional supervised classifier trained only on the labeled data. Basic approaches to semi-supervised learning involve a well-known technique called pseudo-labeling [9], in which method a classification model is first trained on the labeled dataset and then used to infer pseudo-labels on the unlabeled dataset. Then the unlabeled dataset with pseudo-labels is combined with the labeled dataset, so that the predicted labels are used as ground truth. This allows us to essentially scale up the labeled dataset in order to train a more robust model.

Xie et al. [19] conducted an extensive study on pseudo-labeling and presented a self-training method with "student" and "teacher" models on image classification tasks (note that "student" and "teacher" here are not referring to the user of the model), which achieved an outstanding result. The idea is almost the same as the pseudo-labeling approach. Initially a "teacher" model is trained on the labeled dataset and used to predict pseudo-labels on the unlabeled dataset, then a "student" model will be trained on the combination of the labeled and the pseudo-labeled dataset, these steps will be run iteratively. In each iteration, once the "student" model is trained, it will be used as the "teacher" model to generate predictions in the next iteration. This has proven to be a promising approach by creating more labeled data to address the data-insufficiency issue. Pseudo-labeling and self-training strategies have been widely applied in computer vision tasks [5, 19]. However, very few studies have attempted this approach in natural language processing tasks. This paper aims to apply natural language processing techniques and text-classification models to investigate the validity of applying pseudo-labeling to improve the performance of detecting characteristics in the peer review comments.

The main pedagogical contribution of this study to the peer assessment evaluation is to show how to deploy our student taggers (people who labeling the review data) more effectively and eventually build an auto-labeling system. More specifically, in our peer assessment system, the labels can only be collected from student taggers, and, if they are requested to label numerous comments, they may potentially become careless, resulting in the poor labeling quality. We could deploy them more successfully by applying the pseudo-labeling approach to build a text classifier for evaluating peer reviews with considerably less labeled data.

2. METHODOLOGY

2.1 Automated peer-review quality evaluation

Although peer review is a widely accepted approach in the educational setting, the effectiveness of peer review in promoting students' learning can vary significantly. Most research has investigated the overall pedagogical contribution of peer review of writing. However, research on evaluating review quality is particularly lacking.

Nelson et al. [12] demonstrated that high-quality review has proven to be a great benefit in improving students' learning. Their paper proposed an approach to determining what type of feedback is most helpful and why it is helpful to students' writing performance. They also listed the features for identifying high-quality peer reviews. This study laid an excellent foundation for later research projects on automatically detecting characteristics of peer review comments.

The earliest study on automated peer-review quality evaluation was conducted by Cho et al. [1]. This study proposed a machine-learning algorithm to evaluate peer reviews collected from SWoRD—a web-based reciprocal peer-review system. The review data was encoded for multiple characteristics such as problem detection, solution suggestion, etc., and then several traditional machine learning algorithms (Naive Bayes, SVM, and Decision Tree) were applied on the text-classification task to evaluate quality.

Subsequently, automated evaluation became increasingly fashionable in peer assessment. Xiong et al. applied supervised machine learning to automatically identify problem localization (pinpoint the location of where the problem is) [20, 22] and helpfulness [21] in peer review comments using NLP techniques. Zingle et al. [23] describe a method for automatically detecting suggestions in the review text, Xiao et al. [17] proposed to auto-detect problem statement in review comments.

Our study introduces an intriguing approach for automatically assessing review quality by detecting problem statements in the comments and applying a semi-supervised learning approach to address the problem of labeled-data insufficiency. Our goal is to help students get instant and accurate feedback on the reviews they write and enable them to improve their reviewing. This approach can also significantly reduce the workload of student taggers who label those characteristics in peer-review comments.

2.2 Semi-supervised Learning & Pseudo-labeling

Deep learning has achieved great success in the area of artificial intelligence; however, most of the state-of-the-art (SotA) models were trained using supervision, which required a large labeled dataset to attain excellent performance [6]. In most cases, labeling was a difficult and time-consuming task; even if we devote the time to do this, we would still be ignoring potential insights from the unlabeled dataset, which is far easier to collect in the real world. Semi-supervised learning has shown promise from using both labeled unlabeled data. The objective of semi-supervised learning is to improve learning behavior by combining labeled and unlabeled data, or equivalently, to achieve the same model performance with a relatively small labeled dataset.

Pseudo-labeling is one of the most effective and efficient methods in semi-supervised learning [9]. With pseudo-labeling, the initial model is trained on the labeled dataset:

$$D_L = \left\{(x^i, y^i)\right\}_{i=1}^{N_L} \tag{1}$$

, where x^i represents each input, $y^i \subseteq \{0, 1\}$ is the corresponding labels where 0 represents "does not include prob-

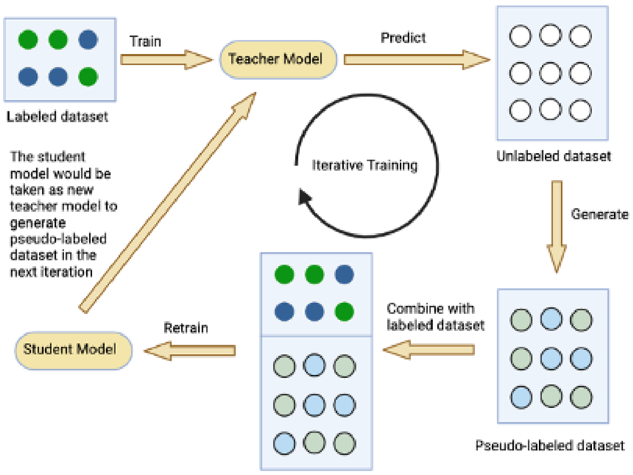
Train
Teacher Model
Predict
Labeled dataset
Unlabeled dataset
Iterative Training
The student model would be taken as new teacher model to generate pseudo-labeled dataset in the next iteration
Generate
Combine with labeled dataset
Retrain
Student Model
Pseudo-labeled dataset

Figure 1: Self-training workflow

lem statement" and 1 represents "does include problem statement", N_L is the size of the labeled dataset. There is also an unlabeled dataset:

$$D_U = \left\{(x^i)\right\}_{i=1}^{N_U} \quad (2)$$

where x^i represents each input without labels, and N_U is the size of the unlabeled dataset. In most cases $N_U \gg N_L$, so we believe that the unlabeled dataset may potentially contain more valuable features than the labeled set. Next, the model trained in the initial step generates predictions $\tilde{y}^i$ as pseudo-labels on the unlabeled set D_U; hence we can construct a pseudo-labeled set:

$$D_U = \left\{(x^i, \tilde{y}^i)\right\}_{i=1}^{N_U} \quad (3)$$

where $\tilde{y}^i$ will be used as the ground-truth label y^i to compute the loss in back-propagation in the next training phase, after this, another model is trained on the combination dataset:

$$D_C = D_L + D_U \quad (4)$$

and we believe that with more training data, the model will become still better.

Pseudo-labeling is often performed through an iterative process rather than one-step label generation. Self-training [19, 5] can be interpreted as an iterative pseudo-labeling approach. Defining the model used to generate labels as "teachers" and the model trained on both pseudo-labeled and labeled dataset as "students", the two roles will be swapped in each iteration after incorporating more pseudo-labeled data into the labeled dataset (as shown in Figure 1). In this way, we can achieve our initial goal of improving model performance with the help of the valuable unlabeled dataset without human intervention or external knowledge.

3. EXPERIMENT

3.1 Datasource

The dataset we used in this paper comes from [Redacted] [4], a web-based peer review system that allows students to provide both numerical ratings and textual feedback to other groups' assignments. Each student must review multiple assignments and provide appropriate peer assessments to earn credits. The scores assigned by peer reviewers help the instructor or TAs to give a final grade to the assignments. The textual feedback also helps the authors to make revisions. Students have the chance to earn extra credit by labeling the review comments they received from the peer reviewers, and these labels support the construction of text classifiers for peer-review evaluation as ground-truth labels.

Student taggers label the review comments for whether or not they contain characteristics such as: problem statements, suggestions, and explanations. In this study, we only use the problem statement label. The quality of these labels cannot be fully guaranteed, but the success of training a robust model depends heavily on the quality of the ground-truth labels. We propose a data-filtering approach to select the review comments with the most reliable labels. When students use the [Redacted] system, up to four students on a team will label the same review comments they received on their team's work. We will not select a review comment for the dataset if any team members disagree on the label (this will be defined as "taggers agreement rule" in the following section). Initially, 48,412 review comments with the corresponding labels were pulled from [Redacted] from the Fall 2017 to Fall 2020 semesters of a masters-level object oriented design class. After the raw data was filtered by the taggers agreement rule, 3100 pieces of "high-quality" labeled data were collected. Since our goal is to investigate the effectiveness of the unlabeled set, only a small labeled subset is required to train the initial model. Because of that, we extract 1600 review comments as the training set and 1500 as the validation set. The remaining 45,312 review comments that do not follow the taggers agreement rule will have their labels stripped and used as the unlabeled dataset.

Another motivation for pseudo-labeling is to compare the effectiveness of this strategy on different sizes of labeled datasets. If the amount of labeled data required can be reduced without harming the model performance, our approach can have a great impact on peer-assessment evaluation. For this paper, we conducted multiple experiments with different sizes of labeled sets. We will report only which size brings the most improvement after applying our strategy, rather than comparing model performance between different-sized datasets, as it is an indisputable fact that more labeled data will produce a better result.

3.2 Model implementation

Comparing the performance of different deep-learning models was not a goal of this study; hence we will only select one language-classification model to train both the teacher and student models. We use the transformer-based language model known as Bidirectional Encoder Representations from Transformers (BERT), which was first introduced by Google in 2019 [2]. Transformers apply a specific self-attention mechanism, which is designed for language understanding [16]. Self-attention emphasizes which part in an input sentence is crucial to the understanding. The transformer is an encoder-decoder-based architecture consisting of a standard feed-forward layer and a special attention layer, as shown in Figure 2 [16].

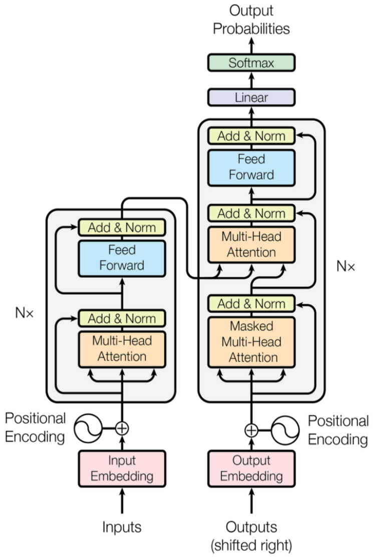

Figure 2: The Transformer - model architecture [16]

The traditional language model reads the input sentence in a single direction, either left to right, or right to left, which is enough for the task of next-word prediction. However, for a deep understanding of the sentence, the context is necessary. For a given word, considering both the previous and next token is valuable in learning the text representations, which is why the Bert model can achieve such superior performance on language-understanding tasks.

The Bert model consists of several layers of transformer blocks; the base model has 12 layers with 110 million parameters. By comparison, the large model has 24 layers with 340 million parameters [16]. Note that the Bert model only uses the encoder part of the transformer, which is responsible for reading the input text and processing it. In this study, we will only apply the Bert base model, considering the training efficiency.

The Bert model is trained in two phases, pre-training and fine-tuning. Pre-training includes two NLP tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), using 3.3 billion words from Wikipedia and BooksCorpus; note that all data is unlabeled. Then the pre-trained model is used for the downstream NLP tasks in the fine-tuning phase, like text classification. In our study, we simply used the pre-trained Bert base model, then fine-tuned the model by feeding our peer-review data to carry out the text-classification task.

3.3 Pseudo-labeling setting

As mentioned in Section 2.2, we initially trained a teacher model only on the labeled dataset. We aimed to assess the improvement achieved by our strategy with three different sizes of the labeled set. Accordingly, 400, 800, and 1600 labeled reviews were randomly selected from the training set. These samples were used to train the initial teacher model and later combined with pseudo-labeling data to train the student model in the self-training loop.

After the pseudo-labels are generated on the unlabeled dataset, another attractive experiment is to investigate whether we should use the entire pseudo-labeled dataset, or just a part of it, to train the student model. There are three common approaches for selecting the pseudo-labeled subset. We define them as—

- **Full selection**: Combine the entire pseudo-labeled set with the labeled set to train the student model.
- **Random selection**: Randomly select a subset from the pseudo-labeled set and combine it with the labeled set; the labels of the remaining samples will be stripped, and those samples will be considered as the unlabeled dataset in the next iteration.
- **Top-k% selection**: Follow almost the same steps as random selection, except for the sampling method, as shown in Figure 3, the teacher model will retain the prediction score while generating the pseudo-labels, and then only the samples with the k% highest prediction scores will be selected.

For this paper, we use the Top-k% selection method to create the subset of the pseudo-labeled dataset in each iteration. This proved to be an effective way to address the confirmation-bias issue (Section 3.4), below. In our study, k = 100% (same as full selection) was chosen as a baseline, and the k = 10, 20, 40% were selected for the experiment.

As previously mentioned, pseudo-labeling is implemented as an iterative process so top-k% selection will be applied in each iteration. Once a pseudo-labeled subset has been selected, the remaining pseudo-labeled data is used as the unlabeled set and new predictions are generated from the teacher model in the next iteration. We ran this process 10 times (epochs = 10) and for consistency, the entire pseudo-labeled dataset will be fed into the model in the last iteration.

3.4 Handling confirmation bias

Machine-learning models predict incorrect labels when they are unable to learn enough patterns from the data. In pseudo-labeling, overfitting the student model to these incorrect labels predicted by the naïve teacher model is defined as confirmation bias. This leads to a significant impairment of the pseudo-labeling strategy. Initially, the teacher model could well be affected by noise, especially with very little labeled data being trained. Although this cannot be avoided fundamentally, there are still some approaches that can help reduce the effects of confirmation bias.

3.4.1 Top-k% selection

As mentioned in section 3.3, random selection would potentially perform better than full selection as it can alleviate the negative impact of bias, since only a subset of the pseudo-labeled data will be fed into the model in each iteration. In this way, relatively less bias will be introduced into the model.

Instead of randomly selecting the subset, the top-k% selection method is based on the prediction scores generated

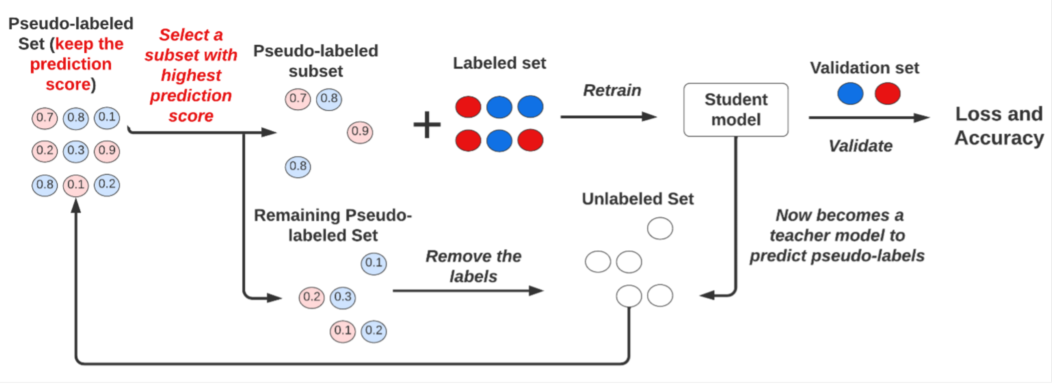

Figure 3: Top-k% selection workflow

by teacher models. Only the data points with the highest predicted probability score will be selected and included into the labeled set. The theoretical justification for this is similar to entropy regularization [7], which is another semi-supervised learning technique that encourages the classifier to infer confident predictions on the unlabeled dataset. For example, we would prefer to assign the unlabeled data a high probability of belonging to a particular class, rather than diffuse probabilities across different classes. However, this confidence-based approach must assume that the data are clustered according to class, which means that neighboring data points should have the same class, while the points in different classes should be widely separated.

3.4.2 Weighted loss

Another approach to handlling confirmation bias is to redefine the cross-entropy loss as a weighted summation between the labeled and pseudo-labeled set. Initially, the naïve teacher model is incapable of generating reliable pseudo-labels. If we simply add the unlabeled loss to the labeled loss, especially when the size of unlabeled dataset is much larger, the model tends to overfit on the unreliable pseudo-labeled data and consequently generate wrong predictions.

Therefore Lee et al. [9] proposed to use weight in the loss function. The overall loss function looks like this:

$$L = \frac{1}{n} \sum_{m=1}^{n} \sum_{i=1}^{C} L\left(y_i^m, f_i^m\right) + \alpha\left(t\right) \frac{1}{n'} \sum_{m=1}^{n'} \sum_{i=1}^{C} L\left(y_i^{\prime m}, f_i^{\prime m}\right) \tag{5}$$

In simple terms, the equation can be interpreted as follows: Loss per Batch = Labeled Loss + Weight × Unlabeled Loss In this equation, the weight (alpha value) is used to control the contribution of the unlabeled loss to the total loss. The value is initialized small and slowly increases during the model training. Since few training epochs are needed to fine-tune the BERT model for text-classification tasks, it does not seem necessary to define the alpha as a function of time. Therefore, in this study, we simply initialize the alpha value to be 0.1 and increase it by 0.1 in each epoch.

Considering the obstacle of calculating the loss after combining the labeled and pseudo-labeled sets, we designed the experiment as training on the pseudo-labeled set for each epoch and calibrating on the labeled set every three epochs. The alpha value is multiplied by the pseudo-labeled loss in each epoch and increases during the training iterations, while the labeled loss will remain the same.

	Accuracy			F1 score		
	Initial	Final	Imp	Initial	Final	Imp
Training with 400 labeled data						
Baseline	85.3%	84.7%	-0.6%	83.4%	82.1%	-1.3%
k=10%	85.3%	91.8%	**6.5%**	83.4%	90.6%	**7.2%**
k=20%	85.3%	90.5%	5.2%	83.4%	87.7%	4.3%
k=40%	85.3%	90.2%	4.9%	83.4%	88.1%	4.7%
Training with 800 labeled data						
Baseline	88.9%	88.0%	-0.9%	86.2%	84.5%	-1.7%
k=10%	88.9%	92.8%	**3.9%**	86.2%	92.1%	**5.9%**
k=20%	88.9%	91.9%	3%	86.2%	90.6%	4.4%
k=40%	88.9%	91.5%	2.6%	86.2%	89.6%	3.4%
Training with 1600 labeled data						
Baseline	89.8%	89.3%	-0.5%	87.2%	86.2%	-1%
k=10%	89.8%	92.6%	**2.8%**	87.2%	90.3%	**3.1%**
k=20%	89.8%	92.1%	2.3%	87.2%	89.7%	2.5%
k=40%	89.8%	91.7%	1.9%	87.2%	88.8%	1.6%

Table 1: The improvement of accuracy and F1 score

4. RESULTS

Figure 4(a) displays the learning curve of validation accuracy and Figure 4(b) displays the F1 score, with different sizes of labeled data over 10 epochs. The performance of different values for k are compared in each plot. Table 1 demonstrates the measurement before and after applying the pseudo-labeling method. Our goal is to compare the improvement in each experiment setting to assess the effectiveness of the Top-k% selection approach and the impact of the labeled data size.

RQ1: Does the pseudo-labeling improve the model performance?

We can see from Figure 4 that in general the learning curve is continuously rising with each experiment setting, and all settings achieved a significant improvement in the last epoch, except where $k = 100\%$; we will analyze this in the following section. These results undoubtedly show that by applying the pseudo-labeling approach, we can obtain a robust classifier using a large unlabeled dataset.

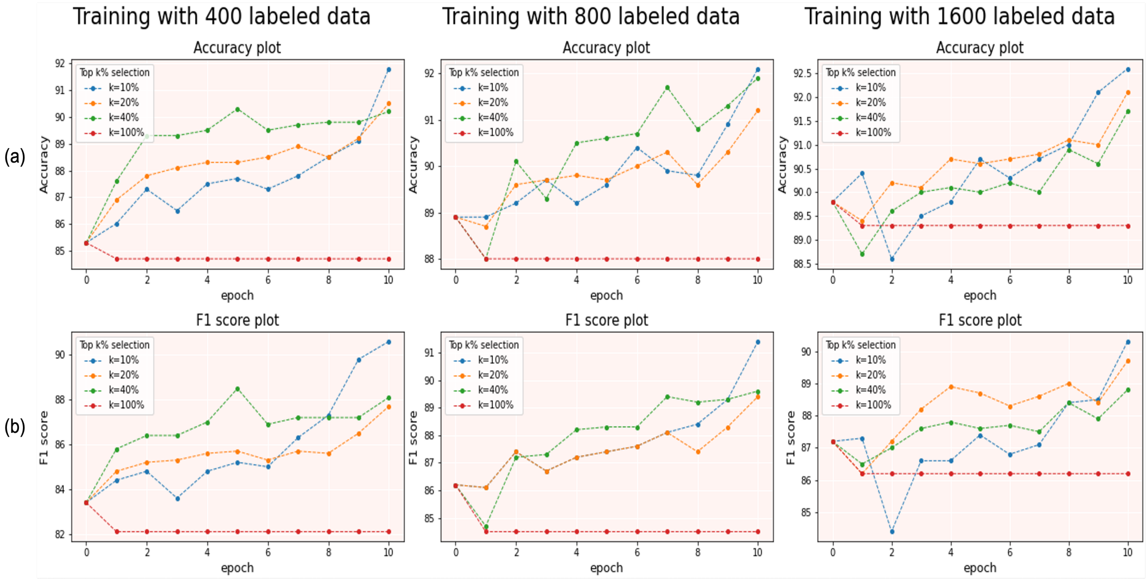

Figure 4: Validation accuracy and F1 score for different size of labeled subset

RQ2: Does the top-$k\%$ selection approach help reduce confirmation bias?
We use k=100%, with the entire pseudo-labeled dataset selected as the baseline, to assess the effectiveness of the top-$k\%$ selection approach. As shown in Figure 4, the learning curve of k=100% for each setting does not change over epochs and there is a notable drop in the early epochs, indicating that the pseudo-labeling does introduce bias into the model training in the absence of top-$k\%$ selection.

Both Figure 4 and Table 1 illustrate that the improvements are slightly different with a different k value; $k = 10\%$ yields the best result regardless of the size of the labeled dataset. Table 1 shows that on an average we are able to achieve 4.4% improvement in accuracy and 5.4% improvement in F1 score with $k = 10\%$, which is higher than with the other k values. This result clarifies that by selecting the pseudo-labeled subset based on the prediction score, we can obtain high-confidence predictions as reliable labels. The result also supports the cluster assumption in our case, which means that the review comments containing the problem statement are distinguishable from the comments not containing it.

RQ3: Does the pseudo-labeling work better on a small labeled set or large labeled set?
From Table 1 we can clearly see that regardless of the k value, the overall improvement on the small labeled set is comparatively higher than on the large labeled set, which indicates that the pseudo-labeling strategy works better on the small labeled set. This also implies that the unlabeled dataset can be more valuable than labeled set in certain practical problems, and using the unlabeled set can significantly improve the learning accuracy.

5. CONCLUSIONS

This paper presents a semi-supervised learning approach based on pseudo-labeling for evaluating peer-assessment quality. We investigated the effectiveness of the pseudo-labeling technique for different sizes of the labeled set. The results indicate that our approach can achieve an outstanding result with a small labeled dataset by augmenting it with an unlabeled set. The main contribution of this study to the peer-review process is the fact that not much labeled data is required to detect problem statements in peer-assessment comments; our student taggers do not have to label so much data. With less labeled data required, student taggers can be more careful to assign correct labels. In addition, we can find some better filtering approach to extract the smaller "high-quality" labeled data, which can greatly facilitate building our automatic labeling system.

Although we achieved a good result by using the top-$k\%$ selection approach as well as the weighted loss function to handle confirmation bias, the same success is not guaranteed on other tasks: confidence-based selection approaches are not always applicable; they will not work well without the cluster assumption.

The results of this study point the way to more efficiently analyzing review comments. Our pseudo-labeling approach can easily calculate how much labeled data for each characteristic is required for training a robust text classifier. For example, given a 91.8% classification accuracy in problem detection achieved with only 400 labeled data (a 6.5% improvement from supervised training alone), we can save a lot of labeling effort—effort that can then be devoted to identifying other salient review characteristics. Further research can explore better filtering approaches (similar to the tagger-agreement rule) for extracting small quantities of higher-quality labeled data in order to build a more reliable auto-labeling system.

6. REFERENCES

[1] K. Cho. Machine classification of peer comments in physics. *Educational Data Mining 2008 - 1st International Conference on Educational Data Mining,*

Proceedings, pages 192–196, 2008.

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186, 2019.

[3] K. S. Double, J. A. McGrane, and T. N. Hopfenbeck. The Impact of Peer Assessment on Academic Performance: A Meta-analysis of Control Group Studies. *Educational Psychology Review*, 32(2):481–509, 2020.

[4] E. Gehringer, L. Ehresman, S. G. GConger, and P. Wagle. Reusable Learning Objects Through Peer Review: The Expertiza Approach. *innovate Journal of On-Line education*, 3(5), 2007.

[5] S. Ghosh, S. Kumar, J. Verma, and A. Kumar. Self Training with Ensemble of Teacher Models. 2021.

[6] X. Goldberg. *Introduction to semi-supervised learning*, volume 6. 2009.

[7] Y. Grandvalet and Y. Bengio. entropy minimization: Semi-supervised Learning by Entropy Minimization. 2002.

[8] Q. Jia, J. Cui, Y. Xiao, C. Liu, P. Rashid, and E. Gehringer. ALL-IN-ONE: Multi-Task Learning BERT models for Evaluating Peer Assessments.

[9] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop: Challenges in Representation Learning*, pages 1–6, 2013.

[10] H. Li, Y. Xiong, C. V. Hunter, X. Guo, and R. Tywoniw. Does peer assessment promote student learning? A meta-analysis. *Assessment and Evaluation in Higher Education*, 45(2):193–211, 2020.

[11] K. Lundstrom and W. Baker. To give is better than to receive: The benefits of peer review to the reviewer's own writing. *Journal of Second Language Writing*, 18(1):30–43, 2009.

[12] M. M. Nelson and C. D. Schunn. The nature of feedback: How different types of peer feedback affect writing performance. *Instructional Science*, 37(4):375–401, 2009.

[13] K. Topping. Peer Assessment between Students in Colleges and Universities Author (s): Keith Topping Source : Review of Educational Research , Autumn , 1998 , Vol . 68 , No . 3 (Autumn , 1998), Published by : American Educational Research Association Stable URL : . 68(3):249–276, 1998.

[14] K. J. Topping. Peer assessment. *Theory into Practice*, 48(1):20–27, 2009.

[15] M. van Zundert, D. Sluijsmans, and J. van Merriënboer. Effective peer assessment processes: Research findings and future directions. *Learning and Instruction*, 20(4):270–279, 2010.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5999–6009, 2017.

[17] Y. Xiao, G. Zingle, Q. Jia, S. Akbar, Y. Song, M. Dong, L. Qi, and E. Gehringer. Problem detection in peer assessments between subjects by effective transfer learning and active learning. (Edm):516–523, 2020.

[18] Y. Xiao, G. Zingle, Q. Jia, H. R. Shah, Y. Zhang, T. Li, M. Karovaliya, W. Zhao, Y. Song, J. Ji, A. Balasubramaniam, H. Patel, P. Bhalasubbramanian, V. Patel, and E. F. Gehringer. Detecting Problem Statements in Peer Assessments. 2020.

[19] Q. Xie, M. T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2020.

[20] W. Xiong and D. Litman. Identifying problem localization in peer-review feedback. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6095 LNCS(PART 2):429–431, 2010.

[21] W. Xiong and D. Litman. Automatically predicting peer-review helpfulness. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2(2009):502–507, 2011.

[22] W. Xiong, D. Litman, and C. Schunn. Assessing reviewers' performance based on mining problem localization in peer-review data. *Educational Data Mining 2010 - 3rd International Conference on Educational Data Mining*, pages 211–220, 2010.

[23] G. Zingle, B. Radhakrishnan, Y. Xiao, E. Gehringer, Z. Xiao, F. Pramudianto, G. Khurana, and A. Arnav. Detecting suggestions in peer assessments. *EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining*, (Edm):474–479, 2019.