Problem detection in peer assessments between subjects by effective transfer learning and active learning

Yunkai Xiao [yxiao28],[1] Gabriel Zingle [gzingle],[1] Qinjin Jia [qjia3],[1] Shoaib Akbar [sakbar],[1]
Yang Song [songy],[2] Muyao Dong [1120172192],[3] Li Qi [1120172633],[3]
and Edward Gehringer [efg][1]

[1] North Carolina State University, Raleigh, North Carolina 27695, USA [@ncsu.edu]
[2] University of North Carolina at Wilmington, Wilmington, NC 28407, USA [@uncw.edu]
[3]Beijing Institute of Technology, Beijing 110819, China [@bit.edu.cn]

ABSTRACT

Peer assessment adds value when students provide "helpful" feedback to their peers. But, this begs the question of how we determine "helpfulness." One important aspect is whether the review detects problems in the submitted work. To recognize problem detection, researchers have employed NLP and machine-learning text classification methods. Past studies have used datasets that were narrowly focused on a small number of classes in specific academic fields. This paper reports on how well models trained on one dataset or field perform on data from classes that are unlike the classes whose data they have been trained on. Specifically we took a model developed with data from a computer science class with several programming assignments, and tried to transfer it onto an education class focused more on writing research papers. We have attempted to perform such a task on a few models including logistic regression classifier, random forest classifier, multinomial naive bayes classifier and support vector machine. We made several attempts to raise the accuracy of classification, including lemmatizing to deduct variation in data input, and active learning strategies.

1. INTRODUCTION

The term "peer assessment" means students reviewing each other's work. The practice has been widely used for at least fifty years. It began as a face-to-face process, with students exchanging their papers. For the last twenty-five years or so, peer assessment has also been performed using online applications. Peer assessment has many advantages. From a pedagogical point of view, the greatest advantage is that it helps students understand the requirements for the assignment, and see how their work measures up to their peers [1, 2, 3, 4, 5]. This helps them to improve their own work product. From an operational standpoint, peer assessment is scalable—no matter how many students are in the course, students' work does not want for personal attention. This makes peer assessment especially useful for MOOCs, where it is frequently to provide feedback and to assign grades.

Student work on a MOOC can be graded in different ways. If objective questions are posed, such as multiple-choice and true/false questions, they can be automatically graded by software that checks whether answers match the key, while for subjective issues such as coding projects and essays, it becomes a bigger challenge. These platforms often utilize quantitative methods such as averaging reviewer scores on multiple sections of peer assessment related to the course assignment.

Current peer grading approaches are based on the numerical scores assigned to rubric items by each reviewer. Rarely do they utilize another very important piece of information: the justification given by reviewers for the grades they're giving.

Fundamentally, the quality of a review is related to whether it identifies ways for the author to improve the work. Thus, it is important for the review to point out shortcomings or problems in the existing work. Other researchers [6] have done preliminary work in this area. They have looked at approaches to detecting suggestions [7], for the reason that suggestions help students act on improving the work they have done. Other work involves recognizing problem statements. A problem statement helps people realize the shortcomings in their work, and pointing out a problem does not require as much thinking as knowing what is wrong and coming up with a solution to correct the problem as making a suggestion does. In the context of peer review, if we could tell whether a comment contains one of these features (suggestion or problem statement), we could compare a reviewer's work with other reviewers' and urge him/her to add more to the review if his/her review lacks these features significantly. In order to accomplish it, a means of automatically detecting these features needs to be devised.

We have built text classifiers that can recognize whether a comment contains a problem statement; however there's a drawback. As researchers know, text classifiers are very domain specific, that is if a classifier is trained on one specific domain, it will probably not perform well when used on another domain [8]. When MOOCs offer classes in multiple fields, the peer reviews in each class will have different language features. Useful sentiment features such as problem statements would not be the same in different classes. Traditionally, there would be multiple classifiers trained on each

Yunkai Xiao, Gabriel Zingle, Qinjin Jia, Shoaib Akbar, Song Yang, Muyao Dong, Li Qi and Edward Gehringer "Problem detection in peer assessments between subjects by effective transfer learning and active learning" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 516 - 523

one of the domains to achieve optimal performance. An issue with this approach is that there needs to be enough labeled data from each of these domains, which in a lot of cases is hard to achieve. Labeling is becoming one of the most expensive steps in machine learning, both from the perspective of time and of money [9]. However, there are a number of ways to work around this problem, if not completely mitigate it.

Researchers have demonstrated that traditional machine learning and deep learning technologies are useful for problem detection in peer review in the computer science field [10]. The researchers aim to generalize the problem detection function to different subjects. There are two potential methods for quickly building a model in a target domain and avoiding much of the time-consuming and expensive data labeling efforts. Such methods include transfer learning and active learning. With these two approaches, problem detection could be transferred quickly to a new field and at a reduced cost.

The first approach is to leverage transfer learning to transfer "knowledge" learned from the problem detection task in the computer science field to the other field. This process can use model insights gained from other datasets to expedite the construction of a new model while including only a small amount of labeled data in the target domain. In our case, we trained a problem detection classifier from data generated in a computer science class. One of the research questions we aim to discuss is leveraging transfer learning to effectively preserve the performance of the model when it is applied to other classes.

The second method is to utilize active learning to label abundant data and then apply machine learning algorithms or train deep neural network models on this automatically labeled data. This method is detailed in the implementations subsection of the experiment section of this paper.

2. LITERATURE REVIEW

2.1 Problem Detection

There have been plenty of attempts to apply natural language processing (NLP) techniques and machine-learning (ML) algorithms on automating various aspects of review assessment. Brun and Hagege [11] leveraged NLP techniques to identify suggestions in review text. Zingle et al [7] attempted to use different ML and Deep learning algorithms to determine whether a review text contains suggestions. Nguyen et al. [12] used logistic regression to train a model that predicted whether a review comment contained a problem solution. They provided this information to the reviewer before the review was submitted, in order to encourage the reviewer to suggest solutions for problems in the work.

However, most of the current research related to applying NLP and ML on peer review is limited to one subject or ones filled with enough labeled data. For example, research from Zingle et al. [7] collected student annotated peer reviews from a graduate level computer science course and used this labeled data to train models for detecting suggestions in the course. The study by the Brun and Hagege [11] did similarly with abundant manually annotated customer reviews. To the best of our knowledge, there are no published papers that address the issue of how to apply NLP and ML on peer reviews in a field without abundant labeled data. This paper is based on previous research about detecting problem statements in peer assessments [10]. This paper focuses on detecting problem statements in a field without abundant labeled reviews by utilizing transfer learning and active learning.

2.2 Transfer Learning

In most traditional machine learning algorithms, an essential hypothesis is that the training data and test data must be in the same feature space and have the same distribution [13, 14]. If the feature space or latent distribution changes, sufficient labeled data from the new domain will be needed and the statistical model must be rebuilt from scratch. This approach can be time-consuming and expensive in many real-world applications like text classification and thus limits its development [15]. The peer-review comments from the computer science field and the peer-review comments from other subjects might be in the same feature space but in different distribution, where plenty of peer-review comments from each field must be labeled and a learner must be reconstructed from scratch for each subject.

In contrast, transfer Learning, which is fundamentally motivated from a discussion in a NIPS-95 workshop [16], relaxes the hypothesis that the training data must be in the same feature space and identically distributed with the test data [13, 14]. The basic idea of transfer learning is to transfer "knowledge" learned from source tasks to different but related target tasks. This is to combat against the problem of an insufficiently large labeled training dataset and to improve the learning of the target task by reducing the labeling cost. In this case, only a small quantity of labeled data in the target domain is required. Negative transfer may occur, but a successful "transfer" would greatly improve the performance and reduce the cost of learning for the target task by avoiding much time-consuming and expensive data labeling efforts.

Pan and Yang [13] summarized various transfer learning settings and categorized transfer learning under three subsettings. These include inductive transfer learning, transductive transfer learning, and unsupervised transfer learning, based on different situations between the source and target domains and task. This paper is under the inductive transfer learning setting, which has different, yet related source and target domain tasks, where a sufficient quantity of labeled data is only required in the source domain. There are five main approaches for conducting the inductive transfer learning from literature. These approaches are instance-based transfer learning [17, 18], feature-representation transfer[19, 20], parameter-transfer [21, 22], relational-knowledge-transfer problem [23, 24], and Hybrid-based (instance and parameter) transfer learning [25, 26].

The parameter-transfer approach mentioned above is a simple but effective method for transferring "knowledge" by sharing parameters. Assumption of the approach is that some parameters are shared by source tasks and target tasks [13]. The "knowledge" is encoded into and transferred across tasks by those shared parameters.

2.3 Active Learning

Active Learning is a significant subfield of machine learning and a helpful technique in many real-world applications where there is abundant unlabeled data, but where labels are difficult, time-consuming, or expensive to acquire [27]. Active learning algorithms are allowed to interactively query a human annotator called teacher or oracle to label the new data point chosen by a predefined strategy and usually perform better with less labeled trained data. There are three main settings in which the learner may be able to query. These settings are membership query synthesis proposed by Angluin [28], steam-based selective sampling proposed by Cohn et al. [29] and pool-based active learning proposed by Lewis and Gale [30].

The most common active learning scenario is the pool-based active learning setting, which assumes that there is a smaller set of labeled data and a large pool of unlabeled instances. The key hypothesis of pool-based active learning is that the learning algorithm would perform better with less training if the algorithm could determine which instances in the pool are the most informative and is allowed to ask queries based on a certain query strategy. This would be in the form of unlabeled instances that are to be labelled by an oracle (e.g. a human annotator) [27, 30]. Hoi and et al. [30] investigated the pool-based scenario on large-scale text classification and first demonstrated the feasibility of batch mode pool-based setting active learning on the text categorization problem.

Under each active learning scenario, there have been a number of query strategies proposed for evaluating the informativeness of unlabeled instances. We evaluated the most common query strategies, uncertainty sampling published by Lewis and Gale [30]. The uncertainty sampling strategy selects the instance in the pool about which model is least certain on how to label observations according to an uncertainty measure like entropy.

In contrast to active learning, traditional passive learning would use a random sampling strategy to select instances from a large pool of unlabeled instances. This strategy generally underperforms compared with uncertainty sampling strategy thus is not adopted here.

3. EXPERIMENT

3.1 Data

To train the problem statement classifier, we used a dataset pulled from the Expertiza system. Expertiza is a web based education platform instructors can use to distribute homework assignments and team projects. The key feature of this platform comes in later stages once students submit their work, where they assess the work product of other students by giving a numeric score as well as a comment to justify their decision. For team assignments, students would assess work done by other teams, as well as the contributions of their teammates.

In some of the classes, students are asked to annotate the comments they received with an incentive of extra credit with a “yes” or “no” on given metrics. For example, some metrics that the students label for include “Does the comment contain a problem statement?”, “Does the comment offers a suggestion?”, or “Was the comment localized to a particular place in the work?“. This is a valuable source of annotated data for our research, as students should be experts at annotating feedback on their own work. However, many times more steps are required to improve the quality of this data. On observing the annotations, we found a number of problems. Sometimes students would rush through the annotation with the goal of getting extra credit with minimal effort, leaving a trail of yes’s or no’s without actually reading the comments. Other times fatigue may set in while annotating a large number of comments, resulting in the accuracy of labels gradually dropping towards the end of the annotation process. To resolve this issue, the course staff and the research team checks labels applied by the students through random sampling of the students’ annotations. If it appeared that a student was not taking adequate care, that student’s annotations would be removed from the dataset.

We extracted data from computer science class projects. Since every member of the team is involved in annotating reviews they received for team projects, we were able to calculate inter rater reliability using Krippendorff’s alpha, which was relatively low at a value of 0.696. To improve the accuracy of our model, we decided to only take those data with consensus among all annotators, by removing those with any conflicted labels, which decreased the size of our dataset by 4649, resulting in an improved Krippendorff’s alpha of 1. We then further altered the dataset by downsampling the majority class by 313 observations to ensure a balanced proportion of classes.

To prove that language features, specifically for problem statements in this particular dataset could be transferred, we run a test on three other datasets. The first composes Hotel product reviews, the second Amazon reviews, and the third a small dataset from a university level education class. The Hotel and Amazon datasets were found on the website Kaggle, which states that the data originated from the website Datafiniti. Two useful columns from the original datasets included a review score from the original 1 to 5 scale, with 1 being very bad to 5 being very good, and a column with the actual review text. From inspecting the data, we found that reviews with low ratings mentioned problems regarding the respective hotels or amazon products they were reviewing, while there was no mention of a problem in well rated reviews. Based on this information, we kept all the reviews with a rating of 1 or 2 and relabeled them all to the value 1 to represent that these reviews mentioned a problem. We then kept an equal quantity of positive reviews, all labeled 5, and relabeled these to the value 0 to represent that these reviews did not mention a problem.

The target domain dataset that we’re primarily trying to transfer is generated from the education class, which had been taught using the Expertiza system. The nature of assignments in this particular class involves much more writing in terms of research papers as compared with the project based assignment in the computer science class. Students in this class are not asked to annotate the feedback reviews they’ve received, thus creating an issue in terms of a lack of labeled data. Different members of the research team did some manual inspection and annotation on small subsets of this data, then removed those data entries with conflicting labels to reach a complete consensus. This dataset was manually labeled by our research team as either 1 mentioning a problem, or 0 not mentioning a problem.

We started by preprocessing the text in all four of the datasets. Specifically, we removed all punctuation aside from sentence ending period marks. We then removed all special characters and numbers. We removed URL links and converted the text to lowercase. Afterwards, we decided to balance the datasets using downsampling in terms of class proportion for observations mentioning and not mentioning a problem. This helps with models, particularly Naive bayes, to prevent overpredicting a class based on the proportion of training data of a certain class instead of the input features. However, we did not balance the Education dataset since it was not being used to train the models and due to its small size. The total number of observations in the Expertiza, Hotel, Amazon, and Education datasets were 18354, 4460, 2442, and 172 (122 labeled 0 and 50 labeled 1) respectively.

Additionally, we have attempted to apply lemmatization and stopword removal to gauge its impact on model performance. The intuition of this is with lemmatization, we would reduce the variation of data embedding, helping the models to focus on important features to achieve better results.

3.2 Models

Before we could transfer knowledge into models that work in the target domain, some machine learning from the source domain is required. For this task, we pick four models including the Random Forest classifier, multinomial naive Bayes classifier, support vector machine, and logistic regression classifier. Each classifier used the same 90-10% train-test split with hyper parameters tuned using 5-fold cross-validation.

Leveraging the power of the Scikit-learn package, we were able to build a data pipeline for this task [31]. Cleaned data was funneled into a count vectorizer, then weight transformed with a TF-IDF transformer, before being used by the classifiers.

The logistic regression classifier uses a regression equation to produce discrete binary outputs through a sigmoid function. It learns the coefficients of each input feature through the fitting process just like in linear regression.

The random forest classifier uses an ensemble approach that fits multiple decision trees, then uses averaging to improve the accuracy of predictions as well as to avoid overfitting. The loss criterion to choose from includes gini and entropy.

The multinomial naïve Bayes classifier is a special instance of a naive bayes classifier that follows a multinomial distribution for each feature p(f i|c). The naïve Bayes model assumes that each of the features it uses for classification are independent of one another.

The support vector machine classifier works by establishing a decision boundary as well as a positive plane and a negative plane between classes. Anything in the positive plane is considered to have the characteristic under study. In our experiment, this is the presence of a problem in a reviewer's comment.

We have also attempted doing the same task with a neural-network based model. One popular network structure in natural language processing is the Long Short Term Memory (LSTM) network. The LSTM takes the cleaned dataset as input, then using GloVe [32] embedding as a feature extractor before feeding them into a stacked LSTM and dense layers.

LSTM is a variation of Recurrent Neural Network (RNN), with the modification of adding the functionality of forgetting information when new information is fed into the network. This particular network leverages existing advantages of memorizing information through timesteps, and in the meantime uses four gates to input, forget, update, and output information.

3.3 Implementations

To validate our ideas on if detecting problem statements could be transferred, we did some initial experiments by training models on one dataset and then test on another. Results of these experiments could be found in the following section of the paper, where we did observe signs of knowledge being transferred and proceeded to the next stage on improving model accuracy on new domains.

Apart from transferring existing knowledge from other domains, the other way to diminish the impact of lacking annotated data is active learning. Active learning helps researchers to lessen the effort annotation by selecting a subset of high value data to annotate. Different active learning strategies may generate different subsets of data, but the essence of doing so is that it would pick data that can bring more knowledge to the models compared with other data points.

During the active learning phase, we attempted applying uncertainty sampling strategy to actively learn the more important groups of data-points listed by each model respectively. Unlabeled data from the education class dataset is exposed to all four models, and they would go through predicting whether a problem statement is present in a comment, generating labels of 1's and 0's as well as corresponding confidence scores. Using the score, we could retain four subsets of data points of which the models' confident scores are between 49% and 51%.

Two researchers then annotate over 100 of these data-points per subset, then remove conflicted entries, leaving 100 labeled data-points which each of these models are "curious" about. These observations are then appended to the computer science dataset which we originally trained the models with. Finally, the four models were re-trained separately.

4. RESULTS

In Tables 1, 2, 3, and 4 the rows represent the dataset that was used to train the model. The columns represent the dataset that was tested on by the model. In the cases marked by the diagonal in the tables, we trained the models using 90% of the dataset and tested on the remaining 10%. The order of the sets of three values within each represent the results without any further text preprocessing, lemmatization, and stopword removal respectively.

Table 1: F1 Score Logistic Regression

TrainTest	Computer Science	Hotel	Amazon	Education
Computer Science	0.89 / 0.89 / 0.83	0.70 / 0.69 / 0.68	0.70 / 0.71 / 0.63	0.73 / 0.69 / 0.64
Hotel	0.68 / 0.68 / 0.55	0.94 / 0.93 / 0.94	0.82 / 0.85 / 0.8	0.65 / 0.63 / 0.59
Amazon	0.60 / 0.58 / 0.47	0.78 / 0.81 / 0.76	0.95 / 0.93 / 0.93	0.65 / 0.63 / 0.53

*without preprocessing / with lemmatization / with stopword removed

Table 2: F1 Score Random Forest

TrainTest	Computer Science	Hotel	Amazon	Education
Computer Science	0.88 / 0.89 / 0.82	0.62 / 0.60 / 0.62	0.66 / 0.65 / 0.57	0.68 / 0.65 / 0.66
Hotel	0.74 / 0.74 / 0.59	0.91 / 0.90 / 0.92	0.73 / 0.74 / 0.73	0.61 / 0.63 / 0.60
Amazon	0.58 / 0.54 / 0.43	0.73 / 0.75 / 0.72	0.91 / 0.93 / 0.91	0.62 / 0.55 / 0.50

When models are trained on one dataset and tested on another dataset without any prior knowledge for the target domain, we could expect some drop in performance. As we tested each model's performance on different datasets, we validated this claim and found that the degradation of model performance is closely related to how much domains differ from each other.

For example, when we initially tested if something constituted a problem statement that was learned from the computer science could be transferred to other domains, we found that despite a drop of 0.2 - 0.3 in F1 score, each model did receive a F1 score larger than 0.6 for most of the runs, which is better than the random guessing average of 50%. This is a sign of positive transferring of knowledge, thus proving our idea could work.

Apart from the naive Bayes classifier, we received good results when testing on the Education dataset. This could be caused by the nature of reviews towards computer science sharing more similarities with the education dataset since they are both done by students towards their peers, unlike the other two. Apart from that, we found the knowledge transferring to the Amazon dataset constantly out performing knowledge transferring to the Hotel dataset. When closely observing the content of the Amazon dataset, we found it is focused on reviewing electronic devices such as Amazon Kindle and Kindle fire. The nature of such projects do share some similarity with reviewing an application built by a computer science student, and as expected we could find knowledge transferred better from a computer science class to Amazon reviews compared with those from the Hotel dataset. All of the findings above can be found in Tables 1, 2, 3, and 4. Unsurprisingly, when we compare transferring knowledge between different domains through datasets that we have acquired, it can also be found that transferring works the best between the two commercial review datasets, being Amazon and Hotel, due to their nature being customer rather than peer reviews.

We analyzed the most important features most models used for prediction by examining feature coefficients from these models. The results of this examination also aligned with our observations. Within the top 20 positive and negative coefficients, we found 5 pairs of shared features between the computer science dataset and Amazon dataset. We also found 6 pairs between the computer science dataset and Hotel dataset. Furthermore, there were 7 pairs of shared features between the Amazon dataset and Hotel dataset.

The models resulted in similar performances with and without the use of lemmatization for training and testing on the same dataset. Lemmatization did increase the accuracy when models were trained on the Hotel dataset and tested on the Amazon dataset, and vice versa. However, stopword removal led to a significant decrease in classifier performance in all cases except for when the models were trained and tested on the same dataset for the Hotel and Amazon dataset, in which case the performance was around the same.

The logistic regression classifier and support vector machine led to the best results when training and testing on the same dataset, with the exception of multinomial naive bayes when using the Hotel dataset. Otherwise, the multinomial naive bayes classifier performed the worst, particularly when attempting to predict observations found in the Education dataset.

When tested and trained on the same dataset, the models performed well with f1-scores ranging from mid 80s to mid 90s.

To bring up the accuracy when we transfer a model onto another domain, we did some active learning attempts. By using the uncertainty sampling strategy, each of the four models were exposed to the unlabeled education dataset, then the top hundred data points denoted unsure by each model is extracted. Each of these data points had a confidence between 49% and 51%, and were presented to an oracle (human annotator) for labeling. After removing conflicting labels, these subsets of data were appended to the original computer science dataset individually based on which model mentioned the uncertainty, then used to retrain each model respectively.

We found that with a very small carefully picked set of data, we could regain a considerable amount of accuracy after transferring a model onto a new domain. As could be seen in Figure 1, which details the affects of adding the target domain data from active learning to the computer science dataset, all models gained accuracy with Naive Bayes benefiting the most from this process.

Table 3: F1 Score Naive Bayes

TrainTest	Computer Science	Hotel	Amazon	Education
Computer Science	0.86 / 0.85 / 0.80	0.55 / 0.55 / 0.53	0.56 / 0.58 / 0.54	0.50 / 0.55 / 0.53
Hotel	0.56 / 0.53 / 0.50	0.95 / 0.95 / 0.93	0.79 / 0.82 / 0.78	0.57 / 0.54 / 0.53
Amazon	0.59 / 0.55 / 0.52	0.80 / 0.82 / 0.77	0.94 / 0.93 / 0.94	0.57 / 0.57 / 0.55

Table 4: F1 Score Support Vector Machine

TrainTest	Computer Science	Hotel	Amazon	Education
Computer Science	0.90 / 0.90 / 0.83	0.69 / 0.68 / 0.67	0.69 / 0.70 / 0.63	0.74 / 0.69 / 0.64
Hotel	0.66 / 0.66 / 0.56	0.93 / 0.94 / 0.94	0.83 / 0.85 / 0.80	0.65 / 0.62 / 0.61
Amazon	0.63 / 0.59 / 0.48	0.79 / 0.80 / 0.75	0.94 / 0.93 / 0.93	0.67 / 0.66 / 0.53

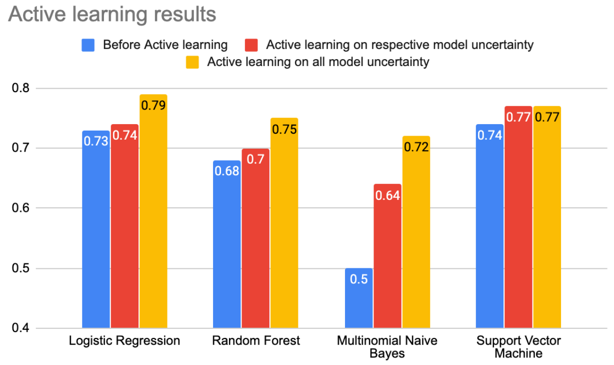

Figure 1: F1 Improvements with Active Learning

There are also a few things we noticed that did not work. Ordinary data preprocessing techniques such as lemmatizing and mainly stopword removal actually reduced model performance in terms of accuracy on all four models. From reviewing the coefficients, we found that many times the tense and plurality of words actually matters, let alone a lot of the stop words. For example auxiliary verbs such as “could” and “should” often implies a problem needs to be corrected, and words implying contrast such as “but” and “however” are used to bring up readers’ attention before mentioning dissatisfaction. When these elements of language are removed, predicting whether a comment contains a problem becomes harder.

Apart from this, attempts on generating uncertain data from Neural network models and then re-train itself with resolved uncertainty does not show significant differences compared with training itself on more randomly selected samples. Results for both approaches have a F1 score fluctuate between 0.69 and 0.71 without significant differences. This could be because each time a neural network is trained, it restructures itself in a different way. With each perceptron (neuron) being a small classifier by itself, what is used to carry important knowledge to one network state might not hold as much value when the network is in a new state.

5. CONCLUSIONS AND FUTURE WORK

In conclusion, we could use models trained on one domain that classify certain sentiment components on other domains. We have tested doing problem detection between two distinctively different classes, and are confident about detecting other useful things such as suggestions or problem localizers. Results in the previous section have presented that with very little human intervention, each of the classifiers could regain a significant amount of its accuracy.

This is a very important step if we are to build a system that could promote students writing better reviews in different domains and different class settings. Furthermore, if we are to automate the grading process by involving inputs from peer assessment, we would certainly want to use features such as “how many suggestions are made” or “how many problems did the reviewer find” to gauge the quality of peer grading. Being able to analyze these features across peer assessments from different subjects becomes increasingly important.

Within this article, we mainly focused on transfer learning on traditional machine learning techniques, while there are many deep transfer learning techniques which could be utilized. With smaller datasets they might not have made much difference in terms of model accuracy. However, other researchers have shown that using layers in these neural networks trained on one dataset could be used as feature extractors for another. Examples of this are GloVe [32] and BERT [33], where both of these models are trained on a much larger dataset, resulting in exposure to a variety of knowledge, then later repurposed as feature extractors for other tasks.

In the future, we plan to explore the possibility of using transfer learning and active learning on neural network models and to continue building a review helpfulness evaluator across different subjects. In the long run, we would like to create a system that automatically assigns grades to students based on both numerical and textual peer assessments.

6. REFERENCES

[1] Hongli Li, Yao Xiong, Charles Vincent Hunter, Xiuyan Guo, and Rurik Tywoniw. Does peer assessment promote student learning? a meta-analysis. *Assessment & Evaluation in Higher Education*, pages 1–19, 2019.

[2] Kristi Lundstrom and Wendy 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.

[3] Yasemin Demiraslan 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.

[4] Lan Li, Xiongyi Liu, and Allen L Steckelberg. Assessor or assessee: How student learning improves by giving and receiving peer feedback. *British Journal of Educational Technology*, 41(3):525–536, 2010.

[5] Esther Van Popta, Marijke Kral, Gino Camp, Rob L Martens, and P Robert-Jan Simons. Exploring the value of peer feedback in online learning for the provider. *Educational Research Review*, 20:24–34, 2017.

[6] Kwangsu Cho. Machine classification of peer comments in physics. In *Educational Data Mining*, 2008.

[7] Gabriel Zingle, Balaji Radhakrishnan, Yunkai Xiao, Edward Gehringer, Zhongcan Xiao, Ferry Pramudianto, Gauraang Khurana, and Ayush Arnav. Detecting suggestions in peer assessments. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 474–479. International Educational Data-Mining Society, 2019.

[8] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[9] P. Perona P. Welinder. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *CVPR*, 2010.

[10] Xiao Yunkai, Gabriel Zingle, Qinjin Jia, Harsh Shah, Yi Zhang, Tianyi Li, Mohsin Karovaliya, Weixiang Zhao, Yang Song, Jie Ji, Ashwin Balasubramaniam, Harshit Patel, Priyankha Bhalasubbramanian, Vikram Patel, and Edward Gehringer. Detecting problem statements in peer assessments. In *Proceedings of the 13th International Conference on Educational Data Mining*. International Educational Data-Mining Society, 2020.

[11] Caroline Brun and Caroline Hagege. Suggestion mining: Detecting suggestions for improvement in users' comments. *Research in Computing Science*, 70(79.7179):5379–62, 2013.

[12] Huy Nguyen, Wenting Xiong, and Diane Litman. Instant feedback for increasing the presence of solutions in peer reviews. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 6–10, 2016.

[13] SJ Pan and Q Yang. A survey on transfer learning. ieee transactions on knowledge and data engineering, 2010.

[14] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer, 2018.

[15] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.

[16] Jonathan Baxter, Rich Caruana, Tom Mitchell, Lorien Y Pratt, Daniel L Silver, and Sebastian Thrun. Learning to learn: Knowledge consolidation and transfer in inductive systems. In *NIPS Workshop, http://plato. acadiau. ca/courses/comp/dsilver/NIPS95_LTL/transfer. workshop*, 1995.

[17] J Quiñonero Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. Dataset shift in machine learning. *The MIT Press*, 1:5, 2009.

[18] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, 2007.

[19] Chuen-Kai Shie, Chung-Hisang Chuang, Chun-Nan Chou, Meng-Hsi Wu, and Edward Y Chang. Transfer representation learning for medical image analysis. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 711–714. IEEE, 2015.

[20] Yi Zhu, Xuegang Hu, Yuhong Zhang, and Peipei Li. Transfer learning with stacked reconstruction independent component analysis. *Knowledge-Based Systems*, 152:100–106, 2018.

[21] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Residual parameter transfer for deep domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4339–4348, 2018.

[22] Chao Chen, Boyuan Jiang, and Xinyu Jin. Parameter transfer extreme learning machine based on projective model. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[23] Donghui Wang, Yanan Li, Yuetan Lin, and Yueting Zhuang. Relational knowledge transfer for zero-shot learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[24] Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 219–228, 2019.

[25] Rui Xia, Chengqing Zong, Xuelei Hu, and Erik Cambria. Feature ensemble plus sample selection: domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3):10–18, 2013.

[26] Rui Xia and Chengqing Zong. A pos-based ensemble model for cross-domain sentiment classification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 614–622, 2011.

[27] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[28] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[29] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[30] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994.

[31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[32] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.