Active Learning for Improving Machine Learning of Student Explanatory Essays

Peter Hastings[1(✉)], Simon Hughes[1], and M. Anne Britt[2]

[1] School of Computing, DePaul University, Chicago, IL, USA
peterh@cdm.depaul.edu
[2] Psychology Department, Northern Illinois University, DeKalb, IL, USA

Abstract. There is an increasing emphasis, especially in STEM areas, on students' abilities to create explanatory descriptions. Holistic, overall evaluations of explanations can be performed relatively easily with shallow language processing by humans or computers. However, this provides little information about an essential element of explanation quality: the structure of the explanation, i.e., how it connects causes to effects. The difficulty of providing feedback on explanation structure can lead teachers to either avoid giving this type of assignment or to provide only shallow feedback on them. Using machine learning techniques, we have developed successful computational models for analyzing explanatory essays. A major cost of developing such models is the time and effort required for human annotation of the essays. As part of a large project studying students' reading processes, we have collected a large number of explanatory essays and thoroughly annotated them. Then we used the annotated essays to train our machine learning models. In this paper, we focus on how to get the best payoff from the expensive annotation process within such an educational context and we evaluate a method called Active Learning.

1 Introduction

There is an increasing emphasis at the educational policy level on improving students' abilities to analyze and create explanations, especially in STEM fields [1,2]. This puts pressure on teachers to create assignments that help students learn these skills. On such assignments, teachers could provide several different kinds of feedback, including identification of spelling and grammatical mistakes, overall holistic evaluations of explanation quality, and detailed analyses of the structure of the explanation — what parts of a good explanation were present and how they were connected together, and what parts were missing. It is much

The assessment project described in this article was funded, in part, by the Institute for Education Sciences, U.S. Department of Education (Grant R305G050091 and Grant R305F100007). The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

easier for teachers (and computers using shallow processing techniques) to provide the first two types of feedback [3]. Deep analysis of explanation structure is much more challenging, but it is necessary for helping students truly improve the quality of their explanations. Holistic and shallow evaluations may help students fix local problems in their explanations, but they do not help students create better chains from causes to effects that are the core of good explanations.

An AI system for analyzing the structure of explanations could be used in a variety of ways: as the back-end of an intelligent tutoring system that would help students write better arguments, as an evaluation system to provide formative assessment to teachers on their students' work, or as the basis of deeper summative assessments of the writing [3].

In educational contexts, as in many others, there is growing availability of large amounts of data. That data can be leveraged by increasingly sophisticated machine learning techniques to evaluate and classify similar data. The bottleneck in many such situations is that most machine learning techniques require a significant amount of labeled data — i.e., data that has been annotated by human coders at high cost of time and money — in order to be effective. A large number of texts may be collected, but what is the best strategy for annotating enough of those texts to produce an automated system that can effectively analyze the rest? That is the research question that we address in this paper.

For several years, we have been working as part of a project aimed at studying students' *reading* processes. To assess how much students understand from what they have read, we collected over 1000 explanatory essays dealing with a scientific phenomenon. Over the course of six months, expert annotators identified the locations of conceptual information and causal statements in these essays. This has provided us with an excellent data set on which to evaluate our research question about how to get the necessary sample size of annotated data for adequate performance; we simply assume that most of the data is unlabeled and try to identify methods that allow machine learning to most quickly create a model that will accurately classify the rest.

This paper focuses on one method called Active Learning in which you start with a small set of labeled data for training. Based on the uncertainty of classification of the rest of the data, you select another batch of data to be labeled, and continue this process until acceptable performance is achieved. The paper describes the specifics of the educational context that our data came from and how the essays were collected and annotated. Then we describe related research and the experiments we performed. We conclude with a discussion of our experiments and results and implications for future research.

2 Student Explanatory Essays

The essays used in this research were scientific explanations generated by 9th grade biology students in 12 schools in a large urban area in the United States. During a 2-day, in-class activity, students were given 5 short documents that included descriptive texts (M = 250 words), images, and several graphs that the student could use to understand the causes of a scientific phenomenon, coral bleaching. Each document was a slightly modified excerpt from an educational website and was presented on a separate sheet with source information at the bottom of the text. In collaboration with our science educators, we co-created all materials and the idealized causal model, shown in Fig. 1, which depicts the ideal explanation that students could make from the documents. The students were told to read the documents and "explain what leads to differences in the rates of coral bleaching." They were told to use information from the texts and make connections clear. They were allowed to refer to the documents while writing their essays.

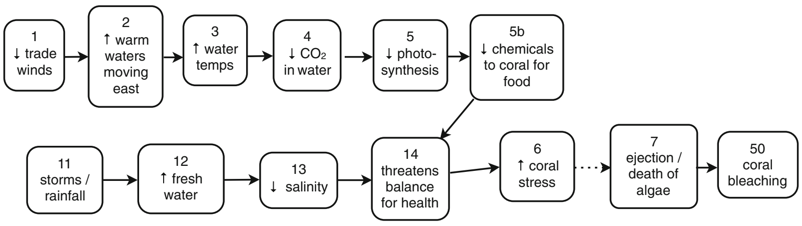
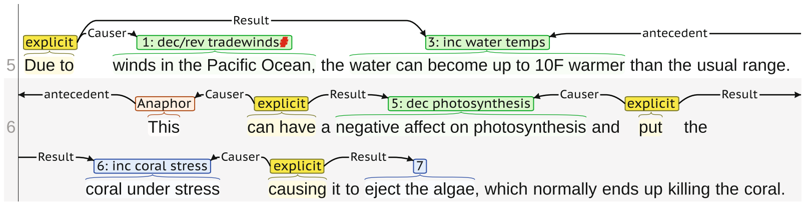


Fig. 1. Causal model for coral bleaching

As mentioned above, the primary goal of the larger research project was to study the students' reading and to try to learn how to help them read more deeply. In support of this goal, all of the essays were closely annotated to determine what the students did and did not include in their explanations. The brat [4,5] annotation tool was used for the annotation process. The mean length of the students' essays was 132 words (SD = 75). The mean number of unique concepts from Fig. 1 in the essays was 3.1 (SD = 2.2) and the mean number of causal connections was 1.3 (SD = 1.7). (See [6] for more details.)



5 Due to winds in the Pacific Ocean, the water can become up to 10F warmer than the usual range.

6 This can have a negative affect on photosynthesis and put the coral under stress causing it to eject the algae, which normally ends up killing the coral.

Fig. 2. Annotation with brat

Figure 2 shows two sentences of one (relatively good) student essay in brat, where an annotator has marked the locations of the concepts from Fig. 1 and the causal connections between them. In the first sentence, the annotator has identified a reference to concept code 1, *decreased trade winds*, and a reference to concept code 3, *increased water temperatures.* The annotator has also identified an explicit causal connection from code 1 to 3. The next sentence has an (anaphoric) causal connection from code 3 to code 5, and further connections to codes 6 and 7. Although brat significantly sped up the annotation, and although the essays were relatively short, it still took trained annotators 15–30 min to annotate each one — a significant manpower cost.

3 Related Research

Along with increased emphasis on standardized testing has come an increased emphasis on creating automatic mechanisms for evaluating written essays or responses. Although these Automatic Essay Scoring systems [7] are becoming increasingly sophisticated, they generally use shallow language analysis tools like lexical and syntactic features and semantic word sets [8,9] to provide a single holistic score for the essay rather than a detailed analysis of the contents of the essay [10]. The holistic-score approach has been criticized for its failure to identify critical aspects of student responses [11] and for its lack of content validity [9,12]. The appeal of the holistic-score approach is partly due to the tasks for which these systems are being used, but also due to the difficulty of performing a deeper analysis. Causal connections in text have been very difficult to identify, with two recent systems getting F_1 scores of 0.41 and 0.39 [13,14].

Our previous research has been more successful at identifying causal connections, producing F_1 scores of 0.73 [15]. We have some advantages however. We know what the students are basing their essays on: documents from a narrowly-defined topic. We also have a large amount of training data that we use to train classification models: 1128 annotated student essays. Recent refinements are producing even higher performance [16]. To identify concept codes, we use a window-based tagging approach, creating a separate classifier for each concept and for both causal connection types (*Causer* and *Result*) based on a sliding window of size 7 of unigrams and bigrams along with their relative positions in the window [15]. We use stacked learning [17] to identify specific causal connections from the results of the window-based tagging models. We view this level of performance as acceptable for the creation of formative feedback on the explanations. The point of this research, however, is to ascertain whether similar performance can be achieved using fewer annotated essays, and if so, what is the best method for choosing essays to annotate.

There has been a wide range of previous work aimed at reducing the amount of training data needed to produce an effective classification model. One approach was to ensure a broadly representative sampling of data, but it was found to be no better than random selection [18]. Other research has applied Active Learning (AL) to various tasks [19–21], but they have generally been concerned with predicting a single class for each instance, have often produced results not significantly better than random sampling [22], and there has been little focus on applying AL to text-based tasks [22], especially multi-class tasks like ours.

One exception [23] aimed to classify newswire articles into one of 10 different categories. In contrast to our situation, however, that was a whole-text task. We have a large set of conceptual and causal codes that we want to identify *within* the students' essays, from which we can infer the structure of their explanations.

The research reported in [24] did focus at a sub-document level, namely on temporal relations between two specified events within the text. In this case, the authors were attempting to identify which of 6 temporal relations (e.g., *Before*, *Simultaneous*) held between the two events. This was also applied to newswire texts, which tend to be longer and less error-ridden than student texts. They combined measures of uncertainty, representativeness of a new instance with previously classified instances, and diversity of the entire set to choose the next items to classify, with the first two having a larger effect than the third.

A related technique to AL is Co-training [25] which could further reduce the requirement for annotated data by applying the predictions of the current model to unlabeled data, then assuming that the instances about which the model was most certain were correctly classified, and adding some portion of those instances to the training set. It requires, however, that the model is trained with two sets of features which are conditionally independent of the target class, and performance can be degraded if the predictions were wrong.

4 Experiments

This section describes the method that we used to evaluate different variants of AL on the explanatory essays, including the overall algorithm that was used, the different selection strategies, the measures used to evaluate performance, and the two experiments that we ran.

4.1 Algorithm

Our dataset consisted of 1128 explanatory essays collected as described above. We used a variant of cross validation described below, 10-fold for the first experiment, and 5-fold for the rest:

1. Randomly select 10%[1] of the essays and put them in the initial training set, and put another 10% into the validation set.[2] The rest of the essays were put in the pool of "unlabeled" essays. (In our case, of course, these essays were actually labeled, but those labels were not used until the essays were selected for inclusion in the training set.)

[1] The percentages are all parameters to the model. These were selected because they allowed us to see the performance of the models at a reasonable granularity. It should be noted, however, that in our case, 10% of the total set represents over 100 additional essays. In real-world settings, a smaller increment would likely be used due to the cost of annotation.

[2] We used a validation or holdout set to provide a consistent basis on which to judge the performance of the models.

2. Repeat until 80% of the essays are in the training set (with 10% of the essays in the validation set and 10% in the remainder pool):
   (a) Train the model on the training set using Support Vector Machines (SVMs) [26,27] to create a classifier for each concept code and each causal connection code. The features for the SVM came from the window-based tagging method described above. The 7-word window was dragged across all the texts. Each training instance consisted of the 7 unigrams and 6 bigrams along with their relative positions in the window. The target class was the code of the word in the middle of the window, and its annotation indicated if that instance was a positive or negative instance of the class.
   (b) For each line in each essay in the remainder pool, calculate the predicted confidence for each code. Because we used SVMs, the decision boundary was at 0, so a positive confidence value was a prediction that the code was present. Negative values were predictions that the code was not present. The higher the absolute value, the more confident the model was that the code was or was not there.
   (c) Calculate Recall (= $Hits/(Hits + Misses)$) and Precision (= $Hits/(Hits + FalseAlarms)$) for each code at the sentence level in the validation set. (Codes rarely occur more than once per sentence.)
   (d) Sort all the essays in the remainder pool based on the average absolute confidence per sentence according to the selection strategy. Selection strategies are described below.
   (e) Move the next 10% of essays from the sorted remainder pool to the training set.

4.2 Instance Selection Strategies

In the AL literature, an Instance Selection Strategy refers to the technique for choosing the next item(s) to have labeled or annotated. Common strategies are based on the uncertainty of classifying the instances, the representativeness of the instances, "query-by-committee", and expected model change [19,21]. Because our instances are complex, containing many different classes (i.e., the codes from the causal model), in this work we focused on the simplest type of strategy, *uncertainty sampling*.

Although the default approach for uncertainty sampling is to prefer to select the least certain items for labeling, researchers have also evaluated other variants of this [20]. We evaluated three: the default (i.e., *Closest to margin*, the least certain), the most certain (*Farthest from margin*), and interleaving certain and uncertain items. These were implemented by sorting the remaining essays in step (d) of the algorithm above based on this criterion. Specifically, because we used SVMs to classify the codes, and because their decision boundary (the threshold between positive and negative predictions) is 0, we took the absolute value of the confidence in the prediction (i.e., the distance from the SVM's marginal hyperplane), and averaged that over all the sentences in the essay. For the different strategies, we used the lowest average confidence, the highest average confidence, and interleaving of the two, respectively. We compared these methods with a control condition: randomly selecting the next items to be added to the training set.

For each of these (non-random) strategies, we also evaluated two different methods for aggregating the confidences for each sentence. In our first experiment, we simply added all of the scores for the 51 different codes that our models were predicting: 13 codes for concepts in the causal model, and 38 codes for the "legal" connections in the model (i.e., those that respected the direction of the arrows, but potentially skipped nodes, e.g., $1 \rightarrow 2$ and $1 \rightarrow 3$, but not $2 \rightarrow 1$). We called this the *Simple Sum* method. Because the causal connection codes are so much more numerous than the concept codes, we also evaluated an aggregation method, which we called the *Split Sum* method, that normalized the confidence scores by the two sets of codes. In other words, we added the confidence scores for all of the concept codes and divided that sum by 13, then added it to the sum of the causal codes divided by 38. The intuition behind using this approach was that we wanted to avoid biasing the selection decision too much toward the (numerous but relatively rare) causal connection codes.

4.3 Measures

As mentioned above in the section describing the algorithm, we calculated Recall and Precision for each concept code and causal connection code in the validation set. From these, we calculated two averaged F_1 scores that could be used to judge the overall performance of the model. F_1 is defined as:

$$\frac{2 * Precision * Recall}{Precision + Recall}$$

The two different ways of combining the F_1 scores for all the codes in the validation set are the mean F_1 and the micro-averaged F_1. The mean F_1 is simply the average across the F_1 scores for all of the 51 different codes. The micro-averaged F_1 is derived from the Precision and Recall from the whole validation set. In other words, the overall Precision and Recall are based on the *Hits*, *Misses*, and *FalseAlarms* from the whole validation set. As a result, micro-averaged F_1 scores are sensitive to the frequency of occurrence of the codes in the set, and mean F_1 scores are not. Micro-averaged scores are representative of how the model performs *in practice*. Mean F_1 scores give equal weighting to each code to take into account rare codes as much as it does frequent ones.[3] Averaging F_1 scores can be seen as a way of evaluating a learning method in an "ideal" situation, when all frequencies are balanced. Micro-averaging evaluates the model based on its overall performance on natural data with imbalanced code frequencies. Thus, it is useful to take both into account.

[3] For what it's worth, these are analogous to the U.S. House of Representatives and Senate, respectively, with one giving more weight to more "populous" (i.e., frequent) entities, and the other giving "equal representation" to each entity.

4.4 Experiment 1: Absolute Confidence Values

As mentioned above, in our first experiment, we combined the uncertainty (or confidence) values by adding all of the absolute values of the predicted confidences for the individual codes, averaged over the number of sentences in the essay. Figures 3 and 4 show the mean F_1 scores and micro-averaged F_1 scores respectively. Each chart shows the percentage of the essays that were in the training set at each iteration on the X axis, and the resulting F_1 score on the Y axis. Each line represents one of the different methods described in Sect. 4.2 for choosing which essays to "label" (move the annotated essay to the training set).

Note that each of the evaluations presented here ends with 80%, or about 900 essays included in the training set. One reason for this is that, at this point, the remaining essays are least typical of the selection method. For example, with the high-uncertainty selection strategy, there would only be the most certain instances remaining. The more significant reason is that in a real world situation where the cost of annotation is high, you would typically want to annotate a much smaller number of items. So data in the left sides of each of the charts are more applicable to practical scenarios.

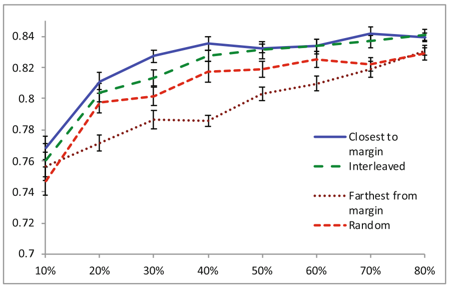


Fig. 3. Absolute mean F1s

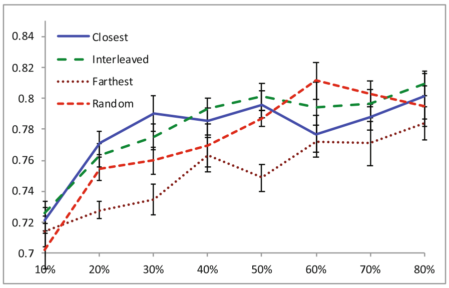


Fig. 4. Absolute micro-averaged F1s

For the mean F_1 scores, the default high-uncertainty/low-confidence/closest-to-marginal-hyperplane method always resulted in the best (or equivalent) scores on the validation set. In other words, one should choose the next set of essays to annotate by selecting those that the classifiers are least sure of. The clear "loser" was the farthest-from-marginal-hyperplane/highest-confidence method. Adding instances which the model was already predicting with high confidence resulted in much slower increase in classification performance.

For the micro-averaged F_1 scores, the results were more mixed. The closest-to-hyperplane method performed best initially, but its performance actually went down with 40% of the essays in the training set. At 60 and 70%, the best scores were produced by the random selection method. However, as mentioned above, results with lower percentages of items in the remainder pool are less indicative of what would be found in practical applications.

While this experiment gave interesting initial results, it also raised some questions. First, we noted that the scores on the randomly selected initial set (at 10%) were higher for the closest-to-hyperplane method, so we wondered what effect that might have on performance. Second, what could be the effect of the frequency of occurrence of the codes (classes) on the overall performance. The codes follow a Zipfian distribution. The most frequent code (50, which is the one the students are asked to explain) occurs in 55% (only!) of the essays. The subsequent frequencies are 12%, 4.7%, 4.2%, and so on. Forty of the 51 codes occur in less that 1% of the essays. While this is the "natural state of affairs" for this set of essays (and for many other natural multi-class situations), we hypothesized that this frequency imbalance would have a differential effect on the mean and micro-averaged F_1 scores. A model could achieve higher mean scores by performing relatively well on very infrequent codes and not so well on more frequent codes. With the micro-averaged scores, the same model would not perform as well. Because of this issue, we wanted to evaluate a method for combining the confidence scores which would take this frequency imbalance into account. We addressed these issues in Experiment 2.

4.5 Experiment 2: Performance Gain, Scaled Confidences

The first question resulting from Experiment 1 was: How does the performance of the initial training set affect increases in performance via AL. To address this question, we additionally calculated the simple performance gain for each method, which we defined as $F_{gain} = F_1@N\% - F_1@10\%$. In other words, we subtracted the method's initial absolute F_1 score from all the F_1 scores for that method. This allowed us to more easily compare performance because each one started at 0. Because the initial training sets were all chosen randomly without regard to the selection strategy, the initial absolute F_1 scores tended to be close anyway. In the results presented in the rest of this paper, we display the simple performance gain values. The initial *absolute* mean F_1 scores were all in the 0.62–0.64 range, and micro-averaged F_1 scores were between 0.70 and 0.73. These values are already relatively good for this complex task — i.e., they classify the components of the essays with sufficient certainty that beneficial feedback could be given, assuming the stakes were not too high, but the focus here is on how to improve the performance of the models most quickly.

The second question raised by Experiment 1 was about the effect of unbalanced frequencies of the codes, and we hypothesized that mean and micro-averaged F_1 scores would be affected differently. To address this question, we scaled the confidence ratings (absolute distance from the marginal hyperplane) for each code in each sentence by dividing by the log of the frequency of the code in the corresponding remainder pool.[4] We assigned a minimum code frequency of 2 to account for rare codes.

[4] Alternatively, we could have used the frequencies from the training set. We used frequencies from the remainder pool because they would be more accurate, especially at the earlier stages. In a real-life setting where the items in the remainder pool would be unlabeled, those frequencies would, of course, be unknown.

Simple Sum Confidence Combination. Figures 5 and 6 show the mean and micro-averaged F_1 gain scores for the Simple Sum combination method described above, which calculates the prediction certainty for a sentence by adding the certainties for all the codes. In Experiment 2, however, the values were scaled by the log frequency of occurrence of the codes before they were summed.

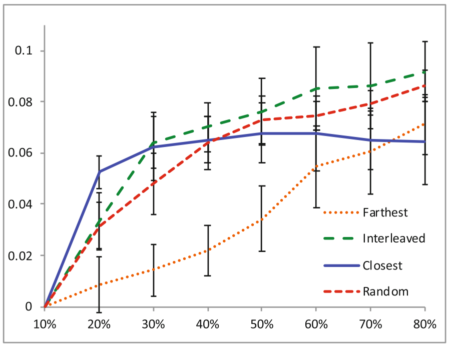


Fig. 5. Simple sum mean F1 gains

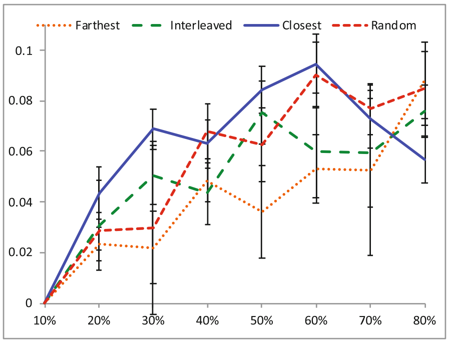


Fig. 6. Simple sum micro F1 gains

These charts make it more obvious that in the first iteration (i.e., going from 10% to 20% of the essays in the training set), choosing the least certain (closest to the marginal hyperplane) items most quickly improves the performance of the models, both for the mean and micro-averaged scores. Conversely, choosing the most confidently-classified items (farthest from the marginal hyperplane) still provides the slowest growth in model performance. The rest of the story is more subtle but supports our hypothesis about differential effects on the mean and micro-averaged F_1 scores.

In the mean F_1 scores in Fig. 5, it is clear that the closest-to-hyperplane approach plateaued, and actually decreased slightly, while the interleaved and random selection strategies kept improving. This provides some support for the idea raised in related research [20] that including a broader range of examples is beneficial, at least later in the training. The behavior of the closest-to-hyperplane selection strategy could be due to the F_1 scores for the whole set of codes not increasing, or, because the mean F_1 score evenly weights all codes, it could be that some subset goes up, and the rest go down. (Even though the weights are scaled by code frequency, the scaled values are all added together in this combination scheme.) Another factor may be that at some point, there are only high-confidence essays in the remainder set, so adding them to training does not improve overall performance.

The micro-averaged F_1 chart in Fig. 6 gives some insight. Here, the closest-to-hyperplane is always the highest, except in the last two iterations, and, by a small amount, at the fourth. This indicates that this method is, in fact, increasing performance on the most-frequent codes (because the micro-average is more sensitive to code frequency). This presumably happens because we are scaling the confidence values by frequency. With this form of scaling, we are discounting the certainty on the more frequent codes. By biasing the selection strategy further toward essays that have low confidence on frequent codes and away from essays that have low confidence on infrequent codes, we have improved the micro-averaged F_1 scores, but at the expense of the mean F_1 scores on the later iterations. Or, to put it another way, using a frequency-scaled AL combination strategy effectively increases the overall performance of the classifications given natural distribution of the classes.

Split-Sum Confidence Combination. Figures 7 and 8 show the mean and micro-averaged F_1 gain scores for the Split Sum combination method described above, which calculates the prediction certainty for a sentence by adding the average of the certainties for the concept codes with the average of the certainties for the causal codes. As above, the values were scaled by the log frequency of occurrence of the codes before they were averaged and summed. To reiterate, the concept codes identify the particular factors or events that students might identify in their explanations. The causal codes identify explicit connections between them, like "X led to Y." The rationale for the Split Sum combination method is to afford equal weight to the *set* of concept codes and the *set* of causal codes.

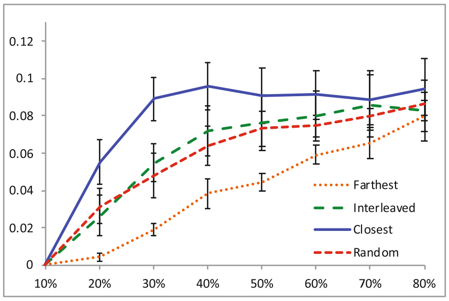


Fig. 7. Split sum mean F1 gains

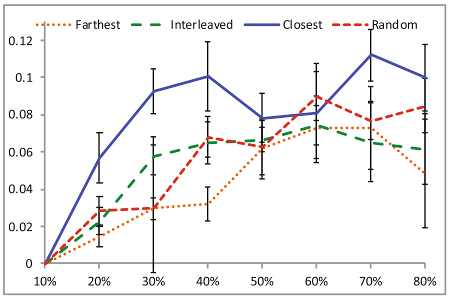


Fig. 8. Split sum micro F1 gains

In the early stages, these charts also show advantages for the closest-to-hyperplane selection strategies in both the mean and micro-average scores, but the advantages appear more pronounced. By the end of the second iteration, the closest-to-hyperplane strategy performs significantly above the others. As before, the performance of the strategy plateaus on the mean F_1 scores, but only at the point at which it is already well above the others, and it maintains its advantage. Comparison with Fig. 5 shows that it outperforms all of those models as well. Performance on the micro-averaged F_1 was also superior across the board, with the exception of one iteration. This method of combining the average of the conceptual codes with the average of the causal codes, along with the frequency-based scaling, produced a model that learned quickly and outperformed the other selection strategies.

5 Discussion, Conclusions and Future Research

The overall goal of our research project is to develop methods for analyzing the causal structure of student explanatory essays. This type of analysis could be provided to teachers to reduce the demands on them, or it could become the foundation of an intelligent tutoring system that will give students feedback on their essays and help direct the focus of their learning. From the limited number of connections included in the essays that we collected, students clearly have a need for additional practice with specific, focused feedback.

Machine learning approaches can create models for performing detailed analyses of texts but require a large amount of relevant *labeled* training data. This paper has provided an evaluation of Active Learning to determine how effectively it can improve accuracy of the machine learning analysis models while minimizing the costs of annotation. Overall, we found that, especially in early iterations, it was best to choose items that the model was least certain of.

These results suggest some directions for future research. Because the closest-to-hyperplane strategy was initially very good, but later plateaued, we would like to evaluate a hybrid model which initially chooses the least certain instances, then at some point, switches to choosing a mixture of more and less certain items. There are also many other instance selection strategies that could be explored. These have previously been applied to tasks in which, unlike ours, there is a single target classification for items [19]. We would like to explore some of the others that have been used for natural language processing [28].

The co-training approach described above could also be another fruitful way to improve model performance with an even lower cost in terms of additional annotation. It should be noted, however, that at its worst, this might be equivalent to a “dumbed down” version of the farthest-from-hyperplane strategy evaluated here; it would take its predictions (which may be noisy) on the highest confidence items. The advantage of co-training would come from the use of complementary feature sets. The trick would be finding feature sets that are conditionally independent of the target classes.

Finally, all of the methods we have evaluated in this paper assume that entire essays would be annotated and added to the training set. To select an essay for annotation, however, we first evaluate the certainty of the predictions at the sentence level, which is, in turn based on predictions at the word level. Instead of selecting entire essays to add to the training set, we could instead select sentences, phrases or words. This could obviously significantly reduce the additional annotation time. The question is how effective it would be at improving model performance.

References

1. Osborne, J., Erduran, S., Simon, S.: Enhancing the quality of argumentation in science classrooms. J. Res. Sci. Teach. **41**(10), 994–1020 (2004)
2. Achieve Inc.: Next generation science standards (2013)
3. Hastings, P., Britt, M.A., Rupp, K., Kopp, K., Hughes, S.: Computational analysis of explanatory essay structure. In: Millis, K., Long, D., Magliano, J.P., Wiemer, K. (eds.) Multi-Disciplinary Approaches to Deep Learning. Routledge, New York (2018). Accepted for publication
4. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: brat: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations Session at EACL 2012, Avignon, France, Association for Computational Linguistics, April 2012
5. Stenetorp, P., Topić, G., Pyysalo, S., Ohta, T., Kim, J.D., Tsujii, J.: BioNLP shared task 2011: Supporting resources. In: Proceedings of BioNLP Shared Task 2011 Workshop, Portland, Oregon, USA, Association for Computational Linguistics, pp. 112–120, June 2011
6. Goldman, S.R., Greenleaf, C., Yukhymenko-Lescroart, M., Brown, W., Ko, M., Emig, J., George, M., Wallace, P., Blaum, D., Britt, M.: Project READI: Explanatory modeling in science through text-based investigation: Testing the efficacy of the READI intervention approach. Technical Report 27, Project READI (2016)
7. Shermis, M.D., Hamner, B.: Contrasting state-of-the-art automated scoring of essays: analysis. In: Annual National Council on Measurement in Education Meeting, pp. 14–16 (2012)
8. Deane, P.: On the relation between automated essay scoring and modern views of the writing construct. Assessing Writ. **18**(1), 7–24 (2013)
9. Roscoe, R.D., Crossley, S.A., Snow, E.L., Varner, L.K., McNamara, D.S.: Writing quality, knowledge, and comprehension correlates of human and automated essay scoring. In: The Twenty-Seventh International Flairs Conference (2014)
10. Shermis, M.D., Burstein, J.: Handbook of Automated Essay Evaluation: Current Applications and New Directions. Routledge (2013)
11. Dikli, S.: Automated essay scoring. Turk. Online J. Distance Educ. **7**(1), 49–62 (2015)
12. Condon, W.: Large-scale assessment, locally-developed measures, and automated scoring of essays: Fishing for red herrings? Assessing Writ. **18**(1), 100–108 (2013)
13. Riaz, M., Girju, R.: Recognizing causality in verb-noun pairs via noun and verb semantics. EACL **2014**, 48 (2014)
14. Rink, B., Bejan, C.A., Harabagiu, S.M.: Learning textual graph patterns to detect causal event relations. In: Guesgen, H.W., Murray, R.C. (eds.) FLAIRS Conference. AAAI Press (2010)
15. Hughes, S., Hastings, P., Britt, M.A., Wallace, P., Blaum, D.: Machine learning for holistic evaluation of scientific essays. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS (LNAI), vol. 9112, pp. 165–175. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19773-9_17
16. Hughes, S.: Automatic inference of causal reasoning chains from student essays. Ph.D. thesis, DePaul University, Chicago, IL (2018)
17. Wolpert, D.H.: Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
18. Hastings, P., Hughes, S., Blaum, D., Wallace, P., Britt, M.A.: Stratified learning for reducing training set size. In: Micarelli, A., Stamper, J., Panourgia, K. (eds.) ITS 2016. LNCS, vol. 9684, pp. 341–346. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39583-8_39

19. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison (2009)
20. Sharma, M., Bilgic, M.: Most-surely vs. least-surely uncertain. In: 13th International Conference on Data Mining (ICDM), pp. 667–676. IEEE (2013)
21. Ferdowsi, Z.: Active learning for high precision classification with imbalanced data. Ph.D. thesis, DePaul University, Chicago, IL, USA, May 2015
22. Cawley, G.C.: Baseline methods for active learning. In: Active Learning and Experimental Design Workshop in Conjunction with AISTATS 2010, pp. 47–57 (2011)
23. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. **2**, 45–66 (2001)
24. Mirroshandel, S.A., Ghassem-Sani, G., Nasr, A.: Active learning strategies for support vector machines, application to temporal relation classification. In: Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 56–64 (2011)
25. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100. ACM (1998)
26. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
27. Joachims, T.: Learning to Classify Text Using Support Vector Machines - Methods, Theory, and Algorithms. Kluwer/Springer, New York (2002)
28. Olsson, F.: A literature survey of active machine learning in the context of natural language processing. Technical Report T2009:06, Swedish Institute of Computer Science (2009). http://eprints.sics.se/3600/1/SICS-T-2009-06-SE.pdf. Accessed 8 Feb 2017