Leveraging Auxiliary Data from Similar Problems to Improve Automatic Open Response Scoring

by

Raysa Rivera-Bergollo

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

April 2022

APPROVED:

Professor Neil Heffernan, Major Thesis Advisor

Professor Craig Wills, Head of Department

Abstract

As computer-based learning platforms have become ubiquitous in educational settings, there is a growing need to provide teachers with better support in assessing open-ended questions. Particularly in the field of mathematics, teachers often rely on open-ended questions, prompting students to explain their reasoning or thought processes, to better assess students' understanding of content beyond what is typically achievable through other types of problems. In recognition of this, the development and evaluation of automated assessment methods and tools has been the focus of numerous prior works and have demonstrated the potential of such systems to help teachers assess open-ended work more efficiently. While showing promise, many of the existing proposed methods and systems require large amounts of student data to make reliable estimates which may vary in real world application. In this work, we explore whether an automated scoring model trained for a single problem could benefit from auxiliary data collected from other similar problems to address this "cold start" problem. Within this, we explore how factors such as sample size and the magnitude of similarity of utilized problem data affect model performance. We find that the use of data from similar problems not only provides benefits to improve predictive performance by increasing sample size, but the incorporation of such data also leads to greater overall model performance than using data solely from the original problem when sample size is held constant.

Acknowledgements

I would like to thank my advisor, Professor Neil Heffernan, for giving me the opportunity to work in his lab and for his constant encouragement. Thank you to Sami Baral and Professor Anthony Botelho for their endless guidance throughout this process. Thank you also to my reader, Professor Yanhua Li, whose feedback helped shape the final work.

In addition, I would like to thank my parents, John Rivera-Poventud and Grisel Bergollo-Ramos, and brother, John Rivera-Bergollo, for always giving their support and answering my endless calls. Thank you to Cristian Joia and our dog, Bailey, for continuously motivating me. Finally, I would like to thank my friends for listening to my consistent updates as well as providing happy distractions when needed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last several decades, the development of online learning platforms [KC+06, HH14] have revolutionized education in various ways, transforming the instructional practices and learning experiences in both traditional and expanded learning environments. With this, there are both great opportunities as well as a growing need to provide better supports for teachers and students using these platforms. In the domain of mathematics, these online-based learning platforms offer automated supports for assessing students' work as well as providing feedback and support to students. While in the past these supports were generally restricted to closed-ended problems with a finite number of accepted correct responses, advancements in machine learning and natural language processing methods have led to the development of automation tools that even support open-ended work [RM13, CKM16, BBE+21]. As open ended questions in mathematics are widely used by teachers to understand the students' knowledge state and their understanding of a topic, these types of tools have great utility for both teachers and students using these systems.

In recent years, there have been several works focused on the development and improvement of automated methods for assessing student open-ended responses in

mathematics [EBM+20, ZHY+22, YZY17, HBVTN21]. These methods are mostly based on evaluating given student answers based on the historical student answers and the scores given by teachers to such data in the past. Similar to this, prior works from [BBE+21] utilize an unsupervised learning approach that compares given student open-response to historical data based on their semantic similarity to suggest a numeric score. Similar approaches are utilized in recommending feedback messages to teachers to give to these students. As is prevalent in many applications of machine learning, however, many of these approaches are susceptible to the cold start problem, where implementations of such methods may lack sufficient data to make informed estimates; this is certainly the case when first implementing models within a system, but may also extend to cases where systems incorporate new content to which the assessment models have not been previously exposed. While the impact will vary depending on the model and the context, most assessment models require non-trivial amounts of data to make accurate predictions (c.f. [BB01]) which may take time and effort to acquire. However, in cases when there is a newer student response, that has not been encountered in the past, these type of methods often fall behind in suggesting an accurate score/feedback message posing this as the cold start problem.

To help illustrate this problem, consider the sampled statistics pertaining to problems from the widely-used curriculum of Illustrative Mathematics collected from ASSISTments [HH14]. The adoption of open educational resources, such as the curricula of Illustrative Mathematics as well as others such as EngageNY, has become ubiquitous in classrooms across the United States. Looking at the data of Illustrative Mathematics curriculum from ASSISTments (i.e. Table 1.1) reveals that nearly half of the content of this curriculum consists of open-ended problems, with over 70% (17,201) of these being regularly assigned to students by teachers using the platform.

Table 1.1: Open Response Statistics for Illustrative Mathematics Content

Title	**Total Problems**
Total Questions	51006
Total Open-ended Questions	23678
Total Open-ended Questions assigned by teachers	17201
Total Scored Open-ended Question	9868
Total Commented Open-ended Questions	8038
Total Student responses on these questions	15,824,851
Total scored responses	2,116,341
Total commented open responses	536,891

However, just over half of those problems assigned contain any teacher-provided assessment scores (e.g. many teachers assign the problems, but are not scoring the student work). In looking at the distribution of scored student responses across problems in Figure 1.1, we see that a large portion of problems contain few-to-no scored student responses on which to train an automated assessment model; conversely, there are a small number of problems that contain a very large number of scored responses. This makes the development of automated scoring models for these open-ended questions more difficult and likely results in large variations in model performance.

In light of this data, it is important to consider ways to mitigate the impact of this cold start problem to provide support for teachers across a wider range of problems. The concept of transfer learning [TS10] is commonly used as a means of addressing the cold start problem in a variety of prediction tasks. Within the field of education, particularly in the comparatively narrow-scoping of mathematics education, we may be able to leverage data from similar content to improve performance in cases where there would otherwise be insufficient data to train an automated assessment model.

In this work, we seek to explore the effectiveness leveraging auxiliary data in the

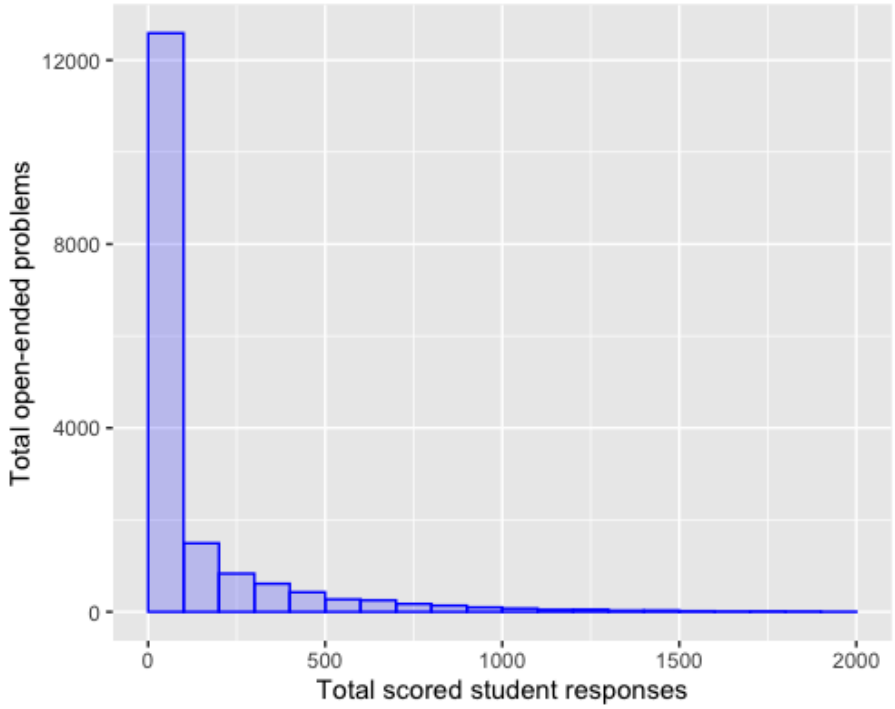

Figure 1.1: Histogram of the distribution of total scored student responses across the open-ended problems in Illustrative Mathematics

form of student responses to similar open-ended problems in the auto-scoring of a new problem with limited labeled data. With the goal of addressing the cold start problem in automatically assessing student open responses, we intend to answer following research questions:[1]

1. Does the addition of new labeled data from a similar open-response problem, improve the predictive performance of single problem based auto-scoring models?

2. Does leveraging data from a similar problem lead to better model performance in comparison to using data from a randomly selected problem?

3. What is the effect of incorporating auxiliary data into the training of an auto-

[1]A portion of this work was also submitted to EDM 2022 where it has been accepted as a poster paper. [RBBBH22]

scoring model and are there any benefits beyond that of increasing sample size?

4. How does the quantity of auxiliary data into the training of an auto-scoring model effect the performance?

Chapter 2

Background

As introduced in the previous section, there has been significant prior research focused on developing automatic assessment methods and tools. In recent years, there have been notable improvements in scoring responses through the use of deep learning techniques for grading both short answers [UU20] and essays [RJO19]. However, most research on scoring open-ended responses has been outside of the domain of mathematics. Automatically scoring mathematical expressions and explanations has several distinctive challenges in comparison to other language-assessment domains due to the interleaving of linguistic and non-linguistic terms (e.g. such as numbers and mathematical expressions). For example, Lan et al. [LVWB15] provides automatic grading and feedback for math open response questions using clustering techniques, but it ignores all text explanations to focus solely on numerical expressions. In the past few years though, there has been progress in the particular task of using language models for mathematics. Erickson and colleagues [EBM+20] compared the performance of different models for scoring math open-ended responses and attempted to establish a benchmark evaluation procedure to evaluate future models. Building on that work, [BBE+21] notably improved performance by us-

ing embeddings produced by Sentence-BERT (SBERT) [RG19] on the same dataset to score student responses. SBERT modifies the pre-trained BERT (Bidirectional Encoder Representations from Transformers) [DCLT19] model to generate sentence-level embeddings and is better suited for comparing semantic similarities. [BBE+21] compares the similarity of a student's response to an open-ended question against previously scored student responses to the same question to generate the score prediction. One of the recurring difficulties in open-ended response grading is the limited quantity of relevant and annotated training data. [CLP21] explores using SBERT with various combinations of content to score unseen questions. For natural language processing problems where data is limited, meta-learning is also becoming a popular approach [Yin20]. Meta-learning attempts to solve a task with limited data after being trained on how to best learn from other tasks. For short answer grading, a meta-learning augmented BERT model (ml-BERT) [WLW+19] has been applied with promising results for biology.

Chapter 3

Preliminary Work

3.1 Methodology

In this chapter, we aim to examine the use of auxiliary data collected from similar problems as a method of addressing the cold-start problem in building automatic assessment models for open-ended mathematics problems. For this, we utilize data collected from ASSISTments in conjunction with the scoring method presented in [BBE+21], known as the "SBERT-Canberra" model. The data and model used in this research, as well as our approach to examining the use of auxiliary data, are described in detail throughout this section.

3.1.1 Dataset

For this study, data[1] consists of student answers to open-ended problems within the ASSISTments. It consists of open-ended responses to problems that have ever been submitted to the system database. For the purpose of this study, we arbitrarily

[1]The data used in this work may contain personally-identifying information but may be shared through an IRB approval process; this process is omitted for blinding purposes but will be included in future versions.

selected a single open response problem within this dataset that contained at least 40 student responses (n=45 for the selected problem) to act as a representative problem. our evaluation will utilize a bootstrapping simulation design using this selected representative problem; while the analyses described in this chapter could be applied to virtually all problems, as will be described, this single problem is sufficient to gain the necessary insights into the utility of using data from other problems. For consistency of terminology, this representative problem will be referred to simply as the "original problem" throughout this work, and will represent the problem for which we would like to train an auto-scoring model (e.g. we will treat it as the problem with insufficient data).

The selected problem pertains to logarithms, and presents the students with the following equation: "$5log(x+4) = 10$"; students are asked to either solve for x and explain their steps to solve or to type "no solution" if no viable solution exists.

In addition to this original problem, we collaborated with a content expert to select a similar open-ended problem for which there was a comparable number of existing labeled student answers (n=43) on which to train a model. This second problem, referred to simply as the "similar problem" throughout the remainder of this work, similarly pertained to logarithms where students were prompted with the following equation: "$log_2(1-x) = 4$"; similar to the original problem, students were asked to solve for x and explain the steps they used to solve or to type "no solution" if no viable solution exists.

While we acknowledge that the selected problems border on the threshold of what might be considered open-ended, much of the content of open curricula pair close-ended and open-ended components within many of their questions (e.g. solve and explain). In this way, the selected problems result in sufficient variation in student answers to examine auto-scoring models, but additionally allowed us to more easily

identify a problem with undeniable similarity both in terms of content and structure; as it is one of our goals to explore how the magnitude of similarity between models affects the effectiveness of model transfer, we are confident in claiming that these problems are in fact similar.

Finally, outside of these two problems, we remove any problem from the remaining dataset containing fewer than 10 labeled student responses. As part of our analyses, we will be sampling random problems and performed this filtering step to mimic a practical application where such problems would not be considered sufficient in providing auxiliary data (arguably, we would in practice even choose a higher threshold, but wanted to utilize as broad, representative dataset with which to conduct our analyses).

Only minor preprocessing was performed on the data to match the same format as was explored in the prior work from which the SBERT-Canberra model was derived [BBE+21]. These steps included the removal of HTML tags that existed in some student responses as well as other special characters and references to images. As was observed in prior works [EBM+20, BBE+21], teacher-provided scores follow a 5-point integer scale ranging from 0, indicating poor performance, and 4, indicating high performance. While ordinal in nature, this scale is converted to a 5-valued categorical one-hot encoded vector and modeled as a multi-class prediction task (i.e. the model treats each score as a mutually-exclusive label). While we acknowledge that the ordinal relationships are lost by representing the labels in this way, we follow this procedure to use the model presented in that prior work.

3.1.2 Model

The SBERT-Canberra model used in this work follows a simple similarity-ranking procedure to generate its predictions. When producing a prediction for a given

student response, it first applies SBERT to generate a high-dimensional feature embedding that describes the response as a whole; this method is intended to capture semantic and syntactic meaning within this embedding, such that similar responses would be mapped to closer points within the embedding space. The SBERT embedding for this student response is compared to SBERT embeddings corresponding to a pool of historic labeled student responses. Using the Canberra distance measure [JRVF09], the score for the historic response corresponding to the smallest distance (i.e. the most similar response) is used as the score prediction. The intuition behind it being that similar answers to the same problem would have the same score. While rather simplistic, particularly as there is no "training" involved in the traditional machine learning sense, we chose to use this model as 1) it outperformed existing benchmarks in assessing student responses in mathematics [BBE+21], 2) as no training is involved, we do not need to optimize hyperparameters, and 3) the model performance is directly linked to the scale and diversity of the historic responses.

3.1.3 Model Evaluation

To examine the use of auxiliary data, we conduct 2 analyses that each compare the SBERT-Canberra model with 3 different training sets. The analyses follow a bootstrapping procedure which samples with-replacement from the available pool of data at increasing intervals. For example, we will observe how well the model performs when trained on just one sample, then two, then three, etc. until the true sample size of the problem is reached (or close to it). At each interval, student responses are randomly sampled to train and evaluate the model using a 10-fold cross validation, where sampling is conducted within the training folds. Since the original problem has 45 scored student responses, the bootstrapped sampling is conducted among 9 training folds and then the model is evaluated on the 10th holdout fold

(and repeated for all folds). This entire process is then repeated 25 times, with the model performance being averaged across these iterations (to reduce noise caused by unlucky sampling).

To evaluate the scoring results, the area under the curve, AUC, (calculated using the simplified multi-class calculation of ROC AUC described in [HT01]) is used as the primary metric to compare the model's predicted score of a student response to the actual score to the student response that was provided by a teacher. For a larger understanding of the performance, RMSE (the root of the average squared errors when comparing the ordinal predictions and the integer labels) and multi-class Cohen's Kappa (measures the inter-rater agreement) were also calculated. Although we focus our later discussion primarily on AUC, the patterns that emerge are consistent with those found with RMSE and Kappa.

The three models are distinguished based on the data used to produce predictions. The *Baseline Model* uses only student responses from the original problem; the number of responses made available to the model will be varied at increasing intervals. The *Similar Problem Model* uses a combination of student responses from the original problem as well as auxiliary responses sampled from the similar problem. Finally, the *Random Problem Model* uses a combination of student responses from the original problem as well as student responses sampled from 5 randomly-selected problems from the remaining dataset; per design and due to the scale of the data used, it is very unlikely for these problems to be similar to the original problem, allowing for comparisons to be made in regard to differing magnitudes of similarity.

Varying Original Problem Sample Size

The first analysis replicates a real-world scenario where we may have a small number of labeled samples for the original problem, but a larger number of samples that may

be leveraged from other similar and non-similar problems. For each bootstrapping interval, we randomly sample data from the original problem ranging from 0 to 40 in increments of 2. As the similar problem has 40 labeled student responses, we similarly randomly sample 40 scored responses from the 5 random problems to create a comparable set. While the Baseline model is limited to only the 0 to 40 original problem samples, both the Similar Problem Model and Random Problem Model is able to use 40-80 samples over the set of intervals (initially using only samples from the other problems and adding an increasing number from the original problem with each interval.

Due to the large variations in sample sizes across problems within the dataset, we sample student responses for the Random Problem Model using a stratified selection method. This helps to ensure that the selected 40 responses are spread evenly over the 5 randomly selected problems rather than from just one of those problems if there is a large difference in sample size (i.e. in the case that the problem with 2000 scored responses is randomly selected with a problem that has only 10). From the 5 randomly-selected problems, 8 scored student responses are randomly selected per iteration in the interval and they compose the 40 samples to supplement the training data from the original problem.

The average performance of each of the three models is then plotted with 95% confidence intervals calculated over the 25 repeated runs per interval.

Varying Sample Proportions

As it is hypothesized that the largest benefit of using auxiliary data is in the added sample size, we conduct a second bootstrapping analysis that observes a constant sample size across intervals while varying the proportion of data used from the original problem. From this analysis, all models (except for the baseline) utilize the

same number of samples, allowing us to observe how the source of content affects model performance independent of data scale.

In this case, the number of training samples is held consistent at 40 scored student responses and the percentage intervals range from 0% to 100% of the original problem at 10% increments. In other words, at the first interval, the 40 samples are composed of only responses from other problems (either from the similar or random problems), while at the last interval, all 40 samples are composed only of the original problem. This proportion is then interpolated between these extremes over each interval. It is hypothesized that the best model performance would be exhibited by each model at the 100% interval, where we use all the data available from the original problem, as this is when the data is most closely related to the test set. In keeping consistent with the previous analysis, for each increment of training data from the original problem, 10-fold cross validation is run 25 times and the reported metric (AUC, RMSE, Kappa) is the average across those runs. The same sampling procedure for the Random Problem Model as was conducted in the previous analysis is utilized here as well. As the Baseline Model only utilizes data from the original problem, we are unable to maintain a consistent sample size across intervals. For comparative purposes, we simply increase the training sample size following the increasing percentage (i.e. using 0 samples, then 4 corresponding with 10%, then 8, etc.).

3.2 Results

3.2.1 Varying Original Problem Sample Size

The performance of each of the models when varying the original problem sample size is reported in Figure 3.1, with the measures of RMSE and Kappa also depicted

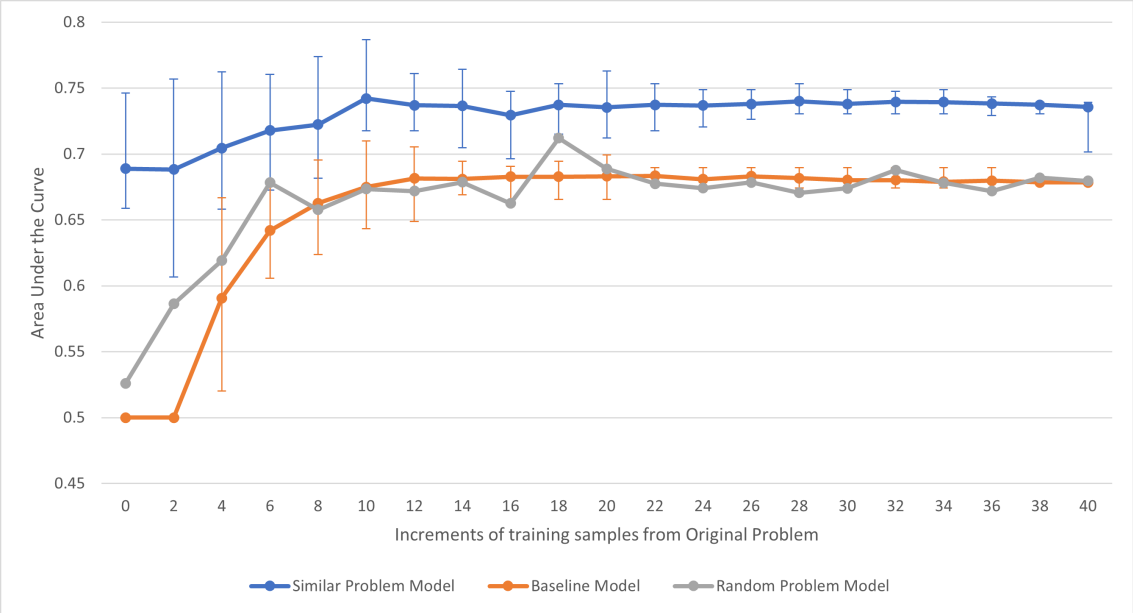

Figure 3.1: Average AUC while varying original problem sample size.

in Figures 3.2 and 3.3, respectively. For interval 0, no training data was provided for the baseline model so there is no recorded performance for comparison for both kappa and RMSE; while we acknowledge that a majority class or other value could have been imputed here to generate some value, but we felt this was unnecessary to observe the performance trends as is our goal.

In regard to the Baseline Model, when the increment of training samples from the original problem is 0, the average AUC of the baseline model is assigned to be 0.5 which is equivalent to chance. The lowest average AUC occurs, rather unsurprisingly, when there are very few samples from the original problem for the model to use. The highest average AUC for the baseline model (0.683) occurs when it is trained with 22 samples from the original problem. However, after just 12 samples from the original problem as the training data, the baseline model has an average AUC equal to 0.682 that seems to converge between 0.678 and 0.683 as the baseline model is trained with increasing amounts of samples from the original problem.

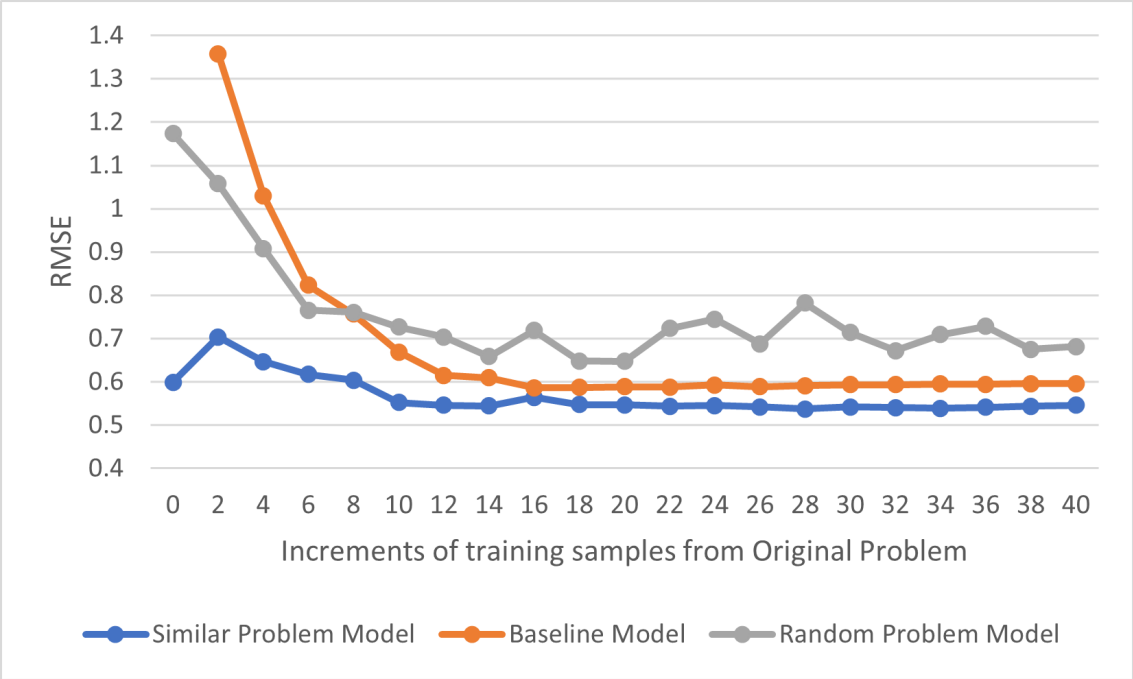

Figure 3.2: Average RMSE while varying original problem sample size.

Observing the Similar Problem Model, when using 40 random samples from the similar problem to supplement the training samples from the original problem, the modified model outperforms the average AUC of the baseline model across every increment of training samples from the original problem; this difference is also statistically reliable across a majority of intervals as determined by comparing the confidence intervals. This model outperforms the baseline model by approximately 0.073 in terms of average AUC per interval. The worst average AUC for the modified similar problem model is 0.688 and it occurs when trained with 40 samples from the similar problem and 2 samples from the original problem. The best average AUC for the modified similar problem model is 0.742 when the model is trained with 40 samples from the similar problem and 10 samples from the original problem. Beyond 8 samples from the original problem, the model arguably converges with an average AUC between 0.662 and 0.740.

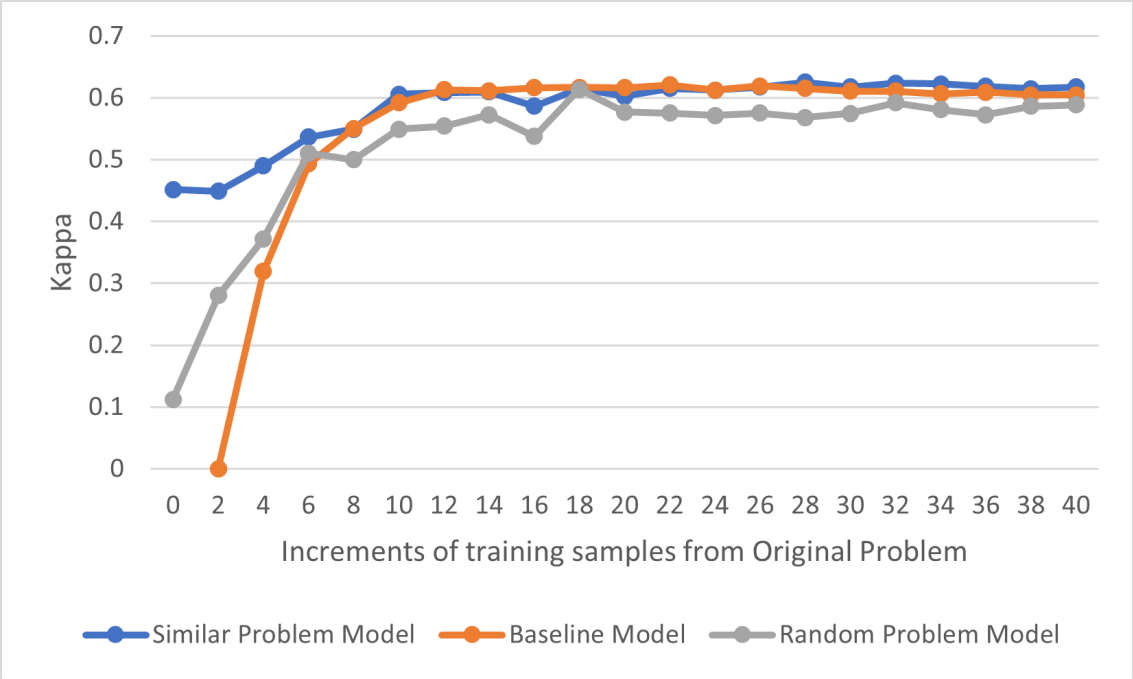

Figure 3.3: Average Kappa while varying original problem sample size.

Finally, regarding the Random Problem Model, when using 40 random samples split evenly from 5 random problems to supplement the training samples from the original problem, the model outperforms the average AUC of the baseline model across 43% of the increments tested. At an average difference of just 0.007 in terms of average AUC per interval, very little meaningful difference is observed between the Random Problem Model and the Baseline Model. It is worth noting that the performance of the Random Problem Model does outperform the Baseline over the initial intervals when sample size is the smallest, suggesting that even randomly-selected problems may provide benefit. However, this model also exhibited large variations in performance, leading us to omit the error bars to improve the readability of the figure; this variation is presumably attributable to the random selection of problems with varying magnitudes of similarity to the original problem. The best average AUC for the modified random problem model is 0.712 when the

model is trained with 40 samples from 5 random problems and 18 samples from the original problem. Beyond 10 samples from the original problem, the model converges with an average AUC between 0.663 and 0.712.

Across the three models, the RMSE and Kappa follow similar trends, with the Similar Problem Model performing the best on average of the methods. While the general trend remains, it is the case that the difference between the methods, particularly by the later intervals, are much smaller than those observed in regard to AUC. For Kappa, for example, all three models seemingly converge by an original problem sample size of 6, but does observe differences in the early intervals.

3.2.2 Varying Sample Proportions

The performance of each of the models when holding sample size constant and varying the sample proportion is reported in Figure 3.4, with the measures of RMSE and Kappa also depicted in Figures 3.5 and 3.6, respectively.

In observing the Baseline Model AUC performance, when the percentage composition of training samples from the original problem is 0%, the average AUC of the baseline model is found to be 0.5, again unsurprisingly as the model has no samples on which to base its predictions. The highest average AUC for the baseline model (0.685) occurs when it is trained with 30% of the possible training samples, which is equivalent to 12 training samples, from the original problem. However, after that percentage composition interval, the remaining percentage composition intervals (between 16 and 40 training samples from the original problem), the average AUC seems to stabilize in a somewhat downward trend between 0.678 and 0.684 even though the baseline model is being trained with increasing amounts of samples from the original problem.

An interesting trend emerged in regard to the Similar Problem Model. When

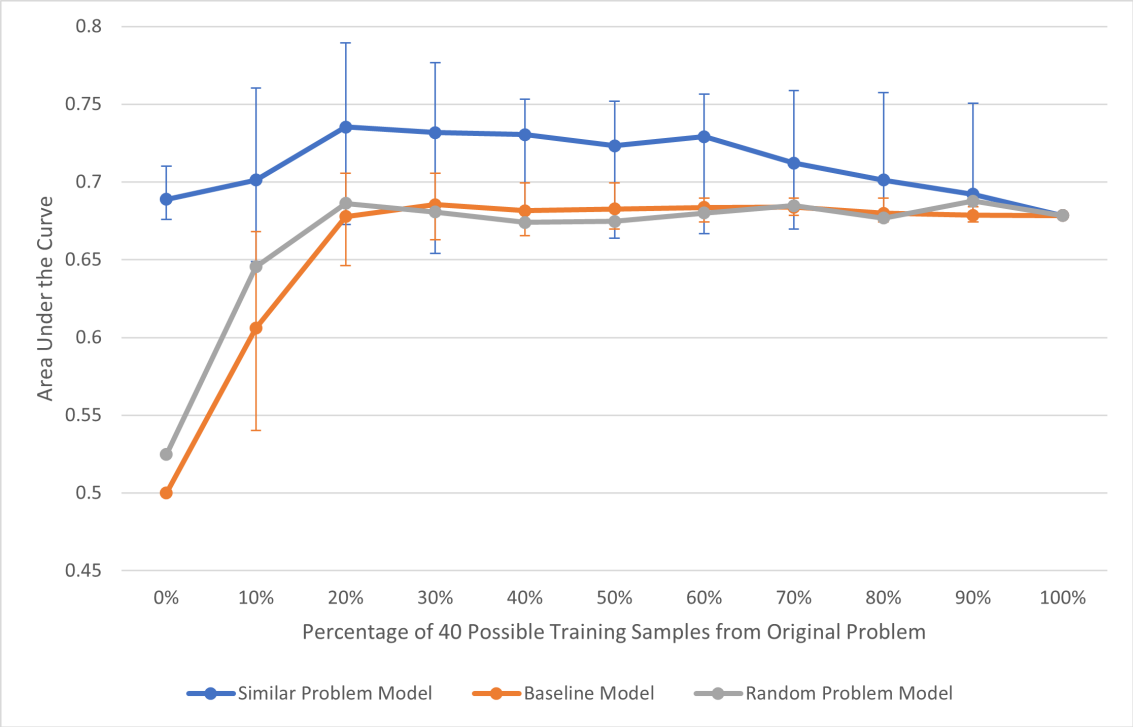

Figure 3.4: Average AUC while varying sample proportion.

using 40 total training samples (and keeping this constant) with some percentage of samples from the original problem and the remaining samples from the similar problem, the modified model outperforms or equals the average AUC of the baseline model across every increment of training samples from the original problem. The modified model that uses training samples from the similar problem and the original problem outperforms the baseline model by around 0.053 in terms of average AUC per interval. The worst average AUC for the similar problem model overall is 0.678 and it occurs when trained with all 40 samples coming from the original problem. However, the worst average AUC for the similar problem model when using some non-zero percentage of the training samples from the similar problem is 0.689 and occurs when trained with all 40 samples coming from the similar problem. The best average AUC for the similar problem model is 0.735 when the model is trained with 20% of the 40 training samples coming from the original problem and the

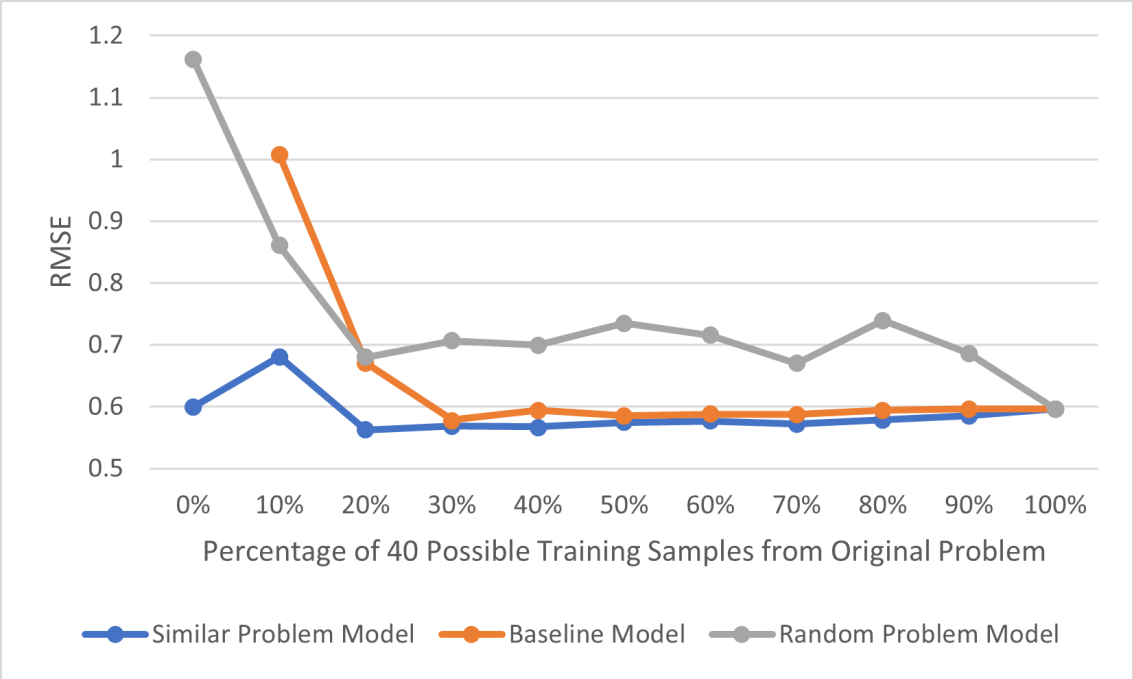

Figure 3.5: Average RMSE while varying sample proportion.

remaining 80% of training samples coming from the similar problem. After the peak performance in terms of average AUC, the model's performance lessens as the percentage of training samples coming from the original problem increases.

The Random Problem Model follows closely with the performance of the baseline. When using 40 total training samples with some percentage of samples from the original problem and the remaining samples from 5 random problems, the modified model outperforms or equals the average AUC of the baseline model across 54% of the percentage composition increments tested. The random problem model that uses training samples from the 5 random problems and the original problem outperforms the baseline model by around 0.005 in terms of average AUC per interval. The worst average AUC for the random problem model overall is 0.525 and occurs when trained with all 40 samples coming from the 5 random problems. The best average AUC for the random problem model is 0.688 when the model is trained with 90%

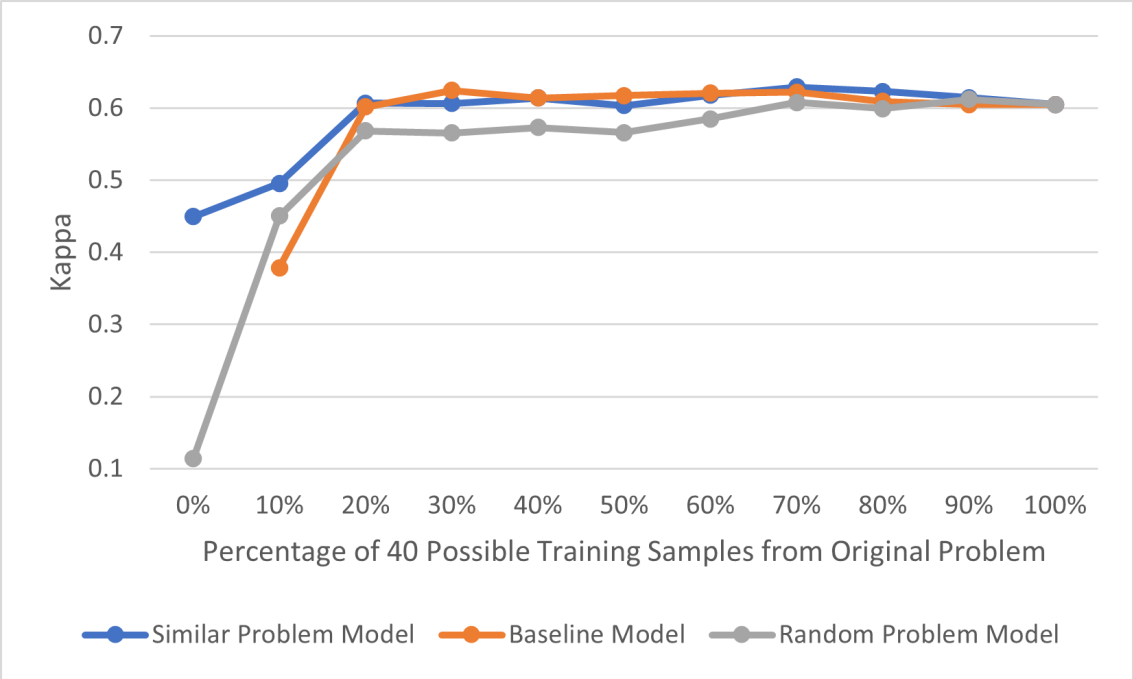

Figure 3.6: Average Kappa while varying sample proportion.

of the 40 training samples coming from the original problem and the remaining 10% of training samples coming from the 5 random problems. After the percentage composition interval of 20% of the training samples coming from the original problem and the remaining 80% coming from the 5 random problems, the performance seems to stabilize between 0.674 and 0.685 even though it is being trained with increasing amounts of samples from the original problem.

3.3 Discussion

Looking deeper into the results of both analyses, we identify consistent trends that are discussed in this section.

The Baseline Model across both analyses provide insights into the current implementation of auto-scoring models. While the performance of the SBERT-Canberra model will likely vary across problems, we can observe here that the model converges

within a relatively small set of samples; after 12 or more samples from the original problem as the training data, the baseline model converges in terms of average AUC performance. It does seem to matter, however, which samples are used to train the model. We can see in both analyses that the Baseline Model's confidence intervals decrease with more samples. The relatively wide bounds over low sample sizes suggests that there are subsets of training samples that are better than others. This is not very surprising as the diversity of data is often considered just as important as the scale in many machine learning applications [HSHA14].

There is a similar trend in regard to the scale of confidence bounds in regard to the Similar Problem Model. Although the average AUC performance stabilized after 10 or more samples, the confidence intervals continued to shrink in the first analysis, but remained relatively constant in the second analysis varying proportion. In both analyses, however, we see consistent, if not statistically reliable differences in comparison to the Baseline Model. In addressing our first research question, this finding suggests that the use of auxiliary data can lead to notable benefits to model performance. We see that in the first analysis that the added sample size leads to notable performance when there are few training samples, but this trend remains through all intervals. While our initial hypothesis was that this benefit would likely be attributable to increased sample sizes, the trend of this Similar Problem Model in the second analysis varying proportion contradicts that hypothesis. While this model does still outperform the baseline, as sample size is held constant, this cannot be the contributing factor to the differences we observe in that analysis. While we expected to observe the final interval of Figure 3.4 to be an upper bound for model performance, we found that the inclusion of data from a similar problem added benefits that extend even beyond the impact of sample size. This finding addresses our third research question, but still remains inconclusive as to what

benefit is provided. It is possible, for example, that the auxiliary data acts as a regularization method (c.f. [Bou98]), but the analyses conducted here are only able to rule out sample size being the contributing factor. These findings further confirm that scoring models can be improved upon when provided with more varied training samples from both the problem it is trying to score and similar problems rather than only being trained from samples of the original problem. Even when trained with the same number of samples, the Similar Problem model's average AUC decreases after a peak training percentage composition which supports the theory that the quality of the training samples from the original problem are less than the quality of the combined samples.

What is perhaps most surprising about this comparison in the second analysis is that the model trained from 100% of data from the similar problem seems to outperform the model trained from 100% of the original problem. We believe that this is an artifact of the selected problems and the level of similarity that they exhibit. As such, we would not expect this finding to extend to every open-ended problem, but rather could extend to a subset where there is strong similarity between problems both in terms of content and the structure of student responses; this is the scenario where we believe this method would provide the most benefit.

This is particularly the case considering that the same level of benefit was not observed in regard to the Random Problem Model across the two analyses. Due to the nature of choosing random problems, the large variance in confidence intervals is expected; while these bounds were omitted from the earlier figures, they can be seen in Figures 3.7 and 3.8 pertaining to the first analysis varying original problem sample size. In essence, these error bars appear to span the gap that is seen between the Similar Problem Model and the Baseline Model. Our hypothesis, as previously introduced, is that the added benefit is likely correlated with the mag-

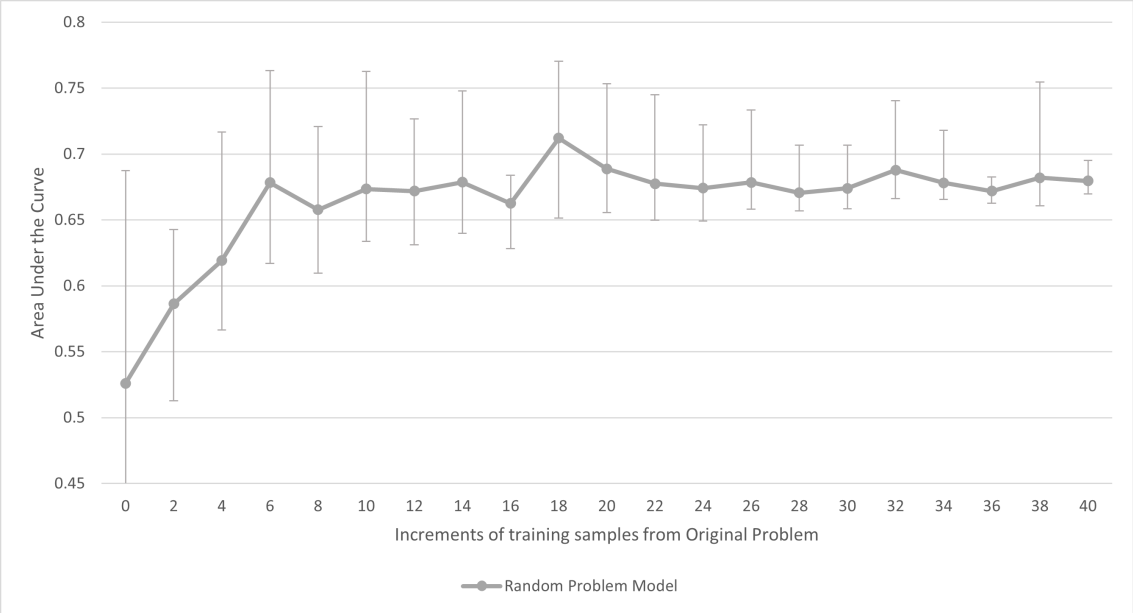

Figure 3.7: Average AUC with confidence intervals for the Random Problem Model while varying original problem sample size.

nitude of problem similarity. Even if this hypothesis is flawed, we are seeing that certain subsets of problems lead to better performance than others, emphasizing the importance in selecting suitable problems from which to draw auxiliary data (e.g. selecting any problem with sufficient sample size may not provide benefits to performance). In light of this finding, we can address our second research question in that problem similarity, loosely defined, does seem to impact performance. In observing the measures of RMSE and Kappa in regard to this Random Problem Model, it would seem that a poor choice of problem may lead to reduced performance than would otherwise be achievable using just data from the original problem (and this was consistent across both analyses).

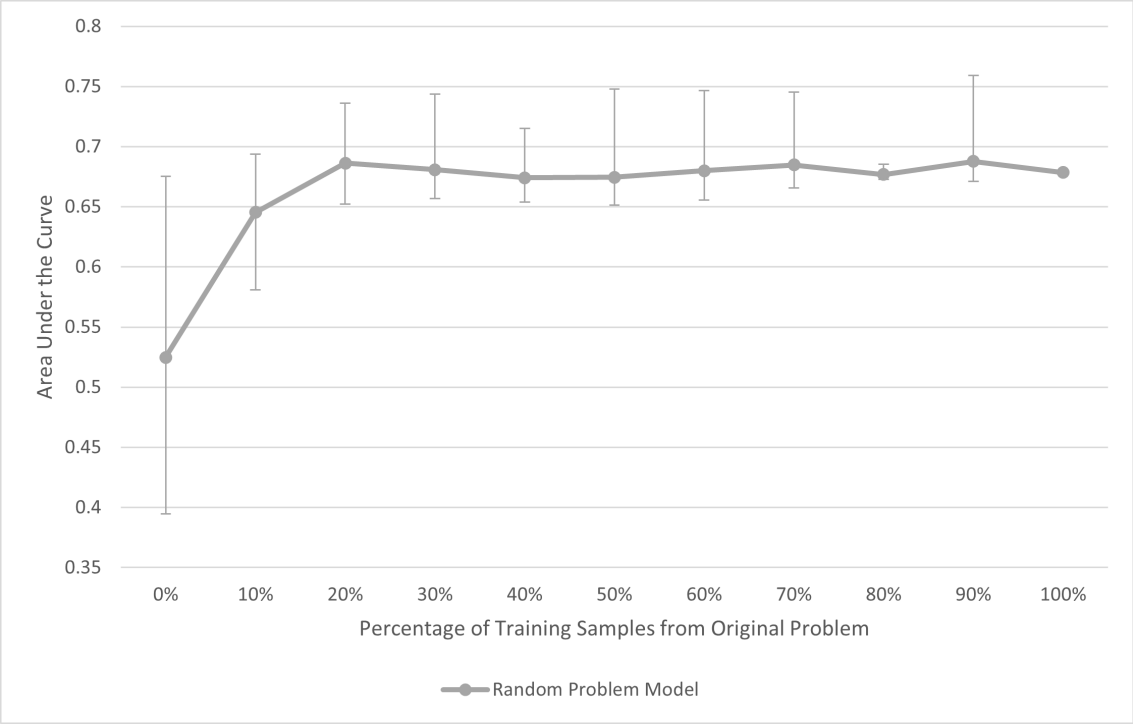

Figure 3.8: Average AUC with confidence intervals for the Random Problem Model while varying sample proportion.

Chapter 4

Extension of Previous Work

4.1 Methodology

We aim to further explore the use of auxiliary data collected from similar problems to supplement the data from the original problem to train the models by modifying the approach of varying sample size from Chapter 3.

4.1.1 Dataset

To ensure consistency across studies, the dataset, the data pre-processing and teacher-provided score encoding process are the same as that from 3.1.1. The only difference is the choice of original problem and similar problem. We explore the same pair of problems from 3.1.1 as well as two different randomly chosen pairs of similar problems. The two additional pairs of similar problems chosen vary in their level of similarity and they are considered to be more open-ended than the original similar problem pair because they ask students to explain their reasoning. Due to this, the answers are more likely to be combinations of words and mathematical expressions rather than only steps to equation solving. Therefore, we feel confident

that these three similar problems pairs provide data that is more representative to the larger set of problems.

Solving Logarithmic Equations

These are the problems used in the main study as described in 3.1.1. As a brief reminder, the designated original problem has 45 scored student responses. It asks students to solve for x and explain their steps to solve or to type "no solution" if no viable solution exists to the equation: "$5log(x+4)=10$". The designated similar problem asks the same question but to the equation: "$log_2(1-x)=4$". The similar problem has 43 scored student responses.

Determining Exponential Decay

The selected problem pertains to interpreting exponential growth or decay and has 42 scored student responses. It presents the students with the following formula: "$f(t)=\frac{2}{3}(\frac{1}{3})^t$". Students were previously asked to identify the initial value in the formula and to determine whether the formula models exponential growth or exponential decay. This specific problem asks students to justify their previous responses regarding the formula. This will be referred to as the "original exponential problem" throughout the remainder of this work.

In addition to this original exponential problem, we selected a similar open-ended problem for which there was a comparable number of existing labeled student answers (n=42) on which to train a model. This second problem, referred to simply as the "similar exponential problem" throughout the remainder of this work, also pertained to exponential growth or decay where students were prompted with the following formula: "$f(t)=2(\frac{2}{5})^t$". Like the original exponential problem, students were asked to justify their reasoning on whether the equation models exponential

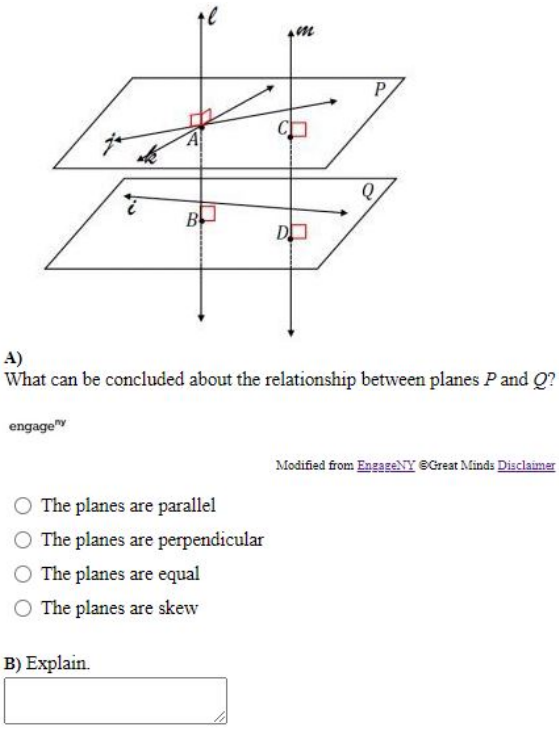

Figure 4.1: Image of the parallel original problem directly from ASSISTments

growth or exponential decay.

Determining Parallelism

The designated original problem of this pair pertains to interpreting and explaining the relationship between planes P and Q as seen in figure 4.1. It has 125 scored student responses.This will be referred to as the "original parallel problem" throughout the remainder of this work.

Along with this original parallel problem, we selected a similar open-ended problem for which there was a comparable number of existing labeled student answers (n=124) on which to train a model. This second problem, referred to simply as the "similar parallel problem" throughout the remainder of this work, involved interpreting and explaining the relationship between lines l and m as seen in the diagram in figure 4.1. Although the original parallel problem is about planes and the similar

parallel problem is about lines, we consider the questions to be similar in nature due to the use of the same diagram, the same relationship between items (parallel) and the process for determining the relationship between items is identical regardless of being about lines or planes.

4.1.2 Model

As described in chapter 3.1.2, the SBERT-Canberra model is used for further analysis.

4.1.3 Model Evaluation

To explore the benefits in using auxiliary data, we conduct exploratory analyses that compare the SBERT-Canberra model when trained with various amounts of auxiliary data. In order to observe the generalized behavior, the same process was performed using 3 different pairs of similar problems. Those problems are described in 4.1.1. The analyses follow a similar bootstrapping procedure from 3.1.3. However, instead of only having the number of training samples from auxiliary data remain consistent, this approach trains with varied quantities of auxiliary data with-replacement from the available pool of data from the associated similar problem at increasing intervals. For example, we will observe how well the model performs following the procedure from 3.1.3 when trained with 0 samples from auxiliary data, then 5 samples from auxiliary data and so on in increments of 5. In doing this approach, we can continue to analyze the performance using only student responses from the associated original problem as well as observe how much auxiliary data is necessary to see changes in performance. Likewise as described in 3.1.3, we repeatedly evaluate the models using 10-fold cross validation so the performance reported by the model at each interval is the average of the 25 iterations performed. Further-

more, the same performance metrics: AUC, RMSE and multi-class Cohen's Kappa were calculated.

4.2 Results

Across all pairs of similar problems, for interval 0, no training data was provided for the model representing 0 training samples from its respective similar problem so there is no recorded performance for comparison for both kappa and RMSE. Furthermore, the models where the training data consists only from its respective original problem are colored in black to make its performance easily seen. These can be considered the baseline model performance for their respective problem pairs. Otherwise, the lines gradually darken in shades of red as the model trains with increasing amounts of auxiliary data from their respective similar problem.

4.2.1 Solving Logarithmic Equations

The performance of models using varying amounts of samples from the similar problem while also varying the original problem sample size is reported in Figure 4.2, with the measures of RMSE and Kappa also depicted in Figures 4.3 and 4.4, respectively. For this problem specifically, the model representing 0 similar problem samples is equivalent to the Baseline Model described in 3.1.3 with results captured in 3.1.3. Likewise for this particular problem, the model representing 40 similar problem samples is equivalent to the Similar Problem Model described in 3.1.3 with results also captured in 3.1.3.

As observed in Figure 4.2, all models with similar problem samples had significantly better performance in terms of AUC across every increment of samples from the original problem. With as few as 5 training samples from the similar problem

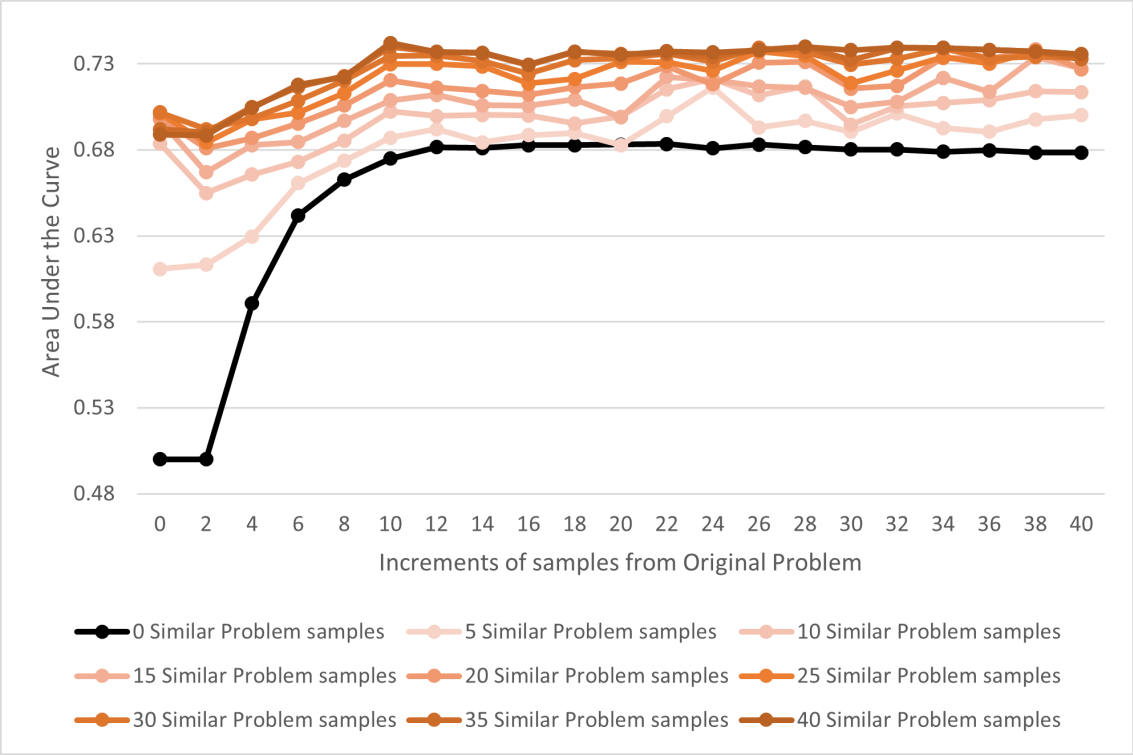

Figure 4.2: Average AUC while varying both the original problem sample size and the similar problem sample size.

to supplement the samples from the original problem, there is noticeable improvements over the model without any samples from the similar problem, the respective baseline model. The model trained with only 5 similar problem samples outperformed the respective baseline model by about 0.024 in terms of average AUC per interval. Each model trained with samples from the similar problems outperforms the respective baseline model by an average of 0.057 in terms of average AUC per interval. Furthermore, each model trained with samples from the similar problem outperforms the model with the next lowest increment of training samples from the similar problem by approximately 0.009 in terms of average AUC per interval. For example, the model trained with 10 similar problem samples outperforms the model trained with 5 similar problem samples by 0.018 in terms of average AUC per interval. The worst average AUC for a model trained with samples from the similar

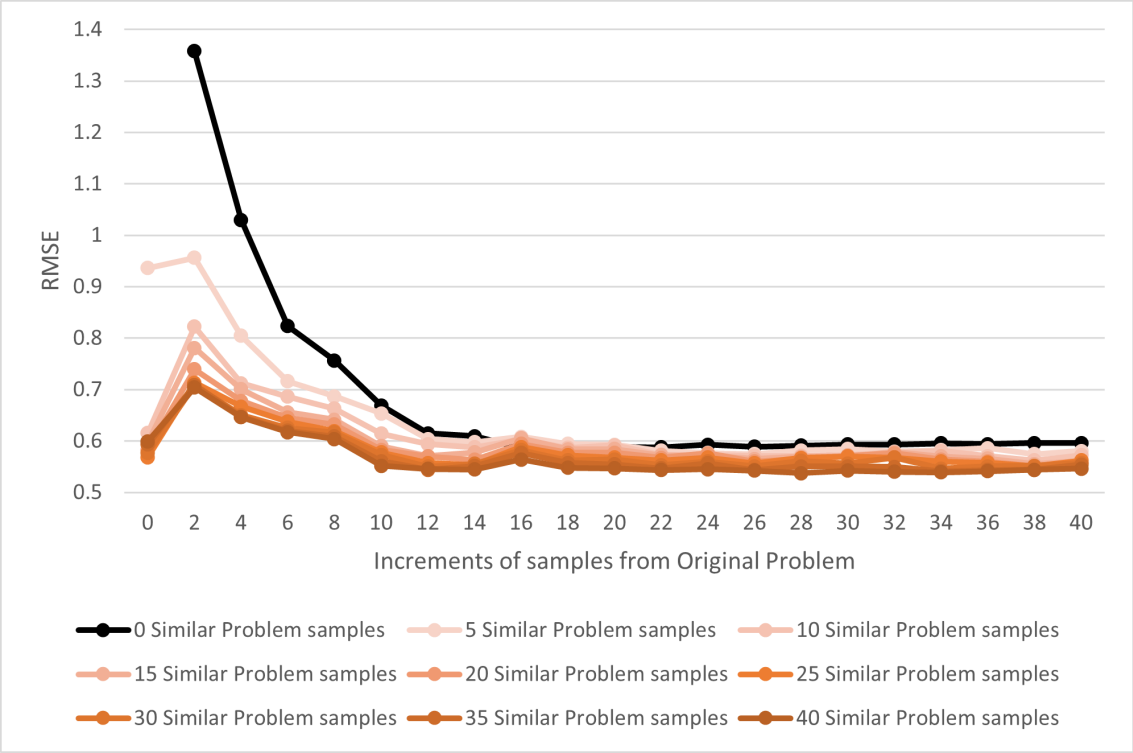

Figure 4.3: Average RMSE while varying both the original problem sample size and the similar problem sample size.

model is 0.611 and it occurs when trained with 5 similar problem samples and 0 samples from the original problem. As was first discovered in 3.1.3, the best average AUC for models using similar problem samples remains 0.742 when the model is trained with 40 similar problem samples and 10 samples from the original problem. Likewise as noticed in 3.1.3, all models generally converge beyond 8 samples from the original problem.

Across the different models, the RMSE and Kappa follow similar trends, with the model trained with 40 similar problem samples performing the best on average of the methods. In this case, the respective baseline model performs the worst and the models generally improve as they are trained with more samples from the similar problem. While this trend remains, the difference between the methods are much less noticeable, particularly by the larger intervals, when compared to AUC.

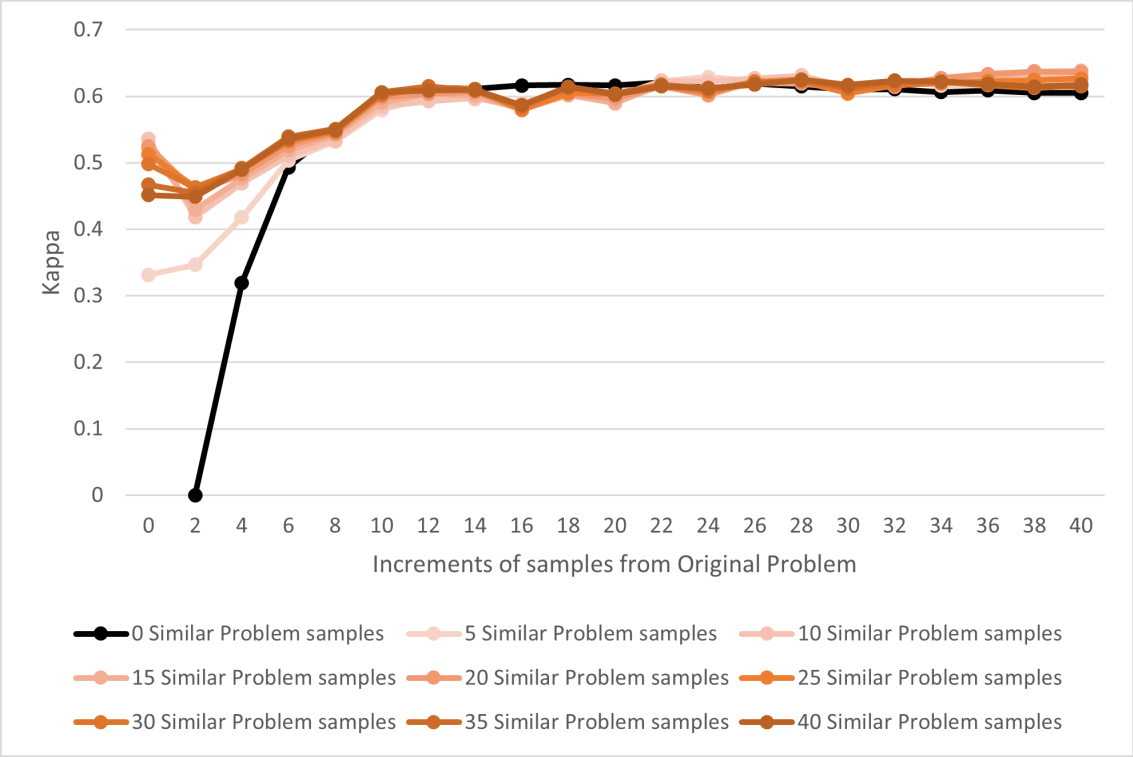

Figure 4.4: Average Kappa while varying both the original problem sample size and the similar problem sample size.

Specifically for Kappa as seen in Figure 4.4, there are a few intervals where the baseline model outperforms those trained with samples from the similar problem. This occurs with the models trained with samples from the similar problem at both 16 and 20 training samples from the original problem. However, before and after those points, all of the models seem to converge. The improvements using training samples from the similar problem for these performance metrics are most seen in the early intervals.

4.2.2 Determining Exponential Decay

The performance of models with varying amounts of training samples from the similar exponential problem when varying the original exponential problem sample

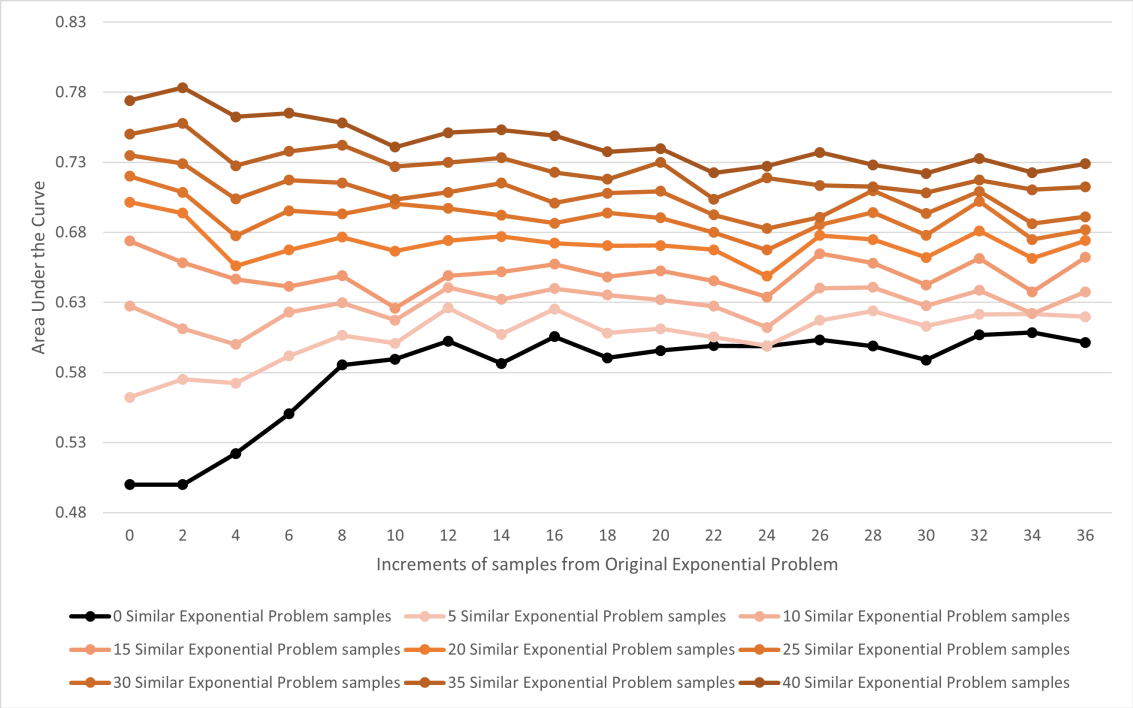

Figure 4.5: Average AUC while varying both the original exponential problem sample size and the similar exponential problem sample size.

size is reported in Figure 4.5, with the measures of RMSE and Kappa also depicted in Figures 4.6 and 4.7, respectively.

As captured in Figure 4.5, all models with similar exponential problem samples had significantly better performance in terms of AUC across every increment of samples from the original exponential problem. The model trained with just 5 similar exponential problem samples outperformed the respective baseline model by about 0.025 in terms of average AUC per interval. Each model trained with samples from the similar exponential problem outperforms the respective baseline model by an average of 0.096 in terms of average AUC per interval. Furthermore, each model trained with samples from the similar exponential problem outperforms the model with the next lowest increment of training samples from the similar exponential problem by approximately 0.020 in terms of average AUC per interval.

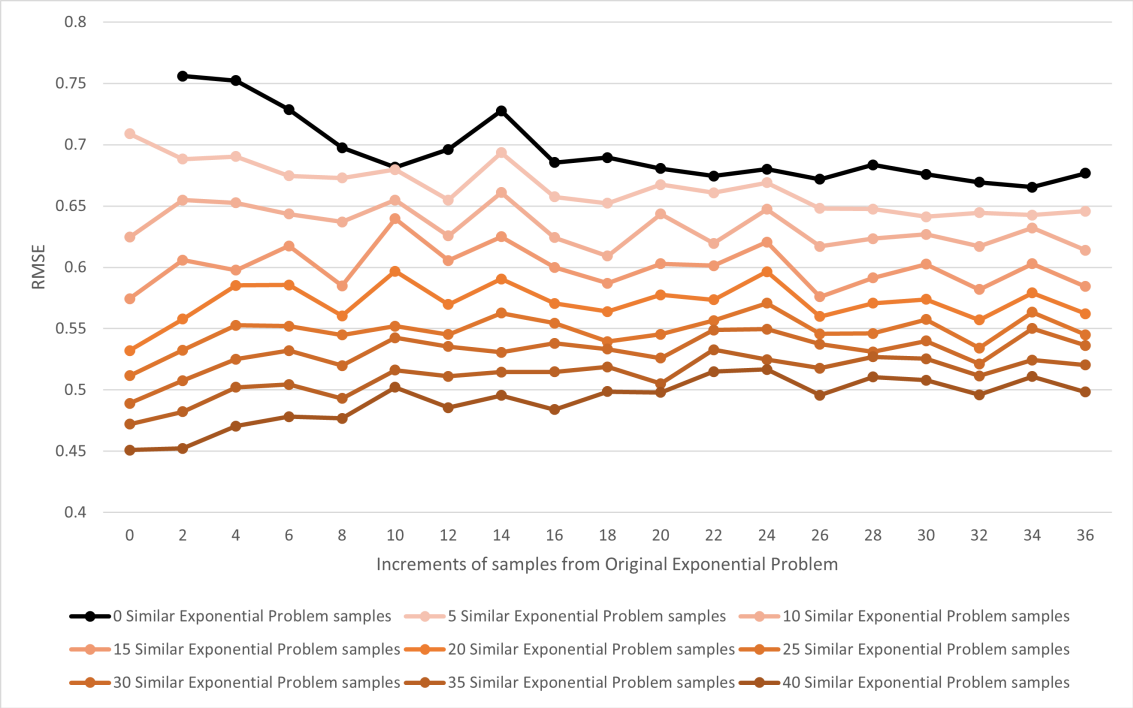

Figure 4.6: Average RMSE while varying both the original exponential problem sample size and the similar exponential problem sample size.

For example, the model trained with 20 similar exponential problem samples outperforms the model trained with 15 similar exponential problem samples by 0.021 in terms of average AUC per interval. The worst average AUC for a model trained with samples from the similar exponential model is 0.606 and it occurs when trained with 5 similar exponential problem samples and 0 samples from the original exponential problem. The best average AUC for models using samples from the similar exponential problem is 0.783 when the model is trained with 40 similar exponential problem samples and 2 samples from the original problem. Furthermore, all models generally converge beyond 8 samples from the original exponential problem.

Across the different models, the RMSE and Kappa have a similar behavior, with the model trained with 40 similar exponential problem samples performing the best on average of the methods. The respective baseline model performs the worst and

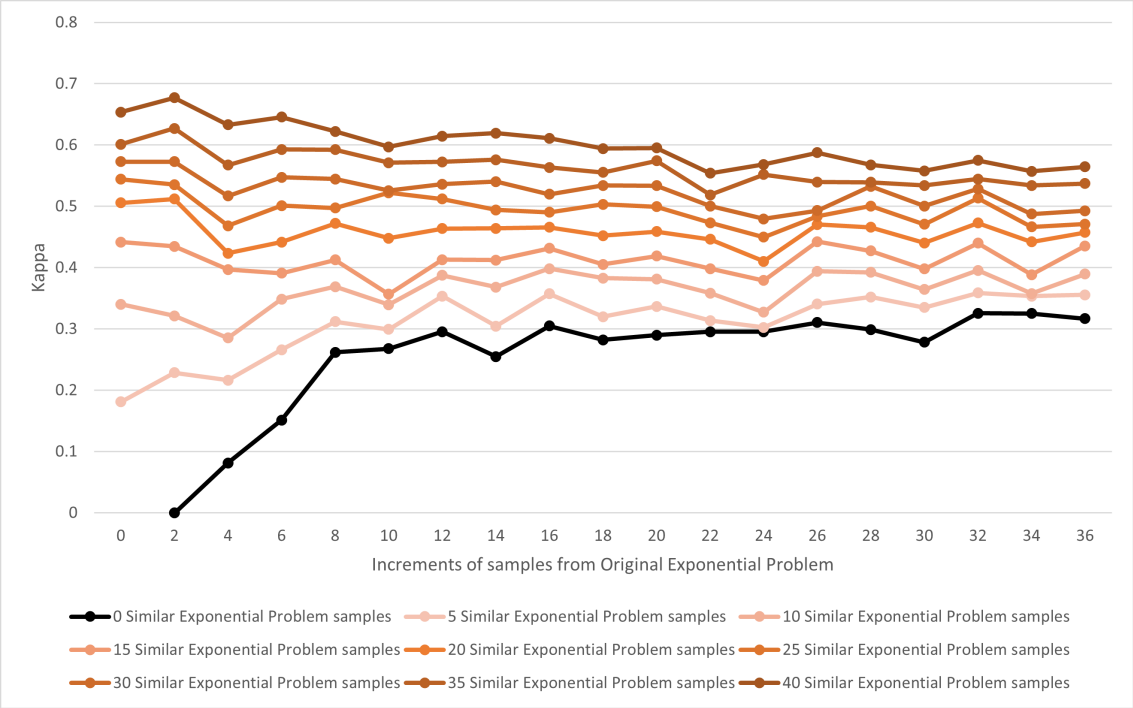

Figure 4.7: Average Kappa while varying both the original exponential problem sample size and the similar exponential problem sample size.

the models improve as they are trained with increasing numbers of similar exponential problem samples. While this pattern is consistent, the difference between the methods continues to lessen by the larger intervals especially when compared to AUC.

4.2.3 Determining Parallelism

The performance of models with varying amounts of training samples from the similar parallel problem when varying the original parallel problem sample size is reported in Figure 4.8, with the measures of RMSE and Kappa also depicted in Figures 4.9 and 4.10, respectively. Due to larger number of samples available from both the original parallel problem and the similar parallel problem, the figures maintain increments of 2 but only show ticks at intervals of 10 to increase readability. In

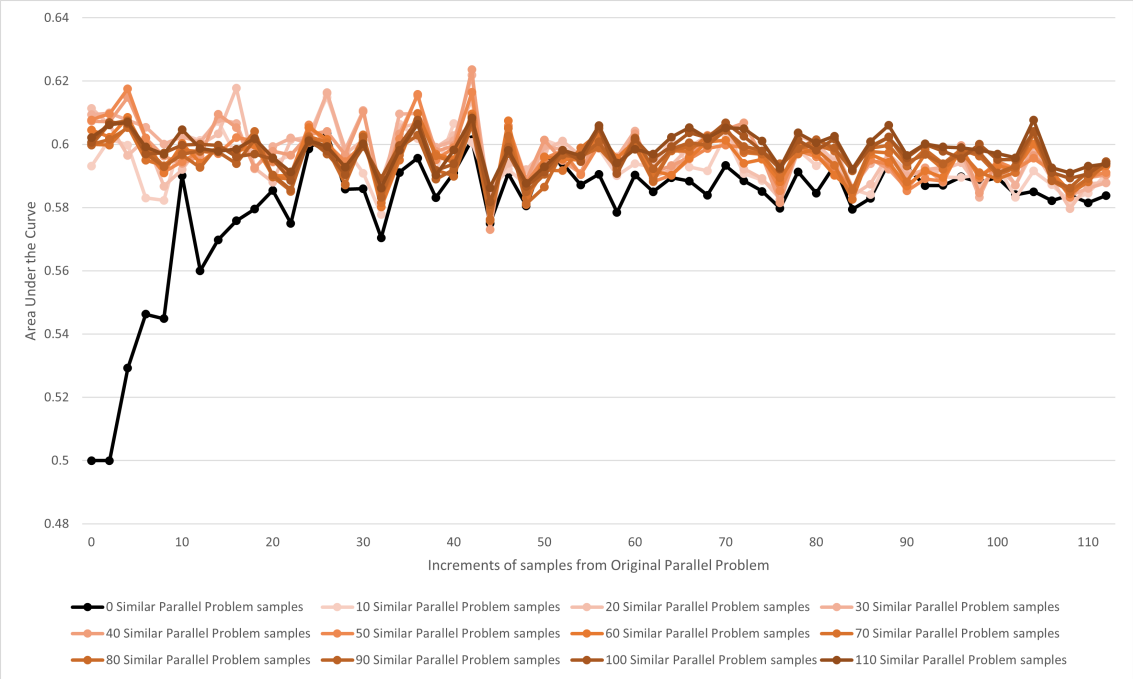

Figure 4.8: Average AUC while varying both the original parallel problem sample size and the similar parallel problem sample size.

addition, models are shown in increments of 10 samples from the similar parallel problem rather than in increments of 5.

As observed in Figure 4.8, all models with similar parallel problem samples generally had significantly better performance in terms of AUC across each increment of samples from the original parallel problem. The model trained with 10 similar parallel problem samples outperformed the respective baseline model by about 0.013 in terms of average AUC per interval. Each model trained with samples from the similar parallel problem outperforms the respective baseline model by an average of 0.016 in terms of average AUC per interval. Unlike in the previous pairs of problems, each model trained with samples from the similar parallel problem does not necessarily outperform the model with the next lowest increment of training samples from the similar parallel problem in terms of average AUC per interval. For example, the model trained with 30 similar parallel problem samples outperforms

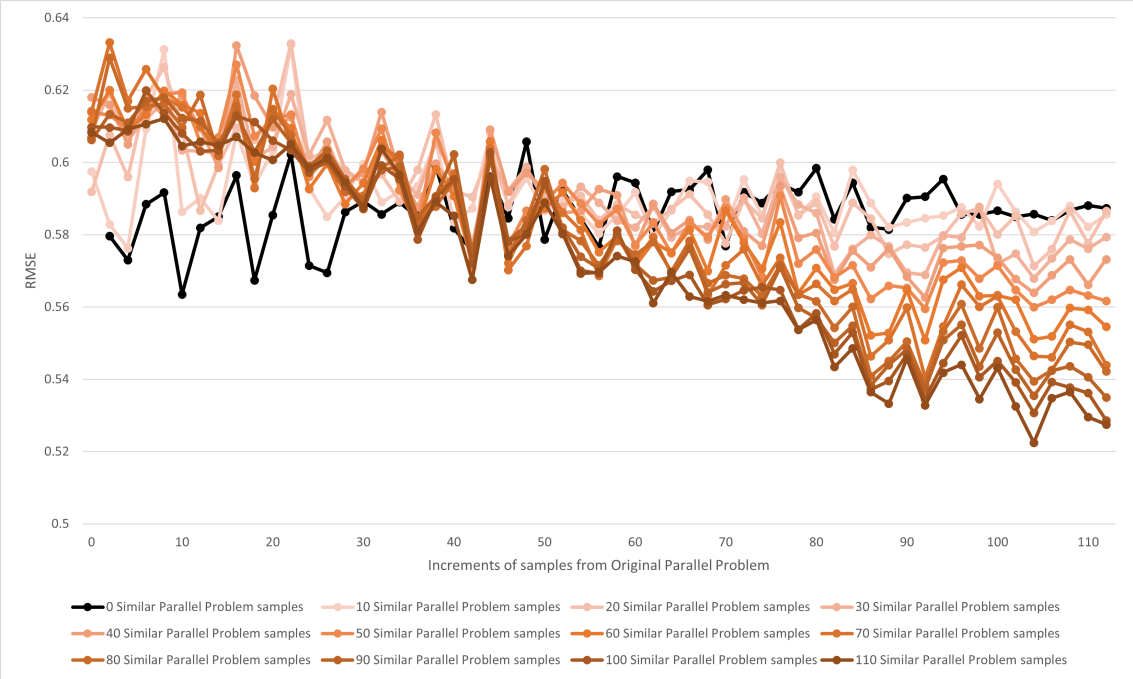

Figure 4.9: Average RMSE while varying both the original parallel problem sample size and the similar parallel problem sample size.

the model trained with 40 similar parallel problem samples by 0.001 in terms of average AUC per interval. The worst average AUC for a model trained with samples from the similar parallel model is 0.556 and it occurs when trained with 10 similar parallel problem samples and 32 samples from the original parallel problem. The best average AUC for models using samples from the similar parallel problem is 0.783 when the model is trained with 40 similar parallel problem samples and 42 samples from the original problem. This problem is unlike the other pairs of problems because none of the models generally converge beyond any number of samples from the original parallel problem despite having the largest number of available samples tested.

For RMSE, the respective baseline model outperforms models using similar parallel problem samples in the earlier intervals up to where the models use 40 samples from the original problem. However after that point, there is a clear improvement

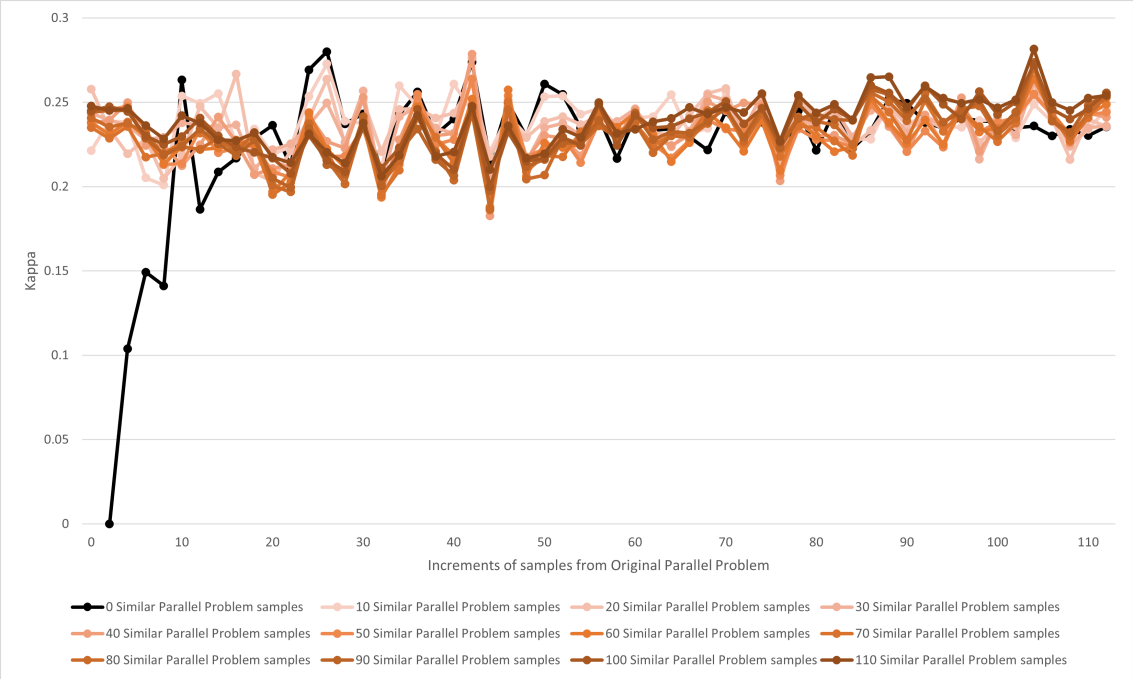

Figure 4.10: Average Kappa while varying both the original parallel problem sample size and the similar parallel problem sample size.

in performance for models using similar parallel problem samples over the respective baseline model. Overall, the baseline model oscillates between 0.56 and 0.61. While nearly all the models using similar parallel samples start above the worst performance of the baseline model as they train with increasing quantities of samples from the original parallel problem, the RMSE on average lessens which amounts to significant improvement over the baseline model. The overall lowest RMSE of 0.52 can be seen from the model using 110 similar parallel problem samples with 104 original parallel problem samples.

Across the different models, Kappa has a similar behavior as with previous pairs of problems, with the model trained with 110 similar parallel problem samples overall performing the best on average of the methods. The respective baseline model performs the worst and the models generally improve as they are trained with increasing numbers of similar parallel problem samples. While this pattern is consis-

tent, the difference between the methods lessens by the larger intervals especially when compared to AUC.

4.3 Discussion

More closely examining the results of the further analyses, we recognize consistent trends across the three pairs of similar problems that are discussed in this section.

Due to the variety of problem pairs, we gain more insight into the current implementation of the automated scoring models with the respective baseline models. For the logarithmic equation solving and exponential decay identification and justification problems in 4.2.1 and 4.2.2 respectively, we see that baseline models converges within 12 samples of their respective original problem. Alternatively for the problem identifying and justifying parallelism in 4.2.3, we see that the baseline model's behavior is more erratic and does not appear to converge until around 90 samples of its respective original problem, if at all. Most of the performance improvements are observed when using AUC. However, the overall decrease in RMSE and increase in Kappa are important achievements that provide a larger picture of the benefits in using any number of samples from similar problems.

Although the performance of all the models generally improves as more samples from their respective similar problem, the trends seen in the improved performance are also often seen in their respective baseline model's performance. In that sense, all the models, regardless of samples from their similar problem, have the same peaks and valleys in regards to the performance at particular original sample size intervals. This is likely a consequence of which samples from the original problem are used to train the model and how diverse all the training samples used are from one another.

As expected, we usually see improvements across all models when trained with more samples from either the respective original problem or the respective similar problem. However, in 4.8, we can also clearly see that quantity is does not exclusively improve performance. As noted previously, the performance varies significantly with the increase of samples from the original parallel problem. Unlike other problems' performances, the AUC does not gradually increase with the increase of respective original samples. Instead, the performance hits a peak at 26 samples and afterwards, continues to hit relative peaks and relative valleys. This suggests that after 26 samples, the increase of original parallel problem samples stops benefiting the performance and instead adds confusion in terms of appropriately scoring student responses.

In addition, there can be a maximum to the benefits provided by supplementing with similar problem samples. Specifically when all the models are trained using solely samples from the similar problem (at interval 0 on the x-axis in 4.8), we can see that the model trained with 110 similar problem samples is outperformed by multiple models trained with fewer similar problem samples. This suggests that the model trained with 110 similar problem samples has worse results because with the increase of samples came additional samples whose scores are inconsistent. Despite not being ideal for training purposes, it should be expected that different teachers would have distinct criteria for students to earn a particular grade. For example, Teacher A could be more lenient and score students mainly full marks (a score of 4) while Teacher B could be searching for clear and concise explanations of concepts. While to our benefit, it is surprising though that the other pairs of similar problems do not seem to come across these scoring inconsistencies in any of the metrics tested. It is possible that this situation is more extreme in 4.2.3 because the problem is so focused on a diagram rather than a focused on a formula or equation like in 4.2.1 and

4.2.2. By having to justify their explanations about interpreting an image, there is likely more variance in the students' responses as well as the teachers' scores. These findings address our fourth research question, but as seen in the variance of performance benefits across problems, it remains inconclusive in determining the strict limits of performance improvement when using increasing quantities of auxiliary data.

Chapter 5

Limitations and Future Work

In Chapter 3, the largest limitation is that the research focuses on predicting the scores of only one specific problem. While we argue that the analyses conducted there were sufficient to address our research questions, there is a larger uncertainty that remains in regard to how representative these results are to the larger set of problems. As a result, in Chapter 4, we test across a variety of problems to ensure that the results generalize well to other possible problems. However, future work should explore even more varied problems in terms of both problem content and the number of available samples to train with.

Another overarching limitation throughout this research that is likely observed in 4.2.3 is that the training data consists of samples of student open-ended responses and their associated teacher provided score. As is the nature of open-ended responses, the quality of the student's response and the teacher provided score can be subjective. Consequently, it opens the possibility of training models with inconsistent scoring standards due to the variance in teachers' scoring requirements within the same problem or its similar problem.

When deciding what constitutes a similar problem, future work could explore

other methods that consider a wide range of comparison characteristics. The choice of problems in this work removed several challenges to identifying similar problems (as the structures of the chosen problems were so similar), but other descriptives including the problem text, knowledge component, grade level, average difficulty, or other such factors may be utilized in comparing problems. Defining such attributes would also provide opportunities to build models to better understand how matching characteristics correlate with model performance gains. By understanding how to better identify similar problems, scoring models that incorporate auxiliary data could better avoid selecting unhelpful or even detrimental samples (e.g. avoid the lower bounds of model performance).

Conversely, the methods explored here may provide insights into the similarity or other relationships between problems and skills. Prior work has focused on developing methods to measure the similarity of problems and skills for the purpose of identifying prerequisite hierarchies among content [ASH+14]. For example, it could be useful to pair problems (Problem A, Problem B) and gauge their similarity or their required knowledge overlap by how well the responses of Problem A could train or supplement the training of a scoring model intended to score the responses of Problem B. Even without the ability to better characterize *how* problems are similar, the magnitude of performance gain by observing model transfer could provide a new measure to gauge these relationships.

Future work could also explore training with more than one similar problem to supplement the original problem's data to see how much the performance can improve or further test if there are limits to the benefits of using other problem's data. Alternatively, future work could explore training using only similar problems' data as a method of transfer learning rather than using the similar problem's data to supplement other original problems' data for training. This would be especially

helpful for open-ended problems that don't yet have any scored responses.

While scoring models are becoming more prevalent in education research and learning systems, teachers often need supports in providing more meaningful forms of feedback beyond that of a numeric score. ASSISTments is already able to recommend feedback for trained problem models, but it requires a lot of data in order to do so (more than for the automated scoring task alone). The use of auxiliary data as explored in this work may prove useful in other such contexts.

Chapter 6

Conclusion

In this work, we explore one possible solution to the cold-start problem in automating the assessment of student open-ended work. We have shown that our SBERT-Canberra method using similar auxiliary problem data consistently and significantly outperformed the model using data solely from the original problem; this trend also held across multiple metrics, with the largest differences observed in AUC. When there are very few training samples from the original problem, even the modified SBERT-Canberra method using random problems' data to supplement helped improve the performance in some cases, and additional research could be conducted to aid in selecting better training samples.

Throughout the exploration of both Chapter 3 and Chapter 4, there is a noticeable benefit to supplementing the training samples with data from similar problems even with as few as 5 samples. By supplementing the original training samples with multiple similar problems, we hypothesize that it will lead to even larger performance improvements to automatic scoring regardless of the number of original training samples. This would be particularly the case if our hypothesis is correct where some of this benefit is derived from regularizing factors.

Future work should use transfer learning to use the SBERT-Canberra model of a similar problem as a starting point to score a new problems' open-ended response. As more data from across problems are collected, we found that there may still be benefits to using auxiliary data even beyond addressing the cold start problem.

Chapter 7

Further Acknowledgments

We thank multiple NSF grants (e.g., 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, & DRL-1031398), as well as the US Department of Education for three different funding lines; the Institute for Education Sciences (e.g., IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, & R305C100024), the Graduate Assistance in Areas of National Need program (e.g., P200A180088 & P200A150306), and the EIR. We also thank the Office of Naval Research (N00014-18-1-2768) and finally Schmidt Futures as well as a second anonymous philanthropy.

Bibliography

[ASH+14] Seth Adjei, Douglas Selent, Neil Heffernan, Zach Pardos, Angela Broaddus, and Neal Kingston. Refining learning maps with data fitting techniques: Searching for better fitting learning maps. In *Educational Data Mining 2014*, 2014.

[BB01] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 26–33, 2001.

[BBE+21] Sami Baral, Anthony Botelho, John Erickson, Priyanka Benachamardi, and Neil Heffernan. Improving automated scoring of student open responses in mathematics. In *Proceedings of the Fourteenth International Conference on Educational Data Mining, Paris, France*, 2021.

[Bou98] J Bouwman. Quality of regularization methods. *DEOS Report 98.2*, 1998.

[CKM16] Scott A Crossley, Kristopher Kyle, and Danielle S McNamara. The tool for the automatic analysis of text cohesion (taaco): Automatic assessment of local, global, and text cohesion. *Behavior research methods*, 48(4):1227–1237, 2016.

[CLP21] Aubrey Condor, Max Litster, and Zachary Pardos. Automatic short answer grading with sbert on out-of-sample questions. In *Proceedings of the Fourteenth International Conference on Educational Data Mining, Paris, France*, 2021.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[EBM+20] John A Erickson, Anthony F Botelho, Steven McAteer, Ashvini Varatharaj, and Neil T Heffernan. The automated grading of student open responses in mathematics. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 615–624, 2020.

[HBVTN21] Georgiana Haldeman, Monica Babeş-Vroman, Andrew Tjang, and Thu D Nguyen. Csf: Formative feedback in autograding. *ACM Transactions on Computing Education (TOCE)*, 21(3):1–30, 2021.

[HH14] Neil T Heffernan and Cristina Lindquist Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[HSHA14] Hiba Jasim Hadi, Ammar Hameed Shnain, Sarah Hadishaheed, and Aziz Ahmad. Big data and five v's characteristics. 2014.

[HT01] David J. Hand and Robert J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.

[JRVF09] Giuseppe Jurman, Samantha Riccadonna, Roberto Visintainer, and Cesare Furlanello. Canberra distance on ranked lists. In *Proceedings of advances in ranking NIPS 09 workshop*, pages 22–27. Citeseer, 2009.

[KC+06] Kenneth R Koedinger, Albert Corbett, et al. *Cognitive tutors: Technology bringing learning sciences to the classroom*. na, 2006.

[LVWB15] Andrew S. Lan, Divyanshu Vats, Andrew E. Waters, and Richard G. Baraniuk. Mathematical language processing: Automatic grading and feedback for open response mathematical questions, 2015.

[RBBBH22] Raysa Rivera-Bergollo, Sami Baral, Anthony Botelho, and Neil Heffernan. Leveraging auxiliary data from similar problems to improve automatic open response scoring. In *Educational Data Mining 2022*, 2022.

[RG19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[RJO19] Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. Language models and automated essay scoring, 2019.

[RM13] Rod D Roscoe and Danielle S McNamara. Writing pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105(4):1010, 2013.

[TS10] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[UU20] Masaki Uto and Yuto Uchida. Automated short-answer grading using deep neural networks and item response theory. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 334–339, Cham, 2020. Springer International Publishing.

[WLW+19] Zichao Wang, Andrew S. Lan, Andrew E. Waters, Phillip J. Grimaldi, and Richard Baraniuk. A meta-learning augmented bidirectional transformer model for automatic short answer grading. In *EDM*, 2019.

[Yin20] Wenpeng Yin. Meta-learning for few-shot natural language processing: A survey, 2020.

[YZY17] Xi Yang, Lishan Zhang, and Shengquan Yu. Can short answers to open response questions be auto-graded without a grading rubric? In *International Conference on Artificial Intelligence in Education*, pages 594–597. Springer, 2017.

[ZHY+22] Lishan Zhang, Yuwei Huang, Xi Yang, Shengquan Yu, and Fuzhen Zhuang. An automatic short-answer grading model for semi-open-ended questions. *Interactive learning environments*, 30(1):177–190, 2022.