Learning to Compare Hints: Combining Insights from Student Logs and Large Language Models

Ted Zhang[1] TEDZ@ANDREW.CMU.EDU
Harshith Arun Kumar[1] HARUNKUM@ANDREW.CMU.EDU
Robin Schmucker[1] RSCHMUCK@ANDREW.CMU.EDU
Amos Azaria[2] AMOS.AZARIA@ARIEL.AC.IL
Tom Mitchell[1] TOMMMITCHELL@GMAIL.COM

[1] *School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA*
[2] *School of Computer Science, Ariel University, Ariel, Israel*

Abstract

We explore the general problem of learning to predict which teaching actions will result in the best learning outcomes for students in online courses. More specifically, we consider the problem of predicting which hint will most help a student who answers a practice question incorrectly, and who is about to make a second attempt to answer that question. In previous work (Schmucker et al., 2023) we showed that log data from thousands of previous students could be used to learn empirically which of several pre-defined hints produces the best learning outcome. However, while that study utilized data from thousands of students submitting millions of responses, it did not consider the actual text of the question, the hint, or the answer. In this paper, we ask the follow-on question "Can we train a machine learned model to examine the text of the question, the answer, and the text of hints, to predict which hint will lead to better learning outcomes?" Our experimental results show that the answer is yes. This is important because the trained model can now be applied to new questions and hints covering related subject matter, to estimate which of the new hints will be most useful, even before testing it on students. Finally, we show that the pairs of hints for which the model makes most accurate predictions are the hint pairs where choosing the right hint has the biggest payoff (i.e., hint pairs for which the difference in learning outcomes is greatest).

Keywords: intelligent tutoring systems, cold start problem, data-driven design

1. Introduction

Providing students with support while they solve practice problems is known to have positive effects on their learning outcomes (Hattie and Timperley, 2007), but determining precisely what support will be the most helpful is a nontrivial task for designers of intelligent tutoring systems (ITSs) (Nathan et al., 2001). In this paper we consider a case study of this problem, in which the task is to choose which of several pre-defined hints will best help a student when they answers a practice problem incorrectly, but before they make a second attempt to answer it. The work we report here builds on our previous work, in collaboration with the online learning platform ck12.org, in which we used log data from over a hundred thousand students creating millions of question answers Schmucker et al. (2023). We used this data to learn, for each practice question, which of its pre-defined hints was most successful. Success of the hint in this previous work was defined in terms of the correctness rate of students in their second attempt to answer the question, and final score over the entire current lesson.

While that work succeeded in discovering useful hints, and has now been deployed to hundreds of thousands of students using the ck12.org learning platform, our approach there did not consider the actual text of the question, its answer, or the hints. In the current paper we present a new approach that does consider this text content. In particular, we train a classifier that takes as input the text embeddings of the question, of two hints, of the answer, and additional features described below, to predict which of the two hints will be most successful. Here we define success in terms of the rate of correct responses obtained from the second attempt of students in answering this question. We use student log data to obtain these second response correctness rates as supervision for training, but once the model is trained it can take new questions and hints as inputs and needs no log data to make its predictions of which hint will work best.

We frame our problem as a classification problem (classifying which action is better) rather than a regression problem (predicting the exact second attempt correctness rate for each hint) for the following reasons: 1.) The goal is to decide which hint is better, rather than predicting the actual reattempt correctness rate. 2.) A voting consensus mechanism can be adapted to pick out the best hint within the set of 4-6 hints typically available for each question, and 3.) Pairwise comparison is an easier and more interpretable task. The main contributions of this paper include:

- We develop a classifier that predicts differences in the effectiveness of hints based on their text content, along with the text of the question and answer, showing that this classifier can be successfully trained on questions and hints from two different courses: Physical Science and Biology.
- We perform an analysis of which features are most useful for this classifier, considering additional input features such as the prediction of GPT-3.5 / GPT-4 regarding which hint will be most effective.
- We study how classifier accuracy is related to the magnitude of the difference in hint effectiveness, finding that the classifier is most accurate in the most important cases (i.e., when the actual difference in learning outcomes for the two hints is greatest).

The results of our study suggest that trained classifiers such as ours may play an important role in the design and continuous improvement of future online education platforms.

2. Related Work

Early research evaluated design principles for intelligent tutor systems (ITS) on the system level, where all decisions points inside the ITS implemented the same design choice (e.g., step-wise hints during problem solving) (Kulik and Fletcher, 2016). Since then, the focus shifted towards data-driven approaches that leverage student data to identify effective teaching actions for individual decision points often via bandit algorithms and randomized experiments (e.g., (Selent et al., 2016; Ostrow et al., 2017; Williams et al., 2018; Fancsali et al., 2022; Schmucker et al., 2023)). Our work also focuses on building a system to identify the best teaching action (e.g., hint). With the advent of online ITSs, large scale student log data is available, and with this data it has become possible to apply machine learning methods in practice to improve online education.

The action of providing assistance (e.g., a hint) when a student answers a question incorrectly, and before they reattempt the question, inherently provides a training signal about the effectiveness of the assistance, which can be used to improve student learning. As such, a survey of using reinforcement learning is described in (Doroudi et al., 2019).

One key difference and benefit of our approach is that prior work relies on student log data to determine which teaching actions are most effective necessitating the collection of new log data for each new teaching action. In this work, we use data from prior large-scale action evaluations as supervision to train a model capable of assessing new teaching hints for which no student log data is available yet based on textual content. The evaluations provided this model can reduce the need for online content evaluations and mitigates the new content cold-start problem (Schmucker and Mitchell, 2022).

3. Dataset

We used the dataset from (Schmucker et al., 2023) to train our models. This data was collected during a randomized experiment inside the Flexbook 2.0 system hosted by the CK-12 foundation. The dataset is comprised of the questions, hints, and student-provided answers for each. In addition, data was aggregated across students to calculate the rate of correct reattempts by students who got a question wrong and were provided a hint before reattempting the question. From the large scale randomized study, each hint has a corresponding reattempt score (correctness rate) which summarizes how effective that hint is in helping the student answer the question correctly on their second attempt.

We focus on two different courses in the CK-12 dataset, Biology, and Physical Science. From these courses we obtain the questions, answers, hints, and student log data. The dataset is based on data from over 300,000 students. For each of the two courses, this yields over 7000 examples, where each example consists of a question, two available hints for that question, and the second attempt score for each hint.

4. Approach

Text Embeddings We process the data from each lesson using Large Language Models (LLMs) to retrieve vector embeddings for each question, hint, and answer text. We choose ADA (OpenAI, 2020) for our embeddings. The embeddings represent the text content, and are used as an input to our model.

Embedding Feature Engineering In addition to the raw text embeddings of questions, answers, and hints, we also introduce a vector which is the z-scored difference between the two input assistance embeddings.

Embedding Classification Model. As shown in Figure 1, our baseline model is a fully connected two layer neural network using a sigmoid output, trained with a cross entropy

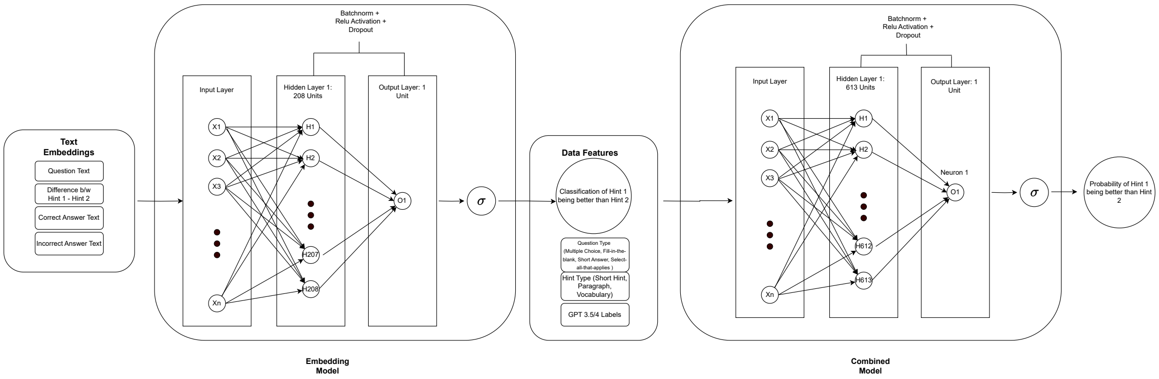
Text Embeddings
Question Text
Difference b/w Hint 1 - Hint 2
Correct Answer Text
Incorrect Answer Text
Batchnorm + Relu Activation + Dropout
Input Layer
Hidden Layer 1: 208 Units
Output Layer: 1 Unit
X1
X2
X3
Xn
H1
H2
H207
H208
O1
σ
Embedding Model
Data Features
Classification of Hint 1 being better than Hint 2
Question Type (Multiple Choice, Fill-in-the-blank, Short Answer, Select-all-that-applies)
Hint Type (Short Hint, Paragraph, Vocabulary)
GPT 3.5/4 Labels
Batchnorm + Relu Activation + Dropout
Input Layer
Hidden Layer 1: 613 Units
Output Layer: 1 Unit
X1
X2
X3
Xn
H1
H2
H612
H613
Neuron 1
O1
σ
Probability of Hint 1 being better than Hint 2
Combined Model

Figure 1: Model Architecture. The neural network to the left (the ”Embedding Model”) first predicts the probability that hint 1 is better than hint 2, based on text embeddings of the question, the difference between the hint embeddings, and optionally the embeddings of the correct answer and incorrect answers. The prediction of this baseline model is then input to the rightmost neural network (the ”combination model”), which combines this prediction with additional inputs to make the final prediction.

loss function, so that the output can be interpreted as the probability that input Hint 1 is more helpful than input Hint 2. The baseline model uses elastic net to regularize the cross entropy loss. The hidden layer contains 208 neurons which is then followed by batch normalization, ReLU activation and a dropout layer before the output layer. We trained two of these embedding classification models using ADA text embeddings described in our previous section. The first uses the question text and difference between embeddings of Hint 1 and Hint 2 text. The second uses these same input features plus the embeddings of the correct answer text, and incorrect answer text (e.g., for multiple choice questions, the embedding of all incorrect answers).

Combination Model. To train our downstream combination model (the rightmost network in Figure 1), we first picked the most accurate embedding classification model. Inputs to our combination model include the binary classification prediction of our embedding model, the question type (a one-hot encoding of alternatives ”fill-in-the-blank”, ”multiple-choice”, ”short-answers”, and ”select-all-that-apply”), types of the two hints being compared and classification labels from GPT-3.5 Instruct Turbo or GPT-4 when they are asked which hint is likely to work best (Figure 4). There are three types of hints that we consider: *short hint* (1-2 sentences), *vocabulary* (relevant keyword definitions), and *paragraph* (short excerpt from lesson text). Our combination model as shown in Figure 1 has the same architecture as the embedding classification model. The only difference being the input features and the hidden layer's size which increased to 613 neurons. When querying GPT-3.5 Turbo and GPT-4 to ask which hint is more helpful, we used the prompt shown in Figure 4.

You are a high school {subject_name} teacher. A student was asked a practice question, but answered incorrectly. Which of the following two hints (hint A or hint B) is more likely to help the student answer correctly on their 2nd attempt? The question, its solution and the two hints provided below: Question: {question_text}
Solution: {answer_text}
Hint A: {hint_1_text}
Hint B: {hint_2_text}
Write your decision in the following format: Hint A OR Hint B

Figure 2: Prompt to GPT-3.5 Turbo and GPT-4 when asking it to compare two hints. We collected both the LLM response, and the token probability for "A" and "B."

5. Results

5.1. Evaluation Metrics

For each example, we evaluate the correctness of the model's prediction based on the actual observed students' reattempt success rate for each question and hint. We evaluate model performance using both *Accuracy* and *Area under ROC curve (AUC)*. We report average standard deviation performance for each metric using a 5-fold cross-validation on the question level. In separating train from test sets, we assure that train and test sets contain only disjoint sets of questions, to avoid the problem of potential overfitting when overlapping content is in train and test set.

5.2. Feature Evaluation

Our embedding classification model, presented in the top row of Table 1, used only question and assistance embeddings as its inputs. We hypothesized that integrating additional textual data, student log data, and LLM predictions could enhance our classifier's accuracy. We therefore trained alternative models, each of which included a single additional feature from this feature pool. As shown in Table 1, adding these features as inputs yielded only slight improvements. This indicates that the essential factors for our neural network's assistance decisions are primarily the question and assistance text.

5.3. Comparison of Trained Model Against Alternative Approaches

To further test our model, we compared it against two other approaches: (1) asking GPT-3.5 or GPT-4 to select the best hint for the question, and (2) "paragraph when possible", in which we choose the paragraph whenever it is one of the available hints, and choose randomly in the case the neither or both hints are paragraphs. Note that among the alternative hints used by CK12, a *paragraph hint* consists of one paragraph of text taken from the reading material for this lesson. Averaged across all questions, these paragraph hints have the greatest success rate, so this "paragraph if possible " hint baseline is considerably more accurate than the baseline of simply selecting hints at random.

Table 2 presents the results of this comparison. Here "Random" refers to the approach of randomly choosing one of the two hints, "Best Embedding Classification" refers to the best

Table 1: Feature evaluation. Each entry reports average ACC and AUC for a MLP classifier trained using question, assistance action, and answer text embeddings augmented with a single additional feature in an ensemble model. Mean and standard error are computed using five-fold test data.

	Biology		Physical Science	
Method	**Accuracy**	**AUC**	**Accuracy**	**AUC**
Embedding Classification (quest. + assist.)	0.641 ± 0.007	0.698 ± 0.013	0.628 ± 0.007	0.666 ± 0.004
Embedding Classification (quest. + assist. + answ.)	0.643 ± 0.006	0.696 ± 0.010	0.617 ± 0.010	0.654 ± 0.008
Best Embedding Model w/ assistance type	0.647 ± 0.010	**0.708** ± 0.011	0.636 ± 0.016	0.678 ± 0.012
Best Embedding Model w/ question type	0.643 ± 0.008	0.658 ± 0.007	0.626 ± 0.008	0.659 ± 0.011
Best Embedding Model w/ GPT-3.5 predictions	0.647 ± 0.010	0.678 ± 0.016	0.636 ± 0.004	0.655 ± 0.008
Best Embedding Model w/ GPT-4 predictions	**0.664** ± 0.011	0.700 ± 0.015	**0.659** ± 0.004	**0.681** ± 0.009

Table 2: Classifier evaluation. We report accuracy (ACC) and AUC for different classifiers trained to identify the more effective assistance action in pairwise comparisons. Mean and standard error are computed using five-fold test data. AUC scores were not available for GPT-4 because its prediction probabilities were unavailable.

	Biology		Physical Science	
Method	**Accuracy**	**AUC**	**Accuracy**	**AUC**
Random	0.500	0.500	0.500	0.500
Paragraph if possible	0.641	0.701	0.602	0.657
GPT-3.5 Prediction	0.616	0.617	0.622	0.627
GPT-4 Prediction	0.666	-	0.659	-
Best Embedding Classification	0.643 ± 0.006	0.696 ± 0.010	0.628 ± 0.007	0.666 ± 0.004
Combination	**0.679** ± 0.014	**0.740** ± 0.012	**0.664** ± 0.006	**0.718** ± 0.007

trained embedding classification model, and ”Combined” refers to our combined model augmented by all of the additional features shown in Table 1 and the output of the best embedding classification model for their respective subject. Notice here that GPT-3.5 and GPT-4 both perform considerably better than the Random baseline in choosing hints, though our combined trained model perform the best by a slight margin.

Finally, we study what are the hint pairs for which our model performs well, versus those for which it does not. Figure 3 shows for each course subject the prediction accuracy of our model plotted against the difference in the effectiveness of the two hints (labeled ”effect delta”). We find that for hint pairs with a large effect delta our model exhibits quite high prediction accuracy. As the effect size decreases so does our prediction accuracy. This result shows that the model is best at choosing the most effective hint precisely in the cases where the choice is most important: when the difference in hint effectiveness is greatest.

6. Conclusion and Future Work

This paper presents the first demonstration, to our knowledge, that a classifier can be trained to choose the best hint for a question, based solely on the text of the question and hints.

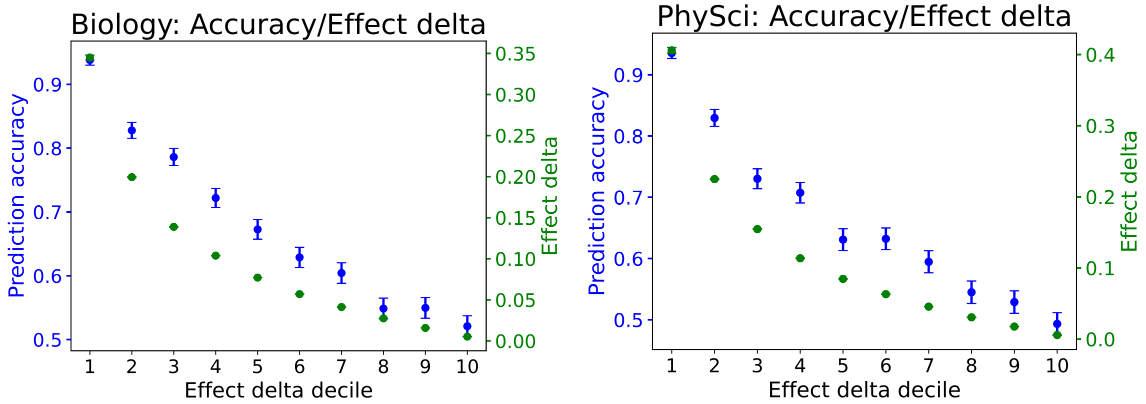

Figure 3: Pairwise classification accuracy for Biology [Left] and Physical Science [Right]. We sort the hints comparison problems based on the difference in effects sizes between the two hints, and group them into deciles. For each group we compute average classifier accuracy (left axes) and effect difference (right axes). We observe highest classification accuracy for hint pairs with large effect size differences.

The classifier is trained to choose the hint with the best learning outcome as determined from high-volume student log data. However, once trained, the model can be applied to new hints and questions on related course topics, even before any student log data is available. This opens up the possibility of taking in new hints suggested by educators, or generated by LLMs, and automatically filtering them with our trained classifier. In this study, we also examined a range of possible input features for the classifier, finding that the text of the question and hints provide most of the information on which successful predictions are based. Most importantly, we find that our model is most accurate in the cases where it's predictions are most critical, that is, when the difference in learning outcomes for the two hints is greatest. In the future, we would like to explore training a more complex model on all subjects instead of separating them by the subject. In doing so, we hope to discover how to best utilize student log information alongside textual information to build a model that leverages the best of both modalities.

Acknowledgments

We thank Microsoft for support in the form of Azure computing and access to the OpenAI API through a grant from their Accelerate Foundation Model Academic Research Program. We thank the CK-12 Foundation (ck12.org) for providing the student log data used here, and for discussions leading up to this problem formulation. This research was supported in part by the AFOSR under award FA95501710218.

References

Shayan Doroudi, Vincent Aleven, and Emma Brunskill. Where's the reward? *Int. Journal of AIED*, 29(4):568–620, 2019.

Stephen Fancsali, April Murphy, and Steven Ritter. “closing the loop” in educational data science with an open source architecture for large-scale field trials. In *Proc. of the 15th Int. Conf. on EDM*, pages 834–838, Durham, UK, July 2022. EDM.

John Hattie and Helen Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

J.A. Kulik and J.D. Fletcher. Effectiveness of intelligent tutoring systems: a meta-analytic review. *Rev. Educ. Res.*, 86(1):42–78, 2016.

Mitchell J Nathan, Kenneth R Koedinger, Martha W Alibali, et al. Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In *Proceedings of the third international conference on cognitive science*, volume 644648, pages 644–648, 2001.

OpenAI. Gpt-3 models. https://platform.openai.com/docs/models/gpt-3, 2020. Accessed: 2023-11-18.

Korinn Ostrow, Neil Heffernan, and Joseph Jay Williams. Tomorrow’s edtech today: Establishing a learning platform as a collaborative research tool for sound science. *Teachers College Record*, 119(3):1–36, 2017.

Robin Schmucker and Tom M Mitchell. Transferable student performance modeling for intelligent tutoring systems. In *Proceedings of the 30th International Conference on Computers in Education*, pages 13–23, Kuala Lumpur, MY, 2022. APSCE.

Robin Schmucker, Nimish Pachapurkar, Shanmuga Bala, Miral Shah, and Tom Mitchell. Learning to give useful hints: Assistance action evaluation and policy improvements. In *European Conference on Technology Enhanced Learning*, pages 383–398. Springer, 2023.

Douglas Selent, Thanaporn Patikorn, and Neil Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proc. of the 3rd ACM Conf. on Learning@ Scale*, pages 181–184, New York, NY, USA, 2016. ACM.

Joseph Jay Williams, Anna N. Rafferty, Dustin Tingley, Andrew Ang, Walter S. Lasecki, and Juho Kim. Enhancing online problems through instructor-centered tools for randomized experiments. In *Proc. of the CHI Conf. on Human Factors in Computing Systems*, CHI ’18, page 1–12, New York, NY, USA, 2018. ACM.