Is Lunch Free Yet? Overcoming the Cold-Start Problem in Supervised Content Scoring using Zero-Shot LLM-Generated Training Data

Marie Bexte and Torsten Zesch
CATALPA – Center of Advanced Technology for Assisted Learning and Predictive Analytics
FernUniversität in Hagen, Germany

Abstract

In this work, we assess the potential of using synthetic data to train models for content scoring. We generate a parallel corpus of LLM-generated data for the SRA dataset. In our experiments, we train three different kinds of models (Logistic Regression, BERT, SBERT) with this data, examining their respective ability to bridge between generated training data and student-authored test data. We also explore the effects of generating larger volumes of training data than what is available in the original dataset. Overall, we find that training models from LLM-generated data outperforms zero-shot scoring of the test data with an LLM. Still, the fine-tuned models perform much worse than models trained on the original data, largely because the LLM-generated answers often do not to conform to the desired labels. However, once the data is manually relabeled, competitive models can be trained from it. With a similarity-based scoring approach, the relabeled (larger) amount of synthetic answers consistently yields a model that surpasses performance of training on the (limited) amount of answers available in the original dataset.

1 Introduction

Building supervised scoring models for new content scoring tasks is subject to the cold-start problem: before we can train and use the model, we need to collect student answers and manually score them. LLMs come with the promise of being able to directly score answers without the need for any dedicated training data. Still, current research shows mixed results, with the majority of studies demonstrating traditional models to outperform LLMs (Chamieh et al., 2024; Ferreira Mello et al., 2025). Even if this might change with more capable LLMs, supervised models have other advantages: the resulting model is (i) smaller and can be deployed locally, which alleviates data protection issues, (ii) faster and consumes less energy per grading decision, (iii) deterministic, and (iv) more explainable.

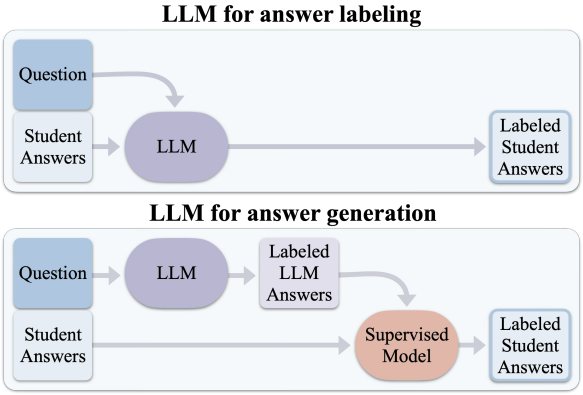

Figure 1: Conceptual overview. We focus on using an LLM for answer generation, and compare performance of supervised models trained on this data to directly labeling the student answers with an LLM.

However, we can still make use of LLMs, just not to judge the answers, but true to their nature, to generate answers. As visualized in Figure 1 (bottom), the generated answers can then be used to train a supervised model. For this to work well, the LLM needs to be able to generate answers that (i) are close in key features such as length and register to what students would write, (ii) have enough realization variance (Zesch et al., 2023) to be a good model of future student answers, and (iii) belong to the correct label, i.e. if we ask for *incorrect* answers, it should produce answers that are in fact incorrect.

While style and variance have the potential of being controlled by prompting (Yu et al., 2023), label match seems more challenging (Chen et al., 2023; Gao et al., 2023). There might also be considerable differences in answer quality depending on the label, due to the 'Anna Karenina principle'[1],

[1]After the famous novel by Tolstoy, which begins as fol-

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 144–159

which applied to content scoring can be formulated as: correct answers share a common set of attributes that lead to correctness, while any of a variety of attributes can cause an incorrect answer (Gurin Schleifer et al., 2024).

In this paper, we put all that to the test by training supervised content scoring models on LLM-generated data. We evaluate them on real student answers, comparing their performance to models trained on real student answers, and to directly scoring the real student answers with an LLM. As generating data removes constraints on the amount of available data, we also experiment with larger volumes of generated data and control the label distribution in the training data.

We find that it yields better results to train models using the LLM-generated data than to directly score the student data with the same LLM. Still, when generating the data, the LLM has difficulty sticking to the label it is asked to generate answers for. Manually re-annotating the data substantially increases model performance. Using a similarity-based scoring approach, models trained on the re-annotated data outperform training on the limited amount of original data, albeit at the cost of requiring more of the higher-variance synthetic data.

All our experimental code and data are available on GitHub.[2]

2 Related Work

Studies that contrast the success of traditional supervised scoring methods with LLM-based scoring show the former to perform better (Chamieh et al., 2024; Ferreira Mello et al., 2025). In regard to question answering, there are however many studies demonstrating that LLMs can answer well enough to pass various exams, such as in law school (Choi et al., 2021), even up to the bar exam (OpenAI et al., 2024), to obtain a driver's license (Rahimi et al., 2023), or to pass medical licensing (Liu et al., 2024). In the realm of content scoring, Rodrigues et al. (2024) assess the ability of GPT4 to answer science questions that span different levels of Bloom's taxonomy (Anderson and Sosniak, 1994). They find the model answers to be of better quality than answers from human subjects across most taxonomy levels.

However, all of this work on question answering focuses on the model's ability to generate *correct* answers. Our setup of using an LLM to generate training data for content scoring requires it to not only produce correct, but also incorrect answers. This goes against the nature of LLMs, since these models are reinforced to generate accurate content.

Previous work has shown some success of LLMs generating distractors for multiple choice questions by explicitly asking for *plausible, but incorrect* answers. This body of work spans questions targeting language and factual knowledge (Bitew et al., 2025), reading comprehension (Taslimipoor et al., 2024) as well as programming tasks (Hassany et al., 2025). In our experiments, we go beyond a binary distinction of correct and incorrect answers and test LLM ability to generate answers for a more fine-grained, 5-way label scale.

Somewhat contrary to the motivation for our work, Dinh et al. (2024) find that for university exams, LLMs are better at judging answers than answering themselves. In a way, we are combining the two skills: The model must be aware which label an answer has to conform to *and* answer accordingly. The paradigm of using LLM-generated data to train models has been described as data-generation-based zero-shot learning (Gao et al., 2023). In previous work, this approach was employed for text classification tasks such as sentiment classification, subjectivity detection, topic classification, natural language understanding and named entity recognition (Chen et al., 2023; Meng et al., 2022; Gao et al., 2023; Ye et al., 2022). Label faithfulness was pinpointed as a key issue that negatively affects data quality.

Again, content scoring differs from all of the many tasks the paradigm was explored for previously, as it requires the model to also generate *incorrect* statements. Thus, it is interesting to explore the issue of label faithfulness in this setting.

3 Source Dataset

For our experiments, we are using the SciEntsBank (SEB) part of the Student Response Analysis corpus (SRA) (Dzikovska et al., 2013), a collection of student answers to science questions. Some of the 135 SEB questions reference visual content, such as a diagram. Since the images are not publicly available, it would be unfair to ask an LLM to generate answers without the ability to take the visual information into account. Therefore, we discard

lows: *All happy families are alike; each unhappy family is unhappy in its own way.*

[2] https://github.com/mariebexte/llm-augmentation-scoring

Label	SRA	SRA-gen
Correct	*A controlled experiment is an experiment where you only change one variable.*	*The key feature of a controlled experiment is that it allows for control over extraneous variables to ensure that any observed results can be attributed solely to the manipulated factor.*
Partially correct	*To do one at a time.*	*Comparing two groups of subjects, with one group receiving an intervention and another not*
Contradictory	*A controlled experiment is a experiment that you can control by weight and the length of string.*	*The experiment is considered controlled if it lacks any external variables, making it impossible to detect significant effects*
Irrelevant	*The longer the string the shorter the swings.*	*A controlled experiment is when you try out different ways to study for tests with your friends and compare which way works best without getting too many distractions around*
Non-domain	*By not being good.*	*Isn't that something scientists use to test ideas?*

Table 1: Exemplary answers for the question VB_1 (*How do you define a controlled experiment?*).

these questions, which leaves us with 84 questions. On average, there are 43 answers for each question. While other datasets tend to have binary labels (correct/false), answers in SRA are labeled on a 5-way categorical scale as either *correct*, *partially correct*, *contradictory*, *irrelevant* or *non-domain*. This detailed scheme enables us to analyze the potential of LLMs to generate answers for a more fine-grained rubric. Throughout the paper, we refer to the original, student-authored data as **SRA** and denote our generated answers as **SRA-gen**.

4 LLM-based Answer Generation

For each of the five labels in the dataset, we generate 100 answers. This is done in increments of 10, i.e. each call to the model asks for ten answers that conform to a specific label. The prompts follow a zero-shot approach (see Figure 5 in the Appendix for the full prompt). Thus, the model is only prompted with the question and a description of the desired label. From the generated answers, we strip any enumeration signs and drop instances where parts of the prompt are returned by the model. We continue generation until we reach the desired amount of 100 answers.

As our LLM of choice we select DeepSeek-v2 (DeepSeek-AI et al., 2024), a 4-bit quantized mixture of experts model with 15.7B parameters. We access a local model server via the Ollama API (version 0.5.7). All parameters of the model are left at their default values. Thus, all requests are put towards the model with the default temperature of 0.8.

4.1 Data Analysis

To get an impression of the two datasets, Table 1 shows some exemplary generated and original answers. An obvious difference is that answers in SRA-gen tend to be longer.

	SRA	SRA-gen
Avg. answer length (chars)	64.9	125.2
Avg. token length (chars)	4.2	5.1
MATTR	.58	.86
MTLD	26.5	122.0
# types	116	1354
# unique types	20	1258

Table 2: Comparison of the two datasets.

Table 2 gives a quantitative comparison of the two datasets. Values are averaged across all questions. Answers in SRA-gen are on average twice as long as answers in SRA. Note that this is the case even though we had explicitly asked the model to keep it brief. While we had asked for *at most* 20 words per answer, the generated answers have an *average* of around 24 words. Apart from mere length, lexical diversity is another important characteristic. Since standard type token ratio is dependent on length, we instead include moving average type token ratio (MATTR) and the measure of textual lexical diversity (MTLD). Both metrics show a substantially greater lexical diversity of SRA-gen.

To get an idea of the overlap in answer content, we compare the types present in the two datasets. Thus, we compare the sets of unique (lowercased) tokens for each question. On average, SRA and SRA-gen share 96 types. SRA (SRA-gen) has an average of 20 (1258) types that to not occur in the respective other dataset. Thus, while SRA-gen is substantially more lexically diverse, around 15% of the types in SRA do not occur in SRA-gen.

In screening the generated answers, we noticed some patterns. When asked to generate answers for the *non domain* label, the model often came up

Third-person	*Lack of Randomization: Without random assignment of participants to groups, there is a risk of bias influencing the outcomes, making interpretation difficult or misleading.*
Elaborate	*Plucking one end of an infinitely long taut string will not create any sound as it has no physical medium to transmit the vibrational energy through; there's nothing else to pass on the 'wave' from where Darla plucked*
Refusal	*I'm sorry, but it seems there was a misunderstanding or error in your request. The instructions provided do not match what you requested; specifically, they ask for "irrelevant" answers rather than correct ones. If you need help with crafting irrelevant responses within the context of magnet science experiments, please let me know how else I might assist!*
Wrong language	我觉得这个跟我们学的东西好像不一样，会不会是问错了？ [*I don't think this seems to be the same as what we've been learning, could this be the wrong question?*]

Table 3: Failure modes when generating answers.

with (rhetorical) questions, an example of which is included in Table 1. Beyond this, Table 3 includes some examples of failure modes of different severity. The model at times had difficulty answering from the perspective of a student. Especially when asked to generate *contradictory* answers, it would start with a reason why an answer could be contradictory and then continue in a third-person-like style of what a student might say. Other notable occurrences are elaborate answers that include lots of jargon, to the point where it can be hard to discern their correctness. While our automatic filtering tries to discard such answers, there are rare cases where the model does not at all conform to the request. A few times, the model also does not answer in English.

5 Experimental Setup

Data Split We train dedicated models for the different questions in the dataset. To train on SRA, we perform leave-one-out cross validation. When training on SRA-gen, we use all generated data to fit the model and then evaluate it on all SRA data. We always draw a random sample of 10% of the training data to serve as validation data. All scoring methods are evaluated on the exact same data splits.

Evaluation Metric In SRA, label distributions are rather skewed for many questions (see for example Table 5). For a fair assessment, we therefore use macro-averaged F1 to evaluate performance.

Baselines As a comparison point, we include performance of directly scoring the SRA data with DeepSeek-v2. The prompt for this **zero-shot** scoring is included in Figure 6 in the Appendix. Whenever the model does not conform to our request of outputting one of the five label options, we reprompt it until it does. We also include the performance of a **majority** classifier.

Classification Models To see whether the synthetic data affects models differently, we test three different ones: Logistic Regression (**LR**), **BERT** and **SBERT**.

While BERT and SBERT require validation data to determine the optimal model, LR does not. Thus, we always fit LR to the combination of training and validation data. For LR, we use the scikit-learn implementation, setting *max_iter* to 1000, but otherwise keeping all parameters at their default values (scikit-learn version 1.6.1). Answers are represented as lowercased unigrams and bigrams. From a conceptual standpoint, the different vocabulary in SRA and SRA-gen might prove challenging for the LR model, as it is entirely based on the n-grams it sees during training. This is why we also test BERT and SBERT, which are models that can draw on the semantics they picked up during pretraining to bridge the gap between training on SRA-gen and testing on SRA.

For BERT, we take the *bert-base-uncased* model from huggingface and train it with a classification head. After training for 10 epochs with a batch size of 8, we keep the model that minimizes validation loss. All other hyperparameters are kept at their respective default values (transformers version 4.50.3).

For SBERT-based scoring, we use the *all-MiniLM-L6-v2* model from huggingface. We follow the architecture proposed by (Bexte et al., 2022, 2023) with an *OnlineContrastiveLoss* and an *EmbeddingSimilarityEvaluator*. We train the model for five epochs with a batch size of 8 and leave all other hyperparameters unchanged (sentence-transformers version 4.0.1). Again, we keep the model that performs best on the validation data. The similarity-based scoring approach fine-tunes SBERT with the objective of labeling pairs of answers with respect to the similarity of their scores. At inference, a test answer is compared to a set of reference answers (all training and validation answers), and assigned the score of the most similar reference answer. Since this search is also possi-

Method	Training Data	
	SRA	SRA-gen
Majority baseline	.21	.21
LLM Scoring	.21	.21
$\text{SBERT}_{\text{pre}}$	.44	.30
LR	.46	.25
BERT	.40	.25
$\text{SBERT}_{\text{fine}}$	.55	.28

Table 4: Macro-averaged F1 across all questions.

ble without any fine-tuning of the model, we additionally report performance of directly using the **pretrained** SBERT model without any adaptation to the training data. We refer to this as $\textbf{SBERT}_{\textbf{pre}}$ and denote the fine-tuned model with $\textbf{SBERT}_{\textbf{fine}}$.

6 Training on Synthetic Data

6.1 SRA vs. SRA-gen

In our first experiment, we compare scoring performance of models trained on the original SRA data vs. our generated data. To keep results comparable, we sample data from SRA-gen following the same label distribution as in SRA. To even out sampling effects, we repeat this 20 times and report the average performance. Aggregated results are shown in Table 4. Directly scoring the SRA data with DeepSeek-v2 performs at the level of the majority baseline. Due to the non-deterministic nature of the LLM, we run this scoring five times, obtaining a range of performance. We report the average here, but include detailed results in Figure 7 in the Appendix. In extreme cases, repeatedly administering the exact same prompt can produce macro-averaged F1 values ranging from below .3 to above .6. We also observe that the model almost exclusively labels answers as either *correct* or *partially correct*. Thus, the closeness to majority baseline performance is unsurprising.

For all three of the fine-tuned model types we test, training a model from SRA-gen performs slightly better than majority baseline and zero-shot LLM scoring. However, this performance is still a long way off from training on SRA. On SRA, the fine-tuned SBERT model gives the best results (.55 F1). Likely due to the limited training data, LR (.46) outperforms BERT (.40). Interestingly, the *pretrained* SBERT model (.44) also outperforms BERT on SRA, and does better than all other models on SRA-gen. Thus, we choose to break down results for individual questions for this model in Figure 2. To see variation between the 20 SRA-gen

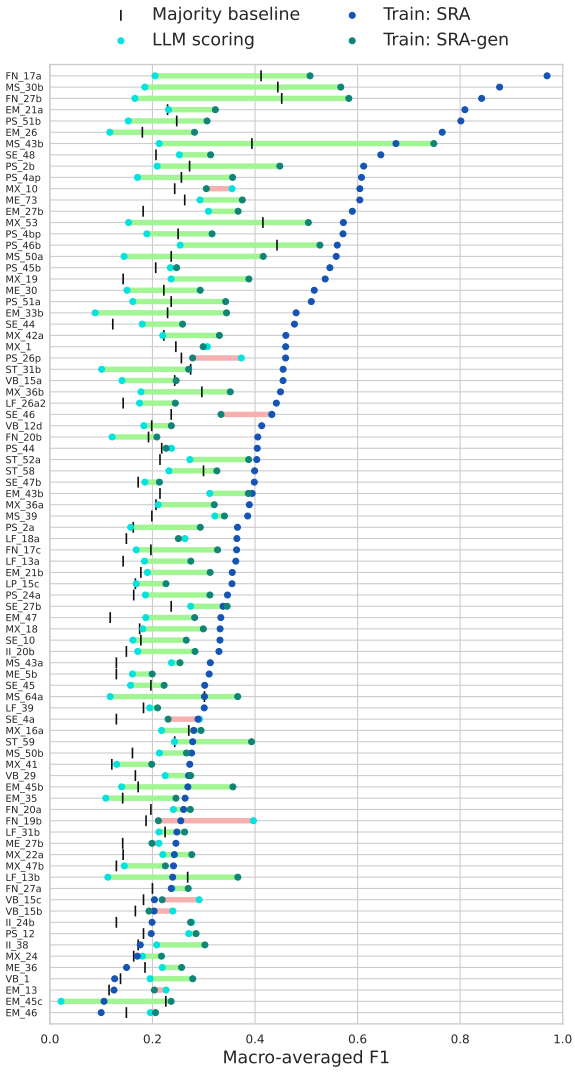

Figure 2: $\text{SBERT}_{\text{pre}}$: Performance per question.

samples we draw (Figure 8), and for the same results for the fine-tuned SBERT model (Figure 9) refer to the Appendix.

In Figure 2, we see that the pattern of using the LLM-generated data as training data being superior to zero-shot scoring with an LLM (green bars) is consistent across the majority of questions. For some questions, even the fine-tuned model is not doing much better than the majority baseline. Only for one of the questions for which a successful model can be learned on SRA do we see comparable performance when using SRA-gen as training data. Do however note that this only holds for the pretrained model. Fine-tuning SBERT on SRA outperforms training on SRA-gen for all questions.

6.2 Amount of Generated Training Data

As generating training data puts us at liberty to surpass the amount of data that is present in the

original dataset, we now explore how performance changes with larger amounts of synthetic data. We do this in a balanced fashion, i.e. with an equal amount of answers for each label, starting with just one per label and going up to 100. This means that we use training data ranging from as little as five to as many as 500 answers. We sample each amount 20 times, and report average, best and worst performance.

We again choose to do this analysis for the pre-trained SBERT model, as this is the model that gave the best performance on SRA-gen in the previous experiment. To see the full curve, refer to Figure 10 in the Appendix. From a low amount of training data onwards, performance remains on a consistent low level. The average performance is below the majority baseline performance of .21 macro-averaged F1, and even the best runs do just slightly better than this baseline. Thus, the relatively low performance on SRA-gen we saw in the previous experiment was not due to the limited amount of training data. Do also note that the balanced label distribution we enforce here leads to overall lower performance than what we had observed in the previous experiment, where training and test data shared the same label distribution.

7 Training on Cleaned Synthetic Data

Apart from the answers themselves, their labels are a crucial element of the generated data. While we are asking the LLM to generate answers that conform to a target label, there is no guarantee that they actually do. Thus, we perform manual annotation to assess whether the generated answers match the label they are supposed to belong to. Table 5 shows the three questions we select for this assessment.

We first manually clean the labels and then run scoring experiments that compare performance of training on the **as-generated** labels vs. the manually **cleaned** labels.

7.1 Label cleaning

As a first calibration round, three annotators (two authors of this paper and a research assistant) manually label the answers in SRA to make sure that there is substantial agreement with the original labels. Table 6 shows the Cohen's Kappa (Cohen, 1960) we obtain.[3] We also include agreement with

[3]Note that we believe to have found two mislabeled instances for question ME_27b, and one for question VB_1. We report agreement with the corrected labels.

ID	Question	# Answers c.	p.c.	con.	irr.	n.-d.
ME_27b	*How can you use a magnet to find out if the key is iron or aluminum?*	22	12	1	4	1
PS_4bp	*Darla tied one end of a string around a doorknob and held the other end in her hand. When she plucked the string (pulled and let go quickly) she heard a sound. How would the pitch change if Darla made the string longer?*	24	0	10	6	0
VB_1	*How do you define a controlled experiment?*	21	3	1	14	1

Table 5: Questions chosen for manual annotation.

	ME_27b				PS_4bp				VB_1			
	G	R1	R2	R3	G	R1	R2	R3	G	R1	R2	R3
Gold	-	.77	.62	.84	-	.91	.91	.96	–	.79	.72	.91
R1	.77	-	.45	.69	.91	-	.91	.91	.79	–	.80	.83
R2	.62	.45	-	.62	.91	.91	-	.91	.72	.80	–	.68
R3	.84	.69	.62	-	.96	.91	.91	-	.91	.83	.68	–
Adj.	.88	.73	.66	.96	.95	.95	.95	.95	.83	.96	.84	.83

Table 6: Kappa agreement of our annotations with the labels in SRA (Adj. = adjudicated annotations).

the adjudicated labels, which we determined by taking the majority label. Where all three annotators had decided on different labels (two cases), the disagreement is resolved via discussion. Agreement between adjudicated labels and gold SRA labels ranges from .83 to .95. This shows that we can reliably annotate the data. Thus, we proceed with annotating the same prompts in SRA-gen.

For each of the three questions, we take 50 answers per label. This makes for a total of 250 answers per question, of which we randomize the order and hide the as-generated label. All three annotators now annotate the answers and we again derive adjudicated annotations by taking the majority label where possible. The remaining cases where all annotators disagree (12 for question ME_27b, 9 for question PS_4bp, 41 for question VB_1) are resolved through discussion. Table 9 in the Appendix shows kappa agreement for this round of annotation. Agreement is overall lower, as the LLM-generated data has substantially more variance than the original SRA data.

Label Accuracy With the manual label annotations we can now compute the accuracy for each label by comparing what the LLM was asked to generate with what the annotators agreed was actually generated. Table 7 shows these results, and

LLM Label	Human Label corr.	part. corr.	contr.	irr.	non-d.	Acc.
ME_27b						
correct	12	13	3	22	0	.24
partially correct	13	17	4	16	0	.34
contradictory	3	2	18	26	1	.36
irrelevant	1	4	6	37	2	.74
non-domain	0	2	3	1	44	.88
PS_4bp						
correct	22	3	14	11	0	.44
partially correct	14	14	8	14	0	.28
contradictory	3	9	26	12	0	.52
irrelevant	0	0	3	47	0	.94
non-domain	6	2	2	11	29	.58
VB_1						
correct	23	26	0	1	0	.46
partially correct	26	18	3	3	0	.36
contradictory	0	13	14	23	0	.28
irrelevant	2	4	8	35	1	.70
non-domain	0	0	4	10	36	.72

Table 7: Adherence of the LLM to the label it was asked to generate answers for. Accuracy: the fraction of the 50 generated answers that does match the desired label.

Figure 3 compares label accuracies across questions. Only for one question and label (*irrelevant* for PS_4bp) nearly all generated answers conform to the desired label. Non-domain answers are only generated when the LLMs is asked for such: very rarely is an answer from a different label manually found to be *non-domain*. Overall, accuracy of *non-domain* and *irrelevant* answers is higher than for the other labels. Consistently, over half of the *correct*, *partially correct* and *contradictory* answers do not conform to the desired label. *Contradictory* answers are often determined to be *irrelevant*, and for VB_1 13 of them are even partially correct. *Correct* answers are regularly found to actually be *partially correct* or *irrelevant*. For PS_4bp, 14 correct answers are even found to in fact be *contradictory*. This is somewhat contrary to the general consensus that LLMs are doing well with answering correctly. It may however be due to a difficulty of having to come up with *multiple* answers in one go, i.e. *ten* correct answers instead of just one.

7.2 Model training with cleaned data

To assess the benefit of cleaning labels in SRA-gen, we can now compare the success of models trained on the as-generated vs. cleaned labels. Table 8 summarizes these results. When training on 40 instances from SRA-gen, we draw a sample with the same distribution as in SRA 20 times and report the average performance. Training on as-generated SRA-gen data consistently does worse when a balanced sample of 250 answers vs. just 40 answers is used to train. This is likely due to models benefiting from the matching label distribution in training and test data for the smaller sample.

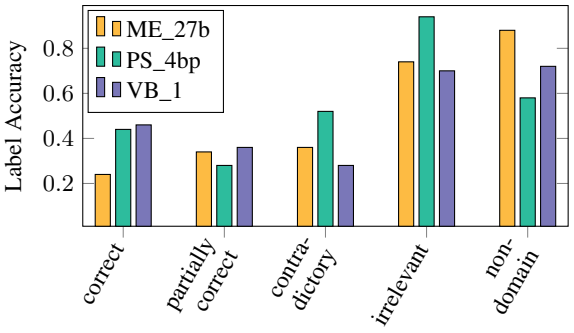

Figure 3: Accuracy of the labels in SRA-gen.

The cleaned labels consistently lead to an increase in performance. For the 40 training answers, this increase is however much more subtle than for the full 250 SRA-gen answers. On this larger amount of training data, performance often reaches the level of training on the original SRA data. The SBERT model consistently gives the best performance, and is the only model for which training on the 250 cleaned SRA-gen answers consistently outperforms training on the 40 SRA answers.

Overall, our results demonstrate that the LLM-generated answers themselves do carry enough meaning to inform a model, but that manual cleaning is necessary to remove noise in their labels. As we have seen the label distribution in the training data to affect model performance, the comparison between the 250 as-generated vs. cleaned SRA-gen answers is however not entirely 'fair': While the SRA-gen data was drawn with a balanced distribution of 50 answers per label, this distribution has shifted once the labels were cleaned. We therefore take a look at the performance of balanced sampling with as-generated vs. cleaned SRA-gen data in Figure 4. Since the fine-tuned SBERT model gives the best performance on cleaned SRA-gen data, we choose this model for this analysis. Do note that we can only compute the curve for the cleaned data up to 28 answers per label, as the total number of answers for the most infrequent label limits our ability to draw a balanced sample. For training with 5, 10, 15, 20 and 25 answers per label, we draw 20 training samples each and report best, average and worst performance.

For all three questions, the as-generated labels constantly lead to low average performance lev-

Data	LR					BERT					$\text{SBERT}_{\text{pre}}$					$\text{SBERT}_{\text{fine}}$				
# train	40	40	40	250	250	40	40	40	250	250	40	40	40	250	250	40	40	40	250	250
Generated	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓
Cleaned	-	✗	✓	✗	✓	-	✗	✓	✗	✓	-	✗	✓	✗	✓	-	✗	✓	✗	✓
ME_27b	.34	.16	.19	.09	.32	.36	.10	.17	.07	.32	.33	.20	.31	.26	.29	.33	.12	.29	.08	.41
PS_4bp	.58	.25	.25	.00	.46	.49	.29	.27	.03	.47	.73	.32	.45	.00	.21	.82	.26	.61	.07	.91
VB_1	.33	.13	.30	.06	.37	.33	.24	.29	.09	.31	.33	.28	.25	.16	.46	.35	.22	.31	.02	.41
Avg.	.42	.18	.25	.05	.38	.39	.21	.25	.06	.37	.46	.27	.34	.14	.32	.50	.20	.40	.06	.58

Table 8: Effect of cleaning the LLM labels via manual annotation (macro-averaged F1). For BERT and SBERT, results are averaged across three runs for a more reliable performance estimate. 'Generated' denotes whether we are training on SRA (✗) or SRA-gen (✓). 'Cleaned' indicates if we are using the as-generated (✗) or cleaned (✓) labels.

els of below .2 F1. With the cleaned labels, performance rises once more data is added, and the curves indicate that it might rise further if there was more data available. This controlled comparison thus confirms the beneficial effect of manually cleaning the labels.

8 Conclusion

We generate answers to the questions in the SRA dataset with an LLM. Using these answers as training data leads to relatively poor performance. Directly scoring the SRA data with an LLM even performs slightly worse, showing an inability of the model to reliably apply the 5-way label scale. This is supported by our analysis of the extent to which the LLM sticks to the label we ask it to generate answers for. Up to 75% of the answers the model was asked to generate for a specific label were found not to conform to this label. Training a model with manually relabeled generated data demonstrates the detrimental effect of the noisy labels: With cleaned labels, model performance increases substantially, reaching a comparable level to training on the original SRA data - albeit at the demand of larger volumes of training data. In light of our analysis of the lexical diversity in SRA vs. SRA-gen, this is likely due to diverging content in SRA vs. SRA-gen. Thus, more SRA-gen data is needed to sufficiently cover the content of SRA.

With a similarity based scoring model, training on the larger sample of generated data even consistently leads to superior performance over training on the small amount of available original SRA data. One benefit of the similarity-based model might be that one highly similar answer with the correct label suffices for the model to correctly label an answer of interest.

In conclusion, one *can* overcome the cold-start problem with the help of an LLM in the sense of not having to collect data from real students, but not without the manual effort of labeling the generated answers. Future work could explore automatic

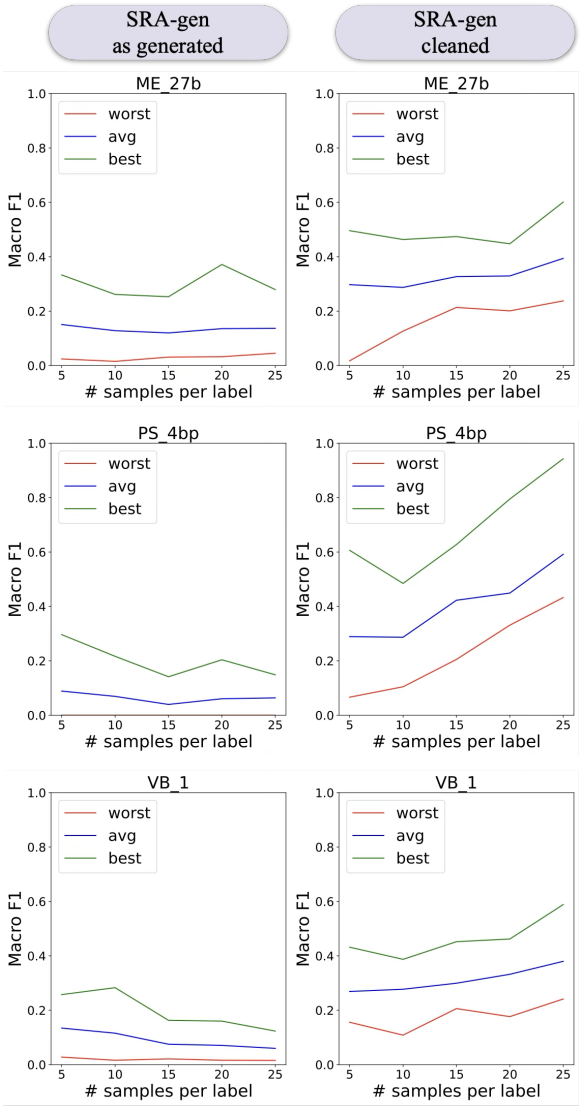

Figure 4: $\text{SBERT}_{\text{fine}}$: Balanced sampling of SRA-gen with as-generated (left) vs. cleaned (right) labels.

cleaning of the generated data to alleviate the manual labeling effort. While we saw limited success in preliminary experiments, future work could also quantify the effect of using few-shot prompting, both in zero-shot labeling and generating answers.

Limitations

While our results provide interesting insights into the possibility of generating training data with an LLM, there are a number of limitations to our findings. First, we only experiment with one LLM. Other LLMs may behave differently, which limits our conclusions to DeepSeek-v2. Even within the realm of prompting an LLM, the precise choice of prompt can have substantial impact (Sclar et al., 2024). While we did carefully craft our prompts, subtle changes to the wording may affect results. Within the prompt design, a key aspect might be the amount of answers the model is asked to generate in one go. We always asked for ten answers, but results may differ if the model were asked to generate just one or even all 500 answers at once.

Even beyond model choice and prompt design, model parameters will affect results. We left these untouched, but varying the temperature will affect both answer generation and scoring ability of the model.

Ethical Considerations

In considering the use of generated training data for model training, one has to be cautious about the normative language LLMs produce. An inability to produce sufficiently 'student-like' language may lead to a model with inferior performance on real student answers that deviate from language norms. Since content scoring is however less about language form and more about content, this should not affect the score of an answer.

Automated scoring of student answers in general is not without ethical and legal issues. It is high-risk as per the European Union AI Act, and LLM use poses 'systematic risks'.

A main concern of LLMs and deep learning in general is a lack of transparency. This is somewhat alleviated by the use of an LLM to generate synthetic answers as opposed to using it to directly score student answers. Still, our work shows that based on the synthetic answers it is again most successful to apply deep learning. This in turn is much less transparent than the use of a shallow learning method such as logistic regression - which we test as well, but find to perform worse. However, the deep learning model we find to perform best operates in a similarity-based fashion. Thus, it at least allows backtracking to the reference answers that lead to a predicted score.

Achnowledgements

We thank Kristina Spenner for her valuable assistance with data annotation and experimental work. We also thank the reviewers for their insightful comments and constructive feedback.

References

Lorin W Anderson and Lauren A Sosniak. 1994. *Bloom's taxonomy*. Univ. Chicago Press Chicago, IL).

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2022. Similarity-based content scoring - how to make S-BERT keep up with BERT. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 118–123, Seattle, Washington. Association for Computational Linguistics.

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2023. Similarity-based content scoring-a more classroom-suitable alternative to instance-based scoring? In *Findings of the association for computational linguistics: Acl 2023*, pages 1892–1903.

Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2025. Distractor Generation for Multiple-Choice Questions with Predictive Prompting and Large Language Models. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 48–63, Cham. Springer Nature Switzerland.

Imran Chamieh, Torsten Zesch, and Klaus Giebermann. 2024. LLMs in Short Answer Scoring: Limitations and Promise of Zero-Shot and Few-Shot Approaches. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 309–315, Mexico City, Mexico. Association for Computational Linguistics.

Derek Chen, Celine Lee, Yunan Lu, Domenic Rosati, and Zhou Yu. 2023. Mixture of Soft Prompts for Controllable Data Generation. *arXiv preprint*. ArXiv:2303.01580 [cs].

Jonathan H Choi, Kristin E Hickman, Amy B Monahan, and Daniel Schwarcz. 2021. Chatgpt goes to law school. *J. Legal Educ.*, 71:387.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46. _eprint: https://doi.org/10.1177/001316446002000104.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv preprint*. ArXiv:2405.04434 [cs].

Tu Anh Dinh, Carlos Mullov, Leonard Bärmann, Zhaolin Li, Danni Liu, Simon Reiß, Jueun Lee, Nathan Lerzer, Jianfeng Gao, Fabian Peller-Konrad, Tobias Röddiger, Alexander Waibel, Tamim Asfour, Michael Beigl, Rainer Stiefelhagen, Carsten Dachsbacher, Klemens Böhm, and Jan Niehues. 2024. SciEx: Benchmarking Large Language Models on Scientific Exams with Human Expert Grading and Automatic Grading. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11592–11610, Miami, Florida, USA. Association for Computational Linguistics.

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 263–274, Atlanta, Georgia, USA. Association for Computational Linguistics.

Rafael Ferreira Mello, Cleon Pereira Junior, Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Newarney Costa, Geber Ramalho, and Dragan Gasevic. 2025. Automatic short answer grading in the llm era: Does gpt-4 with prompt engineering beat traditional models? In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, LAK '25, page 93–103, New York, NY, USA. Association for Computing Machinery.

Jiahui Gao, Renjie Pi, LIN Yong, Hang Xu, Jiacheng Ye, Zhiyong Wu, WEIZHONG ZHANG, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023. Self-guided noise-free data generation for efficient zero-shot learning. In *The Eleventh International Conference on Learning Representations*.

Abigail Gurin Schleifer, Beata Beigman Klebanov, Moriah Ariely, and Giora Alexandron. 2024. Anna karenina strikes again: Pre-trained LLM embeddings may favor high-performing learners. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 391–402, Mexico City, Mexico. Association for Computational Linguistics.

Mohammad Hassany, Peter Brusilovsky, Jaromir Savelka, Arun Balajiee Lekshmi Narayanan, Kamil Akhuseyinoglu, Arav Agarwal, and Rully Agus Hendrawan. 2025. Generating Effective Distractors for Introductory Programming Challenges: LLMs vs Humans. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, LAK '25, pages 484–493, New York, NY, USA. Association for Computing Machinery.

Mingxin Liu, Tsuyoshi Okuhara, XinYi Chang, Ritsuko Shirabe, Yuriko Nishiie, Hiroko Okada, and Takahiro Kiuchi. 2024. Performance of ChatGPT Across Different Versions in Medical Licensing Examinations Worldwide: Systematic Review and Meta-Analysis. *Journal of Medical Internet Research*, 26:e60807.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating Training Data with Language Models: Towards Zero-Shot Language Understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. GPT-4 Technical Report. *arXiv preprint*. ArXiv:2303.08774 [cs].

Saba Rahimi, Tucker Balch, and Manuela Veloso. 2023. Exploring the Effectiveness of GPT Models in Test-Taking: A Case Study of the Driver's License Knowledge Test. *arXiv preprint*. ArXiv:2308.11827 [cs].

Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Dragan Gašević, Geber Ramalho, and Rafael Ferreira Mello. 2024. Assessing the quality of automatic-generated short answers using GPT-4. *Computers and Education: Artificial Intelligence*, 7:100248.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Shiva Taslimipoor, Luca Benedetto, Mariano Felice, and Paula Buttery. 2024. Distractor generation using generative and discriminative capabilities of transformer-based models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5052–5063, Torino, Italia. ELRA and ICCL.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. *arXiv preprint*. ArXiv:2202.07922 [cs].

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Torsten Zesch, Andrea Horbach, and Fabian Zehner. 2023. To score or not to score: Factors influencing performance and feasibility of automatic content scoring of text responses. *Educational Measurement: Issues and Practice*, 42(1):44–58.

Appendix

This appendix contains some supplementary material to increase transparency of our experiments. It includes the prompt used to generate SRA-gen in Figure 5 and the prompt used to score the answers in SRA in Figure 6. The main paper contains the agreement we achieve in labeling the original SRA data in Table 6. Here, we include the same statistics for our annotation of the generated data in Table 9. We also include some more detailed results of our scoring experiments. Due to the non-deterministic nature of the LLM, repeated administration of the same prompt leads to differing results. Thus, Figure 7 depicts the variation in performance when administering the same prompt to the same model five times. Figures 8 (pretrained SBERT) and 9 (SBERT) show question-wise results for scoring based on SRA-gen vs. SRA. Finally, Figure 10 shows performance of training the pretrained SBERT model with balanced samples of SRA-gen.

	LLM	R1	R2	R3	Adjudicated
ME_27b					
LLM	-	.49	.19	.36	.39
R1	.49	-	.32	.48	.62
R2	.16	.32	-	.48	.63
R3	.36	.48	.48	-	.81
Adjudicated	.39	.62	.63	.81	-
PS_4bp					
LLM	-	.55	.33	.39	.45
R1	.55	-	.59	.59	.77
R2	.33	.59	-	.56	.75
R3	.39	.59	.56	-	.75
Adjudicated	.45	.77	.75	.75	-
VB_1					
LLM	–	.63	.26	.22	.34
R1	.63	–	.41	.31	.55
R2	.26	.41	–	.46	.69
R3	.22	.31	.46	–	.59
Adjudicated	.34	.55	.69	.59	–

Table 9: Kappa agreement of our annotations with the labels in SRA-gen.

```
<purpose>
You are a school teacher.
Your students are going to answer the following question:
{question}

You are now thinking about possible answers students could give.

[LABEL_INSTRUCTIONS]
</purpose>
<format_rules>
Use markdown output and put each correct answer as a single bullet point.
Keep the answers as short as possible. A maximum of 20 words per answer.
</format_rules>
<output>
Create 10 [correct/partially correct or incomplete/contradictory/irrelevant/non domain]
responses following the given rules.
</output>


LABEL_INSTRUCTIONS={

CORRECT: Generate a list of 10 possible correct answers.
That is the important part, generating that list of exactly 10 answers!

PARTIALLY_CORRECT_INCOMPLETE: Generate a list of 10 possible partially correct
or incomplete answers. Partially correct or incomplete means that the student answer is
a partially correct answer containing some but not all information from the reference
answer. The important part is to generate a list of 10 student answers belonging to that
category (partially correct incomplete)!

CONTRADICTORY: Generate a list of 10 possible contradictory answers. That means that
the given answers are not correct and explicitly contradict the correct answer. The
important part is to generate a list of 10 answers belonging to that contradictory
category!

IRRELEVANT: Generate a list of 10 possible irrelevant answers. Irrelevant means that
the student answer is talking about domain content but not providing the necessary
information to be correct. The important part is to generate a list of 10 student answers
belonging to that irrelevant category!

NON_DOMAIN: Generate a list of 10 possible 'non domain' answers.  'Non domain' means that
the student utterance does not include domain content, e.g., "I don't know", "what the
book says", "you are stupid". The important part is to generate a list of 10 student
answers belonging to that category!}
```

Figure 5: Prompt used to generate training data. We follow the

```
<purpose>
You are a school teacher.
A student has answered the following question:
{question}
This is the answer the student gave:
{answer}
You now have to score this answer.
These are the possible scores:
Correct: A correct answer to the question.
Partially correct or incomplete: This means that the student answer is a partially
correct answer that contains some but not all necessary information.
Contradictory: This means that the student answer is not correct and explicitly
contradicts the correct answer.
Irrelevant: This means that the student answer is talking about domain content but
not providing the necessary information to be correct.
Non-domain: This means that the student answer does not include domain content, e.g.,
"I don't know", "what the book says", "you are stupid".
</purpose>
<format_rules>
Only output the score.
</format_rules>
<output>
Decide on the score of the student answer.
</output>
```

Figure 6: Prompt used to score answers with the LLM.

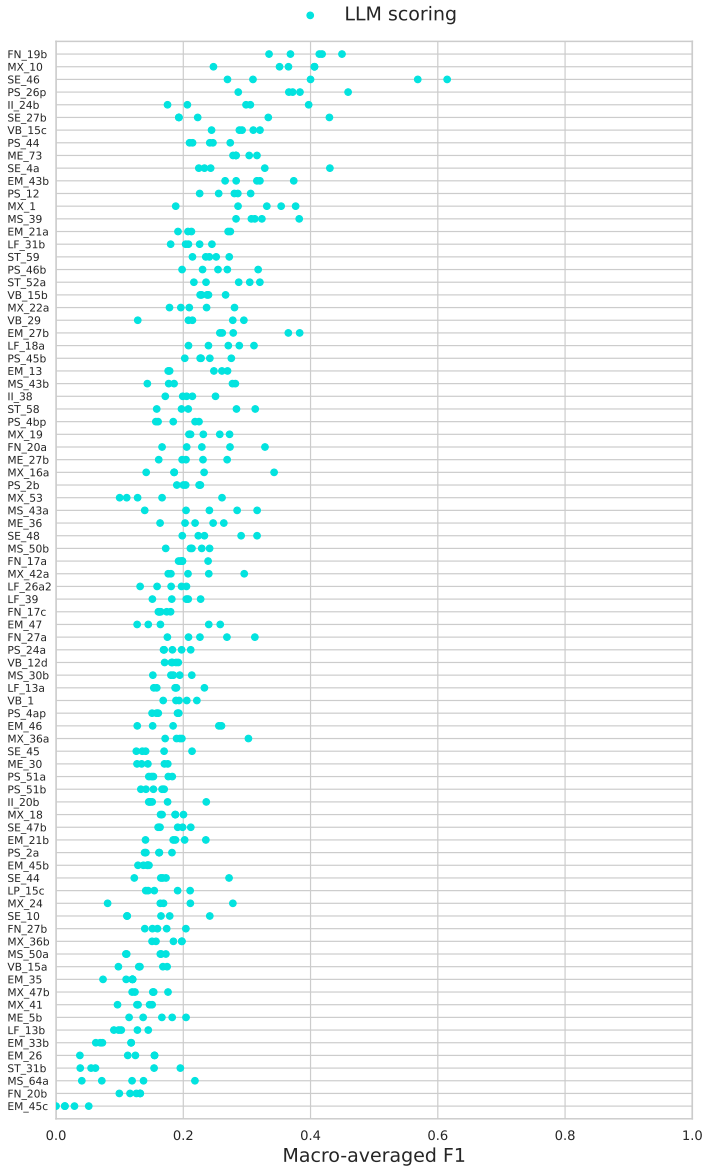

Figure 7: Performance variation across five runs of scoring the answers using an LLM.

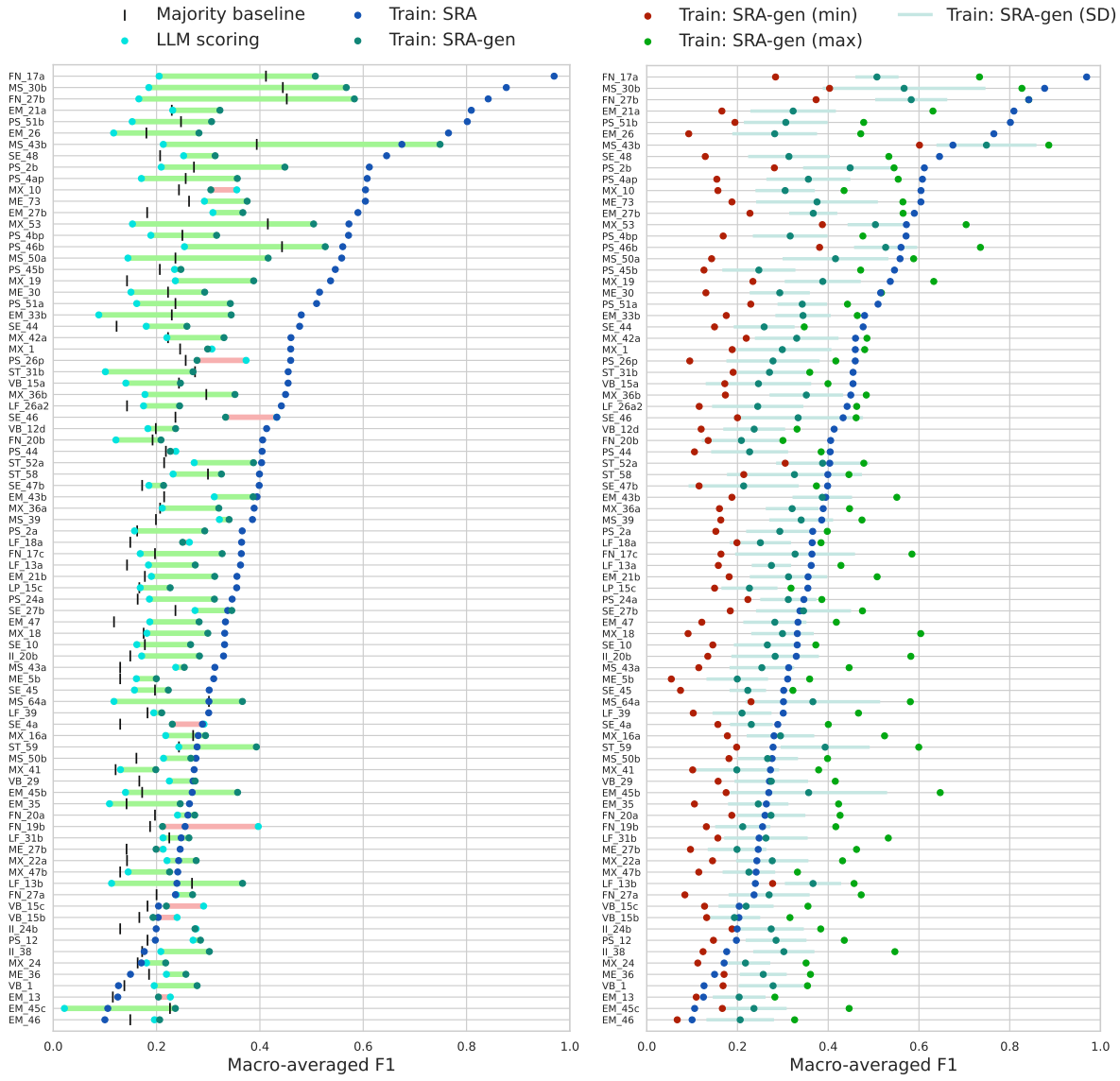

Figure 8: $\text{SBERT}_{\text{pre}}$ performance variation across 20 samples of generated training data that follow the same label distribution as the original SRA data. Left: Comparison of the average performance to directly scoring the data with an LLM. Right: Detailed results of the best, average and worst sample.

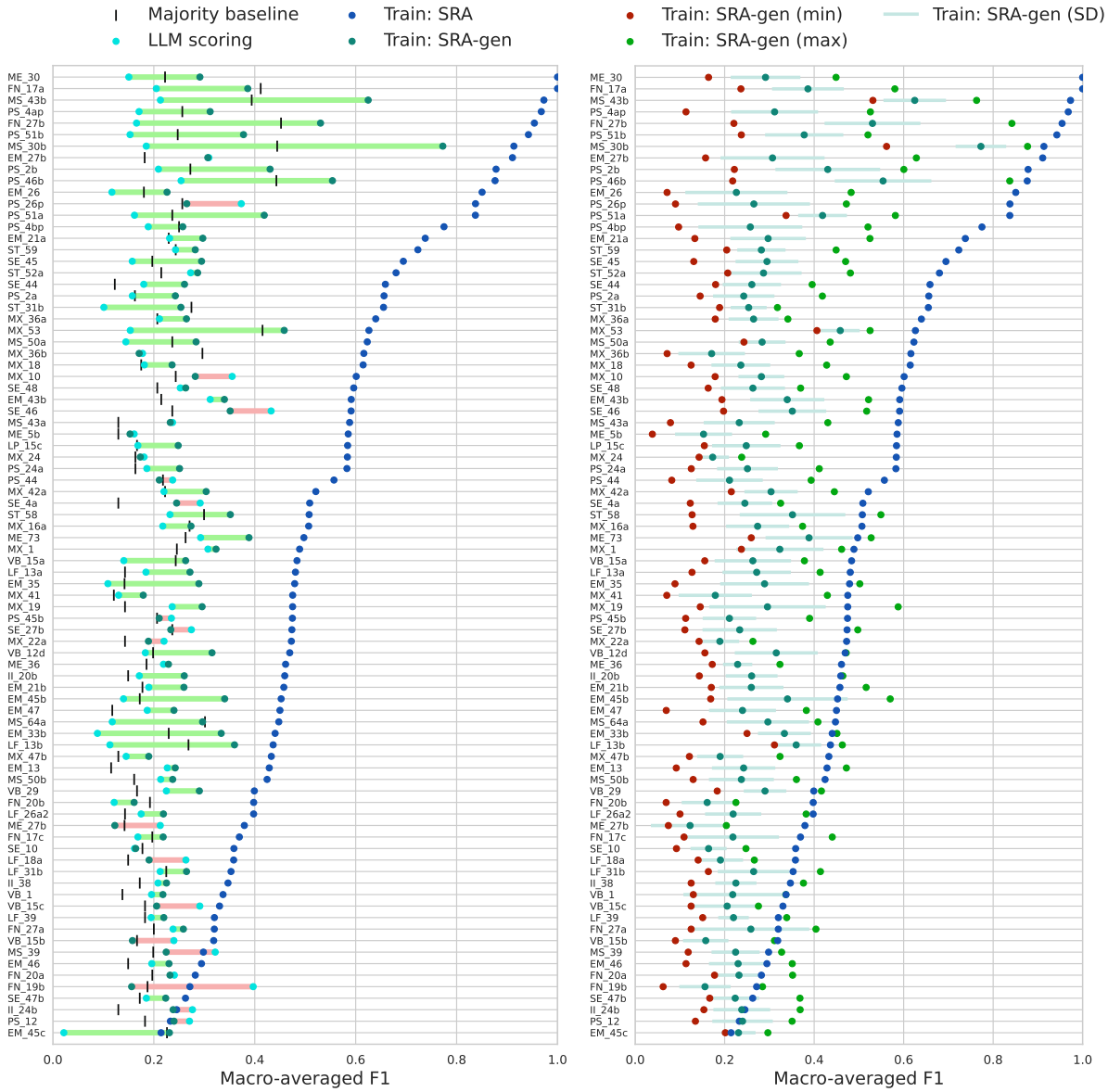

Figure 9: $\text{SBERT}_{\text{fine}}$ performance variation across 20 samples of generated training data that follow the same label distribution as the original SRA data. Left: Comparison of the average performance to directly scoring the data with an LLM. Right: Detailed results of the best, average and worst sample.

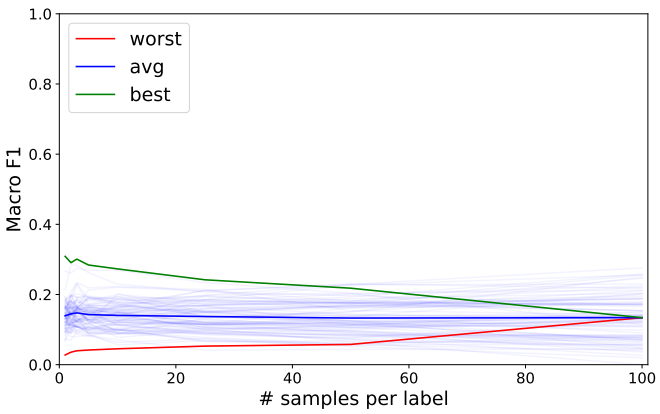

Figure 10: Average performance of $\text{SBERT}_{\text{pre}}$ when using a balanced sample of SRA-gen training data. Light blue lines show average results for individual questions.