Reliability and Robustness of Transformers for Automated Short-Answer Grading

Anonymous ACL submission

Abstract

Short-Answer Grading (SAG) is an application for NLP in education where student answers to open questions are graded. This task places high demands both on the reliability (accuracy and fairness) of label predictions and model robustness against strategic, "adversarial" input. Neural approaches are powerful tools for many problems in NLP, and transfer learning for Transformer-based models specificially promises to support data-poor tasks as this. We analyse the performance of a Transfomer-based SOTA model, zooming in on class- and item type specific behavior in order to gauge reliability; we use adversarial testing to analyze the the model's robustness towards strategic answers. We find a strong dependence on the specifics of training and test data, and recommend that model performance be verified for each individual use case.

1 Introduction

Short-Answer Grading (SAG) is a popular application of NLP in education. Students write one to three sentences in response to open test questions, and the task is to predict the grade based on answer content. The prediction can be passed on directly as feedback to students and teachers or serve as input for human review (in order to reduce manual grading effort). Use cases range from automated feedback to low-stakes quizzes and self-tests to suggestions for human review on higher-stakes tests. (We focus on ad-hoc, non-standardized testing.)

Given the nature of the task, system predictions have to be **reliable** (accurate overall and fair across all labels), and **robust** towards strategic "adversarial" answers in order to be informative and be accepted by teachers and students.

A long-standing challenge for SAG is the small size of annotated corpora (commonly in the thousands of data points, Burrows et al., 2015). Recently, the ascendance of transfer learning for Transformer-based models like BERT (Devlin et al., 2019) allows the use of large amounts of unannotated data to infer a robust language model in pre-training before switching to fine-tuning for a specific task. This strategy has proven very successful on the range of language understanding tasks of the standard GLUE data sets (Wang et al., 2018). Results on SAG data (Ghavidel et al., 2020; Camus and Filighera, 2020) are also promising.

The good performance of Transformer-based methods on literature benchmarks raises the question of whether they are appropriate to provide automated grade feedback. However, overall model F Scores as reported for result benchmarking are not informative enough to estimate both reliability of grades in terms of low, balanced labelling error and robustness to strategic answers.

In this paper, we employ straightforward methods to estimate both aspects of model appropriateness: We first evaluate reliability by label-level F Scores and item-type accuracy (Exp. 1, Sec. 5), and then create adversarial attacks focusing on plausible test-taker strategies (Exp. 2, Sec. 6). We present (1) example results for a literature data set to establish a sense of what can be expected of the models in real life and (2) strategies for practioners to verify model appropriateness for their use case.

2 Previous Work

Task and Data In many real-world settings, systems will be required to work well for previously unseen questions. Researchers have traditionally approached this challenge with feature-based, non-neural algorithms that compare student answers to a correct reference answer (Burrows et al., 2015). There is a strong parallel between this approach to SAG and Natural Language Inference (NLI) emphasized in the SemEval-2013 shared task (Dzikovska et al., 2013): In both, a hypothesis (in SAG, a student answer) is compared to a premise (in SAG, a reference answer). The shared task data sets, the Beetle and SciEntsBank (SEB) corpora,

are still important ressources for SAG. The Beetle data contains several correct reference answers per question; since many systems are designed to work with a single reference answer, results are often reported for SEB only. Three parallel annotations at different levels of granularity (2-way, 3-way, 5-way) are present.

Distributional and neural models Sultan et al. (2016) was the first to show that the use of the distributional information in word embeddings (along with the alignment of reference and student answer) for training a traditional classifier is helpful. Saha et al. (2018) achieved further improvement by combining token-level similarity features and sentence embeddings, but found that the sentence embedding feature is less and less informative the further away from the training domain the test data is.

The first experiments with neural network SAG are reported in Riordan et al. (2017). Adapting and evolving the Taghipour and Ng (2016) approach to essay scoring, they trained an LSTM over word embeddings. The results are less encouraging for SAG than for the longer essays: The model just reaches the state of the art on one of three data sets and underperforms it on the other two. The reason is likely the lack of training data for SAG since corpora are small and each data point is at most a few sentences long.

Transfer Learning for SAG Transformer-based models promise advantages for underressourced tasks like SAG through pre-training on unannotated data. Sung et al. (2019) generalized this idea and collected additional domain-specific data for a second pre-training round before fine-tuning BERT (Devlin et al., 2019) on the SEB corpus. They showed an improvement over the state of the art, but evaluated for the 3-way SAG task only and did not compare to off-the-shelf BERT. Ghavidel et al. (2020) completed this comparison, experimenting with BERT and XLNet (Yang et al., 2019), an auto-regression variant of the Transformer family, on SEB data. They found that off-the-shelf BERT's accuracy is more stable for out-of-domain test sets than the Sung et al. (2019) version, which suggests that their domain-specific pre-training approach is too tightly focused to generalise well. BERT and XLNet performed similarly and appear to come into their own as the number of classes increases and conversely, the number of training instances per class decreases.

Also pursuing the idea of choosing models with optimal pre-training for SAG, Camus and Filighera (2020) looked at the large range of available pre-trained models from the BERT family and found that larger models perform better due to the larger number of parameters available for further learning during fine-tuning and that pre-training for an NLI task before fine-tuning for SAG offers a noticeable performance gain over the base version (especially for RoBERTa, Liu et al., 2019). Evaluation was done on the on the 3-way SemEval test data, using model-level Accuracy and F Scores.

Robustness is in focus in Ding et al. (2020). They are the first to use the adversarial testing paradigm to expose an LSTM-based neural model and a rule-based approach to cheating strategies such as providing a list of relevant keywords instead of a coherent answer text. While the neural model performs better in terms of reliability, they find that the rule-based model is overall more robust and can be tuned more easily by removing features that increase its vulnerability to the adversarial attacks. However, they report that some reliability is lost by removing those features. Unfortunately, their results are not directly comparable due to the use of a different model architecture and different SAG training set.

3 Research Questions

In this paper, we ask whether whether SOTA SAG models are appropriate for real-life automated SAG on two dimensions:

Reliability Around 15% disagreement between human graders have been accepted for published SAG corpora from non-standardized testing situations (Mieskes and Padó, 2018). We verify that SOTA models can reach this overall performance level. Then, we test for grading imbalances like undue strictness or lenience. We also ask whether any imbalance patterns are stable and predictable or whether they vary with the data sets, which would increase the need for case-based investigation of model behavior.

Please note that we cannot look at algorithmic fairness proper (see,e.g., Kizilcec and Lee, 2021) which compares model predictions for different subgroups of students because we have no further information about the students' abilities and background in our data set.

Robustness Any model that provides feedback to students will face strategic input as students attempt to "game the system" and gain points despite be-

ing unsure about the correct answer. Additionally, input may be inadvertently garbled or incomplete and should still be labelled in a human-like fashion. Only models that are robust towards strategies like chaining together relevant keywords or producing very long, irrelevant answers can be used to provide feedback without human monitoring.

Due to the wide range of usage contexts for SAG in education, we do not attempt to define general minimum requirements for model quality. We leave it to practitioners to define requirements for their individual use case and instead aim to provide an intuition of what to expect from a SOTA model.

4 Approach

Reliability To evaluate the reliability of SOTA models, we will first pick such a model from an array of approaches trained on the 2-way SAG task (grading answers as `correct-incorrect`). We compare the base version of the Transformer models, fine-tuning on the GLUE MNLI (Multi-Genre Natural Language Inference) task and fine-tuning on the GLUE MRPC (Microsoft Research Paraphrase Corpus) – the paraphrase recognition task being also highly relevant to the student-reference answer comparison approach to SAG.

In Exp. 1 (Sec. 5), class-based Precision and Recall will identify any grading imbalances towards one of the target classes and the fine-grained 5-way annotation available for SEB and Beetle will help identify the origin of those imbalances. This information is important for teachers and students when interpreting the model output.

Robustness towards garbled or strategic input is in focus in Exp. 2 (Sec. 6). We will use the adversarial testing approach and generate multiple sets of synthetic test data to analyse the model's ability to resist various test gaming strategies.

4.1 Data

We work with the SemEval-2013 data[1]. It is a standard English-language data set consisting of the Beetle and SciEntsBank (SEB) corpora. The corpora contain student answers to science domain questions; Beetle (3.6k answers) was collected from interactions with a tutoring system, while SEB (4.5k answers) stems from a conventional test setting. Both corpora offer in-domain (unseen answers to seen questions, UA) and out-of-domain test sets (answers to unseen questions, UQ, and, for SEB, from an unseen domain, UD). This allows us to gauge the dependence of the models on keywords seen in training and helps avoid data leakage between training and test (Elangovan et al., 2021; Lewis et al., 2021).

In order to cleanly set hyperparameters, we created a development set in the UA setting by pseudo-randomly selecting roughly 10% of the training data.[2] Across all data sets, the `incorrect` answers are the majority class; their percentage is at about 60% consistently across all data subsets.[3]

5 Exp. 1: Reliability

5.1 Model Training and Selection

We begin by creating a SOTA model for 2-way SAG. From the literature, we choose three well-performing models and three pre-training regimes to compare. The models are BERT and XLNet from (Ghavidel et al., 2020) and RoBERTa as the best model in (Camus and Filighera, 2020).

For each model, we choose the *base* version (*uncased* where available) as well as the versions fine-tuned on MRPC and MNLI.[4] The input sequences for SAG fine-tuning were the reference and student answers; the alternative reference answers in Beetle were concatenated. For model sizes, training times and hyperparameter choices, see Appendix B.

Since the three evaluation measures used in SemEval-2013, Accuracy, weighted and macro F_1, are very close in our experiments, we evaluate on weighted F_1 and where needed report macro F_1 for compatibility with the literature.

Table 1 shows the performance of the different models on the development sets. For each combination of model type and previous training regime we give the average weighted F_1 Scores across three different random initialisations for fine-tuning to SAG. F Scores are genereally higher on Beetle, where multiple reference answers offer paraphrases of the correct solution.

Of the three models, BERT performs most consistently and is the best model in all three settings. RoBERTa sometimes achieves similar performance, but twice (Beetle-MRPC, SEB-base) fails

[1] Available from https://www.cs.york.ac.uk/semeval-2013/task7/index.php%3Fid=data.html.

[2] Per selected question, several answers were extracted. See Appendix E for data availability.

[3] See Appendix A for all details on size and label distribution.

[4] All models are available on huggingface.co.

Corpus	Model	Base	MRPC	MNLI
Beetle	BERT	84.20	84.20	**87.10**
	RoBERTa	76.25	47.78	86.94
	XLNet	76.24	70.49	85.72
SEB	BERT	83.87	83.13	**84.56**
	RoBERTa	43.02	82.75	84.45
	XLNet	78.29	64.96	84.52

Table 1: Average weighted F_1 on the development set across three training runs. Fine-tuning for Beetle or SEB on top of the base model, or after first fine-tuning on MRPC or MNLI.

	Beetle		SEB		
	UA	UQ	UA	UQ	UD
SemEval-13	83.3	72.0	76.8	73.7	70.5
Saha et al.	–	–	78.6	**73.9**	**70.9**
BERT$_{MNLI}$	**89.7**	**76.5**	**81.7**	72.8	70.6

Table 2: Macro F_1 on the test sets for literature benchmarks and BERT$_{MNLI}$.

to learn in all three training runs, acquiring only the frequency baseline. XLNet generally lags behind.

Of the three settings, MNLI is clearly the most advantageous for learning SAG on the SemEval-13 data. When using MNLI, the models perform closely together on Beetle and virtually identically on SEB. It appears that the model specifics have very little impact on performance once a sufficient amount of informative training data is used.

We will therefore continue with the robust BERT$_{MNLI}$ models (see Appendix C for the best parameters for each corpus).

5.2 Reliability Analysis

We begin by evaluating BERT$_{MNLI}$ on the test sets on overall F Scores. We report the first results for 2-way Beetle since SemEval-2013[5] and compare to Saha et al. (2018) on SEB.[6]

Table 2 shows that we have succeeded in training a model that outperforms or closely matches the SOTA for both corpora using macro F_1. The performance patterns are the same for weighted F_1 (not available for SemEval-13, see App. D).

In the literature as well as in our results, a clear domain effect is visible: Model performance drops as the test data becomes more dissimilar to the training data. Despite the focus on comparing the input sentences taught by the MNLI data, the model also acquires vocabulary specific to the training data, which becomes less and less relevant for out-of-domain test data. For real-life SAG, this means that as before, models should be expected to be less reliable for unseen questions than for questions seen during training, despite the additional training undergone by Transformers.

On the UA data, BERT$_{MNLI}$ has a prediction error of 10% on Beetle (corresponding to 1-0.90 Accuracy) and of 17.8% on SEB. This is is close to the value of about 15% disagreement between two human annotators that has been accepted for published SAG corpora. Therefore, it appears not implausible to use the model for real-life grading to relieve teacher workloads at this point.

Grading tendencies We look at the F score of the predictions for individual labels in Table 3 to identify imbalances of error. Indeed, the models consistently make more errors predicting `correct`. Also, moving from the in-domain test set UA to the out-of-domain test sets UQ and UD, both labels lose F Score, but the loss for `correct` is much more pronounced.

The reason for this parallel pattern is different for the two data sets when looking at Precision and Recall separately[7]: The Beetle model is overly generous on UA data and labels too many answers as `correct`, with Recall$_c$ at 90.3 and Precision$_c$ at 85.5 for UA (while Precision$_i$ is 93.3). Moving from UA to UQ, it becomes much stricter: Recall$_c$ drops by 28 percentage points, but Precision$_c$ only by four points. Since there are only two possible labels, Precision$_i$ correspondingly drops by 17 percentage points. For real-life applications this means that on seen questions, predictions of `incorrect` are almost certain to be reliable, but the situation reverses drastically on unseen questions, affecting the interpretation of the model output.

The SEB model is too strict on UA data, rejecting a quarter of `correct` answers at a Recall$_c$ of 75.5 (Recall$_i$ is 87.3). Assignments of `correct` conversely are quite reliable at Precision$_c$ of 81.9. Moving to UQ items, the model becomes dramatically more lenient, with Precision$_c$ (and Recall$_i$) dropping by 15 percentage points, while the other measures remain virtually the same. 33% of `correct` and 21% of `incorrect` labels are

[5] Results for the best model for each test set from the top-ranked Heilman and Madnani (2013) and Ott et al. (2013).

[6] Ghavidel et al. (2020) achieved a slightly higher F_1 score for UA at 79.7, but lower scores for UQ and UD.

[7] The full result table can be found in Appendix D.

	Beetle		SEB		
	UA	UQ	UA	UQ	UD
c	87.8	70.4	78.6	68.9	65.0
i	91.5	82.7	84.8	76.8	76.3

Table 3: F for the correct (c) and incorrect (i) classes.

now wrong. On UD, this trend reverses to some extent as Recall_c suffers a drop of nine points; consequently, Precision_i also drops by five points. All categories are now affected strongly by error and labels should be revised by a human grader before being passed on to students.

Item subclasses We can analyse model performance further by using the SemEval-2013 5-way annotation, which applies to the same items as the 2-way annotation, but effectively splits the `incorrect` label into `contradictory`, `irrelevant`, `partially correct/incomplete` and `non-domain`. We bin the test set items into classes according to the 5-way labels and compute the percentage of items for which the binary models appropriately predicted `correct` or `incorrect`. We do not discuss the performance for `non-domain`, which was perfect for both models and corpora.

For Beetle, we know that the UA model is too lenient and accepts `incorrect` answers. The 5-way labels show that the accepted answers are almost exclusively `partially correct` items: 15% were over-generously accepted, while more than 90% of `contradictory` and `irrelevant` items were treated correctly.[8] This is reassuring, since the model errs most in the grey area between correct and incorrect, rather than spuriously accepting clearly incorrect answers. The increase in model strictness on UQ data can be seen exclusively on the `correct` items that are now often being rejected; all other labels are assigned as accurately as before.

For SEB, `correct` answers are rejected too often on UA. This is of course also evident in the 5-way labels, but in addition to 25% rejected `correct` answers, the 5-way classification reveals that 18% of `partially correct` answers are being erroneously accepted.[9] Again, the error is concentrated in the grey area between correct and incorrect answers, but it is not as clearly one-directional as for Beetle, which makes it harder to interpret the labels. Moving to UQ, we see prediction error spreading to other classes as the model's lenience does not improve the amount of accurately labelled `correct` items but rather, `irrelevant` items are now accepted vastly more often(32% of the time instead of 6%). Finally, on UD, the model's ability to recognize `irrelevant` items recovers, (only 14% are erroneously accepted), but `correct` items suffer even more and are rejected 38% of the time.

Discussion Both models struggle more with recognizing `correct` answers and answers in the `partially correct` grey area than they do recognizing clearly `contradictory`, `irrelevant` or `non-domain` answers. However, whether the model is too lenient or too strict depends on the training data. Also, while both models deteriorate on out-of-domain test data, the Beetle model does so only on recognizing `correct` answers, while the error in the SEB model spreads across all classes, making the output much harder to interpret. The reason may be that the students' vocabulary in Beetle is very homegenous and similar to the vocabulary in the reference answers due to alignment with the tutoring system they interacted with. Therefore, question-specific keywords are very informative during training to identify `correct` questions, prompting an over-reliance on this source of information.

The error analysis clearly needs to be carried out for each specific use case, since the error patterns are corpus-dependent and change for out-of-domain test sets: The predictably focused error of the Beetle models is much easier to deal with, for example by human review, than the generalized error of the SEB models.

6 Exp. 2: Robustness

Any grading model used in an educational context also needs to be robust towards strategic input, for example garbled lists of words relevant in the domain. Also, it should not be overly lenient towards insufficient partial answers.

Another Achilles' heel of automated systems is the length bias, since incorrect answers are often much shorter (less detailed, or containing only "I don't know") than correct answers. Indeed, we find this pattern in our data: `correct` Beetle answers have a median length of 54 characters (min: 3,

[8] The full results can be found in Appendix D.

[9] So are 20% of `contradictory` items, but this category makes up only 10% of the data.

max: 367), while `incorrect` answers are only 41 characters long in the median (min: 0, max: 256). For SEB, the numbers are 60 characters (min: 4, max: 532) for correct and 51 (min: 2, max: 413) for incorrect answers.

We use adversarial testing (Goodfellow et al., 2015) and generate synthetic answers to existing questions using several approaches to mimicking these strategies. Ideally, the system will reject all of the synthetic answers, which are highly unlikely to be correct by human standards. We will evaluate system performance using the Attack Rejection Rate (ARR), the percentage of attack items that are labelled as `incorrect` by the system.

Attack strategies There are five different attacks: **Length** items contain sequences of random words that are either very short, very long or of average length for the data. Vocabulary attacks come in different stages of severeness: We begin by randomly stringing together **unigrams**, then move to **bi-** and **trigrams** to create more syntactically and semantically plausible attack items, and finally include **shuffled** versions of the original test items to tease apart the influence of vocabulary and word order. In order to keep the vocabulary attack items comparable, we will clone each real test item using each of the vocabulary attack strategies, preserving its length as closely as possible.

To make the attacks as realistic as possible, we rely on the original vocabulary of the test data. Also, we are interested in the effect of vocabulary differences between `correct` and `incorrect` items and between the different SAG test sets. Therefore, we generate the vocabulary attacks using word frequencies from the test items with the same gold label as the original. Table 4 shows three sample length attack items and the adversarial clones for vocabulary attacks based on a `correct` item from Beetle-UA. The shuffle attack clone differs from the original only in word order, and all n-gram attack clones have the same length as the original and share relevant vocabulary.

In order to isolate length effects, the words for the length attacks are sampled from the complete test set. We generate 200 attack items for each of the three length classes: Short attack items are in the range between the minimum and median length of all relevant answers, the length of medium items is in the range of the first to third quartile and the length of long items is between the median and maximum lengths for the test sets.

Predictions Since our analysis in Section 5 shows a deterioration of performance as the test set vocabulary diverges more from training, we expect to see effects of vocabulary in the n-gram attacks and also expect clones for `correct` test items to be more successful attacks than clones of `incorrect` items. The strongest attack to a vocabulary-based model should be to shuffle `correct` answers. If the models use structure or at least word order, we would expect the n-gram attacks to become more effective with higher n, as longer word sequences are being sampled from real answers. A system that takes word order into account would also be more easily fooled by a trigram-based answer than by a shuffled answer. In addition, it is possible that there will be an effect of length (where longer attack items are more successful) given the observed distribution of answer length over labels.

Vocabulary and structure Table 5 shows the ARRs on the five test sets; a darker cell shade means a higher ARR and more model robustness towards the attack. We also show the percentage of regular test items that the model rejects, as a baseline: if the BERT$_{MNLI}$ labels depend only vocabulary, it will reject the shuffled attack clones exactly like the underlying test items.

The strong impact of vocabulary identity on ARR is immediately visible: Within all the test sets, we find that the n-gram ARRs are very similar across different n and the shuffle attacks (which completely preserve vocabulary) are more successful than the n-gram attacks for all test sets. For Beetle, we also see higher ARRs for the out-of-domain test set with unseen vocabulary.

The model is, however, not ignoring word order: The shuffle ARR is always higher (except for SEB-UD) than the rejection rate of the original test items. This means that quite some items are accepted in their original word order, but rejected when shuffled. Also, we see from the difference between the unigram and shuffle ARRs that the right combination of relevant words (even in the wrong order) is a stronger attack than randomly sampled question- and label-specific words. Therefore, we conclude that the model is not necessarily using word order, but considers word cooccurrences rather than just picking out relevant keywords.

The label-specific ARRs for the shuffle attack (Table 6) allow us to tease apart the model's reaction to `correct` and `incorrect` vocabulary.

As expected, clones of `correct` test items are

Length attacks		Vocabulary Attacks	
Short	was path in is or is closed has incorrect	Unigrams	bulb share are terminal an they
Medium	a affect terminal terminal by bulb off [...] (34 words)	Bigrams	terminal and the bulb electrical state
Long	a and c path state difference bulb [...] (93 words)	Trigrams	an electrical state terminal are connected
		Shuffle	gap they are connected that no
		Original	that they are connected; no gap

Table 4: Adversarial attack items for length and vocabulary attacks.

	Beetle		SEB		
	UA	UQ	UA	UQ	UD
Unigrams	78.8	88.5	68.3	67.5	71.7
Bigrams	77.0	85.6	70.9	66.7	68.8
Trigrams	77.6	84.4	68.2	68.1	67.9
Shuffle	64.9	73.6	62.6	57.6	60.4
Originals	57.6	68.1	60.2	55.5	61.2

Table 5: ARRs for ngram and shuffle attacks and Rejection Rates for the original test items.

rarely, and clones of `incorrect` items almost always, rejected. Importantly, now we see that the items that were accepted in the original but are rejected as shuffle attack clones are almost exclusively `correct` test items (as there is little difference between shuffled and original `incorrect` answers). This again confirms that the model does not do pure keyword spotting, but also considering word order and word cooccurrence.

Length In order to decouple effects of length as much as possible from the strong vocabulary effects identified above, we report the length ARRs for Beetle-UD and SEB-UQ in Table 7. There is a clear trend for long attack items (in the range of the median to the maximum length of the test data set) to be accepted more easily while short and medium attack items are reliably rejected across both corpora: The model learns and uses the correlation between answer length and correct label.

Therefore, caution is needed in a real-world setting if the training data shows length biases: The model is likely to pick them up, and length is a very easily gamed answer property. Fortunately, since only the very longest answers are affected, gaming attempts through answer length can be screened for by a human grader.

Discussion Our adversarial attack experiments have shown that the model pays a lot of attention to correct wording (the shuffle attack from correct original items is the strongest); clearly, combining the right words (as seen in the advantage of shuffle vs. unigram) and putting them in the right order (so that the original item is accepted but the shuffled clone rejected) is also important. This means that a student who tries to pass a question by randomly generating domain keywords is more likely to succeed if they choose a combination of keywords that is relevant for the correct answer – and a student who is able to do this does not really need to strategically fake an answer. Also, our vocabulary experiments have not shown a way to get an `incorrect` attack clone accepted more easily than the original item. Generating extremely long answers does appear to be a promising strategy to fool the model, but can fortunately be easily screened for by human review.

The typical ARRs indicate a need for human oversight, anyway: ARRs of up to 85% for Beetle-UQ may still be acceptable for providing student feedback, but ARRs around 70% for SEB are probably not. Again, we see big corpus-related differences, stressing the need to analyse system behavior specifically for each use case.

On a theoretical note, we observe that a vocabulary-based model's reliability and robustness to strategic answers behave inversely: Reliability is highest when the test and training data are most similar; robustness (i.e. rejection of attack items) is highest when the test vocabulary is different from the training vocabulary, thus avoiding keyword-based erroneous acceptance.

7 Discussion and Conclusions

We have looked at the reliability and robustness of a SAG model by training a Transformer-based model for the 2-way task on the Beetle and SEB corpora and verifying that it matches the SOTA. Specific modelling decisions proved less important in reaching this goal than informative pre-training on the MNLI corpus.

Our focus was on understanding the model's patterns of performance in order to evaluate its appropriateness for real-world settings in an educational

		UA		UQ		UD	
		correct	incorrect	correct	incorrect	correct	incorrect
Beetle	Shuffle	26.1	91.7	49.4	91.8	–	–
	Originals	14.5	93.3	18.6	76.5	–	–
SEB	Shuffle	28.6	88.7	29.7	77.2	37.6	77.5
	Originals	18.1	87.3	33.3	79.1	32.3	74.2

Table 6: Label-specific ARRs for the shuffle attack and Rejection Rates for the original test items.

	Beetle-UQ	SEB-UD
short	97.5	95.5
medium	89.0	83.0
long	43.0	33.0

Table 7: ARRs for length attacks.

context, which requires both correct and balanced predictions and robustness to strategic inputs.

The model's prediction quality as measured in overall F Scores is good and approximates levels of human performance. However, overall F Scores are not detailed enough to understand the usefulness of the model's predictions: A closer look at class-based F Scores and more fine-grained annotation levels revealed that `correct` and `partially correct/incomplete` test items were hardest to label correctly, introducing grading imbalances. These are highly relevant for interpreting the model's grade predictions and for deciding how to use them.

Next, we tested the model's robustness to strategic input (such as chains of relevant keywords or very long, irrelevant content). We found that the model strongly relies on the training vocabulary to spot correct answers, but also considers word-cooccurrence and word order to some degree. In the best case, more than 85% of attack items were rejected, and the best gaming strategy is to combine several keywords relevant to a correct answer, which makes it relatively unlikely that answers with no merit at all will be accepted. However, the model is vulnerable to answer length, so that long answers need to be screened again by a human grader for real-world use. In sum, the model cannot be considered fully tamper-proof.

In the best case, BERT$_{MNLI}$, our SOTA SAG model, is reliable and robust enough to use for formative feedback in real-life, or as a support to human graders for higher-stakes scenarios: Grades are reliable overall with a clearly focused, interpretable grading imbalance and the model is most vulnerable to very long strategic answers, which can be easily identified and screened.

However, this best case performance is by no means guaranteed or predictable. It holds on the Beetle corpus, while on the SEB corpus, both reliability and robustness are generally much worse, and we even see differences in reliability and robustness for different test sets of the same training corpus. Even the direction of grade imbalance can differ between test sets for the same training corpus, despite a similar label distribution. It is therefore vital to closely analyse reliability and robustness of any automated model for the specific use case before deploying it in a real-world education setting.

A second learning again regards the Beetle best case model but generalizes to all SAG models that share its dependence on prompt-specific vocabulary. This dependence causes a trade-off between reliability and robustness: Grade predictions are most reliable for the UA test sets, where prompt-specific vocabulary is helpful to spot correct answers, and deteriorates for out-of-domain test sets. On the other hand, the reliance on informative keywords makes the model more susceptible to vocabulary-based gaming strategies, and robustness increases for out-of-domain test sets. Adversarial training on shuffled correct training items (as the hardest attack category) might be useful here to enforce more use of word order information by the model; Ding et al. (2020) report that this strategy improves the robustness of a non-neural SAG model while hardly hurting overall reliability.

References

Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25:60–117.

Leon Camus and Anna Filighera. 2020. Investigating transformers for automatic short answer grading. In *Artificial Intelligence in Education AIED*, Lecture Notes in Computer Science, pages 43–48.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuning Ding, Brian Riordan, Andrea Horbach, Aoife Cahill, and Torsten Zesch. 2020. Don't take "nswvtnvakgxpm" for an answer –the surprising vulnerability of automatic content scoring systems to adversarial input. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 882–892, Barcelona, Spain (Online). International Committee on 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.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. 2021. Memorization vs. generalization : Quantifying data leakage in NLP performance evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1325–1335, Online. Association for Computational Linguistics.

Hadi Abdi Ghavidel, Amal Zouaq, and Michel C. Desmarais. 2020. Using bert and xlnet for the automatic short answer grading task. In *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU)*, pages 58–67.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations Proceedings*.

Michael Heilman and Nitin Madnani. 2013. ETS: Domain adaptation and stacking for short answer scoring. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 275–279, Atlanta, Georgia, USA. Association for Computational Linguistics.

René Kizilcec and Hansol Lee. 2021. Algorithmic fairness in education. In W. Holmes and K. Porayska-Pomsta, editors, *Ethics in Artificial Intelligence in Education*, page forthcoming. Taylor & Francis.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Margot Mieskes and Ulrike Padó. 2018. Work smart - reducing effort in short-answer grading. In *Proceedings of the 7th workshop on NLP for Computer Assisted Language Learning*, pages 57–68, Stockholm, Sweden. LiU Electronic Press.

Niels Ott, Ramon Ziai, Michael Hahn, and Detmar Meurers. 2013. CoMeT: Integrating different levels of linguistic modeling for meaning assessment. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 608–616, Atlanta, Georgia, USA. Association for Computational Linguistics.

Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 2017. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–168, Copenhagen, Denmark. Association for Computational Linguistics.

Swarnadeep Saha, Tejas I. Dhamecha, Smit Marvaniya, Renuka Sindhgatta, and Bikram Sengupta. 2018. Sentence level or token level features for automatic short answer grading?: Use both. In *Artificial Intelligence in Education*, pages 503–517.

Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075, San Diego, California. Association for Computational Linguistics.

Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. 2019. Improving short answer grading using transformer-based pre-training. In *Artificial Intelligence in Education AIED, Proceedings*, Lecture Notes in Computer Science, pages 469–481.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

A Training and Test Data

Table 8 shows the sizes of test, development and training sets for Beetle and SEB as well as the label distribution for 2-way annotation.

B Model Sizes and Hyperparameters

BERT$_{base}$ and XLNet$_{base}$ have 110M parameters (Devlin et al., 2019; Yang et al., 2019), RoBERTa$_{base}$ has 125M (Liu et al., 2019) parameters.

The models received a maximum of 256 tokens per input sentence. We used the Adam optimizer with an initial learning rate of 5e-5, and ϵ of 1e-8; batch size for training was 8.

We varied the number of training epochs (up to a) maximum of six) and the random seed for weight initialization (1, 42, or 100).

Training times on a single GPU core were short. For six training epochs, BERT and RoBERTa trained in six minutes on Beetle and ten minutes on SEB. XLNet took ten and twelve minutes, respectively.

C Best-Performing Model Parameters

Table 9 shows the optimal random seeds and number of training epochs for the BERT$_{MNLI}$ models on the training corpora, and the resulting individual weighted F Scores on the development sets.

	Seed	Epochs	F Score
Beetle	100	5	89.0
SEB	100	6	85.5

Table 9: Number of training epochs and random seed for weight initialization for the BERT$_{MNLI}$ models. Individual weighted F Scores on the development sets.

D Exp. 1: Reliability

Table 10 shows the weighted F_1 scores on the test set, where available.

	Beetle		SEB		
	UA	UQ	UA	UQ	UD
SemEval-13	–	–	–	–	–
Saha et al.	–	–	79.1	**74.8**	**71.9**
BERT$_{MNLI}$	**90.2**	**77.5**	**83.1**	73.5	71.5

Table 10: Weighted F_1 on the test sets for literature benchmarks and BERT$_{MNLI}$.

Table 11 shows Precision and Recall by label on all the test sets.

Table 12 gives the percentage of accurate labels assigned by the 2-way model to the test items when broken down according to the 5-way classification.

E Code and Data

The code and data used for this study can be downloaded at `https://osf.io/72bzt/?view_only=56900eb27e8e4f88b6e489398fc295db`.

You will find

- SEB and Beetle UA development data (answer IDs only, due to licensing restrictions on redistribution)
- Python code to re-format the corpora, train the models and analyze results
- Bash scripts with the original training calls
- Requirement lists to re-create the server configurations used for training

	Train (% i)	Dev UA (% i)	Test UA (% i)	Test UQ (% i)	Test UD (% i)
Beetle	3570 (61.3)	371 (62.3)	439 (59.9)	819 (58.0)	–
SEB	4491 (59.6)	478 (58.2)	540 (56.9)	733 (58.9)	4562 (58.0)

Table 8: Size of training and test sections for the SemEval-2013 corpora. Label distribution for 2-way annotation (% i: percentage of label incorrect). UA: Unseen Answer, UQ: Unseen Question, UD: Unseen Domain.

	Beetle		SEB		
	UA	UQ	UA	UQ	UD
c	85.5/90.3	81.6/61.9	81.9/75.5	66.3/71.8	67.9/62.5
i	93.3/89.7	76.5/89.9	82.5/87.3	79.1/74.5	74.2/78.4

Table 11: Precision/Recall for the correct (c) and incorrect (i) classes on the Unseen Answer (UA), Unseen Question (UQ) and Unseen Domain (UD) test sets.

	Beetle		SEB		
	UA	UQ	UA	UQ	UD
correct	90.3 (175)	62.0 (344)	75.5 (233)	71.8 (301)	62.5 (1917)
contradictory	91.9 (111)	93.0 (244)	81.3 (58)	67.2 (64)	73.1 (417)
irrelevant	94.1 (17)	100 (18)	94.0 (133)	78.8 (193)	86.4 (1222)
partially correct	84.8 (112)	82.0 (172)	82.3 (113)	72.6 (175)	70.2 (986)
non-domain	100 (23)	100 (40)	100 (3)	– (0)	100 (20)

Table 12: Percentage of correct labels (total number of instances) for each of the 5-way classes on the Unseen Answer (UA), Unseen Question (UQ) and Unseen Domain (UD) test sets.