Filighera, A., Tschesche, J., Steuer, T., Tregel, T., Wernet, L. (2022). Towards Generating Counterfactual Examples as Automatic Short Answer Feedback. In: Rodrigo, M.M., Matsuda, N., Cristea, A.I., Dimitrova, V. (eds) Artificial Intelligence in Education. AIED 2022. Lecture Notes in Computer Science, vol 13355. Springer, Cham. https://doi.org/10.1007/978-3-031-11644-5_17

Towards Generating Counterfactual Examples as Automatic Short Answer Feedback *

Anna Filighera[0000−0001−5519−9959], Joel Tschesche, Tim Steuer[0000−0002−3141−712X], Thomas Tregel[0000−0003−0715−3889], and Lisa Wernet[0000−0002−0870−7225]

Multimedia Communications Lab
https://www.kom.tu-darmstadt.de
Technical University of Darmstadt, Germany

Abstract. Receiving response-specific, individual improvement suggestions is one of the most helpful forms of feedback for students, especially for short answer questions. However, it is also expensive to construct manually. For this reason, we investigate to which extent counterfactual explanation methods can be used to generate feedback from short answer grading models automatically. Given an incorrect student response, counterfactual models suggest small modifications that would have led the response to being graded as correct. Successful modifications can then be displayed to the learner as improvement suggestions formulated in their own words. As not every response can be corrected with only minor modifications, we investigate the percentage of correctable answers in the automatic short answer grading datasets SciEntsBank, Beetle and SAF. In total, we compare three counterfactual explanation models and a paraphrasing approach. On all datasets, roughly a quarter of incorrect responses can be modified to be classified as correct by an automatic grading model without straying too far from the initial response. However, an expert reevaluation of the modified responses shows that nearly all of them remain incorrect, only fooling the grading model into thinking them correct. While one of the counterfactual generation approaches improved student responses at least partially, the results highlight the general weakness of neural networks to adversarial examples. Thus, we recommend further research with more reliable grading models, for example, by including external knowledge sources or training adversarially.

Keywords: Explainable AI · Short Answer Grading · Feedback.

1 Introduction

Feedback is essential for learning as it helps uncover misconceptions, knowledge gaps and avenues for improvement [25]. However, providing feedback is expensive for constructed response questions where each unique answer has to be considered carefully. Nevertheless, since constructed response questions are better

* This research is funded by the Bundesministerium für Bildung und Forschung in the project: Software Campus 2.0 (ZN 01—S17050), Microproject: DA-VBB.

Table 1. Example student answer, common feedback and generated counterfactual.

Question:	What happens to the volume of the sound if you pluck a rubber band harder?
Reference:	The volume increases. The sound is louder.
Response:	It vibrates more and it gets lower. → *Incorrect*
Counterfactual:	It vibrates more and it makes louder sound. → *Correct*

suited to measuring complex skills compared to multiple-choice questions [16], the compromise is often to provide only verification feedback and a reference solution. Generally, verifying responses is much faster than formulating individual improvement suggestions. An example of verification feedback including a reference solution can be found in Table 1. It stems from the SciEntsBank [4] short answer grading dataset.

However, it can be hard to deduce one's mistakes from comparing with a reference solution. Depending on the reference's level of detail and exhaustiveness, a learner's response may not be covered by the solution or key differences may be drowned out by too many details. As there are often multiple correct solutions to short answer questions, learners may also have difficulties comprehending the particular solution provided by the teacher, especially when the teacher uses different terminology [31]. Thus, improvement suggestions in each learner's own words would likely be more helpful for learners [25].

Thus, this work proposes automatically generating counterfactual explanations as response-specific improvement suggestions. Inspired by human counterfactual reasoning [1], counterfactual explanation techniques essentially answer the question "What if the model's input would have looked like this instead?". The goal is to find small changes to the input features that would have changed the initially predicted output to the desired outcome [30]. For instance, given a learner response classified as incorrect by an automatic short answer grading (ASAG) model, what small changes to the learner's response would have led to the answer being predicted as correct? An example can be seen in Table 1.

However, not every learner's response lends itself to counterfactual feedback. Some answers may be so far from correct, such as "I don't know", that only massive changes would flip the predicted label to "correct". Other responses may be close to unreliable decision boundaries and lead to adversarial examples [6] that are only predicted as correct but do not actually improve the response. For this reason, this work addresses the following research question:

***RQ**: To which extent can we generate automatic feedback with counterfactual explanations?*

To this end, we make the following contributions:

– We show that counterfactual generation methods can modify student answers to be automatically graded as correct by comparing three counterfactual

generation models and a paraphrasing model on benchmark automatic short answer grading datasets (Section 3.4).
– Having an expert reevaluate a subset of the modified responses shows that almost all generated counterfactuals are adversarial examples instead of genuine improvements (Section 3.5).

2 Generating Counterfactual Feedback

The main idea of this work is to generate counterfactuals of incorrect student responses and explore their use as feedback. We develop and apply four approaches for this purpose. First are two approaches we developed based on Minimal Contrastive Editing (MiCE) [23]. They aim to iteratively replace the most impactful tokens in a response until it is graded as correct. Next is Polyjuice [32], a framework trained to perform pre-specified modifications, such as negating or shuffling entities in a sentence. Lastly, we develop an approach based on paraphrasing that generates novel responses instead of replacing parts of the original answer.

2.1 Contrastive Infilling

The main idea behind contrastive infilling approaches is finding the input parts detrimental to predicting the target class based on the model's gradients and replacing them with an editor model. MiCE [23] does this in two steps. In the first step, an editor model is trained to reproduce original data inputs. For this purpose, the most impactful tokens for the predicting model are masked so that the editor can learn to fill in critical sections of a response. The editor also receives the input's label to learn to produce responses of a specific class. In the second step, the editor iteratively fills in masked responses to find minimal modifications that cause the predictor to output a target label [23].

Inspired by MiCE, we implement two infilling models, one utilizing target labels and one without labels. The main idea behind cutting the labels used in MiCE is to simplify the task by only correcting wrong responses. Adding the target label does not carry any information in that case; it will always be the class "*correct*". However, one loses the ability to produce partially correct counterfactuals. Cutting the label requires a modification to the editor training proposed. While the label model is trained to reproduce all student answers by infilling masked parts of the student answers similar to MiCE, the other model is only trained to reproduce correct student responses. For both models, we randomly mask 20-55% of the student answer, and both models receive the reference answer in addition to the masked response. The label model is additionally conditioned on the target label in the following format: "label: *target label.* input: *masked student answer* </s> *reference answer*". Since the other model is only trained on correct responses and does not need a label, instead the input is formatted as follows: "input: *masked student answer* </s> *reference answer*".

In the second step, we use the previously fine-tuned models to modify incorrect student answers and perform up to four modification rounds. First, consecutive spans of tokens in the original student answer are masked based on

importance scores provided by the gradient attribution method Integrated Gradients [27]. We create four masked versions in each round with 15, 30, 45, and 60% of the tokens masked. We generate seven candidates for each masked student answer using a combination of top-k=30 and top-p=0.95 sampling. At the end of each modification round, the candidates are graded using an ASAG model, keeping only the candidate with the highest target class probability. The modification process is terminated when the candidate's target class probability exceeds the classification threshold or the maximum number of rounds is reached.

2.2 Polyjuice

In contrast to the previous approach, Polyjuice [32] aims to control the modification process through control codes. Instead of masking the tokens with the highest impact and generating arbitrary replacements, Polyjuice uses a predefined set of possible modifications, such as *negating* the meaning of the input or *shuffling* key phrases or entities around. The type of modification also controls where modifications can be made in the input so that the generated counterfactual should be fluent. Since the modification process is more constrained and, thus, may not be applicable to all student answers, we expect this method to yield less counterfactuals overall compared to the other approaches. However, any counterfactuals found should be more natural. We utilize Wu et al.'s [32] implementation[1] of Polyjuice to generate counterfactuals for incorrect student responses allowing all predefined modification codes: *negation, quantifier, shuffle, lexical, resemantic, insert, delete* and *restructure*.

2.3 Paraphrasing

Finally, we trained a T5 [22] model to paraphrase correct responses. In contrast to the counterfactual methods described above, this model does not fill in masked parts of the student response but generates a novel response instead. The main idea behind this approach was to explore whether a model trained to generate various correct responses to a question could also correct incorrect student answers. For this purpose, we treat correct student responses and reference answers as paraphrases of each other. While this is likely not accurate in practice as reference answers tend to be more comprehensive than student answers, the idea is for the model to learn the characteristics of correct answers. During training, it receives either the student or reference answer and generates the respective counterpart. After training, it gets incorrect student responses as input instead.

3 Experiments

The goal of our experiments is to determine to which extent feedback can be generated with counterfactual explanation methods. For this purpose, we first

[1] `https://github.com/tongshuangwu/polyjuice`

introduce the datasets and the ASAG model whose grading predictions will be explained by the counterfactual approaches. Then we introduce the metrics used to compare the approaches automatically, followed by insights gained from having a domain expert manually reevaluate the generated counterfactuals.

3.1 Datasets

We select three diverse ASAG datasets for our experiments: SciEntsBank [4], Beetle [4] and the English half of the Short Answer Feedback dataset (SAF) [7]. All three datasets offer a 3-way classification task with *correct* and *incorrect* responses. While the third class for SAF is *partially_correct*, the other datasets include *contradictory* as final class. The datasets offer multiple test sets, aimed at different grading scenarios. The unseen answer test split measures how well models perform on new answers to questions they were trained for, while the unseen questions split contains completely novel questions. Since SciEntsBank, in contrast to the others, contains multiple science domains, it also includes an unseen domain test split. Beetle, on the other hand, only contains basic electrical engineering questions and SAF is a computer science dataset in the *communication network* field. While SAF and Beetle consist of undergraduate responses, SciEntsBank's responses stem from American students in the grades 3 to 6. In contrast to the other datasets, Beetle includes multiple reference answers per question. In our experiments, we consider all reference answers.

3.2 Automatic Short Answer Grading Models

For each dataset, we train a BERT model that receives a student and reference answer as input and predicts the response's correctness. These three models form the predictors for the counterfactual search and, thus, should be as reliable as possible. For this reason, we follow the fine-tuning procedure used by Sung et al. [28] and achieve the predictive performance depicted in Table 2.

Table 2. Accuracy (Acc), macro-averaged F1 (M-F1) and weighted F1 (W-F1) of the automatic short answer grading models in percent.

Dataset	Unseen Answers			Unseen Questions			Unseen Domains		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1
SAF	77.1	75.5	77.1	52.9	57.5	52.9	-	-	-
Beetle	71.4	69.7	71.4	54.8	54.8	56.6	-	-	-
SciEnts.	72.9	70.9	72.9	59.7	50.9	59.7	61.5	54.6	61.5

3.3 Evaluation Measures & Experimental Setup

This paper focuses on two dimensions of counterfactuals that influence feedback quality: **validity** and **proximity**. Counterfactuals are often considered valid

when they lead to the desired prediction [30]. Thus, validity is usually measured by calculating the percentage of counterfactuals that flipped the predicted label to the desired outcome irrespective of the class predicted priorly. While that works well for tasks where the predictors achieve nearly perfect accuracy, it would overestimate the generators' performance in our case, as the ASAG model already misclassifies some of the incorrect student responses as *correct* without any modification. For this reason, we exclude all answers already predicted as *correct* from the evaluation. Furthermore, we hypothesize that counterfactual feedback will work better for student responses that are closer to being correct in the first place, such as *partially_correct* responses in contrast to *incorrect* ones. Therefore, we calculate the **flip rate** for each class separately.

Additionally, generated responses should be as close to the original student answer as possible to ensure that only required changes are made and the response follows the learner's wording beyond that. Following related work [23], we also use the word-level Levenshtein **distance** to measure the counterfactual's proximity to the original answer. It provides the minimum number of deletions, insertions and substitutions needed to equalize two strings. The count is then divided by the number of words in the original response to normalize it. As long as the generated response is not longer than the original response, it can be seen as the percentage of words modified.

All models introduced in Section 2 are trained on two Nvidia GX 2080 Ti cards with 11GB of RAM using gradient accumulation and mixed-precision floating-point numbers. The exact hyperparameters used for each approach can be found in our implementation.[2]

3.4 Comparison Results

Table 3 compares the counterfactuals generated by the Polyjuice, paraphrasing and contrastive infilling approaches introduced in Section 2 on the SAF dataset. It can be seen that the paraphrasing model succeeds in flipping the most labels to *correct* with flip rates between 50% and 100%. However, it also generates counterfactuals that differ vastly from the original student answer with an average distance of 2.22 across test splits and classes. Polyjuice is the opposite, generating counterfactuals that are very close to the original with an average distance of 0.02 but seldom flip the label to *correct*. The contrastive infilling methods seem to be more balanced, with an average flip rate of 24.2% without labels (infill) and 21.9% with labels (label infill) and average distances of 0.15 and 0.13, respectively. They also show the expected behaviour of flipping more partially correct responses than completely incorrect ones. While the paraphrasing model actually generates more flips on incorrect student answers compared to partially correct ones, they seem to be even more distant from the original responses.

Table 4 and Table 5 show the same comparison on the Beetle and SciEntsbank datasets. The infilling approaches perform slightly better on Sci-

[2] `https://github.com/joeltsch/CASAF-AIED2022`

Table 3. Flip rate (FR) and average distance (Dist) for counterfactuals generated on SAF's partially correct (Partial) and incorrect responses. Sample sizes are in brackets.

Approach	Unseen Answers				Unseen Questions			
	Partial (52)		Incorrect (9)		Partial (31)		Incorrect (8)	
	FR	Dist	FR	Dist	FR	Dist	FR	Dist
Paraphrase	**50.0**	1.72	**77.8**	3.89	**96.8**	1.60	**100**	1.66
Infill	25.0	0.19	11.1	0.11	35.5	0.12	25.0	0.19
Label Infill	19.2	0.14	11.1	0.10	32.3	0.12	25.0	0.16
Polyjuice	0.0	**0.01**	11.1	**0.01**	3.2	**0.03**	0.0	**0.01**

Entsbank compared to SAF, flipping on average 28.2% of the predictions without using labels and 28.8% utilizing labels, with a comparable average distance of 0.15 for both approaches. On Beetle, the infilling approaches flip considerably more predictions on average - at the cost of the much higher average edit distances. The labelless approach has an average flip rate of 55.9% and an average distance of 2.37. The approach with labels flips 41.0% of the predictions on average with an edit distance of 0.37. The paraphrasing model shows a similar behaviour of high flip rates and high edit distances on all datasets, with distances between 8 and 14 on Beetle. Additionally, Polyjuice produces few counterfactuals on all datasets but has higher average edit distances on Beetle with 0.14 and SciEntsbank with 0.12 compared to SAF.

Table 4. Flip rate (FR) and average distance (Dist) for counterfactuals generated on Beetle's contradictory (Contra) and incorrect responses. Sample sizes are in brackets.

Approach	Unseen Answers				Unseen Questions			
	Contra (453)		Incorrect (480)		Contra (740)		Incorrect (830)	
	FR	Dist	FR	Dist	FR	Dist	FR	Dist
Paraphrase	**74.2**	8.56	**78.3**	10.63	**76.9**	8.04	**74.7**	13.27
Infill	60.9	2.77	63.3	2.18	46.5	2.38	52.8	2.14
Label Infill	44.8	0.42	41.9	0.39	39.1	0.34	38.0	0.33
Polyjuice	1.8	**0.11**	2.1	**0.14**	1.8	**0.12**	3.3	**0.17**

3.5 Expert Regrading

While the flip rate indicates how many modifications lead to successful counterfactuals, it only considers the predictor's judgement and not whether the predictor was fooled into an incorrect prediction. For this reason, we asked one of the communication network experts involved in the original data annotation to reevaluate the generated counterfactuals for the SAF dataset. We selected SAF because it is the only dataset that includes elaborated feedback explaining why the response was graded as incorrect. This dramatically simplifies the

Table 5. Flip rate (FR) and average distance (Dist) for counterfactuals generated on SciEntsBank's contradictory (Contra) and incorrect responses. Sample sizes are in brackets.

Approach	Unseen Answers				Unseen Questions			
	Contra (48)		Incorrect (202)		Contra (35)		Incorrect (238)	
	FR	Dist	FR	Dist	FR	Dist	FR	Dist
Paraphrase	**72.9**	1.54	**74.8**	1.82	**65.7**	1.69	**68.5**	1.65
Infill	31.2	0.16	33.2	0.17	17.1	0.12	31.1	0.15
Label Infill	29.2	0.15	31.7	0.18	20.0	**0.11**	34.5	0.17
Polyjuice	2.1	**0.14**	1.0	**0.12**	5.7	0.12	2.5	**0.11**

reevaluation since the expert only has to determine whether the modification corrects the mistake instead of regrading the responses from scratch.

The expert evaluated all counterfactuals the ASAG model predicted accurately prior to modification and as *correct* after modification. There were 59 examples for the paraphrasing model, 1 for Polyjuice, 21 for the label infilling approach and 25 for infilling without labels. In total, 106 examples were regraded.

Nearly all generated samples (N=103) were adversarial examples and not genuine corrections of the response. Of the 3 correct examples, 2 stem from the paraphrasing model simply generating the reference answer to the question instead of modifying the student answer. In general, the paraphrases were often vastly different from the student responses, which matches the observations from Section 3.4. Sometimes the paraphrasing model would also mix reference solutions to multiple questions, which may be one of the reasons why it is so successful at fooling the predictor. Humorously, some of the content added to the response by the paraphrasing model was utterly absurd, such as "*... 56.648 * 64 bit/sec = 128 bit processing tables = 276 bit data transfer tables + 3 * 1.31 seconds to reach the destination system ...*".

The infilling models also mostly produced adversarial examples with senseless modifications. For example, "*... the issue with this* ***case*** *is ...*" was replaced with "*... the issue with this* ***narcotic*** *is ...*" which does not make any sense in the communication network domain. Sometimes the model would also replace words with special tokens, such as "<extra_id_34>". However, not all modifications made by the infilling models were adversarial. Some modifications truly improved student responses partially, even if they were still incorrect overall. For example, "*extension headers are* ***the way to put additional information in the packet****...*" was correctly replaced with "*extension headers are* ***used to extend the fixed ipv6 header with additional options****...*".

4 Related Work

In recent years, the need for understandable machine learning models has given rise to diverse approaches aiming to explain the inner workings of neural networks. Such explanations can be used to increase the transparency and trustworthiness of automatic predictions [24]. The branch of explainable AI most relevant

to our work is based on counterfactual reasoning, revolving around how an input's features would have to differ as to change a model's prediction. As there are countless counterfactual explanation techniques and we already describe the most relevant ones in Section 2, we recommend one of the excellent surveys summarizing the state-of-the-art [2,12,26,30] for further reading and focus on related work on generating elaborated feedback.

4.1 Elaborated Feedback Generation

Especially in the intelligent tutoring community, generating elaborated feedback has been a hot topic for many years [3,8,14,20,29]. Older approaches mainly focused on hand-crafting domain models and manually tailoring feedback systems to specific tasks [5,10,17]. More recently, research is exploring more flexible feedback systems for structured answer formats, such as programming exercises [13], proofs [18], or multiple-choice questions [15,33]. Here the structure of the response is exploited to automatically identify the kind of mistakes made, for example, by using a compiler. The most similar to our work here is an approach proposed by Olney [21]. They automatically generate elaborated feedback for cloze-style questions by first generating a question about the relationship between the correct cloze solution and the incorrect term provided by the student. The answer to the synthetic question provided by an automatic question answering system is then included as elaborated feedback.

For unstructured question formats, like essays and short answer questions, flexible feedback systems mainly focus on a response's language and style [11], identifying justifications [19] or discovering which topics are covered in an essay [9]. Only recently, a deep learning system to automatically generate elaborated feedback for short answer questions was introduced [7]. However, it relies on feedback data which is still unavailable for most domains.

5 Conclusion & Future Work

In summary, this work compared four approaches to providing counterfactual feedback to short answer questions. Three out of the four methods successfully generated counterfactuals for at least a fifth of the incorrect responses in three diverse short answer grading datasets. Around a quarter of incorrect responses could be modified until the automatic grading model judged them correct without diverging too far from the initial student response. However, a domain expert still deemed nearly all modified responses incorrect. This result illustrates the need for human evaluation of generated counterfactuals. In related work, counterfactuals are mainly evaluated using flip rates and automatic proximity measures [12]. However, considering the high rate of adversarial examples observed in this study, automatic metrics are not sufficient to capture the true usefulness of generated counterfactuals. Thus, future work should include human judgements.

Regarding the research question posed in this work, we conclude that counterfactual explanations are unsuitable as feedback at the current state. However,

they can be even more helpful for teachers aiming to employ automatic grading models in practice. The generated counterfactuals could be used to identify critical weaknesses of the grading model. For example, the humorous example from Section 3.5 may indicate an inability to evaluate mathematical expressions correctly. Generated counterfactuals could also be added to the training data to facilitate adversarial training of more robust grading models. More robust grading models may, in turn, produce better counterfactuals. Since we observed genuine partial improvements in student responses in our experiments, incentivizing the counterfactual model to search beyond adversarial modifications seems like a promising avenue of future research.

Finally, the counterfactual generation methods themselves could be improved. We showed that counterfactual generators vary greatly in the number of label flips they entice and how dissimilar the modifications are to the original. Thus, other approaches may yield more or better counterfactuals. Especially approaches utilizing external knowledge sources and other neuro-symbolic methods may be beneficial for the short answer feedback task. The additional knowledge could inform the search for sensible modifications or help identify which parts of a student's response are incorrect and, thus, should be replaced.

References

1. Buchsbaum, D., Bridgers, S., Skolnick Weisberg, D., Gopnik, A.: The power of possibility: Causal learning, counterfactual reasoning, and pretend play. Philosophical Transactions of the Royal Society B: Biological Sciences **367**(1599), 2202–2212 (2012). https://doi.org/10.1098/rstb.2012.0122
2. Chou, Y.L., Moreira, C., Bruza, P., Ouyang, C., Jorge, J.: Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications. Information Fusion **81**, 59–83 (2022). https://doi.org/10.1016/j.inffus.2021.11.003
3. Deeva, G., Bogdanova, D., Serral, E., Snoeck, M., De Weerdt, J.: A review of automated feedback systems for learners: Classification framework, challenges and opportunities. Computers & Education **162** (2021). https://doi.org/10.1016/j.compedu.2020.104094
4. Dzikovska, M., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., Dang, H.T.: 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). pp. 263–274. Association for Computational Linguistics, Atlanta, Georgia, USA (Jun 2013), https://aclanthology.org/S13-2045
5. Dzikovska, M., Steinhauser, N., Farrow, E., Moore, J., Campbell, G.: Beetle II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. International Journal of Artificial Intelligence in Education **24**(3), 284–332 (2014). https://doi.org/10.1007/s40593-014-0017-9
6. Filighera, A., Ochs, S., Steuer, T., Tregel, T.: Cheating automatic short answer grading: On the adversarial usage of adjectives and adverbs (2022). https://doi.org/10.48550/ARXIV.2201.08318

7. Filighera, A., Parihar, S., Ochs, S., Steuer, T., Meuser, T.: Your answer is incorrect... would you like to know why? Introducing a bilingual short answer feedback dataset. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (in press)
8. Hasan, M.A., Noor, N.F.M., Rahman, S.S.B.A., Rahman, M.M.: The transition from intelligent to affective tutoring system: A review and open issues. IEEE Access **8**, 204612–204638 (2020). https://doi.org/10.1109/ACCESS.2020.3036990
9. Hellman, S., Murray, W., Wiemerslage, A., Rosenstein, M., Foltz, P., Becker, L., Derr, M.: Multiple instance learning for content feedback localization without annotation. In: Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications. pp. 30–40. Association for Computational Linguistics, Seattle, WA, USA → Online (Jul 2020). https://doi.org/10.18653/v1/2020.bea-1.3
10. Jordan, S., Mitchell, T.: e-assessment for learning? The potential of short-answer free-text questions with tailored feedback. British Journal of Educational Technology **40**(2), 371–385 (2009). https://doi.org/10.1111/j.1467-8535.2008.00928.x
11. Ke, Z., Ng, V.: Automated essay scoring: A survey of the state of the art. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 6300–6308. International Joint Conferences on Artificial Intelligence Organization (7 2019). https://doi.org/10.24963/ijcai.2019/879
12. Keane, M.T., Kenny, E.M., Delaney, E., Smyth, B.: If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 4466–4474. International Joint Conferences on Artificial Intelligence Organization (Aug 2021). https://doi.org/10.24963/ijcai.2021/609, survey Track
13. Keuning, H., Jeuring, J., Heeren, B.: A systematic literature review of automated feedback generation for programming exercises. ACM Transactions on Computing Education (TOCE) **19**(1), 1–43 (2018). https://doi.org/10.1145/3231711
14. Kulik, J.A., Fletcher, J.: Effectiveness of intelligent tutoring systems: a meta-analytic review. Review of educational research **86**(1), 42–78 (2016). https://doi.org/10.3102/0034654315581420
15. Ling, W., Yogatama, D., Dyer, C., Blunsom, P.: Program induction by rationale generation: Learning to solve and explain algebraic word problems. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 158–167. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1015
16. Livingston, S.A.: Constructed-response test questions: Why we use them; how we score them. R&D Connections **11** (Sep 2009)
17. Lu, X., Di Eugenio, B., Ohlsson, S., Fossati, D.: Simple but effective feedback generation to tutor abstract problem solving. In: Proceedings of the Fifth International Natural Language Generation Conference. pp. 104–112. Association for Computational Linguistics, Salt Fork, Ohio, USA (Jun 2008)
18. Makatchev, M., Jordan, P.W., VanLehn, K.: Abductive theorem proving for analyzing student explanations to guide feedback in intelligent tutoring systems. Journal of Automated Reasoning **32**(3), 187–226 (2004)
19. Mizumoto, T., Ouchi, H., Isobe, Y., Reisert, P., Nagata, R., Sekine, S., Inui, K.: Analytic score prediction and justification identification in automated short answer scoring. In: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications. pp. 316–325. Association for Computational Linguistics, Florence, Italy (Aug 2019). https://doi.org/10.18653/v1/W19-4433

20. Mousavinasab, E., Zarifsanaiey, N., Kalhori, S.R.N., Rakhshan, M., Keikha, L., Saeedi, M.G.: Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. Interactive Learning Environments **29**(1), 142–163 (2021). https://doi.org/10.1080/10494820.2018.1558257
21. Olney, A.M.: Generating response-specific elaborated feedback using long-form neural question answering. In: Proceedings of the Eighth ACM Conference on Learning @ Scale. p. 27–36. L@S '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3430895.3460131
22. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020)
23. Ross, A., Marasović, A., Peters, M.: Explaining NLP models via minimal contrastive editing (MiCE). In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 3840–3852. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.findings-acl.336
24. Shin, D.: The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai. International Journal of Human-Computer Studies **146**, 102551 (2021). https://doi.org/10.1016/j.ijhcs.2020.102551
25. Shute, V.J.: Focus on formative feedback. Review of Educational Research **78**(1), 153–189 (2008). https://doi.org/10.3102/0034654307313795
26. Stepin, I., Alonso, J.M., Catala, A., Pereira-Fariña, M.: A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. IEEE Access **9**, 11974–12001 (2021)
27. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International Conference on Machine Learning. pp. 3319–3328. PMLR (2017)
28. Sung, C., Dhamecha, T.I., Mukhi, N.: Improving short answer grading using transformer-based pre-training. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) Artificial Intelligence in Education. pp. 469–481. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-23204-7_39
29. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist **46**(4), 197–221 (2011). https://doi.org/10.1080/00461520.2011.611369
30. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: A review. arXiv preprint arXiv:2010.10596 (2020), `https://arxiv.org/abs/2010.10596`
31. Winstone, N.E., Nash, R.A., Parker, M., Rowntree, J.: Supporting learners' agentic engagement with feedback: A systematic review and a taxonomy of recipience processes. Educational Psychologist **52**(1), 17–37 (2017)
32. Wu, T., Ribeiro, M.T., Heer, J., Weld, D.: Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 6707–6723. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.523
33. Xie, Z., Thiem, S., Martin, J., Wainwright, E., Marmorstein, S., Jansen, P.: WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 5456–5473. European Language Resources Association, Marseille, France (May 2020), `https://www.aclweb.org/anthology/2020.lrec-1.671`