Your Answer is Incorrect... Would you like to know why? Introducing a Bilingual Short Answer Feedback Dataset

Anonymous ACL submission

Abstract

Handing in a paper or exercise and merely receiving a "bad" or "incorrect" as feedback is not very helpful when the goal is to improve. Unfortunately, this is currently the kind of feedback given by many Automatic Short Answer Grading (ASAG) systems. One of the reasons for this is a lack of content-focused elaborated feedback datasets. To encourage research on explainable and understandable feedback systems, we present the Short Answer Feedback dataset (SAF). Similar to other ASAG datasets, SAF contains learner responses and reference answers to German and English questions. However, instead of only assigning a label or score to the learners' answers, SAF also contains elaborated feedback explaining the given score. Thus, SAF enables supervised training of models that grade answers and explain where and why mistakes were made. This paper discusses the need for enhanced feedback models in real-world pedagogical scenarios, describes the dataset annotation process, gives a comprehensive analysis of SAF, and demonstrates how SAF challenges T5 Transformer models.[1]

1 Introduction

Assessment and feedback are essential to high-quality education (Shute, 2008). They allow learners and teachers to discover misconceptions, gaps in knowledge, and improvement opportunities. However, manually assessing learners' knowledge and providing helpful feedback is time-consuming and requires pedagogical as well as domain expertise. Here, automatic assessment can free up teachers' time to focus on tutoring learners or adequately preparing classroom activities. Moreover, it can be an alternative to peer-grading when course participant numbers increase beyond the financial feasibility of manual grading (Kay et al., 2013), making it particularly useful for freely accessible online courses.

Besides being cost- and time-efficient, automating assessment also offers unique teaching opportunities. As long as systems give individual, response-specific feedback, learners may retry or take additional assignments and receive instantaneous feedback as often as they need. Additionally, knowing that a system instead of one's teacher or professor will evaluate one's assignment can also reduce anxiety and help learners focus on their work instead of worrying about their reputation (Lipnevich and Smith, 2009). Therefore, it is unsurprising that automatic assessment has been an active research field over the past decades (Burrows et al., 2015; Ihantola et al., 2010; Ke and Ng, 2019; Xi, 2010). So far, significant progress has been made.

In particular, Transformer models are approaching human experts' performance on specific datasets in the Automatic Short Answer Grading (ASAG) field (Sung et al., 2019; Camus and Filighera, 2020). These models are trained to evaluate whether natural language responses fully answer open knowledge questions and typically output a score or label indicating the response's correctness. This kind of feedback is also called *verification* (Shute, 2008). An example can be seen in Table 1. However, merely providing a score or label for a learner's answer is generally not sufficient in real-world pedagogical scenarios. Firstly, learners must understand their feedback to use it effectively (Winstone et al., 2017). That may not be the case when learners only receive a score instead of a clear explanation of where and why they made mistakes. Secondly, the feedback's source needs to be trusted for learners to accept and engage with the given advice (Winstone et al., 2017). Especially assessments by automatic models may be questioned (Lipnevich and Smith, 2009; Filighera et al., 2020a,b). Providing a response-specific, detailed explanation may establish the necessary

[1] Our code, dataset and scoring rubrics will be publicly available at github.com/[anonymized]/ under an MIT license

Question:	What are the challenges of Mobile Routing compared to routing in fixed and wired networks? Please name and describe two challenges.
Answer:	1) Due to hardware constraints, some nodes may be out of the range of others. 2) Mobile routing requires more flexibility. The environment is very dynamic and the routing mechanism has to adapt to that.
Verification:	0.5 out of 1.0 points (Partially Correct)
Elaborated Feedback:	While the second challenge of needing to be able to adapt to a dynamically changing environment is correct, the first challenge stated is not a challenge specific to mobile routing. In a wired network, nodes typically don't have a direct connection to each other node as well.

Table 1: An example answer with annotated feedback contained in SAF.

transparency for learners and teachers to trust the system's predictions. This kind of explanation is also called *elaborated feedback* (Shute, 2008) and is shown in Table 1.

In the Intelligent Tutoring Systems community, the need for elaborated feedback is well-known (Deeva et al., 2021; Hasan et al., 2020). Several researchers have incorporated feedback modules in their systems (VanLehn, 2011; Kulik and Fletcher, 2016; Mousavinasab et al., 2021). However, these approaches are typically constrained to structured answer formats, such as programming exercises (Keuning et al., 2018), focus on the response's language and style instead of the content (Hellman et al., 2020), or are hand-tailored to specific tasks (Dzikovska et al., 2014; Lu et al., 2008). A lack of public, content-centered elaborated feedback datasets that enable supervised, expert-independent approaches may be one of the main reasons for this. To narrow this gap, we provide the Short Answer Feedback dataset (SAF), a German and English collection of learner answers and feedback.

In contrast to other ASAG datasets, the feedback includes a classification or rating of the answers and contains detailed explanations. This allows for automatic scoring and opens the new task of providing response-specific, elaborated feedback that explains the given score. The dataset contains 4,519 submissions, corresponding scores, and response-specific elaborated feedback. Additionally, we provide T5 (Raffel et al., 2020) and mT5 (Xue et al., 2021) baselines for future comparison.

2 Related Work

While elaborated feedback datasets on language learning (Caines et al., 2020; Pilan et al., 2020; Stasaski et al., 2020) appeared recently, they focus on linguistic mistakes, such as grammatical errors, instead of content. Our extensive literature review did not reveal datasets that included content-focused elaborated feedback on short answer responses. However, SAF's feedback can be viewed as a textual explanation of the assigned score. Therefore, comparable NLP datasets with textual explanations and publicly available ASAG datasets without explanations are discussed in the following sections.

2.1 Natural Language Explanation Datasets

In recent years, the need for understandable, interpretable NLP models has been widely discussed (Adadi and Berrada, 2018; Alishahi et al., 2019; Danilevsky et al., 2020; Das and Rad, 2020). One of the possible approaches to make models explainable is to train them or auxiliary models to directly generate explanations of their predictions (Liu et al., 2019; Narang et al., 2020). For this purpose, multiple researchers enhanced NLP datasets with textual explanations.

Camburu et al. (2018) extended the Stanford Natural Language Inference dataset (SNLI) (Bowman et al., 2015) using Amazon Mechanical Turk. The expanded dataset is called *e-SNLI* and contains textual, human-generated explanations for each of SNLI's entailment relation pairs. Rajani et al. (2019), also using Amazon Mechanical Turk, expanded COMMONSENSEQA (Talmor et al., 2019). The resulting *Common Sense Explanations (CoS-E)* dataset consists of commonsense reasoning questions with three possible answers and a textual explanation for every correct selection. Mostafazadeh et al. (2020) introduced

GLUCOSE, a crowdsourced collection of semi-structured causal explanations related to sentences in stories. However, the datasets above do not have a pedagogical focus. This is detrimental to researchers aiming to employ their systems in educational contexts, where explanations should conform to pedagogical guidelines, such as avoiding harm to the learner's self-esteem or motivation.

The closest to our research is the *WorldTree V2* dataset. Here, Xie et al. (2020) used graphs of expert-engineered natural language facts to explain correct answers to multiple-choice science questions. The resulting explanations are essentially lists of scientific and world knowledge facts needed to answer the question correctly. Similarly, Ling et al. (2017) provide textual explanations for the correct solutions to math problems. Their multiple-choice questions, answers, and explanations are obtained by crowdsourcing and standardized tests, such as GMAT. While both Ling et al. (2017)'s and Xie et al. (2020)'s work have an educational focus, they only explain the reference solution instead of mistakes made in incorrect or partially correct solutions.

2.2 Short Answer Grading Datasets

Some of the most well-known ASAG datasets stem from the SemEval 2013 challenge (Dzikovska et al., 2013). BEETLE contains 5,044 student answers to basic electricity questions labeled as *correct, partially_correct_incomplete, contradictory, irrelevant* or *non_domain*. SCIENTSBANK follows the same structure but also contains questions of various other domains, such as biology or geography. Basu et al. (2013) introduced *Powergrading*, a collection of 2,532 unique, crowdsourced answers to ten questions of a United States Citizenship Exam. Each was manually classified as *correct* or *incorrect*. In contrast to the previous datasets, answers in the *ASAP-SAS* [2] dataset are scored on a scale from 0 to 3. Additionally, this dataset is much larger with ∼2,200 responses per question, with 10 questions in total. All of the datasets above only include verification feedback.

Mizumoto et al. (2019) released a Japanese dataset containing 12,600 student responses equally distributed across 6 questions. The answers stem from a commercial achievement test for Japanese high school learners and are annotated with holistic scores and individual marks for manually defined scoring criteria. Additionally, each criterion links to the phrase in the student's answer expressing it. For example, for a criterion like "2 points if the response mentions *Western culture*", the phrase *Western culture* would be marked in the response, if present. This dataset enables elaborated feedback systems. However, the structured nature of criteria and matching answer spans complicates an automatic translation to English. Additionally, the marking scheme is limited in its expressiveness as it is hard to mark missing information in the answer.

Lastly, structured collections of smaller and non-public datasets can be found in surveys by Roy et al. (2015) and Burrows et al. (2015).

3 Short Answer Feedback dataset (SAF)

To remedy the lack of content-focused elaborated feedback datasets, we provide SAF, an English and German short answer dataset with explanations that serve as elaborated feedback. In total, the corpus contains 4,519 submissions, similar to the example in Table 1. There are 22 English short answer questions with reference answers covering a range of college-level *communication network* topics, such as extension headers in IPv6 or frame bursting. Additionally, the dataset contains 8 German short answer questions used in micro-job training on the appJobber [3] crowd-worker platform. The data was collected and annotated between April 2020 and June 2021. While individuals gave the German answers in the context of pre-job training, the English questions were answered in groups of up to three students in voluntary quizzes they could complete for extra points in the final exam. Each quiz consists of 3-4 questions regarding the same overarching topic, such as "Internet protocols". All answers are annotated with a score, label, and feedback as described in Table 2. The dataset can be used for classical automatic short answer grading and elaborated feedback generation.

3.1 Challenges and Requirements

We need reliable scoring and clear, detailed explanations to train understandable feedback models. Providing this is challenging for multiple reasons. Firstly, annotators need to have the necessary domain expertise and the pedagogical knowledge on how to provide understandable, well-received feedback. For instance, they should be aware of their

[2] https://www.kaggle.com/c/asap-sas/

[3] https://appjobber.de/

Field	Description
Score	A numerical value between 0 and 1 indicating the answer's correctness and completeness. Depending on the question, the range is discretized into steps, e.g. 0.125, so that the annotators do not have to make arbitrarily fine distinctions.
Response Feedback	Response-contingent elaborated Feedback. It explains why an answer is wrong or right without using formal error analysis (Shute, 2008). Hints or the correct answer may be used to explain mistakes.
Verification Feed.	An automatic labeling of the score. Includes the following labels: Incorrect (score=0), Correct (score=1), Partially Correct (all intermediate scores)

Table 2: SAF's annotation fields with descriptions.

feedback's emotional effect. At first glance, this may seem obvious, but it is easily overlooked in practice. An example of this became apparent during a pilot study we conducted to uncover pitfalls and train our annotators. Even though we provided guidelines on how to give feedback, questionable phrases like "This response fails to ..." were common as the annotators did not consider that the word "failing" may trigger negative associations and emotions in learners.

Secondly, a common ground truth must be established for each question with clearly defined boundaries because various sources may define concepts differently. For example, the network protocol TCP alone has at least five different variations, all with unique advantages and disadvantages, leading to multiple possible answers to TCP related questions (Chaudhary and Kumar, 2017). In our pilot study, this expressed itself with a low inter-annotator agreement (Krippendorff's Alpha of 0.36), making the need for detailed scoring rubrics clear. We discuss our approaches to these challenges in the following section.

3.2 Dataset Construction

To ensure the necessary domain expertise, we selected two graduate students [4] who had completed the *communication networks* course themselves and two experienced appJobber employees for the crowd-worker platform's answers. For pedagogical training, a researcher first drafted a **general annotation guideline**. It explains the annotation files' structure, the annotation goals, and provides general recommendations for the formulation of feedback and the calculation of scores. For example, it asserts that praise, comparisons with other learners, or emotionally charged words like "fail" should be avoided when writing feedback. Additionally, it points out common biases annotators should be aware of, such as confirmation bias. For instance, answers that contain keywords found in many correct responses may still contain mistakes and should, therefore, still be carefully inspected. The general annotation guidelines were submitted to a psychology doctoral student with prior work in the feedback field for additional advice. Then the annotators applied their knowledge in the pilot study and received further feedback from the researchers. Finally, the guideline was updated to reflect any additional discussion points.

As can be seen in Figure 1, the researcher drafted **grading rubrics** for each question. The rubric consists of the questions, reference answers with detailed grading information, and four example answers per question for illustration. As research suggests that a single author may not suffice to produce reliable and objective scoring rubrics (Carr, 2020), the draft is then discussed and refined with the annotators. The discussion also mitigates the challenge of defining a common ground truth, as multiple sources and opinions can coalesce into a single, exhaustive rubric. Before the discussion, the **answer annotation files** are available to the annotators. The files contain the reference and students' answers.

Subsequently, annotators individually evaluated answers using the scoring rubric and the general annotation guideline. All English answers were annotated twice, while only half of the German answers were annotated doubly due to the prohibitive cost of experienced employees. The first step of combining the independently annotated answer files into a cohesive gold standard involves discussing the disagreements with the annotators and researcher. The disagreements were resolved by either choos-

[4]The students' remuneration consisted of a paid research assistant position for one and partial credit towards a master's thesis and co-authorship of this paper for the other.

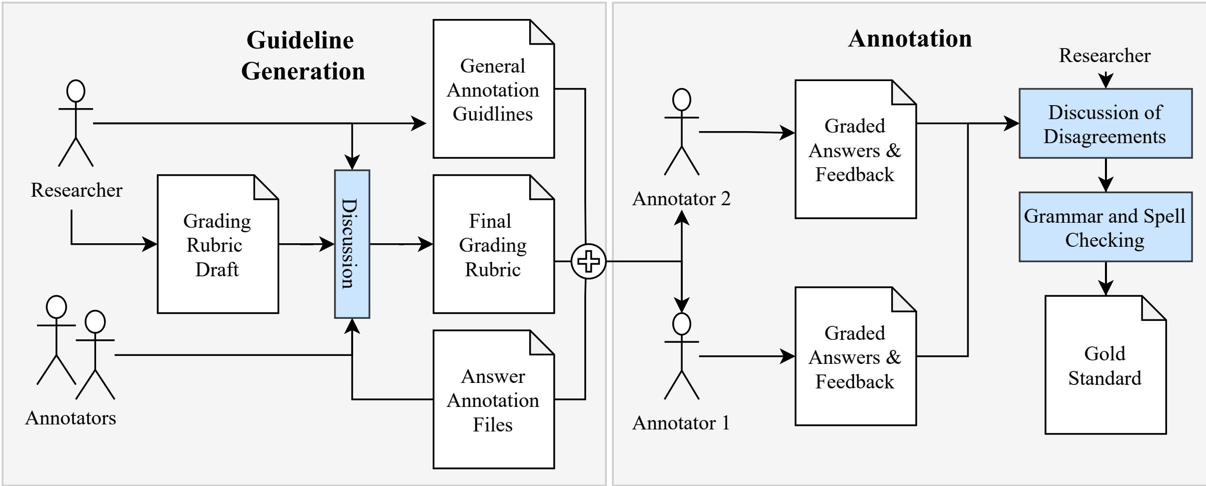


Figure 1: Schematic depiction of the annotation process.

ing one of the annotations, compromising, or fusing them if both had merit. For example, one annotator may notice a missing fact A while the second annotator may find a mistake in B's explanation. Finally, the English gold feedback was checked by Grammarly as well as an English native speaker. Grammar and spelling mistakes were corrected, and sentences were simplified when the same information could be expressed more concisely, for example, by using the possessive form. Learners' answers were not post-processed because models would frequently encounter grammar and spelling mistakes in the wild. Therefore, this is a challenge approaches should overcome.

3.3 Corpus Statistics

The annotation process resulted in a corpus with the following score and label distribution seen in Table 3. Similar to the SemEval dataset BEETLE (Dzikovska et al., 2013), we split the data into training (64% of DE / 70% of EN), unseen answers (11% / 12%) and unseen questions (25% / 18%) test sets. While the unseen answers test split contains new answers to the training's questions, the unseen questions split contains novel questions. This setup enables the investigation of models' ability to generalize to new questions without the need for priming with manually annotated answers first.

Figure 2 shows the length of questions, feedback, reference, and learner answers of the English training set in tokens. We used NLTK's word_tokenize[5] to obtain the tokens, so their count can be seen as the sum of words and punctuation symbols in the text. The learners' answers were between 0 and 589 tokens long (average=82.2, median=68). We did not filter empty submissions (unless all of the group's submissions were empty) from the dataset as models will encounter this in real-world applications. Since the reference answer and learner answer are typically combined as input for ASAG models, this dataset's sensible input sequence length may prove to be computationally expensive for large Transformer models. Feedback tends to be shorter with 5-120 tokens (average=22.4, median=15). The distribution looks similar for the German half of the dataset only that the answers and feedback tend to be slightly shorter. Details can be found in Appendix A.

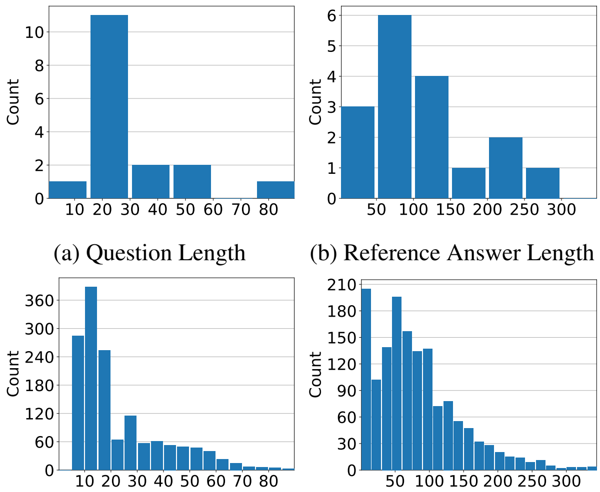
Score	Train		UA		UQ	
	DE	EN	DE	EN	DE	EN
0.0	216	234	47	42	49	87
(0.0, 0.3]	103	43	22	11	37	4
(0.3, 0.6]	385	143	68	19	131	24
(0.6, 1.0)	126	227	31	44	107	90
1.0	704	829	103	136	287	179
$\sum$	1534	1476	271	252	602	384

Table 3: Distribution of gold standard scores. UA stands for Unseen Answers, and UQ denotes Unseen Questions. DE encompasses the German and EN the English half of the dataset.

3.4 Annotation Quality

To estimate our annotations' reliability, we rely on inter-annotator agreement measures. As the scores are interval scaled between 0 and 1, we report the percentage agreement and Krippendorff's Alpha.

[5] https://www.nltk.org/api/nltk.tokenize.html



(a) Question Length (b) Reference Answer Length

(c) Feedback Length (d) Learner Answer Length

Figure 2: Histograms showing the distribution of text lengths (in tokens) in the English training set. The tail ends of c) and d) are trimmed, leaving 3 and 8 samples unrepresented.

The annotators agreed on 89.46% of the cases on the English data, and α is 0.91 (N=2,112). On the German questions, the annotators agreed in 81.38% of the cases, and α is 0.78 (N=1,200). The high agreement on the overall dataset illustrates the effectiveness of our annotation process, especially when compared to the initially low agreement of α=0.36 achieved in our pilot study.

We can assume the validity of our German data to be high, since our experienced annotators were also responsible for accepting or rejecting job results later on. Hence, their judgements should be consistent with the desired learning outcome. To estimate the validity of our English data, we assume that the end-of-term exam is a valid evaluation of students' knowledge. Of course, this is most likely not accurate in practice since the exam was not formally validated and only provides a snapshot of students' performance in a single 120-minute time frame. However, most of the question pool and exam structure have been employed and refined over multiple years. For this reason, we deem it a sufficient approximation. Nevertheless, the following results should be viewed as an indication of validity rather than a fact. The Spearman's rank correlation between the points achieved in the exam and the quizzes is 0.438 ($p < 0.0001$) with a sample size of 186. This is a moderate positive correlation between the exam and quiz results (Dancey and Reidy, 2007) and indicates that they may measure the same or a similar construct. In contrast to the quizzes, exams were not taken in groups, partly explaining the variance.

3.5 Ethical Concerns

It is our responsibility to be transparent in our data collection process and protect the privacy of our learners. Our first step in this regard was to inform our learners of the data collection process. We posted to the college course's online learning platform and the description of the German job training. Both channels usually carry vital information for the learners. In our post, we

- detailed how we would use the learners' answers to research and develop automatic assessment models.
- asked learners to refrain from including personal information in their answers, such as names or addresses. This was also checked during the annotation process.
- gave them contact information if they wanted their answers to be excluded from the data collection. We also clarified that this would not negatively impact them or their grades/access to jobs. None of the learners contacted us.
- clarified that we would only release anonymized data in our publications.

We anonymized German answers by stripping identifying information and randomizing the order. To anonymize the English learners' answers, we randomly assigned each group an ID. The group-to-ID mapping was done locally on one computer and was deleted after the dataset construction. Keeping a consistent group ID allows us to identify responses with *quizID.questionID.groupID* and, thus, publish a dataset where the other answers of a group can be incorporated to refine an assessment model. For example, responses QuizA.1.3 and QuizB.2.3 are written by the group assigned the ID 3. This characteristic is beneficial as it allows for training models that provide personalized feedback, considering the current answer and answers to related questions. Patterns of mistakes spanning multiple questions may be discovered in this setting. For example, if a group answered all performance evaluation questions incorrectly, they may not understand the probability theory underlying the questions. However, note that SAF's annotators only considered the current answer when constructing their feedback.

4 Experiments

The goals of our experiments are threefold. Firstly, we want to provide baselines for the dataset. For this reason, it makes sense to report a wide range of metrics future work may want to utilize. Secondly, we hypothesize that including the question in the model's input would increase performance. Typically, only the student and reference answers are compared in ASAG (Lv et al., 2021) even though the question may contain additional important information. To investigate the question's effect on performance, we run each experiment in two settings: with a student and reference answer pair as model input or with a question, student, and reference answer triplet.

Finally, we want to explore the synergy between the ASAG scoring and classification tasks and feedback generation. We believe that grading and feedback should be trained jointly since the feedback should always match the assigned grade, and both tasks benefit from extracting the same information from the answers. For example, a span of tokens negatively impacting the grade should also affect the feedback accordingly. Our experiments investigate the hypothesis that feedback generation benefits more from being paired with the more informative ASAG scoring task (0-1) than the verification feedback label classification (correct vs. incorrect vs. partially correct).

4.1 Experimental Settings

As baselines, we utilize HuggingFace's implementation of the T5-base and mT5-base models (Wolf et al., 2020). They are fine-tuned to predict the response's score or label and jointly explain it. For computational reasons, the input sequence is trimmed to 512 tokens when using T5 and 256 tokens when using mT5. When the sequence is longer, a part of the reference answer is truncated. While the complete learner answer is always relevant for grading, the reference answer may discuss details or additional aspects irrelevant to the particular response.

The output is limited to 128 tokens and has the following format: "*label/score* feedback: *feedback*". We also enforce a minimum output sequence length of 11 tokens since models tended to refrain from generating feedback otherwise. In all experiments, 10% of the training data was split-off for manual hyperparameter tuning and model selection. All models use gradient accumulation and an Adafactor (Shazeer and Stern, 2018) optimizer with learning rate warm-up. We trained models for maximally 64 epochs utilizing early stopping with a patience of 10 and selected the best performing model/epoch using the following metric m, where f is the macro-averaged F1 score during classification and $1 - MSE$ during scoring.

$$m = \frac{BLEU + ROU. + MET.}{3} * f \qquad (1)$$

We average SacreBLEU,[6] ROUGE-2 and METEOR to compensate for the individual metrics' weaknesses when measuring the generated feedback's quality (Post, 2018; Banerjee and Lavie, 2005). Thus, m balances the feedback generation and labelling performance, such that success on both tasks is required. Each model trained for approximately 1-5 hours on 2 Nvidia RTX 2080 Ti cards with 11 GB of RAM. The mT5 models were trained on a single card, due to the memory overhead of parallelization.

4.2 Results

Table 4 shows T5's, a majority baseline's and the average human performance on the English test sets. We report the accuracy and macro-averaged F1 score for classification and the root-mean-squared-error for scoring. Additionally, we compare the generated and annotated feedback to the gold standard using BERTScore[7] (Zhang et al., 2020) in addition to the metrics used during validation.

We can see that T5 provides a strong baseline for this task. However, there is still room for improvement compared to human performance, especially on unseen questions. A closer inspection of the generated feedback also revealed that the model would often, and often senselessly, copy common phrases it saw during training with minor modifications (see Appendix B). This indicates that elaborated feedback tasks can be challenging even to large language models.

Simultaneously, the models' high text similarity scores indicate a need for new evaluation metrics that measure similarity on a content- instead of lexical-level, enforcing that a text not only sounds well but also makes sense. Contrary to our belief, providing the model with more detailed scores

[6] https://pypi.org/project/sacrebleu/1.4.3/ default parameters (no smoothing, n-gram order=4)

[7] roberta-large_L17_no-idf_version=0.3.7(hug_trans=4.2.1)-rescaled and bert-base-multilingual-cased-rescaled

		Unseen Answers						Unseen Questions					
	Model	Acc.	F1	BLEU	MET.	ROU.	BERT	Acc.	F1	BLEU	MET.	ROU.	BERT
Label	Majority	54.0	23.4	-	-	-	-	47.1	21.4	-	-	-	-
	T5$_{wo_quest}$	74.2	72.0	33.7	59.0	52.8	65.0	66.7	55.9	10.7	36.4	31.1	52.2
	T5$_{w_quest}$	**75.0**	**75.9**	34.0	56.9	49.6	62.2	**67.4**	**69.7**	13.5	39.7	32.1	53.3
		RMSE						RMSE					
Score	Majority	0.470		-	-	-	-	0.512		-	-	-	-
	T5$_{wo_quest}$	0.290		33.7	56.9	50.4	62.8	0.263		9.0	35.3	29.1	49.7
	T5$_{w_quest}$	0.269		32.7	56.4	48.6	61.2	0.248		16.6	45.9	35.5	51.5
	Human	**0.099**		**45.5**	**64.9**	**56.5**	**68.5**	**0.086**		**57.1**	**71.6**	**64.3**	**75.7**

Table 4: T5's, a majority baseline's and the annotator's average results on the English unseen answers and unseen questions test splits. For the scoring and the labeling task, w_quest models additionally received the questions as input and wo_quest did not. Please note that the text similarity measures are in percent.

		Unseen Answers						Unseen Questions					
	Model	Acc.	F1	BLEU	MET.	ROU.	BERT	Acc.	F1	BLEU	MET.	ROU.	BERT
Label	Majority	44.6	20.6	-	-	-	-	46.2	21.1	-	-	-	-
	mT5$_{wo_quest}$	**85.2**	**85.1**	**50.7**	**51.2**	**31.4**	**54.9**	**54.7**	**41.7**	0.7	**20.1**	0.5	**21.9**
	mT5$_{w_quest}$	84.9	84.3	46.0	49.2	30.3	51.7	49.8	36.0	0.6	18.1	0.2	18.1
		RMSE						RMSE					
Score	Majority	0.538		-	-	-	-	0.426		-	-	-	-
	mT5$_{wo_quest}$	0.399		31.5	36.7	21.7	42.9	**0.360**		1.7	12.2	1.1	15.4
	mT5$_{w_quest}$	**0.196**		44.3	43.1	28.7	51.7	0.400		**2.0**	18.1	**1.5**	20.9

Table 5: mT5's results on the German test sets. We do not provide a human limit on the German dataset, as the test sets are only partially annotated by two annotators.

instead of only labels during training does not improve the feedback generation's performance. It even worsens performance slightly for most metrics.

On the English data, we observed that the question provided only a marginal benefit for unseen answers and a larger benefit for unseen questions. Interestingly, this trend does not seem to extend to the German dataset depicted in Table 5, indicating that this effect may be language or dataset dependent. Additionally, we can see that generalizing to new questions is even less successful on the German than on the English data. This may be due to the distribution of questions and answers in the datasets. While both are of similar size, there are significantly fewer German questions with more answers per question than English ones. The divergent answers to questions ratio may also explain why mT5 (German data) outperforms T5 (English data) when classifying or scoring unseen answers.

5 Conclusion and Future Work

This paper introduces the elaborated feedback generation task, challenging the limits of current Transformer models. We provide a benchmarking dataset containing short answers, scores, and textual explanations of given scores to kick off this task. As of yet, the dataset consists of 4,519 submissions to German and English questions. We demonstrate SAF's reliability with high inter-annotator agreements.

In Section 3.3, we presented aspects of the dataset we plan to improve. While the dataset is sizable for a manually annotated task of this complexity, it is small compared to other NLP tasks' crawled, large-scale datasets. We plan to mitigate this by incorporating additional questions in future iterations of the dataset. The focus will be on more complex questions to improve the class balance and questions of other domains and languages to increase diversity. The model's ability to generalize to unseen questions may also benefit from a more diverse dataset.

Finally, the baselines presented in this paper can be improved. Considering the deep understanding human graders require for this task, we believe neuro-symbolic approaches to be an exciting avenue of future research. Current models may especially benefit from incorporating knowledge bases and other reference material.

References

Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for nlp: A report on the first blackboxnlp workshop. *Natural Language Engineering*, 25(4):543–557.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. 2013. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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(1):60–117.

Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chatroom corpus. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20, Gothenburg, Sweden. LiU Electronic Press.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31, pages 9539–9549. Curran Associates, Inc.

Leon Camus and Anna Filighera. 2020. Investigating transformers for automatic short answer grading. In *Artificial Intelligence in Education*, pages 43–48, Cham. Springer International Publishing.

Nathan T Carr. 2020. Consistency of computer-automated scoring keys across authors and authoring teams. In *Another Generation of Fundamental Considerations in Language Assessment*, pages 173–199. Springer.

Pooja Chaudhary and Sachin Kumar. 2017. Comparative study of tcp variants for congestion control in wireless network. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 641–646.

Christine P Dancey and John Reidy. 2007. *Statistics without maths for psychology*. Pearson Education.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Arun Das and Paul Rad. 2020. Opportunities and challenges in explainable artificial intelligence (XAI): A survey. *Computing Research Repository*, arXiv:2006.11371.

Galina Deeva, Daria Bogdanova, Estefanía Serral, Monique Snoeck, and Jochen De Weerdt. 2021. A review of automated feedback systems for learners: Classification framework, challenges and opportunities. *Computers & Education*, 162.

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.

Myroslava Dzikovska, Natalie Steinhauser, Elaine Farrow, Johanna Moore, and Gwendolyn Campbell. 2014. 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.

Anna Filighera, Tim Steuer, and Christoph Rensing. 2020a. Fooling automatic short answer grading systems. In *Artificial Intelligence in Education*, pages 177–190, Cham. Springer International Publishing.

Anna Filighera, Tim Steuer, and Christoph Rensing. 2020b. Fooling it - student attacks on automatic short answer grading. In *Addressing Global Challenges and Quality Education*, pages 347–352, Cham. Springer International Publishing.

Muhammad Asif Hasan, Nurul Fazmidar Mohd Noor, Siti Soraya Binti Abdul Rahman, and Mohammad Mustaneer Rahman. 2020. The transition from intelligent to affective tutoring system: A review and open issues. *IEEE Access*, 8:204612–204638.

Scott Hellman, William Murray, Adam Wiemerslage, Mark Rosenstein, Peter Foltz, Lee Becker, and Marcia Derr. 2020. Multiple instance learning for content feedback localization without annotation. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 30–40, Seattle, WA, USA → Online. Association for Computational Linguistics.

Petri Ihantola, Tuukka Ahoniemi, Ville Karavirta, and Otto Seppälä. 2010. Review of recent systems for automatic assessment of programming assignments. In *Proceedings of the 10th Koli calling international conference on computing education research*, pages 86–93.

Judy Kay, Peter Reimann, Elliot Diebold, and Bob Kummerfeld. 2013. Moocs: So many learners, so much potential ... *IEEE Intelligent Systems*, 28(3):70–77.

Zixuan Ke and Vincent Ng. 2019. 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*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.

Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2018. A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education (TOCE)*, 19(1):1–43.

James A Kulik and JD Fletcher. 2016. Effectiveness of intelligent tutoring systems: a meta-analytic review. *Review of educational research*, 86(1):42–78.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. 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)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Anastasiya A Lipnevich and Jeffrey K Smith. 2009. "I really need feedback to learn:" Students' perspectives on the effectiveness of the differential feedback messages. *Educational Assessment, Evaluation and Accountability*, 21(4):347.

Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5570–5581, Florence, Italy. Association for Computational Linguistics.

Xin Lu, Barbara Di Eugenio, Stellan Ohlsson, and Davide Fossati. 2008. Simple but effective feedback generation to tutor abstract problem solving. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 104–112, Salt Fork, Ohio, USA. Association for Computational Linguistics.

Gaoyan Lv, Wei Song, Miaomiao Cheng, and Lizhen Liu. 2021. Exploring the effectiveness of question for neural short answer scoring system. In *2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC))*, pages 1–4.

Tomoya Mizumoto, Hiroki Ouchi, Yoriko Isobe, Paul Reisert, Ryo Nagata, Satoshi Sekine, and Kentaro Inui. 2019. 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*, pages 316–325, Florence, Italy. Association for Computational Linguistics.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online. Association for Computational Linguistics.

Elham Mousavinasab, Nahid Zarifsanaiey, Sharareh R. Niakan Kalhori, Mahnaz Rakhshan, Leila Keikha, and Marjan Ghazi Saeedi. 2021. Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1):142–163.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. WT5?! Training text-to-text models to explain their predictions. *Computing Research Repository*, arXiv:2004.14546.

Ildiko Pilan, John Lee, Chak Yan Yeung, and Jonathan Webster. 2020. A dataset for investigating the impact of feedback on student revision outcome. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 332–339, Marseille, France. European Language Resources Association.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself!

leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942. Association for Computational Linguistics.

Shourya Roy, Y Narahari, and Om D Deshmukh. 2015. A perspective on computer assisted assessment techniques for short free-text answers. In *International Computer Assisted Assessment Conference*, pages 96–109. Springer.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604, Stockholmsmässan, Stockholm Sweden. PMLR.

Valerie J. Shute. 2008. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.

Katherine Stasaski, Kimberly Kao, and Marti A. Hearst. 2020. CIMA: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, Seattle, WA, USA → Online. Association for Computational Linguistics.

Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. 2019. Improving short answer grading using transformer-based pre-training. In *International Conference on Artificial Intelligence in Education*, pages 469–481. Springer.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.

Naomi E. Winstone, Robert A. Nash, Michael Parker, and James Rowntree. 2017. Supporting learners' agentic engagement with feedback: A systematic review and a taxonomy of recipience processes. *Educational Psychologist*, 52(1):17–37.

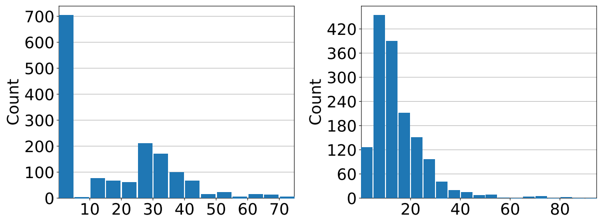
Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiaoming Xi. 2010. Automated scoring and feedback systems: Where are we and where are we heading? *Language Testing*, 27(3):291–300.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. 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*, pages 5456–5473, Marseille, France. European Language Resources Association.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.



(a) Feedback Length (b) Learner Answer Length

Figure 3: Histograms showing the distribution of text lengths (in tokens) in the German training set. The tail ends of b) is trimmed, leaving 3 unrepresented.

A German Length Statistics

The length of questions in the training set ranged from 12 to 20 tokens with reference answers between 48 and 84 tokens. The learners' answers were between 2 and 224 tokens long (average=14.7, median=11) and the corresponding feedback ranged between 2 and 71 tokens (average=17.4, median=14). The distribution of lenghths can be seen in Figure 3.

B Examples of Generated Feedback

Tables 6 and 7 contain example predictions generated by the T5$_{wo_quest}$ models. The examples stem from the English unseen answers test split. While the examples are handpicked, we did not choose them based on the quality of the generated feedback. Instead, they were selected to be as brief as possible while predicting the *partially correct* class or a matching score. This is because feedback for partially correct answers tends to be the most interesting as correct and incorrect aspects of the response are discussed.

Noteworthy is that many of the phrases used in the generated feedback are common in the training set. Take the feedback 4.1 for example, the sentence "The response correctly identifies the spanning-tree property that makes it appealing for broadcast and multicast" appears 20 times in the training split. The rest of the generated feedback is similar to the following feedback which occurs once in the training set: "The link-state modification for constructing spanning trees does not explain how each node shares its multicast information with others by adding it to the link state packet. Each node then has the complete information to build a multicast spanning tree." While the model made some modifications, most of the generated feedback seems to be a collection of various phrases seen during training. The generated feedback for unseen questions would also often match questions in the training set. For example, the feedback would state that the response correctly provided four challenges, even though the unseen question asked for an advantage and drawback of a completely different concept.

Question:
What are extension headers in IPv6 and where are they located in a packet? What is the main advantage of extension headers compared to IPv4?
Reference Answer:
Extension headers are used to extend the fixed IPv6 header with additional, optional network layer information. If present, they are located between the fixed header/main header and payload/upper-layer header/ transport-layer header. Main advantage: One of the following advantages are considered fully correct: 1. It allows the appending of new options without changing the header. 2. IPv6 packets with optional headers are typically processed faster/simpler by intermediate devices as most of the options are ignored (except "Hop-by-Hop Extension") while they are processed by all routers in IPv4 unless ordered otherwise.

Student Answer 1
The extension headers are placed between fixed header and payload. The main advantage is that they are optional. This means that the use of space is more efficient because there won't be any redundant information placed there in order to keep the expected length.
Score & Gold Feedback
0.25 feedback: The response answers only the location of extension headers correctly. The definition is missing and the advantage is incorrect when compared to IPv4 as the option field in IPv4 is also optional. Thus, there is no efficiency benefit.
Predicted Score & Feedback 1.1
0.75 feedback: the advantage part in the response is partially correct because the advantage part is somewhat ambiguous. extension headers are optional but it's more a part of description/feature instead of an advantage.
Predicted Label & Feedback 1.2
partially correct. the advantage is incorrect as the option field in the ipv4 header is optional and can be placed between payload and fixed header. the advantage is incorrect as the option field in the ipv4 header is optional and can be placed between payload and fixed header.

Student Answer 2
1.IPv6 extension headers contain supplementary information used by network devices (such as routers, switches, and endpoint hosts) to decide how to direct or process an IPv6 packet. 2.They are located between fixed header and payload. 3. IPv6 extension headers can be of arbitrary length.And the number of options that a packet carries are not limited.
Score & Gold Feedback
1.0 feedback: The response answers the description, location of extension headers, and advantage correctly.
Predicted Score & Feedback 2.1
1.0 feedback: the response answers all three parts of the question correctly.
Predicted Label & Feedback 2.2
partially correct. the advantage is partially correct as it does not state what extension headers are and how they are used in ipv4 and how they are used in ipv6. the advantage is incorrect as the option field in the ipv4 header is optional and can be of arbitrary length.

Table 6: Example in- and outputs of the T5 models. The question is not part of the model input.

Question:
WHICH PROPERTY of spanning trees makes them appealing for broad- and multicasting? EXPLAIN how you can modify Link State Routing to construct a spanning tree for multicasting.
Reference Answer:
Property: There is a single unique path between every pair of nodes in the tree. (...) This means that no unnecessary duplicates are distributed in the network when forwarding packets using that tree structure. Spanning Tree with Link State Routing: Each intermediate system knows which multicast groups it belongs to, but initially doesn't know which other IS belong to the groups. Therefore, you can add multicast group information to the link state packet and each node can construct multicast trees once the full network topology and group information are distributed in the whole network, as each node then has the complete state information stored locally (e.g. with Prim or Kruskal)

Student Answer 3
Spanning trees are appealing to broad- and multicasting scenarios, because they allow the packets to only travel one path (except travling backwards). This removes the need for looking up specific tables as in RPF / RPB.
Score & Gold Feedback
0.25 feedback: It is true that there is a unique path between nodes but that not only does away with the need to look at routing tables in RPF/RPB but reduces duplicates by removing loops(unnecessary links). No explanation was provided for modifying the link-state algorithm to construct a multicast spanning tree for nodes.
Predicted Score & Feedback 3.1
0.75 feedback: the response is partially correct because the link-state routing modification description is missing how the nodes belong to which groups.
Predicted Label & Feedback 3.2
partially correct. the explanation behind using a spanning tree for multicast and broadcast is partially correct as it does not mention how the link-state algorithm is modified to construct a multicast spanning tree. the explanation for modifying the link-state algorithm to construct a multicast spanning tree is partially correct as it does not mention how each node shares its multicast information with others by adding it to the link-state packet.

Student Answer 4
PROPERTY: The spanning tree does not have any loops.
Score & Gold Feedback
0.5 feedback: The response correctly answers why a spanning-tree usage is ideal in multicast and broadcast. The explanation for modifying the link-state algorithm to construct a multicast spanning tree for nodes is not provided.
Predicted Score & Feedback 4.1
0.75 feedback: the response correctly identifies the appealing property of a spanning tree for broadcast and multicast. the modification of the link state algorithm for constructing a multicast spanning-tree is not complete as it also needs to explain how link-state packets are expanded with multicast group information and how each node shares its multicast information with others by adding it to the link state packet. each node then has the complete information to build a multicast spanning tree.
Predicted Label & Feedback 4.2
partially correct feedback: the response correctly identifies the spanning-tree property that makes it appealing for broadcast and multicast and how the link-state algorithm is modified to construct a multicast spanning tree. the explanation for modifying the link-state algorithm to construct a multicast spanning tree is partially correct as it does not state how the link-state algorithm is modified to construct a multicast spanning tree.

Table 7: Example in- and outputs of the T5 models. The question is not part of the model input.