PAIS 2022
A. Passerini and T. Schiex (Eds.)

doi:10.3233/FAIA220062

Few-Shot Question Generation for Personalized Feedback in Intelligent Tutoring Systems

Devang Kulshreshtha [a,b] Muhammad Shayan [b] Robert Belfer [b] Siva Reddy [a,d]
Iulian Vlad Serban [b] Ekaterina Kochmar [b,c]
[a] *Mila/McGill University, Canada*
[b] *Korbit Technologies Inc., Canada*
[c] *University of Bath, United Kingdom*
[d] *Facebook CIFAR AI Chair*

Abstract. Existing work on generating hints in Intelligent Tutoring Systems (ITS) focuses mostly on manual and non-personalized feedback. In this work, we explore automatically generated questions as personalized feedback in an ITS. Our personalized feedback can pinpoint correct and incorrect or missing phrases in student answers as well as guide them towards correct answer by asking a question in natural language. Our approach combines cause–effect analysis to break down student answers using text similarity-based NLP Transformer models to identify correct and incorrect or missing parts. We train a few-shot Neural Question Generation and Question Re-ranking models to show questions addressing components missing in the student's answers which steers students towards the correct answer. Our model vastly outperforms both simple and strong baselines in terms of student learning gains by 45% and 35% respectively when tested in a real dialogue-based ITS. Finally, we show that our personalized corrective feedback system has the potential to improve Generative Question Answering systems.

Keywords. Intelligent tutoring systems, Natural language processing, Deep learning, Question Generation, Personalized learning and feedback

1. Introduction

Intelligent Tutoring Systems (ITS) are AI-powered instructional systems that provide personalized teaching to students [38]. ITS are a low-cost alternative to conventional classroom teaching, and shown to be more effective for tutoring students [34, 35]. One of the critical aspects of ITS is the ability to provide personalized feedback for exercises.

Many ITS however rely heavily on expert hand-crafted rules to generate feedback which becomes infeasible for large amounts of educational texts. An important research goal is to thus develop automated feedback systems from student-tutor interaction [24, 26]. Existing work mainly focuses on non-personalized hints created using template-based methods [2, 22]. However, students make various type of mistakes (such as grammatical errors, incorrect reasoning, and so on), therefore, showing the same hint to address them is not efficient in improving students' answers, and might even further

Exercise Problem: We want to choose between 2 treatments A and B. For both, we got same mean recovery rate but higher variance for treatment A. Which treatment would you discard, and why?

Student: Treatment A	*Student:* Treatment A
System [Non-personalized]: That's not right. Look at the variances and provide an explanation why you think one treatment is better than the other.	*System [Personalized]:* ”Treatment A” is correct! Try supplying a reason for this idea. Do we prefer more homogeneous results or less? *Student:* Less *System:* Ok, now try to answer original exercise.
Student: Treatment B?	*Student:* Treatment A, because it is less homogeneous than treatment B.
System: Not really. Let's move to another problem.	*System:* That's correct!

Table 1. Non-Personalized vs Personalized Feedback Generation in Korbit ITS. The Personalized Feedback pinpoints correct and missing parts in the answer and provides suggestions on how to improve it. In this case, the student forgot to provide reasoning for their answer and is asked a question about the missing part.

confuse them. As a result, this can lead to lower motivation and a decrease in the overall study time spent on an ITS platform.

We propose a novel automated personalized feedback system based on deep-learning based Transformer models [40, 20] to address the above-mentioned problems. Our model first breaks apart student answer into various components by performing cause-effect relation extraction [4]. Then it matches the components with gold standard answers using Transformers [40], and classifies them into various error categories (such as *missing explanation, incorrect main answer*, and so on). Next, a few-shot Transformer [28] model generates a personalized natural language question which is combined with the output of the cause–effect analysis to generate question-based feedback. We integrate the feedback in the conversation between an AI-tutor and a student. Such questions are easier to answer compared to the original exercises, as they are aimed at guiding a student towards improving their response.

Table 1 demonstrates a real interaction with the feedback system. Consider the case where student supplies the correct answer without an explanation. A non-personalized system would mark this answer incorrect due to the lack of explanation, even though the answer is correct, and provide the student with a generic hint, which may further confuse them and cause them to switch to an incorrect answer in their next attempt. In contrast, our personalized model informs the student that their answer is correct and prompts them to supply explanation. It then asks a clarifying question steering the student towards the reasoning which is missing. As a result, the student is able to provide the correct solution.

We test our method on Korbit ITS[1], a large-scale AI-powered personalized dialogue-based tutor. Students follow a blended-learning framework, which includes watching video lectures on data science and working on problem-solving exercises created by domain experts. Students' answers are compared to reference solutions using an ML-based solution verification model. We trigger hint generation when the Korbit's solution verification model marks a student's answer as incorrect. Student learning gains are measured after showing our feedback in Korbit. We show that our automated hint generation approach outperforms a minimal feedback (simple) baseline by 45% and personalized human feedback (strong) baseline by 35%.

[1] https://www.korbit.ai/

Connective	Reference solution
because	It's a discrete variable *because* it's counting the number of vehicles
,	No, the feature has 0 weight in the model function.
then	If the output is over the threshold *then* x is fraudulent

Table 2. Decomposition of reference solutions in Korbit ITS into their cause and effect.

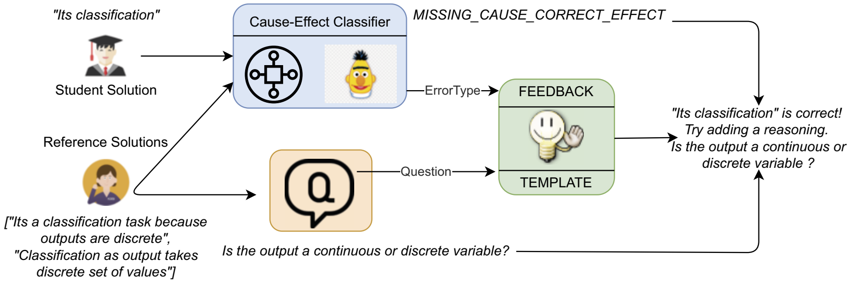
"Its classification"
Student Solution
Reference Solutions
["Its a classification task because outputs are discrete", "Classification as output takes discrete set of values"]
Cause-Effect Classifier
MISSING_CAUSE_CORRECT_EFFECT
ErrorType
FEEDBACK
TEMPLATE
Question
"Its classification" is correct! Try adding a reasoning. Is the output a continuous or discrete variable ?
Is the output a continuous or discrete variable?

Figure 1. An overview of our personalized feedback generation system: (a) Student solution is classified into its error type using cause-effect extractor and BERT similarity. (b) A few-shot QG model generates question from the *cause* of reference solution. (c) Personalized hint is generated using different feedback templates.

2. Background: Exercises in Korbit ITS

Each exercise in Korbit consists of a problem text, and one or more reference solutions. We focus on a particular class of exercises and name them as *cause-effect* exercises. In these exercises, the student is asked about identifying one or several relevant concepts, but they also require to justify the explanation behind their answer. An example of such exercise is *'Can linear regression be applied to classification? Why or why not?'*. Here the expected solution can be decomposed into an *answer (effect)* and *explanation (cause)*. For example, an acceptable solution to the problem above can be *'No, as the output variable of linear regression is continuous'*. Here, the cause is *'The output variable of linear regression is continuous'* and effect is *'No'*. Table 2 illustrates more such examples.

In contrast to reading comprehension exercises such as in SQuAD [29], cause–effect exercises require critical reasoning. The explanation component usually can not be found directly in pre-existing knowledge bases or text paragraphs. Therefore, the need for personalized feedback in such exercises is higher.

3. Personalized Feedback Generation Model

Our model generates feedback in three steps - (i) error classification (ii) Question Generation (iii) Full feedback generation. They are illustrated in Figure 1 and detailed below:

3.1. Cause-Effect Error Classifier

Decomposing a solution into its cause (explanation) and effect (answer) allows classification of student errors. Denote student solution as $s_s \equiv \{c_s, e_s\}$ and gold solution as

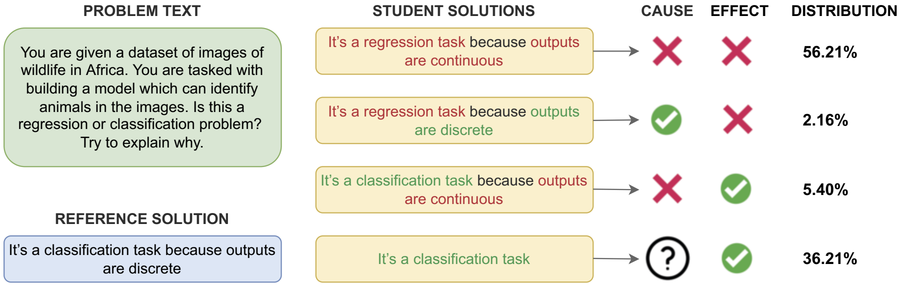
PROBLEM TEXT
You are given a dataset of images of wildlife in Africa. You are tasked with building a model which can identify animals in the images. Is this a regression or classification problem? Try to explain why.
REFERENCE SOLUTION
It's a classification task because outputs are discrete
STUDENT SOLUTIONS
It's a regression task because outputs are continuous
It's a regression task because outputs are discrete
It's a classification task because outputs are continuous
It's a classification task
CAUSE
EFFECT
DISTRIBUTION
56.21%
2.16%
5.40%
36.21%

Figure 2. Illustrating various types of student errors for a *cause-effect* exercise in Korbit ITS.

$s_r \equiv \{c_r, e_r\}$ decomposed into their cause c and effect e by running a cause-effect extractor described in Cao et al. [4]. The student deficiency falls into one of the four categories:

- Incorrect Cause $[c_s \neq c_r]$ Incorrect Effect $[e_s \neq e_r]$
- Correct Cause $[c_s \equiv c_r]$ Incorrect Effect $[e_s \neq e_r]$
- Incorrect Cause $[c_s \neq c_r]$ Correct Effect $[e_s \equiv e_r]$
- Missing Cause $[c_s \equiv \varnothing]$ Correct Effect $[e_s \equiv e_r]$

Figure 2 describes examples of all errors for a given exercise, as well as the error distribution generated by running cause-effect extractor over $7,000$ incorrect solutions.

To detect the error type, we match student cause-effect text with reference solution using BERTScore [40]. BERTScore uses pre-trained BERT [6] contextualised embeddings and computes overall similarity using weighted mean of cosine similarity between their tokens. It correlates better with human judgments compared with n-gram overlap based metrics (e.g. BLEU, ROUGE etc). BERTScore has been used as an evaluation metric for image captioning [40], summarization ([21]), machine translation ([37]) etc. BERTScore returns a score $(0-1)$ between student and reference cause/effect. If similarity exceeds a threshold $(= 0.8$, set manually) then cause/effect is considered correct.

3.2. *Few-shot Question Generation*

Our goal is to generate a question which forces the student to think about the incorrect/missing components in their solutions, and improve their answers. Our QG model pipeline comprises of four steps described below:

3.2.1. *Dataset creation*

We create a dataset by randomly sampling around 112 cause-effect exercises, giving us around 300 reference solutions for those exercises. We then ask four domain experts to write a question from the reference solutions, giving 75 instances to each annotator. Questions are written to *not* reveal the *effect/answer* and hence created only from *explanation* of reference solution. The annotators mainly write three type of questions - open-ended, binary, and binary with alternatives. Examples of such types are shown in Table 3 (the 'Score' column is explained later in Section 3.2.3). All annotators also annotate a shared set of 20 questions to ensure that annotators have low variance in created questions. We confirm that the questions created on the basis of the shared set are quite similar.

Reference Solution: *It is classification because coin flip outcome is discrete.*		
Question Type	**Example**	**Score**
Binary	Is flipping a coin discrete?	0.5
Binary with alternatives	Is flipping a coin discrete or continuous?	0.8
Open-Ended	What kind of action is flipping a coin?	1

Table 3. Taxonomy of questions written by annotators and corresponding scores used for question re-ranking.

3.2.2. Few-Shot Question Generation (QG) model

After data collection, we train a QG model to generate questions from reference solutions. We frame QG as a neural sequence-to-sequence task similar to Du et al. [8] where an encoder reads input text and decoder produces question by predicting one word at a time. We experiment with two pre-trained Transformers: BART [20] and T5 [28].

T5 is an encoder-decoder model pre-trained on a mixture of supervised and unsupervised NLP tasks where each task is converted into text-to-text input-output. T5 works well on a variety of conditional sequence generation tasks such as summarization [31], machine translation and question generation [7]. We name the model as *T5-QG*.

BART is a Transformer autoencoder pre-trained to reconstruct text from noisy text inputs. For QG, it learns a conditional probablity distribution $P(q|r)$ to generate question q from reference solution r. We experiment with two pre-trained checkpoints: original BART-base checkpoint provided by authors, and the BART model trained on 50K MLQuestions dataset using back-training algorithm [17], which generates good-quality questions on data science. We denote them as *BART-QG* and *BART-ML-QG*.

We split the data into 220 train, 40 validation and 40 test examples to train these models. Appendix A provides model training details.

3.2.3. Improving Question Generation using Re-Ranking

To improve question quality, we train a question re-ranker to chose the best question. First, we generate $k = 3$ questions per reference solution using beam search [10] for 80 randomly sampled reference solutions. Then we ask four domain experts to rate the usefulness of 240 generated questions on a scale of 1-5. Rating takes into account *factual correctness, fluency* and *relevance* of question with respect to the reference solution. Additionally, good quality questions based on question type are given higher score based on the preference - $\{$*open-ended* $>$ *binary with alternatives* $>$ *binary*$\}$ question (see Table 3 for question type examples). The 240 examples are distributed equally amongst three annotators. We find that the mean ratings given by each annotator was quite similar - $3.35, 3.4, 3.46$. Additionally, on a shared set of 20 examples annotated by all annotators we record an inter-annotator agreement of 0.75 which is substantial according to Landis and Koch [19].

Finally we train a Linear Regression model to predict usefulness taking the reference solution and generated question as input on 200 examples, and test on 40 examples. The input features to the regression model are -

- **Sentence Embeddings:** We use Sentence-BERT [30] to extract 768 dimensional embeddings from question. The Sentence-BERT uses siamese and triplet network structures to derive sentence embeddings from BERT and have been shown to perform extremely well in a range of tasks [30].

Algorithm 1 Personalized Feedback Generation in Korbit ITS

```
Require: Exercise problem $Q$, reference answers $\mathscr{R} \equiv \{s_r^i\}_{i=1}^m$, incorrect student answer $s_s$, Cause-Effect
    Extractor $\theta_{CE}$, BERTScore model $\theta_{BS}$, BERTScore similarity threshold $\tau_{BS}$, Question Generator $\theta_{QG}$.
Ensure: Personalized hint $h$
/*Find reference answer closest to student answer*/
$sim \leftarrow []$
for $s_r \in \mathscr{R}$ do
    add $\theta_{BS}(s_r, s_s)$ to $sim$
end for
$s_r \leftarrow \arg\max_i(sim)$
/*Classify student error and generate personalized hint*/
$(c_r, e_r) \leftarrow \theta_{CE}(s_r); (c_s, e_s) \leftarrow \theta_{CE}(s_s)$        ▷ Run cause-effect extractor
$q = \theta_{QG}(s_r)$        ▷ Generate question from reference solution.
switch $[c_r, e_r, c_s, e_s]$ do
    case $c_s \neq c_r$ and $e_s \neq e_r$        ▷ $[\theta_{BS}(c_s, c_r) < \tau_{BS}$ and $\theta_{BS}(e_s, e_r) < \tau_{BS}]$
        return "$\{e_s\}$ is incorrect. $\{q\}$?"
    case $c_s \neq c_r$ and $e_s \equiv e_r$        ▷ $[\theta_{BS}(c_s, c_r) < \tau_{BS}$ and $\theta_{BS}(e_s, e_r) \geq \tau_{BS}]$
        if $c_s \equiv \varnothing$ then
            return "$\{e_s\}$ is correct! Try supplying a reason for it. $\{q\}$?"
        else
            return "$\{e_s\}$ is correct! Try changing your reasoning. $\{q\}$?"
        end if
    case $c_s \equiv c_r$ and $e_s \neq e_r$        ▷ $[\theta_{BS}(c_s, c_r) \geq \tau_{BS}$ and $\theta_{BS}(e_s, e_r) < \tau_{BS}]$
        return "Did you mean $\{e_r\}$ because $\{c_s\}$?"
```

- **Well-formedness:** We train a BERT binary classifier to predict whether a question is well-formed or ill-formed on Google Well-formedness dataset [9]. We use the well-formedness probability of generated hint question as the *well-formed* feature.
- **Fluency:** We finetune a GPT-2 LM [3] on the 300 original hand-written questions (Section 3.2.1) using causal language modeling (LM) objective. The negative of LM perplexity of generated question is used as *fluency* feature.
- **Model Confidence:** This feature is computed as the negative loss of model when the generated question is considered as ground truth.
- **Question Type:** We want to penalise simple questions and reward questions which are more diverse and challenging to answer. For that, we apply a type-related score defined in Table 3 and use it as a feature.

We get a 772 dimensional feature vector and train our regression model using Ordinary Least Squares (OLS) objective on 200 examples. During inference, we use this question re-ranker to select the best question from the 5-best list for each reference solution.

After training the Question Generation and reranker model, we generate questions from all 1470 reference solutions in Korbit ITS using above models.

3.3. Providing Feedback

Using the output of cause-effect classifier and question generator, we provide feedback to reveal student deficiencies and suggest improvements. First we find the reference solution s_r closest to student solution s_s using BERTScore similarity. Then according to each error category identified by cause-effect classifier in Section 3.1, we create feedback using Algorithm 1.

Incorrect Cause [$c_s \neq c_r$] Incorrect Effect [$e_s \neq e_r$] First the system reveals error type by saying - *"$\{e_s\}$ is incorrect."*. Then it asks a question generated from s_r using QG model. When the student responds to this question, the system asks them to answer the original exercise again.

Incorrect Cause [$c_s \neq c_r$] Correct Effect [$e_s \equiv e_r$] Since the main answer (effect) is correct, first the system outputs - *"$\{e_s\}$ is correct! Try changing your reasoning."*. Then similar to previous error type, we ask a sub-question generated from s_r. After student answers this sub-question, the interface will ask them to answer original exercise again.

Missing Cause [$c_s \equiv \varnothing$] Correct Effect [$e_s \equiv e_r$] We show similar hint as previous error category, saying - *"$\{e_s\}$ is correct! Try supplying a reason for it. $\{q\}$?"*, where q is the generated question. This example is also illustrated in Figure 1.

Correct Cause [$c_s \equiv c_r$] Incorrect Effect [$e_s \neq e_r$] In practice this scenario rarely occurs. Since student supplied correct explanation, the incorrect answer is likely to be a mistake, which we help repair by asking the student *"Did you mean $\{e_r\}$ because $\{c_s\}$?"* with two answer options to chose from: "Yes, I agree" and "No, I disagree". If the student chooses former option then answer is marked correct.

4. Experimental Results

4.1. Question Generation

We evaluate the generation quality of three models - *T5-QG, BART-QG, BART-ML-QG* using standard language generation metrics: BLEU1-4 [27] and ROUGE-L [32] on the test set of 40 examples. The results are presented in Table 4. BART outperforms T5 by 4 BLEU1 points, showing that it is better suited for conditional generation. Also pre-training on MLQuestions dataset [17] increases BLEU1 by 1.5 absolute points.

Model	B1	B2	B3	B4	R
T5-QG	30.4	18.0	11.9	7.5	30.7
BART-QG	34.5	24.5	17.1	12.1	39.3
BART-ML-QG	**36.1**	**24.7**	**19.6**	**12.2**	**39.7**

Table 4. Results of Question Generation Models on standard language evaluation metrics.

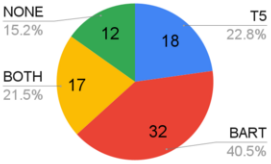

Figure 3. Comparing question quality of T5 with BART based on annotated 80 questions.

4.2. Question Re-ranking

For question re-ranking, we experiment using different combinations of features described in Section 3.2.3 to predict usefulness score of generated question -

1. **Mean Baseline:** This baseline simply outputs the usefulness as the average of all usefulness output in training set.
2. **Linguistic:** Here we only use the four linguistic features - *well-formedness, fluency, model confidence, question type score* as features for the question re-ranker.

Model	MSE	MAE	PCR	Usefulness
Mean Baseline	2.20	1.32	-	3.42
SBERT	1.74	1.16	0.38	3.96
Linguistic	1.78	1.16	0.33	3.85
Ling-SBERT	**1.72**	**1.15**	**0.40**	**4.01**

Table 5. Results of Question Re-ranking.

3. **SBERT:** Here we only use the 768-dimensional SBERT embedding features.
4. **Ling-SBERT:** In this model we concatenate SBERT sentence embeddings with four linguistic features to train our question re-ranker.

For each model we measure standard regression evaluation metrics - Mean Squared Error (MSE), Mean Absolute Error (MAE), Pearson Correlation (PCR). We also measure usefulness metric for each model. To compute usefulness, the re-ranker model predicts usefulness for each of the $k = 3$ questions of the same given reference solution from the test set. Then the actual usefulness label for question achieving highest score is averaged across all reference solutions in test set. The results are presented in Table 5. We find that Ling-SBERT outperforms all other models for all metrics. More importantly, it improves the usefulness rating from 3.42 to 4.01. This means incorporating question re-ranking improves the actual usefulness of question generation by 0.5 on average!

4.3. Human Evaluation

We manually compare the question quality generated by *T5-QG* and *BART-ML-QG* by generating questions for 80 randomly sampled reference solutions. The annotators compare both questions and provide one of the four labels - T5 (meaning T5 question is more useful than BART), BART (BART question is more useful), BOTH (both are equally good), and NONE (neither is a good question). The results from Figure 3 indicate that BART model is the clear winner, which is also supported by the superior BLEU scores.

Based on results on question generation, re-ranking and human evaluation we use the BART-ML-QG model for generation, and the Ling-SBERT model for re-ranking.

4.4. Student Learning Gains

After integrating our models in Korbit ITS, we collect around 146 distinct student interactions with feedback system for 550 exercises and measure student learning gains. The student learning gain is defined as the percentage of times a student answer is labelled correctly by the solution checker after they have received a given feedback. We compare our *Personalized Question-based Feedback* with both simple and strong baselines:

- **Minimal Feedback Baseline:** The system simply tells the student that their solution is incorrect and they should try again.
- **Personalized Human Feedback Baseline:** For every exercise, Korbit already has several hints manually crafted by course designers. To select the best hint from the ones available, the ITS uses a personalized ML model by looking at student performance and responses on the exercise [16]. This personalization is used only during hint selection, and *not* during hint generation itself.

	Average Learning Gain (%)	
Model	First Attempt	All Attempts
Minimal Feedback Baseline	22.58 ± 14.72	21.74 ± 11.92
Personalized Human Feedback Baseline	31.25 ± 16.06	30.43 ± 13.3
Personalized Non-question Feedback	41.67 ± 19.72	34.38 ± 16.46
Personalized Question-based Feedback	**66.67 ± 16.87**	**52.27 ± 14.76**

Table 6. Student learning gains on the Korbit ITS at 95% confidence intervals.

- **Personalized Non-question Feedback:** In this model after informing error type using cause-effect classifier, we reveal a part of the answer rather than asking a question. For e.g. if the student answers *'Its a regression task because outputs are continuous'*, we show the hint as *'"Its a regression task" is incorrect. Observe that outputs are discrete'* and ask the student to try again.

We present results of student learning gains in Table 6. The 'First Attempt' column indicates entries in which the student tried only once previously, while 'All Attempts' considers learning gains across student's all attempts. Our experiments show that our *Personalized Question-based Feedback* model outperforms all models.

The *Non-question Feedback* model improves over *minimal feedback baseline* by 18%, because it additionally informs about correct and incorrect/missing components. However, it cannot tell the student how to correct the incorrect/missing part. Our *Question-based Feedback* model further improves over it by 26%. This shows that asking questions about missing/incorrect parts is the key to help students improve their answers.

For all models, we observe that learning gains for 'First Attempt' are more than 'All Attempts'. This is likely because students who require many hints to solve an exercise may have knowledge gaps to solve exercises.

We find that most frequent student error is *'incorrect cause incorrect effect'* followed by *'missing cause correct effect'*. The error type *'Correct cause incorrect effect'* occurs rarely as students usually know the main answer if they know the explanation behind it.

5. Improving Generative Question Answering using Feedback Intervention

Will a student having access to a feedback generation to correct it's mistakes during training perform better than another student without the feedback system support? Assume Student S_A and S_B are being taught by instructors I_A and I_B. I_A trains S_A by showing the answer for many questions. While I_B trains S_B by showing answers for questions, as well as sending *personalized corrective feedback* when student answers question incorrectly. During test time, both students get same question paper without access to any feedback. We simulate this behaviour by replacing student by QA model and teacher by hint model:

1. Train baseline QA model θ_{QA} to generate reference solution from the question.
2. Generate machine (student) answers for questions in training data using θ_{QA} and generate hints using our feedback system for the incorrect answers.
3. Train hint generator θ_{HG} to generate these hints from question & machine answer.
4. Train hint-assisted QA model θ_{HQA} to generate answer from question and hint text generated by θ_{HG}.

Algorithm 2 Improving Generative QA using Personalized Feedback Generation

```
Require: QA Data $\mathscr{D}_{QA} \equiv \{(q^i, a^i)\}_{i=1}^m$, Personalized Hint Generator $\mathscr{H}$
Ensure: Hint assisted QA model $\theta_{HQA}$
$\theta_{QA} \leftarrow$ Train on $\mathscr{D}_{QA}$                                   ▷ Vanilla QA model
$\mathscr{D}_{HG} \leftarrow [\,]$                                              ▷ Synthetic data for $\theta_{HG}$
for $q, a \in \mathscr{D}_{QA}$ do
    Generate machine answer $\hat{a} = \theta_{QA}(q)$
    Generate personalized hint $h = \mathscr{H}(q, \hat{a}, a)$
    add $(q, \hat{a}, h)$ to $\mathscr{D}_{HG}$
end for
$\theta_{HG} \leftarrow$ Train on $\mathscr{D}_{HG}$ to generate $h$ from $(q, \hat{a})$
$\mathscr{D}_{HQA} \leftarrow [\,]$                                             ▷ Synthetic data for $\theta_{HQA}$
for $q, a \in \mathscr{D}_{QA}$ do
    Generate machine answer $\hat{a} = \theta_{QA}(q)$
    Generate hint $\hat{h} = \theta_{HG}(q)$
    add $(q, \hat{h}, a)$ to $\mathscr{D}_{HQA}$
end for
$\theta_{HQA} \leftarrow$ Train on $\mathscr{D}_{HQA}$ to generate $a$ from $(q, \hat{h})$
```

5. During inference, first generate machine answer using θ_{QA}. Next generate hint using θ_{HG} then generate final answer using θ_{HQA}.

The full algorithm is described in 2. To the best of our knowledge, we are the first to generate intermediate hints and to use them to generate the full answer. Similar inductive bias to learn the output in parts has been show to improve Question Answering [18] Question Generation [13].

5.1. Hint-Answer Entailment Consistency

It is reasonable to expect the generated hint and model answer should be consistent with each other i.e. *machine answer should **entail** model hint*. To ensure such inductive bias in the model, during inference, we generate $k = 3$ model answers and measure the entailment probability of each answer to model generated hint using entailment probability of RoBERTa model [2] trained on multiple entailment datasets [25]. We pick the model answer with the highest entailment probability.

5.2. Experiments and Results

We use BART to train $\theta_{QA}, \theta_{HG},$ and θ_{HQA}. Refer to A for model training details. Since there exists no generative cause-effect QA dataset to the best of our knowledge, we use Korbit dataset of 550 exercises and reference solutions. We split the data into 400 train, 50 validation and 100 test examples and measure BLEU and ROUGE metrics.

Experimental results presented in Table 7 demonstrate that Hint-assisted QA system is superior to Vanilla-QA model by 1 ROUGE point, and enforcing hint-answer entailment further boosts ROUGE by up to 1.5 points. Although the improvements are marginal, note that the task itself is hard as the training data is limited.

[2] https://huggingface.co/ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli

Models	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE-L
Vanilla-QA	24.57	14.89	10.70	8.27	29.68
Hint-assisted QA	25.16	15.07	11.56	9.37	30.63
Hint+Entailment	**25.54**	**16.05**	**12.19**	**9.57**	**31.35**

Table 7. Results on improving Generative Question Answering Using Hint Intervention

6. Related Work

Feedback Generation Previous research on dialogue-based ITS similar to Korbit investigated various aspects of automated feedback generation and adaptation [1, 23, 36]. IFor instance, Stamper et al. [36] investigated ways to augment their Deep Thought logic tutor with a Hint Factory that generated data-driven, context-specific hints for an ITS. The hints were effective in promoting learning, however, their approach mostly focused on the automated detection of the best hint sequence among hints consisting of logic rules, whereas our work focuses on methods of hint generation in natural language. The most similar work to ours is that Grenander et al. [11], who also generate personalized feedback based on cause–effect analysis, but do not use questions in their generated feedback, hence their feedback does not reveal any hint about correct answer.

Question Generation Previous research has focused on training neural Seq2Seq models [8, 41, 15] on supervised full QA datasets such as SQuAD [29]. QG in a few-shot setting under limited data has also been explored recently for multi-hop QG [39, 12].

Chen et al. [5] create a large-scale Educational QG dataset from KhanAcademy and TED-Ed data sources as a learning and assessment tools for students. Kulshreshtha et al. [17] also release a QG dataset comprising of data-science questions to promote research in domain adaptation. Unlike our questions, the questions in Chen et al. [5], Kulshreshtha et al. [17] are static and not personalized to the student. A recent work by Srivastava and Goodman [33] generates personalized questions according to the student's level by proposing a difficulty-controllable QG model. To the best of our knowledge, we are the first to use QG in an education context with real student interaction data.

Improving Question Answering using Hints is not yet studied clearly in NLP paradigm. A related work by Lamm et al. [18] proposes the use of *explanations* for an answer to improve Question Answering. They annotate a dataset of 8,991 QED explanations and use it to learn joint QA and explanation generation. Their explanations however are very different from our hints as they are non-personalized (fixed for a given question/answer).

7. Conclusion and Future Work

We show how can we provide personalized feedback to students in an ITS by combining rule-based models such as cause-effect extraction with deep-learning models such as few-shot Question generation and semantic similarity. Our approach identifies correct and incorrect/missing components in student answers using cause-effect analysis and BERT Transformer. The few-shot Question Generation and re-ranker model then generates questions to help improve student answer. Our model vastly outperforms both simple and strong baselines on student learning gains by a large margin on the Korbit ITS.

One area of future research is to design personalizing feedback for non cause-effect exercises. Another idea is to show multiple feedback to students and have them evaluate it either explicitly or implicitly by trying to answer the question-based feedback. This training signal can be used to further improve the feedback model using active learning.

A. Appendix - Question Generation Model Training Details

All three models - *T5-QG, BART-QG, BART-ML-QG* are trained for 5 epochs with learning rate of $1e-5$ and batch size of 8. For optimization we use Adam [14] with $\beta_1 = 0.9, \beta_2 = 0.999$. The input and output sequence length is padded to 512 and 150 tokens respectively. For generation we use beam search decoding [10] with number of beams set to 3. The initial checkpoint for the models can be found at - T5-QG[3], BART-QG[4], BART-ML-QG[5]. The hint-assisted QA models - $\theta_{QA}, \theta_{HG}, \theta_{HQA}$ are trained using same configurations and vanilla BART checkpoint.

References

[1] Christoph Benzmüller, Helmut Horacek, Ivana Kruijff-Korbayova, Manfred Pinkal, Jörg Siekmann, and Magdalena Wolska. Natural language dialog with a tutor system for mathematical proofs. In *Cognitive Systems*, pages 1–14. Springer, 2007.

[2] Paul Blayney and Mark Freeman. Automated formative feedback and summative assessment using individualised spreadsheet assignments. *Australasian Journal of Educational Technology*, 20(2), 2004.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NIPS 2020.*

[4] Mengyun Cao, Xiaoping Sun, and Hai Zhuge. The role of cause-effect link within scientific paper. In *2016 12th International Conference on Semantics, Knowledge and Grids (SKG)*, pages 32–39. IEEE, 2016.

[5] Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. Learningq: a large-scale dataset for educational question generation. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL 2019*, pages 4171–4186.

[7] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *NIPS*, 2019.

[8] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *ACL*, 2017.

[3] https://huggingface.co/docs/transformers/v4.18.0/en/model_doc/t5#transformers.T5ForConditionalGeneration

[4] https://huggingface.co/docs/transformers/v4.18.0/en/model_doc/bart#transformers.BartForConditionalGeneration

[5] https://huggingface.co/McGill-NLP/bart-qg-mlquestions-backtraining

[9] Manaal Faruqui and Dipanjan Das. Identifying well-formed natural language questions. In *EMNLP*, 2018.

[10] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, 2017.

[11] Matt Grenander, Robert Belfer, Ekaterina Kochmar, Iulian V Serban, François St-Hilaire, and Jackie CK Cheung. Deep discourse analysis for generating personalized feedback in intelligent tutor systems. In *The 11th Symposium on Educational Advances in Artificial Intelligence*, 2021.

[12] Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. Latent reasoning for low-resource question generation. In *Findings of ACL*, 2021.

[13] Junmo Kang, Haritz Puerto San Roman, and Sung-Hyon Myaeng. Let me know what to ask: Interrogative-word-aware question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 163–171, 2019.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Tassilo Klein and Moin Nabi. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. *arXiv preprint arXiv:1911.02365*, 2019.

[16] Ekaterina Kochmar, Dung Do Vu, Robert Belfer, Varun Gupta, Iulian Vlad Serban, and Joelle Pineau. Automated data-driven generation of personalized pedagogical interventions in intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 2021.

[17] Devang Kulshreshtha, Robert Belfer, Iulian Vlad Serban, and Siva Reddy. Back-training excels self-training at unsupervised domain adaptation of question generation and passage retrieval. *arXiv e-prints*, pages arXiv–2104, 2021.

[18] Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. Qed: A framework and dataset for explanations in question answering. *Transactions of the Association for Computational Linguistics*, 9:790–806, 2021.

[19] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[21] Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. *arXiv preprint arXiv:1909.00141*, 2019.

[22] Ming Liu, Yi Li, Weiwei Xu, and Li Liu. Automated essay feedback generation and its impact on revision. *IEEE Transactions on Learning Technologies*, 10(4): 502–513, 2016.

[23] Maxim Makatchev, Pamela W Jordan, Umarani Pappuswamy, and Kurt VanLehn. Representation and reasoning for deeper natural language understanding in a physics tutoring system. *AAAI*, 2011.

[24] Jessica McBroom, Irena Koprinska, and Kalina Yacef. A survey of automated programming hint generation: The hints framework. *ACM Computing Surveys (CSUR)*, 54(8):1–27, 2021.

[25] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[26] Florian Obermüller, Ute Heuer, and Gordon Fraser. Guiding next-step hint generation using automated tests. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 220–226, 2021.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

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

[29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[30] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

[31] Sascha Rothe, Joshua Maynez, and Shashi Narayan. A thorough evaluation of task-specific pretraining for summarization. In *EMNLP*, 2021.

[32] Lin CY ROUGE. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*, 2004.

[33] Megha Srivastava and Noah Goodman. Question generation for adaptive education. In *ACL*, 2021.

[34] Francois St-Hilaire, Nathan Burns, Robert Belfer, Muhammad Shayan, Ariella Smofsky, Dung Do Vu, Antoine Frau, Joseph Potochny, Farid Faraji, Vincent Pavero, et al. A comparative study of learning outcomes for online learning platforms. In *International Conference on Artificial Intelligence in Education*, pages 331–337. Springer, 2021.

[35] Francois St-Hilaire, Dung Do Vu, Antoine Frau, Nathan Burns, Farid Faraji, Joseph Potochny, Stephane Robert, Arnaud Roussel, Selene Zheng, Taylor Glazier, et al. A new era: Intelligent tutoring systems will transform online learning for millions. *arXiv preprint arXiv:2203.03724*, 2022.

[36] John C. Stamper, Michael Eagle, Tiffany Barnes, and Marvin Croy. Experimental Evaluation of Automatic Hint Generation for Logic Tutor. *International Journal of Artificial Intelligence in Education*, 22(1-2):3–17, 2013.

[37] Inigo Jauregi Unanue, Jacob Parnell, and Massimo Piccardi. Berttune: Fine-tuning neural machine translation with bertscore. *arXiv preprint arXiv:2106.02208*, 2021.

[38] Étienne Wenger. *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann, 1987.

[39] Jianxing Yu, Wei Liu, Shuang Qiu, Qinliang Su, Kai Wang, Xiaojun Quan, and Jian Yin. Low-resource generation of multi-hop reasoning questions. In *ACL*, 2020.

[40] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[41] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP*, 2018.