Automated Distractor and Feedback Generation for Math Multiple-choice Questions via In-context Learning

Hunter McNichols[*]**, Wanyong Feng**[*]**, Jaewook Lee**[*]**, Alexander Scarlatos**
University of Massachusetts Amherst
{wmcnichols,wanyongfeng,jaewooklee,ajscarlatos}@umass.edu

Digory Smith, Simon Woodhead
Eedi

Andrew Lan
University of Massachusetts Amherst

Abstract

Multiple-choice questions (MCQs) are ubiquitous in almost all levels of education since they are easy to administer, grade, and are a reliable form of assessment. An important aspect of MCQs is the distractors, i.e., incorrect options that are designed to target specific misconceptions or insufficient knowledge among students. To date, the task of crafting high-quality distractors has largely remained a labor-intensive process for teachers and learning content designers, which has limited scalability. In this work, we explore the task of automated distractor and corresponding feedback message generation in math MCQs using large language models. We establish a formulation of these two tasks and propose a simple, in-context learning-based solution. Moreover, we propose generative AI-based metrics for evaluating the quality of the feedback messages. We conduct extensive experiments on these tasks using a real-world MCQ dataset. Our findings suggest that there is a lot of room for improvement in automated distractor and feedback generation; based on these findings, we outline several directions for future work.

1 Introduction

Multiple-choice questions (MCQs) are widely used to evaluate students knowledge since they enable quick and accurate administration and grading. They are reliable since they are designed to measure specific learning objectives consistently [1–3]. MCQs are constructed in a specific format; see Figure 1 for an example. The *stem* refers to the statement on the problem setup and context, followed by a question that needs to be answered. Among the options, the correct one can be referred to as the *key*, while incorrect ones can be referred to as *distractors*. As the name implies, distractors in multiple-choice questions are typically formulated to align with common mistakes made by students. These distractors are chosen because students either i) lack the necessary comprehension of the concepts or skills tested in the question to accurately identify the key as the correct answer, or ii) hold misconceptions that result in selecting a specific distractor as their response.

While MCQs offer many advantages for evaluating students' knowledge, manually crafting high-quality MCQs is a demanding and labor-intensive process [4]. High-quality distractors should be plausible enough to mislead students but not so evidently incorrect as to be easily discernible. In addition, each distractor should include a feedback message to aide the student in identifying their error or

---

[*]These authors contributed equally to this work

Stem

Write 35 as a fraction of 80. Answer in the simplest form.

Key

A) $\dfrac{7}{16}$ ✓ Correct!

Explanation

LCM of 35 and 80 being 5, dividing both numerator and denominator by 5 results in 35/80 = 7/16.

Distractor

B) $\dfrac{35}{80}$

C) $\dfrac{7}{80}$ ✗ Not quite!

D) $\dfrac{80}{35}$

Feedback

It appears that you have not simplified the fraction.

You simplified the numerator while keeping the same denominator.

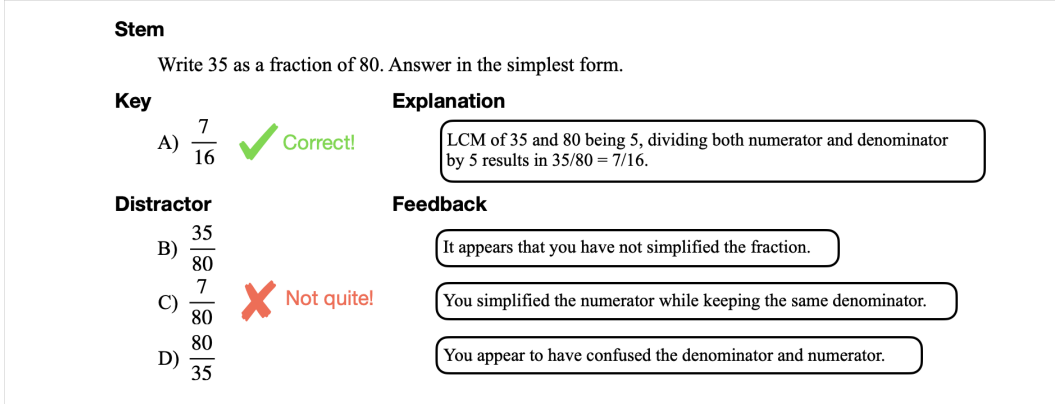You appear to have confused the denominator and numerator.

Figure 1: Different parts of math MCQs illustrated with an example.

misconception and lead them to the correct answer. Therefore, crafting ideal distractors and feedback is a challenging task which requires a deep understanding of possible student misconceptions.

Prior work on automatic distractor generation primarily focused on language learning and reading comprehension tasks, where distractors are used to assess students' comprehension of a given text or article. Early works used a ranking method based on semantic similarity and word collocation information or a pre-defined ontology to produce distractors [5, 6]. More recent works used encoder-decoder models with attention mechanisms for distractor generation, resulting in longer and higher-quality distractors [7–10]. Additionally, several recent works used pre-trained LLMs such as BERT and T5 for distractor generation in the context of Swedish reading and Cloze test [11–13]. However, to the best of our knowledge, there is limited work on distractor generation for math MCQs. Existing works either used constraint logic programming or sent predefined errors to a math problem solver to generate distractors [14, 15]. Prior work on feedback generation primarily focuses on language learning [16, 17] and programming [18–23] settings. These works typically leverage template-based rules or, occasionally, fine-tuned language models. For math, it is possible to use teachersourcing [24] or cognitive models [25–28] to anticipate common student errors and deploy pre-written feedback messages associated with those errors [29, 30], usually in intelligent tutoring systems [31, 32]. However, to the best of our knowledge, there is no prior work on automatically generating math feedback using LLMs or generating math feedback for MCQs.

1.1 Contributions

In this work, we investigate the tasks of automatically generating distractors and corresponding feedback messages for MCQs in the domain of math problems, in particular with LLMs. The first task is to generate plausible and challenging distractors for math MCQs automatically, given the question stem and (optionally) the key and its corresponding explanation. The second task is to generate informative and instructive feedback messages for each distractor automatically, given the question stem and the specific distractor. We believe that these two tasks can potentially be crucial to scaling up the design and support of MCQs in assessment and learning scenarios. We study a simple approach for the distractor and feedback generation tasks using few-shot, in-context learning [33] where the in-context examples are chosen as other MCQs with the highest semantic similarity, an approach that can serve as a baseline for future work. To evaluate the effectiveness of feedback, we explore using two *reference-free* metrics that leverage generative LLMs to evaluate the helpfulness and informativeness of the feedback. We discuss the potential benefits and limitations of these metrics compared to standard text similarity-based metrics in detail. We conduct extensive experiments on a real-world dataset of math MCQs with student response statistics, analyze our results both quantitatively and qualitatively, and suggest directions for future work.

2 Proposed Tasks

In this section, we first define relevant mathematical notation in MCQs and then formally define the automated distractor and feedback generation tasks. We define an MCQ Q as a set of textual
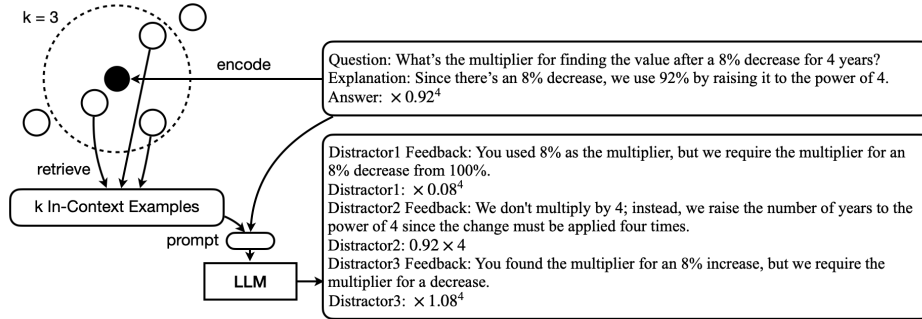
Figure 2: Overview of distractor generation with a math MCQ on "compound percentage decrease".

components defined below. In this paper, we do not consider questions that contain diagrams or images; extending our work to multi-modal question content is left for future work.

$$Q = \{s, k, e_k, D, F\}.$$

Here, Q contains a stem s, a key k, and an (optional) explanation of the key e_k. All of these components are sequences of words and math symbols (e.g., $s = \{w_1, \ldots, w_L\}$ where L is the length of the sequence s). In addition, Q contains a set of distractors D, each of which has an (optional) corresponding feedback message f_i which is shown to a student upon selecting d_i. The set of feedback messages is denoted as F. The distractors and feedback messages are likewise sequences of words and math symbols. See Figure 1 for an example MCQ.

2.1 Distractor Generation

We formulate the task of distractor generation as learning a function g^{dis} that outputs a set of distractors $\hat{D}$ for an MCQ given the question stem and (optionally) the key and its explanation, i.e.,

$$g^{\text{dis}}(s, k, e_k) \rightarrow \hat{D}$$

Our goal is to generate distractor options that students with specific misconceptions or insufficient knowledge on skills required for the question will select. This way, the MCQ can better distinguish between students that master all of the skills required for the question and those that do not. See Section 3.2 for details on our evaluation metrics for generated distractors.

2.2 Feedback Generation

We formulate the task of automated feedback generation as learning a function g^{fb} that outputs a textual feedback message given the question stem, a distractor, and (optionally) the key k and its explanation e_k. Note that in our setting, we generate a single feedback message that corresponds to each distractor, i.e.,

$$g^{\text{fb}}(s, d_i, k, e_k) \rightarrow f_i.$$

Our goal is to generate feedback messages for students who select a distractor to enhance their understanding of why their answer is incorrect and guiding them towards correcting their error.

3 Methodology

In this section, we detail our preliminary approach towards solving the tasks formulated in Section 2. We then define the metrics we use to evaluate our approaches to each task.

3.1 In-Context Learning

Our approach to solving both distractor and feedback generation tasks is to use a LLM to estimate the functions g^{fb}_ϕ and g^{dis}_ϕ, where ϕ are LLM parameters. We utilize in-context learning with similar MCQs chosen by the k-nearest neighbor (**kNN**) algorithm as few-shot examples for LLM input,

since kNN has been shown to be an effective example selection heuristic for LLM prompting [34]. Specifically, given a training dataset of MCQs, we first partition the dataset into two subsets: an example candidate set and a test set. To generate distractors for each target question in the test set, we employ a kNN approach to select a predetermined number of example questions from the candidate set. These selected examples are subsequently integrated as in-context samples in our prompts for the LLM. Figure 2 provides a visual representation of this process. To determine similarity, we calculate the *cosine similarity* between vectorized textual encodings of the examples and the target MCQ. We obtain these encodings using a pre-trained SBERT encoder `mpnet` [35], to which we provide a concatenated string of all MCQ components as input. See Table 3 for details on our prompt format.

3.2 Evaluation Metrics

3.2.1 Distractor Generation

We utilize alignment-based metrics to evaluate the extent to which the generated distractors align with the ground-truth, human-authored distractors. We denote the generated distractors as $\hat{D}$ where $|\hat{D}| = N$, the number of distractors in the MCQ. Prior to computing these metrics, the predicted and ground truth sets must be index-aligned. We introduce two binary metrics **Partial** match h_p (whether at least one generated distractor matches a ground-truth one) and **Exact** match h_e (whether all generated distractors match ground-truth ones), defined formally as:

$$h_p(D, \hat{D}) = \begin{cases} 1 & \exists \hat{d}_i \in \hat{D} : \hat{d}_i = d_i \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad h_e(D, \hat{D}) = \begin{cases} 1 & \forall \hat{d}_i \in \hat{D} : \hat{d}_i = d_i \\ 0 & \text{otherwise.} \end{cases}$$

We also utilize a continuous measure in the range $[0, 1]$ that we call **Proportional** match h_n (the portion of generated distractors that match ground-truth ones), i.e.,

$$h_n(D, \hat{D}) = \frac{\sum_{i=1} \mathbf{1}_{i:\hat{d}_i=d_i}}{N}$$

where $\mathbf{1}$ denotes an indicator function. We report metrics in Table 1 by averaging across all target MCQs in the test set and scale the values by a factor of 100 into percentages.

3.2.2 Feedback Generation

We utilize both existing reference-based and novel reference-free metrics to evaluate generated feedback messages. For reference-based metrics, we compare the actual, ground-truth human-written feedback for a distractor with the generated one using **BLEU** [36], a precision-based metric, **ROUGE-L** [37], a recall-based metric, and **METEOR** [38], a metric that leverages precision, recall, and word order. However, these metrics are imperfect since they do not target the underlying goal of feedback generation: to be informative to a student and help them correct their errors. Therefore, these metrics are susceptible to false positives (when the generated feedback matches well with the reference structurally but has key factual/logic errors) and false negatives (when the generated feedback does not match the reference structurally but is still valid and helpful). While it is common to use human evaluation [23] as a high-quality alternative to these metrics, it can be expensive and time-consuming. As an alternative, we propose two automated reference-free metrics, where we employ a generative LLM to act as a surrogate for human evaluators, along the lines of [39, 40]. We design these metrics based on the observation that feedback messages tend to play two primary roles: to be *helpful* for students to reach a correct answer, and to be an *explanation* for why an answer is incorrect.

Answer Adjustment This metric measures if a feedback message is *helpful* for correcting the corresponding error in a distractor. Specifically, we prompt a generative LLM to take steps indicated by the feedback to make an adjustment to the distractor, or formally, $g^{\text{adj}}(s, d_i, f_i) \to k$. We then measure if the adjusted answer is correct, indicating that the feedback is helpful, or incorrect, indicating that the feedback is misleading or not helpful. When reporting results, we denote the percentage of samples where the LLM adjusts the distractor to the correct answer as **Adj.**

Distractor Prediction This metric measures if a feedback message serves as an *explanation* for why a distractor is incorrect. Specifically, we prompt a generative LLM to use the feedback in order to predict what the corresponding distractor is, or formally, $g^{\text{dp}}(s, k, f_i) \to d_i$. We then measure if

Method	Exact	Partial	Proportional
kNN$^{\text{all}}$	**10.06**	**71.02**	**38.16**
kNN$^{\text{none}}$	6.01	54.52	27.20
kNN$^{\text{key}}$	8.13	61.48	32.39
Random	1.77	52.30	22.85
Zero-shot$^{\text{ChatGPT}}$	1.77	50.09	21.79
Zero-shot$^{\text{GPT-4}}$	3.18	44.52	21.67
kNN$^{\text{all}}_{\neg T}$	3.89	55.83	25.91

Table 1: Results of distractor generation where kNN-based methods often significantly outperform baselines.

Method	BLEU	ROUGE-L	METEOR	Adj.	Dist. Pred.
Ground-truth	–	–	–	**49.00**	**24.73**
kNN$^{\text{one}}$	**33.70**	**42.28**	**43.64**	46.64	18.26
kNN$^{\text{one}}_{\neg T}$	13.04	25.65	26.83	42.05	15.55
Random	4.21	20.08	18.63	42.17	13.19
Zero-shot	3.12	17.62	18.05	_47.70_	_20.49_

Table 2: Evaluation of generated feedback messages on reference-based and reference-free metrics.

the predicted answer is equal to the distractor, indicating that the feedback is informative, or not equal to the distractor, indicating that the feedback is misleading or too generic. When reporting results, we denote the percentage of samples where the LLM correctly predicts the distractor as **Dist. Pred.**

4 Experiments

4.1 Dataset, Methods, Experimental Setup

Our dataset consists of 1.4K MCQs from Eedi's [*] content repository, after filtering out questions with images/diagrams. Each question has four answer options, of which only one is correct, while the remaining options serve as distractors intended to reveal student misconceptions. See Figure 1 for an example of the different components in a MCQ, including explanation for the key and feedback for distractors. The questions are sourced from the broad mathematical topic titled "Number" with subtopics including "Basic Arithmetic", "Fractions", and "Rounding and Estimating". The questions are primarily targeted towards students aged 10-13. We divided the dataset into two subsets, namely a candidate example set and a test set, using an 80:20 ratio. The candidate example set is utilized to select MCQs as in-context examples and the test set contains all the target questions for generation and analysis. Across our experiments, we use a variety of prompt formats and corresponding encoding strategies; these formats are summarized in Table 3 in the Supplementary Material. For example, kNN$^{\text{all}}$ means that i) we use both the key and explanation in addition to the question stem to calculate similarities and ii) we use them together with all distractors and feedback messages of each example MCQ as input to the LLM.

For baselines, we include a **Random** heuristic that randomly selects few-shot examples for in-context learning. The prompt format for Random follows that of kNN$^{\text{all}}$, our best performing prompt format. We use three in-context examples for all few-shot methods. We also include **Zero-shot** as an additional baseline, where a generative LLM is solely provided with detailed instructions and information on the distractor and feedback generation tasks, without any in-context examples. We detail our prompts in the Supplementary Material, Section 5.1. We emphasize that there is no rule-based baeline that we can compare against since the MCQs in our real-world dataset are created by math teachers, not from a set of rigid rules. We use the OpenAI models `Codex` (code-davinci-002) [41] and `ChatGPT` (gpt-3.5-turbo) [42] as our generative LLMs, with greedy decoding and a maximum output length of 256 tokens. We employ `Codex` for both kNN and Random due to its proficient comprehension of mathematical content and ability in few-shot learning. We use the larger `ChatGPT` and `GPT-4` models for Zero-shot. We use `ChatGPT` for the reference-free feedback evaluation metrics; please see the Supplementary Material for all prompts we used in our experiments.

4.2 Results and Discussion

Distractor Generation   Table 1 shows results on distractor generation. We experiment with kNN across three prompt formats together with Random and Zero-shot prompting. To test generalizability, we also experiment with restricting the candidate example set by excluding questions from the same sub-topic. We observe that kNN-based methods have much higher performance on the alignment-based metrics compared to the Random and Zero-shot baselines. This result is not surprising since

---

[*] `https://eedi.com/home`

in some cases the target and example questions are highly similar and only differ in numerical values. Therefore, the LLM can generate distractors that match the ground-truth ones by simply replicating the style of the in-context samples and substituting the correct numbers. The baselines score significantly lower on these metrics, as the methods do not find or have informative in-context samples. Moreover, we observe that kNN$^{\text{all}}$ surpasses other kNN-based methods, as it uses the most information to identify in-context samples that are similar to the target question.

Next, we detail the impact of removing the same sub-topic from the candidate example set with kNN$^{\text{all}}_{\neg T}$. As shown in Table 1, doing so results in a huge performance drop-off from kNN$^{\text{all}}$. To further illustrate this drop-off, we perform a case study of the question described in Figure 2. To answer the MCQ in the Figure correctly, a student should understand and apply two math concepts: percentage decrease and compound interest. Ideally, good distractors should be designed to embed anticipated errors in understanding for both concepts needed to answer the question. For this question, kNN$^{\text{all}}$ generates all ground-truth distractors correctly, with the exact same anticipated errors as the ground-truth. Two distractors anticipate the error of misinterpreting "decrease by 8%" as either an increase by 8% or simply focusing on the 8% number itself. The other one anticipates an error of confusing compound interest with simple interest. Examining the in-context examples, we found that one retrieved example has an identical format to the question (differing only in numerical values), which helps the LLM generate the ground-truth distractors. The exact in-context examples retrieved are shown in Table 11. However, kNN$^{\text{all}}_{\neg T}$ only generates one of the correct distractors, $\times 1.08^4$. The other two non-matching distractors $\times 1.08$ and $\times 1.08^2$ are not related to misconceptions in compound interest. This result can be explained by examining the in-context examples retrieved by kNN$^{\text{all}}_{\neg T}$. The exact in-context examples retrieved are shown in Table 13. Since questions with the same sub-topic are removed from the candidate example set, kNN$^{\text{all}}_{\neg T}$ chooses questions that ask the following: i) multiplier for decreasing by a certain amount and ii) years to reach a certain value with depreciation. Since kNN relies on semantic similarity, it focuses only on the concept of percentage decrease, failing to find in-context examples of compound interest, which results in low-quality distractors. This case study highlights a major limitation in using semantic similarity as a proximity measure for kNN-based methods, which we leave for future work.

Qualitative Analysis for Distractor Generation We now qualitatively investigate the distractors generated by the best performing setting, kNN$^{\text{all}}$, to extract some insights on the distractor generation task and how to improve performance. We group questions into 4 categories according the number of generated distractors that match the human-authored ones, from 0 to 3.

For the group where all generated distractors match the ground truth (3 out of 3), we find that, in all but 2 of the 30 cases, there was an in-context example that is very similar to the target question, with the only difference being different numerical values or names. However, this situation often appears in other categories as well, which is perhaps surprising since it implies that the presence of a near-identical in-context example alone is not sufficient for an LLM to generate quality distractors. Another possible explanation is that for two near-identical questions, their sets of distractors may differ, which suggests that the number of appropriate distractors for each question differs and there is a lot of variance in which set of 3 is actually used.

For the group where none of the generated distractors match the ground truth, we examine 22 of the 82 cases. We find that in 17 of the 22 cases (77%), the generated distractors are reasonable and the ground truth distractors are not clearly superior to the generated ones; see Table 9 for an example. While this observation is entirely subjective, it highlights that alignment-based metrics may not be an appropriate metric to measure the quality of distractor generation since human-authored distractors may not be optimal. Moreover, since many generated distractors are actually reasonable even if they are not among the human-authored ones, there is promise in using automated distractor generation for teacher support during the creation of MCQs.

Finally, for the groups where 1 or 2 generated distractors match the ground truth, we examine which human-authored distractor(s) were generated and which were not. We find that in many cases, one of the generated distractors that matches the ground truth seems to come from a typical misconception relevant to the question, while other human-authored distractors do not seem to correspond to any typical misconception; see Table 10 for an example.

Feedback Generation Table 2 shows results on feedback generation. We experiment with kNN^{one}, Random, and Zero-shot prompting, as well as kNN prompting with questions in the same topic excluded from the pool ($kNN^{one}_{\neg T}$). As a reference, we also show results of the ground-truth feedback messages on the reference-free metrics. We observe that kNN^{one} has the best performance on the reference-based metrics by far compared to other methods. This result is likely due to many feedback messages being almost identical for similar distractors across questions in the same topic; as a result, the LLM can generate feedback messages that are highly similar to the ground message by simply copying the style of the in-context examples. The remaining methods score much lower on these metrics, mainly because they are not able to copy in the same way. However, the evaluation is much closer across methods on the reference-free metrics, which indicates that the reference-based metrics are overly biased with respect to the style of the ground-truth feedback. The ground-truth feedback messages outperform all prompting methods on the Adj. and Dist. Pred. metrics, as expected. Among the prompting methods, Zero-shot performs the best, although we observe that the messages generated with this method are more likely to include the correct answer or distractor in the feedback itself, making it easier to perform well on the metrics without needing high quality feedback. We also observe that kNN^{one} outperforms the remaining methods on these metrics and that it is less likely than Zero-shot to copy the answer or distractor into the feedback. We provide an example of feedback generated by each method on a single question/distractor in Table 4.

There is a relatively small difference between the generation methods and the ground-truth feedback on the reference-free metrics. This observation is surprising since we expect the ground-truth feedback to be significantly better than most generated feedback messages. It is also surprising that the LLM fails to correct the answer even using the ground-truth feedback since recent models like `ChatGPT` are much more capable in math reasoning compared to earlier models. We study several common cases for *false negatives* and *false positives* to explain this result.

For false negatives, many questions are framed in the context of all options, such as asking the student to select the best estimate among options or select the option that satisfies a particular constraint. This setting makes the evaluation tasks difficult since each feedback message is evaluated in the context of a single distractor. Moreover, we observe that the LLM often ignores the instructions on predicting a distractor and simply outputs the correct answer. Furthermore, we observe that the LLM often misinterprets the meaning of the feedback, even if it is indeed high-quality, leading to incorrect predictions.

For false positives, the LLM often correctly solves the problem on its own, even if the given feedback is unhelpful or inaccurate. Moreover, the LLM often makes a lucky guess of the distractor even if the feedback is not specific enough. Overall, we believe that most of these issues are due to limitations of the LLM and could likely be relieved with further prompt engineering or by using more capable models such as `GPT-4`, which we leave for future work.

Despite these limitations, we argue that these metrics provide meaningful results on the *relative* performance of methods compared to each other and are significantly less biased than the reference-based metrics. For more details, we provide instances where the metrics work well in Tables 5, 6, and 7, as well as an instance where the metrics fail in Table 8, in the supplementary material.

5   Conclusions, Future Work, and Acknowledgement

In this paper, we have formulated two tasks for multiple-choice questions: automated distractor and feedback generation. We experimentally validated large language model-based approaches for these tasks on a real-world dataset. We reported our results using a series of metrics, including standard ones and two novel ones we proposed. The results have shown that these tasks are still challenging despite rapid recent advancement in large language models. Our initial exploration opens up many avenues for future work. First, we need to explore approaches other than large language model prompting such as fine-tuning a smaller, task-specific model (although we did not find fine-tuning to be effective in some preliminary experiments). Second, we need to develop modified text encoding methods that are more closely aligned with student errors, which will enable us to move away from using surface semantic similarity in question statement to select in-context examples. Finally, we need to conduct a human evaluation on the generated distractors and feedback messages. The authors thank the NSF (under grants 2118706 & 2237676) and Schmidt Futures for supporting this work.

References

[1] Anthony J. Nitko. *Educational assessment of students.* Prentice-Hall, Iowa, USA, 1996.

[2] Peter Airasian. *Classroom assessment: Concepts and applications.* McGraw-Hill, Ohio, USA, 2001.

[3] Tom Kubiszyn and Gary Borich. *Educational testing and measurement.* John Wiley & Sons, New Jersey, USA, 2016.

[4] Kim Kelly, Neil Heffernan, Sidney D'Mello, Namais Jeffrey, and Amber C. Strain. Adding teacher-created motivational video to an its. In *Proceedings of 26th Florida Artificial Intelligence Research Society Conference*, pages 503–508, 2013.

[5] Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. Automatic distractor generation for multiple-choice english vocabulary questions. *Research and practice in technology enhanced learning*, 13:1–16, 2018.

[6] Katherine Stasaski and Marti A Hearst. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312, 2017.

[7] Zhaopeng Qiu, Xian Wu, and Wei Fan. Automatic distractor generation for multiple choice questions in standard tests. *arXiv preprint arXiv:2011.13100*, 2020.

[8] Pengju Shuai, Li Li, Sishun Liu, and Jun Shen. Qdg: A unified model for automatic question-distractor pairs generation. *Applied Intelligence*, 53(7):8275–8285, 2023.

[9] Jiayuan Xie, Ningxin Peng, Yi Cai, Tao Wang, and Qingbao Huang. Diverse distractor generation for constructing high-quality multiple choice questions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:280–291, 2021.

[10] Yifan Gao, Lidong Bing, Piji Li, Irwin King, and Michael R Lyu. Generating distractors for reading comprehension questions from real examinations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6423–6430, 2019.

[11] Dmytro Kalpakchi and Johan Boye. Bert-based distractor generation for swedish reading comprehension questions using a small-scale dataset. *arXiv preprint arXiv:2108.03973*, 2021.

[12] Shang-Hsuan Chiang, Ssu-Cheng Wang, and Yao-Chung Fan. Cdgp: Automatic cloze distractor generation based on pre-trained language model. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5835–5840, 2022.

[13] Ricardo Rodriguez-Torrealba, Eva Garcia-Lopez, and Antonio Garcia-Cabot. End-to-end generation of multiple-choice questions using text-to-text transfer transformer models. *Expert Systems with Applications*, 208:118258, 2022.

[14] Ana Paula Tomás and José Paulo Leal. Automatic generation and delivery of multiple-choice math quizzes. In *Principles and Practice of Constraint Programming: 19th International Conference, CP 2013, Uppsala, Sweden, September 16-20, 2013. Proceedings 19*, pages 848–863. Springer, 2013.

[15] Neisarg Dave, Riley Bakes, Barton Pursel, and C Lee Giles. Math multiple choice question solving and distractor generation with attentional gru networks. *International Educational Data Mining Society*, 2021.

[16] Su-Youn Yoon, Ching-Ni Hsieh, Klaus Zechner, Matthew Mulholland, Yuan Wang, and Nitin Madnani. Toward automated content feedback generation for non-native spontaneous speech. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 306–315, Florence, Italy, August 2019. Association for Computational Linguistics.

[17] Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, Verena Möller, Florian Nuxoll, and Detmar Meurers. Generating feedback for English foreign language exercises. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–136, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[18] Rishabh Singh, Sumit Gulwani, and Armando Solar-Lezama. Automated feedback generation for introductory programming assignments. *SIGPLAN Not.*, 48(6):15–26, jun 2013.

[19] Sumit Gulwani, Ivan Radiček, and Florian Zuleger. Feedback generation for performance problems in introductory programming assignments. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2014, page 41–51, New York, NY, USA, 2014. Association for Computing Machinery.

[20] Dowon Song, Woosuk Lee, and Hakjoo Oh. Context-aware and data-driven feedback generation for programming assignments. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2021, page 328–340, New York, NY, USA, 2021. Association for Computing Machinery.

[21] Marcus Messer. Grading programming assignments with an automated grading and feedback assistant. In Maria Mercedes Rodrigo, Noburu Matsuda, Alexandra I. Cristea, and Vania Dimitrova, editors, *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium*, pages 35–40, Cham, 2022. Springer International Publishing.

[22] Hemilis Joyse Barbosa Rocha, Patrícia Cabral de Azevedo Restelli Tedesco, and Evandro de Barros Costa. A context-aware approach to personalized feedback for novice programmers. In Maria Mercedes Rodrigo, Noburu Matsuda, Alexandra I. Cristea, and Vania Dimitrova, editors, *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium*, pages 59–64, Cham, 2022. Springer International Publishing.

[23] Qinjin Jia, Mitchell Young, Yunkai Xiao, Jialin Cui, Chengyuan Liu, Parvez Rashid, and Edward Gehringer. Insta-reviewer: A data-driven approach for generating instant feedback on students' project reports. *International Educational Data Mining Society*, 2022.

[24] Thanaporn Patikorn and Neil T Heffernan. Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 115–124, 2020.

[25] Kenneth R. Koedinger and Albert Corbett. *Cognitive tutors: Technology bringing learning sciences to the classroom.* 2006.

[26] Ran Liu, Rony Patel, and Kenneth R. Koedinger. Modeling common misconceptions in learning process data. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 369–377, 2016.

[27] Eleanor O'Rourke, Eric Butler, Armando D. Tolentino, and Zoran Popovic'. Automatic generation of problems and explanations for an intelligent algebra tutor. In *Proceedings of the Artificial Intelligence in Education: 20th International Conference*, pages 383–395, 2019.

[28] Steven Ritter, John R. Anderson, Kenneth R. Koedinger, and Albert Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic bulletin & review*, 14:249–255, April 2007.

[29] Andrew S. Lan, Divyanshu Vats, Andrew E. Waters, and Richard G. Baraniuk. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, L@S '15, page 167–176, New York, NY, USA, 2015. Association for Computing Machinery.

[30] Mengxue Zhang, Zichao Wang, Richard Baraniuk, and Andrew Lan. Math operation embeddings for open-ended solution analysis and feedback, 2021.

[31] Renah Razzaq, Korinn S. Ostrow, and Neil T. Heffernan. Effect of immediate feedback on math achievement at the high school level. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 263–267, Cham, 2020. Springer International Publishing.

[32] Ekaterina Kochmar, Dung Do Vu, Robert Belfer, Varun Gupta, Iulian Vlad Serban, and Joelle Pineau. Automated personalized feedback improves learning gains in an intelligent tutoring system. In Ig Ibert Bittencourt, Mutlu Cukurova, Kasia Muldner, Rose Luckin, and Eva Millán, editors, *Artificial Intelligence in Education*, pages 140–146, Cham, 2020. Springer International Publishing.

[33] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. abs/2005.14165, 2020.

[34] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3. abs/2101.06804, 2021.

[35] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. Bleu: a method for automatic evaluation of machine translation. pages 311–318, 2002.

[37] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[38] Satanjeev Banerjee and Alon Lavie. 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, June 2005. Association for Computational Linguistics.

[39] Roman Rietsche, Andrew Caines, Cornelius Schramm, Dominik Pfütze, and Paula Buttery. The specificity and helpfulness of peer-to-peer feedback in higher education. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 107–117, Seattle, Washington, July 2022. Association for Computational Linguistics.

[40] Qinjin Jia, Yupeng Cao, and Edward Gehringer. Starting from "zero": An incremental zero-shot learning approach for assessing peer feedback comments. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 46–50, Seattle, Washington, July 2022. Association for Computational Linguistics.

[41] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

[42] OpenAI. Introducing chatgpt, 2022.

Method	Encode			Prompt			
	k	e_k	d	k	e_k	$F(f)$	$D(d)$
kNN^{none}	no	no	no	no	no	none	all
kNN^{key}	yes	no	no	yes	no	none	all
kNN^{all}	yes	yes	no	yes	yes	all	all
kNN^{one}	no	no	yes	no	no	one	one

Table 3: Encoding strategies for retrieval and prompt formats used in kNN-based methods.

Supplementary Material

5.1 Prompts

We provide the prompts we use for several zero-shot tasks in the work below. We use $<>$ to indicate that a variable is filled in dynamically.

You are given the following math question along with the correct answer and explanation. Please use the following template to give three alternative incorrect answers to be used as multiple-choice options in a multiple-choice exam. Prior to the incorrect answer, provide feedback to be displayed to the student as an explanation of why that is not the correct answer.
[Template]
Distractor1 Feedback:
Distractor1:
Distractor2 Feedback:
Distractor2:
Distractor3 Feedback:
Distractor3:
Question: <question>
Explanation: <explanation>
Answer: <correct answer>

Figure 3: Distractor generation zero-shot prompt.

You are given the following math question along with an incorrect answer chosen by a student. Please write feedback for the incorrect answer that would help the student understand and correct their mistake. The feedback should be short, only one or two sentences.
Question: <question>
Incorrect Answer: <distractor>
Feedback:

Figure 4: Feedback generation zero-shot prompt.

Given the question: "<question>"
A student gives the incorrect answer: "<distractor>"
Their teacher gives the feedback: "<feedback>"
Using only this feedback, adjust the student's answer. Do not take steps that are not indicated by the feedback. If it is not possible to update the student's answer based on the feedback, respond with "Invalid Feedback". Write your updated answer at the end of the output using the template: "Updated Answer: {updated answer}". Only include the answer in its final form.

Figure 5: Answer adjustment evaluation prompt.

Given the question: "<question>"
The correct answer is: "<correct answer>"
A student gives an incorrect answer, to which their teacher gives the feedback: "<feedback>"
Using only this feedback, predict exactly what the student's incorrect answer was. Do not take steps that are not indicated by the feedback. If there is not enough information to determine exactly what the student's answer was based on the feedback, respond with "Insufficient Information". Write your final prediction at the end of the output using the template: "Student Answer: {the student's answer}". Only include the answer in its final form.

Figure 6: Distractor prediction evaluation prompt.

5.2 Examples of Generated Feedback

Question	which multiplier can be used to find the value after an amount has decreased in value by 8% for 4 years?
Correct Answer	$\times 0.92^4$
Distractor	$\times 1.08^4$
Ground-truth Feedback	It looks like you have found the multiplier for an increase of 8% but we need to find the decrease.
Generated Feedback (kNN$^{\text{one}}$)	It looks like you have found the multiplier for an increase of 8% but we need to find the decrease.
Generated Feedback (kNN$^{\text{one}}_{\neg T}$)	This is the multiplier for increasing by 8% for 4 years. We are decreasing by 8% for 4 years.
Generated (Random)	It looks like you have found the multiplier for a decrease of 8% for 1 year.
Generated Feedback (Zero-shot)	The incorrect answer uses the wrong multiplier. To find the value after a decrease, you need to use a multiplier less than 1.

Table 4: Examples of feedback generation across prompting methods. kNN$^{\text{one}}$ generates the exact same message as the ground-truth feedback, kNN$^{\text{one}}_{\neg T}$ generates a similar message that is still valid, Random generates a similar message that is invalid, and Zero-shot generates a dissimilar message that is still valid though less informative.

Question	sheila is listing the multiples of 7 in ascending order. she start with 7. what number should she write next?
Correct Answer	14
Distractor	1
Ground-truth Feedback	Not quite, I think you divided by 7. We need to find the multiples of 7, the numbers in the 7 times table
Adjustment Prediction	14 ✓
Distractor Prediction	1 ✓
Generated Feedback (kNN$^{\text{one}}$)	I think you're confusing factors and multiples; 1 is the highest common factor of 7 and 7. The word 'multiple', has more of a clue to it ... think 'multiply', so the times table for that number.
Adjustment Prediction	14 ✓
Distractor Prediction	1 ✓
BLEU	0.0
ROUGE-L	33.33
METEOR	35.53

Table 5: Example of feedback evaluation, for both ground-truth feedback and feedback generated by kNN$^{\text{one}}$ prompting. Generated predictions for both reference-free metrics are shown (with generated reasoning steps removed for brevity). Both feedback messages are high quality and pass the reference-free metrics.

Question	a 3 digit number belongs in this box to make the calculation true
	$$467 + \square = 1253$$
	what is the 3rd digit of the number that belongs in the box?
Correct Answer	6
Distractor	4
Ground-truth Feedback	I think you've found what would be added to 3 to get 7. Adding 4 in the units would give an answer with units of 1. Here we need a number to add to 7 to get to 3 (or 13)
Adjustment Prediction	6 ✓
Distractor Prediction	4 ✓
Generated Feedback (kNN$^{\text{one}}$)	4 is the units in the final value, however it is not what we have to add to 467. Adding 4 in the units column would give us 7 units in the answer. What can we add to 7 to get 4 (or 14)?
Adjustment Prediction	7 ✗
Distractor Prediction	4 ✓
BLEU	21.37
ROUGE-L	41.86
METEOR	39.06

Table 6: Example of feedback evaluation, for both ground-truth feedback and feedback generated by kNN$^{\text{one}}$ prompting. Generated predictions for both reference-free metrics are shown (with generated reasoning steps removed for brevity). The ground-truth feedback message is high quality and passes both reference-free metrics. However, the generated feedback message contains a misleading error, and fails the first reference-free metric while passing the second.

Question	
	43.863
	truncate this number to 1 decimal place
Correct Answer	43.8
Distractor	43.9
Ground-truth Feedback	Not quite! Remember with truncating we just cut off the rest of the digits without any rounding.
Adjustment Prediction	43.8 ✓
Distractor Prediction	43.9 ✓
Generated Feedback (kNN$^{\text{one}}$)	43.863 truncated to 1 decimal place is 43.8.
Adjustment Prediction	43.8 ✓
Distractor Prediction	43.86 ✗
BLEU	0.0
ROUGE-L	0.0
METEOR	5.56

Table 7: Example of feedback evaluation, for both ground-truth feedback and feedback generated by kNN$^{\text{one}}$ prompting. Generated predictions for both reference-free metrics are shown (with generated reasoning steps removed for brevity). The ground-truth feedback message is high quality and passes both reference-free metrics. However, the generated feedback message simply gives away the answer without pointing out the error, and thus passes the first reference-free metric but fails the second.

Question	
	$$-(25)^2 = \square$$

Correct Answer	-625
Distractor	625
Ground-truth Feedback	Not quite. As the negative is outside the bracket you need to work out 25 squared and then make the answer negative
Adjustment Prediction	-625 ✓
Distractor Prediction	625 ✓
Generated Feedback (Zero-shot)	Remember to apply the order of operations correctly. The negative sign should be applied first, then the exponent.
Adjustment Prediction	-625 ✓
Distractor Prediction	625 ✓
BLEU	0.0
ROUGE-L	20.0
METEOR	15.35

Table 8: Example of feedback evaluation, for both ground-truth feedback and feedback generated by Zero-shot prompting. Generated predictions for both reference-free metrics are shown (with generated reasoning steps removed for brevity). The ground-truth feedback message is high quality and passes both reference-free metrics. However, the generated feedback states the incorrect order of operations, yet is able to elicit correct predictions for the reference-free metrics.

5.3 Distractor Output Examples

Question	Sheila is listing the multiples of 7 in ascending order. She starts with 7. What number should she write next?		
Correct Answer	14		
Authored Distractors	8	49	1
Generated Distractors	7	21	28

Table 9: Example of generated distractor quality in a "failure" case. No distractors overlap between human-authored and generated but all are reasonable distractor options. Distractors were generated by kNN$^{\text{all}}$ prompting strategy.

Question	Convert 0.6 to a fraction in its simplest form.		
Correct Answer	$\frac{3}{5}$		
Authored Distractors	$\frac{6}{10}$	$\frac{60}{100}$	$\frac{1}{6}$
Generated Distractors	$\frac{6}{10}$	$\frac{5}{3}$	$\frac{10}{6}$

Table 10: Example of generated distractors where the key distractor is generated and the other distractors are unlikely to be selected. In this case the $\frac{6}{10}$ is selected by 28% of students while the others are selected by less than 8%. Distractors were generated by kNN$^{\text{all}}$ prompting strategy.

5.4 Retrieved Examples for In-context Learning

Question	which multiplier can be used to find the value after an amount has decreased in value by 8% for 4 years? A) $\times 0.92^4$ B) $\times 0.08^4$ C)$\times 1.08^4$ D) 0.92×4
Example 1	which multiplier can be used to find the value after an amount has decreased in value by 5% for 5 years? A) $\times 0.05^5$ B) 0.95×5 C) $\times 0.95^5$ D) $\times 1.05^5$
Example 2	the value of a laptop that initially cost \$1100, declines in value by 15% a year. if you wanted to calculate the value of the tablet at the end of 6 years, what number would replace the square below? $$1100 \times \square^6$$ A) 3500×0.85^3 B) 3500×1.5^3 C) $3500 \times 0.85 \times 3$ D) 3500×0.15^3
Example 3	a car depreciates in value by 15% each year. if a car was bought for \$3500, which of the following calculations would find the new value of the car after 3 years? A) 5 B) 4 C) 6 D) 8

Table 11: Three examples retrieved by kNN$^{\text{all}}$.

Question	which multiplier can be used to find the value after an amount has decreased in value by 8% for 4 years? A) $\times 0.92^4$ B) $\times 0.08^4$ C)$\times 1.08^4$ D) 0.92×4
Example 1	$(-\frac{1}{4})^{-2} \equiv \square$ A) $-\frac{1}{16}$ B) 16 C)$\frac{1}{16}$ D) -16
Example 2	what is the correct multiplier for decreasing by 30%? A) $\times 1.3$ B) $\times 0.7$ C) $\times 30$ D) $\times 0.3$
Example 3	dividing by 5 and then by 4 is the same as dividing by ... A) 1.25 B) 20 C) 1 D) 0.8

Table 12: Three examples retrieved by Random.

Question	which multiplier can be used to find the value after an amount has decreased in value by 8% for 4 years? A) $\times 0.92^4$ B) $\times 0.08^4$ C)$\times 1.08^4$ D) 0.92×4
Example 1	which is the correct multiplier for decreasing by 40%? A) $\times 1.4$ B) $\times 40$ C) $\times 0.6$ D) $\times 0.4$
Example 2	a company bought a van for \$12000. the van lost $\frac{1}{10}$ of its value each year. after how many years was the value of the van worth \$6377.29? A) 5 B) 7 C) 6 D) 8
Example 3	a company bought a van for \$15000. the van lost $\frac{1}{10}$ of its value each year. After how many years was the value of the van worth \$8857.29? A) 5 B) 4 C) 6 D) 8

Table 13: Three examples retrieved by kNN$_{\neg\mathcal{T}}^{\text{all}}$.