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

Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, Andrew Lan

*Manning College of Information and Computer Sciences, University of Massachusetts Amherst*

## 1. Background

- Multiple-choice questions (MCQs) are widely used for quick and accurate grading
- However, manually crafting high quality MCQs is demanding and labor-intensive
- Proposed tasks
  - Distractor generation
    - $g^{\text{dis}}(s, k, e_k) \rightarrow \hat{D}$
  - Feedback generation
    - $g^{\text{fb}}(s, d_i, k, e_k) \rightarrow f_i$

**Stem ($s$)**

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

**Key ($k$)** A) $\frac{7}{16}$

**Explanation ($e_k$)** LCM of 35 and 80 being 5, dividing both numerator and denominator by 5 results in 35/80 = 7/16.

**Distractor ($D$)** / **Feedback ($F$)**

B) $\frac{35}{80}$ — It appears that you have not simplified the fraction.

C) $\frac{7}{80}$ — You simplified the numerator while keeping the same denominator.

D) $\frac{80}{35}$ — You appear to have confused the denominator and numerator.

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

## 2. Methodology

### 2-1. In-Context Learning

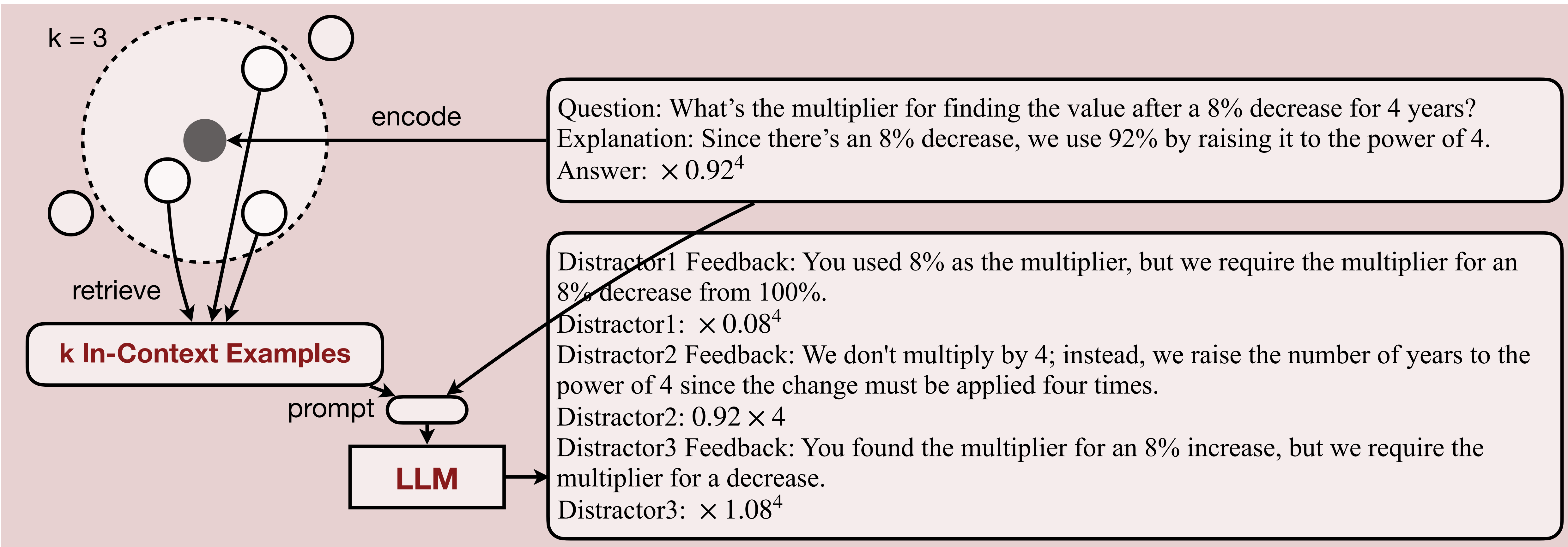


Figure 2. Overview of distractor generation with a math MCQ on "compound percentage decrease".

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

Table 1. Encoding strategies for retrieval and prompt formats used in kNN-based methods.

- Use LLM to estimate the functions $g_\phi^{\text{fb}}$ and $g_\phi^{\text{dis}}$, where $\phi$ are LLM parameters
- Utilize in-context learning with similar MCQs chosen by kNN algorithm as few-shot examples for LLM input

## 3. Results

### 3-1. Evaluation Metrics

**Distractor generation**

- Partial: at least one generated distractor matches a ground-truth one
- Exact: all generated distractors match ground-truth ones
- Proportional: the portion of generated distractors that match ground-truth ones

**Feedback generation**

- Answer adjustment: ask ChatGPT to use feedback to get correct answer - determines if a feedback message is helpful
- Distractor prediction: ask ChatGPT to predict the distractor given the feedback - determines if a feedback message explains why a distractor is incorrect

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

Table 2. 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** |
| $\text{kNN}^{\text{one}}$ | **33.70** | **42.28** | **43.64** | 46.64 | 18.26 |
| $\text{kNN}^{\text{one}}_{\neg T}$ | 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 3. Evaluation of generated feedback messages on reference-based and reference-free metrics.

## 4. Takeaways and Future work

- kNN prompting is an effective tool for distractor and feedback generation, but leaves room for improvement
- Effectiveness of reference-based metrics depends on generation method; reference-free metrics are less biased but have room for improvement
- Our initial exploration opens up many avenues for future work
  - Explore approaches for generation other than LLM prompting (ex: fine-tuning)
  - Use text encodings that closely align with student errors rather than semantic features
  - Conduct a human evaluation on the generated distractors and feedback messages