Beyond Fine-Tuning: In-Context Learning and Chain-of-Thought for Reasoned Distractor Generation

Elaf Alhazmi[1,3], **Quan Z. Sheng**[1], **Wei Emma Zhang**[2]
[1]School of Computing, Macquarie University, Australia
[2]School of Computer Sciences and Information Technology, Adelaide University, Australia
[3]College of Engineering and Computing in Al-Lith, Umm Al-Qura University, Saudi Arabia
elaf.alhazmi@hdr.mq.edu.au, eafhazmi@uqu.edu.sa,
michael.sheng@mq.edu.au, wei.e.zhang@adelaide.edu.au

Abstract

Distractor generation (DG) remains a labor-intensive task that still significantly depends on domain experts. The task focuses on generating plausible yet incorrect options, known as distractors, for multiple-choice questions. A reliable distractor must be contextually relevant to the question and able to mislead examinees through implicit reasoning when identifying the correct answer. While a recent method integrates fine-tuning pre-trained encoder-decoder models with contrastive learning to generate semantically relevant distractors for a given question-answer, it often fails to capture the underlying reasoning process that experts utilize when selecting distractors in benchmarks. In this paper, we explore large language models (LLMs) reasoning for DG through in-context learning with unsupervised semantic retrieval for selecting few-shot examples. We design a rationale-augmented DG framework that jointly generates distractors and their rationales for a given question-answer. Extensive experiments on six benchmarks, with varying average distractor lengths and domains, demonstrate that prompting LLMs with few-shot examples substantially improves the performance compared to recent DG models. It outperforms recent approaches and achieves state-of-the-art results in generating reasoned distractors that align with human-labeled benchmarks.

1 Introduction

Distractor generation (DG) refers to the automated generation of misleading options in multiple-choice questions (MCQs) (Dong et al., 2022; Alhazmi et al., 2024). In practical assessment, the presence of well-selected *distractors* ensures that examinees can thoughtfully distinguish the correct answer from the wrong options. Unfortunately, selecting such distractors remains a challenging, labor-intensive, and expert-dominant task (Ch and Saha, 2018; Kurdi et al., 2020; Das et al., 2021).

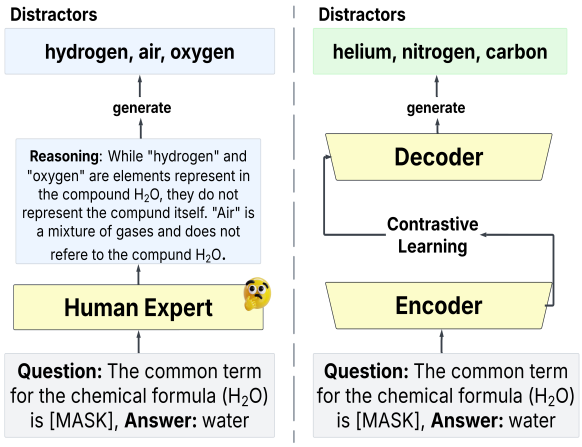


Figure 1: Generated distractors by human reasoning and contrastive-based pre-trained encoder-decoder models.

The DG field has progressed considerably in methodologies, especially with transformer-based models, in textual (Xie et al., 2018), mathematical (Feng et al., 2024), and multi-modal (Yagcioglu et al., 2018) aspects. In text-based DG, an early work (Ren and Zhu, 2021) proposed a *candidate generation and selection framework* that generates a set of distractors and selects the top candidates based on embedding models. Then, several works explored DG as a sequence-to-sequence (Seq2Seq) task, known as the *Text2Text* architecture (Wang et al., 2023), by fine-tuning pre-trained encoder-decoder models without a selection method.

Although Alhazmi et al. (2025) integrates contrastive learning as semantic representation learning in Text2Text models to emphasize in-context DG, they still struggle to align the reasoning behind generated distractors with human-labeled distractors in benchmarks, as indicated in Figure 1. For a question (the chemical formula for H_2O is ...), a recent contrastive approach may generate valid distractors (carbon, nitrogen, helium) that are in-context, but they are often misaligned with human-label reasoning, e.g., (H_2O is a compound made of two hydrogen atoms and one oxygen atom). Thus, *hydrogen* and *oxygen* may refer to more suitable

Proceedings of the 19th International Natural Language Generation Conference: Long Papers, pages 160–190
October 17-21, 2026 ©2026 Association for Computational Linguistics

pedagogically meaningful distractors. In practice, experts rarely select distractors in isolation; they rely on explicit reasoning or rationale (i.e., step-by-step justification) to distinguish the answer from incorrect options. Reasoning-aware distractors play a critical role in effectively challenging examinees. The recent method (Alhazmi et al., 2025) utilizes contrastive learning to improve semantic plausibility, but it fails to model such reasoning in DG.

The emerging LLMs (Brown et al., 2020) have transformed downstream NLP tasks with *in-context learning* (ICL) (Liu et al., 2023; Dong et al., 2024). ICL allows LLMs to learn from a limited number of few-shot examples with a given input (x) and output (y) to generate a desired target for a given test input without fine-tuning or parameter updates. ICL is remarkably effective for a wide range of reasoning tasks, including mathematical problem solving (Brown et al., 2020; Wei et al., 2022), commonsense reasoning (Kojima et al., 2022; Zhang et al., 2024b), and question answering (Li et al., 2023; Patidar et al., 2024).

While recent works utilize ICL in math DG (McNichols et al., 2023; Feng et al., 2024) and multilingual DG benchmark (Bitew et al., 2023), the state-of-the-art (SOTA) text-based DG works still rely on supervised fine-tuning models (Chiang et al., 2022; Wang et al., 2023; Taslimipoor et al., 2024; Yu et al., 2024; Alhazmi et al., 2025).

ICL framework is sensitive to example selection (Zebaze et al., 2025) when using random or semantic-embedding retrieval. While conventional DG methods formulated the task as mapping a given question-answer to a set of distractors, recent work has also shown that rationales, or step-by-step chain-of-thought (COT) (Wei et al., 2022), can improve the performance in multi-step reasoning tasks (Kojima et al., 2022). We utilize COT to generate rationales for each question-answer and distractors in the training set and integrate ICL to generate distractors and rationales for a given question-answer test input. We demonstrate that integrating LLM reasoning to generate distractors can achieve more accurate results than conventional DG approaches.

The main contributions are (i) exploring ICL and the rationale-augmented DG framework for generating reasoned distractors; (ii) benchmarking our approach with SOTA DG models using both automatic and manual evaluation metrics; and (iii) achieving SOTA results in multiple benchmarks, including word-level and long-level distractors in science, general knowledge, and medical domains.

2 Related Works

2.1 Distractor Generation (DG)

Transformer-based pre-trained language models (PLMs) (Vaswani et al., 2017) have significantly advanced DG methods across multi-modal (Luo et al., 2024), mathematical (Fernandez et al., 2024), and text-based (Qu et al., 2024) applications, including question answering (Liang et al., 2018) and reading comprehension (Maurya and Desarkar, 2020).

Text-based approaches in MCQs include two main directions: candidate generation and selection framework (CSG-DS) (Ren and Zhu, 2021) and Text2Text architecture (Wang et al., 2023). The former utilizes a candidate generation step before selecting distractors with embedding models (Taslimipoor et al., 2024), while the latter utilizes fine-tuning pre-trained encoder-decoder models to generate distractors without a selection method.

Recently, the Text2Text DG approach (Yu et al., 2024) has shown stronger performance compared to CSG-DS frameworks (Chiang et al., 2022). Alhazmi et al. (2025) introduced a contrastive-learning objective into Text2Text models to enhance the embedding representation learning between input and output in pre-trained encoder-decoder models to generate in-context distractors.

Despite these advances, the generated distractors often remain misaligned with human-labeled distractors. Human experts usually apply reasoning when selecting distractors for a given question and answer, a process not fully captured by recent fine-tuned pre-trained encoder-decoder models. ICL has emerged as a significant paradigm that leverages LLMs reasoning to generate distractors by prompting the model in zero-shot or few-shot learning. While prompting paradigms have been explored in text-based multilingual (Bitew et al., 2023) and mathematical DG tasks (Feng et al., 2024), fine-tuning PLMs remains more common in text-based DG benchmarks (Wang et al., 2023; Taslimipoor et al., 2024; Yu et al., 2024; Alhazmi et al., 2025).

2.2 In-Context Learning (ICL)

LLMs demonstrate strong ICL (Dong et al., 2024) capabilities, enabling them to solve unseen tasks by conditioning on a few-shot examples without parameter updates. ICL shows significant improvements in mathematical problem solving (Brown et al., 2020; Wei et al., 2022), commonsense reasoning (Kojima et al., 2022; Zhang et al., 2024b), and question answering (Li et al., 2023) tasks.

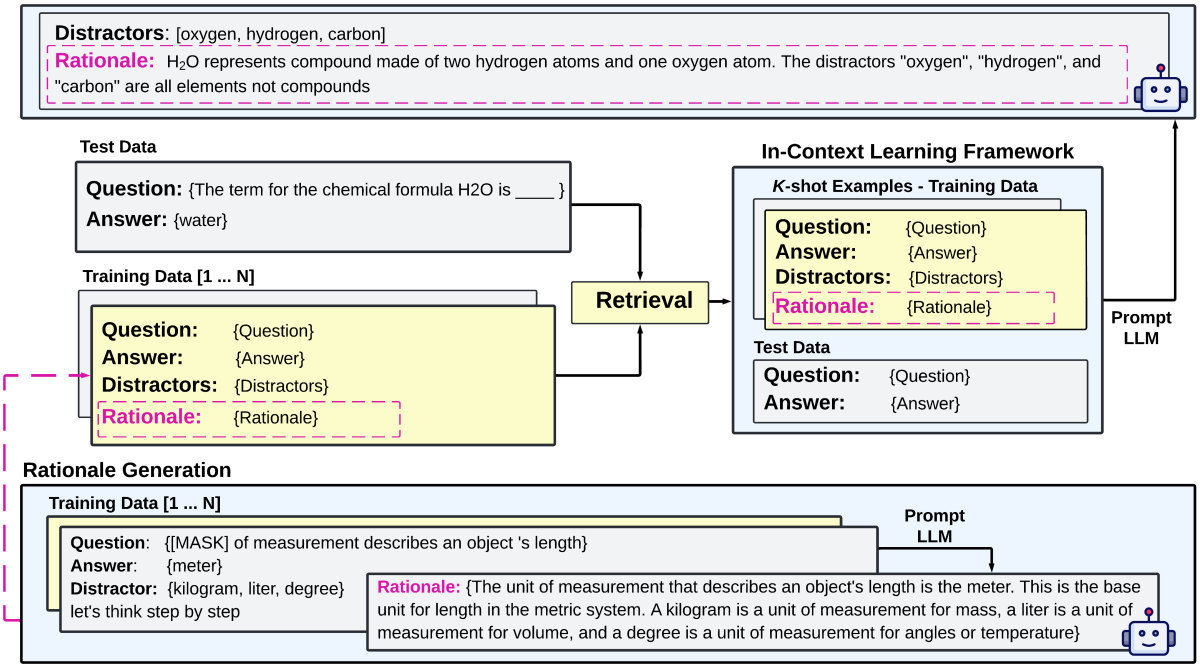


Figure 2: The in-context learning framework by large language model and chain-of-thought rationale generation for the training dataset. The dashed arrows indicate the optional augmented-rationale in-context learning framework.

Initially, ICL performance is highly sensitive to the selection of examples. Liu et al. (2022) proposed k-nearest-neighbor (k-NN) unsupervised retrieval for selecting examples, while Rubin et al. (2022) introduced a two-stage framework that trains a dense retrieval to identify optimal examples. Similarly, in DG domains, while Feng et al. (2024) applied k-NN retrieval for mathematical DG, Bitew et al. (2023) trained supervised bi-encoder retrieval (Bitew et al., 2022) for multilingual DG. We extend the use of unsupervised k-NN retrieval for selecting in-context examples to text-based DG benchmarks.

Chain-of-thought (COT) (Wei et al., 2022) encourages LLMs to generate explicit, step-by-step reasoning traces, known as rationales (Ling et al., 2017), before predicting a final answer. Kojima et al. (2022) outlined strong performance of zero-shot COT by prompting "Let's think step by step". Integrating rationales can serve as auxiliary signals for improving ICL (Wang et al., 2022; Huang et al., 2023) or distilling reasoning (Tian et al., 2025) into smaller pre-trained models. We explore COT rationales as reasoning-based augmentation for DG.

3 Methodology

Section 3.1 formalizes the DG task, and Section 3.2 introduces the ICL framework. The framework incorporates unsupervised similarity-based retrieval, described in Section 3.2.1, and COT rationale-based augmentation, presented in Section 3.2.2.

3.1 Task Definition

The DG task is defined as mapping ($x \mapsto y$), where x denotes the input question $\mathbf{Q}$ and its answer $\mathbf{A}$, and y denotes the target sequence of distractors $\mathbf{D} = \{d_1, d_2, \ldots, d_N\}$ with $N > 0$. This conditional text generation problem is defined as:

$$p_\phi(y \mid x) = \prod_{t=1}^{N} p_\phi(y_t \mid y_{<t}, x) \quad (1)$$

where ϕ are fixed large language model parameters, y_t represents the sequence of letters in the t-th distractor, and $y_{<t}$ denotes the sequences of all distractors generated before y_t.

3.2 In-Context Learning Framework

ICL refers to the ability of large models to generate desired outputs from few-shot examples without updating model parameters. It extends conditional text generation using a context C containing k labeled examples, with the probability defined as:

$$p_\phi(y \mid C, x) = \prod_{t=1}^{N} p_\phi(y_t \mid y_{<t}, C, x) \quad (2)$$

Given a set of training example pairs $e_i = \{(x_i, y_i)\}_{i=1}^{k}$, large language models predict distractors $\hat{y}$ based on the concatenation $\oplus$ of given examples $(e_1 \oplus \ldots \oplus e_k)$ as C and input x_{test}.

$$\hat{y} = LLM(e_1 \oplus e_2 \oplus \ldots \oplus e_k \oplus x_{test}) \quad (3)$$

3.2.1 Unsupervised Retrieval

To select few-shot examples from the training set as illustrated in Figure 2, we utilize a k-nearest-neighbor (k-NN) retrieval (Liu et al., 2022; Feng et al., 2024). All training instances x_i are encoded using a sentence encoder $\mu_\theta(\cdot)$ to obtain their vector representations. Given x_{test}, we compute its embedding v_{test} and identify the top-k most similar training instances using a similarity metric (e.g., cosine similarity), as outlined in Eq. 4. The retrieved neighbors $x_1, \ldots, x_k$ are ordered by descending similarity, i.e., $s(v_i, v_{\text{test}}) \geq s(v_j, v_{\text{test}})$ for $i < j$.

$$\mathrm{s}(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i^\top \mathbf{v}_j}{\|\mathbf{v}_i\|_2 \|\mathbf{v}_j\|_2} \quad (4)$$

Each retrieved x_i is concatenated with its corresponding target y_i to construct the context C.

3.2.2 Chain-of-Thought Augmentation

We introduce rationale augmentation using a zero-shot COT. We prompt a model to generate a rationale r for a given training question-answer x and its distractors y as illustrated in Figure 2.

Then, given a set of training examples $e_{r_i} = \{(x_i, y_i, r_i)\}_{i=1}^{k}$, the large language model predicts $\hat{z}$, consisting of distractors $\hat{y}$ and rationale $\hat{r}$, based on the concatenation $\oplus$ of given retrieved examples $(e_{r_1} \oplus \cdots \oplus e_{r_k})$ as C and input x_{test}.

$$\hat{z} = LLM(e_{r_1} \oplus e_{r_2} \oplus \cdots \oplus e_{r_k} \oplus x_{test}) \quad (5)$$

Algorithm 1 in Appendix A describes the k-NN-based retrieval and its optional COT augmentation.

4 Experiments

4.1 Datasets

We conduct the experiments on six datasets, including science (SciQ (Welbl et al., 2017), MCQL (Liang et al., 2018), ARC (Clark et al., 2018)), general knowledge (MCQ (Ren and Zhu, 2021)), and medical domain (MedQA (Jin et al., 2021)) as detailed in Appendix B. Each dataset contains one answer and three to four distractors, with different token lengths. The statistics are outlined in Table 1.

Datasets	Train	Valid	Test	All
MCQ	1,856	465	259	2,580
SciQ	11,700	1,000	1,000	13,700
MCQL	6,362	600	600	7,562
ARC-Easy	2,251	570	2,376	5,197
ARC-Challenge	1,119	299	1,172	2,590
MedQA	10,178	1,272	1,273	12,723

Table 1: Statistics of the Datasets.

4.2 Baselines Models

CSG-DS: We fine-tune the T5 (Raffel et al., 2020) model as a candidate-generation to generate ten distractors; then we select the top three with beam search (Gao et al., 2019). We also use a fine-tuned T5 model and GPT-3 (Brown et al., 2020), with few-shot k-NN retrieval, as illustrated in Appendix C, for candidate generation, followed by a clustering-based selection method (Taslimipoor et al., 2024).

Text2Text Architecture: We train T5 as Seq2Seq model. We explore multi-task learning (Wang et al., 2023) and contrastive learning (Alhazmi et al., 2025) with InfoNCE (Oord et al., 2018) objective.

Instruction-tuning: We fine-tune the TinyLlama (Zhang et al., 2024a) model under an instruction-tuning paradigm to evaluate whether a lightweight 1.1B model can effectively learn the DG task.

Prompting: We utilize zero-shot prompting and random retrieval for ICL and COT augmentation.

4.3 Evaluation Metrics

We use automatic ranking-based metrics (Wang et al., 2023; Alhazmi et al., 2025) to assess how well models retrieve relevant distractors in the top-k positions. *Order-unaware* metrics include F1 score (F1@K), precision (P@K), and recall (R@K), while *order-aware* metrics include mean reciprocal rank (MRR@K) and normalized discounted cumulative gain (NDCG@K). We also report percentage-based indicators that quantify answer repetition (I_a) and distractor repetition (I_d) to assess, respectively, the extent to which generated distractors repeat the correct answer and the degree of redundancy among the generated distractors.

We also utilize human evaluation metrics, including *relevance*, which measures if the distractors are in-context to a given input; *difficulty*, which captures the ability to mislead from the correct answer; and *fluency*, which ensures the distractors are valid and not duplicated. We select ten questions per dataset and ten evaluators: five academics with over two years of experience to assess fifty science and general-knowledge questions and five medical doctors with over two years of clinical experience to evaluate the remaining ten medical questions. All evaluators independently assess the same set of questions using a five-point rating scale ranging from 1 (strongly irrelevant) to 5 (strongly relevant). In addition, we measure *accuracy* using question-answering models with the generated distractors.

Dataset	Approach	Model	P@1	R@1	F1@3	MRR	NDCG@3	I_a (%)	I_d (%)
MCQ	CSG-DS	T5(beam)	17.37	5.79	13.38	24.20	30.11	36.29	0.39
		T5(clustering)	11.58	3.86	7.72	16.47	20.96	18.15	0.00
		GPT-3(clustering)	14.29	4.76	8.49	18.92	22.95	0.39	0.00
	Text2Text	T5(base)	14.29	4.76	10.81	20.14	24.68	17.76	18.15
		T5(multi-task)	15.06	5.02	12.23	19.95	23.54	19.31	15.83
		T5(contrast)	22.78	7.59	15.70	28.57	32.33	14.67	10.81
	Instruction Tuning	TinyLlama	22.78	7.59	15.06	28.44	33.16	3.09	1.54
	Prompting	GPT-3(zero-shot)	25.87	8.62	17.63	32.56	38.33	0.00	0.00
		GPT-3(few-shot-random)	25.87	8.62	18.28	33.72	40.68	0.00	0.00
		GPT-3(few-shot-k-NN)	29.73	9.91	**19.69**	**36.62**	**42.37**	0.00	0.00
		GPT-3(few-shot-COT-random)	28.96	9.65	19.05	36.04	42.08	0.00	0.00
		GPT-3(few-shot-COT-k-NN)	**30.50**	**10.17**	19.43	36.29	40.60	0.00	0.00
SciQ	CSG-DS	T5(beam)	20.30	6.77	14.33	26.20	31.30	29.10	0.20
		T5(clustering)	10.50	3.50	6.30	14.35	17.95	22.3	0.00
		GPT-3(clustering)	10.80	3.60	5.20	13.33	15.73	0.30	0.00
	Text2Text	T5(base)	18.90	6.30	13.77	23.23	26.66	23.30	12.80
		T5(multi-task)	21.20	7.07	14.27	25.47	28.52	18.90	11.10
		T5(contrast)	25.00	8.33	**17.73**	**31.42**	**36.68**	8.80	6.50
	Instruction Tuning	TinyLlama	20.50	6.83	15.40	27.07	32.73	2.40	1.90
	Prompting	GPT-3(zero-shot)	24.40	8.13	15.47	29.30	33.45	0.10	0.30
		GPT-3(few-shot-random)	25.10	8.37	16.13	30.17	34.57	0.20	0.10
		GPT-3(few-shot-k-NN)	24.90	8.30	16.53	30.45	35.24	0.10	0.00
		GPT-3(few-shot-COT-random)	24.50	8.17	15.70	29.53	33.82	0.20	0.00
		GPT-3(few-shot-COT-k-NN)	**25.50**	**8.50**	17.03	31.08	35.89	0.10	0.00
MCQL	CSG-DS	T5(beam)	20.33	6.78	14.50	25.42	29.65	37.17	1.83
		T5(clustering)	12.17	4.06	5.56	15.69	19.04	20.5	0.00
		GPT-3(clustering)	11.17	3.72	4.28	13.92	16.55	0.17	0.00
	Text2Text	T5(base)	17.83	5.94	10.78	20.07	21.68	28.83	4.83
		T5(multi-task)	19.33	6.44	11.50	22.14	24.45	21.5	4.17
		T5(contrast)	23.00	7.67	13.67	26.42	29.13	16.67	3.67
	Instruction Tuning	TinyLlama	28.83	9.61	20.56	33.33	37.03	2.67	1.17
	Prompting	GPT-3(zero-shot)	24.17	8.06	13.78	27.03	29.37	0.00	0.17
		GPT-3(few-shot-random)	30.17	10.06	19.06	34.36	37.98	0.00	0.50
		GPT-3(few-shot-k-NN)	35.33	11.78	**24.44**	39.61	42.81	0.00	0.33
		GPT-3(few-shot-COT-random)	32.00	10.67	19.44	35.58	38.51	0.00	0.17
		GPT-3(few-shot-COT-k-NN)	**36.17**	**12.06**	23.61	**39.94**	**42.90**	0.00	0.00
ARC-Easy	CSG-DS	T5(beam)	12.75	4.28	9.56	16.88	20.45	33.63	0.63
		T5(clustering)	7.95	2.68	4.50	10.67	13.16	16.16	0.00
		GPT-3(clustering)	9.81	3.28	5.55	13.06	16.08	0.08	0.00
	Text2Text	T5(base)	11.24	3.77	7.38	14.05	16.30	13.97	7.70
		T5(multi-task)	10.06	3.38	6.97	12.73	14.9	13.51	5.77
		T5(contrast)	13.85	4.64	10.39	17.53	20.45	6.69	3.41
	Instruction Tuning	TinyLlama	15.03	5.03	12.73	19.76	23.66	1.98	1.05
	Prompting	GPT-3(zero-shot)	20.75	6.94	15.28	25.55	29.32	0.00	0.00
		GPT-3(few-shot-random)	23.19	7.75	17.74	28.37	32.41	0.00	0.04
		GPT-3(few-shot-k-NN)	**24.24**	**8.10**	**18.10**	**29.08**	**32.82**	0.00	0.17
		GPT-3(few-shot-COT-random)	23.70	7.92	17.77	28.67	32.44	0.00	0.00
		GPT-3(few-shot-COT-k-NN)	23.70	7.92	17.95	28.36	31.74	0.00	0.00
ARC-Challenge	CSG-DS	T5(beam)	7.85	2.67	6.50	11.46	14.71	31.83	0.26
		T5(clustering)	5.46	1.86	2.23	7.35	9.18	12.97	0.00
		GPT-3(clustering)	8.70	2.92	4.00	10.75	12.68	0.00	0.00
	Text2Text	T5(base)	7.17	2.45	3.68	8.79	10.14	19.20	2.90
		T5(multi-task)	5.38	1.85	3.63	7.21	8.81	18.43	2.39
		T5(contrast)	9.13	3.09	5.47	11.26	13.06	9.64	3.07
	Instruction Tuning	TinyLlama	12.37	4.15	8.60	15.15	17.55	1.37	1.11
	Prompting	GPT-3(zero-shot)	17.41	5.85	11.96	20.31	22.53	0.09	0.09
		GPT-3(few-shot-random)	**21.08**	**7.07**	14.96	**24.82**	**27.75**	0.26	0.17
		GPT-3(few-shot-k-NN)	20.48	6.87	15.05	24.06	26.80	0.17	0.09
		GPT-3(few-shot-COT-random)	20.31	6.81	**15.29**	24.23	27.23	0.34	0.09
		GPT-3(few-shot-COT-k-NN)	19.88	6.67	14.60	23.25	25.73	0.17	0.09
MedQA	CSG-DS	T5(beam)	6.83	1.71	4.44	9.54	12.00	21.05	0.55
		T5(clustring)	4.24	1.06	1.89	5.85	7.38	10.05	0.00
		GPT-3(clustring)	9.82	2.45	5.12	12.80	15.47	0.08	0.00
	Text2Text	T5(base)	5.42	1.36	2.51	6.39	7.10	13.28	0.08
		T5(multi-task)	4.95	1.24	2.60	6.11	7.00	11.31	0.08
		T5(contrast)	6.83	1.71	3.50	8.42	9.57	9.19	0.31
	Instruction Tuning	TinyLlama	11.86	2.97	7.90	15.95	19.60	1.02	0.55
	Prompting	GPT-3(zero-shot)	12.49	3.12	7.77	16.17	19.42	0.00	0.00
		GPT-3(few-shot-random)	17.44	4.36	11.90	23.10	26.50	0.00	0.00
		GPT-3(few-shot-k-NN)	20.90	5.22	**15.08**	**27.44**	**30.52**	0.00	0.08
		GPT-3(few-shot-COT-random)	17.20	4.30	11.74	22.52	25.63	0.00	0.00
		GPT-3(few-shot-COT-k-NN)	**21.05**	**5.26**	14.66	27.39	30.34	0.00	0.00

Table 2: Automatic evaluation results. The standard ICL framework is denoted as (few-shot), while COT augmentation is referred to as (few-shot-COT). The best ranking-based scores are outlined in bold.

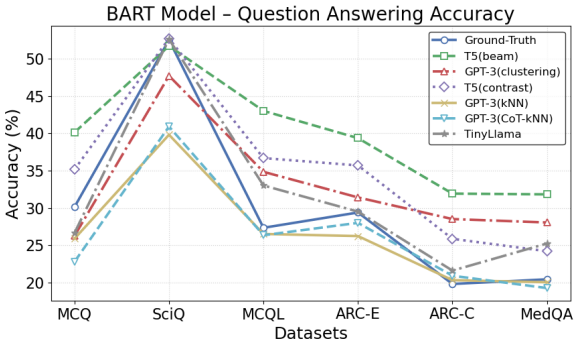


Figure 3: BART – Question Answering Accuracy

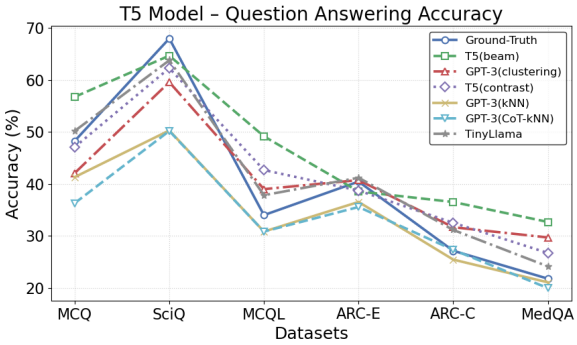


Figure 4: T5 – Question Answering Accuracy

Model	Relevance	Difficulty	Fluency
GPT-3 (COT-k-NN)	3.83	3.33	4.65
GPT-3 (k-NN)	**4.22**	**3.68**	**4.79**
TinyLlama	3.46	3.13	4.49
GPT-3 (clustering)	3.60	3.09	4.58
T5 (beam)	3.06	2.69	3.90
T5 (contrast)	2.66	2.32	3.61
Ground-truth	3.86	3.46	4.64

Table 3: Human evaluation results.

4.4 Implementation Details

We use GPT-3.5-turbo (Brown et al., 2020) for ICL with $k = 5$, MPNet sentence encoder (Reimers and Gurevych, 2019a) for vector representations, and cosine similarity as base similarity metric. Prompt templates and all model configurations are detailed in Appendix C and Appendix D, respectively[1].

4.5 Evaluation Results

4.5.1 Automatic Evaluation Results

Table 2 provides a summary of the automatic evaluation results across all datasets. ICL, with LLMs such as GPT-3, substantially improves DG performance compared to recent SOTA approaches.

From the table, we can observe that ICL significantly outperforms CSG-DS framework results across all datasets and metrics. It increases the beam-search candidate-generation F1@3 score from 13.38 to 19.69 on MCQ, 6.50 to 15.29 on ARC-Challenge, and 4.44 to 15.08 on MedQA. This result underscores the strength of ICL for DG with longer distractors and diverse domains.

ICL also outperforms Text2Text models. While the contrastive-based model provides steady improvements over base fine-tuning and multi-task learning in all datasets and still achieves the highest F1@3 score on SciQ, it remains less competitive with longer distractor tokens and diverse domains, such as the medical field. In fact, ICL shows consistently stronger performance than recent SOTA contrastive-based models in the DG task. Its F1@3 score rises from 15.70 to 19.69 on MCQ and 13.67 to 24.44 on MCQL, with even larger improvements on datasets with longer distractor lengths: 10.39 to 18.10 on ARC-Easy, 5.47 to 15.29 on ARC-Challenge, and 3.50 to 15.08 on MedQA.

Notably, ICL with LLMs provides substantial improvements over fine-tuning small language models (SLMs), including TinyLlama, which shows in some cases stronger automatic results than recent approaches (e.g., CSG-DS and Text2Text), particularly on datasets with longer distractor tokens. It raises the F1@3 score from 12.73 to 18.10 on ARC-Easy, 8.60 to 15.29 on ARC-Challenge, and 7.90 to 15.08 on MedQA. This highlights the reasoning ability of large models to address the DG task effectively (i.e., closer to human reasoning), without fine-tuning or any parameter updates.

ICL presents stronger results when combined with unsupervised k-NN retrieval for selecting few-shot examples. It significantly enhances the performance across several datasets. Compared to random retrieval, k-NN selection boosts the F1@3 score from 19.06 to 24.44 on MCQL and 11.90 to 15.08 on MedQA, while yielding modest improvements from 18.28 to 19.69 on MCQ, 16.13 to 16.53 on SciQ, 17.74 to 18.10 on ARC-Easy, and 14.96 to 15.05 on ARC-Challenge. This demonstrates that embedding semantically similar examples enables LLMs to generate distractors that closely align with human-labeled distractors and reasoning.

Although COT augmentation does not consistently improve all automatic ranking-based metrics compared to the base ICL framework, it achieves higher P@1 and R@1 on datasets such as MCQ, SciQ, MCQL, and MedQA. It also enhances the interpretability of the generated distractors by revealing the reasoning behind predicted options and achieves the second-best performance in multiple datasets (e.g., MCQ, MCQL, ARC, and MedQA) compared to recent fine-tuning models.

[1] https://github.com/reasoning-dis/icl-COT

Dataset	Model	P@1	R@1	F1@3	MRR	NDCG@3	I_a (%)	I_d (%)
MCQ	GPT-3(k-NN)	29.73	9.91	19.69	36.62	42.37	0.00	0.00
	GPT-3(k-NN)/wo-a	24.32	8.11	15.57	29.99	34.71	0.39	1.16
	GPT-3(k-NN)/MiniLM	29.73	9.91	19.82	35.78	40.42	0.00	0.77
	Mistral(k-NN)	27.41	9.14	18.79	34.3	40.15	0.00	0.00
	Qwen(k-NN)	28.19	9.40	18.28	34.17	39.07	0.00	0.00
	GPT-3(k-NN)/k-30	**30.12**	**10.04**	**21.62**	**38.42**	**45.29**	0.00	0.39
SciQ	GPT-3(k-NN)	24.90	8.30	16.53	30.45	35.24	0.10	0.00
	GPT-3(k-NN)/wo-a	19.50	6.50	13.47	24.33	28.46	0.30	0.80
	GPT-3(k-NN)/MiniLM	24.90	8.30	16.77	30.75	35.87	0.00	0.10
	Mistral(k-NN)	20.10	6.70	15.50	26.80	32.62	0.10	0.00
	Qwen(k-NN)	17.40	5.80	12.87	22.97	27.99	0.00	0.00
	GPT-3(k-NN)/k-30	**25.90**	**8.63**	**18.10**	**32.18**	**37.61**	0.10	0.10
MCQL	GPT-3(k-NN)	35.33	11.78	24.44	39.61	42.81	0.00	0.33
	GPT-3(k-NN)/wo-a	23.50	7.83	16.56	27.25	30.33	1.17	3.33
	GPT-3(k-NN)/MiniLM	37.50	12.50	24.28	40.92	43.40	0.00	0.17
	Mistral(k-NN)	27.67	9.22	18.89	32.14	35.93	0.00	0.17
	Qwen(k-NN)	32.67	10.89	18.72	35.31	37.47	0.00	0.00
	GPT-3(k-NN)/k-30	**38.17**	**12.72**	**26.06**	**42.17**	**45.19**	0.17	0.33
ARC-Easy	GPT-3(k-NN)	24.24	8.10	18.10	29.08	32.82	0.00	0.17
	GPT-3(k-NN)/wo-a	15.45	5.16	12.01	18.82	21.41	0.13	0.84
	GPT-3(k-NN)/MiniLM	23.74	7.93	17.95	28.70	32.51	0.00	0.04
	Mistral(k-NN)	17.93	6.00	14.04	22.46	26.14	0.17	0.04
	Qwen(k-NN)	17.63	5.89	13.65	22.29	26.11	0.00	0.04
	GPT-3(k-NN)/k-30	**25.93**	**8.66**	**19.38**	**30.77**	**34.37**	0.00	0.25
ARC-Challenge	GPT-3(k-NN)	20.48	6.87	15.05	24.06	26.80	0.17	0.09
	GPT-3(k-NN)/wo-a	13.91	4.66	10.46	16.54	18.48	0.51	1.11
	GPT-3(k-NN)/MiniLM	19.80	6.64	14.67	23.69	26.85	0.26	0.17
	Mistral(k-NN)	14.59	4.91	10.26	17.09	19.20	0.68	0.17
	Qwen(k-NN)	14.76	4.96	11.13	18.09	20.77	0.09	0.09
	GPT-3(k-NN)/k-30	**21.25**	**7.13**	**16.01**	**25.46**	**28.77**	0.17	0.17
MedQA	GPT-3(k-NN)	20.90	5.22	15.08	27.44	30.52	0.00	0.08
	GPT-3(k-NN)/wo-a	4.87	1.22	3.61	6.99	8.66	0.00	0.08
	GPT-3(k-NN)/MiniLM	19.80	4.95	13.85	25.65	28.36	0.08	0.08
	Mistral(k-NN)	14.61	3.65	10.35	19.48	23.46	0.94	0.00
	Qwen(k-NN)	13.35	3.34	8.98	17.78	21.67	0.00	0.00
	GPT-3(k-NN)/k-30	**22.00**	**5.50**	**15.42**	**28.45**	**31.52**	0.00	0.16

Table 4: Ablation experimental results. wo-a means without answer and k-30 indicates number of examples.

In terms of performance, automatic ranking-based metrics primarily reward exact matches to the ground-truth. It must also be noted that a given question–answer pair may have multiple valid, well-reasoned distractors that do not exactly match the annotated ground truth. Consequently, although the COT variants of the ICL show slight decreases in some automatic metrics, they consistently continue to outperform recent fine-tuning models. This suggests that the observed differences in automatic metrics between ICL and COT variants do not necessarily indicate a decline in distractor quality.

Overall, ICL consistently outperforms zero-shot prompting and is far less prone to generating the correct answer or repeating distractors. For example, CSG-DS with beam search shows the highest rates of answer leakage and distractor repetition, followed by standard Text2Text models. Although contrastive-based Text2Text model reduces these errors as indicated by the I_a and I_d indicators, both Text2Text and CSG-DS methods still fall short of the reliability (i.e., distractors match the correct answer) needed for high-quality distractors.

4.5.2 Human Evaluation Results

Table 3 reports the average human evaluation results for the DG models across each human evaluation metric, including relevance, difficulty, and fluency. While the ICL framework with k-NN retrieval achieves the highest scores across all human metrics, COT augmentation exhibits slightly lower relevance and difficulty scores compared to ground truth, but COT overall evaluation remains closely aligned with ground truth. This indicates the effectiveness of LLMs reasoning in DG task.

Recent approaches show strong capability in generating fluent and relevant distractors. Instruction-tuned models and GPT-3–based clustering achieve higher scores than contrastive models and T5 beam search because the latter models remain vulnerable to answer leakage and option repetitions, thereby reducing their quality during the evaluation.

We also measure inter-annotator agreement for the human evaluation scores using Krippendorff's α. The agreement scores are 0.357 for relevance,

Model	Example (1)	Example (2)	Example (3)
Question [Answer]	Main source of carbohydrates for various animals is [starch]	The common term for the chemical formula H_2O is [water]	[humans] has the greatest ability to learn.
Ground-truth	[glycogen, agar, pectin]	[hydrogen, air, oxygen]	[apes, reptiles, rats]
T5(base)	[glucose, cellulose, starch]	[air, light, air]	[animals, plants, animals]
T5(multi-task)	[glucose, cellulose, starch]	[air, ice, gas]	[animals, animals, plants]
T5(contrast)	[proteins, fats, proteins]	[helium, nitrogen, carbon]	[animals, plants, animals]
T5(beam)	[starch, cellulose, glucose]	[carbon, oxygen, gas]	[animals, mammals, dinosaurs]
T5(clustering)	[starch, cellulose, sugar]	[carbon, air, ions]	[animals]
GPT-3(clustering)	[fat, fiber, maltose]	[H2O2, Hydroxide, Hydronium]	[elephants, monkeys, cats]
TinyLlama	[proteins, cellulose, sap]	[gas, liquid, heat]	[chickens, cars, fish]
Mistral(k-NN)	[cellulose, glycogen, glucose]	[hydrogen peroxide, steam, ice]	[dogs, dolphins, parrots]
Qwen(k-NN)	[cellulose, lactose, sucrose]	[helium, oxygen, carbon dioxide]	[dogs, chimpanzees, parrots]
GPT-3(random)	[protein, cellulose, fat]	[carbon dioxide, oxygen, hydrogen peroxide]	[plants, animals, machines]
GPT-3(k-NN)	[cellulose, glycogen, fructose]	[hydrogen peroxide, carbon dioxide, ammonia]	[birds, dolphins, chimpanzees]
GPT-3(COT-random)	[cellulose, glucose, glycogen]	[hydrogen peroxide, carbon dioxide, sodium chloride]	[plants, animals, machines]
GPT-3(COT-k-NN)	[cellulose, glycogen, fructose]	[oxygen, hydrogen, carbon]	[dogs, birds, dolphins]

Table 5: Examples of DG models. Underline indicates the generated distractor matches the correct answer.

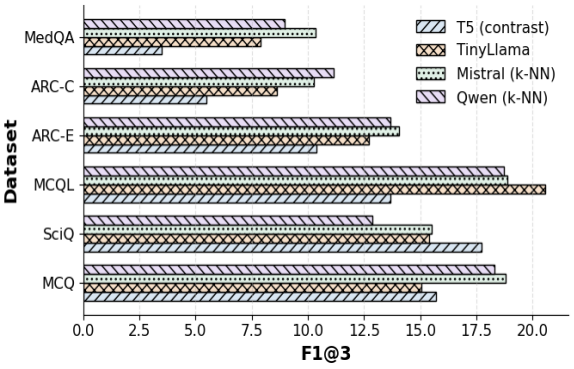


Figure 5: F1@3 score for both Mistral (k-NN) and Qwen (k-NN) with recent fine-tuning models.

0.262 for difficulty, and 0.568 for fluency. The higher agreement on fluency indicates that distractor quality was judged more consistently than relevance and difficulty, whereas relevance and difficulty depend more on evaluators' perceptions.

Furthermore, to complement the human evaluation, we measure the accuracy of predicting the correct answer given the question, answer, and distractors generated by different DG approaches. As shown in Figure 3 and Figure 4, ICL and its COT-augmented variants achieve the lowest accuracy across all datasets for both the BART and T5 models. In addition, ground-truth distractors yield low accuracy on several datasets. These results suggest that ICL generates distractors that more closely reflect human expert reasoning compared with recent DG approaches. Detailed accuracy results for both models are provided in Table 6 in Appendix D.

4.6 Analysis of Different Factors

Table 4 summarizes the analysis of several factors in ICL with unsupervised retrieval. We examine the impact of removing the answer from the few-shot examples (i.e., only the question and its distractors). The results show a substantial performance decline. The F1@3 score decreases from 19.69 to 15.57 on MCQ and drops more sharply from 15.08 to 3.61 on MedQA. These results highlight the importance of providing full-context k-shot examples, including the question, answer, and associated distractors, to support effective reasoning in ICL framework.

Then, we observe that the MPNet sentence encoder may outperform MiniLM (Wang et al., 2020) with k-NN retrieval for DG. The MPNet encoder indicates better automatic metrics, particularly on challenging datasets such as ARC and MedQA. The F1@3 score increases from 14.67 to 15.05 on ARC-Challenge and 13.85 to 15.08 on MedQA.

We also report the performance of Mistral (Jiang et al., 2023) and Qwen (Bai et al., 2023) models with semantic-based k-NN retrieval to examine the generalizability of the ICL framework across different language models. While they achieve better automatic results compared to recent DG approaches, such as contrastive approach and instruction-tuning SLMs (i.e., TinyLlama) across most datasets, especially on harder reasoning benchmarks such as ARC-Challenge, ARC-Easy, and MedQA, GPT-3 consistently outperforms them across all selected datasets, as detailed in Table 4 and summarized in Figure 5. This indicates the effectiveness of the ICL framework reasoning for the DG task.

In addition, we evaluate different numbers of k-shot examples (5, 7, 20, 30), with the results reported in Table 7 of Appendix E. We limit the experiments to k=30 because of the the input-token limitations of prompt-based LLMs, particularly for the MedQA dataset. Notably, with k=30, our approach achieves *new* SOTA performance across all selected datasets based on automatic ranking evaluation metrics, outperforming recent DG methods and further demonstrating the effectiveness of reasoning for the DG task.

4.7 Case Study

Table 5 presents a case study analyzing the distractors generated by fourteen DG models across three selected examples. These models include ground-truth distractors, Text2Text models (base fine-tuning, multi-task learning, and two-stage contrastive learning), candidate generation and selection frameworks (beam search and clustering), instruction-tuning model, ICL framework, and COT-augmented ICL framework.

Initially, candidate generation and selection frameworks indicate notable limitations in DG task. First, in Example (1), T5(beam) is prone to generate the correct answer (starch) as a distractor. Then, in Example (2), GPT-3(clustering) may select distractors (H2O2, Hydroxide, Hydronium) that lack the level of plausibility expected by experts.

The contrastive approach yields clear improvements over the Text2Text baseline and multi-task learning, yet it still struggles to generate competitive distractors. As shown in Examples (1) and (3), the contrastive approach exhibits repetitive outputs and limited reasoning, generating distractors such as (proteins, fats, proteins) and (animals, plants, animals). Although Example (2) demonstrates that the contrastive approach generates more in-context distractors (helium, nitrogen, carbon) than the out-of-context distractors (air, ice, gas) generated by multi-task learning, these distractors still lack the depth of human-like reasoning and may therefore provide limited challenge to examinees.

In addition, instruction-tuning model (e.g., TinyLlama) exhibit notable issues, including generating overly broad and general distractors (proteins, cellulose, sap), as indicated in Example (1) , out-of-context distractors (gas, liquid, heat), as shown in Example (2), or highly implausible options (chickens, cars, fish), as outlined in Example (3). These distractors also lack the reasoning required to generate pedagogically meaningful, human-like options for effectively assessing examinees.

The ICL with k-NN retrieval generates more human-like distractors in all three cases. In Example (1), the distractors (cellulose, glycogen, fructose) are conceptually aligned with the question (main source of carbohydrates for various animals is *[MASK]*) and answer (starch), introducing a meaningful knowledge challenge. In Example (2), the generated distractors (hydrogen peroxide, carbon dioxide, ammonia) correspond to related chemical compounds (H_2O_2, CO_2, NH_3) rather than the target (H_2O), while the COT augmentation framework further refines the output as human-like reasoning distractors (oxygen, hydrogen, carbon). This indicates that the question (the common term for the chemical formula H_2O is *[MASK]*) and answer (water) can have multiple valid, reasoned distractors generated from different reasoning processes or learning objective outcomes such as distinguishing a compound from other chemical compounds or recognizing a compound based on its constituent elements. Example (3) further demonstrates that ICL with semantic-based k-NN retrieval generates distractors (birds, dolphins, chimpanzees) that are more closely aligned with the ground-truth distractors (apes, reptiles, rats) than those generated using random retrieval (plants, animals, machines) for the question (*[MASK]* has the greatest ability to learn) and answer (humans). For further analysis, Appendix F provides case study analyses of the impact of ICL framework and COT variants on DG, and Appendix G extends the case study by providing additional examples across all selected datasets.

5 Conclusion

In this paper, we present the in-context learning (ICL) framework that leverages the reasoning capabilities of large language models (LLMs) to generate distractors that more closely reflect human reasoning. We further incorporate chain-of-thought (COT) augmentation to generate both distractors and their associated rationales. This framework yields substantial improvements across six benchmarks covering challenging domains and varying distractor lengths. Both automatic and human evaluations show that the ICL framework achieves the state-of-the-art performance, generating more plausible, diverse, and contextually aligned distractors than recent fine-tuning models and distractor generation approaches.

Acknowledgments

We express our sincere gratitude to the anonymous reviewers for the constructive feedback. This work was conducted with the collaborative support from Macquarie University and Adelaide University in Australia, along with Umm Al-Qura University in the Kingdom of Saudi Arabia. We sincerely thank the members of the Intelligent Computing Laboratory in the School of Computing at Macquarie University for their valuable support, constructive feedback, and insightful discussions.

Limitations

While the in-context learning (ICL) framework with large language models (LLMs) demonstrates strong capability in generating contextually aligned and human-like distractors, it remains susceptible to hallucination. In particular, zero-shot prompting is more prone to undesired behaviors, such as repeating the question stem or leaking the correct answer among the distractors. Although few-shot prompting alleviates these issues to some extent, it does not fully eliminate them.

Moreover, different LLMs exhibit distinct generation behaviors. For instance, models such as Mistral tend to generate long lists of distractors, whereas GPT-3 more consistently adheres to the intended output format. These observations suggest that distractor quality is highly sensitive to prompt design, model architecture, and decoding strategies, which may limit the generalizability and reproducibility of ICL across different LLMs.

In addition, this work mainly utilizes English-language benchmarks to explore the effectiveness of the in-context learning framework. As a result, the findings may not directly generalize to multilingual or low-resource educational settings, where linguistic structure, cultural context, and domain-specific terminology can substantially affect distractor plausibility. Extending reasoning-aware distractor generation to low-resource languages remains an important direction for future research.

Also, distractor evaluation still largely relies on automatic token-level matching against ground-truth distractors. Although ICL generates substantially fewer distractors that exactly match the ground-truth answer for the improvement of reliability, it remains challenging to validate the correctness of the generated distractors that may constitute plausible but valid answer options. This limitation highlights the difficulty of reliably distinguishing high-quality distractors from semantically acceptable answers using surface-level evaluation alone.

Despite these limitations, our findings underscore the importance of explicit reasoning in the distractor generation task. Reasoning-aware in-context learning generates more plausible, diverse, and contextually grounded distractors than non-reasoning baselines. We hope that this work encourages the research community to further investigate robust output control, validation mechanisms, multilingual adaptation, and reasoning-aware constraints for ICL-based distractor generation.

References

Elaf Alhazmi, Quan Z. Sheng, Wei Emma Zhang, Mohammed I. Thanoon, Haojie Zhuang, Behnaz Soltani, and Munazza Zaib. 2025. Fine-tuning encoder-decoder models with contrastive learning for in-context distractor generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10056–10072.

Elaf Alhazmi, Quan Z. Sheng, Wei Emma Zhang, Munazza Zaib, and Ahoud Alhazmi. 2024. Distractor generation in multiple-choice tasks: A survey of methods, datasets, and evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14437–14458.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Distractor generation for multiple-choice questions with predictive prompting and large language models. *arXiv preprint arXiv:2307.16338*.

Semere Kiros Bitew, Amir Hadifar, Lucas Sterckx, Johannes Deleu, Chris Develder, and Thomas Demeester. 2022. Learning to reuse distractors to support multiple-choice question generation in education. *IEEE Transactions on Learning Technologies*, 17:375–390.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Dhawaleswar Rao Ch and Sujan Kumar Saha. 2018. Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies*, 13(1):14–25.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Bidyut Das, Mukta Majumder, Santanu Phadikar, and Arif Ahmed Sekh. 2021. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning*, 16(1):1–15.

Chenhe Dong, Yinghui Li, Haifan Gong, et al. 2022. A survey of natural language generation. *ACM Computing Survey*, 55(8):1–38.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.

Wanyong Feng, Jaewook Lee, Hunter McNichols, Alexander Scarlatos, Digory Smith, Simon Woodhead, Nancy Ornelas, and Andrew Lan. 2024. Exploring automated distractor generation for math multiple-choice questions via large language models. In *Findings of the Association for Computational Linguistics (NAACL)*, pages 3067–3082.

Nigel Fernandez, Alexander Scarlatos, Wanyong Feng, Simon Woodhead, and Andrew Lan. 2024. Divert: Distractor generation with variational errors represented as text for math multiple-choice questions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9081.

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

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education (IJAIED)*, 30:121–204.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6980.

Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C. Lee Giles. 2018. Distractor generation for multiple choice questions using Learning to Rank. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 284–290.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Haohao Luo, Yang Deng, Ying Shen, See-Kiong Ng, and Tat-Seng Chua. 2024. Chain-of-exemplar: Enhancing distractor generation for multimodal educational question generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7978–7993.

Kaushal Kumar Maurya and Maunendra Sankar Desarkar. 2020. Learning to distract: A hierarchical

multi-decoder network for automated generation of long distractors for multiple-choice questions for reading comprehension. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM)*, pages 1115–1124.

Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2023. Exploring automated distractor and feedback generation for math multiple-choice questions via in-context learning. *arXiv preprint arXiv:2308.03234*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Mayur Patidar, Riya Sawhney, Avinash Singh, Biswajit Chatterjee, Mausam ., and Indrajit Bhattacharya. 2024. Few-shot transfer learning for knowledge base question answering: Fusing supervised models with in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9147–9165.

Fanyi Qu, Hao Sun, and Yunfang Wu. 2024. Unsupervised distractor generation via large language model distilling and counterfactual contrastive decoding. In *Findings of the Association for Computational Linguistics (ACL)*, pages 827–838.

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

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Siyu Ren and Kenny Q Zhu. 2021. Knowledge-driven distractor generation for cloze-style multiple choice questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4339–4347.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

Shiva Taslimipoor, Luca Benedetto, Mariano Felice, and Paula Buttery. 2024. Distractor generation using generative and discriminative capabilities of transformer-based models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pages 5052–5063.

Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V Chawla. 2025. Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 251–260.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

Hui-Juan Wang, Kai-Yu Hsieh, Han-Cheng Yu, Jui-Ching Tsou, Yu An Shih, Chen-Hua Huang, and Yao-Chung Fan. 2023. Distractor generation based on Text2Text language models with pseudo Kullback-Leibler divergence regulation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12477–12491.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text (WNUT)*, pages 94–106.

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

Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 2018. Large-scale cloze test dataset created by teachers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2344–2356.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1358–1368.

Han Cheng Yu, Yu An Shih, Kin Man Law, KaiYu Hsieh, Yu Chen Cheng, Hsin Chih Ho, Zih An Lin, Wen-Chuan Hsu, and Yao-Chung Fan. 2024. Enhancing distractor generation for multiple-choice questions with retrieval augmented pretraining and knowledge graph integration. In *Findings of the Association for Computational Linguistics (ACL)*, pages 11019–11029.

Armel Randy Zebaze, Benoît Sagot, and Rachel Bawden. 2025. In-context example selection via similarity search improves low-resource machine translation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1222–1252.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024a. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Tianhui Zhang, Bei Peng, and Danushka Bollegala. 2024b. Improving diversity of commonsense generation by large language models via in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9226–9242.

A K-Nearest Neighbor Retrieval

The following algorithm describes the k-nearest neighbor (k-NN) retrieval used to select in-context examples. Algorithm 1 presents the in-context learning framework for distractor generation, where the model generates distractors $\hat{y}$ and can optionally incorporate chain-of-thought (COT) augmentation to generate $\hat{z}$, consisting of both distractors $\hat{y}$ and its rationales $\hat{r}$ for a given input x_{test}.

Algorithm 1 In-Context Learning with k-NN-based Retrieval and Optional COT Augmentation

Require: Test input x_{test}; training set $\mathcal{D}_T = \{(x_i, y_i[, r_i])\}_{i=1}^{N}$, where r_i denotes an optional chain-of-thought rationale available only when COT augmentation is enabled; sentence encoder $\mu_\theta(\cdot)$; number of few-shot examples k; binary flag $\gamma \in \{0, 1\}$, where $\gamma = 0$ corresponds to in-context learning and $\gamma = 1$ activates COT augmentation.

$v_{\text{test}} \leftarrow \mu_\theta(x_{\text{test}})$
for each $(x_i, y_i[, r_i]) \in \mathcal{D}_T$ **do**
  $v_i \leftarrow \mu_\theta(x_i)$
  $s_i \leftarrow \frac{v_i^\top v_{\text{test}}}{\|v_i\|_2 \|v_{\text{test}}\|_2}$
end for
Select indices $\mathcal{I}$ of the top-k examples with highest similarity scores
Retrieve examples $\mathcal{C} = \{(x_i, y_i[, r_i]) \mid i \in \mathcal{I}\}$
if $\gamma = 1$ **then**
  $C \leftarrow e_{r_1} \oplus \cdots \oplus e_{r_k}$
  $\hat{z} = (\hat{y}, \hat{r}) \leftarrow \text{LLM}([C; x_{\text{test}}])$
else
  $C \leftarrow e_1 \oplus \cdots \oplus e_k$
  $\hat{y} \leftarrow \text{LLM}([C; x_{\text{test}}])$
end if

B Datasets and Domains

We evaluate our approach on six multiple-choice question datasets across science, general knowledge, and medical domains.

The **Science** domain includes three main datasets. *SciQ* (Welbl et al., 2017) is a crowd-sourced dataset with distractors and questions averaging 1.6 and 14.5 tokens, respectively. *MCQL* (Liang et al., 2018) is a Web-crawled college-level science dataset with distractors averaging 1.2 tokens and questions 9.4 tokens. *ARC* (Clark et al., 2018) covers elementary science questions at Easy and Challenge levels, with question lengths ranging from 21.4 to 24.7 tokens and distractors from 4.1 to 5.5 tokens. Examples (B.1) and Example (B.2) illustrate two examples from SciQ and ARC-Challenge, respectively:

(B.1) **Stem:** *What is a disease caused by the same virus that causes chicken pox?*
Distractors: *a) gout, b) diabetes, c) hepatitis*
Answer: *shingles*

(B.2) **Stem:** *Which is a fact about penguins?*
Distractors: *a) Penguins are fierce competitors, b) Penguins are some of the most beautiful birds, c) Penguins make great pets*
Answer: *Penguins can live in climates with freezing temperatures*

The **General Knowledge** domain refers to the *MCQ* dataset[2] (Ren and Zhu, 2021) that covers diverse domains including science, vocabulary, commonsense, and trivia. It contains fill-in-the-blank questions where we replace "**blank**" with a `[MASK]` token. MCQ features relatively short questions with 19.5 tokens on average and single-token distractors, as presented in Example (B.3):

(B.3) **Stem:** *the only known planet with large amounts of water is [MASK]*
Distractors: *a) saturn, b) jupiter, c) mars*
Answer: *earth*

The **Medical** domain refers to the *MedQA* dataset (Jin et al., 2021) that is constructed from real medical licensing examinations. It contains substantially longer questions with 116.6 tokens on average and multi-token distractors with 3.5 tokens on average, as shown in Example (B.4), making it a challenging dataset for the DG task.

(B.4) **Stem:** *The parents of a 14-year-old patient are concerned and have questions about the use of insulin for their son's recently diagnosed type 1 diabetes. The patient has developed an upper respiratory infection while at school. He is coughing and has a runny nose. His temperature is* 37.8 °C (100.2 °F) *and vital signs are within normal limits. Physical examination is unremarkable. Which of the following modifications to his insulin regimen would you recommend to this patient and his parents?*
Distractors: *a) Reduce the insulin dose, b) Continue same regimen, c) Hold insulin until the patient gets better, d) Increase the insulin dose to double*
Answer: *Increase the frequency of blood glucose checks*

[2] https://github.com/DRSY/DGen

C Prompt Template

We conduct in-context learning by conditioning the model on a given context C, which consists of k example pairs $\{(x_i, y_i)\}_{i=1}^{k}$ selected via a retrieval mechanism and formatted using structured prompt templates. As illustrated in Example (C.1), the prompt begins with the k retrieved examples, followed by a `[stop]` token and the test input x_{test}.

(C.1) **Question**: *If an object is attracted to a magnet, the object is most likely made of [MASK]*
Answer: *metal*
Distractor1: *wood*
Distractor2: *plastic*
Distractor3: *cardboard*
...
`[stop]`
Question: *sugars are broken down into in your digestive system [MASK]*
Answer: *glucose*

We also follow a structured prompt template for chain-of-thought (COT) augmentation, based on a given context C that includes k few-shot examples $\{(x_i, y_i, r_i)\}_{i=1}^{k}$ selected via a retrieval mechanism. As presented in Example (C.2), the prompt template starts with k examples, followed by a `[stop]` token and then the test input x_{test}.

(C.2) **Question**: *If an object is attracted to a magnet, the object is most likely made of [MASK]*
Answer: *metal*
Distractor1: *wood*
Distractor2: *plastic*
Distractor3: *cardboard*
Rationale : *The fact that the object is attracted to a magnet indicates that it contains magnetic properties. The most common materials that exhibit magnetic properties are metals, such as iron, nickel, and cobalt. Therefore, the object is most likely made of metal if it is attracted to a magnet. Wood, plastic, and cardboard do not typically exhibit magnetic properties, so they are not likely to be the correct answer.*
...
`[stop]`
Question: *sugars are broken down into in your digestive system [MASK]*
Answer: *glucose*

We use a zero-shot prompt template, as illustrated in Example (C.3), which maps a given question-answer input x and an explicit instruction to generate distractor outputs y.

(C.3) **Generate three incorrect distractors for the following question.**
Question: *sugars are broken down into in your digestive system [MASK]*
Answer: *glucose*

For candidate generation within the candidate generation and selection framework, we use the prompt template outlined in Example (C.4) to generate ten candidate distractors. Few-shot examples are selected using k-NN-based retrieval.

(C.4) **Question**: *If an object is attracted to a magnet, the object is most likely made of [MASK]*
Answer: *metal*
Distractor1: *wood*
Distractor2: *plastic*
Distractor3: *cardboard*
...
`[stop]`
Generate ten incorrect distractors for the following question and answer
[Template]
Distractor1:
Distractor2:
Distractor3:
Question: *sugars are broken down into in your digestive system [MASK]*
Answer: *glucose*

D Models and Implementation Details

We implement all models using the Hugging Face Transformers framework (Wolf et al., 2020) and conduct experiments on two NVIDIA Tesla P100 GPUs. We also detail the implementation settings and training configurations for each model utilized in distractor generation and question answering.

D.1 Distractor Generation Models

We conduct distractor generation experiments using T5, TinyLlama, GPT-3, Mistral, and Qwen models. The following describes the experimental settings for each model.

T5: We optimize the *T5-base* model by using AdamW with an initial learning rate of 1e-4 for 10 epochs and a batch size of 4 within a candidate generation framework as well as the Text2Text architecture. We integrate a multi-task learning setup

with two objectives: question answering and distractor generation. For the contrastive-based variant, we adopt a two-stage training. The model is first fine-tuned using a standard generation objective, followed by additional fine-tuning with a combined loss consisting of the InfoNCE contrastive loss and the generation loss. The temperature τ is fixed at 0.1, and mean pooling is applied to obtain fixed-size embedding representations. For clustering, we apply agglomerative clustering[3] using Euclidean distance, with a distance threshold of 1.2. The representative element from each resulting cluster is then selected as the final distractor.

One important factor is the training cost of pre-trained language models such as the T5 model. Within the Text2Text architecture, Example (D.1) reports the training time required to fine-tune the two-stage contrastive Text2Text model across datasets, ranging from under 30 minutes on ARC-Challenge to over 5 hours on MedQA. Example (D.2) presents the training time associated with multi-task learning, which consistently incurs slightly higher computational cost across all datasets compared to the contrastive-based approach. These results indicate that, while the contrastive method requires marginally lower computational cost than multi-task learning, it also achieves better performance on the distractor generation task, especially in the Text2Text architecture.

(D.1) **Training Time:** *Contrastive Text2Text*
MCQ: *45 min 4 s*
SciQ: *4 h 28 min 52 s*
MCQL: *2 h 23 min 23 s*
ARC-Easy: *55 min 19 s*
ARC-Challenge: *28 min 30 s*
MedQA: *5 h 12 s*

(D.2) **Training Time:** *Multi-task Learning*
MCQ: *57 min 27 s*
SciQ: *5 h 51 min 11 s*
MCQL: *2 h 55 min 54 s*
ARC-Easy: *1 h 11 min 49 s*
ARC-Challenge: *37 min 22 s*
MedQA: *5 h 17 min 22 s*

Additionally, the results in Example (D.3) indicate that fine-tuning T5 for candidate generation is computationally more expensive, requiring substantially longer training times across the selected datasets than Text2Text-based methods. For instance, training on SciQ and MedQA requires over 8 hours and 10 hours, respectively, while even mid-sized datasets such as MCQL and ARC-Easy demand more than 4 hours and nearly 2 hours of training. In contrast, Text2Text approaches complete training in significantly less time, highlighting the higher computational overhead associated with candidate generation using T5.

(D.3) **Training Time:** *T5 Candidate Generation*
MCQ: *1 h 31 min 17 s*
SciQ: *8 h 17 min 35 s*
MCQL: *4 h 25 min*
ARC-Easy: *1 h 44 min 56 s*
ARC-Challenge: *54 min 7 s*
MedQA: *10 h 8 min 52 s*

TinyLlama: We fine-tune *TinyLlama-1.1B-Chat-v1.0*, using instruction tuning with Low-Rank Adaptation (LoRA) (Hu et al., 2022). The model is trained with prompts structured as question-answer pairs, where the model generates three distractors per question. We apply LoRA with rank $r = 8$, scaling factor $\alpha = 16$, and a dropout rate of 0.05, targeting the projection layers `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. The model is optimized using AdamW with a learning rate of 2e-4 for 10 epochs, a batch size of 4, and mixed-precision (FP16) training. For inference, we apply greedy decoding with a maximum generation length of 50 tokens.

Fine-tuning large language models through instruction tuning requires substantially more computational time than all recent approaches, including pre-trained models such as T5. As shown in Example (D.4), instruction tuning with the TinyLlama language model can exceed 24 hours on SciQ and more than two days on MedQA, highlighting its significant computational overhead.

(D.4) **Training Time:** *Instruction Tuning*
MCQ: *4 h 43 min 44 s*
SciQ: *1 day 23 h 52 min 18 s*
MCQL: *18 h 46 min 47 s*
ARC-Easy: *6 h 26 min 59 s*
ARC-Challenge: *2 h 20 min 38 s*
MedQA: *2 days 1 h 32 min 43 s*

GPT-3: We use the `gpt-3.5-turbo` model within an in-context learning framework, with $k = 5$

[3] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

Dataset	T5 Accuracy (%)						BART Accuracy (%)					
	MCQ	SciQ	MCQL	ARC-E	ARC-C	MedQA	MCQ	SciQ	MCQL	ARC-E	ARC-C	MedQA
Ground-Truth	48.26	68.00	34.00	40.45	27.13	21.76	30.12	52.50	27.33	29.38	19.80	20.42
T5 (beam)	56.76	64.70	49.17	38.55	36.52	32.68	40.12	51.70	43.00	39.35	31.91	31.81
GPT-3 (clustering)	42.08	59.60	39.00	40.74	31.66	29.69	26.25	47.70	34.83	31.40	28.50	28.04
T5 (contrast)	47.10	62.30	42.67	38.72	32.51	26.71	35.14	52.70	36.67	35.69	25.85	24.19
GPT-3 (k-NN)	41.31	50.30	30.83	36.53	25.43	21.05	25.87	39.80	26.50	26.22	20.31	20.03
GPT-3 (COT-k-NN)	36.29	50.20	30.83	35.56	27.30	19.95	22.78	40.90	26.33	27.99	20.90	19.25
TinyLlama	50.19	63.80	37.83	41.20	31.14	24.12	26.64	52.60	33.00	29.50	21.59	25.22

Table 6: Accuracy comparison (%) of distractor generation approaches using the T5 and BART across datasets.

shots, temperature set to 0, a maximum generation length of 350 tokens, and top-$p = 1$, corresponding to greedy decoding. For k-NN-based retrieval, we employ a pre-trained MPNet encoder (Reimers and Gurevych, 2019b) and compute cosine similarity between vectorized textual representations of test and training instances. The implementation leverages the OpenAI API with Azure OpenAI support to generate model responses. Prompts are processed in configurable batches of 20 per request.

Although GPT-3 is not an open-source large language model, it exhibits strong distractor generation performance while achieving significantly lower inference time, as shown in Example (D.5). For instance, inference completes within 2–10 minutes across all datasets, including less than 3 minutes on MCQ and ARC-Challenge, demonstrating its efficiency compared to fine-tuning-based approaches, including candidate generation and selection frameworks, Text2Text models, and instruction-tuned models.

(D.5) **Inference Time:** *GPT-3 ICL*
MCQ: *2 min 6 s*
SciQ: *9 min 29 s*
MCQL: *4 min 38 s*
ARC-Easy: *5 min 32 s*
ARC-Challenge: *2 min 46 s*
MedQA: *10 min 9 s*

Mistral: We explore the *Mistral-7B-Instruct* model via Ollama within the in-context learning framework, with $k = 5$ shots, temperature set to 0, top-$p = 1$, and `num_predict` = 1500, corresponding to greedy decoding. The model is deployed locally through the Ollama interface. Unlike GPT-3, we select only the first three generated distractors from the output results, as the Mistral large language model tends to generate long distractor lists.

(D.6) **Inference Time:** *Mistral ICL*
MCQ: *37 min 59 s*
SciQ: *1 h 24 min 51 s*
MCQL: *30 min 41 s*
ARC-Easy: *1 h 55 min 44 s*
ARC-Challenge: *48 min 16 s*
MedQA: *1 h 28 min 48 s*

Qwen: We explore the *qwen2.5:7b* model via Ollama within the in-context learning framework, using $k = 5$ shots, temperature set to 0, top-$p = 1$, and `num_predict`=1024, corresponding to greedy decoding. The model is deployed locally through the Ollama interface. A system-level instruction is incorporated to constrain the model to generate exactly three plausible but incorrect distractors that follow the format of the provided examples.

(D.7) **Inference Time:** *Qwen ICL*
MCQ: *21 min 14 s*
SciQ: *54 min 56 s*
MCQL: *37 min 7 s*
ARC-Easy: *2 h 39 min 8 s*
ARC-Challenge: *1 h 30 min 3 s*
MedQA: *2 h 57 min 58 s*

Compared to the fine-tuning models, Mistral and Qwen under the in-context learning paradigm eliminate the need for costly training, enabling immediate and flexible deployment. However, this advantage comes at the cost of higher inference latency, as shown in Example (D.6) and Example (D.7). Although fine-tuned approaches incur substantial training costs (Examples (D.1)–(D.4)), they benefit from significantly faster inference once training is completed. This highlights a clear trade-off between training efficiency and inference efficiency.

D.2 Question Answering Models

We employ two encoder–decoder models, T5 (Raffel et al., 2020) and BART (Lewis et al., 2020), as question-answering models. The models *t5-base*

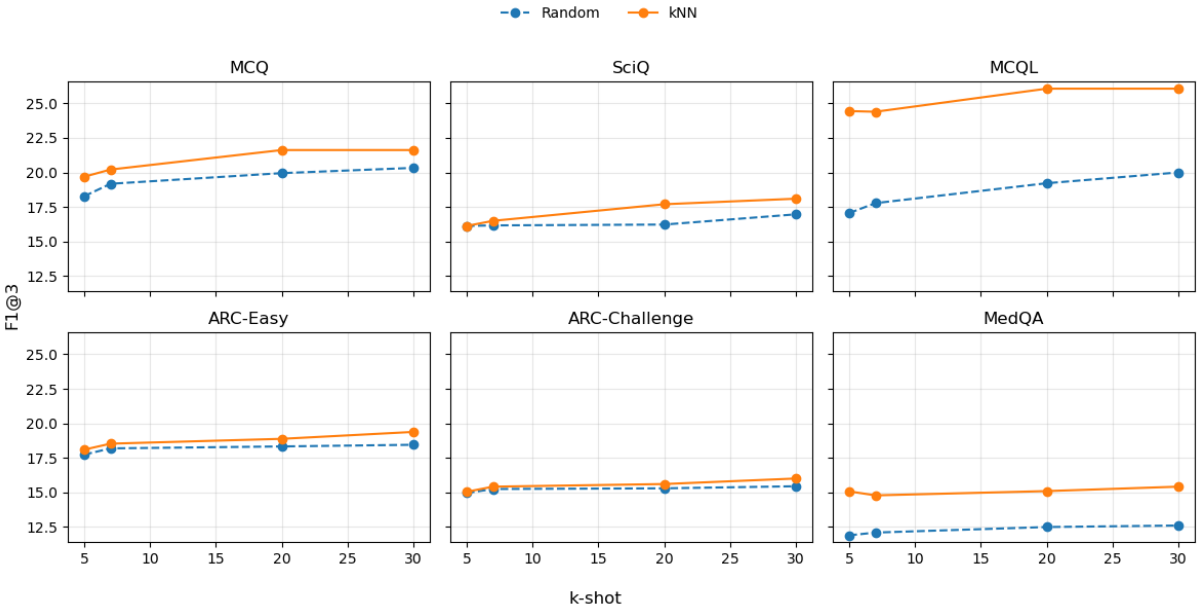


Figure 6: F1@3 score comparison with varying number of few-shot ($k \in \{5, 7, 20, 30\}$) across datasets.

and *facebook/bart-base* are fine-tuned in a supervised manner, conditioned on the given question and its associated options, which randomly include the generated distractors and the correct answer. The input is prefixed with the labels "`question:`" and "`options:`" before the corresponding content.

The models are optimized using AdamW with a learning rate of 1e-4 and weight decay of 0.01, trained for 10 epochs with a batch size of 4. We set maximum input length to 512 tokens and maximum target length to 128 tokens. During inference, we use greedy decoding with early stopping and a maximum generation length of 50 tokens. The model is evaluated based on exact match accuracy between generated and ground-truth answers.

Let $\mathcal{D} = \{(a_i, \hat{a}_i)\}_{i=1}^{N}$ denote a set of N evaluation instances, where a_i is the ground-truth answer and $\hat{a}_i$ is the generated answer. We define the indicator function $\mathbb{I}(\cdot)$ as:

$$\mathbb{I}(a_i = \hat{a}_i) = \begin{cases} 1, & \text{if } a_i = \hat{a}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

the accuracy is computed as:

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(a_i = \hat{a}_i) \times 100 \tag{7}$$

where both a_i and $\hat{a}_i$ are normalized by lowercasing and trimming whitespace. Table 6 reports accuracy results for distractor generation approaches.

E k-Shot Ablation

Table 7 reports the automatic ranking-based evaluation metrics for analyzing the effect of varying the number of k-shot examples $\{5, 7, 20, 30\}$ across all selected datasets, where the experiments stop at $k = 30$ because the input token limitations of prompt-based large language models. For example, questions in the MedQA dataset contain an average of 116.6 tokens, and increasing k further may exceed the model maximum input length. In terms of evaluation, the performance shows a clear and consistent trend. The automatic results steadily improve as k increases for both random and k-NN retrieval. This indicates that richer in-context examples substantially benefit distractor generation in aligning closely with human-labeled distractors.

Figure 6 summarizes the impact of varying the number of k-shot examples on the F1@3 score. k-NN-based retrieval consistently outperforms random retrieval across all datasets, with particularly significant improvements observed on MCQL and MedQA. Specifically, the F1@3 score increases from 20.00 to 26.06 on MCQL and from 12.61 to 15.42 on MedQA when using k-NN selected examples with $k = 30$. This highlights the importance of selecting semantically relevant examples rather than relying on randomly sampled contexts.

Furthermore, the highest F1@3 scores are consistently achieved when $k = 30$ across all selected datasets, indicating *new* state-of-the-art (SOTA) automatic results compared to recent approaches, as

reported in Table 2. Under k-NN-based retrieval, increasing k from 5 to 30 improves F1@3 from 19.69 to 21.62 on MCQ, from 16.53 to 18.10 on SciQ, from 24.44 to 26.06 on MCQL, from 18.10 to 19.38 on ARC-Easy, from 15.05 to 16.01 on ARC-Challenge, and from 15.08 to 15.42 on MedQA. These results further demonstrate that leveraging k-NN-based retrieval with larger *k*-shot contexts provides more informative in-context examples, leading to consistent improvements in distractor generation performance across varying distractor token lengths and diverse domains.

F Analysis of Case Study

We analyze the case studies presented in Table 5 for Examples (1) and (2) by examining the effectiveness of different few-shot example selection methods in generating distractors. Specifically, we conduct two comparisons: (i) the benefit of semantic-based k-NN retrieval over random retrieval for selecting examples and (ii) the impact of integrating COT rationales within the ICL retrieval methods (i.e., random and k-NN) for distractor generation.

First, the comparison between random example selection and k-NN retrieval with the base ICL framework demonstrates the importance of semantic similarity in few-shot prompting for distractor generation. In the random selection method, the retrieved examples are randomly selected, which may relate to different topics to a given test question-answer pair. As indicated in Table 8 for Example (1) mentioned in Table 5, the GPT-3 model receives weak contextual examples regarding the target question (main source of carbohydrates for various animals is ...etc.) and its related answer (starch). Consequently, the generated distractors include broad and less semantically related options (protein, fat) that belong to different nutritional categories rather than closely related carbohydrate concepts. Although these distractors are incorrect, they may be relatively easy to eliminate by examinees because they differ from the correct answer.

In contrast, the k-NN retrieval method selects semantically similar examples centered on carbohydrates, glycogen, starch, polysaccharides, and related biological concepts as indicated in Table 8. These retrieved examples provide GPT-3 with a focused semantic context and expose the model to plausible distractors (cellulose, glycogen, fructose) within the same conceptual domain, making them difficult for examinees to distinguish from the correct answer. This example highlights that k-NN retrieval effectively improves distractor generation quality by retrieving contextually relevant examples that guide GPT-3 toward generating semantically coherent and educationally effective distractors. We also observe the effectiveness of integrating COT with both random and k-NN retrieval methods in generating more domain-related distractors (cellulose, glucose, glycogen, fructose) as illustrated in Tables 10–11.

Second, the effectiveness of distractor generation depends not only on semantic plausibility but also on aligning the distractors with the intended conceptual focus and cognitive level of the examinees. As mentioned in Example (2) in Table 5, for the question ("the common term for the chemical formula H_2O is ...") with the correct answer ("water"), the ground-truth distractors ("hydrogen", "air", "oxygen") are conceptually associated with the elemental composition and formation of the compound rather than with other alternative chemical compound formulas. This indicates that the educational objective of the MCQ item is not simply to distinguish water from other compounds, but to assess whether examinees understand the relationship between a compound and its constituent elements. Although multiple valid distractor types may exist for the same question, such as compound-related options or element-related options, the distractors should remain aligned with the underlying conceptual intention of the MCQ learning objective.

However, COT with random retrieval often generates distractors (hydrogen peroxide, carbon dioxide, sodium chloride), which mainly represent alternative chemical compounds with different formulas (H_2O_2, CO_2, and NaCl) as indicated in Table 12. Similar to standard baseline ICL distractors with both random and k-NN retrieval, as indicated in Table 9, these generated options shift the focus of the question toward distinguishing among compounds rather than evaluating knowledge of the elemental composition of water, as reflected in the human-selected ground-truth distractors.

In contrast, COT with k-NN retrieval successfully selects semantically related examples whose rationales emphasize the relationship between compounds and their constituent elements as indicated in Table 13. The COT and semantic similarity between the retrieved examples and the target question help the model to transition prediction to generating distractors (oxygen, hydrogen, carbon), which are conceptually closer to the intent and rea-

Dataset	k	Model	P@1	R@1	F1@3	MRR	NDCG@3
MCQ	5	GPT-3 (few-shot-random)	25.87	8.62	18.28	33.72	40.68
	7		27.41	9.14	19.18	35.20	42.08
	20		30.50	10.17	19.95	37.77	44.11
	30		30.89	10.30	20.33	38.48	45.27
	5	GPT-3 (few-shot-k-NN)	29.73	9.91	19.69	36.62	42.37
	7		30.12	10.04	20.21	36.49	41.44
	20		**32.05**	**10.68**	**21.62**	**39.32**	45.26
	30		30.12	10.04	21.62	38.42	**45.29**
SciQ	5	GPT-3 (few-shot-random)	25.10	8.37	16.13	30.17	34.57
	7		24.90	8.30	16.17	29.92	34.24
	20		25.00	8.33	16.23	30.12	34.36
	30		25.60	8.53	16.97	31.07	35.65
	5	GPT-3 (few-shot-k-NN)	24.90	8.30	16.53	30.45	35.24
	7		24.70	8.23	16.50	30.37	35.32
	20		25.20	8.40	17.70	31.63	37.25
	30		**25.90**	**8.63**	**18.10**	**32.18**	**37.61**
MCQL	5	GPT-3 (few-shot-random)	30.17	10.06	19.06	34.36	37.98
	7		29.00	9.67	17.78	32.53	35.48
	20		31.83	10.61	19.22	35.22	37.98
	30		33.83	11.28	20.00	37.17	39.88
	5	GPT-3 (few-shot-k-NN)	35.33	11.78	24.44	39.61	42.81
	7		36.00	12.00	24.39	40.36	43.86
	20		37.50	12.50	**26.06**	42.08	**45.69**
	30		**38.17**	**12.72**	26.06	**42.17**	45.19
ARC-Easy	5	GPT-3 (few-shot-random)	23.19	7.75	17.74	28.37	32.41
	7		23.95	8.00	18.19	29.03	32.95
	20		24.37	8.14	18.33	29.48	33.39
	30		23.86	7.98	18.25	29.15	33.29
	5	GPT-3 (few-shot-k-NN)	24.24	8.10	18.10	29.08	32.82
	7		24.66	8.24	18.53	29.63	33.39
	20		24.87	8.31	18.88	29.85	33.59
	30		**25.93**	**8.66**	**19.38**	**30.77**	**34.37**
ARC-Challenge	5	GPT-3 (few-shot-random)	21.08	7.07	14.96	24.82	27.75
	7		20.48	6.87	15.25	24.96	28.60
	20		20.90	7.01	15.30	25.24	28.76
	30		21.16	7.10	15.45	25.20	28.36
	5	GPT-3 (few-shot-k-NN)	20.48	6.87	15.05	24.06	26.80
	7		20.65	6.93	15.42	24.63	27.77
	20		20.48	6.87	15.61	24.64	27.88
	30		**21.25**	**7.13**	**16.01**	**25.46**	**28.77**
MedQA	5	GPT-3 (few-shot-random)	17.44	4.36	11.90	23.10	26.50
	7		18.22	4.56	12.10	23.46	26.44
	20		18.70	4.67	12.50	24.09	27.18
	30		19.80	4.95	12.61	24.89	27.79
	5	GPT-3 (few-shot-k-NN)	20.90	5.22	15.08	27.44	30.52
	7		20.42	5.11	14.79	26.90	30.02
	20		21.45	5.36	15.10	27.89	30.70
	30		**22.00**	**5.50**	**15.42**	**28.45**	**31.52**

Table 7: Ablation results with varying number of few-shot examples ($k \in \{5, 7, 20, 30\}$) across datasets.

soning structure of the original human-based MCQ, as they evaluate whether examinees can distinguish the compound from its composing elements.

This direction also indicates that incorrect options are not only concerned with generating in-context distractors, but also with validating why these distractors can effectively assess examinees' knowledge through reasoning, thereby satisfying the learning objectives of the MCQ items. In this context, distractors should reflect common misconceptions, incomplete reasoning processes, or plausible errors that learners are likely to make, highlighting the essential of reasoning-aware mechanisms in distractor generation.

Question [Answer]: Main source of carbohydrates for various animals is [starch] **Ground-truth**: [glycogen, agar, pectin]	
Prompt: **GPT-3(random)**	**Prompt**: **GPT-3(k-NN)**
Question: Cell surface membrane is also called as **Answer**: plasma membrane **Distractor1**: cell wall **Distractor2**: cytoplasm **Distractor3**: nucleus [stop]	**Question**: Animal starch is called **Answer**: glycogen **Distractor1**: glucose **Distractor2**: cellulose **Distractor3**: fructose [stop]
Question: Value of 'm' depends upon values of **Answer**: l **Distractor1**: s **Distractor2**: d **Distractor3**: f [stop]	**Question**: Animals store carbohydrates mainly in form of **Answer**: glycogen **Distractor1**: cellulose **Distractor2**: starch **Distractor3**: galactose [stop]
Question: Type of force that opposes motion in fluids (water and air) is termed as **Answer**: resistance **Distractor1**: tension **Distractor2**: weight **Distractor3**: friction [stop]	**Question**: Starch is commonly formed and stored in **Answer**: plants **Distractor1**: animals **Distractor2**: fish **Distractor3**: insects [stop]
Question: Sea water is alkaline with a pH of **Answer**: 8 **Distractor1**: less than 8 **Distractor2**: 9 **Distractor3**: 10 [stop]	**Question**: Cereals, potatoes and tapioca are common sources of **Answer**: polysaccharides **Distractor1**: simple sugars **Distractor2**: complex sugars **Distractor3**: monosaccharides [stop]
Question: Heart can be detected in pulse only by **Answer**: arteries **Distractor1**: veins **Distractor2**: capillaries **Distractor3**: lymphatic vessels [stop]	**Question**: Starch $(C_6H_{10}O_5)_n$ is a form of **Answer**: carbohydrate **Distractor1**: proteins **Distractor2**: fats **Distractor3**: minerals [stop]
Question: Main source of carbohydrates for various animals is **Answer**: starch **Distractor1**: protein **Distractor2**: cellulose **Distractor3**: fat	**Question**: Main source of carbohydrates for various animals is **Answer**: starch **Distractor1**: cellulose **Distractor2**: glycogen **Distractor3**: fructose

Table 8: The prompt illustrates baseline ICL with both random sampling and k-NN retrieval using GPT-3 for **Example (1)** in Table 5.

G Supplementary Examples

To further support the case study examples presented in Table 5, additional extended case studies are provided in Tables 14–18. These supplementary examples span a broader range of question types, domains, and distractor difficulty levels across the selected datasets. The generated distractors are also presented together with their corresponding rationales generated through COT augmentation.

Consistent with the main case study in Section 4.7, the extended examples reveal the limitations of the candidate generation and selection framework, contrastive learning Text2Text models, and instruction-tuned models, including answer leakage, repetitive distractors, and the generation of out-of-context and implausible distractors.

In contrast, the in-context learning framework, particularly when combined with semantic k-NN-based retrieval and COT augmentation, consistently generates more contextually appropriate and human-like distractors compared to recent approaches, highlighting the importance of reasoning in the distractor generation task. By leveraging semantically similar examples, the framework is better able to capture the underlying educational context of the question and generate distractors that reflect realistic misconceptions and plausible reasoning errors across all selected datasets. These extended examples demonstrate the robustness of the in-context learning and its variants with COT augmentation for effective distractor generation across varying average distractor lengths and domains.

Question [Answer]: The common term for the chemical formula H_2O is [water] **Ground-truth**: [hydrogen, air, oxygen]	
Prompt: GPT-3(random)	**Prompt: GPT-3(k-NN)**
Question: the long , thin protein extensions in most prokaryotic cells are called [MASK] **Answer**: flagellum **Distractor1**: analyte **Distractor2**: cingulum **Distractor3**: neutrons [stop]	**Question**: In the presence of oxygen , hydrogen can interact to make [MASK] **Answer**: water **Distractor1**: carbon **Distractor2**: helium **Distractor3**: acid [stop]
Question: The variety of cell shapes seen in prokaryotes and eukaryotes reflects the functions that each [MASK] has **Answer**: cell **Distractor1**: proteins **Distractor2**: life **Distractor3**: organ [stop]	**Question**: Oxygen is combined with [MASK] to form water **Answer**: hydrogen **Distractor1**: glucose **Distractor2**: helium **Distractor3**: carbon [stop]
Question: The organization of roles, patterns of communication, and power in a group defines the group's [MASK] **Answer**: structure **Distractor1**: status **Distractor2**: cohesiveness **Distractor3**: norms [stop]	**Question**: a compound is [MASK] **Answer**: ammonia **Distractor1**: butter **Distractor2**: nitrogen **Distractor3**: hydrogen [stop]
Question: Jack Weiss recently won a $10 million lottery. Jack's life will change a great deal, particularly how he spends money. According to Engel's laws, Jack is LEAST likely to change the percentage of his income that he spends on [MASK] of the following **Answer**: housing **Distractor1**: food **Distractor2**: entertainment **Distractor3**: utilities **Distractor4**: savings [stop]	**Question**: the chemical formula for carbon dioxide is [MASK] **Answer**: CO2 **Distractor1**: CO **Distractor2**: C **Distractor3**: O2 [stop]
Question: the moon 's shadow 's inner part is called [MASK] **Answer**: umbra **Distractor1**: mullah **Distractor2**: penumbra **Distractor3**: aurora [stop]	**Question**: [MASK] is formed when an oxygen atom picks up a pair of hydrogen ions from a solution **Answer**: water **Distractor1**: liquid **Distractor2**: turpentine **Distractor3**: ammonia [stop]
Question: the common term for the chemical formula h 2 0 is [MASK] **Answer**: water **Distractor1**: carbon dioxide **Distractor2**: oxygen **Distractor3**: hydrogen peroxide	**Question**: the common term for the chemical formula h 2 0 is [MASK] **Answer**: water **Distractor1**: hydrogen peroxide **Distractor2**: carbon dioxide **Distractor3**: ammonia

Table 9: The prompt illustrates baseline ICL with both random sampling and k-NN retrieval using GPT-3 for **Example (2)** in Table 5.

Question [Answer]: Main source of carbohydrates for various animals is [starch] **Ground-truth**: [glycogen, agar, pectin]

Prompt: GPT-3(COT-random)

Question: Cell surface membrane is also called as
Answer: Plasma Membrane
Distractor1: Nuclear Envelope
Distractor2: Nuclear Membrane
Distractor3: Cell Wall
Rationale: The cell surface membrane is the outermost boundary of a cell, separating the cell's internal environment from its external environment. The cell surface membrane is composed of a phospholipid bilayer with embedded proteins that regulate the passage of molecules in and out of the cell. The term "plasma membrane" is often used interchangeably with "cell surface membrane" to refer to this structure. Nuclear Envelope, Nuclear Membrane, and Cell Wall are all different structures with different functions within a cell. The nuclear envelope surrounds the nucleus, the nuclear membrane is the double membrane surrounding the nucleus, and the cell wall is a rigid structure found in plant cells and some prokaryotic cells. These structures are not synonymous with the cell surface membrane.
[stop]
Question: Value of 'm ' depends upon values of
Answer: l
Distractor1: s
Distractor2: d
Distractor3: f
Rationale: We need to determine the relationship between 'm' and 'l'. Once we understand how 'm' depends on 'l', we can then consider the other variables 's', 'd', and 'f' to see if they have any impact on the value of 'm'. Since the question states that the value of 'm' depends upon the value of 'l', we can focus on understanding that relationship. The distractors 's', 'd', and 'f' are not relevant in this context and can be ignored when determining the value of 'm'.
[stop]
Question: Type of force that opposes motion in fluids (water and air) is termed as
Answer: resistance
Distractor1: Tension
Distractor2: Weight
Distractor3: Friction
Rationale: Tension is a force that occurs in a rope or cable that is being pulled from both ends. It is not directly related to opposing motion in fluids. Weight is the force of gravity acting on an object, which is not specific to opposing motion in fluids. Friction is a force that opposes motion when two surfaces are in contact, but it is not specific to fluids like water and air. The answer is resistance, which is the force that opposes motion in fluids like water and air.
[stop]
Question: Sea water is alkaline with a pH of
Answer: 8
Distractor1: less than 8
Distractor2: 9
Distractor3: 10
Rationale: The question states that sea water is alkaline with a pH of 8. The distractors are "less than 8", "9", and "10". Since sea water has a pH of 8, the distractor "less than 8" is incorrect. The distractor "9" is incorrect because it is higher than the pH of sea water. The distractor "10" is also incorrect because it is even higher than a pH of 9.
[stop]
Question: Heart ca be detected in pulse only by
Answer: arteries
Distractor1: veins
Distractor2: capillaries
Distractor3: lymphatic vessels
Rationale: The heart pumps blood through the arteries, which carry oxygenated blood away from the heart to the rest of the body. The pulse is the rhythmic throbbing of the arteries as blood is pumped through them by the heart. Therefore, the heart can be detected in the pulse only by the arteries, as they are directly connected to the heart and carry the blood that creates the pulse. Veins, capillaries, and lymphatic vessels do not directly carry the blood being pumped by the heart, so they would not be able to detect the heart in the pulse.
[stop]

Question: Main source of carbohydrates for various animals is
Answer: starch

Distractor1: cellulose
Distractor2: glucose
Distractor3: glycogen
Rationale: Starch is a complex carbohydrate found in plants and is a common source of carbohydrates for various animals. Cellulose is also a complex carbohydrate found in plants, but it is not easily digestible by many animals and is not a main source of carbohydrates for them. Glucose is a simple sugar and can be a source of carbohydrates, but it is not the main source for various animals. Glycogen is a form of stored glucose in animals, but it is not the main source of carbohydrates for them. Therefore, the correct answer is starch.

Table 10: The prompt illustrates COT-ICL with random selection using GPT-3 for **Example (1)** in Table 5.

Question [Answer]: Main source of carbohydrates for various animals is [starch] **Ground-truth**: [glycogen, agar, pectin]

Prompt: **GPT-3(COT-k-NN)**

Question: Animal starch is called
Answer: glycogen
Distractor1: glucose
Distractor2: cellulose
Distractor3: fructose
Rationale: Animal starch is a term used to describe glycogen, a polysaccharide that serves as a form of energy storage in animals. Glucose is a simple sugar that is a component of many carbohydrates, but it is not specifically referred to as animal starch. Cellulose is a polysaccharide found in the cell walls of plants and is not related to animal starch. Fructose is another simple sugar found in fruits and honey, but it is not the same as animal starch.
[stop]
Question: Animals store carbohydrates mainly in form of
Answer: glycogen
Distractor1: cellulose
Distractor2: starch
Distractor3: galactose
Rationale: Animals store carbohydrates mainly in the form of glycogen, which is a polysaccharide made up of glucose molecules. Glycogen is the primary storage form of glucose in animals, particularly in the liver and muscles. Cellulose is a structural polysaccharide found in plant cell walls and is not a form of carbohydrate storage in animals. Starch is another form of carbohydrate storage found in plants, not animals. Galactose is a monosaccharide sugar and is not typically stored in large amounts in animals. Therefore, the correct answer is glycogen.
[stop]
Question: Starch is commonly formed and stored in
Answer: plants
Distractor1: animals
Distractor2: fish
Distractor3: insects
Rationale: Starch is a carbohydrate that serves as a storage form of energy in plants. Plants produce starch through photosynthesis, where they convert sunlight into energy and store it in the form of starch. Animals, fish, and insects do not produce starch in the same way that plants do. They may consume starch-containing plants for energy, but they do not store starch in their bodies in the same way that plants do. Therefore, the correct answer is plants ...etc.
[stop]
Question: Cereals , potatoes and tapioca are common sources of
Answer: polysaccharides
Distractor1: simple sugars
Distractor2: complex sugars
Distractor3: monosaccharides
Rationale: Cereals, potatoes, and tapioca are all sources of carbohydrates, specifically polysaccharides. Polysaccharides are complex carbohydrates made up of multiple sugar units bonded together. They are found in foods like grains, starchy vegetables, and root vegetables. Simple sugars, also known as monosaccharides or disaccharides, are not the correct answer because cereals, potatoes, and tapioca are sources of polysaccharides, which are more complex carbohydrates made up of multiple sugar units. Complex sugars - While this term may seem related to polysaccharides, it is not the most accurate term to describe the carbohydrates found in cereals, potatoes, and tapioca. Polysaccharides specifically refer to complex carbohydrates made up of multiple sugar units. Monosaccharides are single sugar units, such as glucose and fructose. They are not the correct answer in this context because cereals, potatoes, and tapioca are sources of polysaccharides, which are made up of multiple sugar units bonded together.
[stop]
Question: Starch (C6H10O5) n is a form of
Answer: carbohydrate
Distractor1: proteins
Distractor2: fats
Distractor3: minerals
Rationale: Starch is a type of carbohydrate, a macronutrient that provides energy to the body. Proteins are another type of macronutrient, but they are made up of amino acids, not glucose molecules like carbohydrates. Fats are also a type of macronutrient, but they are made up of fatty acids and glycerol, not glucose molecules like carbohydrates. Minerals are micronutrients that are essential for various bodily functions, but they are not a form of starch.
[stop]

Question: Main source of carbohydrates for various animals is
Answer: starch

Distractor1: cellulose
Distractor2: glycogen
Distractor3: fructose
Rationale: The main source of carbohydrates for various animals is starch, a polysaccharide found in plants. ...etc. Cellulose is also a polysaccharide found in plants, but it is not easily digestible by many animals and is not their main source of carbohydrates. Glycogen is a form of carbohydrate storage in animals, not a source of carbohydrates from external sources. Fructose is a simple sugar found in fruits, but it is not the main source of carbohydrates for animals.

Table 11: The prompt illustrates COT-ICL with k-NN retrieval using GPT-3 for **Example (1)** in Table 5.

Question [Answer]: The common term for the chemical formula H_2O is [water] **Ground-truth**: [hydrogen, air, oxygen]

Prompt: GPT-3(COT-random)

Question: the long , thin protein extensions in most prokaryotic cells are called [MASK]
Answer: flagellum
Distractor1: analyte
Distractor2: cingulum
Distractor3: neutrons
Rationale: The question mentions that the long, thin protein extensions are found in prokaryotic cells. This narrows down the options to structures specific to prokaryotic cells. The term "flagellum" fits the description of a long, thin protein extension found in prokaryotic cells and is used for movement. The distractors "analyte," "cingulum," and "neutrons" do not match the description because they refer to unrelated scientific concepts. Therefore, "flagellum" is the correct answer.
[stop]
Question: The variety of cell shapes seen in prokaryotes and eukaryotes reflects the functions that each [MASK] has
Answer: cell
Distractor1: proteins
Distractor2: life
Distractor3: organ
Rationale: The question refers to the different shapes observed in prokaryotic and eukaryotic cells. The correct answer is "cell" because cell shape is related to the function performed by each cell. The distractors "proteins," "life," and "organ" do not directly describe the biological unit whose shape varies according to function. Therefore, "cell" is the correct answer.
[stop]
Question: The organization of roles, patterns of communication, and power in a group defines the group's [MASK]
Answer: structure
Distractor1: status
Distractor2: cohesiveness
Distractor3: norms
Rationale: The organization of roles determines how responsibilities are distributed within the group. Patterns of communication describe how members exchange information and interact. Power relationships influence authority and decision-making within the group. Together, these elements define the group's overall "structure.". Therefore, "structure" is the correct answer.
[stop]
Question: Jack Weiss recently won a $10 million lottery. Jack's life will change a great deal, particularly how he spends money. According to Engel's laws, Jack is LEAST likely to change the percentage of his income that he spends on [MASK] of the following
Answer: housing
Distractor1: food
Distractor2: entertainment
Distractor3: utilities
Distractor4: savings
Rationale: Engel's law states that as income increases, the proportion of income spent on food decreases. Spending on categories such as entertainment and savings is more likely to increase with higher income. Housing expenditure tends to remain relatively stable as a percentage of income. Therefore, "housing" is the correct answer.
[stop]
Question: the moon 's shadow 's inner part is called [MASK]
Answer: umbra
Distractor1: mullah
Distractor2: penumbra
Distractor3: aurora
Rationale: The moon's shadow has two main regions: the umbra and the penumbra. The "umbra" is the darkest inner part where light is completely blocked. The "penumbra" is the lighter outer region where light is partially blocked. "Mullah" and "aurora" are unrelated terms. Therefore, "umbra" is the correct answer.
[stop]

Question: the common term for the chemical formula h 2 0 is [MASK]
Answer: water

Distractor1: hydrogen peroxide
Distractor2: carbon dioxide
Distractor3: sodium chloride
Rationale: The chemical formula H_2O represents water. "Hydrogen peroxide" has the formula H_2O_2. "Carbon dioxide" has the formula CO_2. "Sodium chloride" has the formula NaCl. Therefore, "water" is the correct answer for H_2O.

Table 12: The prompt illustrates COT-ICL with random selection using GPT-3 for **Example (2)** in Table 5.

Question [Answer]: The common term for the chemical formula H_2O is [water] **Ground-truth**: [hydrogen, air, oxygen]

Prompt: **GPT-3(COT-k-NN)**

Question: In the presence of oxygen , hydrogen can interact to make [MASK]
Answer: water
Distractor1: carbon
Distractor2: helium
Distractor3: acid
Rationale: In the presence of oxygen, hydrogen can react to form a compound. The most common compound formed when hydrogen and oxygen react is water (H_2O). Water is made up of two hydrogen atoms and one oxygen atom. Therefore, the correct answer is water.
[stop]
Question: Oxygen is combined with [MASK] to form water
Answer: hydrogen
Distractor1: glucose
Distractor2: helium
Distractor3: carbon
Rationale: The question states that oxygen is combined with something to form water. In the chemical formula for water, H_2O, hydrogen is the other element combined with oxygen. Glucose, helium, and carbon are not elements typically combined with oxygen to form water. Therefore, the correct answer is hydrogen.
[stop]
Question: a compound is [MASK]
Answer: ammonia
Distractor1: butter
Distractor2: nitrogen
Distractor3: hydrogen
Rationale: The compound mentioned is likely a gas, as it is commonly used in household cleaning products and fertilizers. It is composed of nitrogen and hydrogen atoms bonded together. The compound is also known for its pungent smell. Based on these clues, the compound is most likely ammonia.
[stop]
Question: the chemical formula for carbon dioxide is [MASK]
Answer: CO2
Distractor1: CO
Distractor2: C
Distractor3: O2
Rationale: The chemical symbol for carbon is C. The chemical symbol for oxygen is O. When carbon and oxygen combine to form carbon dioxide, the chemical formula is CO_2. This means that there is one carbon atom (C) and two oxygen atoms (O) in each molecule of carbon dioxide. Therefore, the correct chemical formula for carbon dioxide is CO_2.
[stop]
Question: [MASK] is formed when an oxygen atom picks up a pair of hydrogen ions from a solution
Answer: water
Distractor1: liquid
Distractor2: turpentine
Distractor3: ammonia
Rationale: The question mentions that an oxygen atom picks up a pair of hydrogen ions from a solution. This process describes the formation of water through the chemical reaction between oxygen and hydrogen. Water is formed when two hydrogen atoms bond with one oxygen atom, creating a water molecule (H_2O). The distractors "liquid," "turpentine," and "ammonia" are not formed in the same way as water. Therefore, the correct answer is water.
[stop]

Question: the common term for the chemical formula h 2 0 is [MASK]
Answer: water

Distractor1: oxygen
Distractor2: hydrogen
Distractor3: carbon
Rationale: The chemical formula H_2O represents a specific compound. The common term for the chemical formula H_2O is water. Oxygen, hydrogen, and carbon are not the common terms used to refer to the chemical formula H_2O. Therefore, the correct answer is water.

Table 13: The prompt illustrates COT-ICL with k-NN retrieval using GPT-3 for **Example (2)** in Table 5.

Example: G.1
Dataset: MCQ
Question [Answer]: [humans] has the greatest ability to learn.

Model	Generated Distractors	Rationale
Ground Truth	[apes, reptiles, rats]	–
T5(base)	[animals, plants, animals]	–
T5(multi-task)	[animals, animals, plants]	–
T5(contrast)	[animals, plants, animals]	–
T5(beam)	[animals, mammals, dinosaurs]	–
T5(clustering)	[animals]	–
GPT-3(clustering)	[elephants, monkeys, cats]	–
TinyLlama	[chickens, cars, fish]	–
Mistral(k-NN)	[dogs, dolphins, parrots]	–
Qwen(k-NN)	[dogs, chimpanzees, parrots]	–
GPT-3(random)	[plants, animals, machines]	–
GPT-3(k-NN)	[birds, dolphins, chimpanzees]	–
GPT-3(COT-random)	[plants, animals, machines]	[The question is asking about which entity has the greatest ability to learn. The answer is humans because humans are known for their advanced cognitive abilities and capacity for learning and adaptation. The distractors plants, animals, and machines are not as capable of learning as humans are, making them incorrect answers.]
GPT-3(COT-k-NN)	[dogs, birds, dolphins]	[The question is asking about the greatest ability to learn, which implies a high level of cognitive ability. Humans are known for their advanced cognitive abilities and capacity for learning, making them the correct answer. Dogs, birds, and dolphins are also known for their ability to learn, but humans are generally considered to have the greatest ability to learn due to their advanced cognitive capabilities and complex societies. Therefore, the correct answer is humans as they have the greatest ability to learn compared to the other options.]

Example: G.2
Dataset: ARC-Challenge
Question [Answer]: Which items would take the least amount of time to decompose in nature? [cut grass]

Model	Generated Distractors	Rationale
Ground Truth	[paper cups, newspapers, cut down trees]	–
T5(base)	[a sandbag, ice cubes, leaves]	–
T5(multi-task)	[sand, ice, water]	–
T5(contrast)	[grass, mosses, leaves]	–
T5(beam)	[sand, weeds, moss]	–
T5(clustering)	[ice, weeds, a tree]	–
GPT-3(clustering)	[glass bottles, aluminum cans, styrofoam]	–
TinyLlama	[a dead bird, a mushroom, a tree trunk]	–
Mistral(k-NN)	[plastic bags, aluminum cans, glass bottles]	–
Qwen(k-NN)	[plastic bottles, aluminum cans, cardboard boxes]	–
GPT-3(random)	[plastic bottles, aluminum cans, glass bottles]	–
GPT-3(k-NN)	[plastic bottles, aluminum cans, glass bottles]	–
GPT-3(COT-random)	[plastic water bottle, aluminum can, glass bottle]	[Cut grass would take the least amount of time to decompose in nature because it is an organic material and can be broken down by natural processes such as bacteria and fungi. In contrast, plastic water bottles, aluminum cans, and glass bottles are all non-biodegradable materials and would take much longer to decompose, if at all. Therefore, cut grass is the correct answer as it would decompose the fastest in nature.]
GPT-3(COT-k-NN)	[plastic bottles, aluminum cans, glass bottles]	[Cut grass would take the least amount of time to decompose in nature because it is an organic material that can easily break down with the help of microorganisms and other decomposers. Plastic bottles, aluminum cans, and glass bottles are all non-biodegradable materials that can take hundreds of years to decompose in nature, making them much slower to break down compared to organic materials like cut grass.]

Table 14: Representative examples (**G.1**, **G.2**) of distractor generation models across selected datasets. Underline indicates cases where the generated distractor matches the correct answer.

Example: G.3
Dataset: MCQ
Question [Answer]: The average human body contains 5,830 g of [blood]

Model	Generated Distractors	Rationale
Ground Truth	[muscle, bacteria, water]	–
T5(base)	[urine, saliva, lungs]	–
T5(multi-task)	[urine, urine, saliva]	–
T5(contrast)	[urine, fat, urine]	–
T5(beam)	[urine, blood, saliva]	–
T5(clustering)	[liver, fat, bone]	–
GPT-3(clustering)	[water, fat, calcium]	–
TinyLlama	[sand, water, bones]	–
Mistral(k-NN)	[water, muscle tissue, bone]	–
Qwen(k-NN)	[water, fat, bone]	–
GPT-3(random)	[water, bone, muscle]	–
GPT-3(k-NN)	[bone, muscle, fat]	–
GPT-3(COT-random)	[water, bone, muscle]	[The question is asking for the substance that the average human body contains 5,830 g of. The correct answer is blood as it is a substance found in the human body and the average amount is specified in the question. The distractors water, bone, and muscle are also substances found in the human body, but the question specifically mentions the amount of a substance, which is blood in this case.]
GPT-3(COT-k-NN)	[water, fat, muscle]	[The average human body contains approximately 7-8% of its weight as blood, which equates to around 4.5-6 liters of blood in an average adult. Water is a major component of the human body, but the average human body contains around 60% water, not 5,830 g specifically. Fat and muscle are also major components of the human body, but their average quantities in the body do not match the specific weight mentioned in the question. Therefore, the correct answer is blood.]

Example: G.4
Dataset: SciQ
Question [Answer]: A bee will sometimes do a dance to tell other bees in the hive where to find what? [food]

Model	Generated Distractors	Rationale
Ground Truth	[enemies, water, honey]	–
T5(base)	[water, food, water]	–
T5(multi-task)	[water, shelter, food]	–
T5(contrast)	[honey, eggs, shelter]	–
T5(beam)	[water, food, pollen]	–
T5(clustering)	[water, nectar, insects]	–
GPT-3(clustering)	[shelter, drones, nectar]	–
TinyLlama	[flowers, trees, safety]	–
Mistral(k-NN)	[water, nectar, pollen]	–
Qwen(k-NN)	[water, sunlight, nectar]	–
GPT-3(random)	[water, shelter, pollen]	–
GPT-3(k-NN)	[water, shelter, pollen]	–
GPT-3(COT-random)	[water, shelter, pollen]	[The behavior of a bee doing a dance to communicate with other bees in the hive is known as the waggle dance. The purpose of the waggle dance is to communicate the location of a food source, such as nectar or pollen, to other members of the hive. Therefore, the correct answer is food. The distractors water, shelter, and pollen are not typically communicated through the waggle dance.]
GPT-3(COT-k-NN)	[water, shelter, predators]	[Bees communicate the location of food sources to other bees through a dance known as the waggle dance. The dance conveys information about the direction and distance of the food source. Therefore, the purpose of the dance is to tell other bees where to find food. The distractors provided (water, shelter, predators) do not accurately represent the information communicated through the waggle dance, as it specifically relates to the location of food sources.]

Table 15: Representative examples (**G.3**, **G.4**) of distractor generation models across selected datasets. Underline indicates cases where the generated distractor matches the correct answer.

Example: G.5
Dataset: MCQL
Question [Answer]: Main sources of carbohydrates are [green plants]

Model	Generated Distractors	Rationale
Ground Truth	[milk, water, proteins]	–
T5(base)	[red plants, blue plants and green plants alike, yellow plants]	–
T5(multi-task)	[red plants, blue plants and green plants alike, yellow plants]	–
T5(contrast)	[animals, plants, fish]	–
T5(beam)	[red plants, white plants, brown plants]	–
T5(clustering)	[red plants, green vegetables, starch]	–
GPT-3(clustering)	[animal products, fast food, water sources]	–
TinyLlama	[herbivores, red meat, processed foods]	–
Mistral(k-NN)	[animals, fungi, bacteria]	–
Qwen(k-NN)	[fungi, bacteria, soilorganisms]	–
GPT-3(random)	[meat, dairy products, fish]	–
GPT-3(k-NN)	[meat, dairy products, seafood]	–
GPT-3(COT-random)	[meat, dairy products, fish]	[Green plants are the main sources of carbohydrates, as they produce carbohydrates through the process of photosynthesis. Meat, dairy products, and fish are sources of protein and fat, but they are not the main sources of carbohydrates. Therefore, the correct answer is green plants, as they are the primary source of carbohydrates in the diet.]
GPT-3(COT-k-NN)	[meat and dairy products, seafood, nuts and seeds]	[The main sources of carbohydrates are green plants, such as fruits, vegetables, grains, and legumes. These plant-based foods are rich in carbohydrates, providing energy and essential nutrients for the body. Meat and dairy products are not the main sources of carbohydrates, as they primarily provide protein and fats. Seafood is also not a significant source of carbohydrates. Nuts and seeds contain some carbohydrates, but they are not the main sources compared to green plants. Therefore, the correct answer is green plants.]

Example: G.6
Dataset: ARC-Easy
Question [Answer]: George is studying erosion. He learns how rocks break apart over time. Which of these most likely forms when a rock weathers and breaks into small pieces? [sediments]

Model	Generated Distractors	Rationale
Ground Truth	[fossils, crystals, minerals]	–
T5(base)	[igneous rocks, sandstones, sediments]	–
T5(multi-task)	[igneous rocks, sandstones, sediments]	–
T5(contrast)	[crystals, igneous rocks, metamorphic rocks]	–
T5(beam)	[sediments, volcanic eruptions, sand]	–
T5(clustering)	[sand, volcanic activity]	–
GPT-3(clustering)	[fossils, lava, glaciers]	–
TinyLlama	[igneous, metamorphic, mappable rocks]	–
Mistral(k-NN)	[crystals, igneous rocks, metamorphic rocks]	–
Qwen(k-NN)	[magma, fossils, sand dunes]	–
GPT-3(random)	[fossils, minerals, lava]	–
GPT-3(k-NN)	[magma, fossils, minerals]	–
GPT-3(COT-random)	[minerals, fossils, magma]	[Sediments are small pieces of rock that are the result of weathering and erosion. When rocks break apart over time, they form sediments.Minerals are the individual components of rocks and are not formed when rocks break into small pieces. Fossils are the preserved remains or traces of ancient organisms and are not formed directly from the weathering and breaking of rocks. Magma is molten rock beneath the Earth's surface and is not formed from the weathering and breaking of rocks.Therefore, the most likely formation when a rock weathers and breaks into small pieces is sediments.]
GPT-3(COT-k-NN)	[magma, fossils, crystals]	[Sediments: When a rock weathers and breaks into small pieces, these small pieces are called sediments. This is the most likely outcome of the weathering and breaking apart of rocks. Magma: Magma is molten rock beneath the Earth's surface and is not directly formed from the weathering and breaking apart of rocks. Fossils: Fossils are the preserved remains or traces of animals, plants, and other organisms from the past, and are not directly formed from the weathering and breaking apart of rocks. Crystals: Crystals are formed through the process of crystallization from a liquid or gas, and are not directly formed from the weathering and breaking apart of rocks. Therefore, the most likely form when a rock weathers and breaks into small pieces is sediments.]

Table 16: Representative examples (**G.5**, **G.6**) of distractor generation models across selected datasets. Underline indicates cases where the generated distractor matches the correct answer.

Example: G.7
Dataset: MedQA
Question [Answer]: A 24-year-old pregnant woman at 28 weeks gestation presents to the emergency department with complaints of fever with chills and pain in her knee and ankle joints for the past 2 days. She also complains of headaches and difficulty moving her neck. Further questioning reveals that she had a tick bite on her arm while gardening a few days ago. Past medical history is noncontributory. She takes a multivitamin with iron and folate every day and has been receiving regular prenatal care and the pregnancy is progressing normally. On examination, an erythematous rash is seen on her right arm, as shown in the accompanying photograph. Her obstetric examination is normal. Ultrasound of the fetus is reassuring with a normal heartbeat and no gross abnormalities. A specimen is collected to test for Lyme disease. What is the next best step for this patient? [Amoxicillin]

Model	Generated Distractors	Rationale
Ground Truth	[Ibuprofen, Doxycycline, Tetracycline, Gentamicin]	–
T5(base)	[Amoxicillin, Vaccination, Lyme disease screening, Immunoglobulin]	–
T5(multi-task)	[Amphotericin B, Vaccination against Lyme disease, Antibiotics, Vitamin B12]	–
T5(contrast)	[Antibiotics, Iron supplementation, Antistreptolysin O titer, Lysosomal trafficking screen]	–
T5(beam)	[Ceftriaxone, Amphotericin B, Azithromycin]	–
T5(clustering)	[Ceftriaxone, Amphotericin, Antibiotics]	–
GPT-3(clustering)	[Ceftriaxone, Azithromycin, Penicillin G]	–
TinyLlama	[No treatment indicated, Penicillin, Doxycycline]	–
Mistral(k-NN)	[Doxycycline, Cefotaxime, Azithromycin]	–
Qwen(k-NN)	[Ceftriaxone, Azithromycin, Metronidazole]	–
GPT-3(random)	[Doxycycline, Ceftriaxone, Acetaminophen, Prednisone]	–
GPT-3(k-NN)	[Doxycycline, Ceftriaxone, Azithromycin, Penicillin]	–
GPT-3(COT-random)	[Doxycycline, Ceftriaxone, Acetaminophen, Ibuprofen]	[The patient's symptoms, including fever, joint pain, headaches, and neck stiffness, along with the history of a tick bite and the presence of an erythematous rash at the site of the bite, are highly suggestive of Lyme disease. The next best step for this patient is to start treatment with amoxicillin, which is the first-line antibiotic for the treatment of Lyme disease in pregnant women. Doxycycline is contraindicated in pregnancy as it can affect fetal bone and teeth development. Ceftriaxone is not the first-line treatment for Lyme disease in pregnancy. Acetaminophen and ibuprofen may help with symptom management, but the priority is to start antibiotic treatment for the suspected Lyme disease. In conclusion, the next best step for this patient is to start treatment with amoxicillin for suspected Lyme disease.]
GPT-3(COT-k-NN)	[Doxycycline, Ceftriaxone, Azithromycin, Penicillin]	[The patient's history of a tick bite, along with the symptoms of fever, joint pain, headaches, and the characteristic erythematous rash, are highly suggestive of Lyme disease. The next best step for this patient would be to start treatment with amoxicillin, which is the recommended antibiotic for treating Lyme disease in pregnant women. Doxycycline is contraindicated in pregnancy as it can affect fetal bone and teeth development. Ceftriaxone is not the first-line treatment for Lyme disease in pregnancy. Azithromycin is not the first-line treatment for Lyme disease, and penicillin is not the preferred antibiotic for treating Lyme disease.]

Table 17: Representative example (**G.7**) of distractor generation models across selected datasets. Underline indicates cases where the generated distractor matches the correct answer.

Example: G.8
Dataset: MedQA
Question [Answer]: The parents of a 14-year-old patient are concerned and have questions about the use of insulin for their son's recently diagnosed type 1 diabetes. The patient has developed an upper respiratory infection while at school. He is coughing and has a runny nose. His temperature is 37.8 °C (100.2 °F) and vital signs are within normal limits. Physical examination is unremarkable. Which of the following modifications to his insulin regimen would you recommend to this patient and his parents? [Increase the frequency of blood glucose checks]

Model	Generated Distractors	Rationale
Ground Truth	[Reduce the insulin dose, Continue same regimen, Hold insulin until the patient gets better, Increase the insulin dose to double]	–
T5(base)	[Increase the frequency of blood glucose checks, Increase dietary intake of glucagon, Obtain a blood sugar level]	–
T5(multi-task)	[Increase the frequency of blood glucose checks, Increase his daily insulin intake, Add a small amount of glycemic control]	–
T5(contrast)	[Increase the frequency of insulin checks, Add a small amount of water per day to his insulin regimen, Switch to glucagon]	–
T5(beam)	[Increase the frequency of blood glucose checks, Increase the frequency of insulin doses, Increase the duration of insulin therapy]	–
T5(clustering)	[Increase the frequency of insulin intake, Increase the amount of insulin he takes]	–
GPT-3(clustering)	[Use insulin only when the patient is physically active]	–
TinyLlama	[Increase the frequency of blood glucose checks, Administer insulin every 3 hours, Administer insulin every hour and provide pre-meal]	–
Mistral(k-NN)	[Decrease the dose of insulin, Stop insulin therapy temporarily, Continue insulin therapy as prescribed without modification]	–
Qwen(k-NN)	[Decrease all insulin doses by 20%, Switch from rapid-acting to long-acting insulin temporarily, Stop all insulin until symptoms resolve]	–
GPT-3(random)	[Increase the dose of long-acting insulin, Switch to oral hypoglycemic agents, Decrease the frequency of insulin injections, Increase the dose of short-acting insulin]	–
GPT-3(k-NN)	[Increase the insulin dosage, Decrease the insulin dosage, Switch to a different type of insulin, Stop insulin therapy temporarily]	–
GPT-3(COT-random)	[Increase the dose of insulin, Switch to a different type of insulin, Decrease the frequency of blood glucose checks, Stop insulin temporarily]	[The patient has developed an upper respiratory infection, which can affect blood glucose levels and insulin requirements. It is important to increase the frequency of blood glucose checks to monitor for any changes in blood sugar levels due to the illness. Increasing the dose of insulin without monitoring blood glucose levels could lead to hypoglycemia. Switching to a different type of insulin may not be necessary if the patient's blood glucose levels can be managed with increased monitoring. Decreasing the frequency of blood glucose checks would not be appropriate, especially during an illness. Stopping insulin temporarily is not recommended, as it could lead to uncontrolled hyperglycemia.Therefore, the most appropriate modification to the insulin regimen would be to increase the frequency of blood glucose checks to closely monitor the patient's blood sugar levels during the illness.]
GPT-3(COT-k-NN)	[Increase the insulin dosage, Switch to a different type of insulin, Decrease the frequency of blood glucose checks, Stop insulin therapy temporarily]	[The patient has developed an upper respiratory infection, which can affect blood sugar levels and insulin requirements. It is important to increase the frequency of blood glucose checks to closely monitor the patient's blood sugar levels during the illness. Increasing the insulin dosage without proper monitoring can lead to hypoglycemia, so it is not the most appropriate recommendation in this situation. Switching to a different type of insulin may not be necessary at this time, as the focus should be on closely monitoring blood sugar levels. Decreasing the frequency of blood glucose checks is not recommended, especially during an illness when blood sugar levels can fluctuate. Therefore, the most appropriate modification to the insulin regimen would be to increase the frequency of blood glucose checks to closely monitor the patient's blood sugar levels during the upper respiratory infection.]

Table 18: Representative example (**G.8**) of distractor generation models across selected datasets. Underline indicates cases where the generated distractor matches the correct answer.