Fine-Tuning Encoder-Decoder Models with Contrastive Learning for In-Context Distractor Generation

Elaf Alhazmi[1,3], **Quan Z. Sheng**[1], **Wei Emma Zhang**[2], **Mohammed I. Thanoon**[3]
Haojie Zhuang[2], **Behnaz Soltani**[1], **Munazza Zaib**[1]

[1]School of Computing, Macquarie University, Australia
[2]School of Computer and Mathematical Sciences, The University of Adelaide, Australia
[3]College of Engineering and Computing in Al-Lith, Umm Al-Qura University, Saudi Arabia
elaf.alhazmi@hdr.mq.edu.au, michael.sheng@mq.edu.au
wei.e.zhang@adelaide.edu.au, {eafhazmi, mithanoon}@uqu.edu.sa

Abstract

Distractor generation is the task of automatically generating plausible yet incorrect options (i.e., distractors) for fill-in-the-blank and multiple-choice questions. In assessment, distractors must be contextually relevant to the given question and answer. Even though recent research works focus on fine-tuning pre-trained encoder-decoder models with data augmentation techniques to generate distractors, these models often fail to capture the full semantic representation of a given question-answer and related distractors. The augmentation methods often rely on expanding the quantity of proposed candidates (i.e., questions or distractors), which can introduce noise into the models without necessarily enhancing their understanding of the deeper semantic relationships between question-answer and related distractors. This paper introduces a novel distractor generation model based on contrastive learning to train the model to recognize essential semantic features necessary to generate in-context distractors. The extensive experiments on two public datasets indicate that contrastive learning introduces a strong baseline model to the distractor generation task. It significantly outperforms recent models, increasing the NDCG@3 score from 24.68 to 32.33 on the MCQ dataset and from 26.66 to 36.68 on the SciQ dataset.

1 Introduction

In assessments, objective questions (Das et al., 2021), including multiple-choice and fill-in-the-blank questions, are widely used in education for fair evaluation across various domains and subjects (Ch and Saha, 2018; Kurdi et al., 2020). These questions require an examinee to select one correct answer from a set of wrong options. The quality of these questions relies on the quality of selecting these wrong options, known as *distractors*. Distractor generation (DG) (Alhazmi et al., 2024) refers to the automated process of generating plausible yet incorrect options in objective types of questions.

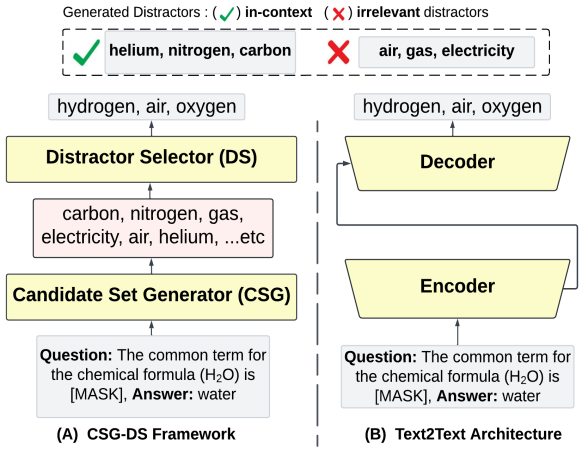

Figure 1: Distractor generation methods via pre-trained language models. On the left, **CSG-DS** refers to the candidate generation and selection framework. On the right, **Text2Text** represents the sequence-to-sequence generation task by pre-trained encoder-decoder models.

For decades, research works have shown interest in DG using several approaches, ranging from feature-based learning (Liang et al., 2018) to deep neural networks (Gao et al., 2019; Maurya and Desarkar, 2020). Then, pre-trained language models (PLMs), based on Transformer architecture (Vaswani et al., 2017), have notably enhanced DG through fine-tuning (Yu et al., 2024) and prompting (Doughty et al., 2024) paradigms.

Two primary methods have been proposed for DG based on PLMs, as illustrated in Figure 1. First, *candidate generation and selection framework* (CSG-DS) (Chiang et al., 2022) uses fine-tuning or prompting methods to generate a candidate set of distractors, then selects the top distractors based on embedding models or feature-based rules (Taslimipoor et al., 2024). Second, *Text2Text* architecture (Wang et al., 2023) utilizes fine-tuning pre-trained encoder-decoder models (Raffel et al., 2020; Lewis et al., 2020) to generate distractors as a sequence-to-sequence (Seq2Seq) task, where the distractors are generated directly from PLMs rather than selected from the generated candidates.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 10056–10072
November 4-9, 2025

Aligning Text2Text models with the DG task is still challenging. As shown in Figure 1, they may generate distractors (air, gas, electricity) that are only plausible to the answer (water) rather than in-context with the question (the common term for the chemical formula H_2O is ...etc).

Recent state-of-the-art (SOTA) Text2Text works incorporate augmentation techniques (Wang et al., 2023) and adopt a retrieval augmented pre-training method (Yu et al., 2024) to enhance the knowledge of the encoder-decoder models, but these models are designed to restore and denoise entire text sequences during pre-training, rather than capturing fine-grained semantic distinctions required for generating contextually relevant distractors. This limitation causes the models to generate distractors (air, gas, electricity) that simply complete the denoised sequence for the given question-answer, rather than in-context distractors (helium, nitrogen, carbon) that semantically align with the question-answer, as shown in Figure 1. Thus, we propose to integrate a contrastive learning (CL) approach inspired by computer vision and text generation works (Li et al., 2020; Radford et al., 2021; Zhang et al., 2022a; Dong et al., 2023; Zhuang et al., 2024) to enhance the semantic learning in these models.

While CL is applied to the DG in ranking-based models (Bitew et al., 2022), visual question answering task (Ding et al., 2024), knowledge graph reasoning (Guo et al., 2024), and most recently still under-explored in reading comprehension (Dong et al., 2025), it is not yet explored, particularly for the DG task, within the Text2Text architecture.

Initially, the encoding of the input (i.e., question–answer) and the decoding of the output (i.e., distractors) can be regarded as two representational views with respect to the same semantics, forming a positive pair. They are contrasted with negative pairs (i.e., a given question-answer with unrelated distractors in the mini-batch). Integrating a contrastive loss with the generation loss encourages the model to bring semantically similar pairs closer in the representation space while pushing dissimilar pairs. This joint learning enhances the model to capture semantic features and generate in-context distractors. We explore two contrastive objectives: *InfoNCE* (Oord et al., 2018) with multiple negatives and *Triplet* loss (Schroff et al., 2015) with a single negative. Our automatic and human evaluation results on two public datasets indicate that this method aligns the DG task with encoder-decoder models better than augmentation techniques.

The main contributions can be summarized as follows: (i) introducing a contrastive-based learning approach within Text2Text architecture for the DG task (ii) benchmarking our approach against SOTA models using both automatic and human evaluation metrics; (iii) achieving new SOTA results, increasing NDCG@3 score from 24.68 to 32.33 on MCQ and from 26.66 to 36.68 on SciQ.

We organize this paper as follows. Sec. 2 reviews the related works on DG and CL. Sec. 3 presents the details of the proposed methodology. Sec. 4 reports the experimental details along with performance analysis, and Sec. 5 offers a conclusion.

2 Related Work

2.1 Distractor Generation (DG)

The tasks in DG are typically divided into two formats: *multiple-choice questions* (MCQs) and *fill-in-the-blank* (FITB). They have been applied in textual (Xie et al., 2018) and multi-modal (Yagcioglu et al., 2018) aspects. These tasks have also been explored in question answering (Liang et al., 2017, 2018), reading comprehension (Xie et al., 2021; Qu et al., 2024), and multi-modal question answering (Zhu et al., 2016; Luo et al., 2024).

Over the years, the field of DG has progressed significantly in methodologies, transitioning from conventional techniques to cutting-edge artificial intelligence approaches. Initially, conventional methods include the use of corpus features (Chen et al., 2006), phonetic and morphological features (Pino and Eskenazi, 2009), knowledge-based structures (Mitkov et al., 2003, 2009), and word embedding models (Guo et al., 2016; Yoshimi et al., 2023).

Recently, transformer-based PLMs have revolutionized DG tasks. The two main approaches proposed for the DG in text-based contexts include the CSG-DS framework and the Text2Text architecture. Ren and Zhu (2021) proposed using knowledge-based structures such as Probase (Wu et al., 2012) and WordNet (Miller, 1995) to retrieve a small set of candidates, followed by a feature-rich learning-to-rank model to identify the top distractors. Chiang et al. (2022) utilized PLMs, which showed significant improvements, to generate the candidates as compared to knowledge-based structures. Taslimipoor et al. (2024) proposed using the pre-trained encoder-decoder model for generating both correct and incorrect options, and then discriminate between options with a classifier. The generated options are then clustered to remove duplicates.

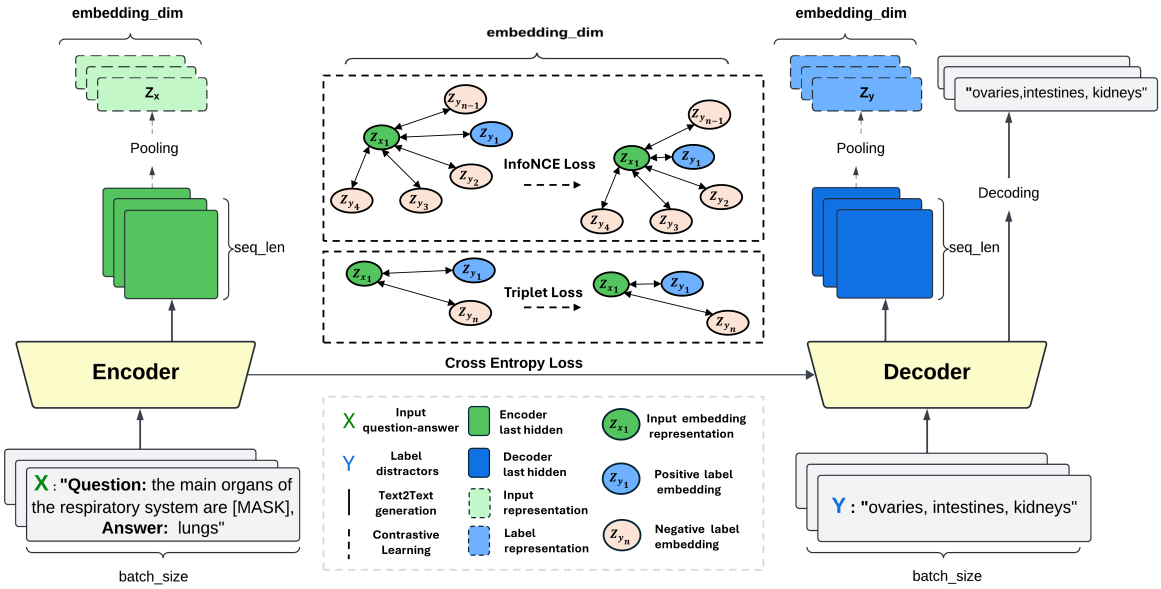

Figure 2: The training pipeline describes the integration of contrastive learning (CL) in Text2Text distractor generation (DG). Solid arrows represent the data flow of generation loss, and dashed arrows indicate the data flow of CL. Our approach outlines two-stage training. The first stage trains the model as generation task, including only cross-entropy loss. The second stage trains the model with both cross-entropy loss and one of the CL losses.

Wang et al. (2023) framed DG as a Text2Text architecture by fine-tuning encoder-decoder models and used data augmentation to reduce repeated distractor generation. Then, Yu et al. (2024) applied retrieval-augmented pre-training and used knowledge graph triplets for data augmentation. Unlike prior methods, we address the challenge of fine-grained semantic learning by integrating CL.

2.2 Contrastive Learning in NLP

CL is a machine learning technique that trains models to distinguish between semantically similar and dissimilar data pairs (Chopra et al., 2005; Hadsell et al., 2006). It has shown success in various domains, starting with computer vision to NLP tasks.

Initially, Schroff et al. (2015) proposed the FaceNet system that trains face recognition and clustering based on triplet loss learning, while Sohn (2016) proposed multi-class N-pair loss for a variety of tasks on several visual recognition benchmarks. Chen et al. (2020) introduced the SimCLR framework using data augmentation to generate diverse views of the same image. This approach used a CL objective to ensure that representations from the same source image are similar, while those from different source images remain distinct. Radford et al. (2021) utilized CL to pre-train a vision-language model to align representations between images and their textual descriptions.

Recently, many works applied CL to learn better sentence embeddings (Gao et al., 2021; Wu et al., 2022; Zhang et al., 2022b; Xu et al., 2023). Beyond embeddings, Karpukhin et al. (2020) applied CL to develop an innovative dense passage retrieval strategy for question-passage pairs, advancing the field of open-domain question answering (Zaib et al., 2024). Qin et al. (2021) explored CL to obtain a deeper understanding of the entities and their relations in texts, and Chen et al. (2022) utilized CL to tackle both discriminative representation and overfitting problems in few-shot text classification.

In text generation, CL approach is also recognized for addressing degeneration issues such as undesirable generated content and repetition (Su et al., 2022). It is also applied in various NLP tasks, including machine translation (Pan et al., 2021), definition generation (Zhang et al., 2022a), closed-book question generation (Dong et al., 2023), multi-document question generation (Cho et al., 2021), paraphrase generation (Yang et al., 2021), and summarization (Zhuang et al., 2022, 2024). Although CL is explored in the DG task for multilingual ranking models (Bitew et al., 2022), multi-modal distractor generation (Ding et al., 2024), and graph knowledge-based for commonsense questions (Guo et al., 2024), CL has not yet been particularly explored for the DG task in Text2Text architecture (Wang et al., 2023; Yu et al., 2024).

3 Methodology

This section outlines the details of our approach. Sec. 3.1 defines DG task formulation. Sec. 3.2 and Sec. 3.3 detail the training of Text2Text and the implementation of CL in encoder-decoder models, respectively. Sec. 3.4 presents a two-stage training to incorporate contrastive loss with generation task.

3.1 Task Formulation

Given a query $\mathbf{Q}$ and its answer $\mathbf{A}$, the task of distractor generation involves generating a set sequence of distractors $\mathbf{D} = \{d_1, d_2, \ldots, d_N\}$, where $N > 0$ represents the number of distractors. The generation process is formally defined as:

$$P(D \mid Q, A) = \prod_{t=1}^{N} p(d_t \mid \mathbf{d}_{<t}, \mathbf{Q}, \mathbf{A}) \quad (1)$$

where d_t represents the sequence of letters in the t-th distractor, $\mathbf{d}_{<t}$ denotes the sequences of all distractors generated before d_t.

3.2 Text2Text Generation

For each training instance (Q, A, D), the objective is to fine-tune a generative model, which is conditioned on the given query Q and the answer A, aiming to minimize the negative log-likelihood for each correct token t_i in the sequence D, based on its preceding tokens and the given conditions, where the generation loss function is defined as:

$$\mathcal{L}_g = -\sum_{i=1}^{|D|} t_i \log p(\hat{t}_i \mid \hat{t}_{<i}, Q, A, \theta) \quad (2)$$

As depicted in Figure 2, the input consists of the query Q and the answer A, with the prefix "Question: " before the given query and "Answer: " before the given answer. The generated output is a sequence of distractors, expressed as $d_1 \oplus d_2 \oplus d_3$.

3.3 Contrastive Loss

CL aims to optimize semantic representations by pulling positive pairs closer in feature space while pushing negative pairs further apart. In DG models, this process requires an understanding of the semantics of a question, an answer, and their relationships with related ground-truth distractors. Initially, the encoder takes an input sequence of source words $x = (x_1, x_2, \ldots, x_n)$, which includes the given question and answer as illustrated in Figure 2. The encoder then maps x to a sequence of continuous representations $z = (z_1, z_2, \ldots, z_n)$.

Subsequently, the decoder utilizes z to generate a sequence of target words, which are the sequence of distractors $y = (y_1, y_2, \ldots, y_m)$ at a time. The question-answer encoding should be semantically similar to its ground-truth distractors and dissimilar to incorrect distractors. The objective is to develop a similarity function that minimizes the distance between the question-answer sequence and the representations of its correct distractors, enhancing the model to generate relevant in-context distractors.

First, we implement the *InfoNCE* contrastive loss in the representation space to enhance model training. For a positive pair $S = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i (question-answer) and y_i (distractors) represent semantically related inputs, we treat the remaining $(n-1)$ examples within a mini-batch as negative examples. The training loss objective for each pair (x_i, y_i) is:

$$\mathcal{L}_c = -\log \frac{e^{\mathrm{d}(\mathbf{z}_{x_i}, \mathbf{z}_{y_i})/\tau}}{\sum_{j=1}^{n} e^{\mathrm{d}(\mathbf{z}_{x_i}, \mathbf{z}_{y_j})/\tau}} \quad (3)$$

where z_{x_i} and z_{y_i} are the embedding representations of an input question-answer x_i and its related distractors y_i, respectively. Here, z_{y_j} refers to the embedding representation of the distractors y_j in the mini-batch. $\mathrm{d}(z_i, z_j)$ denotes the cosine similarity, and τ is a temperature parameter.

$$\mathrm{d}(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|} \quad (4)$$

Second, we implement the *Triplet* contrastive loss in the representation space. For each positive pair $S = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i and y_i are semantically related inputs, we randomly select a negative example n_j from the mini-batch, ensuring $j \neq i$. The training loss objective for the (x_i, y_i, n_j) is:

$$\mathcal{L}_t = \max(\mathrm{d}(\mathbf{z}_{x_i}, \mathbf{z}_{y_i}) - \mathrm{d}(\mathbf{z}_{x_i}, \mathbf{z}_{n_j}) + m, 0) \quad (5)$$

where $\mathbf{z}_{x_i}$, $\mathbf{z}_{y_i}$, and $\mathbf{z}_{n_j}$ represent the semantic embeddings of the anchor, positive, and negative, examples respectively. Here, $\mathbf{z}_{x_i}$ and $\mathbf{z}_{y_i}$ are semantically similar, whereas $\mathbf{z}_{n_j}$ is semantically dissimilar. The margin m ensures a minimum distance between the anchor-positive pairs and the anchor-negative pairs. The distance function d can be either implemented with *cosine similarity* (c.s) in Eq 4 or *Euclidean distance* (e.d) in Eq 6.

$$\mathrm{d}(\mathbf{z}_i, \mathbf{z}_j) = \sqrt{\sum_{k=1}^{d} (\mathbf{z}_i[k] - \mathbf{z}_j[k])^2} \quad (6)$$

3.4 Overall Two-Stage Training

The training approach combines both generation loss with a newly implemented contrastive objective loss. As illustrated in Figure 2, the model is fine-tuned using solely the generation loss (i.e., Text2Text generation) as the first stage. In the subsequent stage, contrastive loss is fine-tuned with the generation loss, optimizing the model with a mixed loss function $\mathcal{L}_{Final}$:

$$\mathcal{L}_{Final} = \lambda_g * \mathcal{L}_g + \lambda_c * \mathcal{L}_{cl} \quad (7)$$

As described in Sec. 3.3, the contrastive objective $\mathcal{L}_{cl}$ can be implemented as either the InfoNCE loss $\mathcal{L}_c$ or the Triplet loss $\mathcal{L}_t$. Here, λ_g and λ_c serve as a hyper-parameters to balance the generative and contrastive types of losses, respectively. The two-stage training enables the model to learn semantic information from a given question-answer (the term for the chemical formula H_2O is ...etc) and its related distractors (hydrogen, air, oxygen), as shown in Figure 1. The given question-answer and its valid distractors forms a positive pair, while pairing the same question-answer with randomly sampled unrelated distractors (spring, fall, summer) from the mini-batch forms a negative pair. By jointly applying a semantic learning objective such as Triplet loss alongside the generation loss, the training process encourages the models to minimize the distance between the embeddings of positive pairs, to capture semantic features, and maximize the distance from negative pairs. This encourages the model to generate in-context distractors (helium, nitrogen, carbon) that are semantically aligned with the question context (i.e., chemical elements) rather than less relevant distractors (air, gas, electricity) that are only plausible with the answer (water). This semantic learning is essential in encoder-decoder models for the DG task.

4 Experiments

4.1 Datasets

We conduct the experiments on the SciQ (Welbl et al., 2017) and MCQ (Ren and Zhu, 2021) datasets as outlined the statistics in Table 1.

SciQ contains crowd-sourced multiple-choice questions from natural sciences, each with one correct answer and three distractors. Average token in options is 1.6, word-level, and 14.5 in questions. In test data, we remove few articles (e.g., a, an) from the answers or distractors.

MCQ or Dgen[1] dataset, contains fill-in-the-blank questions with one correct answer and three distractors, collected from various sources. We replace "**blank**" with a $[MASK]$ token. The questions cover science, vocabulary, commonsense, and trivia. Average token in options is 1, and 19.5 in questions. We use 80% of the 2,321 questions for training and 20% for validation.

Datasets	Train	Valid	Test	All
SciQ	11,700	1,000	1,000	13,700
MCQ	1,856	465	259	2,580

Table 1: Statistics of SciQ and MCQ datasets.

4.2 Baselines Models

Text2Text Architecture: we fine-tune T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) as base models. We also incorporate multi-task learning, candidate augmentation, and retrieval augmented pre-training methods (Wang et al., 2023; Yu et al., 2024). App. A covers the details of the methods.

CSG-DS: we fine-tune T5 for generating distractor candidates. Then, we utilize two selection methods: beam search (Gao et al., 2019) and clustering (Taslimipoor et al., 2024). Beam search is utilized to select the top three predicted distractors from a set of ten. For clustering, we utilize agglomerative clustering[2] with Euclidean distance to measure the similarity between clusters, setting a threshold of 1.2. The heads of different clusters are then selected as the final set of distractors.

Prompting: we utilize one random example for one-shot (Bitew et al., 2023) and three random examples for few-shot (Feng et al., 2024) learning to generate three distractors per a query. An example includes a query and three distractors.

4.3 Evaluation Metrics

For *automatic* evaluation, we utilize ranking-based metrics that measure the models ability to retrieve relevant distractors from the top-k locations, as used in previous DG studies (Ren and Zhu, 2021; Yu et al., 2024). *Order-unaware* metrics, include F1 score (F1@3), precision (P@1, P@3), and recall (R@1, R@3). We also include *order-aware* metrics such as mean reciprocal rank (MRR@K) and normalized discounted cumulative gain (NDCG@K).

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

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

We utilize *human* evaluation metrics to assess the model performance. They include, *relevance* to assess if the distractors are relevant to the context of the query, *difficulty* to evaluate the level of distraction provided in finding the correct answer, and *fluency* to determine if the distractors are not duplicated and semantically different. We randomly select twenty examples from both datasets, that were assessed by five human participants, each having more than two years of academic experience. We use a five-point quantitative rating system from 1 (strongly irrelevant) to 5 (strongly relevant).

4.4 Implementation Details

Our models are built using Hugging Face frameworks (Wolf et al., 2020), including T5 and BART as generative models. We optimize using AdamW, with initial learning rates of 1e-4 for T5 and 2e-5 for BART. We conduct the experiments on two NVIDIA Tesla P100 GPUs. The T5 model trains for 10 epochs and the BART model for 20, both with a batch size of 4. For InfoNCE, the temperature τ is set at 0.1, and in the Triplet, the margin m is set at 0.01. The weights λ_g and λ_c are both set at 0.5, and mean pooling is used as the standard pooling method for embedding dimensions. We implement the two-stage training and utilize the gpt-3.5-turbo model for prompting[3].

4.5 Evaluation Results

4.5.1 Automatic Evaluation Results

Table 2 shows automatic evaluation results for various models on both datasets. A two-stage training with either InfoNCE or Triplet loss significantly enhances the performance of DG in Text2Text architecture compared to all recent SOTA approaches.

Initially, the contrastive-based approach outperforms recent Text2Text methods across all metrics in both datasets. In T5 model, retrieval augmented pre-training improves the NDCG@3 score from 24.68 to 27.44 on MCQ, and multi-task learning increases the scores from 26.66 to 28.52 on SciQ, but T5 with InfoNCE loss achieves the best NDCG@3 scores. The model raises the scores to 32.33 on MCQ and 36.68 on SciQ. Then, Triplet loss, especially with Euclidean distance, achieves the second-best NDCG@3 scores, which increase to 30.46 on MCQ and 35.25 on SciQ. This illustrates the benefits of batch-wide optimization in InfoNCE that uses multiple negative examples rather than Triplet, which relies only on one negative example. It is worth mentioning that the contrastive-based approach, including both InfoNCE and Triplet objectives, shows a performance increase across all automatic metrics in both datasets, but the retrieval augmented pre-training approach only shows an increase on MCQ rather than SciQ.

Also, contrastive-based approach demonstrates successful results in BART model. These improvements in automatic metrics highlight the effectiveness of contrastive learning in aligning pre-trained encoder-decoder models to DG task. Furthermore, the choice of distance metric in Triplet loss plays a critical role in performance. Euclidean distance demonstrates significantly better results compared to cosine similarity across both models and datasets. For example, on the SciQ dataset, T5 model achieves an NDCG@3 score of 33.72 with Triplet loss using cosine similarity, whereas using Euclidean distance increases the score to 35.25. However, while BART-Contrast(Triplet)/e.d occasionally outperforms BART-Contrast(InfoNCE) on metrics such as P@1, R@1, and F1@3, the contrastive-based approaches, including both InfoNCE and Triplet losses, consistently surpass recent Text2Text methods, including multi-task training, retrieval augmented pre-training, and candidate augmentation, in both datasets and models.

Furthermore, contrastive-based approach outperforms prompting methods such as one-shot and few-shot. The approach also exceeds the performance of candidate generation with beam search selection. In fact, CSG-DS(beam) shows remarkable results in both datasets. Even though it yields higher NDCG@3 scores (i.e., 30.11 on MCQ and 31.30 on SciQ) than T5-base, both scores in T5 model with InfoNCE (i.e., 32.33 on MCQ and 36.68 on SciQ) and Triplet/e.d (i.e., 30.46 on MCQ and 35.25 on SciQ) objectives surpass CSG-DS(beam). This improvement underscores the effectiveness of contrastive learning in enhancing the relevance and ranking quality of generated distractors over recent SOTA approaches. Therefore, these results highlight contrastive learning as a strong and promising approach for DG task. We provide analysis on hyper-parameters in App. B.

4.5.2 Human Evaluation Results

Table 3 presents human evaluation results for the DG task across ten models on both datasets. Initially, T5 with contrastive objective InfoNCE scores the highest across all human evaluation met-

[3] https://github.com/contrastivelearningDG/contrastive_learning_in_encoder_decoder_models

Dataset	Approach	Model	P@1	R@1	F1@3	MRR	NDCG@3
MCQ	Text2Text	BART-base	8.49	2.83	10.55	14.99	20.60
		BART-Contrast(Triplet)/c.s	11.58	3.86	11.58	18.53	24.51
		BART-Contrast(Triplet)/e.d	15.44	5.15	13.13	22.84	29.20
		BART-Contrast(InfoNCE)	13.90	4.63	12.61	21.24	27.71
		BART(Multi-task)	5.79	1.93	9.65	13.06	19.50
		BART(CA)	6.18	2.06	7.46	11.58	16.31
		BART(RAP)	8.49	2.83	9.78	14.09	18.99
		T5-base	14.29	4.76	10.81	20.14	24.68
		T5-Contrast(Triplet)/c.s	20.46	6.82	13.38	25.61	29.51
		T5-Contrast(Triplet)/e.d	22.01	7.34	14.16	26.96	30.46
		T5-Contrast(InfoNCE)	**22.78**	**7.59**	**15.70**	**28.57**	**32.33**
		T5(Multi-task)	15.06	5.02	12.23	19.95	23.54
		T5(CA)	3.47	1.16	4.50	7.34	10.93
		T5(RAP)	18.15	6.05	14.67	23.49	27.44
	CSG-DS	T5-CG(beam)	17.37	5.79	13.38	24.20	30.11
		T5-CG(clustring)	11.58	3.86	7.72	16.47	20.95
	Prompting	GPT-3(one-shot)	11.19	3.73	9.13	16.57	20.75
		GPT-3(few-shot)	13.89	4.63	11.06	20.07	25.50
SciQ	Text2Text	BART-base	10.50	3.50	12.60	17.77	24.02
		BART-Contrast(Triplet)/c.s	15.50	5.17	15.27	23.67	30.45
		BART-Contrast(Triplet)/e.d	16.30	5.43	15.33	24.15	30.63
		BART-Contrast(InfoNCE)	16.00	5.33	15.43	24.73	32.35
		BART(Multi-task)	11.70	3.90	13.17	18.78	24.71
		BART(CA)	8.20	2.73	11.13	15.72	22.25
		BART(RAP)	9.10	3.03	10.20	15.00	20.06
		T5-base	18.90	6.30	13.77	23.23	26.66
		T5-Contrast(Triplet)/c.s	22.20	7.40	16.60	28.58	33.72
		T5-Contrast(Triplet)/e.d	24.80	8.27	17.50	30.62	35.25
		T5-Contrast(InfoNCE)	**25.00**	**8.33**	**17.73**	**31.42**	**36.68**
		T5(Multi-task)	21.20	7.07	14.27	25.47	28.52
		T5(CA)	11.00	3.67	5.40	13.70	15.98
		T5(RAP)	12.80	4.27	9.30	16.73	19.87
	CSG-DS	T5-CG(beam)	20.30	6.77	14.33	26.20	31.30
		T5-CG(clustering)	10.50	3.49	6.30	14.35	17.95
	Prompting	GPT-3(one-shot)	11.00	3.66	8.69	15.36	19.07
		GPT-3(few-shot)	12.50	4.16	9.63	17.08	21.01

Table 2: Automatic evaluation results for two datasets MCQ and SciQ in three main approaches (e.g., Text2Text, CSG-DS and Prompting). Triplet loss includes either (c.s) cosine similarity or (e.d) Euclidean distance. (CA) refers to candidate augmentation, and (RAP) is retrieval augmented pre-training. The best scores are highlighted in bold.

Model	Relevance	Difficulty	Fluency
T5(InfoNCE)	**4.46**	**4.33**	**4.27**
T5(Triplet)/e.d	4.03	3.58	4.02
GPT-3(few-shot)	3.79	3.55	3.53
T5-CG(beam)	3.32	2.93	2.79
T5-CG(clustering)	2.72	2.42	2.50
T5-base	2.41	2.26	2.18
T5(Multi-task)	2.44	2.27	2.19
T5(CA)	1.73	1.44	1.46
T5(RAP)	2.16	1.89	1.81
Ground-truth	3.81	3.92	3.60

Table 3: Human evaluations in MCQ and SciQ datasets.

rics, and T5 with objective Triplet, especially with Euclidean distance, surpasses ground-truth distractors in relevance and fluency, indicating the benefits of semantic fine-grained training in encoder-decoder models. This improvement shows the success of contrastive-based models in generating in-context distractors in Text2Text architecture.

Then, few-shot learning and candidate generation framework with beam selection method are successfully capable of generating relevant distractors, but they often underperform compared to contrastive-based models due to the absence of fine-grained semantic optimization. It is worth to mention that CSG-DS(beam) usually performs lower than few-shot learning and contrastive-based approaches, including both InfoNCE and Triplet objectives, because it usually generates distractors that may seem plausible to the answer but are not strongly related to the context of the question.

Furthermore, contrastive-based approach significantly outperforms the Text2Text methods, such

Dataset	Model	P@1	R@1	F1@3	MRR	NDCG@3
MCQ	T5-base	14.29	4.76	10.81	20.14	24.68
	T5-Contrast(Triplet)/c.s	20.46	6.82	13.38	25.61	29.51
	T5-Contrast(Triplet)/c.s/one-stage	20.08	6.69	15.06	26.00	30.60
	T5-Contrast(Triplet)/c.s/max	21.62	7.21	15.06	26.83	30.67
	T5-Contrast(Triplet)/e.d	22.01	7.34	14.16	26.96	30.46
	T5-Contrast(Triplet)/e.d/one-stage	20.46	6.82	12.74	25.42	29.05
	T5-Contrast(Triplet)/e.d/max	20.85	6.95	13.26	25.23	28.21
	T5-Contrast(InfoNCE)	**22.78**	**7.59**	**15.70**	**28.57**	**32.33**
	T5-Contrast(InfoNCE)/one-stage	21.24	7.08	14.80	26.64	30.64
	T5-Contrast(InfoNCE)/max	16.99	5.66	11.71	22.46	27.09
SciQ	T5-base	18.90	6.30	13.77	23.23	26.66
	T5-Contrast(Triplet)/c.s	22.20	7.40	16.60	28.58	33.72
	T5-Contrast(Triplet)/c.s/one-stage	23.90	7.97	17.33	30.58	35.91
	T5-Contrast(Triplet)/c.s/max	22.90	7.63	17.03	29.23	34.27
	T5-Contrast(Triplet)/e.d	24.80	8.27	17.50	30.62	35.25
	T5-Contrast(Triplet)/e.d/one-stage	24.80	8.27	17.33	30.33	34.62
	T5-Contrast(Triplet)/e.d/max	24.50	8.17	17.07	30.38	35.28
	T5-Contrast(InfoNCE)	25.00	8.33	**17.73**	**31.42**	**36.68**
	T5-Contrast(InfoNCE)/one-stage	24.90	8.30	17.50	31.23	36.33
	T5-Contrast(InfoNCE)/max	**25.40**	**8.47**	16.70	30.83	35.26

Table 4: Ablation experiment results on both MCQ and SciQ datasets using the T5 model.

as multi-task learning and retrieval augmented pre-training. While multi-task learning shows slightly better results than T5-base, recent Text2Text methods are significantly more vulnerable to generating low-relevance distractors to the context of the question compared to the ground truth. These results demonstrate the essential role of contrastive learning in effectively aligning Text2Text models to DG.

4.6 Ablation Study

We conduct an ablation study as the results outlined in Table 4. We propose two contrastive objectives, and the Triplet loss incorporates cosine similarity or Euclidean distance. We also utilize mean or max pooling functions and a two-stage training strategy.

Replacing the *mean* pooling function with *max* pooling in the T5-Contrast methods using InfoNCE and Triplet loss across both datasets shows different results. In the InfoNCE objective, max pooling generally underperforms compared to mean pooling across most metrics. With Triplet cosine similarity, max pooling slightly improves the performance; but it reduces with Triplet Euclidean distance.

Removing the first stage and directly training the model with the second stage (i.e., contrastive loss and generation loss) in T5 generally shows a decline in performance across all metrics in both datasets, indicating that the complexity of the two-stage process is beneficial for the InfoNCE. Conversely, the one-stage Triplet model with cosine similarity presents improvements in several metrics, particularly in the SciQ dataset, while the one-stage Triplet model with Euclidean distance shows a decline in performance across both datasets. All ablated variants still outperform T5-base in all metrics, indicating the robustness of CL in DG.

4.7 Case Study

Table 5 presents a case study for the generated distractors based on ten models. These models include contrastive-based methods (InfoNCE, Triplet), prompting-based generation (few-shot), candidate set generation and selection frameworks (beam search and clustering), Text2Text approaches (multi-task learning, candidate augmentation, and retrieval-augmented pre-training), and the ground-truth distractors for comparison.

Initially, T5-base model and recent Text2Text approaches obviously generate distractors that lack semantic relevance to the given question and are less similar to the ground-truth distractors. In T5-base model, the distractors (*air*, *light*, *air*) in example (1) might seem plausible to the answer (*water*), but they are not contextually relevant to the question (the common term for the chemical formula H_2O ...etc), that are mostly related to chemical elements. In addition, the distractors (*ice*, *moon*, *planet*) in example (2) might be contextually plausible to the answer (*earth*), but fail to maintain meaningful semantic connection to the subsequent

(1) Question [Answer]	The common term for the chemical formula H_2O is [water]
Models	**Generated Distractors**
T5(InfoNCE)	[helium, nitrogen, carbon]
T5(Triplet)/e.d	[nitrogen, carbon, oxygen]
GPT-3(few-shot)	[carbon dioxide, nitrogen, oxygen]
T5-CG(beam)	[carbon, oxygen, gas]
T5-CG(clustering)	[carbon, air, ions]
T5-base	[air, light, air]
T5(Multi-task)	[air, ice, gas]
T5(CA)	[water]
T5(RAP)	[air, electricity, gas]
Ground-Truth	[hydrogen, air, oxygen]
(2) Question [Answer]	The only known planet with large amounts of water is [earth]
Models	**Generated Distractors**
T5(InfoNCE)	[Mars, Venus, Jupiter]
T5(Triplet)/e.d	[earth, moon, planet]
GPT-3(few-shot)	[Venus, Mars, Mercury]
T5-CG(beam)	[moon, planet, earth]
T5-CG(clustering)	[planet, ice]
T5-base	[ice, moon, planet]
T5(Multi-task)	[moon, moon]
T5(CA)	[earth]
T5(RAP)	[sun, moon, planet]
Ground-Truth	[Saturn, Jupiter, Mars]

Table 5: Examples from MCQ for DG by ten models.

question (the only known planet with large amount of water is ...etc), which are mostly related to the specific name of planets.

Contrastive-based approach shows the benefits of capturing semantic features from given question-answer and distractor pairs to generate in-context relevant distractors. The InfoNCE-based approach presents remarkable distractors in both examples. The distractors (*helium*, *nitrogen*, *carbon*) in example (1) and the distractors (*Saturn*, *Jupiter*, *Mars*) in example (2) are strongly relevant to the context of the questions and closely similar to the ground-truth distractors. Triplet/e.d shows varied success. Unlike example (2), only example (1) output is relevant to the question. This outlines the benefit of InfoNCE loss compared to Triplet loss.

While few-shot learning generates promising distractors in both questions, the contrastive based-approach with InfoNCE objective learning consistently outperforms candidate generation and selection framework, including both beam search and clustering approaches. In addition, InfoNCE-based contrastive objective with Text2Text architecture show remarkable improvement in generating high-quality in-context distractors for both questions compared to recent Text2Text methods. We include additional examples in Table 14 for MCQ and Table 15 for SciQ in App. B.

5 Conclusion

We propose a distractor generation model that integrates contrastive learning in a Text2Text architecture to better train pre-trained encoder-decoder models for generating relevant distractors for multiple-choice and fill-in-the-blank questions. We explore InfoNCE and Triplet losses, as contrastive objectives, with the generation task to align semantically similar question-answer and distractor pairs closer in feature space while distancing negative pairs. This joint training improves the models to capture semantic features to generate in-context distractors. We validate our work through automatic and manual evaluations across two datasets and recent state-of-the-art approaches. The contrastive-based approach represents a strong baseline model and an insightful contribution to the DG field. It shows the success in improving the automatic results and the quality of generated distractors.

Limitations

We identify the following limitations of our research work. While contrastive learning has enhanced the semantic alignment between generated distractors and human-created ones, Text2Text models are still vulnerable to generating distractors that are either similar to the answer, repetitive, or semantically valid as potential answers. Furthermore, automatic evaluation metrics primarily rely on token-level matching to ground-truth distractors, failing to fully capture the quality of the generated distractors. Although contrastive learning showed a significant improvement over the candidate generation and selection framework, the latter often generate a more diverse set of distractors. We hope our work will encourage the community to explore integrating contrastive learning for a novel selection method within these frameworks. Remarkably, our work still provides a strong baseline approach for the distractor generation field.

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 the University of Adelaide in Australia, along with Umm Al-Qura University in the Kingdom of Saudi Arabia. We are thankful for the support received from the members of the Intelligent Computing Laboratory in the School of Computing at Macquarie University.

References

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. Association for Computational Linguistics.

Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Distractor generation for multiple-choice questions with predictive prompting and large language models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pages 48–63. Springer.

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.

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.

Chia-Yin Chen, Hsien-Chin Liou, and Jason S. Chang. 2006. FAST – an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 1–4.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2022. Contrastnet: A contrastive learning framework for few-shot text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10492–10500.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607. PmLR.

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. Association for Computational Linguistics.

Woon Sang Cho, Yizhe Zhang, Sudha Rao, Asli Celikyilmaz, Chenyan Xiong, Jianfeng Gao, Mengdi Wang, and Bill Dolan. 2021. Contrastive multi-document question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 12–30. Association for Computational Linguistics.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–546. IEEE.

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):5.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. Association for Computational Linguistics.

Wenjian Ding, Yao Zhang, Jun Wang, Adam Jatowt, and Zhenglu Yang. 2024. Can we learn question, answer, and distractors all from an image? a new task for multiple-choice visual question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pages 2852–2863. ELRA and ICCL.

Xiangjue Dong, Jiaying Lu, Jianling Wang, and James Caverlee. 2023. Closed-book question generation via contrastive learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 3150–3162. Association for Computational Linguistics.

Xiaohui Dong, Zhengluo Li, Haoming Su, Jixiang Xue, and Xiaochao Dang. 2025. Transformer-enhanced hierarchical encoding with multi-decoder for diversified mcq distractor generation. *Artificial Intelligence Review*, 58(8):229.

Jacob Doughty, Zipiao Wan, Anishka Bompelli, Jubahed Qayum, Taozhi Wang, Juran Zhang, Yujia Zheng, Aidan Doyle, Pragnya Sridhar, Arav Agarwal, et al. 2024. A comparative study of ai-generated (gpt-4) and human-crafted mcqs in programming education. In *Proceedings of the 26th Australasian Computing Education Conference (ACE)*, pages 114–123.

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. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910. Association for Computational Linguistics.

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.

Qi Guo, Chinmay Kulkarni, Aniket Kittur, Jeffrey P Bigham, and Emma Brunskill. 2016. Questimator: generating knowledge assessments for arbitrary topics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 30.

Yingshuang Guo, Jianfei Zhang, Junjie Dong, Chen Li, Yuanxin Ouyang, and Wenge Rong. 2024. Optimization strategies for knowledge graph based distractor generation. In *International Conference on Knowledge Science, Engineering and Management*, pages 189–200. Springer.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. Association for Computational Linguistics.

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*, 30(1):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. Association for Computational Linguistics.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proceedings 16th European Conference Computer Vision (ECCV)*, pages 121–137. Springer.

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. Association for Computational Linguistics.

Chen Liang, Xiao Yang, Drew Wham, Bart Pursel, Rebecca Passonneaur, and C Lee Giles. 2017. Distractor generation with generative adversarial nets for automatically creating fill-in-the-blank questions. In *Proceedings of the 9th Knowledge Capture Conference (K-CAP)*, pages 1–4.

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 (ACL)*, pages 7978–7993. Association for Computational Linguistics.

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.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Ruslan Mitkov, Le An Ha, Andrea Varga, and Luz Rello. 2009. Semantic similarity of distractors in multiple-choice tests: Extrinsic evaluation. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics (GEMS)*, pages 49–56. Association for Computational Linguistics.

Ruslan Mitkov et al. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17–22.

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

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 244–258. Association for Computational Linguistics.

Juan Pino and Maxine Eskenazi. 2009. Semi-automatic generation of cloze question distractors effect of students' l1. In *International Workshop on Speech and Language Technology in Education*.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 3350–3363. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763. PmLR.

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

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.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29. Curran Associates, Inc.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 21548–21561. Curran Associates, Inc.

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. ELRA and ICCL.

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)*, pages 12477–12491. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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)*, pages 38–45. Association for Computational Linguistics.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022. PCL: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12052–12066. Association for Computational Linguistics.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492.

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

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. Association for Computational Linguistics.

Jiahao Xu, Wei Shao, Lihui Chen, and Lemao Liu. 2023. SimCSE++: Improving contrastive learning for sentence embeddings from two perspectives. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12028–12040. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Haoran Yang, Wai Lam, and Piji Li. 2021. Contrastive representation learning for exemplar-guided paraphrase generation. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 4754–4761. Association for Computational Linguistics.

Nana Yoshimi, Tomoyuki Kajiwara, Satoru Uchida, Yuki Arase, and Takashi Ninomiya. 2023. Distractor generation for fill-in-the-blank exercises by question

type. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 276–281. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Munazza Zaib, Quan Z Sheng, Wei Emma Zhang, Elaf Alhazmi, and Adnan Mahmood. 2024. Learning contrastive representations for dense passage retrieval in open-domain conversational question answering. In *International Conference on Web Information Systems Engineering (WISE)*, pages 3–13. Springer.

Hengyuan Zhang, Dawei Li, Shiping Yang, and Yanran Li. 2022a. Fine-grained contrastive learning for definition generation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (AACL-IJCNLP)*, pages 1001–1012. Association for Computational Linguistics.

Yanzhao Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. 2022b. Unsupervised sentence representation via contrastive learning with mixing negatives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11730–11738.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4995–5004.

Haojie Zhuang, Wei Emma Zhang, Chang Dong, Jian Yang, and Quan Sheng. 2024. Trainable hard negative examples in contrastive learning for unsupervised abstractive summarization. In *Findings of the Association for Computational Linguistics (EACL)*, pages 1589–1600. Association for Computational Linguistics.

Haojie Zhuang, Wei Emma Zhang, Jian Yang, Congbo Ma, Yutong Qu, and Quan Z. Sheng. 2022. Learning from the source document: Unsupervised abstractive summarization. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 4194–4205. Association for Computational Linguistics.

A Text2Text Architecture Methods

We utilize recent Text2Text distractor generation approaches. First, we integrate multi-task learning and candidate augmentation methods, proposed by Wang et al. (2023), and retrieval augmented pre-training, suggested by Yu et al. (2024). The following sections describe our implementation of the strategies for fair comparison settings.

A.1 Multi-Task Training

We utilize two tasks, including distractor generation and answer selection, to train a generative model $\mathbb{M}$, as illustrated in Figure 4. For a given query **Q**, answer **A** and set of distractors **D** = $\{d_1, d_2, \ldots, d_N\}$, we construct two forms of inputs. The first is formed as Text2Text generation objective that starts with label "*generate distractors*" and defined as $\mathbb{M}\big(Q\ \texttt{[SEP]}\ A\big) \rightarrow D$. The second input is formed as finding an answer that starts with label "*choose answer*" and defined as $\mathbb{M}\big(Q\ \texttt{[SEP]}\ \{A, D\}\big) \rightarrow A$.

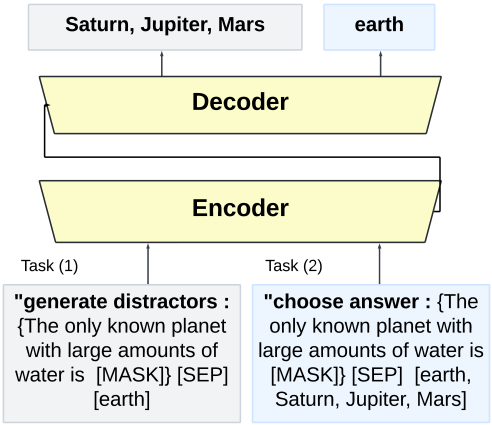

Figure 3: Multi-Task Learning.

A.2 Candidate Augmentation

We observe the effectiveness of few-shot learning in generating in-context distractors, as demonstrated in Table 5; therefore, we utilize this approach for candidate augmentation. This approach is also practical for both fill-in-the-blank and multiple-choice questions than masked language models (MLM) (Devlin et al., 2019) that mainly covers generating distractors for mainly fill-in-the-blank (Chiang et al., 2022) task. Initially, we generate three candidates for augmentation and further examine the impact of expanding to K candidates, as shown in Table 6. Candidate augmentation often introduces noise into the training process (Wang et al., 2023; Yu et al., 2024), leading to a decline in distractor quality instead of improvement.

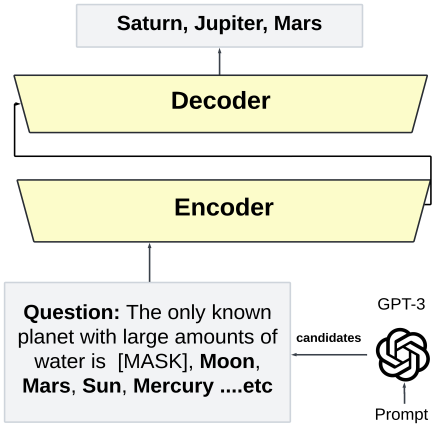

Figure 4: Candidate Augmentation.

K	P@1	R@1	F1@3	MRR	NDCG@3
3	3.47	1.16	4.56	7.34	10.93
5	3.09	1.03	3.47	6.24	9.25
10	1.93	0.64	0.64	2.25	2.56

Table 6: K candidates augmentation in T5 MCQ

A.3 Retrieval Augmented Pre-training

We utilize Wikipedia to expand the knowledge of the selected datasets as proposed by Yu et al. (2024). Since the retrieved datasets are not public, we used each an answer and a distractor in datasets to retrieve a sentence without repetition, then we masked the answer with a [MASK] token, as illustrated in Figure 5. Table 7 shows the statistics of the retrieved datasets, including MCQ and SciQ.

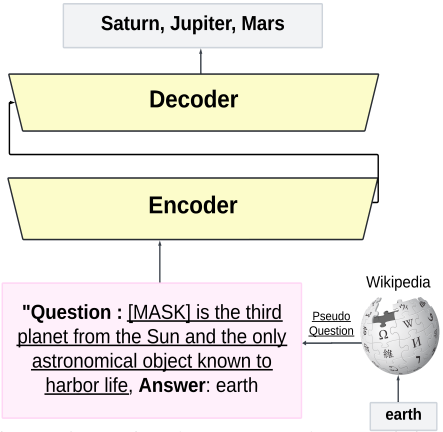

Figure 5: Retrieval Augmented Pre-training.

Datasets	Train	Valid	Test	All
RAP-SciQ	18,889	2,824	1,000	22,713
RAP-MCQ	3,316	830	259	4,405

Table 7: Statistics of the retrieved datasets.

B Analysis on Hyper-Parameters

In the following sections, we study the hyper-parameters used in our approach. Sec. B.1 examines the influence of generation loss (λ_g) and contrastive loss (λ_c) weights on the InfoNCE objective, as defined in Eq. 7, on learning variations within the T5 model. Then, Sec. B.2 details the effects of temperature (τ) in the InfoNCE loss and margin (m) in the Triplet loss as defined in Eq. 3 and Eq. 5, respectively. The hyper-parameters are studied across both the MCQ and SciQ datasets.

B.1 Loss Weights

Adjusting the weights of the generation loss λ_g and contrastive loss λ_c in the T5 model, using the InfoNCE contrastive objective, resulted in varied outcomes across both the MCQ and SciQ datasets.

Firstly, Table 8 presents the results in the MCQ dataset. The optimal performance is achieved when both λ_c and λ_g are set to 0.5. Secondly, Table 9 outlines the results in the SciQ dataset. Unlike the MCQ dataset, SciQ achieves optimal performance metrics - P@1 and R@1 with both λ_c and λ_g set to 0.2, F1@3 at 0.1, and MRR and NDCG@3 at 0.3.

Then, both MCQ and SciQ datasets show a slight decrease when the contrastive loss is omitted in the second stage of training, underscoring the importance of contrastive loss weight in the task of distractor generation at pre-trained encoder-decoder models. Notably, the performance significantly deteriorates when the generation loss λ_g is set to 0.0 in the second stage, highlighting the crucial role of generation loss in aligning T5 objectives with its generative goals.

B.2 Temperature and Margin

Another crucial hyper-parameter impacting model performance is the temperature (τ) in the InfoNCE loss and the margin (m) in the Triplet loss.

Table 10 presents the results of varying the temperature τ for the InfoNCE objective in the T5 model within the MCQ dataset, with optimal performance observed at 0.1 across all automatic metrics. Table 11 outlines the performance of the T5-Triplet/Euclidean model in the MCQ dataset, where the best results are achieved with margins ranging from 0.01 to 0.1. In contrast, Table 12 shows that the optimal performance of the temperature in the SciQ dataset ranges between 0.1 and 0.5, while Table 13 indicates that the best margin performance in SciQ occurs at 0.1.

λ_g	λ_c	P@1	R@1	F1@3	MRR	NDCG@3
0.1	0.1	19.69	6.56	14.03	25.48	29.70
0.2	0.2	20.46	6.82	14.16	26.51	31.19
0.3	0.3	19.31	6.44	13.51	24.07	27.54
0.4	0.4	20.46	6.82	13.64	26.38	31.16
0.5	0.0	20.46	6.82	13.64	24.58	27.37
0.5	0.5	**22.78**	**7.59**	**15.70**	**28.57**	**32.33**
0.0	0.5	1.16	0.39	1.54	2.45	3.61
0.6	0.6	20.46	6.82	14.54	26.58	31.33
0.7	0.7	20.08	6.69	13.90	25.55	29.64
0.8	0.8	20.08	6.69	15.06	26.71	31.66
0.9	0.9	19.69	6.56	12.87	25.10	29.15
1.0	1.0	19.31	6.44	13.26	24.13	27.71

Table 8: λ_g and λ_c settings on T5 InfoNCE at MCQ.

λ_g	λ_c	P@1	R@1	F1@3	MRR	NDCG@3
0.1	0.1	25.00	8.33	**18.13**	31.48	36.64
0.2	0.2	**25.70**	**8.57**	17.60	31.55	36.22
0.3	0.3	25.50	8.50	18.03	**31.95**	**37.08**
0.4	0.4	23.60	7.87	17.60	30.42	35.95
0.5	0.0	24.30	8.10	17.70	30.35	35.16
0.5	0.5	25.00	8.33	17.73	31.42	36.68
0.0	0.5	0.90	0.30	1.07	1.81	2.50
0.6	0.6	24.30	8.10	17.87	30.97	36.43
0.7	0.7	24.00	8.00	17.67	30.63	36.14
0.8	0.8	25.00	8.33	17.50	30.73	35.03
0.9	0.9	23.40	7.80	17.50	30.12	35.78
1.0	1.0	24.10	8.03	16.87	30.12	35.06

Table 9: λ_g and λ_c settings on T5 InfoNCE at SciQ.

τ	P@1	R@1	F1@3	MRR	NDCG@3
0.08	18.92	6.31	13.51	25.16	30.24
0.1	**22.78**	**7.59**	**15.70**	**28.57**	**32.33**
0.5	20.85	6.95	14.80	26.71	31.12
1.0	21.24	7.08	14.93	25.42	27.97

Table 10: τ on T5 InfoNCE loss at MCQ

m	P@1	R@1	F1@3	MRR	NDCG@3
0.04	21.62	7.21	13.77	26.06	29.27
0.01	**22.01**	**7.34**	14.16	**26.96**	30.46
0.1	21.24	7.08	**14.29**	26.64	**30.63**
0.4	20.08	6.69	14.29	25.42	29.33

Table 11: m on T5 Triplet loss (Euclidean) at MCQ.

τ	P@1	R@1	F1@3	MRR	NDCG@3
0.08	24.70	8.23	17.33	31.08	36.26
0.1	25.00	8.33	**17.73**	31.42	**36.68**
0.5	**25.80**	**8.60**	17.47	**31.47**	35.91
1.0	23.90	7.97	17.70	30.52	36.04

Table 12: τ on T5 InfoNCE loss at SciQ.

m	P@1	R@1	F1@3	MRR	NDCG@3
0.04	23.60	7.87	16.90	29.98	35.23
0.01	24.80	8.27	17.50	30.62	35.25
0.1	**25.40**	**8.47**	**17.60**	**31.08**	**35.67**
0.4	24.70	8.23	17.00	30.15	34.47

Table 13: m on T5 Triplet loss (Euclidean) at SciQ

Question [Answer]	A [lead] is typically used to shield things from gamma rays
Models	Generated Distractors
T5-Contrast(InfoNCE)	[aluminum, copper, zinc]
T5-Contrast(Triplet)/e.d	[aluminum, lead, copper]
GPT-3(few-shot)	[aluminum, lead, copper]
T5-CG(beam)	[glass, carbon, mercury]
T5-CG(clustering)	[iron, metals, cellulose]
T5-base	[helium, sulphur, carbon]
T5(Multi-task)	[helium, carbon, aluminum]
T5(CA)	[lead, helium, aluminum]
T5(RAP)	[copper, zinc, copper]
Ground-Truth	[gold, silver, aluminum]
Question [Answer]	[inches] of measurement is typically used for rainfall
Models	Generated Distractors
T5-Contrast(InfoNCE)	[grams, meters, liters]
T5-Contrast(Triplet)/e.d	[grams, minutes, liters]
GPT-3(few-shot)	[unit, method, scale]
T5-CG(beam)	[kilograms, meters, grams]
T5-CG(clustering)	[kilograms, meters, degrees]
T5-base	[meters, meters]
T5(Multi-task)	[meter, kilometer, meters]
T5(CA)	[inches, meters, meters]
T5(RAP)	[meters, meters]
Ground-Truth	[gallons, feet, pounds]
Question [Answer]	Sugars are broken down into in your digestive system [glucose]
Models	Generated Distractors
T5-Contrast(InfoNCE)	[fats, vitamins, protein]
T5-Contrast(Triplet)/e.d	[fat, protein, water]
GPT-3(few-shot)	[proteins, vitamins, minerals]
T5-CG(beam)	[glucose, fat, cellulose]
T5-CG(clustering)	[sugar, fats, cellulose]
T5-base	[lipids, fat, protein]
T5(Multi-task)	[lipids, proteins, carbohydrates]
T5(CA)	[glucose, glucosamine, protein]
T5(RAP)	[cellulose, sulfonate, magnesium]
Ground-Truth	[starch, insulin, nicotine]

Table 14: Examples of distractors generated by ten models from the **MCQ** dataset. The models include contrastive learning (InfoNCE, Triplet), prompting (few-shot), candidate set generation and selection framework (beam search, clustering), Text2Text methods (multi-task, candidate augmentation, retrieved augmented pre-training) and ground-truth distractors.

Question [Answer]	A cochlear implant aims to restore loss of what sense? [hearing]
Models	Generated Distractors
T5-Contrast(InfoNCE)	[visual, smell, touch]
T5-Contrast(Triplet)/e.d	[vision, touch, taste]
GPT-3(few-shot)	[taste, touch, smell]
T5-CG(beam)	[hearing, senses, vision]
T5-CG(clustering)	[hearing, senses, smell]
T5-base	[vision, vision loss, sight]
T5(Multi-task)	[vision, smell, touch]
T5(CA)	[hearing, vision, hearing]
T5(RAP)	[vision, hearing, vision]
Ground-Truth	[vision, touch, taste]
Question [Answer]	Parasites infect the host's what [red blood cells]
Models	Generated Distractors
T5-Contrast(InfoNCE)	[white blood cells, marrow cells, plateletes]
T5-Contrast(Triplet)/e.d	[lungs, white blood cells, marrow cells]
GPT-3(few-shot)	[digestive system, circulatory system, respiratory system]
T5-CG(beam)	[white blood cells, lungs, blood cells]
T5-CG(clustering)	[blood cells, liver]
T5-base	[lungs, intestines, brain]
T5(Multi-task)	[lungs, spleen, liver]
T5(CA)	[lungs, intestines, brain]
T5(RAP)	[white blood cells, white clots, red blood cell]
Ground-Truth	[platelets, plasma, white blood cells]
Question [Answer]	A bee will sometimes do a dance to tell other bees in the hive where to find what? [food]
Models	Generated Distractors
T5-Contrast(InfoNCE)	[honey, eggs, shelter]
T5-Contrast(Triplet)/e.d	[shelter, water, honey]
GPT-3(few-shot)	[water, shelter, predators]
T5-CG(beam)	[water, food, pollen]
T5-CG(clustering)	[water, nectar, insects]
T5-base	[water, food, water]
T5(Multi-task)	[water, shelter, food]
T5(CA)	[food, water, food]
T5(RAP)	[water, food, water]
Ground-Truth	[enemies, water, honey]

Table 15: Examples of distractors generated by ten models from the **SciQ** dataset. The models include contrastive learning (InfoNCE, Triplet), prompting (few-shot), candidate set generation and selection framework (beam search, clustering), Text2Text methods (multi-task, candidate augmentation, retrieved augmented pre-training) and ground-truth distractors.