Enhancing Objective Query Distractor Generation in Pre-trained Encoder-Decoder Models via Contrastive Learning

Anonymous ACL submission

Abstract

Distractor generation is a critical task in objective types of assessments, including fill-in-the-blank and multiple-choice questions. Recent advances in pre-trained transformer-based models have shown success in generating distractors. Prior research efforts focus on fine-tuning pre-trained encoder-decoder models with data augmentation strategies to improve this task, but these models often fail to capture the full semantic representation of a given query-answer and related distractors. Data augmentation methods often rely on expanding the quantity of proposed distractors, which can introduce noise into the models without necessarily enhancing its understanding of the deeper semantic relationships between distractors. This paper introduces a novel distractor generation model based on contrastive learning to capture semantic details from the query-answer and distractor sequence encodings. The contrastive learning method trains the model to recognize essential semantic features, necessary to generate in-context distractors. The extensive experiments on two public datasets indicate that contrastive learning is essential in encoder-decoder models. It significantly outperforms baseline models and advances the NDCG@3 score from 24.68 to 32.33 in the MCQ dataset and 26.66 to 36.68 in the SciQ dataset.

1 Introduction

In assessments, objective questions (Das et al., 2021) such as multiple-choice and fill-in-the-blank questions are widely used in education because they contribute to fair assessment 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. Notably, the quality of these questions relies on the quality of selecting false plausible options, known as *distractors*. Distractor generation (Dong et al., 2022; Alhazmi et al., 2024) refers to the automated process of generating plausible yet incorrect options in objective types of questions.

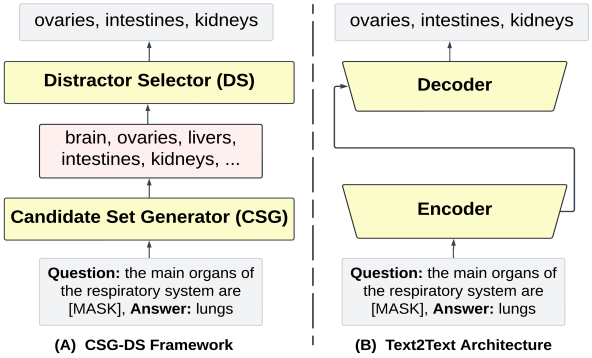

Figure 1: Distractor generation methods via PLMs. **CSG-DS** refers to the candidate generation and selection framework. **Text2Text** represents the sequence-to-sequence generation task by encoder-decoder models.

For decades, research communities have shown interest in generating distractors using several approaches, ranging from feature-based learning (Liang et al., 2018) to deep neural networks (Maurya and Desarkar, 2020). Also, recent advancements in artificial intelligence, particularly in pre-trained language models (PLMs), have significantly enhanced the task of distractor generation through fine-tuning (Bitew et al., 2022; Wang et al., 2023; Yu et al., 2024) and prompting (Feng et al., 2024; Maity et al., 2024; Doughty et al., 2024) methods.

Two primary approaches have been proposed for distractor generation by using PLMs as showed in Figure 1. First, *candidate generation and selection framework* (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. Second, *Text2Text* architecture (Wang et al., 2023) utilizes encoder-decoder models to generate distractors as a sequence-to-sequence (Seq2Seq) task.

While pre-trained encoder-decoder models have shown success in distractor generation, aligning these models specifically with distractor generation

as a Seq2Seq task remains challenging. Recent state-of-the-art approaches incorporate augmentation techniques on distractor candidates (Wang et al., 2023) and adopt retrieval augmented pre-training method (Yu et al., 2024) to enhance the knowledge of pre-trained encoder-decoder models. These models are primarily designed to restore and denoise entire text sequences during pre-training, rather than capturing fine-grained semantic distinctions required for distractor generation models. Therefore, we propose to integrate a contrastive learning 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 pre-trained encoder-decoder models for the distractor generation task.

Initially, the encoding and decoding of the target input and output can be regarded as two representational views with respect to the same semantics. The encoded representation of the question-answer as an input and the sequence of distractors as an output is considered a positive pair. Then, the model utilizes these two representations in contrastive learning with other selected negative pairs in the mini-batch to capture fine-grained semantics.

Since contrastive learning has not yet been conducted in the distractor generation, we explore two contrastive objectives, including InfoNCE and Triplet loss, which both enhanced the performance of distractor generation. InfoNCE utilizes multiple negative examples, while Triplet loss relies on a single negative example. When a contrastive objective integrated with generation loss in encoder-decoder models, a contrastive objective effectively trains the model to bring semantically similar pairs (positives) closer together in the feature space while push dissimilar pairs (negatives) further apart. This training teaches the model to capture semantic features and generate contextually relevant distractors.

Our experimental results, derived from both automatic and human evaluations on two public datasets, demonstrate that this method successfully aligns the distractor generation task with pre-trained encoder-decoder models without relying on augmentation or external data sources. The main contributions of this work can be summarized as follows: (i) introducing a contrastive learning-based approach to enhance distractor generation, marking its first application in pre-trained encoder-decoder models specifically tailored for distractor generation tasks, (ii) validating the effectiveness of our approach by benchmarking it against the state-of-the-art models on two public datasets, using both automatic and manual evaluation metrics, and (iii) conducting extensive analysis to thoroughly examine our approach in encoder-decoder models.

This paper is organized as follows. Sec. 2 reviews the related works on distractor generation and contrastive learning. Sec. 3 presents the details of the proposed methodology. Sec. 4 reports the experimental details along with performance analysis, and Sec. 5 offers some concluding remarks.

2 Related Work

2.1 Distractor Generation

Distractor generation (DG) tasks are typically divided into two primary formats: *multiple-choice questions* (MCQs) and *fill-in-the-blank* (FITB). These formats are applied across various contexts, ranging from textual (Xie et al., 2018) to multi-modal (Yagcioglu et al., 2018) aspects. These tasks are explored across various domains, including question answering (Liang et al., 2017, 2018), reading comprehension (Gao et al., 2019; Xie et al., 2021; Qu et al., 2024), and multi-modal question answering (Zhu et al., 2016; Ding 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 (Kumar et al., 2015; Guo et al., 2016; Yoshimi et al., 2023), e.g., word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2017), to select distractors that are semantically similar to the answer based on cosine similarity of word vectors.

Recently, transformer-based PLMs have revolutionized DG tasks. The two main approaches proposed for generating distractors in text-based contexts include the candidate generation and selection 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 distractor candidates, followed by a feature-rich learning-to-rank model to identify the top distractors. Chiang et al. (2022) utilized PLMs to generate the candidate sets instead of knowledge-

based structure approaches, which showed significant improvement. Additionally, Taslimipoor et al. (2024) proposed using a pre-trained encoder-decoder model for generating both correct and incorrect answer options, and then discriminate between options with a classifier. The generated options are then clustered to remove duplicates.

Wang et al. (2023) treated distractor generation as a Text2Text problem through fine-tuning pre-trained encoder-decoder models. To improve the DG performance, data augmentation was proposed to reduce repeated generation. Yu et al. (2024) applied retrieval-augmented pre-training and used knowledge graph triplet as data augmentation. Although Text2Text DG models have been improved with the previous approaches, the key to improve DG models is to address the lack of fine-grained semantic learning. Thus, we propose to exploit contrastive learning methods to Text2Text DG models.

2.2 Contrastive Learning

Contrastive learning (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). The goal is to bring semantically related instances closer in the feature space, while pushing apart unrelated instances. It has shown success in various domains, starting with applications in computer vision.

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, the CL has been widely used in enhancing semantic information for NLP tasks. Many works applied CL to learn better sentence embeddings (Gao et al., 2021; Giorgi et al., 2021; Kim 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, substantially 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 the few-shot text classification.

In text generation (An et al., 2022), CL is recognized for addressing the degeneration problem, including issues like undesirable generated content and repetitions (Su et al., 2022). Although it has been applied in machine translation (Pan et al., 2021), definition generation (Zhang et al., 2022a), closed-book question generation (Dong et al., 2023), and summarization (Zhuang et al., 2022, 2024), it is not yet applied to DG in pre-trained encoder-decoder models.

3 Methodology

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

3.1 Task Formulation

Given a query $Q = \{q_1, \ldots, q_n\}$ and its corresponding answer $A = \{a_1, \ldots, a_m\}$, the task of DG involves generating a set sequence of distractors $D = \{\{d_{1,1}, \ldots, d_{1,j}\}, \ldots, \{d_{N,1}, \ldots, d_{N,j}\}\}$, 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. $\mathbf{Q}$ and $\mathbf{A}$ denote the query and answer representations, respectively.

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:

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

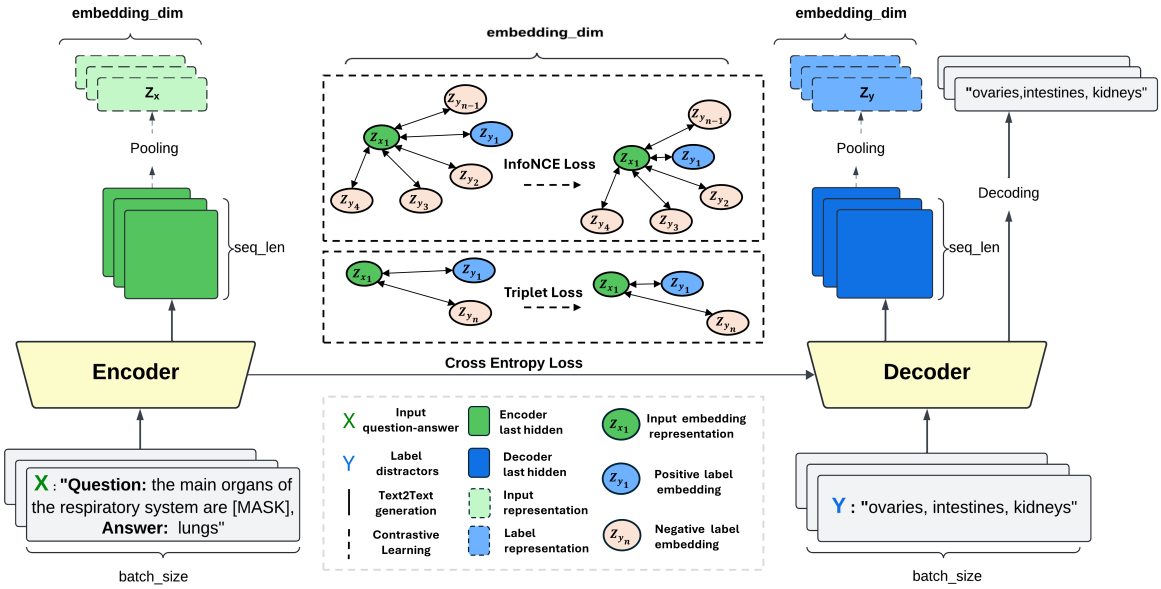

Figure 2: The training pipeline describes the integration of contrastive learning in the Text2Text distractor generation. Solid arrows represent the data flow of generation loss, and dashed arrows indicate the data flow of contrastive learning. Our approach outlines two-stage training. The first stage 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 contrastive learning losses.

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 Learning

Contrastive learning (CL) aims to optimize semantic representations by pulling positive pairs closer in feature space while pushing negative pairs further apart. In DG models, this requires an understanding of the semantics of a question, answer and their relationships with related ground-truth distractors. 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 and y_i 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 representations of inputs x_i and y_i, respectively, $\mathrm{d}(z_i, z_j)$ 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* in Eq 4 or *euclidean distance* 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 learning loss is introduced and trained with the conventional 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 learning 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 serves as a hyper-parameter to balance the generative and contrastive types of losses, respectively. The two-stage training strategy is designed to enable the model to learn semantic information from given question-answer and distractors, thereby improving the model's ability to capture semantic features to align DG with the encoder-decoder models.

4 Experiments

4.1 Datasets

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

SciQ dataset, collected by crowd workers, consists of multiple-choice questions, each with one correct answer and three incorrect distractors. These questions are open-ended and span various domains, including physics, chemistry, biology, and other natural sciences. This dataset contains word-level options, with the average token count for options at 1.6 and 14.5 for the question. We remove unnecessary articles in the answers or distractors.

MCQ or Dgen dataset, collected from several datasets and websites, includes fill-in-the-blank sentences, each with a '**blank**', one correct answer, and three distractors. We replace '**blank**' with $[MASK]$ token. These cloze sentences are also open-ended and span the fields of science, vocabulary, commonsense, and trivia. This dataset contains word-level options, with the average token count for options at 1 and 19.5 for the cloze stem. It is available on GitHub link[1] and comprises training and testing data with 2,321 and 259 instances, respectively. We allocate 80% of the training data for training and the remaining 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 the datasets.

4.2 Baselines Models

We conduct comparative experiments with the following baseline models and recent approaches:

T5-Base (Raffel et al., 2020) and **BART-Base** (Lewis et al., 2020) models. We fine-tune pre-train encoder-decoder models based on the Text2Text architecture using generation loss only.

T5-Base candidate generation, we fine-tune the T5 model using a candidate generation and selection framework. We utilize two approaches for selection: 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.

One-shot and **few-shot** learning (Bitew et al., 2023). We utilize a single random example for one-shot and three random examples for few-shot to generate three distractors for each 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 on the previous 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

Dataset	Method	P@1	R@1	F1@3	MRR	NDCG@3
MCQ	BART-base	8.49	2.83	10.55	14.99	20.66
	BART-Contrast(Triplet)/Cosine	11.58	3.86	11.58	18.53	24.51
	BART-Contrast(Triplet)/Euclidean	15.44	5.15	13.13	22.84	29.20
	BART-Contrast(InfoNCE)	13.90	4.63	12.61	21.24	27.71
	T5-base	14.29	4.76	10.81	20.14	24.68
	T5-Contrast(Triplet)/Cosine	20.46	6.82	13.38	25.61	29.51
	T5-Contrast(Triplet)/Euclidean	22.01	7.34	14.16	26.96	30.46
	T5-Contrast(InfoNCE)	**22.78**	**7.59**	**15.70**	**28.57**	**32.33**
	T5-CG(beam)	17.37	5.79	13.38	24.20	30.11
	T5-CG(clustering)	11.58	3.86	7.72	16.47	20.95
	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	BART-base	10.50	3.50	12.60	17.77	24.02
	BART-Contrast(Triplet)/Cosine	15.50	5.17	15.27	23.67	30.45
	BART-Contrast(Triplet)/Euclidean	16.30	5.43	15.33	24.15	30.63
	BART-Contrast(InfoNCE)	16.00	5.33	15.43	24.73	32.35
	T5-base	18.90	6.30	13.77	23.23	26.66
	T5-Contrast(Triplet)/Cosine	22.20	7.40	16.60	28.58	33.72
	T5-Contrast(Triplet)/Euclidean	24.80	8.27	17.50	30.62	35.25
	T5-Contrast(InfoNCE)	**25.00**	**8.33**	**17.73**	**31.42**	**36.68**
	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
	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: The results of automatic evaluation on the MCQ and SciQ datasets. The best scores are highlighted in bold.

Method	Relevance	Difficulty	Fluency
T5-Contrast(InfoNCE)	**4.16**	**3.56**	**3.98**
T5-Contrast(Triplet)/Euclidean	3.88	3.04	3.76
GPT-3(few-shot)	3.80	2.98	3.74
T5-CG(beam)	3.74	2.70	3.44
T5-CG(clustering)	3.06	2.64	3.16
T5-base	2.82	2.30	2.30
Ground-truth	3.44	3.28	3.52

Table 3: Human evaluations in the MCQ dataset.

We utilize *human* evaluation metrics to assess the model performance. The metrics 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 ten examples, which are then 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-4 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 1.0, 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 employ the gpt-3.5-turbo model for prompting[3].

4.5 Evaluation Results

4.5.1 Automatic Evaluation Results

Table 2 shows a comparison of automatic evaluation results for various models on both datasets. Implementing a two-stage training approach with either InfoNCE or Triplet loss significantly enhanced the performance of Text2Text architecture in both BART and T5 compared to baseline models.

T5-Contrast (InfoNCE) shows superior performance across all metrics. For the MCQ dataset, this model achieves a 8.49% increase in P@1 and a 7.65% rise in NDCG@3 over its baseline T5-base. Similarly, the SciQ dataset records a 6.10% increase in P@1 and a 10.02% improvement in NDCG@3, underscoring the effectiveness of the InfoNCE loss in aligning the T5 model closely with ground-truth distractors. In addition, T5-Contrast

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

Dataset	Method	P@1	R@1	F1@3	MRR	NDCG@3
MCQ	T5-base	14.29	4.76	10.81	20.14	24.68
	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
	T5-Contrast(Triplet)/Cosine	20.46	6.82	13.38	25.61	29.51
	T5-Contrast(Triplet)/Cosine/one-stage	20.08	6.69	15.06	26.00	30.60
	T5-Contrast(Triplet)/Cosine/max	21.62	7.21	15.06	26.83	30.67
	T5-Contrast(Triplet)/Euclidean	22.01	7.34	14.16	26.96	30.46
	T5-Contrast(Triplet)/Euclidean/one-stage	20.46	6.82	12.74	25.42	29.05
	T5-Contrast(Triplet)/Euclidean/max	20.85	6.95	13.26	25.23	28.21
SciQ	T5-base	18.90	6.30	13.77	23.23	26.66
	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
	T5-Contrast(Triplet)/Cosine	22.20	7.40	16.60	28.58	33.72
	T5-Contrast(Triplet)/Cosine/one-stage	23.90	7.97	17.33	30.58	35.91
	T5-Contrast(Triplet)/Cosine/max	22.90	7.63	17.03	29.23	34.27
	T5-Contrast(Triplet)/Euclidean	24.80	8.27	17.50	30.62	35.25
	T5-Contrast(Triplet)/Euclidean/one-stage	24.80	8.27	17.33	30.33	34.62
	T5-Contrast(Triplet)/Euclidean/max	24.50	8.17	17.07	30.38	35.28

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

(Triplet), utilizing Euclidean distance, shows the second-best performance, which may illustrate the benefits of batch-wide optimization in InfoNCE that considers multiple negative examples, compared to one negative example in Triplet.

Conversely, while BART-Contrast (InfoNCE) achieves significant improvements over the BART-base model, BART-Contrast (Triplet), also utilizing Euclidean distance, occasionally surpasses the InfoNCE variant in metrics such as P@1, R@1, and F1@3 in both datasets. The choice of distance metric in Triplet loss plays a critical role in performance, with Euclidean distance demonstrating notably better results compared to the cosine similarity method across both models and datasets. It is worth noting that applying CL within encoder-decoder models for the DG task achieves state-of-the-art results and outperforms other methods (e.g., beam search and clustering-based selection).

4.5.2 Human Evaluation Results

Table 3 presents the results of human evaluations comparing different methods for generating distractors in the MCQ dataset. Remarkably, T5-(InfoNCE) scores the highest across all evaluation metrics: relevance, difficulty, and fluency, demonstrating the substantial benefits of semantic fine-grained training in encoder-decoder models. Additionally, the T5-Triplet/Euclidean distance model surpasses ground-truth distractors in relevance and fluency, underscoring the impact of semantic objective learning in pre-trained models. Comparing T5-base models with contrastive learning models confirm the critical role of semantic training in enhancing DG models. While approaches like few-shot learning and candidate generation and selection methods can generate in-context relevant distractors, they often under-perform CL models due to the lack of semantic fine-grained learning.

(1) Question [Answer]	The common term for the chemical formula H_2O is [water]
Models	Generated Distractors
T5-Contrast(InfoNCE)	[helium, nitrogen, carbon]
T5-Contrast(Triplet)/Euclidean	[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]
Ground-Truth	[hydrogen, air, oxygen]
(2) Question [Answer]	The only known planet with large amounts of water is [earth]
Models	Generated Distractors
T5-Contrast(InfoNCE)	[Mars, Venus, Jupiter]
T5-Contrast(Triplet)/Euclidean	[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]
Ground-Truth	[Saturn, Jupiter, Mars]

Table 5: Examples of distractors generated by seven models from the MCQ dataset.

4.6 Ablation Study

To assess the impact of each component in our methodology, we conduct an ablation study and the results are presented in Table 4. We propose two types of contrastive objectives, as discussed in Sec. 3. Triplet loss incorporates *cosine similarity* or *Euclidean distance*. We also utilize a pooling function and a *two-stage training* strategy.

Test Dataset	Method (Pre-train Dataset)	P@1	R@1	F1@3	MRR	NDCG@3
MCQ	T5-base (SciQ)	25.86	8.62	22.26	33.59	39.03
	T5-Contrast(InfoNCE) (SciQ)	63.70	21.23	55.08	74.38	81.24
	T5-Contrast(Triplet)/Euclidean (SciQ)	66.79	22.26	48.64	74.90	79.72
	BART-base (SciQ)	15.05	5.01	17.76	25.22	33.79
	BART-Contrast(InfoNCE) (SciQ)	**86.48**	**28.82**	**85.84**	**90.54**	**93.37**
	BART-Contrast(Triplet)/Euclidean (SciQ)	82.62	27.54	81.33	86.87	89.85
SciQ	T5-base (MCQ)	15.80	5.26	10.29	19.51	22.38
	T5-Contrast(InfoNCE) (MCQ)	34.10	11.36	27.53	38.34	**41.71**
	T5-Contrast(Triplet)/Euclidean (MCQ)	29.59	9.86	22.03	33.93	37.21
	BART-base (MCQ)	7.50	2.49	10.36	13.71	18.91
	BART-Contrast(InfoNCE) (MCQ)	**36.30**	**12.09**	**35.36**	**38.53**	40.42
	BART-Contrast(Triplet)/Euclidean (MCQ)	34.90	11.63	34.13	37.75	40.19

Table 6: Cross-domain training on the two datasets. The best scores are highlighted in bold.

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 method, 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 (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 method. 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. We also provide analysis on hyper-parameters in App. A. All ablated variants still outperform T5-Base in all metrics, indicating the robustness of CL in DG.

4.7 Case Study

Table 5 presents the distractors generated by seven models. Firstly, it is obvious that the distractors generated by the T5-base model lack semantic relevance to the question. As demonstrated in example (1), the distractors (e.g., *air*, *light*, *air*) might seem plausible in relation to the answer *water*, and in example (2), the distractors (e.g., *ice*, *moon*, *planet*) might also be contextually plausible to *earth*. However, both sets of distractors fail to maintain meaningful semantic connections to the questions, making them unsuitable for real-world applications.

Contrastive learning shows semantic fine-grained in the generated distractors. First, InfoNCE presents remarkable outputs as showed in both example (1) and (2). Secondly, Triplet/Euclidean objective shows varied success in generated outputs. The distractors in example (1) (e.g., *nitrogen*, *carbon*, *oxygen*) are successfully relevant to the question, but the distractors (e.g., *earth*, *moon*, *planet*) in example (2) are semantically less relevant. This outlines the benefit of InfoNCE, including several negative examples compared to Triplet loss, using only one negative example. We provide additional cases in Table 13 in App. A.

We further investigate cross-domain training using CL, leading to notable improvements in the automatic metrics for both datasets, as detailed in Table 6. Contrastive learning has enhanced these metrics even though the training dataset MCQ is smaller than the testing dataset SciQ. This outlines the critical role of contrastive learning in distractor generation models. We present examples of distractors generated through cross-domain training in Table 14 in App. A.

5 Conclusion

In this paper, we integrate contrastive learning (CL) into pre-trained encoder-decoder models for enhancing the objective query distractor generation (DG). We introduce contrastive objectives like InfoNCE and Triplet losses, integrating each one of them with the generation task to align semantically similar question-answer and distractor pairs closer in feature space while distancing negative pairs. This training improves the models to capture semantic features from given pairs to generate in-context relevant distractors. We demonstrate the effectiveness of our approach, which are validated through both automatic and manual evaluations across two datasets. Our work represents a novel contribution to the field of DG. It underscores the significance of CL in improving the automatic results and the quality of generated distractors without using external data augmentation techniques.

Limitations

We identify the following limitations of contrastive learning (CL) in Text2Text-based distractor generation (DG). While contrastive learning has enhanced the semantic alignment between generated distractors and human-created ones, Text2Text models are still vulnerable to producing distractors that are either too similar to the correct answer, repetitive, or semantically valid as potential answers. Furthermore, automatic evaluation metrics still rely on token scores, which only reflect similarity to the ground truth and do not comprehensively represent the quality of the generated output. Although contrastive learning has been effectively applied in Text2Text architectures, candidate generation frameworks can produce a more diverse set of distractors that may are suitable in real-world applications. However, these frameworks require detailed semantic analysis to select high-quality distractors. We hope our work will encourage the community to explore integrating contrastive learning as a novel selection method within candidate generation frameworks. Finally, we would like to declare that our approach and all baseline models are implemented without relying on external data augmentation resources.

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.

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 2197–2210.

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*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146.

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.

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.

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)*, volume 1, pages 539–546.

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.

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 2024)*, pages 2852–2863.

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

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.

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.

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.

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.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. 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 879–895.

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)*, pages 3726–3732.

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.

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.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. 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 2528–2540.

Girish Kumar, Rafael Banchs, and Luis Fernando D'Haro. 2015. RevUP: Automatic gap-fill question generation from educational texts. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 154–161.

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.

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.

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.

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 Knowledge Capture Conference (K-CAP)*, pages 1–4.

Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. 2024. A novel multi-stage prompting approach for language agnostic mcq generation using gpt. In *Proceedings of European Conference on Information Retrieval (ECIR)*, pages 268–277.

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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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.

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.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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.

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.

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)*, volume 139, pages 8748–8763.

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.

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.

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.

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.

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.

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.

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.

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.

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*, pages 12028–12040.

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.

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) (Volume 4: Student Research Workshop)*, pages 276–281.

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.

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 (Volume 1: Long Papers)*, pages 1001–1012.

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 2024*, pages 1589–1600.

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.

A Analysis on Hyper-Parameters

In the following sections, we study the hyper-parameters used in our approach. Sec. A.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. A.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.

A.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 7 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 8 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.

A.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 9 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 10 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 11 shows that the optimal performance of the temperature in the SciQ dataset ranges between 0.1 and 0.5, while Table 12 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 7: λ_g and λ_c settings on T5 InfoNCE loss 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 8: λ_g and λ_c settings on T5 InfoNCE loss 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 9: 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 10: 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 11: 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 12: 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)/Euclidean	[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]
Ground-Truth	[gold, silver, aluminum]
Question [Answer]	Gymnosperms have seeds but do not have [flowers]
Models	Generated Distractors
T5-Contrast(InfoNCE)	[leaves, cells, roots]
T5-Contrast(Triplet)/Euclidean	[leaves, stems, roots]
GPT-3(few-shot)	[leaves, flowers, roots]
T5-CG(beam)	[seeds, leaves, stems]
T5-CG(clustering)	[eggs, flowers, stems]
T5-base	[leaves, leaves]
Ground-Truth	[leaves, stems, roots]
Question [Answer]	[inches] of measurement is typically used for rainfall
Models	Generated Distractors
T5-Contrast(InfoNCE)	[grams, meters, liters]
T5-Contrast(Triplet)/Euclidean	[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]
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)/Euclidean	[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]
Ground-Truth	[starch, insulin, nicotine]

Table 13: Examples of distractors generated by seven models from the MCQ dataset. The models include contrastive learning (InfoNCE, Triplet), prompting (few-shot), candidate set generation and selection framework (beam search, clustering) and ground-truth distractors.

Test Dataset	MCQ
Question [Answer]	While similar to insects, [arachnids] eight-legged invertebrates lack antennae or wings
Models (Pre-train Dataset)	Generated Distractors
T5-base (SciQ)	[reptiles, mammals, amphibians]
T5-Contrast(InfoNCE) (SciQ)	[spiders, reptiles, crustaceans]
T5-Contrast(Triplet)/Euclidean (SciQ)	[spiders, reptiles, insects]
BART-base (SciQ)	[parasites, invertebrates, arthropods]
BART-Contrast(InfoNCE) (SciQ)	[spiders, reptiles, crustaceans]
BART-Contrast(Triplet)/Euclidean (SciQ)	[spiders, reptiles, crustaceans]
Ground-Truth	[spiders, reptiles, crustaceans]
Test Dataset	MCQ
Question [Answer]	The common term for the chemical formula H_2O is [water]
Models (Pre-train Dataset)	Generated Distractors
T5-base (SciQ)	[air, sand, air]
T5-Contrast(InfoNCE) (SciQ)	[acid, air, plasma]
T5-Contrast(Triplet)/Euclidean (SciQ)	[oxygen, helium, nitrogen]
BART-base (SciQ)	[carbon dioxide, oxygen, nitrogen]
BART-Contrast(InfoNCE) (SciQ)	[hydrogen, air, oxygen]
BART-Contrast(Triplet)/Euclidean (SciQ)	[glass, smog, salt]
Ground-Truth	[hydrogen, air, oxygen]
Test Dataset	SciQ
Question [Answer]	What are by far the most common type of invertebrate? [insects]
Models (Pre-train Dataset)	Generated Distractors
T5-base (MCQ)	[worms, fungi, insects]
T5-Contrast(InfoNCE) (MCQ)	[spiders, crustaceans, corals]
T5-Contrast(Triplet)/Euclidean (MCQ)	[crustaceans, crusted, mammals]
BART-base (MCQ)	[mammals, reptiles, crustaceans]
BART-Contrast(InfoNCE) (MCQ)	[spiders, crustaceans, corals]
BART-Contrast(Triplet)/Euclidean (MCQ)	[spiders, crustaceans, corals]
Ground-Truth	[spiders, crustaceans, corals]
Test Dataset	SciQ
Question [Answer]	What kind of organisms helps hold soil in place to slow erosion? [plants]
Models (Pre-train Dataset)	Generated Distractors
T5-base (MCQ)	[insects, insects, animals]
T5-Contrast(InfoNCE) (MCQ)	[animals, insects, bacteria]
T5-Contrast(Triplet)/Euclidean (MCQ)	[roots, bacteria, insects]
BART-base (MCQ)	[mammals, animals, insects]
BART-Contrast(InfoNCE) (MCQ)	[flowers, bacteria, animals]
BART-Contrast(Triplet)/Euclidean (MCQ)	[flowers, bacteria, animals]
Ground-Truth	[flowers, bacteria, animals]

Table 14: Additional distractors generated through cross-domain training using both base fine-tuning and contrastive learning in two PLMs (T5, BART). Each example specified with test dataset and each model indicates the pre-train dataset in parentheses.