Unsupervised Distractor Generation via Large Language Model Distilling and Counterfactual Contrastive Decoding

Fanyi Qu Hao Sun Yunfang Wu*
National Key Laboratory for Multimedia Information Processing, Peking University
MOE Key Laboratory of Computational Linguistics, Peking University
School of Computer Science, Peking University
{fanyiqu,2301213218,wuyf}@pku.edu.cn

Abstract

Within the context of reading comprehension, the task of Distractor Generation (DG) aims to generate several incorrect options to confuse readers. Traditional supervised methods for DG rely heavily on expensive human-annotated distractor labels. In this paper, we propose an unsupervised DG framework, leveraging Large Language Models (LLMs) as cost-effective annotators to enhance the DG capability of smaller student models. Specially, to perform knowledge distilling, we propose a dual task training strategy that integrates pseudo distractors from LLMs and the original answer information as the objective targets with a two-stage training process. Moreover, we devise a counterfactual contrastive decoding mechanism for increasing the distracting capability of the DG model. Experiments show that our unsupervised generation method with Bart-base greatly surpasses GPT-3.5-turbo performance with only 200× fewer model parameters. Our proposed unsupervised DG method offers a cost-effective framework for practical reading comprehension applications, without the need of laborious distractor annotation and costly large-size models.

1 Introduction

Reading comprehension assessment holds significant importance in the educational field. Typically, a reading comprehension sample consists of four components: passage, question, answer and multiple distractors. In recent years, while the cloze-style Distractor Generation (DG) task has received wide interest (Ren and Zhu, 2021; Chiang et al., 2022; Wang et al., 2023), the DG task with complete long sentences is less addressed, primarily due to a scarcity of available supervised data.

Limited by the expensive annotation cost, there are just a few reading comprehension datasets for DG from examination scene (Lai et al., 2017; Sun et al., 2019). Methods on these single-sourcing datasets (Gao et al., 2019; Zhou et al., 2020) all face challenges of insufficient generalization in real-world applications. On the other hand, unsupervised generation methods remain inaccessible considering the difficulty brought by the reading comprehension context.

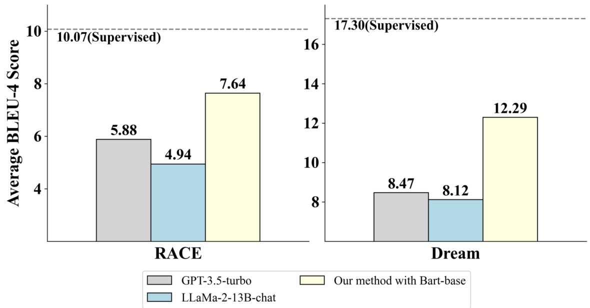


Figure 1: Performance of different unsupervised methods generating 3 distractors on two datasets. We also display results with the supervised Bart-base model for comparison.

Recently, LLMs like GPT (OpenAI, 2023) and LLaMa (Touvron et al., 2023) have demonstrated powerful ability as automatic annotators to label training data (Arora et al., 2022; Gilardi et al., 2023). LLMs have been successfully applied in various fields of NLP, including multi-choice question-answering task (Bitew et al., 2023; Nasution, 2023; Doughty et al., 2024). However, compared to previous fine-tuned methods, mainstream LLMs often fail to achieve a satisfactory performance on DG, as illustrated in Figure 1. Additionally, deploying LLMs in real world applications is challenging due to their substantial computational resource requirements and the closed-source model parameters.

To meet the high need of DG for real applications, where there are no distractor labelling data and only limited computational resource, we propose an unsupervised DG framework with a small model. We adopt the distilling paradigm to en-

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 827–838
August 11-16, 2024

hance the smaller student model's generation capability with pseudo labels from LLMs (Smith et al., 2022; Arora et al., 2022), regarding LLMs as data annotators to assist the training of the smaller model (Wang et al., 2021). Recognizing the suboptimal performance of LLMs on DG, we propose a dual task training strategy by integrating both pseudo distractors and golden answers as training targets. Furthermore, we devise a two-stage training framework to reduce the negative impact caused by the conflict semantics presented in answers and distractors. Note that we do not use the reference distractors for the unsupervised setting.

The utilization of answer information as training target, although improves the model's performance in generation quality, makes harm to the counterfactual capability of the generative model. To address this issue, we introduce contrastive decoding (Li et al., 2023) into the inference process of DG. Specifically, we penalize factual text patterns favored by the answer generation module while encourage counterfactual results generated by the distractor generation agent. Additionally, we apply plausibility constraint to restrict the effect of contrastive decoding for more stable generation results. Note that we do not leverage LLMs during inference to ensure an easy deployment in real applications.

We conduct experiments on RACE (Lai et al., 2017) and Dream (Sun et al., 2019). As illustrated in Figure 1, our unsupervised method with Bart-base significantly outperforms zero-shot LLMs with 200× fewer model parameters. Experimental results also show that our proposed counterfactual contrastive decoding method greatly improves the distracting capability of the generation model. Moreover, we leverage GPT-4 (OpenAI, 2023) to evaluate the generated distractors, demonstrating that our method obtains a better performance than GPT-3.5-turbo both in generation quality and distracting level.

To sum up, our contributions are as follows:

- We propose an unsupervised DG framework with dual task training, integrating the original answer information and pseudo distractors generated by LLMs.

- We devise a new counterfactual contrastive decoding method to improve the distracting level of generated outputs. The proposed optimization can be transferred to other counterfactual generation tasks.

- Our method greatly outperforms teacher LLMs across various evaluation metrics. Our method provides a valuable approach for constructing reading comprehension data in diverse real-world applications, eliminating the need for costly human-annotated data and large-size models.

2 Related Work

2.1 Distractor Generation

Previous researches on distractor generation in reading comprehension mostly concentrate on designing attention framework based on end-to-end models (Gao et al., 2019). Co-attention (Zhou et al., 2020) and reforming modules (Qiu et al., 2020) among passage, question and answer are proposed to extract key information about question-related and counterfactual details. Besides, Mixture of Expert is utilized in some works to ensure the quality and diversity of the generation results (Qiu et al., 2020; Qu et al., 2023).

Recently, DG studies with LLMs mostly focus on knowledge-based distractor generation in education field. Bitew et al. (2023) explores question-similarity based example selection method to enhance LLMs' DG performance in in-context learning. Doughty et al. (2024) designs complex prompt to generate multi-choice question answering data for Python programming learning. Nasution (2023) asks ChatGPT to construct multi-choice data with the input of biology subject for biology learning.

2.2 LLM Knowledge Distillation

LLMs have been widely applied to generate pseudo labels to reduce the labeling cost in unsupervised situation. Recent works have proved the effectiveness of LLMs in annotating accuracy compared to human-annotated results in various NLP tasks (Gilardi et al., 2023; He et al., 2023). Related works have already been applied in question answering (Saad-Falcon et al., 2023), information retrieval (Bonifacio et al., 2022), text summarization (Wang et al., 2021) and common sense reasoning (Whitehouse et al., 2023) , covering both NLU and NLG fields.

Similar unsupervised methods are still under-explored in DG task, though previous works suffer from the expensive labeling cost. Addressing this gap is the main target of our paper.

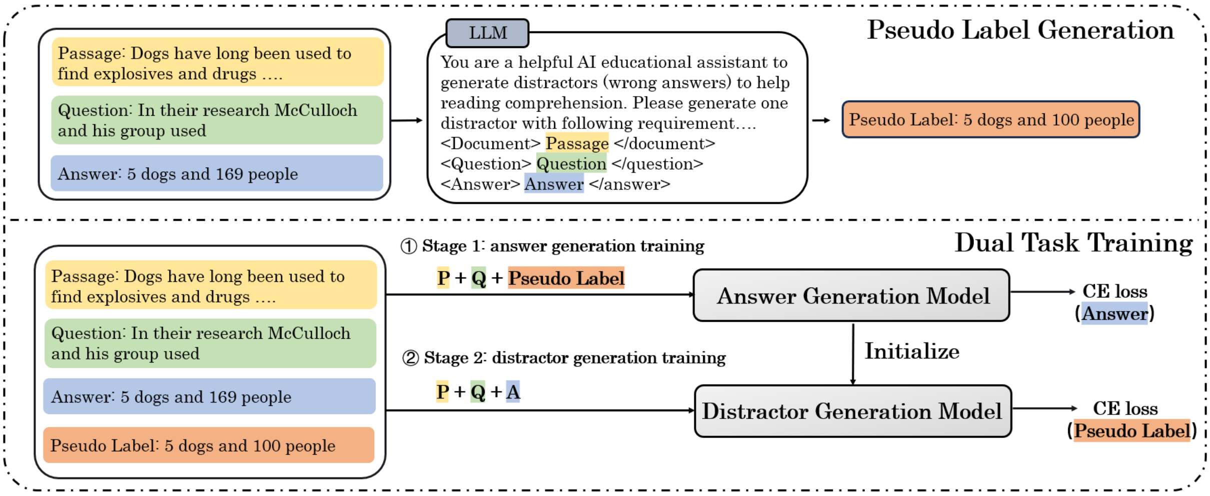


Figure 2: Overview of our proposed unsupervised distractor generation framework, which can be divided into two parts: pseudo distractor generation and dual task training.

3 Method

3.1 Task Definition

The reading comprehension data generally consists of four components: passage, question, answer and multiple distractors. In this context, the DG model regards the triplet of passage(p), question(q) and answer(a) as input and generates results with a probability of p_d:

$$p_d = p(d_i|p, q, a, d_{<i}) \tag{1}$$

In this work, we propose an unsupervised framework for DG task, where the reference distractors in the training data are unseen but only the passage(p), question(q) and answer(a) are provided. In this way, we would liberate experts from the laborious work of distractor annotation.

An overview of our unsupervised framework is displayed in Figure 2. We first apply LLM as a teacher model to generation pseudo distractors. Next, we train a smaller student model with both pseudo labels and answers as generating targets through a two-stage training process.

3.2 Generating Distractors with LLMs

Previous works on DG task (Gao et al., 2019; Zhou et al., 2020) mostly depend on human-annotated data. The common-used dataset like RACE (Lai et al., 2017) is sourced from the educational domain and annotated by professional teachers, exhibiting a high quality but expensive cost. The powerful generation capability of LLMs presents a chance for DG task in reading comprehension to overcome the problem of limited data.

Instead of utilizing human-annotated distractors, we obtain pseudo distractors from a LLM teacher with the input passage, question and answer. To save space, we display the prompts in Appendix A.1.

To guarantee the distracting level of the generated pseudo distractors, we filter out results that exhibit high similarity to the answer by calculating the BLEU-4 score. All pseudo outputs with a BLEU-4 score greater than 30 are dropped, along with their associated question-answer pairs.

3.3 Dual Task Training with Student Models

The pseudo distractors from LLMs (denoted as d') can be directly distilled to a student model as supervised signals during training. However, employing this straightforward augmentation method may not lead to satisfactory results because of the suboptimal performance of LLM in DG task.

As a pair of dual tasks, the answer generation task exhibits similarities with the DG task. Both tasks require a comprehensive understanding on the input passage and deep analysis on question-related contexts. To this end, we introduce answer generation into our unsupervised distractor training process as an auxiliary task.

To take advantage of both pseudo distractor generation and answer generation, we devise a two-stage training procedure for the smaller student model. Firstly, the model treats $[p, q, d']$ as input and generates a, addressing answer generation:

$$p_a = p(a_i|p, q, d', a_{<i}) \tag{2}$$

Secondly, we just interchange answers and pseudo distractors, applying $[p, q, a]$ as input and d' as output, serving as distractor generation:

$$p_d = p(d'_i|p, q, a, d'_{<i}) \quad (3)$$

In the experiment, we prepend task-specific tokens to distinguish these two tasks. For distractor generation, we replace the decoder start token (often `eos` token) with the special token `[DIS]` and `[ANS]` for answer generation.

We denote the model training in answer generation as M_a, and distractor generation as M_d. In the two-stage training procedure, M_d is initialized with the parameters of M_a. Both models apply the cross-entropy (CE) loss for training.

3.4 Contrastive Decoding

The introduction of answer generation in dual task training raises up a challenge on the distracting level of the generated result, as the model may generate correct content. Inspired by the contrastive decoding method (Li et al., 2023), we propose a novel decoding strategy called counterfactual contrastive decoding (CCD) to solve this issue.

3.4.1 Counterfactual Contrastive Decoding

Generally speaking, CCD rewards counterfactual text patterns favoured by the distractor generation model while penalizes factual text patterns generated by the answer generation model. To obtain the output probabilities of both models, we propose a two-stage inference process:

Stage 1 Perform the inference process with p, q, a as input on DG model M_d to generate one distractor result, referring as d_{inter}.

Stage 2 Regard M_d as the expert model and M_a as the amateur model, and apply contrastive decoding as Equation 4:

$$\text{CCD-score}_i = \log \frac{p_d(y_i|p, q, a, y_{<i})}{p_a(y_i|p, q, d_{inter}, y_{<i})} \quad (4)$$

where y is the output sequence of M_d.

Please note that different from the training process, d_{inter} replaces the pseudo distractor d' in the input sequence of M_a, thereby avoiding the dependence on LLM during inference.

Taking into account the high similarity between M_d and M_a models, some common text patterns with high probability in both models will become hard to be generated while the generation probability of some implausible tokens will greatly improve. To address this issue, we optimize the counterfactual contrastive decoding as:

$$\begin{aligned} \text{CCD-score} &= \text{log-softmax}(\text{logit}'_d) \\ \text{logit}'_d &= \text{logit}_d * f(\text{logit}_d, \text{logit}_a) \\ \text{logit}_a &= M_a(p, q, d_{inter}, y_{<i}) \\ \text{logit}_d &= M_d(p, q, a, y_{<i}) \end{aligned} \quad (5)$$

The scaling function f is calculated as:

$$f(x, y) = \exp(\text{sgn}(x) * (\sigma(\frac{x-y}{t}) - 0.5)) \quad (6)$$

where σ is the Sigmoid function, $\text{sgn}(*)$ is the Signum function. t is a hyper-parameter to control the scaling degree and avoid the saturation of the sigmoid function.

As the final value of CCD-score mostly depends on the absolute value of logit_d, the implausible tokens with low generation probability are still hard to be generated after the logit scaling. On the other hand, the scaling factor is determined by the difference between logit_d and logit_a. Notably, it will be close to 1 when the logit_d is approximately equal to logit_a, maintaining the high probability of tokens that both models exhibit high confidence on.

3.4.2 Plausibility Constraint

To improve the stability of the decoding results, we introduce a plausibility constraint method to restrict the effect of counterfactual contrastive decoding. For distractor generation, we propose a rank-based plausibility constraint:

$$\text{CCD-score} = \begin{cases} \text{Equation 5} & \text{if } x_i \in \mathcal{V}_{adj} \\ \text{log-softmax}(\text{logit}_d) & \text{otherwise} \end{cases} \quad (7)$$

$$\mathcal{V}_{adj} = \{y_i \in \mathcal{V} : \text{rank}(p_d(y_i|y_{<i})) > r\} \quad (8)$$

We rank the probability across the vocabulary. For tokens with extremely high probability, we will fix their logits and only adjust tokens ranked after r-th, where r is a hyper-parameter. Compared to the value-based plausibility constraint method proposed in Li et al. (2023), our rank-based method is more stable and controllable.

	Dataset	# q-a pair	# passage	# d(pseudo)
Train	RACE	45120	20028	35452
	Dream	6116	3862	4721
Test	RACE	5787	2519	-
	Dream	2041	1283	-

Table 1: The statistics of RACE and Dream dataset, where q, a refers to the question and answer respectively. d(pseudo) is the pseudo distractor generated by GPT-3.5-turbo.

	Dataset	**Faithful Score**
Answer	RACE	78.34
	Dream	76.23
Distractor	RACE	10.49
	Dream	10.81

Table 2: Faithful Score evaluation on the test set of RACE and Dream.

4 Experimental Setup

4.1 Student Model Training

We choose Bart-base as the student model , which has only 139M parameters and can be easily deployed in real applications. We train the student model with 5-epoch answer generation and 10-epoch distractor generation in the two-stage training process. In both stages we set the maximum learning rate, batch size and warmup ratio to 10^{-5}, 48 and 0.1. During inference, we adopt Jaccard Distance for generating 3 different results with the beam size as 20, and other detail settings can be referred to Zhou et al. (2020). r and t are set to 15 and 2 in experiment.

4.2 Applying LLMs

We conduct experiments on two mainstream series LLMs: LLaMa-2 and GPT-3.5. For LLaMa-2, we apply LLaMa-2-13B-chat implemented by HuggingFace. For GPT-3.5, we apply GPT-3.5-turbo with the official API.

LLM distractor generation for distilling We apply the prompt shown in Appendix A.1 to generate one pseudo distractor for each input p, q and a. For the sake of reproductivity, we turn off the sampling and set the temperature to zero.

LLM zero-shot inference To explore the performance of LLM, we apply LLMs to generate three distractors with the prompt displayed in Appendix A.2. Due to the high cost of applying LLMs with a large beam size, we do not ask LLMs to return 20 results for filtering like student model does.

4.3 Datasets

We conduct extensive experiments on RACE (Lai et al., 2017) and Dream (Sun et al., 2019), which are collected from the English exams of middle and high schools in China. The detailed statistics of the cleaned dataset are shown in Table 1.

We drop the human-annotated distractors in training set and utilize these datasets as unsupervised data. LLMs are applied to generate one pseudo distractor for each question-answer pair in the training set for student model training. The generated results will be filtered with BLEU score as mentioned in Section 3.2.

4.4 Automatic Evaluation Metrics

We apply BLEU[1], Rouge[2] and BertScore[3] to evaluate the generation quality, and Distinct (Li et al., 2016) to evaluate the generation diversity.

As an important evaluation aspect, the distracting level of generated distractors has not received sufficient attention in previous works. Therefore, we propose a new automatic evaluation metric called **Faithful Score** to measure the distracting level of the generated results.

Faithful Score Based on RACE and Dream, we follow Jiang et al. (2020) and train an Alberta model on the machine reading comprehension task. This model aims to judge whether a given candidate is a correct answer to the corresponding passage-question pair and return a classification score ranging [0, 100]. The DG task prefers models that generate distracting results with low Faithful Score.

We conduct evaluation on the test set of RACE and Dream, and results are shown in Table 2. The Faithful Score values on reference answers are between 75 and 80 for both datasets and about 10 on distractors. The experiment results prove the excellent discrimination ability of this metric.

5 Results and Analysis

5.1 Main Results

The experiment results on RACE and Dream are shown in Table 3 and Table 4. For RACE, we display the results on previous works including

[1]https://pypi.org/project/sacrebleu/
[2]https://pypi.org/project/pyrouge/
[3]https://github.com/huggingface/evaluate

Models	1-st B4	2-nd B4	3-rd B4	Avg B4	Avg BS	Avg R-L	Distinct 1	Distinct 2	Avg FS(↓)
Fully Fine-tuned									
HSA	6.43	5.17	4.59	5.40	-	14.67	-	-	-
HCA	7.01	5.51	4.88	5.80	-	15.12	-	-	-
EDGE	7.57	6.27	5.70	6.51	-	18.27	-	-	-
HMD-Net	7.66	6.37	5.33	6.45	-	24.99	-	-	-
MSG-Net	8.87	8.86	8.53	8.75	-	26.39	-	-	-
DG-MoE	9.52	9.12	9.59	9.41	89.78	26.80	69.61	82.40	-
Bart-base	11.22	9.83	9.15	10.07	88.83	26.45	71.39	85.24	24.66
Unsupervised									
GPT-3.5-turbo	6.82	5.75	5.07	5.88	86.73	26.13	75.46	87.68	33.39
LLaMa-2-13B-chat	5.99	4.82	4.00	4.94	70.03	19.48	80.74	91.48	30.30
Our Full Model	8.36	7.43	7.14	7.64	88.38	26.38	70.33	83.86	25.64

Table 3: Experimental results on RACE dataset with 3 distractors comparing with baselines. B4 refers to BLEU-4, BS refers to BertScore, R-L refers to Rouge-L and FS refers to Faithful Score. The pseudo labels for model training are from GPT-3.5-turbo.

Models	1-st B4	2-nd B4	3-rd B4	Avg B4	Avg BS	Avg R-L	Distinct 1	Distinct 2	Avg FS(↓)
Fully Fine-tuned									
Bart-base	22.65	16.37	12.88	17.30	93.30	41.52	76.03	81.25	27.20
Unsupervised									
GPT-3.5-turbo	10.91	8.18	6.33	8.47	91.80	33.00	71.73	77.26	28.67
LLaMa-2-13B-chat	11.81	7.63	5.92	8.12	87.13	31.01	80.61	82.91	24.22
Our Full Model	15.83	11.09	9.93	12.29	92.07	38.44	77.64	81.71	23.83

Table 4: Experimental results on DREAM dataset with 3 distractors. The pseudo labels are from GPT-3.5-turbo.

HSA (Gao et al., 2019), HCA (Zhou et al., 2020), EDGE (Qiu et al., 2020), HMD-Net (Maurya and Desarkar, 2020), MSG-Net (Xie et al., 2022), DG-MoE (Qu et al., 2023). The results of fine-tuned Bart-base and LLMs are also displayed for comparison. For our full model, we apply GPT-3.5-turbo to generate pseudo distractors. We mainly evaluate the results from three aspects: quality, diversity and distracting level.

Performance on the generation quality Both GPT-3.5-turbo and LLaMa-2-13B-chat just manage to obtain half of the SOTA BLEU-4 score on two datasets. Compared to GPT-3.5-turbo, LLaMa-2-13B-chat is more unstable. On the more challenging dataset RACE, LLaMa-2-13B-chat achieves far inferior performance on BertScore (70.03) and Rouge-L (19.48) than other methods.

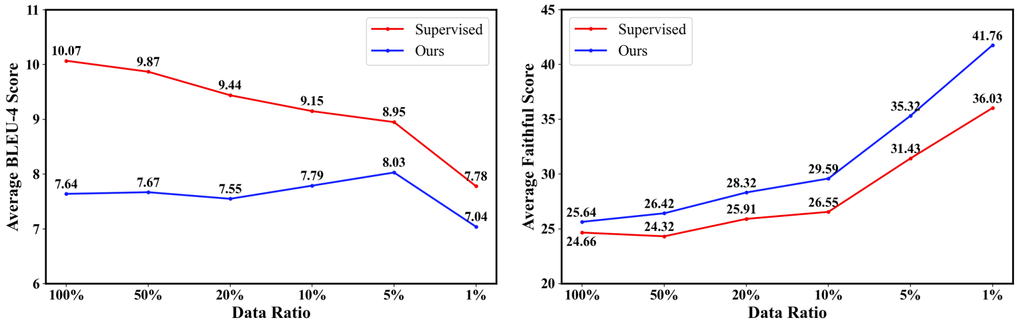
Our proposed unsupervised method greatly outperforms LLMs on the generation quality. In terms of BLEU-4, we achieve 1.76 and 3.82 points improvements on RACE and Dream compared to GPT-3.5-turbo. And our model has achieved an approximate performance on BertScore and Rouge-L compared to the fully fine-tuned SOTA result. As for BLEU-4, there still exists an obvious gap between the unsupervised and fine-tuned results.

Performance on the generation diversity We apply Distinct-1 and Distinct-2 to measure the generation diversity. On both RACE and Dream, our method achieves a close performance to SOTA. Among all these methods, LLaMa-2-13B-chat achieves greatest diversity performance. However, this may be due to the high randomness of its result that improves the generation diversity while makes harm to the generation quality (Fang and Jiang, 2022).

Performance on the distracting level We apply Faithful Score to evaluate the distracting level of the generated results. Our proposed method outperforms GPT-3.5 and LLaMa-2-13B-chat on both datasets and performs closely to the fully fine-tuned method. On RACE, our method is just 0.98 points lower than fine-tuned method, and on Dream our method outperforms 3.37 points.

5.2 Effect of Dual Task Training

We conduct experiments on RACE to investigate the impact of the dual task training based on GPT-3.5-turbo. We train the student model in four different settings: with only pseudo distractors (Pseudo Label); with only answer targets (Answer Label); with the mixed data of pseudo distractors and answers (Mixed Data) and two-stage training process



(a) BLEU-4 Score (b) Faithful Score

Figure 3: Low-resource experimental results on RACE.

Models	Avg B4	Avg BS	Avg R-L	Distinct 1	Distinct 2	Avg FS(↓)
Two-stage training	7.75	88.48	26.77	69.34	83.37	27.59
+ Counterfactual Contrastive Decoding	7.17	88.12	26.22	68.29	82.71	25.85
+ Plausibility Constraint	7.64	88.38	26.38	70.33	83.86	25.64

Table 5: Ablation studies on the contrastive decoding method.

Models	Avg B4	Distinct 1	Avg FS(↓)
Pseudo Label	6.69	72.56	27.31
Answer Label	6.81	70.07	45.31
Mixed Data	8.02	69.16	32.68
Two Stage	7.75	69.34	27.59

Table 6: Experimental results on dual task training.

(Two Stage). In all these settings, we do not apply our proposed counterfactual contrastive decoding during inference.

The results are shown in Table 6. Compared to pseudo-distractor-only method, results of answer-only method have a slight improvement on BLEU-4 and a tiny decline on Distinct-1. However, the Faithful Score of pseudo-distractor-only method outperforms that of answer-only method significantly, with a decline of 18.0 points. As for the dual task training with mixed data, the incorporation of answer information brings significant improvement on the generation quality. Nevertheless, it raises up a negative impact on the result's distracting level, as evidenced by the Faithful Score value.

The two-stage training process proves to be effective to address the above limitation. We observe a decrease of 5.09 points in Faithful Score, accompanied by only a slight 0.27 points reduction in BLEU-4.

5.3 Effect of Counterfactual Contrastive Decoding

Further, we explore the effect of counterfactual contrastive decoding based on GPT-3.5-turbo. We apply the DG model with two-stage training as the base model. The results on RACE dataset are shown in Table 5.

Despite a decline on the generation quality, CCD contributes to an improvement on model's counterfactual generation capability, with a 1.74 points decline on Faithful Score. Besides, plausibility constraint successfully enhances the stability of the generated results, further reducing Faithful Score by 0.21 points and achieving a great trade-off between the generation quality and distracting level.

5.4 Performance with Low-Resource Setting

Our unsupervised method leverages answer information to enhance the generation quality. However, obtaining a sufficient number of annotated answer labels still requires investment, even if cheaper than distractors. In this section, we simulate the low-resource scenario of real-world applications on RACE dataset, and investigate the performance of both supervised and our unsupervised methods. The results are shown in Figure 3. The experiments of our unsupervised method are conducted based on GPT-3.5-turbo.

As the data ratio of the training set decreases, the generation quality of the supervised method declines continuously. In contrast, our unsupervised method maintains a stable BLEU-4 score until the data ratio decreases to 1% (about 400 samples) and demonstrates comparable performance to the supervised method in low-resource situation.

With the decrease of the unannotated data num-

Model	Shots	Avg B4	Avg FS(↓)
Llama-2-13b-chat	0	4.94	30.30
	5	5.24	28.76

Table 7: Experimental results on few-shot setting.

(%)	vs. Supervised			vs. GPT-3.5-turbo		
	Win	Tie	Lose	Win	Tie	Lose
Quality	36.4	23.2	40.4	41.8	18.2	40.0
Distracting	47.1	12.8	40.1	49.1	7.5	43.4

Table 8: Evaluation results from GPT-4 with respect to the quality and distracting level of the generated distractors. We compare our unsupervised model with the supervised Bart-base and GPT-3.5-turbo zero-shot inference results.

ber, the second stage training for distractor generation becomes insufficient, resulting in a heightened impact from the first stage answer generation training on the final results. This leads to a faster increase in the Faithful Score of unsupervised method compared to supervised one.

Besides, we conduct experiments to explore LLMs' performance on few-shot settings. The results in Table 7 indicate that adding 5 randomly selected demonstrations to the prompt only increases the BLEU-4 score by 0.3 points for LLaMa-2-13B-chat.

5.5 Evaluation from GPT-4

N-gram based quality metrics like BLEU are not completely consistent with the actual performance in generation task. Thereby we apply GPT-4 as a professional evaluator to measure the performance of different methods from two aspects: the quality and distracting level of the generated results.

Concretely, we apply GPT-4 to compare two distractors generated by different methods. GPT-4 is asked to return "Win", "Lose" or "Tie" according to the comparison result. The order of two input distractors will be randomly shuffled in the input prompt for a fair comparison. We randomly select 1000 samples from the test set of RACE for evaluation. The prompt for GPT-4 evaluation is proposed in Section A.3. The results are shown in Table 8.

For the generation quality, GPT-4 favors results from the supervised method more, and our method's performance is slightly better than GPT-3.5-turbo. As for the distracting level, our method performs significantly better than the other two methods, demonstrating the effectiveness of our counterfactual contrastive decoding strategy.

Teacher Models	Avg B4	Distinct 1	Avg FS(↓)
GPT-3.5-turbo	7.64	70.33	0.2564
LLaMA-2-13B	5.84	71.27	0.2583

Table 9: Experimental results with different large language models as the teacher models.

5.6 Performance of Different LLMs

Further, to explore the impact of different LLMs on the performance, we adopt GPT-3.5-turbo and LLaMa-2-13B as different teacher models, and the results are shown in Table 9.

Generally, the generation diversity and distracting level of the generated results are not significantly related to the selection of teacher model. However, the unsupervised method based on GPT-3.5-turbo outperforms LLaMa-2-13B by an average of 1.40 points in BLEU-4 score, which is positively correlated with the zero-shot performance of LLMs.

5.7 Case Study

Table 10 illustrates two examples of the generated distractors by three models. All three models produce fluent results without grammar errors. However, in case 1, the results from GPT-3.5-turbo suffers a problem of low diversity. And in case 2, both fine-tuned Bart-base and GPT-3.5-turbo generate **correct answers** like 'how the fathers raise their children', 'what kind of food the fathers and their children eat' and 'how the fathers survive in the desert without cooking skills'. This mistake is not observed in our unsupervised results with counterfactual contrastive decoding.

6 Conclusion

In this paper, we propose an unsupervised distractor generation method. We apply Large Language Models as distractor labelers and construct dual task training process to enhance the student model's generation capability. Moreover, we optimize contrastive generation method in counterfactual generation context. Experiment results indicate that our method generally outperforms LLM-based approaches and is comparable to fully fine-tuned results in low-resource situations. Given the absence of human-annotated distractor dataset, our work can make contribution to building solid reading comprehension data in more future scenarios.

Passage: Until late in the 20th century, most Americans spent time with people of generations. Now mid-aged Americans may not keep in touch with old people until they are old themselves .... The young, in turn, save the old. Once I was in a rest home when a visitor showed up with a baby ....
Question: Now in an American family, people can find that
Answer: not all working people live with their parents

Reference distractors:
(1) children never live with their parents (2) aged people are supported by their grandchildren (3) grandchildren are supported by their grandparents
Results of fully fine-tuned Bart-base:
(1) all working people live with their parents (2) parents don't care about their children (3) all the old people live with their children
Results of GPT-3.5-turbo:
(1) all working people live with their parents (2) working people live with their parents (3) working people do not live with their parents
Results of proposed unsupervised method:
(1) only working people live with parents (2) parents don't care about their children (3) the young save the old in turn

Passage: 'Where Are We Going , Dad ?' presents a new generation of men, in a break from Chinese tradition, now take an active role in their children's lives .... Another one must survive with his son for three days in the desert .... In traditional Chinese culture, the common conception of parenthood is that the father is strict and the mother is kind ....
Question: In Where Are We Going , Dad , we can probably learn about
Answer: how the fathers look after and guide their children

Reference distractors:
(1) how the fathers do housework at home (2) how the children study in their spare time (3) how the children help their father with their travel around China
Results of fully fine-tuned Bart-base:
(1) how the children grow up at the same time (2) how the father is strict and the mother is kind (3) how the fathers raise their children
Results of GPT-3.5-turbo:
(1) what kind of food the fathers and their children eat (2) whether the fathers know how to do their daughters' hair
(3) how the fathers survive in the desert without cooking skills
Results of proposed unsupervised method:
(1) the negative impact of fathers on children's lives (2) the history of Chinese culture (3) why the father is strict

Table 10: Case study for fully fine-tuned model, GPT-3.5-turbo and our proposed unsupervised generation method.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62076008) and the Key Project of Natural Science Foundation of China (61936012).

Limitations

First, there still remains an obvious gap between our proposed unsupervised method and supervised method in DG task, especially for the quality of the generated results. Second, we only experiment on two English dataset from reading comprehension data due to the lack of high-quality testing data. The performance of our method on other application scenarios requires further exploration. Third, we utilize human-annotated answer labels in our unsupervised method, which brings some cost of manual annotation. Fourth, while recent works succeed to enhance student models with rationales generated by LLMs in various NLU tasks, we fail to introduce these methods into DG task. We analyze that for NLG tasks, the rationales generated by LLMs rather complicate the generation process and put negative impact on the model's performance. Related work can be further explored in the future. Last, due to limited time, we do not explore the performance on more LLMs and student models with different parameter scales.

References

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models. *ArXiv preprint*, abs/2210.02441.

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*, abs/2307.16338.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *ArXiv preprint*, abs/2202.05144.

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 2022, Abu Dhabi, United Arab*

Emirates, December 7-11, 2022, pages 5835–5840. Association for Computational Linguistics.

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*, pages 114–123.

Weike Fang and Meng Jiang. 2022. Investigating relationships between accuracy and diversity in multi-reference text generation.

Yifan Gao, Lidong Bing, Piji Li, Irwin King, and Michael R. Lyu. 2019. Generating distractors for reading comprehension questions from real examinations. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6423–6430. AAAI Press.

F Gilardi, M Alizadeh, and M Kubli. 2023. Chatgpt outperforms crowd-workers for textannotation tasks. *ArXiv preprint*, abs/2303.15056.

Xingwei He, Zhenghao Lin, Yeyun Gong, Alex Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, et al. 2023. Annollm: Making large language models to be better crowdsourced annotators. *ArXiv preprint*, abs/2303.16854.

Yufan Jiang, Shuangzhi Wu, Jing Gong, Yahui Cheng, Peng Meng, Weiliang Lin, Zhibo Chen, and Mu Li. 2020. Improving machine reading comprehension with single-choice decision and transfer learning. *ArXiv preprint*, abs/2011.03292.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12286–12312. 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 *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 1115–1124. ACM.

NEA Nasution. 2023. Using artificial intelligence to create biology multiple choice questions for higher education. *Agricultural and Environmental Education*, 2(1).

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Zhaopeng Qiu, Xian Wu, and Wei Fan. 2020. Automatic distractor generation for multiple choice questions in standard tests. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2096–2106, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Fanyi Qu, Che Wang, and Yunfang Wu. 2023. Accurate, diverse and multiple distractor generation with mixture of experts. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 761–773. Springer.

Siyu Ren and Kenny Q. Zhu. 2021. Knowledge-driven distractor generation for cloze-style multiple choice questions. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4339–4347. AAAI Press.

Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Arafat Sultan, and Christopher Potts. 2023. Udapdr: Unsupervised domain adaptation via llm prompting and distillation of rerankers. *ArXiv preprint*, abs/2303.00807.

Ryan Smith, Jason A Fries, Braden Hancock, and Stephen H Bach. 2022. Language models in the loop: Incorporating prompting into weak supervision. *ArXiv preprint*, abs/2205.02318.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

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, Toronto, Canada, July 9-14, 2023*, pages 12477–12491. Association for Computational Linguistics.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. 2023. Llm-powered data augmentation for enhanced crosslingual performance. *ArXiv preprint*, abs/2305.14288.

Jiayuan Xie, Ningxin Peng, Yi Cai, Tao Wang, and Qingbao Huang. 2022. Diverse distractor generation for constructing high-quality multiple choice questions. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:280–291.

Xiaorui Zhou, Senlin Luo, and Yunfang Wu. 2020. Co-attention hierarchical network: Generating coherent long distractors for reading comprehension. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9725–9732. AAAI Press.

A Prompt for LLMs

In this section we display the prompts applied to different LLMs.

A.1 Prompt for generating pseudo distractors

We generate pseudo distractors by LLMs for the training of student model. The instructions are shown in Table 11

A.2 Prompt for zero-shot inference

When applying LLMs for zero-shot inference in distractor generation, we ask LLMs to generate three different results in one generation process. The prompt are shown in Table 12

A.3 Prompt for GPT-4 evaluation

We compare the quality and distracting level of results from two different models with GPT-4, and the prompts are shown in Table 13.

GPT-3.5-turbo	LLaMa-2-13B-chat
"system": You are a helpful AI educational assistant to generate distractors (wrong answers) to help reading comprehension. Please generate one distractor with following requirement: 1. The generated distractor is a wrong answer to the input question according to the given document. 2. Return the generated result directly in one line that begin with ’<result>’ and end with ’</result>’. "user": Now I will provide you with a reading comprehension document, a question and an answer. <document> p </document> <question> q </question> <answer> a </answer>	You are a helpful AI educational assistant to generate distractors (wrong answers) for reading comprehension. You are required to generate one distractor with the given document, question and answer. There are some requirements for you: 1. The generated result should begin with ’<result>’ and end with ’</result>’. 2. If the input question is an incomplete sentence, the generated result should complete the syntax of the question. 3. You should not return any explanations except the distractors. There is the document, question and answer: <document> p </document> <question> q </question> <answer> a </answer> <result>

Table 11: Prompt for GPT-3.5-turbo and LLaMa-2-13B-chat to generate pseudo distractors.

GPT-3.5-turbo	LLaMa-2-13B-chat
"system": You are a helpful AI educational assistant to generate distractors (wrong answers) to help reading comprehension. Please generate three distractors with following requirement: 1. The generated distractors are a wrong answer to the input question according to the given document. 2. The generated results should be returned in three lines and each result should begin with ’<result>’ and end with ’</result>’. "user": Now I will provide you with a reading comprehension document, a question and an answer. <document> p </document> <question> q </question> <answer> a </answer>	You are a helpful AI educational assistant to generate distractors (wrong answers) for reading comprehension. You are required to generate three distractors with the given document, question and answer. Now I will provide you with a document. <document> p </document> There are some requirements for you: 1. The generated result should begin with ’<result>’ and end with ’</result>’. Between <result> and </result>, return three results split by ’;’. 2. If the input question is an incomplete sentence, the generated result should complete the syntax of the question. 3. You should not return any explanations except the distractors. Then I will give you a question-answer pair about the input document. <question> q </question> <answer> a </answer> The three distractors can be: <result>

Table 12: Prompt for GPT-3.5-turbo and LLaMa-2-13B-chat to generate three different distractors.

Prompt for GPT-4	
"system": You are a helpful AI educational assistant that can evaluate distractors (wrong answers) and find the better one from two candidates. "user": Now I will provide you with a reading comprehension document, a question, an answer and a reference distractor. <document> p </document> <question> q </question> <answer> a </answer> <reference> d </reference> Then I will give you 2 distractor candidates and you should judge which one is a better result. The detailed comparison requirements are as follow: ************ $requirement$ ************ I will show you two candidate distractors. If the first candidate is obviously greater than the second candidate, return ’Win’; If the first candidate is obviously worse than the second candidate, return ’Lose’; If you think there are not obvious gap between these two candidates, return ’Tie’. Do not return any explanations about your result. The candidates are: 1. $candidates\_1$; 2. $candidates\_2$.	
Requirements for quality evaluation	**Requirements for distracting level evaluation**
1. You should compare the candidates according to their quality. 2. If the candidate is consist of fluent sentences without any grammar errors, the candidate has high quality. 3. If there are just some small errors like tense error and voice error, the candidate has medium quality. 4. If there are obvious syntactic or grammatical errors, the candidate has low quality.	1. You should compare the candidates according to their distracting level. 2. If the candidate is correct to the input question, it has low distracting level. 3. If the candidate is wrong to the input question, it has high distracting level. 4. The given answer has low distracting level and the given reference has high distracting level. These two sentences can serve as the reference for your comparison.

Table 13: Prompt for GPT-4 to compare distractors generated by two different models.