Distractor Generation based on Text2Text Language Models with Pseudo Kullback-Leibler Divergence Regulation

Hui-Juan Wang, Kai-Yu Hsieh, Han-Cheng Yu, Jui-Ching Tsou,
Yu-An Shih, Chen-Hua Huang, Yao-Chung Fan*
Department of Computer Science and Engineering,
National Chung Hsing University, Taiwan
yfan@nchu.edu.tw

Abstract

In this paper, we address the task of cloze-style multiple choice question (MCQs) distractor generation. Our study is featured by the following designs. First, we propose to formulate the cloze distractor generation as a Text2Text task. Second, we propose *pseudo Kullback-Leibler Divergence* for regulating the generation to consider the *item discrimination index* in education evaluation. Third, we explore the *candidate augmentation* strategy and *multi-tasking training with cloze-related tasks* to further boost the generation performance. Through experiments with benchmarking datasets, our best performing model advances the state-of-the-art result from 10.81 to 22.00 (p@1 score).

1 Introduction

Cloze-style multiple choice question (MCQ) is a common form of exercise used to assess the knowledge of learner. Manual crafting of cloze questions demands significant time and effort for educator, which motivates the need for automatic cloze question generation.

An important challenge in the preparation of cloze questions lies in the selection of appropriate wrong options (distractors). Carefully designing distractors is crucial for enhancing the effectiveness of learner ability assessment, but it also requires significant time and effort. As a result, there has been a growing motivation to explore automatic distractor generation (DG) techniques.

The paradigm for cloze DG is the *candidate generating-and-ranking* (CGR) framework. The CGR paradigm consists of two stages/components: (1) *candidate generator* and (2) *candidate selector*. The candidate generator is generally based on knowledge bases (such as Probase (Wu et al., 2012) or pre-trained language model (Devlin et al., 2018)) to have a distractor candidate set, and the candidate selector ranks the candidates by linguistic features (e.g., morphological, POS, word embedding similarity). The SOTA methods (Chiang et al., 2022; Ren and Zhu, 2021) in recent years are all based on the CGR paradigm.

Question Stem	I was in a _ to reach my office
Options	(a) hurry, (b) way, (c) dream, (d) deferral

Table 1: Item discrimination for Distractor Generation: To consider the validity of the test questions, distractors with different levels of difficulty are needed. In this example, *hurry* is the correct answer, *dream* is an obviously wrong option, and the rest are in the middle.

While the CGR framework shows promise, it overlooks the importance of the *item discrimination index* (Hingorjo and Jaleel, 2012) when evaluating the quality of questions. When teachers design multiple-choice questions (MCQs), it is crucial to consider the validity of the test questions by including distractors of varying difficulty levels. For example, in a four-option MCQ, one option may be easily eliminated, while the remaining two options pose a greater challenge in distinguishing the correct answer, as shown in Table 1. This allows for differentiation among students with varying levels of knowledge during the test. Therefore, the objective of this paper is to incorporate this factor into the process of distractor generation.

Our study incorporates the following notable designs. First, we introduce a formulation that treats cloze distractor generation as a Text2Text task. As demonstrated in the experiment section, this approach yields a significant improvement in performance compared to traditional CGR methods. Second, we propose the utilization of the "pseudo Kullback-Leibler Divergence" technique to regulate the inter-correlation between the generated distractors. This ensures the diversity and relevance of the distractors. Third, we investigate two additional strategies: the "candidate augmentation" strategy and the "multi-tasking training with cloze-related tasks" approach, both of which aim to further enhance the generation performance.

The contributions of this paper are

Findings of the Association for Computational Linguistics: ACL 2023, pages 12477–12491

	Distractor Level		Answer Type		Method Type		Model
	Word/phrase	Sentence	Cloze	R.C.	Extractive	Generative	Type
Gao et al. 2019	Y	Y		Y		Y	RNN
Zhou et al. 2019	Y	Y		Y		Y	RNN
Araki et al. 2016	Y		Y		Y		Non-neural model
Welbl et al. 2017	Y			Y	Y		Random forests
Guo et al. 2016	Y		Y		Y		Word2Vec
Kumar et al. 2015	Y	Y	Y		Y		SVM
Liang et al. 2017	Y		Y			Y	GAN
Liang et al. 2018	Y	Y		Y	Y		Non-neural model
Chung et al. 2020		Y		Y		Y	PLM
Ren and Q. Zhu 2021	Y		Y			Y	Knowledge-base
Peng et al. 2022		Y		Y		Y	PLM
Chiang et al., 2022	Y		Y			Y	PLM
this work	Y		Y			Y	Text2Text

Table 2: An Overview of the Existing Distractor Generation Methods

- Our best performing model achieves a significant advancement in state-of-the-art results, increasing the P@1 score from 10.81 to 22.00. This remarkable improvement represents an almost two-fold increase in performance compared to previous approaches.

- Our study demonstrates that the generative Text2Text framework outperforms the traditional candidate generating-and-ranking framework in the context of distractor generation. This finding suggests that the Text2Text approach serves as a superior alternative for generating high-quality distractors.

- We introduce the concept of pseudo Kullback-Leibler divergence as a means of regulating distractor generation. By incorporating this approach, we aim to address the item discrimination factor when designing multiple-choice questions (MCQs).

- Extensive experimental evaluation with the benchmarking datasets are conducted and the insights of incorporating large models, multi-tasking setting, and context-sentence provision are discussed.

The rest of this paper is organized as follows. Section 2 reviews the works of automatic distractor generation in the literatures. In Section 3 we present the proposed methods. Section 4 reports the performance evaluation and Section 5 concludes this work and discuss the future work.

2 Related Work

In this section, we review the literature related to this work.

Datasets The available distractor datasets are CLOTH (Xie et al., 2017), MCQ (Ren and Zhu, 2021), SCDE (Kong et al., 2020), and RACE (Lai et al., 2017). The CLOTH dataset (Xie et al., 2017) collects word-level cloze questions from English exams designed by teachers. MCQ dataset is a cross-domain cloze-style dataset, that includes the domains of science, vocabulary, common sense, and trivia. MCQ consists of various open-source multiple choice question datasets, including SciQ (Welbl et al., 2017), MCQL (Liang et al., 2018), AI2 Science Questions, and vocabulary and trivia MCQ scraped from websites. SCDE (Kong et al., 2020) consists of cloze question but with sentence-level distractors. Specifically, the SCDE question setting is to fill up multiple blanks in a given passage from a *shared* candidate set of sentence level distractors. The RACE datasets also consists of sentence-level distractors. However, the RACE question setting is a reading comprehension form (instead of cloze form). As our goal is to generate word-level distractors for cloze question, we mainly use CLOTH and MCQ datasets for model learning and evaluation.

Distractor Generator The methods on distractor generation (DG) can be sorted into the following two categories: *cloze distractor generation* and *reading comprehension (RC) distractor generation*.

In cloze DG task, it is viewed as a word filling problem. In general, the first step is to extract dis-

tractor candidates from context or some knowledge base, and then the next step is to rank the extracted distractors as a final result. Along this direction, the models are mainly based on similarity heuristic (Sumita et al., 2005; Mitkov et al., 2006; Guo et al., 2016; Ren and Q. Zhu, 2021) or supervised learning (Liang et al., 2018; Yeung et al., 2019; Ren and Zhu, 2021; Chiang et al., 2022).

The SOTA method for cloze distractor generation is the work by Chiang et al. (Chiang et al., 2022). The work is also based on the CGR framework. The major performance gain comes from the employment of pre-trained language models (PLMs) as a candidate generator. The idea is that PLMs are essentially equipped with the ability of fill-in-the-blank rooted from its MLM (masked token prediction) training process. However, as mentioned, CGR-based methods do not take into account the inter-relationship between generated distractors.

On the other hand, the RC-type DG focuses on generating sentence-level distractors for reading comprehension level testing, such as summarizing article or understanding author opinion (Gao et al., 2019; Zhou et al., 2019; Chung et al., 2020; Peng et al., 2022). For sentence-level distractor generation, neural models are commonly employed.

For clarity of comparison, we summarize the existing DG studies in Table 2.

3 Methodology

Our approach employs a two-stage training process. In the first stage (Subsection 3.1), we utilize a Text2Text framework to generate distractors. This involves training the model to generate plausible distractors based on a given cloze question and its corresponding answer.

In the second stage (Subsection 3.2), we introduce pseudo KL-divergence as a means to regulate the generation of distractors. This step is crucial for ensuring the validity of testing when designing multiple-choice questions (MCQs). By incorporating this technique, we aim to control the quality and relevance of the generated distractors.

Furthermore, we delve into the exploration of boosting techniques in Subsections 3.3 and 3.4. These techniques are intended to enhance our overall approach. They may play a role in improving the distractor generation process or optimizing the design of MCQs.

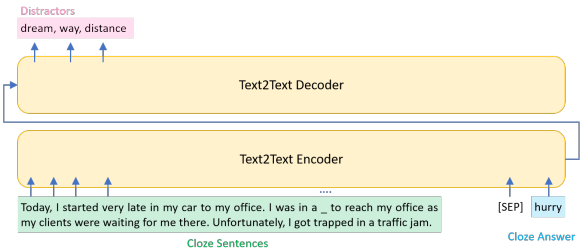

Figure 1: Text2Text Distractor Generation

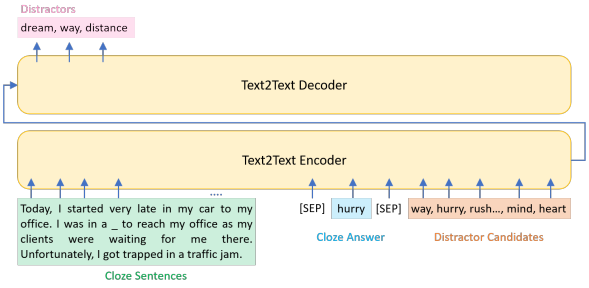

Figure 2: Candidate Augmented Sentence Level Generation

3.1 Text2Text Generation

For a given training instance (C, A, D), the goal is to train a generation model conditioned on C and A by minimizing the negative log-likelihood of the correct token t_i of D given the preceding tokens and the conditions.

$$L_{t2t}(\theta) = -\sum_{i=1}^{|D|} t_i \log p(\hat{t}_i|\hat{t}_1, \hat{t}_2, ..., \hat{t}_{i-1}, C, A; \theta)$$

- C: a cloze question stem (a context passage with a blank gap)
- A: the answer of the blank gap
- D: the set of ground truth distractors d_i.

As illustrated in Figure 1, the input text is a concatenation of a cloze stem C and an answer phrase A (separated by [Sep]). The output target is a distractor sequence $d_1 \oplus d_2 \oplus d_3$.

3.2 Pseudo KL-Divergence Regulation

Let $\mathbb{M}$ be a PLM model and C_{d_i} be the cloze question stem with d_i being placed at the blank gap. Please refer to the table below as an example.

C	I was in a _ to reach my office...
d	dream
C_d	I was in a dream to reach my office...

Furthermore, let the likelihood of d_i conditioned at C and $\mathbb{M}$ be

$$p_{d_i} = p(C_{d_i}|C, \mathbb{M})$$

Let P_D be the probability distribution given by all p_{d_i}s. Given a ground truth distractor set D and the generated distractor set $\hat{D}$, our pseudo KL-divergence regulation is defined as follows.

$$D_{KL}(P_D \| P_{\hat{D}}) = \sum_i P_D(i) \log \frac{P_D(i)}{P_{\hat{D}}(i)}$$

During the second stage training, the training loss is set to the sum of the orginal Text2Text loss and the pseudo KL-divergence loss as follows.

$$L(\theta) = L_{t2t}(\theta) + D_{KL}(P_D \| P_{\hat{D}})$$

3.3 Candidate Augmentation

To further boost the performance, we propose *Candidate Augmentation* strategy. The idea is to generate a set of candidate distractors $\{\hat{d}_1, ..., \hat{d}_k\}$ (top-k results) by a MLM neural candidate generator (we use candidate generator of the state-of-the-art CGP-based method by Chiang et al., 2022) and concatenate the candidates with the original input text as an augmented text input for generation. Specifically, the loss function is

$$L(\theta) = -\sum_{i=1}^{|D|} t_i \log p(\hat{t}_i | \hat{t}_{<i}, C, A, \{\hat{d}_1, ..., \hat{d}_k\}; \theta)$$

The observation behind the candidate augmentation strategy is to inject more information for generation through the MLM candidate generator in hope to boost the performance.

As a concrete example, as illustrated in Figure 2, we align the input text by concatenating the input text with the candidates by MLM neural candidate generator.

3.4 Multi-tasking with Distractor-Related Tasks

To boost the performance, we also explore the employment of multi-task training with the following tasks:

- **Distractor Finding:** The distractor finding task is to detect a distractor span from C. The idea is to place d at the blank gap in question stem C, denoted as $C \otimes d$, and train $\mathbb{M}$ to generate d based on input $C \otimes d$. Specifically, the distractor finding model is with the following generation objective

$$\mathbb{M}(C \otimes d) \rightarrow d$$

- **Cloze Test Answering**: The cloze test answering task is to answer cloze questions. We take C and the option sequence $Opts$ (the option sequence formed by a random permutation of $\{A, D_1, D_2, D_3\}$) as input. The output is the question answer A. Specifically, we have

$$\mathbb{M}(C[\texttt{SEP}]Opts) \rightarrow A$$

4 Experiment

In this section, we introduce the training datasets, the automatic metrics, the implementation details, and the performance results of the compared methods.

4.1 Dataset

We use CLOTH (Xie et al., 2017) and MCQ dataset (the dataset releated by Ren and Zhu, 2021) for performance evaluation.

CLOTH dataset CLOTH is a dataset with a cloze test answer task, it contains an article, options, answers, and source, the source is divided into middle and high, the middle is middle-school English exams and high is high-school English exams. CLOTH contains 7,131 passages with 99,433 questions from China entrance exams. The dataset is divided into train/dev/test with 5,513, 805, and 813.

Note that we find that in the original CLOTH dataset there are two forms of cloze questions: the major form is the one with cloze gaps indicated by _ (a blank) and the other is with cloze gaps indicated by _ and a number (a question number). To avoid the training data insistence, we select to remove the later form (_ with a number). The remaining data for train/dev/test are 5041, 720, and 739. We use the remaining data experiment. The detailed statistics of the dataset are presented in Table 3.

MCQ dataset MCQ dataset is a cross-domain cloze-style dataset, that includes the domains of science, vocabulary, common sense, and trivia. Each data is composed of a sentence containing **blank** of cloze stem, answer, and distractors. According to the setting reported by (Ren and Q. Zhu, 2021), MCQ contains 2880 questions and is randomly divided into train/dev/test with a ratio of 8:1:1. One thing to note for MCQ is sentence-level cloze test while CLOTH is passage-level cloze test.

Dataset	CLOTH				CLOTH-F (Filtered)				MCQ			
	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test	All
# of Passages	5,513	805	813	7,131	5,041	720	739	6500	-	-	-	-
# of Questions	76,850	11,067	11,516	99,433	69,009	9,696	10,233	88,938	2088	233	258	2580

Table 3: The statistics of the training, development and test sets of CLOTH, CLOTH-F (filtered), and MCQ. Note that MCQ consists of sentence-level questions, and therefore # of passages of MCQ is N/A.

We obtain the MCQ dataset from GitHub link shared by (Ren and Q. Zhu, 2021). However, we find there is a slight difference between the numbers in the shared dataset and reported in the paper. In the shared dataset, it only contains train and test data (with 2321/258). Thus, we use this data setting in our experiments. For dev data, we use 9:1 split from train as dev data.

4.2 Evaluation Metrics

Automatic Metric Following the approach by Chiang et al. (Chiang et al., 2022), we evaluate the quality of the generated distractors using several metrics, including F1 score (F1@3), precision (P@1, P@3), and recall (R@1, R@3). P@k represents the ratio of correctly labeled top-k generated distractors, while R@k indicates the ratio of correctly predicted labels among the ground truth. F1@k is the harmonic mean of P@k and R@k. Notably, when the label size is 3, P@3 and R@3 will be the same, resulting in the same F1@3 score. Since both the CLOTH test data and MCQ test data contain 3 distractors, we report the scores of P@1 and F1@3 in the experiments.

Human Evaluation Metric Following (Ren and Zhu, 2021), we asked an English teacher to evaluate the *reliability* and *plausibility* of distractors by showing her the cloze passage and answers. We randomly select 5 passages from the CLOTH-F test set, each passage contains multiple questions, and each question contains multiple distractors, including three generated by each method of the T5 model and three ground truth distractors from the dataset. For each distractor, the judgement based on whether it is correct or incorrect based on the context. For a generated result considered as a feasible distractor, a reliability score of 1 was given and further assessed its plausibility on a 3-point scale: "Obviously Wrong" (0 points), "Somewhat Plausible" (1 point), or "Plausible" (2 points).

4.3 Implementation Details

Our models are implemented based on models from Hugging Face (Wolf et al., 2019). We experiment with BART (Lewis et al., 2019) and T5 (Raffel et al., 2020) as base generation models. For neural candidate generator, we use BERT. For pseudo KL-divergence regulation, we use BART to estimate the likelihood of d_i. During training, we use AdamW as the optimizer and an initial learning rate of 2e-5 for BERT, BART, and 1e-4 for T5 models. All experiments are conducted using two NVIDIA GeForce RTX 3090 GPUs.

BART-based generator With CLOTH data, the maximum number of epochs is set to 20 with a batch size of on two NVIDIA GeForce RTX 3090 GPUs for the Text2Text sentence-level (Len 1) and candidate augmentation (Len 1), the Text2Text passage-level with a batch size of 8, and other methods with a batch size of 32. With MCQ data, the maximum number of epochs is set to 50 with a batch size of 64 on two NVIDIA GeForce RTX 3090 GPUs for the Text2Text sentence-level generation method, and other methods with a batch size of 32. The average running time for BART-based generators is 5 hours (21 minutes) on CLOTH (MCQ).

T5-based generator With CLOTH data, the maximum number of epochs is set to 30 with a batch size of 8 on two NVIDIA GeForce RTX 3090 GPUs for the Text2Text passage-level generation method, and other methods with a batch size of 16. With MCQ data, the maximum number of epochs is set to 50 with a batch size of 64 on two NVIDIA GeForce RTX 3090 GPUs for the Text2Text sentence-level generation method, and other methods with a batch size of 32. The average running time for T5-based generators is 24 hours (39 minutes) on CLOTH (MCQ).

Multi-Tasking and Candidate Augmentation Setting The default top-k for candidate augmentation is set to 20. In the multi-task training, for having a training data balance, When considering a two-tasks setting, we train the sentence-level generation model with full data and sample the same number of data for the distractor finding task (as there are three distractors for each question, the

Dataset	Method	Len	P@1	R@1	F1@3	MRR	NDCG@3
CLOTH-F	Chiang et al., 2022	1	23.17	7.72	18.98	**35.71**	29.13
	BART Text2Text (passage-level generation)	-	22.62	7.54	16.66	28.87	30.86
	BART Text2Text (sentence-level generation)	1	24.84	8.28	18.70	31.53	33.61
	BART Text2Text (sentence-level generation)	3	25.48	8.49	19.34	32.26	34.37
	BART Text2Text with PKL	1	24.05	8.02	18.46	30.74	32.65
	BART candidate augmentation	1	24.25	8.08	19.73	32.17	34.71
	BART candidate augmentation	3	23.69	7.90	19.40	31.49	33.98
	BART multi-task (+ DF)	1	25.16	8.39	19.27	31.97	34.11
	BART multi-task (+ DF)	3	25.74	8.58	19.39	32.33	34.33
	BART multi-task (+ CTA)	3	25.70	8.56	19.62	32.53	34.63
	BART multi-task (+ DF, CTA)	3	25.64	8.54	19.52	32.55	34.66
	T5 Text2Text (passage-level generation)	-	23.03	7.67	14.80	27.42	28.77
	T5 Text2Text (sentence-level generation)	3	28.18	9.39	18.92	33.56	35.15
	T5 Text2Text with PKL	1	25.72	8.57	17.36	30.89	32.28
	T5 candidate augmentation	3	26.07	8.69	18.79	32.45	34.41
	T5 multi-task (+ DF)	3	28.50	9.50	19.10	33.84	35.42
	T5 multi-task (+ CTA)	3	**28.75**	**9.58**	19.20	34.06	35.64
	T5 multi-task (+ DF, CTA)	3	28.47	9.49	**19.82**	**34.46**	**36.26**
MCQ	Ren and Zhu, 2021	1	10.58	-	9.19	17.51	-
	Chiang et al., 2022	1	10.81	3.60	7.72	18.15	15.39
	BART Text2Text (sentence-level generation)	1	14.28	4.76	11.45	21.49	23.70
	BART Text2Text with PKL	1	6.56	2.18	5.92	10.74	12.23
	BART candidate augmentation	1	19.69	6.56	13.12	25.03	26.26
	BART multi-task (+ DF)	1	17.37	5.79	12.61	23.29	25.30
	BART multi-task (+ CTA)	1	16.21	5.40	11.96	22.45	24.33
	BART multi-task (+ DF, CTA)	1	16.60	5.53	12.99	23.61	25.79
	T5 Text2Text (sentence-level generation)	1	18.53	6.17	11.45	23.61	25.08
	T5 Text2Text with PKL	1	9.65	3.21	9.65	16.66	19.07
	T5 candidate augmentation	1	16.60	5.53	13.64	24.90	27.61
	T5 multi-task (+ DF)	1	**22.00**	**7.33**	**13.64**	**27.15**	**28.50**
	T5 multi-task (+ CTA)	1	21.23	7.07	13.51	27.15	28.40
	T5 multi-task (+ DF, CTA)	1	17.76	5.92	12.61	24.00	25.85

Table 4: Distractor Generation Results on the Compared Datasets. In the table, DF denotes the distractor finding task, CTA denotes the cloze test answering task, and PKL denotes the pseudo KL divergence regulation.

amount of data in the distractor finding task will be three times that of the task1. Thus, we randomly select 1/3 of the data for training) to have a 50%:50% data balance. For the three-tasks setting, we randomly select 1/6 data from distractor finding and 1/2 from cloze test answering to have a 50%:25%:25% data balance. The average running time for Multi-Tasking is 28.5 hours (37 minutes) on CLOTH (MCQ).

4.4 Evaluation Results

Table 4 presents the results of the compared methods on the two benchmarking datasets. We have the following notes for the results.

First, Text2Text generation shows best performing results. By comparing MCQ results, we can see that all our Text2Text generation methods surpass the SOTA result reported in (Chiang et al., 2022). Our best performing method (T5 with DF multi-task) advances the SOTA result from 10.81 to 22.00 in terms of P@1.

Second, using large model brings performance improvement. By comparing the result of CLOTH-F and MCQ, T5 (with more parameters) brings near two-points improvements.

Third, the candidate augmentation strategy plays a crucial role in reducing the occurrence of generated distractors that are the same as the answer or previously generated distractors. Initially, it may seem that the candidate augmentation strategy is not effective based on a direct comparison with and without its implementation. However, upon further investigation, we observe that the candidate augmentation strategy leads to significant performance gains by addressing two critical issues: (1) the generation of distractors identical to the answer

Dataset	Method	Len	# of distractors are the same as answer				# of repeatly generated distractor(s)		
			0	**1**	**2**	**3**	**0**	**1**	**2**
CLOTH-F	BART Text2Text (passage-level generation)	-	66.27	18.13	15.54	0.00	35.62	64.37	0.01
	BART Text2Text (sentence-level generation)	1	73.21	17.67	9.06	0.03	47.44	52.51	0.03
	BART Text2Text (sentence-level generation)	3	78.86	15.49	5.61	0.01	60.49	39.48	0.01
	BART Text2Text with PKL	1	68.49	19.91	11.47	0.11	60.28	39.57	0.13
	BART candidate augmentation	1	90.33	6.87	2.78	0.00	**85.09**	14.90	0.00
	BART candidate augmentation	3	89.23	8.59	2.16	0.00	85.06	14.93	0.00
	BART multi-task (+ DF)	1	79.25	14.88	5.85	0.01	64.11	35.87	0.01
	BART multi-task (+ DF)	3	79.68	14.93	5.35	0.02	64.04	35.92	0.02
	BART multi-task (+ CTA)	3	82.85	13.43	3.68	0.01	69.57	30.39	0.02
	BART multi-task (+ DF, CTA)	3	81.08	14.32	4.54	0.04	66.25	33.69	0.04
	T5 Text2Text (passage-level generation)	-	83.11	8.59	2.24	6.03	29.14	37.53	34.26
	T5 Text2Text (sentence-level generation)	3	88.26	6.21	1.91	3.60	45.11	37.45	17.42
	T5 Text2Text with PKL	1	78.81	19.59	1.58	0.00	79.28	20.71	0.00
	T5 candidate augmentation	3	**92.71**	2.62	1.10	3.54	68.91	16.25	14.83
	T5 multi-task (+ DF)	3	88.71	6.03	1.81	3.43	47.38	34.90	17.70
	T5 multi-task (+ CTA)	3	89.60	5.39	1.54	3.45	44.19	37.64	18.16
	T5 multi-task (+ DF, CTA)	3	90.99	6.14	1.12	1.72	60.06	31.35	8.58
MCQ	BART Text2Text (sentence-level generation)	1	67.18	26.64	6.17	0.00	67.56	32.43	0.00
	BART Text2Text with PKL	1	53.28	29.34	16.98	0.38	61.38	38.22	0.38
	BART candidate augmentation	1	77.60	20.46	1.93	0.00	72.20	27.79	0.00
	BART multi-task (+ DF)	1	69.49	27.02	3.47	0.00	82.23	17.76	0.00
	BART multi-task (+ CTA)	1	70.65	23.16	6.17	0.00	65.25	34.74	0.00
	BART multi-task (+ DF, CTA)	1	70.65	24.32	5.01	0.00	78.37	21.62	0.00
	T5 Text2Text (sentence-level generation)	1	85.71	10.03	1.93	2.31	53.66	35.52	10.81
	T5 Text2Text with PKL	1	71.42	26.25	2.31	0.00	88.41	11.58	0.00
	T5 candidate augmentation	1	76.83	22.39	0.77	0.00	**97.68**	2.31	0.00
	T5 multi-task (+ DF)	1	**86.10**	11.96	1.93	0.00	72.97	26.25	0.77
	T5 multi-task (+ CTA)	1	85.32	12.35	1.93	0.38	78.37	20.84	0.77
	T5 multi-task (+ DF, CTA)	1	81.85	15.83	1.93	0.38	72.20	26.64	1.15

Table 5: Statistics on percentage of generating distractor same as answer and generating the same distractors

and (2) the repetition of the same distractors.

To illustrate this, Table 5 presents the percentage of these two cases in the generation results of the compared methods. Notably, in the CLOTH-F comparison, approximately 90.33% of the results obtained from BART candidate augmentation do not contain distractors identical to the answer, and 85.09% of the results do not exhibit repeated distractors generated.

These findings highlight the effectiveness of the candidate augmentation strategy in mitigating the issues related to generating redundant or answer-matching distractors, leading to improved overall performance.

Fourth, from the tables, we observe that PKL does not perform well. In the Cloze dataset, its performance lags behind the best-performing method, T5 multi-task+CTA, by about two to three points. Moreover, in the MCQ comparison, PKL falls far behind other methods. Regarding this issue, we offer the following observations. First, in the Cloze dataset, we find that PKL generates higher-quality outputs to meet the item discrimination index to generate incorrect options (please refer to the case study in the Appendix). Second, in the MCQ task, we noticed that the data in MCQ often consist of more challenging words (this factor causes the language model tokenizer to split complex words into two or more tokens). As a result, our current regulation based on the MLM probability distribution is not effective. Currently, we only calculate PKL distribution for individual words.

Further, the employment of multi-tasking boosts the BART-based and T5-based performance. By comparing the results of CLOTH-F and MCQ, we see that the BART with multi-tasking further advance the performance from 25.48 (14.28) to 25.64 (17.37) (P@1) and the T5 with multi-tasking further advance the performance from 28.18 (19.30) to 28.75 (21.62) (P@1).

Human Evaluation Results Table 6 shows the results of the human evaluation of 5 passages randomly selected on the CLOTH-F test dataset. From the results of human evaluation, we found that the reliability of both ground truth and model-generated distractors are very high. In Plausibility, neither the ground truth nor the generated distractor score is high, because the distractor is too simple and not very suitable for questioning in the English test. Among all methods of T5, the multi-task (+ CTA) method produces distractors the highest in both reliability and plausibility, as well as with the score closest to the ground truth.

4.5 Parameter Study

k value in candidate augmentation We also investigated the effect of distractor candidate top-k in candidate augmentation. We use top-1, 3, 5, 10, and 20 distractor candidates to experiment on CLOTH-F. Table 10 shows that when the candidate distractor is top-5, all metrics are the highest, which means that the generated distractor is closer to the label. Table 7 shows that when the candidate distractor is top-20, the generated distractor has a higher ratio not the same as the answer, and the generated distractor has a higher ratio of not generating repeated distractors.

Impacts on Distractor Order We also investigated whether the order of distractors affects model performance. We conduct experiments on CLOTH-F using the distractors of lexicographical and length-ordered (short-to-long) and compare them with the original dataset ordering. Table 8 shows that the training of the distractor using the dataset order has the highest performance on most metrics, which means that the distractor generated using the original dataset order is closer to the label. The special ordering distractor may make the learning of the distractor more difficult. Table 9 shows that when using the dataset distractor sort, the generated distractors have a higher proportion of different answers; the distractors generated using special order distractors have a higher proportion of not repeating.

5 Conclusion

In this paper, we introduce the utilization of a Text2Text formulation for generating cloze-style multiple-choice questions. Our experimental results highlight a significant performance improvement achieved through the adoption of the Text2Text formulation. Specifically, our approach yields a nearly two-fold increase in performance compared to the current state-of-the-art method. These results strongly suggest that the generative Text2Text framework represents a superior alternative to the traditional candidate generating-and-ranking (CGR) framework.

6 Limitations

We report the following limitations for the Text2Text-based distractor generator (the major

Method	Len	Reliability	Plausibility
T5 Text2Text (passage-level generation)	-	81.51%	0.45±0.59
T5 Text2Text (sentence-level generation)	3	87.22%	0.47±0.56
T5 candidate augmentation	3	84.14%	0.45±0.57
T5 multi-task (+ DF)	3	83.82%	0.51±0.62
T5 multi-task (+ CTA)	3	**87.77%**	**0.56±0.59**
T5 multi-task (+ DF, CTA)	3	86.99%	0.53±0.60
ground truth	-	**88.89%**	**0.63±0.64**

Table 6: Randomly select 5 passages (60 questions in total) from the CLOTH-F test dataset for human evaluation. The value after ± in Plausibility is the standard deviation.

top-k	# of distractors are the same as answer				# of repeatly generated distractor(s)		
	0	**1**	**2**	**3**	**0**	**1**	**2**
1	72.93	18.69	8.37	0.00	57.05	42.94	0.00
3	78.54	17.52	3.93	0.00	76.85	23.14	0.00
5	85.90	11.12	2.97	0.00	82.92	17.07	0.00
10	86.06	9.83	4.10	0.00	79.20	20.79	0.00
20	**90.33**	6.87	2.78	0.00	**85.09**	14.90	0.00

Table 7: Investigation on the ratio of the same and repeated distractors and answers generated by different top-k in candidate augmentation (using the CLOTH-F dataset of length 1)

proposal in this study):

- The Text2Text-based generator still suffers from the concern of generating distractor same as answer or previous generated distractor. In fact, generating repeated incoherent or factual inconsistent results are commonly concerns for neural text generators (Durmus et al., 2020)(Wang et al., 2020). Although the concern is mitigated through the candidate augmentation strategy, there still are certain portions of generating the distractor of those types, as can be seen in Table 5.
- Although the CGR-based methods show their disadvantage in the evaluation, we find that CGR-based method might be a more practical one for facilitating the cloze-style MCQ preparation. The CGR-based method is able to generate ten or more candidates for educators to select, while the Text2Text generators are only capable of generating three or four distractors.

Acknowledgement

This work is supported by National Science and Technology Council, Taiwan, under grant No. NSTC 111-2634-F-005-001 - project Smart Sustainable New Agriculture Research Center (SMARTer), grant No.109-2221-E-005-058-MY3, and Delta Electric Research Center.

References

Jun Araki, Dheeraj Rajagopal, Sreecharan Sankaranarayanan, Susan Holm, Yukari Yamakawa, and Teruko Mitamura. 2016. Generating questions and multiple-choice answers using semantic analysis of texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1125–1136.

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*.

Ho-Lam Chung, Ying-Hong Chan, and Yao-Chung Fan. 2020. A bert-based distractor generation scheme with multi-tasking and negative answer training strategies. *arXiv preprint arXiv:2010.05384*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Esin Durmus, He He, and Mona Diab. 2020. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Distractor Order	P@1	R@1	F1@3	MRR	NDCG@3
dataset order	**28.30**	**9.43**	19.83	**34.64**	**36.51**
dictionary order	23.68	7.89	20.25	32.10	34.87
length order	24.82	8.27	**20.43**	33.06	35.72

Table 8: The performance of the different order of the distractors on the model (using T5 multi-task (+ DF, CTA) on the CLOTH-F dataset)

Distractor Order	# of distractors are the same as answer				# of repeatly generated distractor(s)		
	0	1	2	3	0	1	2
dataset order	**90.53**	6.32	1.52	1.62	61.85	30.22	7.91
dictionary order	88.66	7.99	1.41	1.92	**75.58**	18.09	6.31
length order	89.34	7.74	1.68	1.22	75.57	19.24	5.17

Table 9: Analysis on the ratio of the same and repeated distractors and answers generated by training with different orders of distractors (using T5 multi-task (+ DF, CTA) in the CLOTH-F dataset)

top-k	P@1	R@1	F1@3	MRR	NDCG@3
1	24.22	8.07	18.47	31.08	33.25
3	23.89	7.96	19.74	32.26	34.91
5	**24.80**	**8.27**	**20.02**	**32.79**	**35.30**
10	24.46	8.15	19.70	32.57	35.11
20	24.25	8.08	19.73	32.17	34.71

Table 10: The performance of varying top-k values in candidate augmentation (using the CLOTH-F dataset of length 1)

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

Qi Guo, Chinmay Kulkarni, Aniket Kittur, Jeffrey P Bigham, and Emma Brunskill. 2016. Questimator: Generating knowledge assessments for arbitrary topics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16). AAAI Press*.

Mozaffer Rahim Hingorjo and Farhan Jaleel. 2012. Analysis of one-best mcqs: the difficulty index, discrimination index and distractor efficiency. *JPMA-Journal of the Pakistan Medical Association*, 62(2):142.

Xiang Kong, Varun Gangal, and Eduard Hovy. 2020. SCDE: Sentence cloze dataset with high quality distractors from examinations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5668–5683, Online. Association for Computational Linguistics.

Girish Kumar, Rafael E 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*, pages 154–161.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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*, 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*, pages 1–4.

Ruslan Mitkov, Ha Le An, and Nikiforos Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. *Natural language engineering*, 12(2):177–194.

Hsien-Yung Peng, Ho-Lam Chung, Ying-Hong Chan, and Yao-Chung Fan. 2022. Misleading inference generation via proximal policy optimization. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 497–509. Springer.

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

Siyu Ren and Kenny Q. Zhu. 2021. Knowledge-driven distractor generation for cloze-style multiple choice questions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4339–4347.

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.

Eiichiro Sumita, Fumiaki Sugaya, and Seiichi Yamamoto. 2005. Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 61–68.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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.

Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 2017. Large-scale cloze test dataset created by teachers. *arXiv preprint arXiv:1711.03225*.

Chak Yan Yeung, John SY Lee, and Benjamin K Tsou. 2019. Difficulty-aware distractor generation for gap-fill items. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 159–164.

Xiaorui Zhou, Senlin Luo, and Yunfang Wu. 2019. Co-attention hierarchical network: Generating coherent long distractors for reading comprehension.

Appendix

A Qualitative Study

In Table 11 and Table 12 we present two generation results, selected from CLOTH test set. In each result, we present the cloze passage, cloze answer, and three distractors. We list the distractor results generated by the T5 model using Text2Text (sentence-level), candidate argumentation, the generation with pseudo KL divergence regulation, and multi-task (Distractor Finding Task and Cloze Test Answering Task.)

In Example 1, we observe that using T5 Text2Text produces effective distractors for certain questions, specifically questions 1, 2, 5, 6, 7, and 16. The generated distractors are distinct from the answers and vary among the three options. However, in other questions, we notice instances where Text2Text generates repeated or answer-based distractors. In such cases, the distractors generated by the candidate and multi-task approaches exhibit less repetition, as seen in questions 10, 11, and 13. Notably, the multi-task-generated distractors outperform Text2Text and candidate approaches in questions 3, 8, 9, 12, 14, and 17. These multi-task-generated distractors neither contain duplicates nor share the same part of speech as the answers. Additionally, we find positive outcomes with PKL regulation in questions 2, 7, 8, 9, 11, and 14. For instance, in question 2, "doctors" and "parents" are generated, providing discriminative distractors among the three options, with one being relatively straightforward while the other two pose more difficulty.

Moving to Example 2, we observe that the T5 Text2Text generator generates distinct distractors with the same part of speech as the answers for questions 1, 5, 6, 13, and 14. On the other hand, candidate augmentation generates three distinct distractors for questions 2, 4, 8, 9, 10, 12, and 15, while the Text2Text generator occasionally produces duplicated or answer-based distractors. When both Text2Text and candidate augmentation fail to provide satisfactory distractors in questions 3 and 11, the multi-task generator successfully generates three non-repetitive and non-answer-based distractors. Furthermore, we note favorable outcomes with PKL regulation in questions 1, 7, 3, and 14, showcasing the desired discrimination feature among the options.

Passage	Carly's eyes filled with tears as the dusty bus drove down a dirt road in southern Vietnam.The 14-year-old girl and her _1_ had traveled by plane from Canton, Ohio, to Ho Chi Minh City and then by bus deep into the Mekong Delta. Now, as they reached the village, hundreds of cheering _2_ lined the entrance to the Hoa Lac School, a two-story building that Carly had _3_ money for, When Carly was eight, she started _4_ others by giving Thanksgiving baskets in the church to families in need. It was a snowy day, _5_ she saw that one girl was wearing only a shirt and that others didn't have _6_ coats. The next November, she went door to door asking for uesed coats, hats, gloves, and scarves, and then _7_ them out with the baskets. But Carly wanted to do more —she wanted to"change their lives".She _8_ that her grandmother's Rotary club had, years, earlier, collected money to build a _9_ in Vietnam. That was it, she decided. She'd build a school too. She tried to let people _10_ more about Vietnam and the _11_ there. She gave speeches. She _12_ with enthusiasm." The kids in rural Vietnam don't have beautiful schools, "she told a room of 200 Rotarians. "That's not _13_ .I want to give them a _14_ to make their lives better. "That summer, Carly set off with her family across Ohio, _15_ three or four Rotary clubs a week."We traveled like crazy people to all these _16_ , "recalled her mother, Kris. In two year, Carly had collected $50,000.At the dedication ceremony in Hoa Lac, the school principal was _17_ with the girl. "How wonderful it was that a girl of her age wanted to do something for kids so far away, "he said through a translator.	
Answer / Distractor	1. family / classmates, friends, team 2. schoolchildren / villagers, farmers, workers 3. raised / earned, spent, borrowed 4. helping / encouraging, teaching, engaging 5. and / before, though, because 6. warm / beautiful, big, thin 7. handed / took, left, put 8. remembered / considered, thought, believed 9. school / hospital, factory, hospital	10. know / feel, save, study 11. children / culture, economy, scenery 12. spoke / played, laughed, traveled 13. fair / true, exciting, careful 14. place / room, house, playground 15. visiting / passing, watching, scanning 16. meeting / discussions, topics, suggestions 17. impressed / pleased, satisfied, shocked
T5 Text2Text (sentence-level)	1. mother, father, classmates 2. parents, workers, friends 3. saved, earned, saved 4. helping, helping, helping 5. but, or, so 6. cheap, expensive, dirty 7. took, sent, lent 8. forgot, recalled, recalled 9. hospital, factory, hospital	10. think, think, find 11. children, children, children's 12. sang, cried, cried 13. fair, fair, fair 14. chance, chance, lesson 15. meeting, meeting, meeting 16. clubs, schools, countries 17. satisfied, satisfied, familiar
T5 candidate	1. friend, mother, sister 2. friends, neighbors, students 3. borrowed, spent, spent 4. helping, helping, helping 5. but, so, or 6. dirty, dirty, dirty 7. took, took, took 8. guessed, guessed, guessed 9. church, hospital, hospital	10. think, talk, look 11. teachers, students, parents 12. cried, cried, cried 13. interesting, interesting, interesting 14. gift, prize, prize 15. meeting, meeting, meeting 16. clubs, clubs, schools 17. satisfied, satisfied, satisfied
T5 multi-task (+DF, CTA)	1. mother, father, brother 2. adults, drivers, workers 3. borrowed, earned, saved 4. rescuing, praising, praising 5. but, or, for 6. dirty, dirty, ugly 7. threw, sent, took 8. doubted, guessed, thought 9. church, village, market	10. think, hear, guess 11. women, people, schools 12. sang, told, cried 13. difficult, impossible, impossible 14. room, school, project 15. joining, forming, forming 16. meetings, clubs, trips 17. satisfied, compared, concerned
BART Text2Text with PKL	1. mother, father, sister 2. doctors, workers, parents 3. borrowed, spent, saved 4. helped, helped, helping 5. but, or, so 6. cold, warm, cold 7. took, brought, carried 8. wondered, doubted, imagined 9. hospital, factory, museum	10. say, talk, tell 11. boys, girls, teachers 12. talked, talked, spoke 13. unfair, unfair, unimportant 14. time, place, room 15. visited, visited, visited 16. traveling, traveling, travelling 17. satisfied, satisfied, pleased

Table 11: Generated Distractors Example 1

Passage	Ellen Sims is an 18-year-old college student. She has an important history exam tomorrow morning. Ellen is going to study all night. She is not going to _1_ at all. Many college students, like Ellen, do this often. They think that in the morning, they will _2_ everything that they studied the night before. Ellen thinks that this is a good way to study, but many doctors _3_ . They say that sleep is very important for memory and brain development. Scientists at Harvard Medical School in the USA studied sleep and memory. They studied 24 people. First, they asked the people to look at a picture and _4_ t. At night, they put the people in _5_ groups of 12. Group One went to sleep. Group Two did not. A few days later, scientists showed some _6_ to both groups. They asked the people to find the picture they _7_ before. The people in Group Two did not do so _8_ as those in Group One. It wasn't _9_ for them to remember the picture. What happened? Scientists say that sleep _10_ our memory. After we learn something new, sleep helps us remember it. And when we don't sleep, we can _11_ new things. Scientists say that many teenagers, like Ellen, sleep too _12_ They go to school and work, too. They also _13_ time with their friends. They're always _14_ and they think sleep isn't important. But scientists say the brains of teenagers are still _15_ , and sleeping is a very important part of the development. When teens sleep less than six hours, they can't think clearly. That is not very helpful for a student who is taking an exam.	
Answer / Distractor	1. study, play, eat 2. remember / learn, use, forget 3. disagree / discuss, dislike, discover 4. remember / sell, hold, copy 5. two / three, four, eight 6. pictures / pencils, books, newspapers 7. saw / remembered, threw, drew 8. well / nice, glad, good	9. hard, difficult, difficult 10. helps / steals, takes, worries 11. forget / understand, grasp, lose 12. little / many, much, few 13. spend / cost, take, pay 14. busy / lazy, relaxed, worried 15. developing / getting, cloning, dreaming
T5 Text2Text (sentence-level)	1. mother, father, classmates 2. forget, remember, remember 3. agree, agree, agree 4. see, hear, see 5. one, three, four 6. books, letters, computers 7. looked, saw, look 8. much, much, soon	9. think, think, find 10. helps, helps, helped 11. remember, make, take 12. much, few, few 13. cost, take, pay 14. free, happy, sad 15. developing, developing, developing
T5 candidate	1. eat, work, play 2. read, study, learn 3. agree, approve, agree 4. remind, say, ask 5. one, three, four 6. books, experiments, news 7. made, took, had 8. slowly, quickly, badly	9. hard, difficult, important 10. hurts, destroys, ruins 11. remember, remember, remember 12. many, much, long 13. take, pay, cost 14. happy, tired, lazy 15. growing, recovering, working
T5 multi-task (+DF, CTA)	1. study, eat, speak 2. forget, forget, remember 3. agree, help, study 4. write, read, write 5. three, four, five 6. books, clothes, money 7. looked, found, wrote 8. good, nice, fine	9. hard, difficult, important 10. helps, helps, helped 11. remember, make, take 12. much, many, few 13. take, cost, pay 14. free, happy, sad 15. developing, developing, developing
T5 Text2Text with PKL	1. work, play, eat 2. forget, remember, forget 3. agree, agrees, disagrees 4. forget, forgetting, remembering 5. one, three, four 6. books, papers, books 7. admired, bought, viewed 8. well, wells, good	9. difficult, important, necessary 10. destroys, destroy, ruins 11. remember, remembers, forgetting 12. much, many, much 13. take, cost, pay 14. tired, happy, sad 15. developing, developings, development

Table 12: Generated Distractors Example 2

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Left blank.

☑ A2. Did you discuss any potential risks of your work?
Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☒ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4, Page 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4, Page 5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4, Page 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4, Page 5

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.