CDGP: Automatic Cloze Distractor Generation based on Pre-trained Language Model

Shang-Hsuan Chiang and **Ssu-Cheng Wang** and **Yao-Chung Fan**
Department of Computer Science and Engineering,
National Chung Hsing University, Taichung, Taiwan

Abstract

Manually designing cloze test consumes enormous time and efforts. The major challenge lies in wrong option (distractor) selection. Having carefully-design distractors improves the effectiveness of learner ability assessment. As a result, the idea of automatically generating cloze distractor is motivated. In this paper, we investigate cloze distractor generation by exploring the employment of pre-trained language models (PLMs) as an alternative for candidate distractor generation. Experiments show that the PLM-enhanced model brings a substantial performance improvement. Our best performing model advances the state-of-the-art result from 14.94 to 34.17 (NDCG@10 score). Our code and dataset is available at https://github.com/AndyChiangSH/CDGP.

1 Introduction

A cloze test is an assessment consisting of a portion of language with certain words removed (cloze text), where the participant is asked to select the missing language item from a given set of options. Specifically, a cloze question (as illustrated in Figure 1) is composed by a sentence with a word removed (a blank space) and list of options (one answer and three wrong options).

The cloze test with carefully-design distractors can improve the effectiveness of learner ability assessment. However, manually designing cloze test consumes enormous time and efforts. The major challenge lies in wrong option (distractor) selection. As a result, automatic cloze distractor generation is proposed (Ren and Q. Zhu, 2021; Kumar et al., 2015; Narendra et al., 2013).

In this paper, we extend the candidate-ranking framework reported in (Ren and Q. Zhu, 2021) by exploring the employment of PLMs as an alternative for candidate distractor generation. In this paper, we propose a cloze distractor generation framework called CDGP (Automatic Cloze Distractor Generation based on PLMs) which incorporates a serial of training and ranking strategies to boost the performance of distractor generation based on PLMs.

Stem	If you want recovery soon, start by feeling grateful that you are still ___.
Options	(A) alive — **Answer** (B) lovely (C) lively (D) living — **Distractors**

Figure 1: A Cloze Test Example: the challenge to cloze test preparation lies in wrong option selection. A good wrong option selection improve the effectiveness of learner ability assessment.

The contribution of this work is as follows.

- We show that PLM-based methods brings significant performance improvement over the knowledge-driven methods (Ren and Q. Zhu, 2021) (generating candidates from Probase (Wu et al., 2012) or Wordnet (Miller, 1995))
- We conduct evaluation using two benchmarking datasets. The experiment results indicates that our CDGP significantly outperforms the state-of-the-art result (Ren and Q. Zhu, 2021). We advance NDCG@10 score from 19.31 to 34.17 (improving up to 177%).

2 Related Work

The methods on cloze distractor generation can be sorted into the following two categories. The first category (Correia et al., 2010; Lee and Seneff, 2007) is to prepare cloze distractors based on linguistic heuristic rules. The problem with these methods is that the results are far from practically satisfactory. The second category (Kumar et al., 2015; Narendra et al., 2013) is to construct candidate distractors from domain-specific vocabulary or taxonomies and employ classifiers for selecting final distractors. The results by the methods of this category are still less than satisfactory due to the

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5835–5840
December 7-11, 2022

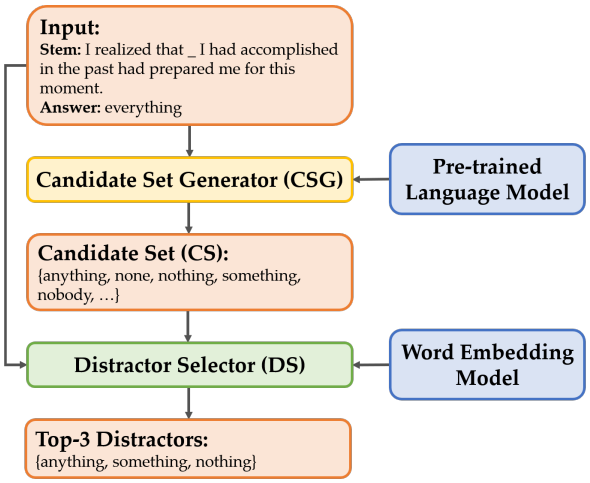

Figure 2: CDGP Framework

domain generalization and the generation quality. To improve the quality, (Ren and Q. Zhu, 2021) proposes to use knowledge bases (Wordnet (Miller, 1995) and Probase (Wu et al., 2012)) to analyze the word semantic and hypernym-hyponym relations for generating candidate distractors. In this paper, we explore the employment of PLMs as a alternative for the knowledge bases in (Ren and Q. Zhu, 2021) and also explore various linguistic features for candidate selection.

3 Methodology

3.1 CDGP Framework

We extend the framework proposed by (Ren and Q. Zhu, 2021) by exploring the employment of pre-trained language models as an alternative for candidate distractor generation. Specifically, as illustrated in Figure 2, the framework consists of two stages: (1) Candidate Set Generator (CSG) and (2) Distractor Selector (DS). In this paper, we revisit the framework by considering (1) PLMs at CSG and (2) various features at DS.

3.2 Candidate Set Generator (CSG)

The input to CSG is a question stem and the corresponding answer. The output is a distractors candidate set of size k.

In this study, we use PLM to generate candidates. Let $\mathbb{M}()$ be PLM model. For a given training instance (S, A, D), where S is a cloze stem, A is the answer, and D is a distractor. We explore the following two training setting for generating distractor candidates.

1. **Naive Fine-Tune**:

$$\mathbb{M}(S_{\otimes[\texttt{Mask}]}) \rightarrow D$$

The input is a given stem S with the cloze blank filled in [Mask] (denoted by $S_{\otimes[\texttt{Mask}]}$). The idea is to fine-tune the PLMs to predict D. The training objective is to find a parameter set θ minimizing the following loss function

$$-log(p(D|S;\theta))$$

2. **Answer-Relating Fine-Tune**: The input is further concatenated with cloze answer A. The idea is to guide the model to refer A to generate D. Specifically,

$$\mathbb{M}(S_{\otimes[\texttt{Mask}]}[\texttt{Sep}]A) \rightarrow D$$

The training objective is to find a parameter set θ minimizing the following loss function

$$-log(p(D|S,A;\theta))$$

3.3 Distractor Selector (DS)

The input to DS is a question stem S, an answer A, and a candidate set $\{D_i\}$ from CSG. We investigate the following features for ranking candidates.

- Confidence Score s_0: the confidence score of D_i given by the PLM at CSG. Specifically,

$$s_0 = p(D_i|S,A;\theta)$$

- Word Embedding Similarity s_1: the word embedding score between A and D given by the cosine similarity between $\vec{A}$ and $\vec{D}$. Specifically,

$$s_1 = 1 - cos(\vec{A}, \vec{D_i})$$

- Contextual-Sentence Embedding Similarity s_2: the sentence-level cosine similarity between the stem with the blank filled in A (denoted by $\vec{S}_{\otimes A}$) and the stem with the blank filled in D (denoted by $\vec{S}_{\otimes D_i}$).

$$s_2 = 1 - cos(\vec{S}_{\otimes A}, \vec{S}_{\otimes D_i})$$

- POS match score s_3: the POS (part-of-speech) matching indicator. $s_3 = 1$, if A and D_i has the same POS tag. Otherwise $s_3 = 0$.

Dataset	CLOTH-M			CLOTH-H			CLOTH (Total)		
	train	dev	test	train	dev	test	train	dev	test
#passages	2341	355	355	3172	450	478	5513	805	813
#questions	22056	3273	3198	54794	7794	8318	76850	11067	11516
Vocab. size	15096			32212			37235		
Avg. #sentence	16.26			18.92			17.79		
Avg. #words	242.88			365.1			313.16		

Table 1: The statistics of the training, developing and testing datasets of CLOTH-M (middle school), CLOTH-H (high school). (Xie et al., 2017)

Dataset	Short-term		Long-term		
	GM	STR	MP	LTR	O
CLOTH	0.265	0.503	0.044	0.180	0.007
CLOTH-M	0.330	0.413	0.068	0.174	0.014
CLOTH-H	0.240	0.539	0.035	0.183	0.004

Table 2: The question type statistics of 3000 sampled questions where GM, STR, MP, LTR and O denotes grammar, short-term-reasoning, matching paraphrasing, long-term-reasoning and others respectively. (Xie et al., 2017)

The final score of a distractor D_i is then computed by a weighted sum over the individual score with MinMax normalization.

$$score(D_i) = \sum_{i=0}^{3} w_i \cdot \texttt{MinMax-Norm}(s_i)$$

Distractors with Top-3 scores are selected as the final resultant distractors.

4 Performance Evaluation

4.1 Dataset

To validate the performance of our methodology, we use the following two datasets.

- **CLOTH datatset** (Xie et al., 2017) The dataset comes from English cloze exercises. The datasets consists of a passage with cloze stems, answers and distractors. The data statistics are summarized in Table 1 and Table 2.
- **DGen dataset** (Ren and Q. Zhu, 2021) The DGen dataset released by (Ren and Q. Zhu, 2021), which is a reorganized dataset from SciQ (Welbl et al., 2017) and MCQL (Liang et al., 2018). We compare our methods with the SOTA method (Ren and Q. Zhu, 2021) based on this dataset. The data statistics are listed in Table 3 and Table 4.

4.2 Implementation Details

We select bert-base-uncased (Devlin et al., 2018) as the default PLM. We use Adam optimizer with an initial learning rate setting to 0.0001. We set the PLM maximal input length to 64. The default batch size is set to 64. All models are trained with NVIDIA® Tesla T4.

For computing word embedding similarity in DS, we use the fasttext model (Bojanowski et al., 2016) as the default embedding model. The fasttext is trained with the cbow setting. The minimal and maximal n-gram parameter are set to 3 and 6. The vector dimension is set to 100. The initial learning rate is 0.05. In addition, the size of distractor candidate set k is set to 10 as a default value.

Data Split	# of questions
total	2880
train	2321
valid	300
test	259

Table 3: DGen statistics. (Ren and Q. Zhu, 2021)

Domain	Total	Science	Vocab.	Commen Sense	Trivia
# of questions	2880	758	956	706	460
#of distractors	3.13	3.00	3.99	3.48	2.99

Table 4: DGen statistics in different domains. (Ren and Q. Zhu, 2021)

4.3 Evaluation Metric

Automatic Score We use the same setting of (Ren and Q. Zhu, 2021); the models are compared by the following automatic scores: Precision (P@1), F1 score (F1@3, F1@10), Mean Reciprocal Rank (MRR@10), and Normalized Discounted Cumulative Gain (NDCG@10).

4.4 Evaluation Results

4.4.1 Results on DGen

In this set of experiment, our goal is to compare our method with the SOTA method (Ren and Q. Zhu, 2021). Table 5 shows the comparison results. In addition to the BERT model, we also report our CDGP variants based on (SciBERT, RoBERTa, and BART). From Table 5, it can be seen that the NDCG@10 of CDGP with SciBERT was improved from 19.31 to 34.17, surpassing the existing SOTA method by 77%.

An interesting finding here is that in this set of experiment, we see CDGP using SciBERT show the best results. We think this confirms the domain matchesness between DGen dataset. Note SciBERT which is pre-trained based on science

Models	P@1	F1@3	MRR@10	NDCG@10
DGen (Wordnet CSG)	9.31	7.71	14.34	14.94
DGen (Probase CSG)	10.85	9.19	17.51	19.31
DGen (w/o CSG)	5.01	5.59	9.28	11.6
CDGP (BERT)	10.81	7.72	18.15	24.47
CDGP (SciBERT)	**13.13**	**12.23**	**25.12**	**34.17**
CDGP (RoBERTa)	13.13	9.65	19.34	24.52
CDGP (BART)	8.49	8.24	16.01	22.66

Table 5: Comparison Results: Comparing CDGP with the DGen (Ren and Q. Zhu, 2021)

Models	P@1	F1@3	F1@10	MRR@10	NDCG @10
Naive	12.60	10.00	12.45	22.70	30.32
Answer Relating	**18.50**	**13.80**	**15.37**	**29.96**	**37.82**

Table 6: The Results of Naive and Answer-Relating Fine-Tuning Comparison

Models	P@1	F1@3	F1@10	MRR@10	NDCG @10
BERT	**18.50**	**13.80**	**15.37**	**29.96**	**37.82**
SciBERT	8.10	9.13	12.22	19.53	28.76
RoBERTa	10.50	9.83	10.25	20.42	28.17
BART	14.20	11.07	11.37	24.29	31.74

Table 7: Results on Comparing the Employment of Different Pre-trained Language Models (fine-tuned with CLOTH dataset)

w_0	w_1	w_2	w_3	P@1	F1@3	MRR@10	NDCG @10
0.25	**0.25**	**0.25**	**0.25**	18.50	13.80	29.96	37.82
0.4	**0.2**	**0.2**	**0.2**	**19.40**	15.33	31.11	39.12
0.6	**0.15**	**0.15**	**0.1**	19.30	**15.50**	**31.26**	39.49
0.8	**0.05**	**0.05**	**0.1**	18.90	15.43	30.88	**39.56**

Table 8: Distractor Selector Features Weighting Comparison

literature and DGen is a dataset related to scientific domains.

4.4.2 Results on CLOTH dataset

In this experiment, we evaluate the performance of our models on CLOTH dataset and conduct ablation studies for our CDGP model.

Comparing Fine-Tuning Strategy In this set of experiment, we compare the performance of naive fine-tuning and answer-relating fine-tuning. The results are presented in Table 6.

From the above results, it can be observed that the overall score of answer-relating fine-tuning is higher than that of naive fine-tuning. Therefore, we select answer-relating fine-tuning as a default fine-tuning strategy.

Comparing Pre-trained Language Models In this set of experiment, we experiment with using different pre-trained language models. The following are the pre-trained language models used in the experiments. (1) BERT (Devlin et al., 2018), (2) SciBERT (Beltagy et al., 2019), (3) RoBERTa (Liu et al., 2019), (4) BART (Lewis et al., 2019).

Table 7 shows the comparison result. Through this experiment, we see that the BERT model has the most outstanding performance, so we use the BERT model for subsequent experiments.

Comparing DS Factors There are four scoring factors in DS, namely s_0 (confidence score), s_1 (word embedding similarity), s_2 (contextual sentence similarity) and s_3 (part-of-speech match score). In this experiment, we adjust the weighting of each scoring index of DS (from w_0 to w_3), and compare the difference of using different weight ratios. Table 8 shows the experiment results.

From the results in Table 8, we see that if the weights of s_1 and s_2 is adjusted lower, a better distractor generation performance is observed, but if they are set too low, the performance starts to degrade.

After the experiments, we see that the DS weights setting to (0.6, 0.15, 0.15, 0.1) show the best performance. We use this weighting setting as default values for other experiments.

Comparing w/o CDGP Components Through the above experiment studies, we obtain the besting parameter settings for CDGP. In order to prove the effectiveness of the CDGP design, in this set of experiments, we compare the use or not of each component in the framework. Table 9 presents the experimental results.

From the results, we can see that the whole CDGP framework (CSG+DS with

Methods	P@1	F1@3	F1@10	MRR@10	NDCG @10
CSG+DS	**19.30**	**15.50**	**15.37**	**31.26**	**39.49**
CSG	18.50	14.90	15.37	30.57	38.73
DS	4.00	6.43	5.05	12.02	19.12
None	4.10	6.03	5.05	11.81	18.65

Table 9: Ablation study on CDGP components

$(w_0, w_1, w_2, w_3) = (0.6, 0.15, 0.15, 0.1))$ shows the best performing results compared with the options using only one or none of the components. Furthermore, we see that using only CSG improves the performance (107.7%, in terms of NDCG@10, compared with *none* scheme (which uses BERT's MLM capability to have distractor candicate without any fine-tuning), while using only DS brings slightly performance improvement (2.5%). Such results indicate that the major performance improvement comes from the CSG employment.

4.4.3 Result on Human Evaluation

We also recruit 40 human evaluators from our campus. The evaluation process is as follows. First, the evaluator takes a cloze exam (a passage with 10 cloze multiple choice questions). The passages are randomly selected from the CLOTH dataset. For a selected passage, we keep five original questions and replace the rest five questions with the generation results by our model. Our goal is to observe the answering correct rate over the manually designed distractors and the automatically designed distractors. Furthermore, we also ask the evaluators to exam the quality of the generated distractors. Specifically, after the exam, we ask (1) the evaluators to guess which questions are generated by CDGP and (2) rank the distractor difficulty by Likert scale ranging from 1-5.

Answering Correct Rate We find that the correct rate of the human cloze questions is 50.5%, while the correct rate of CDGP questions is 66%. The correct rate of CDGP questions is slightly higher than that of human questions, which shows that the difficulty of CDGP distractors is slight easier than that of human questions. Improving and controlling the difficulty of automatically generated distractors will be an interesting future work direction.

Distinguishing Human-design or CDGP Question In the test of judging whether a question is a CDGP question, the correct rate of the evaluators' guess is 53%, which nearly to a random guess, showing that the evaluator cannot effectively distinguish between human and CDGP questions.

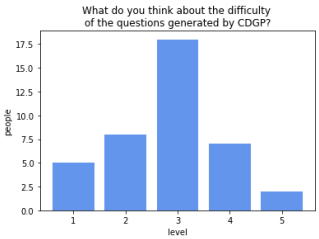

Figure 3: The testers' feedback on the difficulty of the questions generated by CDGP (1: easiest, 5: most difficult)

Examining Difficulty of Generated Distractors From the tester feedback, as shown in Figure 3, the testers' ratings of difficulty are normally distribution, indicating that the difficulty level of the questions is moderate. It can be seen that the performance of CDGP questions is close to that of manual-design questions, which confirms that CDGP can assist in the cloze distractor preparation.

5 Conclusion

Our study indicates that PLM-based candidate distractor generator is a better alternative for knowledge-based component. The experiment results show that our model significantly surpassed the SOTA method, demonstrating the effectiveness of PLM-based distractor generation on Cloze Test. Also, the result shows that using domain-specific PLM will further boost the generation quality.

6 Limitations

The major limitation for this study is that the current evaluation on the test dataset cannot truly reflect the distractor generation quality. A mismatch with the ground truth distractors do not imply the generated distractor is not a feasible one. Also, we have no way to control the difficulty and the correctness of distractor generation.

Acknowledgement

This work is supported by NSTC 110-2634-F-005-006-project Smart Sustainable New Agriculture Research Center (SMARTer), NSTC Taiwan Project under grant 109-2221-E-005-058-MY3, and Delta Research Center, Delta Electronics, Inc. We thank to National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP*. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Rui Pedro dos Santos Correia, Jorge Baptista, Nuno Mamede, Isabel Trancoso, and Maxine Eskenazi. 2010. Automatic generation of cloze question distractors. In *Second language studies: acquisition, learning, education and technology*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

John Lee and Stephanie Seneff. 2007. Automatic generation of cloze items for prepositions. In *Eighth Annual Conference of the International Speech Communication Association*. Citeseer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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

Annamaneni Narendra, Manish Agarwal, and Rakshit Shah. 2013. Automatic cloze-questions generation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 511–515.

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.

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

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*.