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. Thus, automatic cloze test generation is motivated by research community. In this paper, we extend the traditional candidate-ranking framework by exploring the employment of pre-trained language models (PLM) 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).

1 Introduction

In this paper, we investigate automatic cloze test generation based on pretrained language models. 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.

Manually designing cloze test consumes enormous time and efforts. A cloze question (as illustrated in Figure 1) is composed by (a) a sentence with a word removed (a blank space) and list of options (one answer and three wrong options). The challenge lies in wrong option (distractor) selection. Having carefully-design distractors improves the effectiveness of learner ability assessment, but take time and efforts. As a result, automatic cloze distractor generation is motivated (Ren and Q. Zhu, 2021)(Jiang and Lee, 2017)(Kumar et al., 2015).

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 **Distractors** (D) living

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 the NDCG@10 score from 19.31 to 34.17 (nearly 1.7 times the improvement)

2 Related Work

The methods on cloze distractor generation (Jiang and Lee, 2017)(Kumar et al., 2015)(Correia et al., 2010)(Lee and Seneff, 2007) 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 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 domain

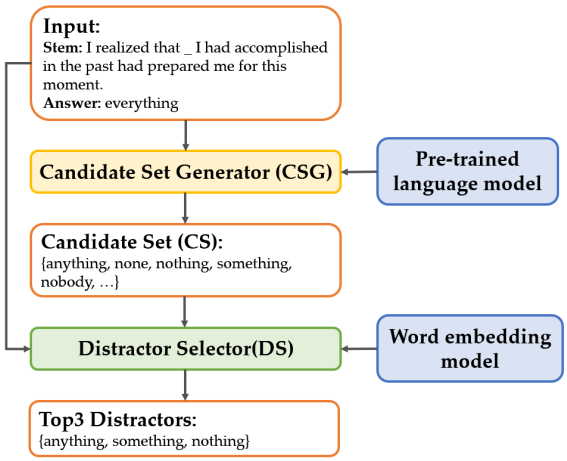

Figure 2: CDGP Framework

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 linguistc 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 distractor 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**:

$$\mathbb{M}(S[\texttt{Sep}][\texttt{Mask}]) \rightarrow D$$

For a given stem S, we concatenate it with a [Sep] and a [Mask] token as the input to PLM. The idea is to adapt the original MLM ability by fine-tuning the PLM. 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,

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

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 Q, an answer A, and a candidate set $\{D_i\}$ from CDG. 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 similarity between the stem with the blank filled in A (denoted by $\vec{S}_{\otimes A}$) and the stem 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. $s_4 = 1$, if A and D_i has the same POS tag. Otherwise $s_4 = 0$.

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-k scores are selected as the final resultant distractors.

4 Performance Evaluation

4.1 Dataset

To validate the performance of our methodology, we use the CLOTH datatset (Xie et al., 2017). The dataset comes from English cloze exercises. The datasets consists of a passage with cloze stems, answers and the corresponding answers.

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 settin 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 embedding similarity in DS, we use the fasttext model (Wu and Manber, 1992) 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 default 10.

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), and Normalized Discounted Cumulative Gain (NDCG@10).

Human Evaluation Score 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 CLOZE dataset. For a selected passage, we keep five original questions and replace the rest five questions with the generation results by our model. After the exam, we ask (1) the evaluators to guess which questions are generated by AI and (2) rank the distractor difficulty by Likert scale ranging from 1-5.

4.4 Evaluation Results

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 1.

Models	P@1	F1@3	F1@10	MRR	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 1: The Result of Naive and Answer-Relating Fine-Tuning Comparison

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

Table 2: Results on Comparing the Employment of Different Pre-trained Language Models (trained with CLOTH)

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 in subsequent experiments.

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

Table 2 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 Features There are four scoring features 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 weight ratio of each scoring index of DS, and compare the difference of using different weight ratios. Table 3 is the experiment result.

From the results in Table 3, 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. For such outcomes, please refer to our discussion in Appendix.

After the experiments, it is found that the DS feature weights setting to (0.6, 0.15, 0.15, 0.1) show the best performance. We use this weighting setting in all subsequent experiments.

w_0	w_1	w_2	w_3	P@1	F1@3	MRR	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 3: Distractor Selector Feature Weighting Comparison

Methods	P@1	F1@3	F1@10	MRR	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 4: Comparison on CDGP components

4.4.1 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 4 presents the experimental results.

It can be seen from the experimental results that the generation based on the CDGP framework has a significant improvement compared with the use of the pre-training model alone, which proves that CDGP's effectiveness on the cloze distractor generation. It can also be seen that using only CSG improves the performance, and using only DS brings slightly performance improvement (0.8%). Such results indicate that the major performance boosting comes from the CSG employment.

4.4.2 Comparison with the SOTA Method

We also compare CDGP with the SOTA method (Ren and Q. Zhu, 2021). We use 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). The question domain is scientific passage. Table 5 shows the comparison results.

From Table 5, it can be seen that the NDCG@10 of CDGP has increased 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 based on SciBERT show the best results. We think this confirms the domain matches between DGen dataset and SciBERT pretrained corpus.

Models	P@1	F1@3	MRR	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_BART	8.49	8.24	16.01	22.66
CDGP_RoBERTa	13.13	9.65	19.34	24.52

Table 5: Comparing CDGP with the SOTA method

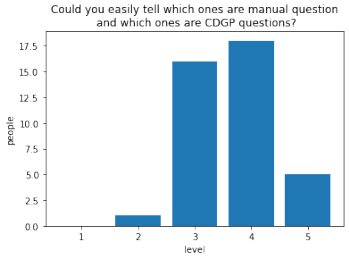

Figure 3: The testers' feedback of whether they can recognize the difference between manual and CDGP's distractors (1: most easiest, 5: most difficult)

4.4.3 Result on Human Evaluation

The results of human evaluation show that in terms of the correct answer rate of testers, the correct rate of the human cloze questions (designed by teacher) is 50.5%, and 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 perplexity of CDGP distractors is still different from that of human questions.

In the test of judging whether a question is an AI question, the correct rate of the tester's guess is 53%, which nearly to a random guess. It shows that the tester cannot really distinguish between manual and CDGP questions.

From the tester feedback, shown in Figure 3, 58% testers score the generation quality higher than 4. It can be seen that the performance of CDGP questions is very close to that of manual questions, which confirms that CDGP has the ability to assist in the cloze distractor preparation.

5 Conclusion

Our study indicates that PLM-based candidate 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. 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.

References

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

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.

Shu Jiang and John SY Lee. 2017. Distractor generation for chinese fill-in-the-blank items. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148.

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.

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*.

Sun Wu and Udi Manber. 1992. Fast text searching: allowing errors. *Communications of the ACM*, 35(10):83–91.

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*.

A Appendix

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 6: The statistics of the training, development and test sets of CLOTH-M (middle school), CLOTH-H (high school) and CLOTH. (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 7: 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)

DS feature discussion From the experiment on DS feature weighting, we see that lowering the weight of word and sentence similarity will enhance the performance of the model, but at the same time, it should be noted that too low weight will also lead to violation of reliability. We have the following discussion for such results.

First, the reason of considering word-level and sentence level similarity scores is that we cannot avoid the generation of answer synonyms by CSG, so the higher the similarity, the lower the score. In this case, however, distractor options that are semantically similar but not identical in meaning to the answer also have lower scores, and these ideal distractors rank worse. Therefore, if the weight of the similarity of words and sentences is adjusted lower, the higher the score of these ideal distractors will be, as shown in Table 8 and Table 9. But if it is set too low, the candidates which are synonymous of the answer will get a higher score and have a chance to be selected as final distractor, thus violating reliability, as shown in Table 10. In Table 10, it can be seen that the ranking of the answer's synonym (everything) has risen from 9 to 5. In other cases, if the confidence score in CSG is high enough, it will have the chance to become the final selected distractor.

Generation Examples We also show three generation examples in Table 11, 12, and 13. In Table 11, we see there are two generated distractors are the same as the ground truth. Also, the third generated example "suggestion" is also a feasible one. We have also similar observation for the rest examples.

Stem: I realized that __ I had accomplished in the past had prepared me for this moment.
Answer: everything
Distractors: anything, nothing, something

Ranking of the DS:

1. (none, 0.7572)
2. (anything, 0.5430)
3. (something, 0.4480)
4. (nothing, 0.4437)
5. (nobody, 0.4123)
6. (somebody, 0.3849)
7. (someone, 0.3764)
8. (any, 0.2746)
9. (everything, 0.2687)
10. (somewhere, 0.1555)

Table 8: DS Features(0.25, 0.25, 0.25, 0.25)

Answer: everything
Distractors: anything, nothing, something

Ranking of the DS:

1. (anything, 0.7258)
2. (something, 0.5058)
3. (nothing, 0.4858)
4. (none, 0.4173)
5. (nobody, 0.1986)
6. (somebody, 0.1838)
7. (someone, 0.1758)
8. (any, 0.1683)
9. (everything, 0.1449)
10. (somewhere, 0.0966)

Table 9: DS Features(0.6, 0.15, 0.15, 0.1)

Answer: everything
Distractors: anything, nothing, something

Ranking of the DS:

1. (anything, 0.9086)
2. (something, 0.6179)
3. (nothing, 0.5880)
4. (none, 0.2231)
5. (everything, 0.1599)
6. (nobody, 0.1339)
7. (somebody, 0.1319)
8. (someone, 0.1253)
9. (any, 0.0609)
10. (somewhere, 0.0366)

Table 10: DS Features(0.8, 0.05, 0.05, 0.1)

Stem	I made my __ to start my own company and leave my worry-less position after attending a regional sales meeting.		
Answer	decision		
Manual	**plan** **promise** mind	**CDGP's**	**plan** **promise** suggestion

Table 11: CDGP example 1

Stem	Having __ a course as a reading adviser, I can now help others to read and write.		
Answer	completed		
Manual	**attended** **organized** started	**CDGP's**	**attended** **organized** taught

Table 12: CDGP example 2

Stem	And they wondered __ I would risk everything for a dream.		
Answer	why		
Manual	**how** **when** if	**CDGP's**	**how** **when** where

Table 13: CDGP example 3