Distractor Generation Using Generative and Discriminative Capabilities of Transformer-based Models

Shiva Taslimipoor†, Luca Benedetto†, Mariano Felice†‡, Paula Buttery†
†ALTA Institute, Department of Computer Science & Technology, University of Cambridge, U.K.
‡British Council, U.K.
{firstname.lastname}@cl.cam.ac.uk

Abstract

Multiple Choice Questions (MCQs) are very common in both high-stakes and low-stakes examinations, and their effectiveness in assessing students relies on the quality and diversity of distractors, which are the incorrect answer options provided alongside the correct answer. Motivated by the progress in generative language models, we propose a two-step automatic distractor generation approach which is based on text-to-text transfer transformer models. Unlike most previous methods for distractor generation, our approach does not rely on the correct answers. Instead, it first generates both correct and incorrect answer options, and then discriminates between potential correct options and distractors. Identified distractors are finally grouped into separate clusters based on semantic similarity, and cluster heads are selected as our final distinct distractors. Experiments on two publicly available datasets show that our approach outperforms previous models both in the case of single-word answer options and longer-sequence answers for reading comprehension questions.

Keywords: multiple-choice questions, distractor generation, transformers

1. Introduction

Automatic distractor generation (DG) refers to the process of generating plausible incorrect answers, also known as distractors, for Multiple-Choice Questions (MCQs). MCQs are widely used to test language learners. Generating suitable distractors can be a very time-consuming process for item writers. The aim of DG is to automate the creation of challenging exercises that can accurately assess students knowledge and understanding.

Automated DG is particularly challenging due to the difficulty of defining what characterises good distractors, as they must be semantically and syntactically coherent with the correct answer but unambiguously wrong. Also, they should not be obviously incorrect, as this would make them easy to detect and therefore unhelpful. In other words, *ideal* distractors can be thought of as being very similar to the correct answer option in most aspects, but different in at least one of them, which is what makes them incorrect. Distractors are often built to try and capture common misconceptions and comprehension errors of students, but this process can be fairly subjective and therefore very difficult to automate.

Before the popularity of neural models, natural language processing methods for automatic DG involved choosing distractors that shared the same part-of-speech (POS) with the correct answer (Susanti et al., 2015), had a similar frequency of occurrence to the correct answer (Jiang and Lee, 2017), had a similar semantic representation based on distributional similarity (Afzal and Mitkov, 2014), or followed certain patterns or rules, such as synonyms, antonyms, hyponyms, hypernyms, etc (Correia et al., 2010). When designing conceptual questions, anthologies such as WordNet or term extraction methodologies have also been used (Mitkov et al., 2006). However, these similarity-based models are susceptible to i) generating alternative correct answers – which leads to poor test items with more than one correct answer – and ii) creating multiple distractors that are almost the same and thus interdependent (i.e., a student would know that eliminating one implies eliminating the other). Combinations of the above-mentioned features are also used in machine learning models to rank distractors (Liang et al., 2018).

Recently, generative neural models have proved effective in DG (Gao et al., 2019; Liang et al., 2017), although some of this work is focused on other goals (e.g., summarisation (Manakul et al., 2023)), thus giving limited attention to evaluating the quality of the generated distractors. Also, they mostly generate distractors given the correct answer option (Vachev et al., 2022), and/or focus on proposing one distractor only (Gao et al., 2019). Unlike previous work, we propose a two-step approach. First, we use transfer learning with text-to-text transformer models to generate a combination of correct answers and distractors. Second, we classify the generated options into correct answers and distractors, with the help of clustering to remove duplicate distractors. Since correct answers and distractors can be expressed in many different ways and are rarely unique, we hypothesise that generative models can benefit from predicting both

LREC-COLING 2024, pages 5052–5063
20-25 May, 2024.

of them together.

We experiment with two different reading comprehension Multiple-Choice Question Answering (MCQA) tasks: i) cloze tests (*fill-in-the-gap* exercises) with single-word answer options and ii) standard reading comprehension questions, which require longer sequences as answer options. The contributions of this work can be summarised as follows: i) we improve DG by generating distractors and correct answers together, ii) we apply clustering to minimise the number of duplicate distractors, iii) we show that our approach outperforms previous models through extensive experimentation, and iv) we carry out human evaluation to get more accurate perceptions of the usefulness of our approach.

2. Background: Question Answering Using Text-to-text Transformers

When using neural models, MCQA is typically modelled as a sequence classification problem: an encoding module extracts contextual semantic representations from the text – the context (i.e., the reading passage), the question, and the answer options – and a following classifier layer chooses the correct answer from the options. The input sequence to such a model is usually encoded as a concatenation of i) the question, ii) one of the answer options, and iii) the context, and the task is to predict whether the answer option in the sequence is correct or not (Lai et al., 2017).

Recently, transformer-based models using self and cross-attention mechanisms proved successful in MCQA, where most systems have a *softmax* layer over the outputs of the four options to perform the classification (Chen et al., 2016). However, T5 (Raffel et al., 2022) is a text-to-text generative transformer model which has a decoder rather than a classification layer to model the output. This gives the advantage of modeling all NLP tasks using a unified text-to-text format. For reading comprehension MCQs, as shown in Figure 1, it adds the texts "`multirc question:`" as the task-specific prefix to the input question, "`answer:`" as the prefix to the answer option, then it generates the texts `True` or `False` for each answer option.

T5 has also been trained for non-MCQs such as SQUAD (Rajpurkar et al., 2016), where it takes the question and the context paragraph, and generates the full span of the answer sequence. Following Khashabi et al. (2020) we find that by fine-tuning on appropriate training samples, this text-to-text transformer model is suitable for generating both correct answer options and distractors.

Input: **`multirc question:`** `What kind of room is it?` **`answer:`** `It's a bedroom.` **`paragraph:`** `Here are the twin sisters, Lily and Lucy. They are in Miss Gao's class. They are two new students. They're eleven. This is their room. It's a nice room. There are two beds in the room. One is Lucy's and the other is Lily's. They look the same. Their coats are on their beds. We can't see their shoes. The twins have two desks and chairs. Their clocks, books and pencil boxes are on the desks. Their schoolbags are behind the chairs.`
Target: `True`

Figure 1: An example of input and target representation for multiple choice reading comprehension question answering using T5.

3. Answer and Distractor Generation

In automatic DG, the quality of a distractor is established based on two factors: *plausibility* and *incorrectness* (Qiu et al., 2020). *Plausibility* implies that the distractor should be syntactically and semantically relevant to the question and also contextually relevant to the passage. We learn this by using a pre-trained transformer-based language model representing the combination of the question and the passage, where the task is to generate answer options (see Section 3.1). *Incorrectness* indicates that while being relevant to the question and the passage, the distractor should be unambiguously different from the correct answer and wrong. We achieve this by training a classifier which – given the passage, the question, and one answer option – determines whether the answer option is correct (see Section 3.2).

Most DG systems take the original correct answer as input while outputting one distractor (Gao et al., 2019). However, we argue that even correct answers are not always unique, so having the model learn them alongside the distractors would be a better approach. T5 is particularly suitable for this. Specifically, we propose a two-step approach where we first generate a combination of correct and incorrect answers for each question (see Section 3.1), and then design a discrimination model to distinguish distractors from correct answers and group them into dissimilar clusters where cluster heads constitute our final distinct distractors (see Section 3.2).

Input	Target
<question> <context>	<option1>
<question> <context>	<option2>
<question> <context>	<option3>
<question> <context>	<option4>

Figure 2: Overview of how four data entries are created from one MCQ with question <question>, context <context>, and four answer options <option1>, <option2>, <option3>, <option4>, during the generation step.

3.1. Text-to-text generation of the answer options

The first step in our model is to generate correct answers and distractors for each question. We employ the T5 model, which is a state-of-the-art question answering architecture.[1] The publicly available pretrained T5 model is originally trained using standard maximum likelihood in the form of unsupervised denoising objectives similar to BERT, as explained in Raffel et al. (2022). We fine-tune the model by providing it with additional training data. Each training example is the concatenation of the question and the reading passage, separated by a delimiter. The desired output is one of the four answer choices (without distinction between correct answer and distractors). Figure 2 shows the structures of the four data points extracted for one MCQ with the question (question), the context paragraph (context), and options (option1, option2, option3 and option4). As this generation task is different from standard question answering models (since we generate incorrect answer options as well), we do not add to the input the question answering prefixes used in the original T5 model. At the time of inference, we specify the number of answer options, which the system generates using a beam search.

3.2. Binary classification of correct-incorrect answer options

The second step of our approach is to classify the generated options into correct and incorrect answers. Both generative encoder-decoder style models (e.g., T5) and encoder style models (e.g., ELECTRA (Clark et al., 2020)) can be used for this step. T5 is initially pretrained for multiple choice reading comprehension. The T5 input formatting for this step is a bit different from the previous one (Figure 2) as the question is followed by one answer option, followed by the context, all separated by a delimiter, and the target output is either True or False. This makes the generative T5 model suitable for a discrimination task by generating True/False values. The input sequence is also augmented with the prefixes “multirc question:" before the question, “answer:" before the answer option, and “paragraph:" before the context. [2] Therefore, fine-tuning T5 with appropriate prefixes is suitable for this task.

We also experimented with fine-tuning ELECTRA for sequence classification. However, we stuck with T5 for the discrimination as well as the generation step since the difference in results was negligible.

Post-processing The decoder layer in step 1 (Section 3.1) generates *N* answer options, and usually many of them are conceptually similar. To overcome this issue and to have a set of diverse distractors, we perform semantic clustering. We use Sentence Transformers (Reimers and Gurevych, 2019) to extract vector representations for the predicted distractor options, which we use for agglomerative clustering[3]. Agglomerative is a common type of hierarchical clustering used to group objects into clusters based on their similarity/distance. It starts by treating each vector representation as a singleton cluster and recursively merges pair of clusters until a distance or similarity threshold is reached. We use Euclidean distance between clusters as the metric and 1.2 as the threshold, selecting these two values after performing some preliminary experiments to prove their effectiveness. The heads of different clusters are selected as the final set of distractors. Among all sequences in a cluster, we choose the cluster head to be the sequence which has the highest probability (confidence) score based on the generation model.

4. Experimental setup

4.1. Datasets

We run our experiments on two language learning MCQA datasets: CLOTH (Xie et al., 2018), and RACE (Lai et al., 2017).

CLOTH includes cloze test paragraphs, each with up to 20 gaps, and four single-word options for each gap. Neither T5 nor any of its variations are pre-trained for cloze test QA. In order to represent the input for CLOTH items, we separately extract the sentences with a gap (a gap is shown with an

[1] https://huggingface.co/docs/transformers/v4.14.1/model_doc/t5

[2] In the original T5, the sentences in a paragraph are split and prefixed by their sentence number. We skip this sentence splitting process.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

underscore) and consider them as the questions[4]. The question sequence is followed by the single-word answer option (for the discrimination phase), followed by the whole text (with gaps) as the context. We use the same prefixes in T5 as described in Section 3.2. The output for the generation phase are single-word answer options, and the output of the discrimination phase is either `True` (for the correct answer) or `False` (for distractors).

RACE is a large-scale Reading Comprehension Multiple Choice Question Answering (RC-MCQA) dataset, collected from English examinations in China. It includes passages containing multiple questions, each with 4 answer options, one of them being correct. RACE is a standard question answering dataset which has been popularised for distractor generation by Gao et al. (2019), who modified the dataset by filtering out distractors that were not relevant to the articles. They also removed the fill-in-the-blank questions with gaps at the beginning or in the middle of the questions. For a fair comparison of our work with Gao et al. (2019) and Qiu et al. (2020), we report our results on the modified version of RACE, also known as RACE-DG.

4.2. Evaluation metrics

For automatic evaluation, we first focus on exact matching between original and generated distractors. CLOTH answer options are single words, so we report *precision*@1 (*P*@1), *F*1@3, and *NDCG*@10 (Normalized Discounted Cumulative Gain), which are rank-based evaluation metrics used in previous single-word DG studies (Ren and Q. Zhu, 2021). In contrast, RACE contains longer sentences as answer options, therefore we report the BLEU scores (BLEU1, BLEU2, BLEU3, and BLEU4) for generating the 1st, the 2nd and the 3rd distractors, as in Gao et al. (2019).

Following previous work that challenged the adequacy of machine translation metrics such as BLEU for evaluating DG, we also use similarity-based metrics (Rodriguez-Torrealba et al., 2022). Specifically, we report the semantic similarity between i) the generated distractors and the correct answer, ii) the generated distractors and the reference distractors available in the dataset, and iii) the generated distractors themselves. We use SentenceTransformer (Reimers and Gurevych, 2019) (as explained in the post-processing step in Section 3.2), to extract embedding vectors for sequences and then calculate cosine similarity between the vectors.

zero-shot GPT:
"Generate 10 plausible but incorrect answers for the following question.
Question: {reading passage} {question text}
Answer: {answer text}"

one-shot GPT:
"Generate 10 plausible but incorrect answers for the following question.
Question: {reading passage} {question text}
Answer: {answer text}
Incorrect answers: {known distractors}
Question: {reading passage} {question text}
Answer: {answer text}
Incorrect answers:"

Figure 3: Prompts for zero-shot and one-shot GPT.

Finally we perform human evaluation on a subset of questions from RACE-DG, to get an additional perspective towards the generation capabilities of the proposed model.

4.3. Models

We use several baselines taken from previous literature. Chiang et al. (2022) use a **BERT**-based model to generate single-word distractors; therefore, it can be used only on the CLOTH dataset. Gao et al. (2019) propose a Hierarchical Static Attention (**HSA**) mechanism to generate distractors for reading comprehension questions. **EDGE** (Qiu et al., 2020) follows Gao et al. (2019) and is a combination of LSTM, self-attention and gated layers to encode the passage and question. **Baseline T5** (Manakul et al., 2023) is a standard T5 model, trained to generate three distractor options when given the context, the question, and the correct answer. We use the publicly available code for implementing this model. [5]

Following Bitew et al. (2023), we evaluate DG with **GPT-3.5** in a zero-shot and one-shot fashion. The prompts used for these models are shown in Figure 3. Finally, our model, **two-step DG** uses pretrained T5 and is fine-tuned using the architecture explained in Section 3.

5. Results

We first report the experiments on CLOTH (Section 5.1) and RACE (Section 5.2), then present the results of the human evaluation in Section 5.3 and a study of how our model performs on different types of questions in Section 5.4.

[4] If the sentence has more than one gap, all other gaps are replaced with a random word. Here we choose the one-character stopword 'a'. We do not replace the gaps with their original words in order not to make the answer generation task easier for the system by giving away the correct answers to other gaps.

[5] `https://huggingface.co/potsawee/t5-large-generation-race-Distractor`

Models	P@1	F1@3	NDCG@10
Baseline T5	9.22	10.29	27.5
BERT	18.50	13.80	37.82
two-step DG	**26.57**	**22.05**	**47.29**

Table 1: Performance of different systems on generating distractors for CLOTH questions.

5.1. Single-word cloze items

All the answer options in the CLOTH dataset are single words, therefore we use information retrieval measures based on string matching for evaluation, as explained in Section 4. Table 1 presents the results of the quantitative comparison between our system (two-step DG), Baseline T5 (Manakul et al., 2023), and the BERT-based baseline (Chiang et al., 2022). Since the experiments by Chiang et al. (2022) are performed with *bert-base*, we use *t5-base* for our system and Baseline T5.

Results show that our system significantly outperforms the two baselines on all metrics. Specifically, *P*@1 shows that on average the first distractor generated by our model is relevant (i.e., matches one of the gold distractors in the dataset) for more than 26% of questions, almost 50% higher than the BERT-based model and almost three times the precision of the T5 baseline. Improvements are slightly lower for the other two metrics but still significant, both considering *F*1@3, which is the weighted average of precision and recall when generating three distractors, and *NDCG*@10, which measures the effectiveness of the first ten generated distractors (and gives higher scores when the relevant distractors are ranked higher in the list).

5.2. Reading comprehension MCQA

5.2.1. BLEU scores

Table 2 presents a comparison of the BLEU scores achieved by our model and previous approaches on the RACE-DG dataset, focusing separately on the first three distractors generated by each model and averaging the scores across them. We focus on three distractors for our evaluation because that is the original question format in RACE.

Starting from the average scores between distractors (the last row in the table), we can see that the proposed two-step DG model is better than all the baselines for generating longer sequences that match the original distractors. This is captured by BLEU2, BLEU3, and BLEU4, which look at two-, three-, and four-word matching between the generated distractors and the reference: the improvement over the baselines is particularly visible for longer sequences (BLEU4). Considering BLEU1, which measures the capability of generating distractors that match one word of the original distractors, EDGE is the best model overall, followed by our two-step DG model and HSA. EDGE is particularly good in terms of BLEU1, but its accuracy is much lower when considering longer sequences.

Baseline T5 is capable of generating one very accurate distractor, but its performance drops significantly for the second and third distractors – making it the worst model considering the BLEU scores on the third distractor. The GPT models are consistently outperformed by our two-step DG model and EDGE – which prove to be the best and second best models overall – but the performance of one-shot GPT is closer to the state of the art compared to the zero-shot model.

5.2.2. Similarity based evaluation

As explained in Section 4.2, an analysis based only on BLEU scores can give only partial insight into the quality of the generated distractors, since it is based on string matching between the generated distractors and the reference. Therefore, we also perform further analyses based on the semantic similarity of the generated distractors. Specifically, we study the similarity between the generated distractors and the original correct answer and distractors, as well as the similarity among the generated distractors themselves. Table 3 shows the semantic similarity based scores for our model and the three baselines that we re-implemented: zero-shot GPT, one-shot GPT, and Baseline T5.

The first column of results (*gd2c*) in the table shows the similarity of the predicted distractors to the original correct answer. Previous work (Rodriguez-Torrealba et al., 2022) argue that plausible distractors have higher similarity to the correct answer. While we find this very debatable, we report this measure and compare it with the average similarity of the original distractors to the correct answers (shown in the first row).

The next three columns (*gd2d*) show the similarities of the three generated distractors to the reference distractors in the dataset. The higher similarities between the predicted distractors and the original ones show that the model has better abilities to generate relevant distractors. As can be seen in the table our model outperforms other models in generating more similar distractors to the original data, one-shot GPT improves upon the zero-shot GPT, but still slightly underperforms our model. Consistent with the results in Table 2, the strength of Baseline T5 lies in generating the first distractor, but it lags behind when generating more than one distractor.

Lastly, column *gd2gd* shows the average pairwise similarity among the three generated distractors. Since we are aiming for generating distractors which are conceptually distinct – so that rejecting

Generated distractor	System	BLEU1	BLEU2	BLEU3	BLEU4
1st distractor	HSA (Gao et al., 2019)	0.28	0.15	0.09	0.06
	EDGE (Qiu et al., 2020)	0.33	0.18	0.11	0.08
	Baseline T5 (Manakul et al., 2023)	**0.34**	**0.23**	**0.16**	**0.12**
	zero-shot GPT (Bitew et al., 2023)	0.25	0.13	0.07	0.04
	one-shot GPT	0.28	0.16	0.10	0.06
	two-step DG	0.31	0.20	0.15	**0.12**
2nd distractor	HSA (Gao et al., 2019)	0.28	0.13	0.08	0.05
	EDGE (Qiu et al., 2020)	**0.32**	0.17	0.1	0.06
	Baseline T5 (Manakul et al., 2023)	0.21	0.13	0.09	0.07
	zero-shot GPT (Bitew et al., 2023)	0.23	0.12	0.06	0.04
	one-shot GPT	0.27	0.14	0.09	0.05
	two-step DG	0.29	**0.18**	**0.13**	**0.09**
3rd distractor	HSA (Gao et al., 2019)	0.27	0.13	0.07	0.05
	EDGE (Qiu et al., 2020)	**0.31**	0.16	0.09	0.06
	Baseline T5(Manakul et al., 2023)	0.04	0.02	0.02	0.01
	zero-shot GPT (Bitew et al., 2023)	0.22	0.11	0.06	0.04
	one-shot GPT	0.26	0.14	0.08	0.05
	two-step DG	0.27	**0.17**	**0.12**	**0.08**
Average	HSA (Gao et al., 2019)	0.28	0.14	0.08	0.05
	EDGE (Qiu et al., 2020)	**0.32**	0.17	0.10	0.07
	Baseline T5(Manakul et al., 2023)	0.20	0.13	0.09	0.07
	zero-shot GPT (Bitew et al., 2023)	0.23	0.12	0.06	0.04
	one-shot GPT	0.27	0.15	0.09	0.05
	two-step DG	0.29	**0.18**	**0.13**	**0.10**

Table 2: Performance of different systems when generating distractors for RACE. We report BLEU scores for the top-three generated distractors separately and on average.

	gd2c	gd2d ↑			gd2gd ↓
		d1	d2	d3	
Gold	46.79	-	-	-	33.80
Baseline T5	49.25	**56.26**	37.90	17.73	58.55
zero-shot GPT	42.21	48.06	46.79	46.29	51.99
one-shot GPT	43.00	50.07	48.81	47.49	49.87
two-step DG	42.28	54.00	**51.84**	**49.39**	**43.56**

Table 3: Semantic similarity scores between i) the generated distractors and the correct answer (*gd2c*), ii) the generated distractors and the reference distractors (*gd2d*, higher is better) for the three generated distractors d1, d2, and d3 separately, and iii) the generated distractors themselves (*gd2gd*, lower is better). The first row presents the similarity values of the original distractors in the dataset.

one does not entail rejecting the other – we believe that lower semantic similarity between the generated distractors leads to better sets of distractors. In this sense, our model outperforms all the baselines, and our hypothesis that lower *gd2gd* is better, is also supported by the average similarity among the distractors in the original dataset, which is reported in the first row.

5.3. Human evaluation

To better evaluate the quality of the distractors generated by our model, we also perform human annotation on a subset of the RACE-DG dataset. Three annotators were shown 12 reading passages, each containing 4 or 5 questions, for a total of 53 questions. Passages were shown one at a time. Each question was followed by the correct answer (`key`) and five generated distractors (for a total of 265).

The annotators were asked to label each distractor as *acceptable*, *unacceptable* or *uncertain* (in case they could not decide between the two). Annotators were instructed to consider the set of distractors for a question as a whole. This assures that the set is ready to be used as is. In other words, if the model generates two acceptable distractors which are too similar, only one of them should be labelled as *acceptable*. We use *majority* voting to merge the labels from the three annotators, to reduce bias and subjectivity. Inter-annotator agreement is moderate, with Krippendorff's α = 0.43. We show the annotation guidelines in Appendix A.

We present the results of two systems, our proposed two-step DG and one-shot GPT, to the annotators at random. Specifically, each annotator encounters and annotates all the 53 questions twice, with the difference being that the generated distractors are the result of two different models. Annotators are aware that they encounter each question twice in a randomised order, and they are blind to which model generated the distractors.

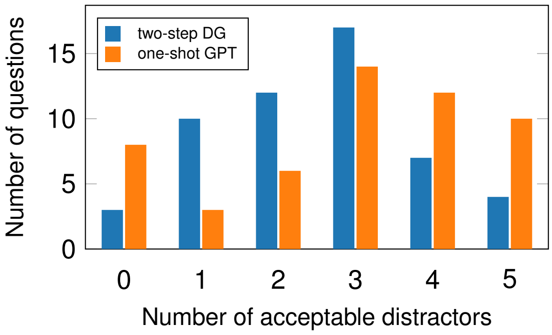

Figure 4: Human annotation results for one-shot GPT and our proposed two-step DG model.

Figure 4 shows the results of the annotation for the two models. Specifically, the bar chart shows the number of questions that contain any number of acceptable distractors (from 0 to 5). Overall, human annotators tend to prefer the GPT model, as shown by questions with 3 or more acceptable distractors. However, our model is more likely to generate at least one acceptable distractor for almost all questions (50 out of 53), when compared to one-shot GPT. It is more likely that GPT fails to generate any good distractors (8 out of 53).

In order to investigate further, we categorise the questions in the RACE dataset as either *general* (e.g., '*What is the best title for the passage?*') or *specific* questions, with the difference being that *general* questions could be used for any reading passage (with different answer options). We argue that models perform differently on these two types of questions, and therefore show in Table 4 the annotation results separately for *generic* and *specific* questions, after manually labelling the questions in the annotation set as belonging to either category. Examples of specific and general questions are added to Appendix B.

Specifically, we show on the left the percentage of questions for which at least three distractors were labelled as *acceptable*, and on the right the overall percentages of acceptable distractors. These results show a significant difference between the two classes: while our model is outperformed by one-shot GPT on *generic* questions, performance is very similar on *specific* questions, even though there is a major difference in the size and complexity of the two models. It is worth noting that the annotation guidelines state that an acceptable distractor should have the same sentence structure as the key. [6] Unlike the GPT model, our proposed model does not rely on the original correct answer for generating new distractors. While this is in principle an advantage, we find our system is actually penalised for generating distractors that are not compatible with the key. This motivates future post-processing of the results of our model.

	$\geq$ 3 *acceptable*		% *acceptable*	
Model	Gen.	Spec.	Gen.	Spec.
one-shot GPT	88.9%	57.1%	79.3%	50.8%
two-step DG	50.0%	54.3%	53.3%	48.6%

Table 4: Human annotation results after distinguishing between *generic* and *specific* questions.

5.4. Analysis on the types of questions

Questions in reading comprehension tasks like RACE can be very heterogeneous, ranging from very general questions such as '*what is the best title for the passage*' to very specific questions such as '*What is Jenny doing in the park*?' [7]. Following our human evaluation experiments, we speculate that automatic systems work differently on different types of questions. Therefore, in this section, we analyse the performance of our models on different "types" of questions. We are interested in this analysis when generating both the correct answers and distractors, since our model treats them interdependently by generating them together. In order to have a clearer picture of the performance of our system, we perform these experiments on the original RACE test set, which has the full set of questions, each with 4 answer options. This is because the RACE-DG dataset by Gao et al. (2019) omits many distractors, affecting the performance of any system as there is a smaller number of original reference distractors.[8] The RACE test dataset that we focus on contains 1436 questions. Moreover, in order to have a general overview of the performance of the model, we report recall@N ($R@N$, N = 50) for the generated answer options after classifying them into correct answers and distractors.

In order to categorise question types, we use SentenceTransformers to extract semantic representations for all the questions and cluster them with DBSCAN.[9] DBSCAN is a density-based clus-

[6] The annotation guidelines will be added as an appendix in the final version if the paper is accepted.

[7] Some general questions are arguably subjective and might be seen as lower quality items with respect to the rest. However, evaluating the quality of questions is out of the scope of this work.

[8] While RACE-DG is not a standard MCQA datastet, we had to report our results on it in order to compare our approach with previous work.

[9] https://scikit-learn.org/stable/

tering algorithm that groups nearby points within a certain distance. The algorithm assumes that clusters are regions of high density that are separated by regions of low density. Any data point that is not close to a high density region is considered an outlier. We use DBSCAN to find high density regions that are general questions of high frequency such as, *'what is the best title for this text?'* and *'which is the title of the passage?'*. All specific questions that are related to a specific event in a specific passage and would occur very few times are considered outliers and belong to what is called cluster -1 in DBSCAN and we refer to it as SPECIFIC. We set the parameters of DBSCAN (EPS= 0.6 and min-sample=40)[10] to make sure that we have three clusters as below:

[TRUE-FALSE] questions asking which option is true or false according to the passage (e.g., '*which of the following statements is true according to the article?*'),

[TITLE] questions about the best title for the passage, e.g., '*what is the best title for the passage?*'

[SPECIFIC] questions related to specific information in the passage (e.g., '*what is Jenny doing in the park?*').

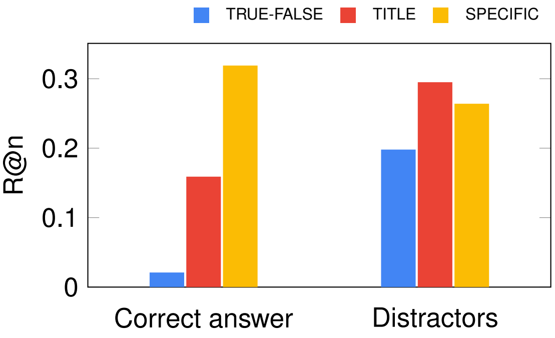

Figure 5: Evaluation of the system performance on different question types; we show *Recall@n* (R@n) separately for the correct answer and the distractors, as well as for different question types.

Figure 5 shows the performance of our model when generating the correct answer and distractors for different types of questions.[11] Although system performance varies greatly when generating the correct answers (i.e. it performs much better at generating answers for specific questions), the difference in performance is less significant when generating distractors. Performance on TRUE-FALSE questions is worse than the ones for TITLE and SPECIFIC questions. While this experiment is a bit different from the question categorisation in Section 5.3 where we have TITLE and TRUE-FALSE as one general category, these results are in line with those, as they show that our model is more robust (than GPT) when generating distractors for different types of questions.

6. Related Work

Earlier work on DG has been motivated by semantic text similarity techniques. Distractors are defined as words that have high similarity to the correct answer (Afzal and Mitkov, 2014). However, there is no guarantee that the correct answer is unique and the identified distractors are not alternative correct answers. Qiu et al. (2020) propose special modules to control for the inherent incorrectness of the generated distractors. Gao et al. (2019) is one of the pioneering works on using encoder-decoder mechanisms to generate long-sequence distractors for reading comprehension questions.

Chung et al. (2020) improve the quality of generated distractors with the purpose of generating multiple distractors, as they criticise the existing methods focusing on single distractors only. Their work introduces 'answer negative loss' to discourage generating distractors that are similar to the correct answer. Leaf (Vachev et al., 2022) uses T5 to fine-tune DG on RACE. Rodriguez-Torrealba et al. (2022) also use T5 to generate distractors. Liang et al. (2018)'s learning to rank method and Ren and Q. Zhu (2021)'s knowledge-driven approach use standard IR-based evaluation measures. Chiang et al. (2022) focuses on generating single-word distractors for cloze questions using BERT.

In all cases, previous models require the correct answer to work. Qiu et al. (2020) encodes the correct answer and reforms the passage and question by erasing any answer-relevant information. However, we believe that only measuring the semantic similarity between each distractor and the correct answer, and adapting the model to maximise that, does not guarantee the incorrectness of the distractor. Also correct answers are not unique, so it seems more natural to let the model produce correct answers and distractors at the same time. Most previous work, however, includes an extra step to remove distractor candidates that are also considered to be valid answers (Susanti et al., 2018).

modules/generated/sklearn.cluster.DBSCAN.html

[10] These parameters can be different depending on the size of the dataset and the diversity of the questions.

[11] Performance is computed as the recall for generating the answer option (either the correct answer or a distractor) among the first 50 generated options.

7. Conclusions and Future Work

This paper proposes a two-step Distractor Generation (DG) model which generates both distractors and correct answer options together, and leverages clustering as a way to avoid generating duplicate distractors. By performing extensive experiments on two publicly available reading comprehension datasets, we show that our proposed model outperforms the previous state of the art according to automatic evaluation metrics. Our semantic similarity analyses show that our model is more effective than similar models at generating a diverse set of distractors that resemble the original distractors. Although automatic evaluation shows that our model outperforms GPT3.5, human evaluation suggests that annotators prefer GPT results. While this is the case in general, we see that the two systems fare equally well on questions eliciting specific information from the passage.

According to human evaluation, some distractors are rejected as they do not follow the same sentence structure as the key. Future work could look into post-processing techniques to make the generated distractors follow a specific format. Motivated by the success of our current two-step approach, we would like to investigate multitask learning as an alternative to our generation and discrimination pipeline. Lastly, we observe that model performance can vary by question type, when classified into *specific* questions (which are specific to the passage) and *general* questions (which can theoretically be used on any passage). We believe this relationship should be further explored, as different models could be better suited to different types of questions, possibly differentiating the training phase as well. A prerequisite for this would be to have an approach for question clustering that is transferable across datasets, which is yet to be explored as another avenue for future research.

8. Limitations

We perform most of our experiments on the RACE-DG dataset in order to compare our performance with previous work. However, this dataset contains both high-quality and low-quality questions, according to our human annotators, which might have an impact on some of our results. Also, the creators of the RACE-DG dataset have filtered out bad quality distractors from the original RACE dataset. While this improves the average quality of the remaining distractors, it also means that not all questions have 4 answer options, which is not completely representative of the standard settings of MCQ exams.

9. Ethical Considerations

The two-step distractor generation model is based on the publicly available pretrained T5 model and is further fine-tuned on the standard RACE dataset. As such, it might be affected by potential bias in the training data and thus produce biased distractors. During the human evaluation, the annotators identified only two distractors which were potentially biased (and should have been removed) out of the 265 they annotated, but this is only a portion of the dataset and there is no guarantee that the same holds true for all the other distractors. Also, it is worth emphasising that these models are developed with the goal of generating distractors to assess language learners and, as such, should minimise generating output that could be disadvantageous to learners from a particular group or background, such as students with a particular first language. These issues are out of the scope of this initial experimental work.

The annotation was carried out by three authors of this paper, who received no compensation for the job. None of the authors were aware of which system produced which output.

10. Acknowledgement

This paper reports on research supported by Cambridge University Press & Assessment. Some of this work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council.

11. Bibliographical References

Naveed Afzal and Ruslan Mitkov. 2014. Automatic generation of multiple choice questions using dependency-based semantic relations. *Soft Comput.*, 18(7):1269–1281.

Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Distractor generation for multiple-choice questions with predictive prompting and large language models. *arXiv preprint arXiv:2307.16338*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

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*, pages 5835–5840, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4390–4400, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

Rui Correia, Jorge Baptista, Nuno J. Mamede, Isabel Trancoso, and Maxine Eskenazi. 2010. Automatic generation of cloze question distractors. In *Second Language Studies: Acquisition, Learning, Education and Technology*. SLaTE: the ISCA SIG on Speech and Language Technology in Edu.

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 Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Shu Jiang and John 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, Copenhagen, Denmark. Association for Computational Linguistics.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. *CoRR*, abs/2005.00700.

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

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, New Orleans, Louisiana. Association for Computational Linguistics.

Chen Liang, Xiao Yang, Drew Wham, Bart Pursel, Rebecca Passonneaur, and C. Lee Giles. 2017. Distractor generation with generative adversarial nets for automatically creating fill-in-the-blank questions. In *Proceedings of the Knowledge Capture Conference*, K-CAP 2017, New York, NY, USA. Association for Computing Machinery.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Mqag: Multiple-choice question answering and generation for assessing information consistency in summarization. *arXiv preprint arXiv:2301.12307*.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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.

Ricardo Rodriguez-Torrealba, Eva Garcia-Lopez, and Antonio Garcia-Cabot. 2022. End-to-end generation of multiple-choice questions using text-to-text transfer transformer models. *Expert Systems with Applications*, 208:118258.

Yuni Susanti, Ryu Iida, and Takenobu Tokunaga. 2015. Automatic generation of english vocabulary tests. In *Proceedings of the 7th International Conference on Computer Supported Education - Volume 1*, CSEDU 2015, page 77–87, Setubal, PRT. SCITEPRESS - Science and Technology Publications, Lda.

Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. 2018. Automatic distractor generation for multiple-choice english vocabulary questions. *Research and Practice in Technology Enhanced Learning*, 13.

Kristiyan Vachev, Momchil Hardalov, Georgi Karadzhov, Georgi Georgiev, Ivan Koychev, and Preslav Nakov. 2022. Leaf: Multiple-choice question generation.

Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 2018. Large-scale cloze test dataset created by teachers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2344–2356, Brussels, Belgium. Association for Computational Linguistics.

A. Annotation guidelines

While performing the task, human annotators were asked to stick to the following guidelines as much as possible.

- “Please select as many distractors as are acceptable together (i.e., they are good distractors as a set and no two distractors are too similar or referring to the same thing). If two distractors are too similar and you want to accept one of them, choose the first one in the list, in order to reduce unnecessary disagreement between annotators.”
- “For all other distractors please choose *Not Acceptable*. If you are uncertain about any of the distractors you can tag them with *Uncertain* (ideally try as much as you can to choose either *Acceptable* or *Not Acceptable*). Please make sure to tag all distractors in a page.”
- “Acceptable distractors should be related to what is stated in the passage and only the candidates who fully understand the passage should be able to detect that they are not the answer to the question.”
- “Distractors should not be completely off-topic in relation to the correct answer or the question.”
- “Acceptable distractors and correct answer should all have the same format follow the same sentence structure (e.g., if the key is *on the desks*, the distractor *they are under their beds* is not acceptable, while *under their beds* might be acceptable depending on the context).”
- “Distractors and the key do not need to all be of uniform length, but the key option should not stand out by being significantly shorter or longer, or of a more complex structure.”
- “Please note that you will see repeated articles with same questions. Generated distractors might be similar or different. Please annotate them independently.”

B. Questions used for human evaluation

Table 5 lists the questions that were used for human evaluation (without their passages or distractors). Questions were classified into *general* and *specific* questions as described in Section 5.3. It must be noted that while some general questions are repeated across different passages, they are reported only once in the table.

General questions
Which of the following is TRUE according to the passage ?
Which of the following is TRUE ?
Which of the following statements is TRUE ?
From the passage we can infer that
We can infer from the passage that
What can we infer from the passage ?
What might be the title of the passage ?
What is the best title of this passage ?
Which is the best title for the passage ?
What would be the best title for the passage ?
According to the passage , we can know that
What can we learn from the passage ?
What is mainly talked about in the text ?
What is the article about ?
The text is mainly about
Specific questions
In the report , who studies hardest ?
In China , how many students fall asleep in class ?
What do American students do in their free time ?
Why did n't Chief Joseph want to leave the land ?
After some of the young men in White Bird 's group killed eleven white persons, _
Morgan invented volleyball to
What did Morgan think of basketball ?
Specific volleyball rules were formed probably because
What is included in the volleyball rules ?
What did the group of old classmates get together for ?
What cups did the old professor give to his students ?
According to the old professor , why did they have so much stress ?
What can we learn from the old professor 's words ?
Many birds travel in large groups because
Rabbits spend the cold winter by
In winter , snakes
Some animals like squirrels
Doherty and his team of scientists did an experiment to evaluate
When asked to find the larger circle ,
Why are younger children not fooled ?
How did Profe treat his class and his students ?
What 's the job of West and Jernigan at school ?
They love the job because they can
Which of the following is true of the two men ?
The most significant revolution refers to
Using Orange Money , people can
Most people in the West do n't use mobile banking because
By saying " Gies tried to play down her own role " , the writer means Gies
According to the passage , the chemical accident that caused by the fault of management happened in
From the passage we know that " ammonium nitrate " is a kind of
From the discussion among some experts we may conclude that
What did the author 's grandmother always ask her to do during her summer vacation ?
How did the author first shop in the store ?
What can we infer about Miss Bee ?
The author mentioned her daughter to

Table 5: List of questions that are used for the human evaluation presented in Section 5.3, grouped into general and specific questions.