Better Distractions: Transformer-based Distractor Generation and Multiple Choice Question Filtering

Jeroen Offerijns
Graduation Thesis
Media Technology MSc program, Leiden University
July 2020
Thesis advisors: Tessa Verhoef and Suzan Verberne

Abstract

For the field of education, being able to generate semantically correct and educationally relevant multiple choice questions (MCQs) could have a large impact. While question generation itself is an active research topic, generating distractors (the incorrect multiple choice options) receives much less attention. A missed opportunity, since there is still a lot of room for improvement in this area. In this work, we train a GPT-2 language model to generate three distractors for each question, using the RACE dataset. Our model outperforms earlier work on distractor generation (DG) and achieves state-of-the-art performance. Next, we train a BERT language model to answer MCQs, and use this model as a filter, to select only questions that can be answered and therefore presumably make sense. This improves not only our own results, but can also be used to enhance other question generation models.

1 Introduction

Over the last two years, Transformer-based language models have gone from development to being adopted in all parts of natural language processing (NLP). This started with ULMFiT (Howard and Ruder, 2018) and BERT (Devlin et al., 2019), which showed the potential of pretraining a large neural network using unsupervised learning. After pre-training, these neural networks can be fine-tuned on specific tasks. During fine-tuning, the weights of the model are tweaked to perform well on a specific task, building upon the knowledge learned during pre-training. This has led to substantial improvements in the state of the art for tasks such as sentiment classification, question answering, and many others. When GPT-2 (Radford et al., 2019) was released, a huge improvement in text generation ability was obtained. The performance has even been shown to continue to improve with an increase in the size of the language models, ranging from 117M parameters for the smallest GPT-2 model, to 175B parameters for the largest of the GPT-3 (Brown et al., 2020) models.

Within natural language processing, question answering (QA) is a heavily researched field, while the inverse task receives much less attention: question generation (QG) (Pan et al., 2019). For education, being able to generate semantically correct and educationally relevant questions is a task with clear applications. Yet most of the work in this field focuses on using QG for generating synthetic datasets for question answering, rather than seeing it as an end on its own. For this reason, these papers tend to concentrate only on the task of generating a question from a given context and answer, while the other elements required for multiple choice questions (MCQs) receive much less attention. These elements include selecting the answer and generating the incorrect answers. It is this last part that we decided to work on: generating incorrect answers, also known as distractors.

For the distractor generation task, we use the RACE dataset (Lai et al., 2017), which contains almost $100,000$ questions. Each of these questions is paired with a context of a single paragraph, the correct answer, and three distractors. We use this to create a distractor generation model, which gives us the ability to generate complete multiple choice questions. This opens up other capabilities, including the ability to create a QA model which chooses the correct answer from four options. We will show that such a multiple choice QA model can be used to filter only correctly answered questions in order to improve the overall quality of question generation models.

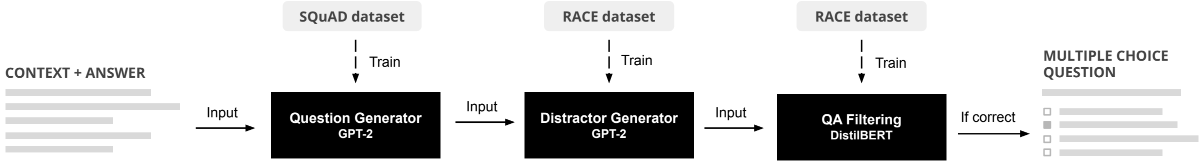

Figure 1: Overview of the model architecture: we take a context and answer as input, generate a question, generate three distractors, and use the QA model to filter only correctly answered questions. The distractor and question generator models are based on GPT-2, which is ideal for text generation, while the QA filtering model is based on BERT, which is better for classification problems.

The key contributions of this work include:

- We fine-tuned a GPT-2 language model for distractor generation on the RACE dataset.
- We fine-tuned a BERT language model for multiple choice question answering on the RACE dataset.
- We proposed a new QA filtering technique for improving QG results, by filtering using a multiple choice QA model.

2 Related work

Question generation Early question generation models were mainly rule-based: defining patterns of word types and using these to extract phrases from the text, which would be transformed into questions (Mitkov and Ha, 2003; Chen et al., 2006; Heilman, 2011). In the last decade, these rule-based models were mostly replaced by neural networks, primarily sequence-to-sequence architectures (Du et al., 2017; Kim et al., 2019). However, in the last year, these again are being replaced, now with Transformer-based language models.

The first of such works used BERT to generate questions (Alberti et al., 2019). By now, GPT-2 (Radford et al., 2019) has mostly replaced BERT for QG tasks (Klein and Nabi, 2019; Liu et al., 2020; Cho et al., 2019; Lopez et al., 2020). GPT-2 is a better text generator overall (Wang and Cho, 2019) due to it being trained solely in a left-to-right fashion, predicting the next word in a sequence of words. This is in contrast with bidirectional models such as BERT, which are trained primarily by predicting masked words. Such masked language modeling training leads to better performance on many NLP tasks, due to the bidirectional nature, but is worse at the specific task of text generation.

Distractor generation Several works for distractor generation (DG) are actually ranking models. These include Liang et al. (2018), which ranks distractors from a given candidate set using both feature-based and neural network-based ranking models, and Ren and Zhu (2020), who use a knowledge base to generate a distractor candidate set and a learning-to-rank model for selecting distractors.

In 2017, the RACE dataset (Lai et al., 2017) was published. This was the first dataset to include a large number of distractors along with the questions. Several papers since then have used this to create distractor generation models, including Gao et al. (2019), which used a hierarchical encoder-decoder model with attention to generate distractors. Zhou et al. (2020) improved upon this model by adding co-attention layers and using more tricks to gain better performance.

Multiple choice QA The original RACE paper used several models to establish baselines on the multiple choice QA task. Their Gated AR model achieved an accuracy of 44.1%, which showed the limitations of the models available at that time of publication (2017) for such a complex dataset. Recently, language models have been able to greatly surpass this accuracy, with BERT achieving an accuracy of 73.9% (Lan et al., 2019), and the largest variant of ALBERT (Lan et al., 2019) even achieving an accuracy of 82.3%.

QA filtering Alberti et al. (2019) introduced the concept of QA filtering to the domain of question generation. They generate a question, then answer that question using an extractive text QA model. Only when the QA model generates the correct answer, do they keep it. This is to ensure roundtrip consistency. Liu et al. (2020) also used a similar filtering method, but with the explicit goal of generating human-like questions.

3 Method

Our system consists of three separate models: a question generator, a distractor generator, and a QA filter. We will outline how we created and trained these models separately, and then we will explain how we used these jointly to improve the overall results. Figure 1 provides a high-level overview of our complete architecture.

3.1 Question generation

While question generation is not the goal of our research, we do use it as input for the other two models. It is used to evaluate the ability of the QA model to filter generated question—answer—distractor tuples. Similar to many recent works (Klein and Nabi, 2019; Liu et al., 2020; Lopez et al., 2020), we decided to fine-tune a GPT-2 model, in particular the "small" variant with 117 million parameters. For this task, we used the SQuAD dataset (Rajpurkar et al., 2016), specifically the training dataset of SQuAD v2. We remove questions which are highlighted as being impossible to answer (as specified by humans when the dataset was created), because we want our model to generate answerable questions. After removing these, $86,821$ questions remained.

We extract context—answer—question tuples from the SQuAD dataset, and tokenize these using the Byte-Pair-Encoding (BPE) tokenizer (Sennrich et al., 2016) that GPT-2 uses. Since GPT-2 is a model that learns to generate the next word after a sequence of words, we use special tokens to identify the segments of the inputs. This forces the model to learn to generate the correct elements. The input format is shown in Figure 2.

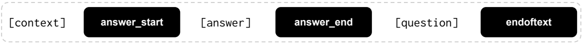

Figure 2: Format of the input to the QG model. The black boxes denote special tokens supplied to the tokenizer.

This model was implemented in PyTorch (Paszke et al., 2019) using the Transformers library (Wolf et al., 2019). The model was already pre-trained by OpenAI on a large text corpus, and we fine-tuned it on our dataset. It was fine-tuned for 3 epochs on the full dataset, using a batch size of 4. The Adam optimizer (Kingma and Ba, 2015) was used with a learning rate of 5×10^{-5} and an epsilon value of 1×10^{-8}. This optimizer improves upon classical stochastic gradient descent by using first and second moments of the gradients to speed up convergence. Using the Adam optimizer is standard practice for Transformer-based models. The learning rate and epsilon values are based on recommendations from Wolf et al. (2019).

3.2 Distractor generation

Similar to the question generation model, we again fine-tune GPT-2, but this time to generate distractors. Since the SQuAD dataset does not contain distractors, we used the RACE dataset (Lai et al., 2017) for this model. We do not do any filtering, so we use the full training dataset of $87,866$ questions. We provide the context, question, and answer as input. The context is where the model can draw stylistic influence from and which can be used for finding similar words and phrases to the correct answer. The question is what the distractors should be written in relation to. And finally, the answer should be used to make sure that the distractors are different from the answer. The input format is shown in Figure 3.

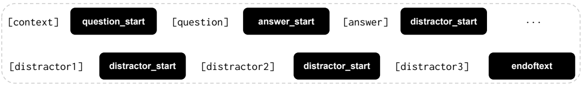

Figure 3: Format of the input to the DG model.

This is again tokenized using the BPE tokenizer, and we train the model with the same settings. However, besides training the small GPT-2 model, we also train another model based on the medium GPT-2 variant, with 355 million parameters. We keep the settings the same, except for the batch size which we reduce to 1, since we are limited by the memory usage. Ideally, we would have also trained the large or extra large GPT-2 variant, but this was not possible[1] on the RTX 2080 TI GPUs that we used.

During generation, we also apply a repetition penalty, as proposed by the authors of the CTRL language model (Keskar et al., 2019). This penalizes the model for generating similar texts, which helps to generate syntactically dissimilar distractors. Moreover, we noticed that the model could

[1] Although it is technically possible with the use of gradient checkpointing, we did not get this to work.

sometimes generate less than three distractors, generate non-unique distractors, or generate empty strings as distractors. To alleviate this, we decided to filter non-unique and empty distractors, and to repeat the generation step until three unique and non-empty distractors were found.

3.3 QA filtering

In order to be able to filter multiple choice questions, we need to have a model which can answer them. To create this, we decided to fine-tune the DistilBERT model (Sanh et al., 2019), with 66 million parameters. This is a distilled version of BERT, retaining 97% of the performance of the small BERT model, with 40% less parameters. Most QA research focuses on extractive QA: models where the output is a string, which is extracted from the source document. In our case, we want a QA model which chooses one of the multiple choice options as the correct answer. To accomplish this, we feed context—question—answer tuples into BERT. We then combine the four outputs and feed it through a dropout layer (Srivastava et al., 2014) for regularization, a fully connected layer for classification, and finally a softmax layer in order to model it a multi-class classification problem. The input format and the model architecture is shown in Figure 4.

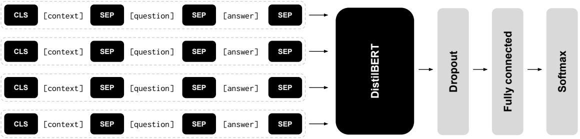

Figure 4: Overview of the input and architecture of the QA filtering model. We feed each distractor separately into the DistilBERT model, then use the four outputs to determine the answer.

This model was trained for 3 epochs, with a fully connected layer dimension of 768, a dropout ratio of 10%, a batch size of 2, and 8 gradient accumulation steps per batch [2]. Again, the Adam optimizer was used, with a learning rate of 3×10^{-5} and an epsilon value of 1×10^{-8}.

Once we have the multiple choice QA model, we can use it to filter question—answer—distractor tuples. The intuition behind this QA filter is that when a multiple choice QA model is given perfect information, it should almost always be able to answer a generated question correctly. If not, there could be two type of errors: (I) either the QA model does not have the capability to answer it, (II) or the question or distractors are somehow incorrect (i.e. this is a bad question). As for the type I errors, this should be unlikely because the model receives the exact context which is needed to answer the question. Imagine if you had a test and the students would be provided the paragraph which contained the answer for the question right next to every question: students would surely receive high grades. Moreover, QA models have already surpassed human performance on the SQuAD dataset (Zhang et al., 2020) and are nearing human performance on the RACE dataset (Lan et al., 2019), further decreasing the chance of type I errors. Type II errors are exactly what the QA model aims to filter. Therefore, whether the QA model can answer the question should be a good filter for high-quality questions.

4 Results

To evaluate our work, we chose several approaches: evaluating the text generation quality using standardized metrics, evaluating the ability for the QA model to answer the generated questions, and using a human evaluation to complement these two automatic metrics with a human perspective.

4.1 Quantitative evaluation

We compare our models against three baselines: the basic sequence-to-sequence distractor generator model from Gao et al. (2019), the improved hierarchical encoder-decoder model with static attention (HSA) from Gao et al. (2019), and the hierarchical model enhanced with co-attention (CHN) from Zhou et al. (2020).

Text generation quality As a high-level overview, we use several metrics to calculate the quality of the generated distractors. Specifically, we use the BLEU metric, which uses modified precision of n-grams to determine the correspondence to human-written text; and we use the ROUGE-L metric, which looks at the longest common subsequence and focuses on sentence level structure. The results of this evaluation can be found in Table 1. By default, we use questions from the dataset as input to the distractor generator. As a comparison, we also show the case where

[2] This simulates a larger batch size, which is required for good performance with a QA model on the RACE dataset (Liu et al., 2019).

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
Dataset questions					
SEQ2SEQ (Gao et al., 2019)	25.25	11.99	6.54	3.92	13.34
HSA (Gao et al., 2019)	26.93	13.57	8.00	5.21	14.45
CHN (Zhou et al., 2020)	27.53	13.80	8.46	5.80	**15.11**
GPT-2 SMALL	59.48	26.30	**13.68**	**9.28**	12.57
GPT-2 MEDIUM	**60.35**	**26.44**	13.37	8.92	12.33
GPT-2 MEDIUM (after QA filtering)	59.67	26.21	13.39	9.02	12.27
Generated questions					
GPT-2 SMALL	56.80	24.00	11.75	7.65	10.60
GPT-2 MEDIUM	57.20	23.80	11.37	7.30	10.10
GPT-2 MEDIUM (after QA filtering)	56.21	23.23	11.04	7.11	9.85

Table 1: Text generation quality of the distractor generation model. The DG scores are calculated separately for each distractor, and then averaged over all three distractors.

we are generating the questions as well, to show what the impact is on the results of the distractor generator.

The distractors in the RACE dataset are on average 5.4 words long, with a standard deviation of 3.3. This means that for evaluating distractors, the BLEU-1 and BLEU-2 scores are more relevant than BLEU-3 and BLEU-4, since 3-grams and 4-grams occur much less.

Question answering ability As a second quantitative evaluation, we decided to measure the number of questions answered correctly by the QA model, when the distractors are generated by our model. The better the distractors, the higher this percentage should be, as good distractors should be clearly incorrect answers to the QA model, given the fact that the model has full access to the context. However, as previously noted, the error rate of the QA model is a summation of two errors: errors due to bad distractors or questions, as well as errors made by the QA model itself due to other reasons. Therefore, the accuracy on its own is not meaningful to evaluate the distractors, but it is meaningful as a relative number to compare models.

For the results, see Table 2. We compare the GPT-2 SMALL and MEDIUM models. Again, we also compare the case for which we generate the questions with our question generator, with the case where we use the questions provided by the dataset and only generate the distractors.

4.2 Human evaluation

Metrics such as BLEU and ROUGE are based merely on comparing text similarity to reference sentences and are therefore limited in its ability to measure the quality of generated text as a human would (Callison-Burch et al., 2006). So, we decided to run a human evaluation to supplement the quantitative evaluation. Specifically, we wanted to test the ability of the QA filtering model to filter high quality questions which are answerable by a human. We set up a human evaluation with 4 assessors, each rating 100 generated questions with the following questions:

1. **Is the question well-formed and can you understand the meaning?** Possible answers include "Both understandable and well-formed", "Understandable, but not well-formed.", and "Neither".

2. **If the question is at least understandable, does the answer make sense in relation to the question?** This is a yes, no, or I don't know question.

These questions are based on work done by Liu et al. (2020), but we removed the relevancy question since it did not provide for a good indicator of quality in their results, and we rewrote the questions and answers to improve clarity. Of the 100 generated questions rated by each assessor, 30 questions were the same for each assessor, while the other 70 were unique questions. This enabled us to estimate inter-rater reliability, while still rating a large number of questions overall. Of these

	Dataset questions	Generated questions
GPT-2 SMALL	51.08%	54.55%
GPT-2 MEDIUM	**53.60%**	**56.49%**

Table 2: Accuracy of the QA model for generated distractors by both DG models.

Accepted questions

- What did the Wahhabism mean for the Muslims?
- What does the climate change report do?
- What do Wankel engines use?
- What river divides the city?
- What was the aim of the new law that the EU created?

Rejected questions

- Who was the composer for Destiny's Child?
- What was Zia-ul-Haq's primary ideology?
- Who was the Duke Yansheng Kong Duanyou's brother?
- What is the effect of inequality on human capital formation?
- What is a very short period with short epochs?

Figure 5: Randomly chosen examples of generated questions used for the human evaluation.

310 unique questions, 155 are questions that the QA filtering model accepted, while the other 155 are questions that the QA filtering model rejected. This should highlight the effect of the QA filtering model and show whether it is a good measure of the quality of questions. 10 example questions used as part of the evaluation are shown in Figure 5.

5 Discussion

Text generation quality Looking at the quantitative results in Table 1, the BLEU scores are substantially higher than previous works. This is in line with what other works have shown with the use of Transformer-based language models for text generation: these are much better at generating coherent text than previous sequence-to-sequence model based approaches were. However, interestingly, the ROUGE-L score is actually slightly lower than the ROUGE-L scores of previous works. While BLEU score is a measure of precision, ROUGE-L is a measure of recall. ROUGE measures how many words in the human references appear in the generated distractors. A potential explanation for this difference in scores is that the GPT-2 based models are more creative and therefore diverge further from the word distribution of the references than previous models do. This can lead to lower relative recall.

When looking at the differences between our own models, these seem to be relatively minor. The larger GPT-2 MEDIUM model, which has twice the number of parameters as the GPT-2 SMALL model, only gains less than a percentage point (when looking at BLEU-1 and BLEU-2). This minor change is likely due to the dataset size: the small model is already able to model the distribution well and can already learn to generate distractors like the outputs from the dataset. Furthermore, it appears that only rating distractors after the QA filtering step does not lead to better results. Lastly, the scores for when we generate questions are on average several percentage points lower than we use questions from the dataset. This makes sense: the question generator will occasionally generate incoherent questions, which will complicate the work of the distractor generator, and lead to outputs which differ more from the reference dataset.

Question answering ability As for the results shown in Table 2, we can clearly see that using GPT-2 MEDIUM for distractor generation, which has twice the number of parameters as GPT-2 SMALL, results in more accurate question answering than the smaller model. Moreover, as is to be expected, the scores for when the questions are also generated, are worse than when the questions are taken from the dataset.

Human evaluation The output of the human evaluation can be found in Table 3. The questions which the QA filtering model accepted are overall slightly better than those it rejected. 88% of accepted questions are either only understandable (18%) or are both well-formed and understandable (70%). This is 5% higher than 83% for rejected

		Accepted	Rejected
Question	Well-formed and understandable	70%	69%
	Only understandable	18%	14%
	Neither	12%	18%
Answer	Yes	50%	56%
	No	41%	37%
	I don't know	8%	7%

Table 3: Results from the human evaluation. We compare the quality of the questions which were accepted by the QA filtering model with those which were rejected.

questions. However, this is still a pretty small difference. To evaluate this, we applied Pearson's chi-squared test. This test showed that the likelihood of the observed data being drawn from the expected distribution is 21.40% for question 1 from the human evaluation and 40.18% for question 2. This means that the difference between the accepted and rejected questions was not shown to be statistically significant (for $p \leq 0.05$).

We also estimated the inter-rater reliability using the Fleiss' kappa measure (Fleiss, 1971). This led to a κ value of 0.413 for question 1 and a κ value of -0.147 for question 2. Using the interpretation table[3] from Landis and Koch (1977), the assessors would appear to be in moderate agreement for question 1, but in poor agreement for question 2. Since there is some subjectivity in how the generated questions are rated by the assessors, we would say that moderate agreement for question 1 is a positive result. The low score for question 2 can be explained by a combination of the question being even more subjective, as well as the fact that question 2 was likely not explained well in the evaluation setup. Therefore, we should rely primarily on the results of question 1.

Limitations & summary In order to make sense of the results, we need to be aware of the limitations of the different evaluation methods. As for the text generation quality measures such as BLEU and ROUGE, the main issue is that they do not consider the meaning of the text. There is some recent work in using language models for evaluating the text quality (Sellam et al., 2020), which should better incorporate meaning into the score, but we were not yet able to use this. Moreover, these metrics do not evaluate sentence structure as part of their calculation. As for the question answering ability, the main issue is that the model can accept bad questions or reject good questions. These types of errors are included in the total score. Ideally, we would need a QA model which always answers a good question correctly and always answers a bad question incorrectly. Although we think this is technically possible, our model is far from this level of performance. This means that the absolute values from Table 2 are irrelevant, but we can still look at the relative differences. As for the human evaluation, the main issue is the low number of total assessed questions, leading to statistically insignificant results.

These limitations have led us to use three different evaluation methods. Using the combined results to draw our conclusions, it looks like whether the question is answerable by the multiple choice QA model, is only a minor indicator of question quality. There was only a small difference in the quantitative results and no statistically significant difference in the qualitative results. One possible reason for this result is that the QA model will guess one of the four options if it does not know the answer for certain, leading to a high false positive rate. This could potentially be resolved by using bayesian neural networks to determine the QA uncertainy and set a threshold, ensuring that the model is sure about its prediction. Or a fifth ”I don't know” option could be added to the QA output and we could teach the model to choose this option when it is not certain.

6 Conclusion

Overall, we can conclude that distractor generation using GPT-2 works well, but filtering using a

[3] It should be noted that there is extensive debate about the validity of these ranges of interpretation, but it seems no better alternative exists.

multiple choice QA model does not help much to improve the results. However, we have multiple options to fix this and we still believe that it could be a worthwhile addition to the field of question generation. Besides being applied to our own question generator, we could apply our QA filtering model to improve the results of other question generation models. This will be explored in future work. Moreover, we could also attempt to use larger pretrained Transformer-based language models. It would be interesting to see how much of an improvement such larger pretrained models could bring.

As part of our research, we attempted to improve the distractor generation model by setting up an end-to-end training pipeline with the question answering model. Inspired by Klein and Nabi (2019), we wanted to generate distractors for a question, then feed this to the QA model, and backpropagate the loss of the QA model with regards to the weights of the DG model. This way, we wanted to teach the DG model to generate distractors such that the QA model could still correctly identify the correct answer, as the current DG model does not have enough inductive bias to generate distractors which are actually incorrect answers. Unfortunately, due to issues with catastrophic forgetting, this never materialized.

In summary, we have shown that generating multiple choice questions with distractors is technically possible using Transformer-based language models. This opens up many new possibilities and interesting applications. For example, it could be used to assist teachers in creating multiple choice exams. Or it could be used to automatically quiz students when they are learning. These developments are getting closer to reality and we aimed for this work to provide a valuable contribution towards this hopeful future.

Acknowledgements

This work was performed using resources provided by the Academic Leiden Interdisciplinary Cluster Environment (ALICE).

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Chia-Yin Chen, Hsien-Chin Liou, and Jason S. Chang. 2006. FAST – an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 1–4, Sydney, Australia. Association for Computational Linguistics.

Woon Sang Cho, Yizhe Zhang, Sudha Rao, Asli Celikyilmaz, Chenyan Xiong, Jianfeng Gao, Mengdi Wang, and Bill Dolan. 2019. Contrastive multi-document question generation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Association for Computational Linguistics (ACL)*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Yifan Gao, Lidong Bing, Piji Li, Irwin King, and Michael R. Lyu. 2019. Generating distractors for reading comprehension questions from real examinations. In *AAAI-19 AAAI Conference on Artificial Intelligence*.

Michael Heilman. 2011. *Automatic factual question generation from text*. Ph.D. thesis, Carnegie Mellon University.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6602–6609.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tassilo Klein and Moin Nabi. 2019. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

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.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. In *Proceedings of The Web Conference 2020*, WWW '20, page 2032–2043, New York, NY, USA. Association for Computing Machinery.

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.

Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Cheng. 2020. Transformer-based end-to-end question generation.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17–22.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019. Recent advances in neural question generation.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Siyu Ren and Kenny Q. Zhu. 2020. Knowledge-driven distractor generation for cloze-style multiple choice questions.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of* *the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2020. Retrospective reader for machine reading comprehension.

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