End-to-End Synthetic Data Generation for Domain Adaptation of Question Answering Systems

Siamak Shakeri[†][*], Cicero Nogueira dos Santos[*], Henry Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, Bing Xiang
AWS AI
New York City, NY
{cignog,henghui,patricng,nanfen}@amazon.com
{zhiguow,rnallapa,bxiang}@amazon.com
siamaks@google.com

Abstract

We propose an end-to-end approach for synthetic QA data generation. Our model comprises a single transformer-based encoder-decoder network that is trained end-to-end to generate both answers and questions. In a nutshell, we feed a passage to the encoder and ask the decoder to generate a question and an answer token-by-token. The likelihood produced in the generation process is used as a filtering score, which avoids the need for a separate filtering model. Our generator is trained by fine-tuning a pretrained LM using maximum likelihood estimation. The experimental results indicate significant improvements in the domain adaptation of QA models outperforming current state-of-the-art methods.

1 Introduction

Improving question answering (QA) systems through automatically generated synthetic data is a long standing research goal (Mitkov and Ha, 2003; Rus et al., 2010). Although many past works have proposed different strategies for question generation, they have limited or no success in improving the downstream QA task (Du et al., 2017; Sun et al., 2018; Song et al., 2018; Klein and Nabi, 2019; Wang et al., 2020; Ma et al., 2020; Chen et al., 2020; Tuan et al., 2019).

Some recent approaches for synthetic QA data generation based on large pretrained language models (LM) have started to demonstrate success in improving the downstream Reading Comprehension (RC) task with automatically generated data (Alberti et al., 2019; Puri et al., 2020). However, these approaches typically consist of multi-stage systems that use three modules: span/answer detector, question generator and question filtering. Given an input passage, the *span detector* is responsible for extracting spans that will serve as answers for which questions will be generated. This module normally combines a pretrained QA model with handcrafted heuristics. The *question generator* is a large LM fine-tuned for the task of conditional generation of questions given passage and answer. The *question filtering* comprises another RC model that is used to score and filter the generated QA pairs. Each module of this synthetic data generation pipeline is trained/tuned separately and errors from one stage can propagate to the next stages. Additionally, each module is expensive to be computed because all use large transformer networks (Vaswani et al., 2017).

In this work, we propose an end-to-end approach for synthetic QA data generation. Our model comprises a single transformer-based encoder-decoder network that is trained end-to-end to generate both the answer and the question. In a nutshell, we feed a passage to the encoder and ask the decoder to generate the question and the answer token-by-token. The likelihood produced in the generation process is used as a filtering score, which avoids the need of a separate filtering model. Our generator is trained by fine-tuning a pretrained LM using maximum likelihood estimation (MLE). We use BART (Lewis et al., 2019) as the pretrained LM in our experiments.

We perform experiments with three different variations of our synthetic QA data generator: (1) *AQGen*, which generates first the answer then the question; (2) *QAGen*, which generates first the question then the answer; (3) *QAGen Two-step (2S)*, which generates first the question, concatenates it to the passage, then generates the answer in a second pass through the same encoder-decoder.

We focus our empirical evaluation on the task of data augmentation for domain adaptation of reading comprehension (RC) models trained on

[*]equal contribution. † Siamak Shakeri is currently with Google. The work was done when he was at AWS AI.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5445–5460,
November 16–20, 2020. ©2020 Association for Computational Linguistics

SQuAD 1.1 dataset. We assess the effectiveness of our QA data generators for domain adaptation of four different target domain datasets: Natural Questions (NQ), BioASQ, NewsQA and DuoRC. We compare our results with recent work on domain adaptation for QA as well as with a three-stage synthetic data generator. QAGen performs better than AQGen and the baselines for all datasets, while QAGen2S provides the best results overall because it allows bidirectional attention between passage and question. For NQ dataset, QAGen2S improves the SQuAD baseline by more than 8 points in EM and more than 7 points in F1. For NewsQA and BioASQ the gains in EM are also above 4 points. Additionally, we also demonstrate that synthetically generated data by QAGen2S can improve the in-domain performance of both small and large RC models, leading to F1/EM improvements of 1/0.5 and 3.1/2.2 on `RoBERTa-large` and `bert-base-uncased` trained RC models on SQuAD dev.

The main contributions of this work can be summarized as follows: (1) we propose the first effective end-to-end approach for synthetic QA data generation; (2) our approach solves an important issue in previous methods for QA data generation: the detection of good spans. We show that span detection can be effectively solved as a generation task, just like question generation; (3) as it uses a single end-to-end model, our data generation pipeline is simpler, faster and more efficient; (4) we perform comprehensive experiments that demonstrate the effectiveness of our proposed approach for domain adaptation of QA systems.

2 End-to-End Model for Question and Answer Generation and Filtering

We model the problem of synthetic QA data generation as a conditional language modeling task. More specifically, we use an encoder-decoder (enc-dec) conditional LM as described in what follows.

2.1 Enc-Dec Conditional Language Models

Language modeling consists of learning the probability distribution $p(x)$ over variable-length token sequences $x = (x_1, x_2, ..., x_{|x|})$, where the tokens come from a fixed size vocabulary V. The training of LMs typically involves solving the task of predicting the next token based on past tokens. The distribution $p(x)$ can be represented by the conditional probability of the next token given the previous ones (Bengio et al., 2003):

$$p(x) = \prod_{i=1}^{|x|} p(x_i|x_{<i}) \tag{1}$$

In the case of conditional LMs, the generation is conditioned on an additional context c:

$$p(x|c) = \prod_{i=1}^{|x|} p(x_i|x_{<i}, c) \tag{2}$$

Transformer-based encoder-decoder conditional LMs (Lewis et al., 2019; Raffel et al., 2019) use bidirectional self-attention in the encoding step to create vector representations of the tokens in the context c. The decoding step generates the tokens of the sequence x in an auto-regressive manner, while performing self-attention on previously generated tokens of x and all the representations output by the encoder for c.

2.2 Question-Answer Generation

In the case of end-to-end synthetic data generation for QA, we need to model the joint conditional distribution $p(a, q|c)$, where the input context c is a passage, q is a question and a is the correct answer, which is a span in c. Our approach to model $p(a, q|c)$ involves fine-tuning a pretrained Enc-Dec conditional LM using a training set $D = \{(c^1, q^1, a^1), (c^2, q^2, a^2), ..., (c^{|D|}, q^{|D|}, a^{|D|})\}$. We train the Enc-Dec with parameters θ through maximum likelihood estimation (MLE) by minimizing the negative log-likelihood over D:

$$\mathcal{L}(D) = -\sum_{i=1}^{|D|} \log p_\theta(a^i, q^i|c^i) \tag{3}$$

We can have different variations of the generator depending on how we place the items in the output sequence: answer-question or question-answer. This difference in the ordering is crucial because it defines which part is conditioned on the other. Based on this observation, we propose three variations of our generative model:

AQGen: this model generates answer and question jointly given the input context: $(q, a) \sim p(a, q|c)$. During sampling, the answer tokens are generated, which are followed by question tokens. This makes the generation of the question conditioned on both input context (through attention on the encoder) and answer (through self-attention in the decoder). Fig. 1 depicts this model.

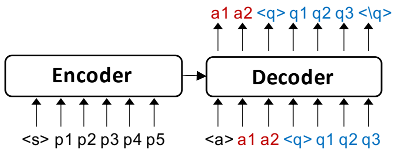

Figure 1: AQGen Model: given an input passage the model generates an answer followed by a question.

QAGen: this model generates question and answer jointly given the input passage: $(q, a) \sim p(a, q|c)$. During sampling, the question tokens are generated, which are followed by answer tokens. This makes the generation of the answer conditioned on both input context (through attention on the encoder) and question (through self-attention in the decoder). Fig. 2 depicts this model.

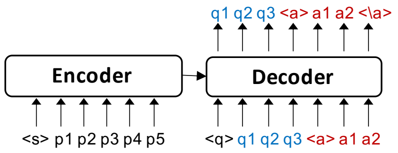

Figure 2: QAGen Model: given an input passage the model generates a question followed by an answer.

QAGen Two-Step (2S): this model performs question generation and answer generation in two separate passes over the Enc-Dec LM. First, the question is generated given the input context $q \sim p(q|c)$, (Step 1). Next, the question is concatenated with the input context and the resulting sequence is given as input to the Enc-Dec, which finally generates the answer $a \sim p(a|q, c)$, (Step 2). QAGen 2S sampling approach is illustrates in Fig. 3. This model uses a single Enc-Dec LM that is trained with samples of both $p(q|c)$ and $p(a|q, c)$. We use control codes $<q>$ and $<a>$ to inform the decoder whether to generate a question or an answer, respectively.

2.3 Decoding

A natural choice for decoding with conditional neural LMs is beam search. However, our preliminary experiments with beam search showed a lack of diversity and a high repetition of generated question-answer pairs. Generating diverse question-answer pairs is crucial to the performance of downstream RC models. Particularly, diversity of answer spans ensures that various parts of the passage are used, and different question types are generated. We use a variant of nucleus sampling (Holtzman et al., 2019), where we pick top k tokens, and within top k, we pick tokens that comprise top 95% probability mass. We set k to 20 in our experiments. We refer to this setting as *Topk+Nucleus*. This decoding was used in QAGen, AQGen, and question sampling step in QAGen2S. The answer generation of QAGen2S was performed by greedy decoding. We discard generated (q, a) pairs whose answers do not occur in the input passage, as non-extractive QA is outside the scope of this work. We observed between 10% to 15% of samples being dropped because of this issue.

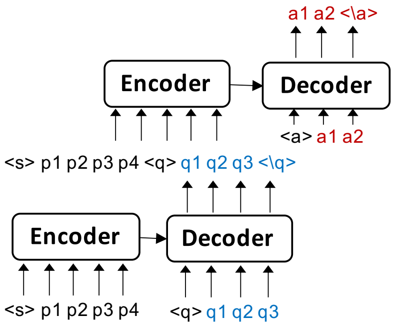

Figure 3: QAGen Two-Step: given an input passage the model first generates a question (Step 1). Next, the question is concatenated with the passage and both are given to the encoder-decoder that generates the answer (Step 2).

2.4 Filtering

Recent work have used the *round-trip filtering* method (Alberti et al., 2019; Puri et al., 2020) to prune the synthetic QA set and improve data quality. This method consists of two steps: (1) using an RC model to provide answers to the generated questions; (2) dropping the QA pairs for which the answer of the RC model does not match the span detected answer. While round-trip filtering has shown to be effective, it is not the most efficient approach because it involves the application of an RC system over the whole set of generated data. Additionally, there might exist cases that are difficult for the filtering model, but in fact are of high quality.

We propose using the likelihood of the generated question-answers as a measure to perform filtering and address the efficiency issue, as it avoids the use of an RC model for filtering. We argue that such a likelihood score, albeit noisy, is an indicator of whether a generated question-answer is high quality for training a downstream RC model. We refer to this approach as *LM filtering*. Essentially, given an input passage, we sample n different QA pairs, rank them according to decreasing order of

PubMed

Lymph node status has major prognostic importance in colorectal cancer and greater precision in the diagnosis of lymph node metastases should provide better prognostic and therapeutic guidance. **Keratin 20** (K20) gene expression has been used as a marker of lymph node metastases, but the evidence for this remains circumstantial. This study has therefore sought to determine K20 specificity and to correlate K20 expression with mutant K-RAS expression, in order to provide direct evidence that K20 expression in lymph nodes of colorectal cancer patients genuinely reflects metastatic disease. Specificity of K20 expression was established against a range of tissue types and 289 lymph nodes from 41 non-cancer control patients. K20 expression was restricted to **gastrointestinal epithelia** and was only present in one of the 289 control lymph nodes, giving a calculated specificity of 97.6 % (95% confidence limits: **87.1-99.9**%)...

Q: *What is K20 expression found to be restricted to?*	A: *gastrointestinal epithelia*
Q: *What was the 95% confidence range of the mutation analysis?*	A: *87.1-99.9%*
Q: *What is the name of the gene that can be used as a marker of metastatic disease?*	A: *Keratin 20*

CNNDM

By. Emily Allen. PUBLISHED:. 06:27 EST, 12 June 2012. |. UPDATED:. 09:35 EST, 12 June 2012. Teachers have apologised to parents after a group of **primary school** children were forced to stay in the canteen until they had finished all the food on their plates. Parents of children attending Kaizen Primary School in Plaistow, East London, were left fuming after a group of pupils, some as young as **five**, were told they had to clear their plates before being allowed out into the playground. Even though years ago parents would not have batted an eyelid and would have welcomed schools encouraging their children to eat, dozens of parents complained, saying that children should 'not be forced to eat' by teachers. Upset: Parents of children at Kaizen Primary School in Plaistow, East London, said pupils were told they had to clear their plates (file picture) Candeece Kenlock said her five-year-old son **Kehyan** was 'so scared' of being forced to eat everything on his plate he didn't want to go to school anymore....

Q: *what is the name of a five year old boy whose parents said he was 'so scared' he didn't want to go to school?*	A: *Kehyan*
Q: *What type of school were children forced to stay in the canteen to finish their meals?*	A: *primary school*
Q: *How old were the children who were forced to stay in the canteen until they had finished their food?*	A: *five*

IMDB

Clark Russell, a prominent writer, concludes that he will visit the south in the capacity of a farm hand and thus secure atmosphere for a new story. He learns that laborers are needed on a certain farm and as he journeys into the country he rescues a young woman whose horse is running away. When Clark applies for work he is treated lightly by **Bud**, the foreman, until the owner of the farm arrives with his daughter, Anna, who recognizes her hero of the afternoon. A few days later at the dinner table Clark defends Polly, a maid, when she is annoyed by Bud and after the hands departed for the fields the two men settle their score in a fight, the bully receiving a severe lesson. Polly overhears Bud declaring that he will be revenged but she is unable to warn Clark. Later in the day **the bully** tries to force Clark into the hopper of the threshing machine but Anna sees the struggle from a distance and stops the engine...

Q: *What is the name of the foreman at the farm?*	A: *Bud*
Q: *Who saves Anna?*	A: *Clark Russell*
Q: *Who tries to force Clark into a hopper of the threshing machine?*	A: *the bully*

Natural Questions

<Table> <Tr> <Th colspan="2"> Tampa Bay Lightning </Th> </Tr> <Tr> <Td colspan="2"> 2018 – 19 Tampa Bay Lightning season </Td> </Tr> <Tr> <Td colspan="2"> </Td> </Tr> <Tr> <Th> Conference </Th> <Td> Eastern </Td> </Tr> <Tr> <Th> Division </Th> <Td> Atlantic </Td> </Tr> <Tr> <Th> Founded </Th> <Td> **1992** </Td> </Tr> <Tr> <Th> History </Th> <Td> Tampa Bay Lightning 1992 – present </Td> </Tr> <Tr> <Th> Home arena </Th> <Td> Amalie Arena </Td> </Tr> <Tr> <Th> City </Th> <Td> Tampa , Florida </Td> </Tr> <Tr> <Td colspan="2"> </Td> </Tr> <Tr> <Th> Colors </Th> <Td> Tampa Bay blue , white </Td> </Tr> <Tr> <Th> Media </Th> <Td> Fox Sports Sun 970 AM </Td> </Tr> <Tr> <Th> Owner (s) </Th> <Td> Tampa Bay Sports and Entertainment (Jeffrey Vinik , chairman) </Td> </Tr> <Tr> <Th> General manager </Th> <Td> **Steve Yzerman** </Td> </Tr> <Tr> <Th> Head coach </Th> <Td> **Jon Cooper** </Td> </Tr> <Tr> <Th> Captain </Th> <Td> Steven Stamkos </Td> </Tr> <Tr> <Th> Minor league affiliates </Th> <Td> Syracuse Crunch (AHL) Orlando Solar Bears (ECHL) </Td> </Tr> <Tr> <Th> Stanley Cups </Th> <Td> 1 (2003 – 04) </Td> </Tr> <Tr> <Th> Conference championships </Th> <Td> 2 (2003 – 04 , 2014 – 15) </Td> </Tr> <Tr> <Th> Presidents ' Trophy </Th> <Td> 0 </Td> </Tr> <Tr> <Th> Division championships </Th> <Td> 3 (2002 – 03 , 2003 – 04 , 2017 – 18) </Td> </Tr> <Tr> <Th> Official website </Th> <Td> www.nhl.com/lightning </Td> </Tr> </Table>

Q: *What year was the Tampa Bay Lightning established??*	A: *1992*
Q: *Who is the head coach of the Tampa Bay Lightning?*	A: *Jon Cooper*
Q: *Who is the Tampa Bay Lightning general manager?*	A: *Steve Yzerman*

Table 1: Samples of generated question-answer pairs using QAGen2S model for four target domains. The generated answers are shown in **bold**. The paragraphs are truncated from their original sizes due to space limitations.

LM score and pick the top m samples. This is similar to the *sample-and-rerank* approach suggested by Holtzman et al. (2019) and Adiwardana et al. (2020). Formally, for QAGen and QAGen2S, we use the score:

$$\text{LM score} = \sum_{i=1}^{N_a} \log p(a^i|c, q)$$

And for AQGen :

$$\text{LM score} = \sum_{i=1}^{N_a} \log p(a^i|c) + \sum_{i=1}^{N_q} \log p(q^i|c, a)$$

Where N_q and N_a indicate the lengths of generated question and answer, respectively. We use answer-only scores for QAGen and QAGen2S because question quality would have a dominant effect on LM scores since questions are usually longer than answers. Additionally, using answer-only scores when conditioned on the generated question is more suitable for the RC tasks because it better mimics the score of a downstream RC model, which is answer centric. With AQGen, we use both answer and question LM scores, as answer generation is not conditioned on the question. We use likelihood summation instead of averaging because experiments showed that the former works slightly better. Further details included in Appendix B.3. We speculate this is due to average pooling encouraging longer question-answers, which could be of lower quality than shorter question-answer pairs.

3 Related Work

Question generation (QG) has been extensively studied from the early heuristic-based methods (Mitkov and Ha, 2003; Rus et al., 2010) to the recent neural-base approaches. However, most work (Du et al., 2017; Sun et al., 2018; Zhao et al., 2018; Kumar et al., 2019; Wang et al., 2020; Ma et al., 2020; Tuan et al., 2019; Chen et al., 2020) only takes QG as a stand-alone task, and evaluates the quality of generated questions with either automatic metrics such as BLEU, or human evaluation. Tang et al. (2017), Duan et al. (2017) and Sachan and Xing (2018) verified that generated questions can improve the downstream answer sentence selection tasks. Song et al. (2018) and Klein and Nabi (2019) leveraged QG to augment the training set for machine reading comprehend tasks. However, they only got improvement when only a small amount of human labeled data is available. Recently, with the help of large pre-trained language models, Alberti et al. (2019) and Puri et al. (2020) have been able to improve the performance of RC models using generated questions. However, they need two extra BERT models to identify high-quality answer spans, and filter out low-quality question-answer pairs. Lee et al. (2020) follow a similar approach while using InfoMax Hierarchical Conditional VAEs. Nishida et al. (2019) showed improvements by fine-tuning the language model on the target domains.

4 Experimental Setup and Results

4.1 Datasets

We used **SQuAD 1.1** dataset (Rajpurkar et al., 2016) to train the generative models as well as in-domain supervised data for the downstream RC task in this work. We used the default train and dev splits, which contain 87,599 and 10,570 (q, a) pairs, respectively.

Similar to (Nishida et al., 2019), we selected the following four datasets as target domains:
Natural Questions (Kwiatkowski et al., 2019), which consist of Google search questions and the annotated answers from Wikipedia. We used MRQA Shared Task (Fisch et al., 2019) preprocessed training and dev sets, which consist of 104,071 and 12,836 (q, a) pairs, respectively. The training set passages were used as the unlabeled target domain corpus, while the evaluations were performed on the dev set.
NewsQA (Hermann et al., 2015), which consists of question and answer pairs from CNN news articles. We used the dev set from the MRQA Shared Task, which removes unanswerable questions and those without annotator agreement. We prefer this version as we focus only on the generation of answerable questions. The dev set consists of 4,212 (q, a) pairs. Passages from CNN/Daily Mail corpus of Hermann et al. (2015) are used as unlabeled target domain corpus.
BioASQ (Tsatsaronis et al., 2015): we employed MRQA shared task version of BioASQ, which consists of a dev set with 1,504 samples. We collected PubMed abstracts to use as target domain unlabeled passages.
DuoRC (Saha et al., 2018) contains question-answer pairs from movie plots which are extracted from both Wikipedia and IMDB. ParaphraseRC task of DuoRC dataset was used in our evaluations, consisting of 13,111 pairs. We crawled IMDB movie plots to use as the unlabeled target domain corpus.

4.2 Experimental Setup

We used Pytorch (Paszke et al., 2019) and Transformers (Wolf et al., 2019) to develop the models and perform experiments. Generative models are trained on SQuAD 1.1 for 5 epochs, and the best model is selected based on the cross entropy loss on the SQuAD dev set. AdamW (Loshchilov and Hutter, 2017) optimizer with learning rate of 3×10^{-5} is employed.

For RC model training, we use `bert-base-uncased` model (Devlin et al., 2018). AdamW optimizer is used with learning rate of 3×10^{-5} and batch size 24 for 2 epochs without linear warmup. We set maximum sequence length 384 with document stride 128. SQuAD 1.1 dev set is used to select the best model during training. As a baseline for QA data generation, we implemented a three-stage pipeline similar to the state-of-the-art approach of Puri et al. (2020). We call this baseline ***QGen***, which generates a question given a passage and extracted span, $q \sim p(q|a, c)$. The span detection module consists of `bert-base-uncased` fine-tuned on SQuAD 1.1 passage and spans, where the start and end classification heads are trained to perform span detection. For QGen, we experimented with sampling top 5 spans and generating two questions per each, as suggested by (Puri et al., 2020), as well as sampling top 10 spans and generating one question per each. Our results showed the latter outperforming the former. Henceforth, we used this configuration in our evaluations.

We trained QGen models on both BART-Large and GPT2-Medium (Radford et al., 2019), which have an equivalent number of parameters, 406M (BART) vs 350M (GPT2), and evaluated BLEU score of the generated question w.r.t. the ground truth question on the SQuAD dev set. BART and GPT2 achieved 21.29 and 18.31 BLEU, respectively. We believe the bi-directional encoding in BART is superior to uni-directional encoding in GPT2. Hence, we used BART for the rest of the experiments.

4.3 Synthetic Data Generation

For each of the unlabeled target domain corpora, we randomly selected 100,000 passages to perform synthetic data generation. Passages shorter than 100 tokens were discarded. Selected ones were truncated to maximum length of 550 tokens. We removed the passages that existed in the dev sets.

Question-answer generation with AQGen, QAGen, and QAGen2S is performed using Topk+Nucleus, as discussed in Sec. 2.3. For each passage, 10 samples are generated. Unless otherwise mentioned, LM filtering is applied by sorting the 10 samples of each passage according to LM scores as detailed in Sec. 2.4, and the top 5 samples are selected. The number of synthetically generated pairs is between 860k to 890k without filtering and 480k to 500k after LM filtering. Tab. 1 shows generated question-answer pairs from four target domain (see Appendix for more examples). We can observe that the generative model is able to generate question answer pairs even from raw HTML input that corresponds to a table. The rendered table can be seen in Tab. 12 (Appendix C.3). Considering the fact that the training data of the generative model does not include any HTML input, this further demonstrates the robustness and efficacy of our proposed approach.

4.4 Domain Adaptation Results

Tab. 2 shows the results of domain adaptation experiments. Each experiment was performed by training the RC model on the synthetic data generated on the target domain corpus. We refer to the dataset to which the downstream model is being adapted as the target domain. Source domain indicates the supervised training dataset (SQuAD).

We also performed experiments by using both Synthetic + SQuAD1.1 data. Our QAGen and QAGen2S models outperform by wide margins the baseline models trained on SQuAD 1.1 only, as well as unsupervised domain adaptation approaches (UDA) suggested by Nishida et al. (2019) and Lee et al. (2020). Additionally, QAGen and QAGen2S significantly outperforms QGen, our implementation of the three-stage pipeline of Puri et al. (2020).

Even though our SQuAD 1.1 baselines are generally higher than both Nishida et al. (2019) and Lee et al. (2020), our best model achieves more point-wise improvements in all of the target domain datasets, except with BioASQ, where Nishida et al. (2019) observe 4.3 points in EM versus 4 points with ours, and 4.2 points in F1 versus 2.2 with ours.

Comparing LM and round-trip filtering when applied to the best performing model, QAGen2S, we can observe that the LM filtering approach (Sec. 2.4) is more effective than round-trip filtering in BioASQ and DuoRC target domains. It barely underperforms ($\sim$ 1 point) in F1 and EM in the other two domains. This demonstrates the efficacy of the suggested filtering approach, which also simplifies the question-answer generation pipeline.

The highest (EM/F1) domain adaptation gains seen with BioASQ (4/2.2) and DuoRC (1.2/1.1) are smaller than those with Natural Questions (8.5/7.5) and NewsQA (5.5/4.5). We postulate this is due to two reasons: Firstly, both BioASQ and DuoRC domains are more dissimilar to the source domain, SQuAD, compared to NewsQA and Natural Questions; Secondly, BioASQ and DuoRC are more difficult datasets. Comparing our results with supervised target domain training of DuoRC, we observe that with using only synthetic data outperforms the DuoRC training set, which consists of 39144 pairs. While our domain adaptation methods show substantial gains with NewsQA and Natural Questions domain, there is still room for improvements to match the performance of supervised target domain training (last row in Tab. 2).

While results in Tab. 2 suggest that generating synthetic QA data from target domain text leads to significant gains on the target domain dev set, one can argue whether it is essential to generate synthetic data from the corpus matching the target dev set's domain to achieve good performance. Hence, we performed cross-domain experiments to check this argument. Tab. 3 shows the performance

Model	fine-tune Data	NQ		NewsQA		BioASQ		DuoRC	
		EM	F1	EM	F1	EM	F1	EM	F1
SQuAD 1.1 Nishida et al. (2019)	SQuAD	44.4	57.5	35.2	50.7	41.1	53.6	24.5	33.0
UDA Nishida et al. (2019)	SQuAD	43.8	56.7	35.9	51.4	45.4	57.8	25.5	34.1
SQuAD 1.1 Lee et al. (2020)	SQuAD	42.77	57.29	–	–	–	–	–	–
UDA Lee et al. (2020)	SQuAD+Synthetic	48.44	62.69	–	–	–	–	–	–
Our SQuAD 1.1	SQuAD	44.66	58.94	39.51	56.36	44.35	56.06	28.85	34.92
QGen + round-trip filtering	Synthetic	48.04	61.28	39.03	54.37	35.31	46.80	28.74	34.10
	+ SQuAD	49.02	62.61	40.79	56.79	39.43	50.42	29.39	34.80
AQGen (ours) + LM filtering	Synthetic	47.80	61.29	38.55	55.42	39.49	52.11	27.09	33.47
	+ SQuAD	49.04	62.56	39.62	56.88	42.89	54.90	27.88	34.40
QAGen (ours) + LM filtering	Synthetic	49.81	63.36	43.09	57.9	42.49	51.95	29.46	35.25
	+ SQuAD	50.01	63.10	44.06	59.20	45.74	55.06	29.91	35.82
QAGen2S (ours) + LM filtering	Synthetic	52.64	65.56	43.99	59.95	46.74	57.76	29.91	35.81
	+ SQuAD	52.03	65.70	43.57	59.8	**48.40**	**58.33**	**30.06**	**36.05**
QAGen2S (ours) + round-trip	Synthetic	**53.11**	**66.45**	**45.04**	60.79	45.01	57.01	29.47	35.32
	+ SQuAD	51.91	65.62	44.78	**60.92**	46.14	57.96	30.01	35.83
Supervised target domain	Target	66.50	78.55	51.09	66.67	–	–	27.35	33.28

Table 2: Domain adaptation results for different methods. **Bold** cells indicate the best performing model on each of the target domain dev sets, excluding supervised target domain training results.

Target Domain Corpus	fine-tune Data	NQ		NewsQA		BioASQ		DuoRC		SQuAD	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
SQuAD 1.1	SQuAD	44.66	58.94	39.51	56.36	44.35	56.06	28.85	34.92	80.78	88.20
Natural Questions	Synthetic	*52.64*	65.56	40.48	55.40	42.69	52.56	27.88	33.39	79.95	86.89
	+ SQuAD	52.03	*65.70*	40.55	56.37	44.15	55.87	30.04	*36.14*	83.05	89.91
CNN/DM	Synthetic	47.05	60.27	*43.99*	*59.95*	45.28	55.25	27.02	33.22	76.81	84.62
	+ SQuAD	45.92	60.24	43.56	59.8	44.88	57.06	27.62	34	82.29	89.32
PubMed	Synthetic	44.48	57.98	39.27	54.88	46.74	57.76	26.21	32.03	78.65	85.82
	+ SQuAD	48.08	61.73	41.74	58.30	***48.40***	*58.33*	*30.23*	36.13	82.95	89.74
IMDB	Synthetic	48.82	61.77	43.09	58.90	45.28	55.59	29.91	35.81	79.86	86.79
	+ SQuAD	49.56	63.10	43.40	59.37	46.68	57.27	30.06	36.05	83.33	89.92
All 4 data sources	Synthetic	53.28	66.32	43.64	60.43	47.41	57.88	29.91	36.37	82.71	89.06
	+ SQuAD	**53.30**	**66.73**	**44.23**	**60.79**	47.01	**58.35**	**30.36**	**36.50**	**84.57**	**90.90**

Table 3: Cross domain experiments using QAGen2S as the generative model. Underlined cells indicate best EM/F1 value for each of the target domain dev sets (column-wise) and individual target domain corpus.

on every target domain dev set of RC models fine-tuned on synthetic data of different target domain corpora. We can see that diagonal elements, which have same domain of dev set and target corpus, show either the best performance (underlined results) or are within a narrow margin of top EM/F1 scores. Therefore, the most effective strategy is achieved when the passages used in the generation of synthetic samples are from the same domain as the target, which is expected in a domain adaptation method. Additionally, we trained an RC model with the synthetic data from all the four domains (last two rows in Tab. 3). This produced our best F1 results for all datasets, indicating that mixing synthetic data from different domains is beneficial for the QA task. Tab. 3 also shows EM/F1 scores of the cross-domain RC models on SQuAD 1.1 dev set. We can see that using synthetic data from any of the four domains significantly improved the performance for SQuAD. In particular, when training the RC model with data from all domains + SQuAD training data (last row), there is a large gain in both EM (3.8) and F1 (2.7).

4.5 Comparison of AQGen, QAGen and QAGen2S models

Comparing our proposed LM filtering-based models in Tab. 2, we propose the following explanations: (1) QAGen2S and QAGen outperform AQGen because generating answers conditioned on the question results in better spans, which is crucial in the training of the downstream RC task. Generated answer spans not conditioned on questions could include spurious tokens, or be a partial span. (2) QAGen2S outperforms QAGen because including the generated question in the bidirectional encoder allows cross attention between the passage and generated question, which results in even more accurate answer generation. Comparing the performance when only synthetic question-answer pairs are employed versus adding SQuAD training pairs, we can observe that the addition of labeled data results in marginal gains. This becomes even more evident for the best performing data generators. In fact, in some cases, adding SQuAD data degrades EM, such as QAGen2S + LM filtering with Natural

Model	Beam Search		Topk+Nucleus		Topk+Nucleus		Topk+Nucleus	
	N=5		N=5		N=10		N=20	
	EM	F1	EM	F1	EM	F1	EM	F1
Synthetic	49.73	63.19	52.20	66.19	52.64	65.56	51.08	63.50
Synthetic + SQuAD	49.95	64.08	49.68	64.47	52.03	65.70	51.87	64.82

Table 4: Beam search vs. Topk+Nucleus sampling with various sample sizes per passage. NQ is used as target domain and QAGen2S with LM filtering is used as generator. For N > 5, top 5 samples per passage were selected according to LM scores.

Questions and NewsQA.

4.6 Ablation Studies

Sampling Design Choices

Tab. 4 shows a comparison between beam search and Topk+Nucleus sampling with different number of samples (5, 10, and 20). The results indicate that beam search underperforms Topk+Nucleus. We attribute this to the lack of diversity in the generated samples using beam search. We observed that beam search tends to select fewer distinct spans, compared to Topk+Nucleus, and generates minor variations of the same question. Appendix C.1 examines this issue.

When training the RC model we only used the top 5 samples based on LM score per each passage. We can observe that sampling 10 pairs per document leads to the best EM/F1 on the target domain. By sampling many QA pairs per passage, we increase the chance of generating good samples. However, if we sample too many qa pairs the top ranked ones might be too similar. Therefore, we used sample size of 10 in this work since a higher sample size incurs higher computation cost while not showing improvements.

LM Filtering

We argue that using LM filtering, as discussed in section 2.4, results in improvements in the target domain downstream RC models by enhancing the quality of the generated (q, a) pairs. Results in Tab. 5 indicate that in the majority of the experiments using LM filtering leads to improved F1/EM metrics. AQGen benefits the most from LM filtering as it generates data with lower quality than the other two models. Tables 10 and 12 in the Appendix show examples of QA pairs and their LM scores.

Fig. 4 shows experimental results when varying the number of (q, a) pairs selected from the 10 pairs sampled per each passage. We chose the value of 5 as this configuration outperforms other values overall. A high value is more likely to allow undesired pairs, while a low value might discard plenty of high quality samples.

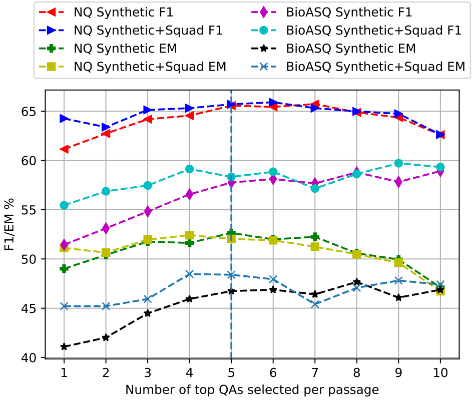

Figure 4: Effect of number of QAs selected per passage in LM filtering. QAGen2S model is used for generation. The likelihood score of the generated answer is used to sort the generated question answer pairs decreasingly.

Model	FT Data	NQ		BioASQ	
		EM	F1	EM	F1
AQGen w/o filter.	Synth.	46.93	60.71	41.49	53.59
	+ SQ	46.84	61.00	41.36	53.84
AQGen + LM filter.	Synth.	47.80	61.29	39.49	52.11
	+ SQ	**49.04**	**62.56**	**42.89**	**54.90**
QAGen w/o filter.	Synth.	50.67	64.04	43.15	53.20
	+ SQ	**51.35**	**64.99**	45.21	54.94
QAGen + LM filter.	Synth.	49.81	63.36	42.49	51.95
	+ SQ	50.01	63.10	**45.74**	**55.06**
QAGen2S w/o filter.	Synth.	47.12	62.61	46.88	58.92
	+ SQ	46.73	62.63	47.41	**59.33**
QAGen2S + LM filter.	Synth.	**52.64**	65.56	**48.40**	58.33
	+ SQ	52.03	**65.70**	46.74	57.76

Table 5: Comparison of using LM filtering versus no filtering. **Bold** values indicate best performance on each target domain for each model (per rows separated by sold lines).

Correlation between LM and F1 Scores

In this work, we proposed using the LM score of the generated samples as a surrogate to round-trip filtering. We postulate that the LM score correlates with the F1 score used in round-trip filtering. To more thoroughly examine this, we devised an experiment where we sorted the generated samples by their answer LM scores, divided them into contiguous buckets each with 200 samples, and calculated the average F1 score of the samples in each bucket. Fig. 5 shows the results of this experiment. As we can see, there exists a strong correlation between the two scores.

While the correlation looks promising, a challenge with using the LM score is that it is relatively noisy. For example, to use the LM score to get only samples whose F1 scores are 1, a very high threshold needs to be set, forcing the vast majority of samples to be dropped. Future work can explore how to reduce this noise.

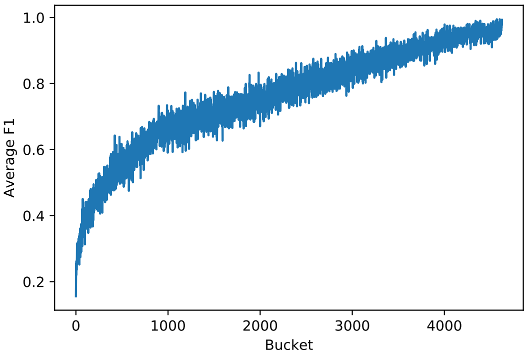

Figure 5: Average F1 score of sorted items based on LM scores. Samples were generated using QAGen2S on Natural Questions passages.

Impact of Synthetic Dataset Size

In Fig. 6, we present plots that correlate synthetic dataset size (in # of passages) and RC model performance (EM/F1). We can see that with increasing the number of generated (q, a) pairs (5 pairs per passage), RC model performance improves. Such correlation is more evident when not using the SQuAD training data. This is expected as with added supervised training samples, there would be less need for a large number of synthetic samples.

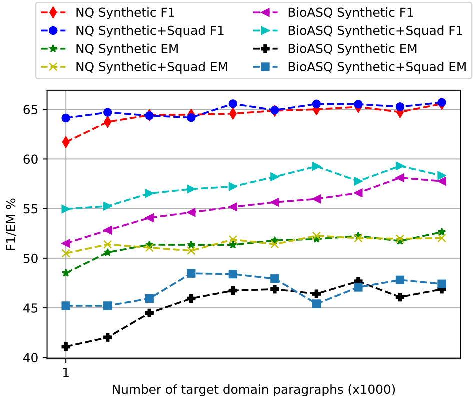

Figure 6: The effect of number of target domain passages on the RC task with synthetically generated QA pairs. QAGen2S is employed to generate questions on NQ and PubMed.

4.7 Experiments with Large QA Models

The downstream RC models presented in previous sections were based on fine-tuning `BERT-base` model, which has 110 million parameters. In this section, we assess the efficacy of our proposed domain adaptation approach on a higher capacity transformer as the RC model. For these experiments, we chose pretrained `RoBERTa-large` (Liu et al., 2019) model from transformers library (Wolf et al., 2019), which has 355 million parameters. Tab. 6 displays the domain adaptation results on the NQ domain using QAGen2S generated samples. It also includes performance on the source domain dev set. Although the SQuAD 1.1 baselines (first row), is significantly higher than those with `BERT-base` in Tab. 2, EM/F1 gains of 5.8/3.4 are achieved on the target domain. 1/0.5 gains in EM/F1 are observed in SQuAD 1.1 dev set. These results demonstrate that our proposed end-to-end synthetic data generation approach is capable of achieving substantial gains even on state-of-the-art RC baselines such as `RoBERTa-large`.

Model	**FT**	**SQuAD 1.1**		**NQ**	
	Data	EM	F1	EM	F1
SQuAD1.1 (SQ)	SQ	86.43	93.18	50.57	67.09
QAGen2S w/o filter.	Synth.	85.39	92.15	51.20	67.25
	+ SQ	86.23	93.19	50.73	67.07
QAGen2S + LM filter.	Synth.	85.77	92.07	55.06	68.83
	+ SQ	86.75	93.50	55.73	70.04
QAGen2S + RT filter.	Synth.	85.80	92.15	56.46	70.39
	+ SQ	**87.46**	**93.67**	**56.35**	**70.47**

Table 6: Source and target domain performance with RoBERTa-large as downstream RC model.

5 Conclusions

We presented a novel end-to-end approach to generate question-answer pairs by using a single transformer-based model. Our experiments showed that by proper decoding, significant improvements in domain adaptation of RC models can be achieved. We concluded that using LM filtering improves the quality of synthetic question-answer pairs; however, there is still a gap with round-trip filtering with some of the target domains. Improving LM-score-based filtering is a future direction of our work.

While we were able to generate diverse, high quality and challenging synthetic samples on the target domains, the types of the questions produced still were limited to those of SQuAD, since the generative models were trained on SQuAD. It would be interesting to explore how one can adapt the generative models to the type of target domain questions.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot.

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.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2020. Reinforcement learning based graph-to-sequence model for natural question generation. In *International Conference on Learning Representations*.

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.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Tassilo Klein and Moin Nabi. 2019. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. *arXiv preprint arXiv:1911.02365*.

Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuan-Fang Li. 2019. Putting the horse before the cart: A generator-evaluator framework for question generation from text. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 812–821.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. 2020. Generating diverse and consistent qa pairs from contexts with information-maximizing hierarchical conditional vaes.

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

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.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Xiyao Ma, Qile Zhu, Yanlin Zhou, and Xiaolin Li. 2020. Improving question generation with sentence-level semantic matching and answer position inferring. In *AAAI 2020*.

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 - Volume 2*, HLT-NAACL-EDUC '03, page 17–22, USA. Association for Computational Linguistics.

Kosuke Nishida, Kyosuke Nishida, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Unsupervised domain adaptation of language models for reading comprehension.

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 8024–8035.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Christian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*.

Mrinmaya Sachan and Eric Xing. 2018. Self-training for jointly learning to ask and answer questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 629–640, New Orleans, Louisiana. Association for Computational Linguistics.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension. In *Meeting of the Association for Computational Linguistics (ACL)*.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574, New Orleans, Louisiana. Association for Computational Linguistics.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, Brussels, Belgium. Association for Computational Linguistics.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel Ngonga, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138.

Luu Anh Tuan, Darsh J Shah, and Regina Barzilay. 2019. Capturing greater context for question generation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Xiaochuan Wang, Bingning Wang, Ting Yao, Qi Zhang, and Jingfang Xu. 2020. Neural question generation with answer pivot. In *AAAI 2020*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

A Additional Details Regarding the Datasets Used

SQuAD 1.1 dataset is used to train the generative models as well as in-domain supervised data for the downstream RC task. We use the default train and dev splits, which contain 87,599 and 10,570 (q, a) pairs, respectively. SQuAD 1.1 questions exhibit high lexical overlap with answers, since annotators were presented with passages and extracted answers when creating questions.

Natural Questions dataset consists of Google search questions, the Wikipedia pages from top 5 search results, and the corresponding annotated answers. This dataset and SQuAD are both derived from Wikipedia pages, however, questions from Natural Questions have considerably less ngram overlap with annotated answers compared to those from SQuAD. Also different from SQuAD, Natural Questions dataset contains passages with HTML tables and tags. We use MRQA Shared Task preprocessed training and dev sets, which consist of 104,071 and 12,836 (q, a) pairs, respectively. We utilize training set passages as the target domain (unlabeled) corpus, while preforming evaluations on the dev set.

NewsQA consists of question and answer pairs from CNN news articles. We use the dev set from the MRQA Shared Task, which removes unanswerable questions and those without annotator agreement. We believe this version better suits our work, as we focus only on generation of answerable questions. The train and dev sets consists of 74,160 and 4,212 samples, respectively. Passages from CNN/Daily Mail corpus are used as target domain passages.

BioASQ challenge is a competition on semantic indexing and question answering tasks based on annotated PubMed documents. As with the previous dataset, we employ MRQA shared task version of BioASQ, which consists of a dev set with 1,504 pairs. We collected PubMed abstracts to use as target domain passages. Being from Biomedical domain, BioASQ makes a clear domain shift from other datasets.

DuoRC contains question answer pairs from movie plots which are extracted from both Wikipedia and IMDB. This dataset has been developed to have question and answer pairs with minimal lexical overlap, which makes it more challenging. ParaphraseRC task of DuoRC dataset is used in our evaluations. Training and dev sets include 39,144 and 13,111 pairs, respectively. We crawled IMDB movie plots to use as the target domain unlabeled corpus. The dataset has been developed by selecting the same movie plot from both sources, and generating question from one source and selecting the answer from the other. This approach has resulted in question and answer pairs with minimal lexical overlap.

All of the validation sets of the aforementioned out-of-domain tasks are identical to those used by Nishida et al., except DuoRC, where we use MRQA shared task formatted DuoRC dev set.

B Additional Ablation Studies

B.1 Performance on SQuAD 1.1 with Different Filtering Approaches

While the performance of the RC models on the target domains is important, weak performance on the source domain could inhibit the use of our proposed methods in applications that require strong performance in both source and target domains. Tab. 7 shows EM/F1 scores of the `bert-base-uncased` RC models trained with synthetic data generated from the IMDB corpus on SQuAD 1.1 dev set. We can observe that adding synthetic samples to the SQuAD training set always improves the performance on the dev set compared to using the SQuAD training set only. In fact, with QAGen2S, impressive 3.1(EM)/2.2(F1) gains are achieved. Synthetic only samples from the same model outperform the SQuAD baseline. Similar to previous domain adaptation results, we observe that QAGen2S outperforms QAGen, and QAGen exceeds AQGen.

B.2 Comparison of Using Filtering vs. No Filtering

Tab. 8 presents comprehensive results of using LM filtering over all the of the target domains. We can observe that the arguments made in Sec. 4.7 hold for NewsQA and DuoRC as well.

B.3 Impact of Language Model Score Pooling

To aggregate the LM scores of a given question-answer pair, one can use either sum or average of the token scores, as defined in Sec. 2.4. We experimented with both options and summarized the results in Tab. 9 for QAGen and AQGen models. We can observe that using summation generally outperforms averaging. We speculate this is because average pooling encourages longer question-

Model	fine-tune Data	None		LM		Rountrip	
		EM	F1	EM	F1	EM	F1
QGen	Synthetic	70.31	80.34	–	–	77.11	84.81
	+ SQuAD	81.50	89.01	–	–	**82.94**	**89.68**
AQGen	Synthetic	74.58	84.14	74.34	83.55	78.51	86.21
	+ SQuAD	82.10	89.47	82.15	89.31	**82.88**	**89.78**
QAGen	Synthetic	79.65	87.14	78.40	85.98	78.51	86.21
	+ SQuAD	**83.07**	**90.00**	82.53	89.51	83.03	89.74
QAGen2S	Synthetic	81.25	88.20	79.86	86.79	80.61	87.36
	+ SQuAD	**83.87**	**90.40**	83.33	89.92	83.29	89.84

Table 7: Performance on SQuAD 1.1 development set when training with LM-filtered synthetically generated question-answer pairs on IMDB corpus. **Bold** values indicate best performance per each model (row-wise). Our baseline EM and F1 numbers (on SQuAD 1.1 training set) are 80.78 and 88.20, respectively.

Model	fine-tune Data	NQ		NewsQA		BioASQ		DuoRC		Synthetic #
		EM	F1	EM	F1	EM	F1	EM	F1	
AQGen w/o filtering	Synthetic	46.93	60.71	36.21	53.83	41.49	53.59	26.94	33.46	860k
	+ SQuAD	46.84	61.00	36.99	54.47	41.36	53.84	26.87	33.43	
AQGen + LM filtering	Synthetic	47.80	61.29	38.56	55.42	39.49	52.11	27.09	33.47	490k
	+ SQuAD	**49.04**	**62.56**	**39.62**	**56.89**	**42.89**	**54.90**	**27.88**	**34.40**	
QAGen w/o filtering	Synthetic	50.67	64.04	43.07	**59.53**	43.15	53.20	29.68	35.78	890k
	+ SQuAD	**51.35**	**64.99**	42.64	59.4	45.21	54.94	29.87	**35.87**	
QAGen + LM filtering	Synthetic	49.81	63.36	43.1	57.94	42.49	51.95	29.46	35.25	500k
	+ SQuAD	50.01	63.10	**44.06**	59.20	**45.74**	**55.06**	**29.91**	35.82	
QAGen2S w/o filtering	Synthetic	47.12	62.61	43.38	60.1	46.88	58.92	30.04	**36.58**	890k
	+ SQuAD	46.73	62.63	43.87	**60.51**	47.41	**59.33**	30.00	36.49	
QAGen2S + LM filtering	Synthetic	**52.64**	65.56	**43.99**	59.94	**48.40**	58.33	29.91	35.81	480k
	+ SQuAD	52.03	**65.70**	43.57	59.8	46.74	57.76	**30.06**	36.05	

Table 8: Comparison of using LM filtering versus no filtering. **Bold** values indicate best performance on each target domain for each model (per rows separated by sold lines).

answer pairs, which are more likely to consist of incorrect samples. By using summation, shorter question-answer pairs would be more likely to be selected during LM filtering.

C Examples of Generated Samples

C.1 Illustration of Answer LM Score

Tab. 10 presents unfiltered question-answer pairs and associated answer LM scores generated from a randomly selected Natural Questions corpus using the QAGen2S model. As can be seen from Topk+Nucleus decoded samples, the last two generated samples are incorrect and would be filtered out using the LM filtering approach that is used in this work. The last sample, which consists of an answer that is entirely irrelevant to its question, has a considerably lower answer LM score than the rest of the samples.

With beam search, due to the high number of repetitions, the scores are close. While beam search generates samples with high likelihood, due to the lack of diversity, as evident here, the performance of the trained RC models on such synthetic samples underperforms those of Topk+Nucleus.

C.2 Comparison of Generated Samples by AQGen, QAGen and QAGen2S

Tab. 11 presents unfiltered question-answers pairs generated using each of our proposed models on a randomly selected passage from CNN/Daily Mail corpus. We can observe that generated samples using AQGen have lower quality than the other two models. Also, the selected spans are repetitive. Only 3 out of the 6 properly generated samples are correct question-answer pairs. Comparing QAGen and QAGen2S samples, we can observe that QAGen2S generates more diverse and longer answer spans. In this example, we can see that more repeated spans are generated by QAGen than QAGen2S.

While the Topk+Nucleus sampling approach improves the diversity of generated question-answer pairs, we can still see repetitions and incorrect pairs. We believe using the LM score filtering, the vast majority of incorrect pairs are discarded. However, this also means there is room for improving the generative models.

C.3 Question Answers from Table

The Natural Questions dataset includes HTML formatted passages. We noticed that some of them

Model	LM Pooling	fine-tune Data	NQ		NewsQA		BioASQ		DuoRC	
			EM	F1	EM	F1	EM	F1	EM	F1
AQGen	Sum	Synthetic	47.80	61.29	38.56	55.42	39.49	52.11	27.09	33.47
		+ SQuAD	**49.04**	**62.56**	**39.62**	**56.89**	**42.89**	**54.90**	**27.88**	**34.40**
	Avg	Synthetic	47.73	61.98	34.19	52.05	39.03	51.52	26.31	32.84
		+ SQuAD	45.03	59.87	35.21	53.08	40.82	53.74	26.7	33.26
QAGen	Sum	Synthetic	49.81	63.36	43.1	57.94	42.49	51.95	29.46	35.25
		+ SQuAD	50.01	63.10	**44.06**	**59.20**	**45.74**	**55.06**	**29.91**	**35.82**
	Avg	Synthetic	**50.3**	**63.93**	43.14	58.82	41.82	52.22	28.5	34.51
		+ SQuAD	50.18	63.71	42.76	58.65	42.15	52.21	29.01	35.05

Table 9: Comparison of using average versus summation of LM scores when doing LM filtering. **Bold** values indicate the best performance on each target domain for each model (per rows separated by solid lines).

Passage: <P> The United States is estimated to have a population of **327,589,916** as of April 23 , 2018 , making it the **third** most populous country in the world . It is very urbanized , with **81** % residing in **cities and suburbs** as of 2014 (the worldwide urban rate is 54 %) . **California and Texas** are the most populous states , as the mean center of U.S. population has consistently shifted westward and southward . **New York City** is the most populous city in the United States . </P>		
Topk+Nucleus		
Q: *As of April 23, 2018, what is the estimated population of the US?*	A: *327,589,916*	LM score: *-0.00577*
Q: *How many people lived in the US in April of 2018?*	A: *327,589,916*	LM score: *-0.00707*
Q: *What is the population of the United States?*	A: *327,589,916*	LM score: *-0.01358*
Q: *What is the most populous city in the United States?*	A: *New York City*	LM score: *-0.04131*
Q: *Where do 81 percent of Americans live?*	A: *cities and suburbs*	LM score: *-0.05360*
Q: *Where does the United States rank among most populous countries on the planet?*	A: *third*	LM score: *-0.07449*
Q: *How much of the US's population is concentrated in the metropolitan areas of the country?*	A: *81 %*	LM score: *-0.09509*
Q: *How much of the US population is urbanized?*	A: *81 %*	LM score: *-0.1375*
Q: *What two cities have the highest populations in America?*	A: *California and Texas*	LM score: *-0.18128*
Q: *What country is considered the most populous?*	A: *third*	LM score: *-1.85929*
Beam Search		
Q: *What is the population of the United States as of April 23, 2018?*	A: *327,589,916*	LM score: *-0.00492*
Q: *As of April 23, 2018, what was the population of the United States?*	A: *327,589,916*	LM score: *-0.00529*
Q: *As of April 23, 2018, how many people live in the United States?*	A: *327,589,916*	LM score: *-0.00618*
Q: *How many people live in the United States?*	A: *327,589,916*	LM score: *-0.0132*
Q: *What is the population of the United States?*	A: *327,589,916*	LM score: *-0.0135*

Table 10: Samples of generated question-answers pairs using QAGen2S model from Natural Questions passages with their LM scores. Sum of answer likelihood scores is used to sort the pairs decreasingly. The generated answers are shown in **bold**. Samples shown from Beam Search with beam size of 5, and Topk+Nucleus with sample size of 10.

are web tables. Tab. 12 illustrates one such example. The content under ***Passage*** is the input string, as seen by the generative models, and ***Rendered Passage*** indicates how the table appears in a browser. We experimented with using **QGen** model on this passage, and noticed that the span detection model was not capable of distinguishing between textual content and HTML tags properly, resulting in selecting spans that included HTML tags. However, the samples generated by the joint span and question generation model, QAGen2S in this example, show surprisingly high-quality spans and questions. Only one sample is not correct (*What team is Tampa Bay's home arena?*). We believe this is because when the span generation is conditioned on the generated question, the likelihood of spans that include spurious tokens, HMTL tags in this example, diminishes sharply. This opens the door to the possibility of using our proposed models in structured corpora without any extra effort.

D Training and Platform Details

All of the experiments in this work were performed on Amazon EC2 instances. We employed p3.8xlarge, p3.16xlarge, and p3dn.24xlarge GPU instances. In the training of the generative models, warmup was set to 10% of total training steps. We used a batch size of 24. Each epoch took 2 to 3 hours on 3 GPUs. We observed that usually, the best model is achieved within the first two epochs.

The RC models with Synthetic+SQuAD samples were trained by combining synthetic samples and SQuAD training set and randomly shuffling them. Each epoch of training took 2 to 12 hours, depending on the average length of target domain passages on 1 GPU.

All of the hyperparameters of both generative and RC downstream models were fixed. We only performed hyperparameter tuning on those mentioned in the paper.

Passage: (CNN) – Fifteen people have now died after consuming **cantaloupe** contaminated with the listeria monocytogenes bacteria, the Centers for Disease Control and Prevention said **Friday**. At least **84** people in 19 states have become ill with the bacteria, the agency said. And the **number of illnesses** could still grow, added the CDC, citing **reporting lags and how the disease can develop slowly in some people**. On Tuesday, the CDC was reporting 13 deaths and 72 illnesses in what was already then the deadliest food-borne illness outbreak in the United States since **1998**. Five people have died in New Mexico from eating the tainted cantaloupes, the CDC said. **Three** people died in Colorado, **two** in Texas and one each in Kansas, Maryland, Missouri, Nebraska and Oklahoma. Illnesses have also been reported in Alabama, Arkansas, California, Illinois, Indiana, Montana, North Dakota, Virginia, West Virginia, Wisconsin and Wyoming. What you need to know about Listeria. **Most of those who fell ill are more than 60 years old**, the CDC said. Doctors also are closely monitoring the pregnancies of two women who ate contaminated cantaloupe, with the agency noting that **listeriosis** can cause miscarriages and stillbirths. Older adults and people with compromised immune systems are also especially susceptible. Public health investigators have traced the source of the bacteria to a farm in **Granada, Colorado**. Food Poisoning 101. The grower, **Jensen Farms**, issued a recall for its Rocky Ford-brand cantaloupes on September 14. By now, the cantaloupes should all be off store shelves, the CDC said. The agency warned that people should not eat Rocky Ford cantaloupes, even if they have eaten part of one and have not yet fallen ill. It also said that consumers **should be wary of eating any cantaloupes if they don't know where they came from**. How to keep your food safe.	
AQGen :	
Q: *What can cause miscarriages?*	A: *listeriosis*
Q: *Which state has had the most deaths?*	A: *Colorado*
Q: *Where is the farm where the bacteria came from?*	A: *Colorado*
Q: *How many people have died from eating listeria from cantaloupe?*	A: *14*
Q: *Where has the worst case happened?*	A: *Colorado*
Q: *Where were the listeria monocytogenes bacteria come from?*	A: *Granada*
QAGen :	
Q: *What year was the deadliest food-borne illness outbreak in the United States since?*	A: *1998*
Q: *How old were most of the victims of the outbreak?*	A: *more than 60 years old*
Q: *How old were most of the people who died from the listeria infection?*	A: *more than 60 years old*
Q: *How many people in the US have become seriously ill with Listeria?*	A: *84*
Q: *How many people in Texas were killed by tainted cantaloupes?*	A: *two*
Q: *How old were most of the people who died from the listeria infection?*	A: *more than 60*
Q: *How many people were reported killed in Colorado?*	A: *Three*
Q: *Where has the food poisoning been traced to?*	A: *Granada, Colorado*
Q: *Who did the CDC have in custody over the tainted cantaloupes?*	A: *Jensen Farms*
Q: *Who released the recall announcement?*	A: *Jensen Farms*
QAGen2S :	
Q: *What can cause miscarriages and stillbirths?*	A: *listeriosis*
Q: *What type of food was it?*	A: *cantaloupe*
Q: *What was the first year of death from this outbreak?*	A: *1998*
Q: *How does the food-borne illness outbreak effect those over 60?*	A: *Most of those who fell ill are more than 60 years old*
Q: *When did the CDC start reporting the Listeria monocytogenes bacteria in cantaloupes?*	A: *Friday*
Q: *How old are most of those in the recent outbreak?*	A: *more than 60 years old*
Q: *How could the number of sickened listeria possibly grow?*	A: *reporting lags and how the disease can develop slowly in some people*
Q: *When did the CDC start reporting the Listeria monocytogenes bacteria?*	A: *Friday*
Q: *What could still grow?*	A: *number of illnesses*
Q: *How can listeriosis be avoided?*	A: *should be wary of eating any cantaloupes if they don't know where they came from*

Table 11: Samples of generated question-answers pairs from randomly selected passage from CNN/Daily Mail corpus. Samples are sorted according to LM scores.

Passage:

<Table> <Tr> <Th colspan="2"> Tampa Bay Lightning </Th> </Tr> <Tr> <Td colspan="2"> 2018 – 19 Tampa Bay Lightning season </Td> </Tr> <Tr> <Td colspan="2"> </Td> </Tr> <Tr> <Th> Conference </Th> <Td> Eastern </Td> </Tr> <Tr> <Th> Division </Th> <Td> Atlantic </Td> </Tr> <Tr> <Th> Founded </Th> <Td> 1992 </Td> </Tr> <Tr> <Th> History </Th> <Td> Tampa Bay Lightning 1992 – present </Td> </Tr> <Tr> <Th> Home arena </Th> <Td> Amalie Arena </Td> </Tr> <Tr> <Th> City </Th> <Td> Tampa , Florida </Td> </Tr> <Tr> <Td colspan="2"> </Td> </Tr> <Tr> <Th> Colors </Th> <Td> Tampa Bay blue , white </Td> </Tr> <Tr> <Th> Media </Th> <Td> Fox Sports Sun 970 AM </Td> </Tr> <Tr> <Th> Owner (s) </Th> <Td> Tampa Bay Sports and Entertainment (Jeffrey Vinik , chairman) </Td> </Tr> <Tr> <Th> General manager </Th> <Td> Steve Yzerman </Td> </Tr> <Tr> <Th> Head coach </Th> <Td> Jon Cooper </Td> </Tr> <Tr> <Th> Captain </Th> <Td> Steven Stamkos </Td> </Tr> <Tr> <Th> Minor league affiliates </Th> <Td> Syracuse Crunch (AHL) Orlando Solar Bears (ECHL) </Td> </Tr> <Tr> <Th> Stanley Cups </Th> <Td> 1 (2003 – 04) </Td> </Tr> <Tr> <Th> Conference championships </Th> <Td> 2 (2003 – 04 , 2014 – 15) </Td> </Tr> <Tr> <Th> Presidents ' Trophy </Th> <Td> 0 </Td> </Tr> <Tr> <Th> Division championships </Th> <Td> 3 (2002 – 03 , 2003 – 04 , 2017 – 18) </Td> </Tr> <Tr> <Th> Official website </Th> <Td> www.nhl.com/lightning </Td> </Tr> </Table>

Rendered Passage:

Tampa Bay Lightning

2018 -- 19 Tampa Bay Lightning season

Conference	Eastern
Division	Atlantic
Founded	1992
History	Tampa Bay Lightning 1992 -- present
Home arena	Amalie Arena
City	Tampa , Florida
Colors	Tampa Bay blue , white
Media	Fox Sports Sun 970 AM
Owner (s)	Tampa Bay Sports and Entertainment (Jeffrey Vinik , chairman)
General manager	Steve Yzerman
Head coach	Jon Cooper
Captain	Steven Stamkos
Minor league affiliates	Syracuse Crunch (AHL) Orlando Solar Bears (ECHL)
Stanley Cups	1 (2003 -- 04)
Conference championships	2 (2003 -- 04 , 2014 -- 15)
Presidents ' Trophy	0
Division championships	3 (2002 -- 03 , 2003 -- 04 , 2017 -- 18)
Official website	www.nhl.com/lightning

Q: *What year was the Tampa Bay Lightning established??*	A: *1992*	LM score: *-0.001539*
Q: *Who is the head coach of the Tampa Bay Lightning?*	A: *Jon Cooper*	LM score: *-0.0015659*
Q: *Who is the Tampa Bay Lightning general manager?*	A: *Steve Yzerman*	LM score: *-0.002090*
Q: *Who is the Head coach of the Tampa Bay Lightning?*	A: *Jon Cooper*	LM score: *-0.003044*
Q: *Who is the General Manager of the Tampa Bay Lightning?*	A: *Steve Yzerman*	LM score: *-0.003877*
Q: *What team is Tampa Bay's home arena?*	A: *Amalie Arena*	LM score: *-0.00543*
Q: *For whom did Jeffrey Vinik serve as chairman?*	A: *Tampa Bay Sports and Entertainment*	LM score: *-0.0215854*
Q: *Tampa Bay Sports and Entertainment is owned by what?*	A: *Jeffrey Vinik*	LM score: *-0.087364*

Table 12: Generated samples using QAGen2S model from a Natural Questions passage consisting of a table. Sum of answer likelihood scores are chosen to sort the pairs decreasingly.