Synthetic QA Corpora Generation with Roundtrip Consistency

Chris Alberti **Daniel Andor** **Emily Pitler** **Jacob Devlin** **Michael Collins**
Google Research
{chrisalberti, andor, epitler, jacobdevlin, mjcollins}@google.com

Abstract

We introduce a novel method of generating synthetic question answering corpora by combining models of question generation and answer extraction, and by filtering the results to ensure roundtrip consistency. By pretraining on the resulting corpora we obtain significant improvements on SQuAD2 (Rajpurkar et al., 2018) and NQ (Kwiatkowski et al., 2019), establishing a new state-of-the-art on the latter. Our synthetic data generation models, for both question generation and answer extraction, can be fully reproduced by finetuning a publicly available BERT model (Devlin et al., 2018) on the extractive subsets of SQuAD2 and NQ. We also describe a more powerful variant that does full sequence-to-sequence pretraining for question generation, obtaining exact match and F1 at less than 0.1% and 0.4% from human performance on SQuAD2.

1 Introduction

Significant advances in Question Answering (QA) have recently been achieved by pretraining deep transformer language models on large amounts of unlabeled text data, and finetuning the pretrained models on hand labeled QA datasets, e.g. with BERT (Devlin et al., 2018).

Language modeling is however just one example of how an auxiliary prediction task can be constructed from widely available natural text, namely by masking some words from each passage and training the model to predict them. It seems plausible that other auxiliary tasks might exist that are better suited for QA, but can still be constructed from widely available natural text. It also seems intuitive that such auxiliary tasks will be more helpful the closer they are to the particular QA task we are attempting to solve.

Based on this intuition we construct auxiliary tasks for QA, generating millions of synthetic question-answer-context triples from unlabeled passages of text, pretraining a model on these examples, and finally finetuning on a particular labeled dataset. Our auxiliary tasks are illustrated in Table 1.

Input (C)		... in 1903, boston participated in the first modern world series, going up against the pittsburgh pirates ...
(1)	$C \rightarrow A$	1903
(2)	$C, A \rightarrow Q$	when did the red sox first go to the world series
(3)	$C, Q \rightarrow A'$	1903
(4)	$A \stackrel{?}{=} A'$	Yes

Table 1: Example of how synthetic question-answer pairs are generated. The model's predicted answer (A') matches the original answer the question was generated from, so the example is kept.

For a given passage C, we sample an extractive short answer A (Step (1) in Table 1). In Step (2), we generate a question Q conditioned on A and C, then (Step (3)) predict the extractive answer A' conditioned on Q and C. If A and A' match we finally emit (C, Q, A) as a new synthetic training example (Step (4)). We train a separate model on labeled QA data for each of the first three steps, and then apply the models in sequence on a large number of unlabeled text passages. We show that pretraining on synthetic data generated through this procedure provides us with significant improvements on two challenging datasets, SQuAD2 (Rajpurkar et al., 2018) and NQ (Kwiatkowski et al., 2019), achieving a new state of the art on the latter.

2 Related Work

Question generation is a well-studied task in its own right (Heilman and Smith, 2010; Du et al., 2017; Du and Cardie, 2018). Yang et al. (2017) and Dhingra et al. (2018) both use generated

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6168–6173
Florence, Italy, July 28 - August 2, 2019.

question-answer pairs to improve a QA system, showing large improvements in low-resource settings with few gold labeled examples. Validating and improving the accuracy of these generated QA pairs, however, is relatively unexplored.

In machine translation, modeling consistency with dual learning (He et al., 2016) or back-translation (Sennrich et al., 2016) across both translation directions improves the quality of translation models. Back-translation, which adds synthetically generated parallel data as training examples, was an inspiration for this work, and has led to state-of-the-art results in both the supervised (Edunov et al., 2018) and the unsupervised settings (Lample et al., 2018).

Lewis and Fan (2019) model the joint distribution of questions and answers given a context and use this model directly, whereas our work uses generative models to generate synthetic data to be used for pretraining. Combining these two approaches could be an area of fruitful future work.

3 Model

Given a dataset of contexts, questions, and answers: $\{(c^{(i)}, q^{(i)}, a^{(i)}) : i = 1, \ldots, N\}$, we train three models: (1) answer extraction: $p(a|c; \theta_A)$, (2) question generation: $p(q|c, a; \theta_Q)$, and (3) question answering: $p(a|c, q; \theta_{A'})$.

We use BERT (Devlin et al., 2018)* to model each of these distributions. Inputs to each of these models are fixed length sequences of wordpieces, listing the tokenized question (if one was available) followed by the context c. The answer extraction model is detailed in §3.1 and two variants of question generation models in §3.2 and §3.3. The question answering model follows Alberti et al. (2019).

3.1 Question (Un)Conditional Extractive QA

We define a question-unconditional extractive answer model $p(a|c; \theta_A)$ and a question-conditional extractive answer model $p(a|q, c; \theta_{A'})$ as follows:

$$p(a|c; \theta_A) = \frac{e^{f_J(a,c;\theta_A)}}{\sum_{a''} e^{f_J(a'',c;\theta_A)}}$$

$$p(a|c, q; \theta_{A'}) = \frac{e^{f_I(a,c,q;\theta_{A'})}}{\sum_{a''} e^{f_I(a'',c,q;\theta_{A'})}}$$

*Some experiments use a variant of BERT that masks out whole words at training time, similar to Sun et al. (2019). See https://github.com/google-research/bert for both the original and whole word masked versions of BERT.

where a, a'' are defined to be token spans over c. For $p(a|c; \theta_A)$, a and a'' are constrained to be of length up to L_A, set to 32 word piece tokens. The key difference between the two expressions is that f_I scores the start and the end of each span independently, while f_J scores them jointly.

Specifically we define $f_J : \mathbb{R}^h \rightarrow \mathbb{R}$ and $f_I : \mathbb{R}^h \rightarrow \mathbb{R}$ to be transformations of the final token representations computed by a BERT model:

$$f_J(a, c; \theta_A) = \text{MLP}_J(\text{CONCAT}(\text{BERT}(c)[s], \text{BERT}(c)[e]))$$

$$f_I(a, q, c; \theta_{A'})) = \text{AFF}_I(\text{BERT}(q, c)[s]) + \text{AFF}_I(\text{BERT}(q, c)[e]).$$

Here h is the hidden representation dimension, $(s, e) = a$ is the answer span, $\text{BERT}(t)[i]$ is the BERT representation of the i'th token in token sequence t. MLP_J is a multi-layer perceptron with a single hidden layer, and AFF_I is an affine transformation.

We found it was critical to model span start and end points jointly in $p(a|c; \theta_A)$ because, when the question is not given, there are usually multiple acceptable answers for a given context, so that the start point of an answer span cannot be determined separately from the end point.

3.2 Question Generation: Fine-tuning Only

Text generation allows for a variety of choices in model architecture and training data. In this section we opt for a simple adaptation of the public BERT model for text generation. This adaptation does not require any additional pretraining and no extra parameters need to be trained from scratch at finetuning time. This question generation system can be reproduced by simply finetuning a publicly available pretrained BERT model on the extractive subsets of datasets like SQuAD2 and NQ.

Fine-tuning We define the $p(q|c, a; \theta_Q)$ model as a left-to-right language model

$$p(q|a, c; \theta_Q) = \prod_{i=1}^{L_Q} p(q_i|q_1, \ldots, q_{i-1}, a, c; \theta_Q)$$

$$= \prod_{i=1}^{L_Q} \frac{e^{f_Q(q_1,\ldots,q_i,a,c;\theta_Q)}}{\sum_{q_i'} e^{f_Q(q_1,\ldots,q_i',a,c;\theta_Q)}},$$

where $q = (q_1, \ldots, q_{L_Q})$ is the sequence of question tokens and L_Q is a predetermined maximum question length, but, unlike the more usual

encoder-decoder approach, we compute f_Q using the single encoder stack from the BERT model:

$$f_Q(q_1, \ldots, q_i, a, c; \theta_Q) = \text{BERT}(q_1, \ldots, q_{i-1}, a, c)[i-1] \cdot W_{\text{BERT}}^{\intercal},$$

where W_{BERT} is the word piece embedding matrix in BERT. All parameters of BERT including W_{BERT} are finetuned. In the context of question generation, the input answer is encoded by introducing a new token type id for the tokens in the extractive answer span, e.g. the question tokens being generated have type 0 and the context tokens have type 1, except for the ones in the answer span that have type 2. We always pad or truncate the question being input to BERT to a constant length L_Q to avoid giving the model information about the length of the question we want it to generate.

This model can be trained efficiently by using an attention mask that forces to zero all the attention weights from c to q and from q_i to $q_{i+1} \ldots q_{L_Q}$ for all i.

Question Generation At inference time we generate questions through iterative greedy decoding, by computing $\text{argmax}_{q_i} f_Q(q_1, \ldots, q_i, a, c)$ for $i = 1, \ldots, L_Q$. Question-answer pairs are kept only if they satisfy roundtrip consistency.

3.3 Question Generation: Full Pretraining

The prior section addressed a restricted setting in which a BERT model was fine-tuned, without any further changes. In this section, we describe an alternative approach for question generation that fully pretrains and fine-tunes a sequence-to-sequence generation model.

Pretraining Section 3.2 used only an encoder for question generation. In this section, we use a full sequence-to-sequence Transformer (both encoder and decoder). The encoder is trained identically (BERT pretraining, Wikipedia data), while the decoder is trained to output the next sentence.

Fine-tuning Fine-tuning is done identically as in Section 3.2, where the input is (C, A) and the output is Q from tuples from a supervised question-answering dataset (e.g., SQuAD).

Question Generation To get examples of synthetic (C, Q, A) triples, we sample from the decoder with both beam search and Monte Carlo search. As before, we use roundtrip consistency to keep only the high precision triples.

3.4 Why Does Roundtrip Consistency Work?

A key question for future work is to develop a more formal understanding of why the roundtrip method improves accuracy on question answering tasks (similar questions arise for the back-translation methods of Edunov et al. (2018) and Sennrich et al. (2016); a similar theory may apply to these methods). In the supplementary material we sketch a possible approach, inspired by the method of Balcan and Blum (2005) for learning with labeled and unlabeled data. This section is intentionally rather speculative but is intended to develop intuition about the methods, and to propose possible directions for future work on developing a formal grounding.

In brief, the approach discussed in the supplementary material suggests optimizing the log-likelihood of the labeled training examples, under a constraint that some measure of roundtrip consistency $\beta(\theta_{A'})$ on unlabeled data is greater than some value γ. The value for γ can be estimated using performance on development data. The auxiliary function $\beta(\theta_{A'})$ is chosen such that: (1) the constraint $\beta(\theta_{A'}) \geq \gamma$ eliminates a substantial part of the parameter space, and hence reduces sample complexity; (2) the constraint $\beta(\theta_{A'}) \geq \gamma$ nevertheless includes 'good' parameter values that fit the training data well. The final step in the argument is to make the case that the algorithms described in the current paper may effectively be optimizing a criterion of this kind. Specifically, the auxiliary function $\beta(\theta_{A'})$ is defined as the log-likelihood of noisy (c, q, a) triples generated from unlabeled data using the $C \rightarrow A$ and $C, A \rightarrow Q$ models; constraining the parameters $\theta_{A'}$ to achieve a relatively high value on $\beta(\theta_{A'})$ is achieved by pre-training the model on these examples. Future work should consider this connection in more detail.

4 Experiments

4.1 Experimental Setup

We considered two datasets in this work: SQuAD2 (Rajpurkar et al., 2018) and the Natural Questions (NQ) (Kwiatkowski et al., 2019). SQuAD2 is a dataset of QA examples of questions with answers formulated and answered by human annotators about Wikipedia passages. NQ is a dataset of Google queries with answers from Wikipedia pages provided by human annotators. We used the full text from the training set of NQ (1B words) as

	Dev		Test	
	EM	F1	EM	F1
Fine-tuning Only				
BERT-Large (Original)	78.7	81.9	80.0	83.1
+ 3M synth SQuAD2	80.1	82.8	-	-
+ 4M synth NQ	81.2	84.0	82.0	84.8
Full Pretraining				
BERT (Whole Word Masking)[†]	82.6	85.2	-	-
+ 50M synth SQuAD2	85.1	87.9	85.2	87.7
+ ensemble	86.0	88.6	86.7	89.1
Human	-	-	86.8	89.5

Table 2: Our results on SQuAD2. For our fine-tuning only setting, we compare a BERT baseline (BERT single model - Google AI Language on the SQuAD2 leaderboard) to similar models pretrained on our synthetic SQuAD2-style corpus and on a corpus containing both SQuAD2- and NQ-style data. For the full pretraining setting, we report our best single model and ensemble results.

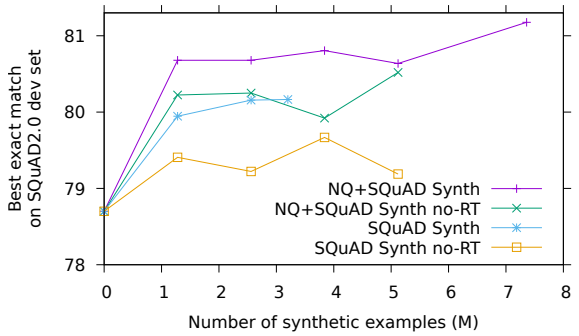

Figure 1: Learning curves for pretraining using synthetic question-answering data (fine-tuning only setting). "no-RT" refers to omitting the roundtrip consistency check. Best exact match is reported after fine-tuning on SQuAD2. Performance improves with the amount of synthetic data. For a fixed amount of synthetic data, having a more diverse source (NQ+SQuAD vs. just SQuAD) yields higher accuracies. Roundtrip filtering gives further improvements.

a source of unlabeled data.

In our fine-tuning only experiments (Section 3.2) we trained two triples of models $(\theta_A, \theta_Q, \theta_{A'})$ on the extractive subsets of SQuAD2 and NQ. We extracted 8M unlabeled windows of 512 tokens from the NQ training set. For each unlabeled window we generated one example from the SQuAD2-trained models and one example from the NQ-trained models. For A we picked an answer uniformly from the top 10 extractive answers according to $p(a|c; \theta_A)$. For A' we picked the best extractive answer according to $p(a|c, q; \theta_{A'})$. Filtering for roundtrip consistency gave us 2.4M and 3.2M synthetic positive instances from SQuAD2- and NQ-trained models respectively. We then added synthetic unanswerable instances by taking the question generated from a window and associating it with a non-overlapping window from the same Wikipedia page. We then sampled negatives to obtain a total of 3M and 4M synthetic training instances for SQuAD2 and NQ respectively. We trained models analogous to Alberti et al. (2019) initializing from the public BERT model, with a batch size of 128 examples for one epoch on each of the two sets of synthetic examples and on the union of the two, with a learning rate of $2 \cdot 10^{-5}$ and no learning rate decay. We then fine-tuned the the resulting models on SQuAD2 and NQ.

In our full pretraining experiments (Section 3.3) we only trained $(\theta_A, \theta_Q, \theta_{A'})$ on SQuAD2. However, we pretrained our question generation model on all of the BERT pretraining data, generating the next sentence left-to-right. We created a synthetic, roundtrip filtered corpus with 50M examples. We then fine-tuned the model on SQuAD2 as previously described. We experimented with both the single model setting and an ensemble of 6 models.

4.2 Results

The final results are shown in Tables 2 and 3. We found that pretraining on SQuAD2 and NQ synthetic data increases the performance of the fine-tuned model by a significant margin. On the NQ short answer task, the relative reduction in headroom is 50% to the single human performance and 10% to human ensemble performance. We additionally found that pretraining on the union of synthetic SQuAD2 and NQ data is very beneficial on the SQuAD2 task, but does not improve NQ results.

The full pretraining approach with ensembling obtains the highest EM and F1 listed in Table 2. This result is only $0.1 - 0.4\%$ from human performance and is the third best model on the SQuAD2 leaderboard as of this writing (5/31/19).

Roundtrip Filtering Roundtrip filtering appears to be consistently beneficial. As shown in Figure 1, models pretrained on roundtrip consistent data outperform their counterparts pretrained without filtering. From manual inspection, of 46 (C, Q, A) triples that were roundtrip consistent

[†] https://github.com/google-research/bert

	Long Answer Dev			Long Answer Test			Short Answer Dev			Short Answer Test		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
$\text{BERT}_{\text{joint}}$	61.3	68.4	64.7	64.1	68.3	66.2	59.5	47.3	52.7	**63.8**	44.0	52.1
+ 4M synth NQ	**62.3**	**70.0**	**65.9**	**65.2**	**68.4**	**66.8**	**60.7**	**50.4**	**55.1**	62.1	**47.7**	**53.9**
Single Human	80.4	67.6	73.4	-	-	-	63.4	52.6	57.5	-	-	-
Super-annotator	90.0	84.6	87.2	-	-	-	79.1	72.6	75.7	-	-	-

Table 3: Our results on NQ, compared to the previous best system and to the performance of a human annotator and of an ensemble of human annotators. $\text{BERT}_{\text{joint}}$ is the model described in Alberti et al. (2019).

	Question	Answer
NQ	what was the population of chicago in 1857?	over 90,000
SQuAD2	what was the weight of the brigg's hotel?	22,000 tons
NQ	where is the death of the virgin located?	louvre
SQuAD2	what person replaced the painting?	carlo saraceni
NQ	when did rick and morty get released?	2012
SQuAD2	what executive suggested that rick be a grandfather?	nick weidenfeld

Table 4: Comparison of question-answer pairs generated by NQ and SQuAD2 models for the same passage of text.

39% were correct, while of 44 triples that were discarded only 16% were correct.

Data Source Generated question-answer pairs are illustrative of the differences in the style of questions between SQuAD2 and NQ. We show a few examples in Table 4, where the same passage is used to create a SQuAD2-style and an NQ-style question-answer pair. The SQuAD2 models seem better at creating questions that directly query a specific property of an entity expressed in the text. The NQ models seem instead to attempt to create questions around popular themes, like famous works of art or TV shows, and then extract the answer by combining information from the entire passage.

5 Conclusion

We presented a novel method to generate synthetic QA instances and demonstrated improvements from this data on SQuAD2 and on NQ. We additionally proposed a possible direction for formal grounding of this method, which we hope to develop more thoroughly in future work.

References

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*.

Maria-Florina Balcan and Avrim Blum. 2005. A pac-style model for learning from labeled and unlabeled data. In *Proceedings of the 18th Annual Conference on Learning Theory*, COLT'05, pages 111–126, Berlin, Heidelberg. Springer-Verlag.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. 2018. Simple and effective semi-supervised question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 582–587.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. *arXiv preprint arXiv:1805.05942*.

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. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, pages 489–500. Association for Computational Linguistics.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617. Association for Computational Linguistics.

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

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, et al. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

Mike Lewis and Angela Fan. 2019. Generative question answering: Learning to answer the whole question. *International Conference on Learning Representations (ICLR)*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you dont know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 784–789.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 86–96.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-supervised qa with generative domain-adaptive nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1040–1050.