Improving the Robustness of QA Models to Challenge Sets with Variational Question-Answer Pair Generation

Kazutoshi Shinoda[†‡] **Saku Sugawara**[‡] **Akiko Aizawa**[†‡]
[†]The University of Tokyo
[‡]National Institute of Informatics
shinoda@is.s.u-tokyo.ac.jp
{saku,aizawa}@nii.ac.jp

Abstract

Question answering (QA) models for reading comprehension have achieved human-level accuracy on in-distribution test sets. However, they have been demonstrated to lack robustness to challenge sets, whose distribution is different from that of training sets. Existing data augmentation methods mitigate this problem by simply augmenting training sets with synthetic examples sampled from the same distribution as the challenge sets. However, these methods assume that the distribution of a challenge set is known a priori, making them less applicable to unseen challenge sets. In this study, we focus on question-answer pair generation (QAG) to mitigate this problem. While most existing QAG methods aim to improve the quality of synthetic examples, we conjecture that diversity-promoting QAG can mitigate the sparsity of training sets and lead to better robustness. We present a variational QAG model that generates multiple diverse QA pairs from a paragraph. Our experiments show that our method can improve the accuracy of 12 challenge sets, as well as the in-distribution accuracy.[1]

1 Introduction

Machine reading comprehension has gained significant attention in the NLP community, whose goal is to devise systems that can answer questions about given documents (Rajpurkar et al., 2016; Trischler et al., 2017; Joshi et al., 2017). Such systems usually use neural models, which require a substantial number of question-answer (QA) pairs for training. To reduce the considerable manual cost of dataset creation, there has been a resurgence of studies on automatic QA pair generation (QAG), consisting of a pipeline of answer extraction (AE) and question generation (QG), to augment question answering (QA) datasets (Yang et al., 2017a; Du and Cardie, 2018; Subramanian et al., 2018; Alberti et al., 2019).

For the downstream QA task, most existing studies have evaluated QAG methods using a test set from the same distribution as a training set (Yang et al., 2017a; Zhang and Bansal, 2019; Liu et al., 2020). However, when a QA model is evaluated only on an in-distribution test set, it is difficult to verify that the model is not exploiting unintended biases in a dataset (Geirhos et al., 2020). Exploiting an unintended bias can degrade the robustness of a QA model, which is problematic in real-world applications. For example, recent studies have observed that a QA model does not generalize to other QA datasets (Yogatama et al., 2019; Talmor and Berant, 2019; Sen and Saffari, 2020). Other studies have found a lack of robustness to challenge sets, such as paraphrased questions (Gan and Ng, 2019), questions with low lexical overlap (Sugawara et al., 2018), and questions that include noise (Ravichander et al., 2021).

While existing studies have proposed data augmentation methods targeting a particular challenge set, they are only effective at the expense of the in-distribution accuracy (Gan and Ng, 2019; Ribeiro et al., 2019; Ravichander et al., 2021). These methods assume that the target distribution is given a priori. However, identifying the type of samples that a QA model cannot handle in advance is difficult in real-world applications.

We conjecture that increasing the diversity of a training set with data augmentation, rather than augmenting QA pairs similar to the original training set, can improve the robustness of QA models. Poor diversity in QA datasets has been shown to result in the poor robustness of QA models (Lewis and Fan, 2019; Geva et al., 2019; Ko et al., 2020), supporting our hypothesis. To this end, we propose

[1] Our code and data are available at https://github.com/KazutoshiShinoda/VQAG.

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 197–214
August 5–6, 2021. ©2021 Association for Computational Linguistics

a variational QAG model (VQAG). We introduce two independent latent random variables into our model to learn the two one-to-many relationships in AE and QG by utilizing neural variational inference (Kingma and Welling, 2013). Incorporating the randomness of these two latent variables enables our model to generate diverse answers and questions separately. We also study the effect of controlling the Kullback–Leibler (KL) term in the variational lower bound for mitigating the posterior collapse issue (Bowman et al., 2016), where the model ignores latent variables and generates outputs that are almost the same. We evaluate our approach on 12 challenge sets that are unseen during training to assess the improved robustness of the QA model.

In summary, our contributions are three-fold:

- We propose a variational question-answer pair generation model with explicit KL control to generate significantly diverse answers and questions.
- We construct synthetic QA datasets using our model to boost the QA performance in an in-distribution test set, achieving comparable scores with existing QAG methods.
- We discover that our method achieves meaningful improvements in unseen challenge sets, which are further boosted using a simple ensemble method.

2 Related Work

2.1 Answer Extraction

AE aims to extract question-worthy phrases, which are worth being asked about, from each textual context without looking at the questions. AE has been performed mainly in two ways: rule-based and neural methods. Yang et al. (2017a) extracted candidate phrases using rule-based methods such as named entity recognition (NER). However, not all the named entities, noun phrases, verb phrases, adjectives, or clauses in the given documents are used as gold answer spans. As such, these rule-based methods are likely to extract many trivial phrases.

Therefore, there have been studies on training neural models to identify question-worthy phrases. Du and Cardie (2018) framed AE as a sequence labeling task and used BiLSTM-CRF (Huang et al., 2015). Subramanian et al. (2018) treated the positions of answers as a sequence and used a pointer network (Vinyals et al., 2015). Wang et al. (2019) used a pointer network and Match-LSTM (Wang and Jiang, 2016, 2017). Alberti et al. (2019) made use of pretrained BERT (Devlin et al., 2019).

However, these neural AE models are trained with maximum likelihood estimation; that is, each model is optimized to produce an answer set closest to the gold answers. In contrast, our model incorporates a latent random variable and is trained by maximizing the lower bound of the likelihood to extract diverse answers. In this study, we assume that there should be question-worthy phrases that are not used as the gold answers in a manually created dataset. We aim to extract such phrases.

2.2 Question Generation

Traditionally, QG was studied using rule-based methods (Mostow and Chen, 2009; Heilman and Smith, 2010; Lindberg et al., 2013; Labutov et al., 2015). After Du et al. (2017) proposed a neural sequence-to-sequence model (Sutskever et al., 2014) for QG, neural models that take context and answer as inputs have started to be used to improve question quality with attention (Bahdanau et al., 2014) and copying (Gulcehre et al., 2016; Gu et al., 2016) mechanisms. Most works focused on generating relevant questions from context-answer pairs (Zhou et al., 2018; Song et al., 2018; Zhao et al., 2018; Sun et al., 2018; Kim et al., 2019; Liu et al., 2019; Qiu and Xiong, 2019). These works showed the importance of answers as input features for QG. Other works studied predicting question types (Zhou et al., 2019; Kang et al., 2019), modeling a structured answer-relevant relation (Li et al., 2019), and refining generated questions (Nema et al., 2019). To further improve question quality, policy gradient techniques have been used (Yuan et al., 2017; Yang et al., 2017a; Yao et al., 2018; Kumar et al., 2019). Dong et al. (2019) used a pretrained language model.

The diversity of questions has been tackled using variational attention (Bahuleyan et al., 2018), a conditional variational autoencoder (CVAE) (Yao et al., 2018), and top p nucleus sampling (Sultan et al., 2020). Our study is different from these studies wherein we study QAG by introducing variational methods into both AE and QG. Lee et al. (2020) is the closest to our study in terms of the modeling choice. While Lee et al. (2020) introduced an information-maximizing term to improve the consistency of QA pairs, our study uniquely controls the diversity by explicitly controlling KL values.

Despite the potential of data augmentation with QAG to mitigate the sparsity of QA datasets and avoid overfitting, not much is known about the robustness of QA models reinforced with QAG to more challenging test sets. We comprehensively evaluate QAG methods on challenging QA test sets, such as hard questions (Sugawara et al., 2018), implications (Ribeiro et al., 2019), and paraphrased questions (Gan and Ng, 2019).

2.3 Variational Autoencoder

The variational autoencoder (VAE) (Kingma and Welling, 2013) is a deep generative model consisting of a neural encoder (inference model) and decoder (generative model). The encoder learns to map from an observed variable to a latent random variable and the decoder works vice versa. The techniques of VAE have been widely applied to NLP tasks such as text generation (Bowman et al., 2016), machine translation (Zhang et al., 2016), and sequence labeling (Chen et al., 2018).

The CVAE is an extension of the VAE, in which the distribution of a latent variable is explicitly conditioned on certain variables and enables generation processes to be more diverse than a VAE (Li et al., 2018; Zhao et al., 2017b; Shen et al., 2017). The CVAE is trained by maximizing the variational lower bound of the log likelihood.

3 VQAG: Variational Question-Answer Pair Generation Model

3.1 Problem Definition

Our problem is to generate QA pairs from textual contexts. We focus on extractive QA in which an answer is a text span in context. We use c, q, and a to represent the context, question, and answer, respectively. We assume that every QA pair is sampled independently given a context. Thus, the problem is defined as maximizing the conditional log likelihood $\log p(q, a|c)$ averaged over all samples in a dataset.

3.2 Variational Lower Bound with Explicit KL Control

Generating questions and answers from different latent spaces makes sense because multiple questions can be created from a context-answer pair and multiple answer spans can be extracted from a context. Thus, we introduce two independent latent random variables to assign the roles of diversifying AE and QG to z and y, respectively.

VAEs often suffer from *posterior collapse*, where the model learns to ignore latent variables and generates outputs that are almost the same (Bowman et al., 2016). Many approaches have been proposed to mitigate this issue, such as weakening the generators (Bowman et al., 2016; Yang et al., 2017b; Semeniuta et al., 2017), or modifying the objective functions (Tolstikhin et al., 2018; Zhao et al., 2017a; Higgins et al., 2017).

To mitigate this problem, we use a variant of the modified β-VAE (Higgins et al., 2017) proposed by Burgess et al. (2018), which uses two hyperparameters to control the KL terms. Our modified objective function is:

$$\begin{aligned}\mathcal{L} = &\mathbb{E}_{q_\phi(z,y|q,a,c)}[\log p_\theta(q|y, a, c) \\ &+ \log p_\theta(a|z, c)] \\ &- |D_{\mathrm{KL}}(q_\phi(z|a, c)||p_\theta(z|c)) - C_a| \\ &- |D_{\mathrm{KL}}(q_\phi(y|q, c)||p_\theta(y|c)) - C_q|, \end{aligned} \quad (1)$$

where D_{KL} is the KL divergence, θ (ϕ) is the parameters of the generative (inference) model, and $C_a, C_q \geq 0$. See Appendix A for the derivation of the objective. Tuning C_a and C_q was enough to regularize the KL terms in our case (see Appendix B). C_a and C_q can explicitly control the KL values because the KL terms are forced to get closer to these values during training. We mathematically show that the KL control can be interpreted to control the conditional mutual information $I(z; a)$ and $I(y; q)$. This is the major difference between our model and Lee et al. (2020), where $I(q; a)$ is maximized to improve consistency of QA pairs. See Appendix C for the mathematical interpretation.

3.3 Model Architecture

An overview of VQAG is given in Figure 1. We denote c_i, q_i, and a_i as the i-th word in context, question, and answer, respectively. See Appendix D for the details of the implementation.

Embedding and Contextual Embedding Layer First, in the embedding layer, the i-th word, w_i, of a sequence of length L is simultaneously converted into word- and character-level embedding vectors by using a CNN based on Kim (2014). Then, we concatenate the embedding vectors. After that, we pass the embedding vectors to the contextual embedding layer consisting of bidirectional LSTMs (BiLSTM). We obtain $H \in \mathbb{R}^{L \times 2d}$, which is the

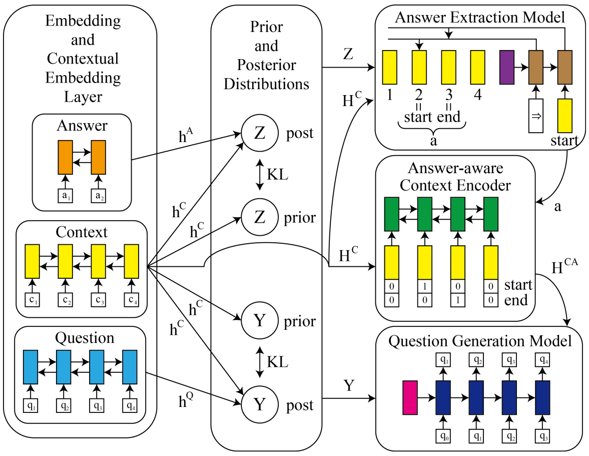

Figure 1: Overview of the model architecture. The latent variables z and y are sampled from the posteriors when computing the variational lower bound and from the priors during generation. See §3.3 for the details.

concatenated outputs from the LSTMs in each direction at each time step, and $h \in \mathbb{R}^{2d}$, which is the concatenated last hidden state vectors in each direction. The superscripts of the outputs H and h shown in Figure 1 indicate where they come from. C, Q, and A denote the context, question, and answer, respectively.

Prior and Posterior Distributions Following Zhao et al. (2017b), we hypothesized that the prior and and posterior distributions of the latent variables follow multivariate Gaussian distributions with diagonal covariance. The mean μ and log variance $\log \sigma^2$ of these prior and posterior distributions of z and y are computed with linear transformation from h^C, h^A, and h^Q. Next, latent variable z and y are obtained using the reparameterization trick (Kingma and Welling, 2013). Then, z and y are passed to the AE and QG models, respectively. z and y are sampled from the posteriors during training and the priors during testing.

Answer Extraction Model We regard AE as two-step autoregressive decoding, i.e., $p(a|c) = p(c_{start}|c)p(c_{end}|c_{start}, c)$, that predicts the start and end positions of an answer span in this order. For AE, we modify a pointer network (Vinyals et al., 2015) to take as input the initial hidden state computed from linear transformation from z, which in the end diversifies AE by learning the mappings from z to a. We use an LSTM as a decoder and compute attention scores over H^C.

Answer-aware Context Encoder To compute answer-aware context information for QG, we use another BiLSTM. We concatenate H^C and one hot vectors of start and end positions of answer, which are fed to the BiLSTM. We obtain $H^{CA} \in \mathbb{R}^{L \times 2d}$, which is the concatenated outputs from the LSTMs in each direction. H^{CA} is used as the source for attention and copying in QG.

Question Generation Model For QG, we modify an LSTM decoder with attention and copying mechanisms to take as input the initial hidden state computed from linear transformation from y, which in the end diversifies QG. At each time step, the probability distribution of generating words from vocabulary $P_v(q_i)$ is computed using attention (Bahdanau et al., 2014), and the probability distributions of copying words (Gulcehre et al., 2016; Gu et al., 2016) from context $P_c(c_j)$ are computed using attention. In parallel, the switching probability p_s is linearly estimated from the hidden state vector. Lastly, we compute the probability of q_i as:

$$p(q_i) = p_s P_v(q_i) + (1 - p_s) \sum_{j:c_j = q_i} P_c(c_j). \quad (2)$$

4 Experiments and Results

4.1 Dataset

We used SQuAD v1.1 (Rajpurkar et al., 2016), a large scale QA dataset consisting of documents collected from Wikipedia and 100k QA pairs created by crowdworkers, as a source dataset for QAG. Answers to questions in SQuAD can be extracted from textual contexts. Since the SQuAD test set has not been released, we use the split of the dataset, SQuAD-Du (Du et al., 2017), where the original training set is split into the training set ($\text{SQuAD}^{\text{Du}}_{\text{train}}$) and the test set ($\text{SQuAD}^{\text{Du}}_{\text{test}}$), and the original development set is used as the dev set ($\text{SQuAD}^{\text{Du}}_{\text{dev}}$). The sizes of $\text{SQuAD}^{\text{Du}}_{\text{train}}$, $\text{SQuAD}^{\text{Du}}_{\text{dev}}$, and $\text{SQuAD}^{\text{Du}}_{\text{test}}$ are 75,722, 10,570, and 11,877, respectively. See Appendix E for the training details of VQAG.

4.2 Answer Extraction

First, we conducted the AE experiment where inputs were the contexts and outputs were a set of multiple answer spans. The objective of this experiment is to measure the diversity and the extent to which our extracted answers cover the ground truths. We also study the effect of C_a in Eq. 1.

	Relevance				Diversity
	Precision		Recall		Dist
	Prop.	Exact	Prop.	Exact	
NER	34.44	19.61	64.60	45.39	30.0k
HarQG	45.96	33.90	41.05	28.37	-
InfoHCVAE	31.59	16.18	78.75	59.32	70.1k
VQAG					
$\mathrm{C_a} = 0$	**58.39**	**47.15**	21.82	16.38	3.1k
$\mathrm{C_a} = 5$	30.16	13.41	**83.13**	**60.88**	71.2k
$\mathrm{C_a} = 20$	21.95	5.75	72.26	42.15	**103.3k**

Table 1: Results of AE on the test set.

	Relevance				Diversity		
	B1-R	ME-R	RL-R	Token	D1	E4	SB4
SemQG	**62.32**	**36.77**	**62.87**	7.0M	15.8k	18.28	91.44
VQAG							
$\mathrm{C_q} = 0$	35.57	18.31	33.92	7.6M	14.4k	17.33	97.61
$\mathrm{C_q} = 5$	44.19	25.84	45.18	11.5M	19.0k	19.71	82.59
$\mathrm{C_q} = 20$	48.19	25.29	48.26	4.9M	**22.4k**	**19.72**	**44.41**

Table 2: Results of answer-aware QG on the test set. One question per input is evaluated in the upper part, while 50 questions per input are evaluated in the lower part to assess their diversity.

Metrics To measure the accuracy of multi-span extraction, we computed Proportional Overlap (Prop.) and Exact Match (Exact) metrics (Breck et al., 2007; Johansson and Moschitti, 2010; Du and Cardie, 2018) for each pair of extracted and ground truth answer spans, and then we report their precision and recall.[2] Prop. is proportional to the amount of overlap between two phrases. Our models extracted 50 answers from each context. To measure the diversity, we defined a Dist score, which is the the total number of distinct context-answer pairs.

Baselines We used three baselines: named entity recognition (NER), Harvesting QG (HarQG) (Du and Cardie, 2018), and InfoHCVAE (Lee et al., 2020). For NER, we used spaCy (Honnibal et al., 2020). For HarQG, we directly copied the scores from Du and Cardie (2018). For InfoHCVAE, we trained the model on the training set, and extracted 50 answers randomly from each context for a fair comparison.

Result Table 1 shows the result. While we tested various values of $\mathrm{C_a}$ ranging from 0 to 100, we only report the selected values here for brevity. When using $\mathrm{C_a}$ larger than 20, the scores did not get improved. Our model with $\mathrm{C_a} = 5$ performed the best in terms of the recall scores, while surpassing the diversity of NER. The highest Dist scores did not occur together with the highest recall scores. When $\mathrm{C_a}$ is 0, the Dist score is fairly low. This implies the posterior collapse issue, though the precision scores are the best. We assert that low precision scores do not necessarily mean poor performance in our experiment because even the original test set does not cover all the valid answer spans.

4.3 Answer-aware Question Generation

We also conducted answer-aware QG experiments where the contexts and ground truth answer spans were the inputs to assess diversity and relevance to the gold questions.

Metrics To evaluate the diversity of the generated questions, our models generated 50 questions from each context-answer pair. We reported the recall scores (denoted as "-R") of BLEU-1 (B1), METEOR (ME), and ROUGE-L (RL) per reference question. We do not report precision scores here because our motivation is to improve diversity. To measure diversity, we reported Dist-1 (D1), Entropy-4 (E4) (Serban et al., 2017; Zhang et al., 2018), and Self-BLEU-4 (SB4) (Zhu et al., 2018).[3]

Baselines We compared our models with SemQG (Zhang and Bansal, 2019).[4] We used diverse beam search (Li et al., 2016b), sampled the top 50 questions per answer from SemQG, and used them to calculate the metrics as the baseline for a fair comparison

Result The results in Table 2 show that our model can improve diversity while degrading the recall scores compared to SemQG. Using $\mathrm{C_q}$ larger than 20 did not lead to improved diversity. More detailed analysis of $\mathrm{C_a}$ and $\mathrm{C_q}$ is provided in Appendix F.

4.4 Synthetic Dataset Construction

We created three synthetic QA datasets, denoted as $\mathcal{D}_{5,5}$, $\mathcal{D}_{20,20}$, and $\mathcal{D}_{5,20}$, using VQAG

[2] We exclude Binary Overlap, which assigns higher scores to systems that extract the entire input context and is not a reliable metric as Breck et al. (2007) discussed.

[3] We computed Dist-1 following the definition of Xu et al. (2018), wherein Dist-1 is the number of distinct unigrams. Dist-1 is often defined as the ratio of distinct unigrams (Li et al., 2016a) but this is not fair when the number of generated sentences differs among models, so we did not use this. SB4 was calculated per 50 questions generated from each input.

[4] We reran the ELMo+QPP&QAP model, which is available at `https://github.com/ZhangShiyue/QGforQA`.

beyoncé ’s vocal range spans four octaves . jody rosen highlights her tone and timbre as particularly distinctive , describing her voice as ” one of the most compelling instruments in popular music ” . while another critic says she is a ” vocal acrobat , being able to sing long and complex melismas and vocal runs effortlessly , and in key . her vocal abilities mean she is identified as the centerpiece of destiny ’s child . the daily mail calls beyoncé ’s voice ” versatile ” , capable of exploring power ballads , soul , rock belting , operatic flourishes , and hip hop . jon pareles of the new york times commented that her voice is ” velvety yet tart , with an insistent flutter and reserves of soul belting ” .

Q: how can one find her vocal abilities in key music ?	A: she is identified as the centerpiece of destiny ’s child
Q: how many octaves spans beyoncé ’s vocal range ?	A: spans four
Q: how many octaves ’s vocal range spans the beyoncé hop vocal range ?	A: four
Q: who commented that her voice is tart yet tart ?	A: jon pareles

Table 3: Heatmap of extracted answer spans and generated samples using our model. The darker the color is, the more often the word is extracted. The phrases surrounded by black boxes are the ground truth answers in SQuAD.

with the different configurations, $(C_a, C_q) = (5,5), (20,20), (5,20)$ respectively. These configurations are chosen based on the recall-based metrics and diversity scores in the AE and QG results.

VQAG generated 50 QA pairs from each paragraph in SQuAD$^{\text{Du}}_{\text{train}}$ to construct each $\mathcal{D}$. It is generally known that VAEs generate diverse but low-quality data unlike GANs. We used heuristics to filter out low-quality generated QA pairs, dropping questions that are longer than 20 words or shorter than 5 words and answers that are longer than 10 words, keeping questions that have at least one interrogative word, and removing n-gram repetition in questions. While some existing works used the BERT QA model or an entailment model as a data filter (Alberti et al., 2019; Zhang and Bansal, 2019; Liu et al., 2020), our heuristics are enough to obtain improvement in the downstream QA task as shown in §4.6. Some samples in our datasets are given in Table 3, showing that the diverse QA pairs are generated. See Appendix G to see how VQAG maps latent variables to QA pairs.

4.5 Human Evaluation

We assess the quality of the synthetic QA pairs by conducting human evaluation on Amazon Mechanical Turk. For human evaluation, we randomly chose 200 samples from synthetic QA pairs generated by Zhang and Bansal (2019) and our model with $(C_a, C_q) = (5,5), (20,20)$ from the paragraphs in SQuAD$^{\text{Du}}_{\text{test}}$. We also chose 100 samples from SQuAD$^{\text{Du}}_{\text{test}}$. In addition to the three items proposed by Liu et al. (2020), we asked annotators if an given answer is important, i.e., it is worth being asked about. We showed the workers a triple (passage, question, answer) and asked them to answer the four questions shown in Table 4. See Appendix H for the details. We report the responses obtained using the majority vote.

Experiments		SemQG	$(C_a, C_q) =$ $(5,5)$	 $(20,20)$	SQuAD
Question is well-formed	No	2.9%	23.1%	27.8%	2.3%
	Understandable	34.5%	16.0%	17.0%	10.5%
	Yes	62.6%	60.9%	55.1%	87.2%
Question is relevant	No	2.5%	9.5%	11.5%	4.0%
	Yes	97.5%	90.5%	88.5%	96.0%
Answer is correct	No	2.8%	28.8%	30.5%	7.5%
	Partially	21.8%	28.1%	26.6%	11.8%
	Yes	75.4%	43.2%	42.9%	80.6%
Answer is important	No	1.5%	10.0%	5.0%	6.0%
	Yes	98.5%	90.0%	95.0%	94.0%

Table 4: Human evaluation of the quality of QA pairs.

According to the results in Table 4, nearly 25% of our questions are not understandable or meaningful, and 30% of our answers are incorrect for the generated questions. This result indicates that our synthetic datasets contain a considerable number of noisy QA pairs in these two aspects. However, 90 % of the generated questions are relevant to the passages, and 90% of the answers extracted by our models are question-worthy. As we will verify in §4.6, our noisy but diverse synthetic datasets are effective in enhancing the QA performance in the in- and out-of-distribution test sets.

4.6 Question Answering

We evaluated QAG methods on the downstream QA task. We evaluated our method on 12 challenge sets in addition to the in-distribution test set.

4.6.1 Baselines

We compared our method with the following baselines.

- **SQuAD$^{\text{Du}}_{\text{train}}$** BERT-base model trained on SQuAD$^{\text{Du}}_{\text{train}}$ without data augmentation.
- **HarQG** (Du and Cardie, 2018) uses neural AE and QG models and generates over one million QA pairs from top ranking Wikipedia articles not included in SQuAD. We used the publicly available dataset.[5]

[5] https://github.com/xinyadu/harvestingQA

- **SemQG** (Zhang and Bansal, 2019) uses reinforcement learning to generate more SQuAD-like questions. We reran the trained model, and generated questions from the same context-answer pairs as HarQG.
- **InfoHCVAE** (Lee et al., 2020) uses a variational QAG model with an information-maximizing term. We trained this model[6] on $\text{SQuAD}^{\text{Du}}_{\text{train}}$, and then generated 50 QA pairs from each context in $\text{SQuAD}^{\text{Du}}_{\text{train}}$ for a fair comparison with VQAG.

4.6.2 Training Details

We trained pretrained BERT-base models (Devlin et al., 2019) on each synthetic dataset, and then fine-tuned it on $\text{SQuAD}^{\text{Du}}_{\text{train}}$. We adopted this procedure following existing data augmentation approach for QA (Dhingra et al., 2018; Zhang and Bansal, 2019). In our study, the order in which our synthetic datasets $\mathcal{D}$ were given to a QA model was tuned on the dev set.

We used the Hugging Face's implementation of BERT (Wolf et al., 2020). We used Adam (Kingma and Ba, 2014) with epsilon as 1e-8 for the optimizer. The batch size was 32. In both the pretraining and fine-tuning procedure, the learning rate decreased linearly from 3e-5 to zero. We conducted the training for one epoch using a synthetic dataset and two epochs using the original training set.

In addition to the performance of *Single* models, we reported the performance of *Ensemble* models, where the output probabilities of three different QA models are simply averaged. In practice, the top 20 candidate answer spans predicted by each QA model were used for the final prediction.

4.6.3 Challenge Sets

We assessed the robustness of the QA models to the following 12 challenge sets, as well as $\text{SQuAD}^{\text{Du}}_{\text{test}}$.

- **NewsQA (News)** (Trischler et al., 2017): 5,166 QA pairs created from CNN articles by crowdworkers, transformed into the SQuAD format following Sen and Saffari (2020).
- **Natural Questions (NQ)** (Kwiatkowski et al., 2019): 2,356 questions from real users for Wikipedia articles. We reframed NQ as extractive QA by using long answers in NQ as contexts following Sen and Saffari (2020).[7]
- **Non-Adversarial Paraphrased Test Set (Para)** (Gan and Ng, 2019): 1,062 questions paraphrased with slight perturbations from SQuAD using a trained paraphrased model.
- **Adversarial Paraphrased Test Set (APara)** (Gan and Ng, 2019): 56 questions manually paraphrased using context words near a confusing answer from SQuAD.
- **Hard Subset (Hard)** (Sugawara et al., 2018): A subset of the SQuAD dev set, which consists of 1,661 questions that require less word matching and more knowledge inference and multiple sentence reasoning.
- **Implications (Imp)** (Ribeiro et al., 2019): 13,371 QA pairs automatically derived from the SQuAD dev set with a linguistic rule-based method.[8]
- **AddSent (Add) & AddOneSent (AddO)** (Jia and Liang, 2017): Adversarial SQuAD dataset created using handcrafted rules designed for fooling a QA model. The sizes of Add and AddO are 3,560 and 1,787, respectively.
- **Quoref (Quo)** (Dasigi et al., 2019): 2,418 questions requiring coreference resolution created by humans. We used the dev set.
- **Natural Machine Translation Noise (MT)** (Ravichander et al., 2021): A subset of NoiseQA, consisting of 1,190 English translated questions produced by Google's commercial translation system from the XQuAD dataset (Artetxe et al., 2020). This creation introduces naturally occurring noise caused by machine translation.
- **Natural Automatic Speech Recognition Noise (ASR)** (Ravichander et al., 2021): Another subset of NoiseQA, consisting of 1,190 questions that include automatic speech recognition error.
- **Natural Keyboard Noise (KB)** (Ravichander et al., 2021): Another subset of NoiseQA, consisting of 1,190 questions that include natural character-level typos introduced by typing questions on a keyboard.

These challenge sets enable us to evaluate the QA models' robustness to other domain corpora, varia-

[6] `https://github.com/seanie12/Info-HCVAE`

[7] We used answerable questions for NewsQA and NQ provided by Sen and Saffari (2020). We did not use the MRQA shared task version as Lee et al. (2020) did.

[8] For example, "Q: *Who died in 1285?* A: *Zhenjin*" is derived from "Q: *When did Zhenjin die?* A: *1285*"

		Challenge Sets												
	Training Data (Size)	$\text{SQuAD}^{\text{Du}}_{\text{test}}$	News	NQ	Quo	Para	APara	Hard	Imp	Add	AddO	MT	ASR	KB
Single	$\text{SQuAD}^{\text{Du}}_{\text{train}}$ (76k)	83.5	49.2	67.7	30.1	85.7	50.2	75.6	64.7	62.9	71.8	79.7	67.5	80.1
	+HarQG (1,205k)	*83.3*	*48.5*	*66.2*	31.3	*85.2*	56.5	*73.0*	*63.5*	65.1	73.1	*78.6*	70.0	80.3
	+SemQG (1,204k)	84.7	50.5	69.8	***34.5***	86.2	51.8	*75.0*	65.1	***66.5***	74.3	*79.5*	71.0	80.7
	+InfoHCVAE (824k)	**84.8**	**51.3**	**71.2**	33.8	85.6	53.3	**77.7**	64.8	66.1	**74.5**	**81.3**	**71.6**	**82.8**
	+VQAG (432k)	84.5	49.2	70.1	32.0	**86.7**	**59.0**	76.1	**66.3**	64.8	73.9	79.9	70.5	81.1
Ensemble	{$\text{SQuAD}^{\text{Du}}_{\text{train}}$}*3	84.2	50.4	69.4	31.3	86.4	53.2	76.6	65.7	63.6	72.6	80.3	68.7	81.2
	{+SemQG}*3	85.5	51.8	71.3	**35.1**	87.5	57.8	78.2	66.5	67.0	75.1	80.8	72.9	82.5
	{+InfoHCVAE}*3	85.3	52.0	**72.2**	34.0	88.0	56.9	**79.0**	65.7	**67.7**	**75.9**	81.4	73.1	**83.2**
	{+VQAG}*3	84.9	50.9	70.1	32.3	**88.1**	**58.6**	77.3	**67.5**	64.9	73.9	80.8	71.2	81.6
	{+Sem,+Info,+V}	**85.8**	**52.1**	72.0	34.2	88.0	55.1	78.8	67.0	66.3	74.7	**82.2**	**73.5**	83.0
If challenge set is known		-	62.9	83.0	66.9	88.6	73.9	-	-	-	-	80.8	75.9	82.6

Table 5: QA performance (F1 score) on $\text{SQuAD}^{\text{Du}}_{\text{test}}$ and the 12 challenge sets. The abbreviations of the challenge sets are explained in §4.6. Curly brackets denote an ensemble of different models (e.g., {+VQAG}*3 denotes the ensemble of three QA models, trained with different random seeds after data augmentation with VQAG). The best scores for each of the *Single* and *Ensemble* models are **boldfaced**. The degraded scores compared to the no data augmentation baseline (the 1st line) are in *red*. Sem: SemQG, Info: InfoHCVAE, V: VQAG.

tions in questions, adversarial examples, and noise that may occur in real-world applications.

4.6.4 Results

The overall results are given in Table 5. First, we discovered that the QA model without data augmentation degraded the performance on the 12 challenge sets, showing a lack of the robustness to the natural and adversarial distribution shifts in contexts, questions, and answers.[9]

With data augmentation using QAG, the in-distribution scores were generally improved, except for HarQG. In the *Single* model setting on the challenge sets, SemQG achieved the best performance on Quo and Add. InfoHCVAE achieved the best performance on News, NQ, Hard, AddO, MT, ASR, and KB. VQAG achieved the best performance on Para, APara, and Imp. These results imply that different QAG methods have different benefits. In the *Ensemble* setting, taking the best of the three, the scores on $\text{SQuAD}^{\text{Du}}_{\text{test}}$, News, MT, and ASR were further improved with {+Sem,+Info,+V}.

We also attached scores that are obtained *if challenge set is known* in Table 5; that is, natural or synthetic samples from the same distributions as the challenge sets are available during training. For News, NQ, and Quo, we trained the BERT-base model on the corresponding training sets, which are annotated by humans. For paraphrased questions (Para, APara) and NoiseQA (MT, ASR, and KB), the scores were taken from Gan and Ng (2019) and Ravichander et al. (2021), respectively. These scores can be considered as the upper bounds. In NoiseQA, the QAG methods consistently improved the scores, even though they were not designed for the noise. This may be because the lack of quality in synthetic datasets, as shown in Table 4, unintentionally improved the robustness to the noise. However, the most significant performance gap ($>$ 30 F1) between the upper bound and the no data augmentation baseline was observed in Quo. This result indicates that a QA model does not acquire coreference resolution from SQuAD, even though approximately 18% of SQuAD questions require coreference resolution (Sugawara et al., 2018). The QAG methods mitigated this gap to some extent, but there is a significant room for improvement.

The improvement in NQ is generally more prominent than that in News. This may be because both SQuAD and NQ contain paragraphs in Wikipedia. Utilizing unlabeled documents in domains such as news articles may improve the generalization to other domains, such as News.

In our experiment, our model and InfoHCVAE improved the scores despite generating QA pairs from only the paragraphs in $\text{SQuAD}^{\text{Du}}_{\text{train}}$, unlike SemQG and HarQG, which generated QA pairs from paragraphs out of SQuAD in Wikipedia. Using paragraphs in and out of $\text{SQuAD}^{\text{Du}}_{\text{train}}$ as the source for QAG may be more effective.

In paraphrased questions (Gan and Ng, 2019), implications (Ribeiro et al., 2019), and NoiseQA

[9] The score on Para—85.7 F1 is degraded when compared to the score on the SQuAD dev set—87.9 F1, which is the source for creating Para. This means the lack of robustness to paraphrased questions.

Training Source (Size)	EM	F1
VQAG (432k)	81.49	88.61
$-\mathcal{D}_{5,5}$ (251k)	81.04	88.39
$-\mathcal{D}_{5,20}$ (113k)	81.00	88.48
$-\mathcal{D}_{20,20}$ (68k)	81.14	88.52

Table 6: Ablation study on $\text{SQuAD}^{\text{Du}}_{\text{dev}}$.

Dataset	what	how	who	which	when	where	why
$\text{SQuAD}^{\text{Du}}_{\text{train}}$	58.3	10.4	10.3	6.7	6.7	4.2	1.5
$\text{SQuAD}^{\text{Du}}_{\text{test}}$	56.5	12.1	11.5	8.6	6.0	3.8	0.8
HarQG	61.3	7.8	13.8	0.7	10.1	5.8	0.5
SemQG	71.1	8.1	12.8	1.3	3.6	2.7	0.2
InfoHCVAE	77.1	6.6	5.0	1.6	5.6	3.3	0.5
VQAG							
$\mathcal{D}_{5,5}$	36.6	54.9	4.9	0.5	0.3	0.5	2.3
$\mathcal{D}_{5,20}$	9.5	35.5	3.6	49.2	1.2	0.9	0.0
$\mathcal{D}_{20,20}$	28.2	36.7	6.3	23.2	0.2	1.6	3.9

Table 7: Percentages (%) of each question type in each dataset. The largest number in each line is underlined.

(Ravichander et al., 2021), augment questions that are similar to the corresponding challenge sets, that is, generating paraphrases, implications, and questions including the noise, successfully improved the robustness to these perturbations. While these methods slightly degraded or maintained the in-distribution score, we showed that QAG methods are less likely to exhibit a trade-off between the in- and out-of-distribution accuracies. Notably, VQAG did not degrade the scores on all the 12 challenge sets while improving the in-distribution score. In contrast, SemQG degraded the scores on Hard and MT, and InfoHCVAE degraded the score on Para. This property of VQAG may be because it can significantly improve the diversity by combining different configurations of the KL control.

Moreover, the size of synthetic dataset created by VQAG was the smallest among the QAG methods. If the diversity is assured sufficiently, significantly increasing the quantity may not be necessary. In Add and AddO, we showed that the QAG methods consistently improved adversarial robustness, which has not been studied in the QAG literature.

4.6.5 Analysis

To assess the usefulness of each dataset $\mathcal{D}$ in VQAG, we conducted an ablation study. As shown in Table 6, each dataset $\mathcal{D}$ has meaningful effect on the performance. This result implies that creating more synthetic datasets using different configurations may further improve the performance.

To understand the differences in each dataset in terms of diversity, we conducted a simple analysis on the question type. As shown in Table 7, VQAG with different configurations corresponds to different distributions of question types, while more than 50% of the questions in the other datasets contain "what". Among the QAG methods, this point is unique to VQAG.

5 Discussion and Conclusion

We presented a variational QAG model, incorporating two independent latent random variables. We showed that an explicit KL control can enable our model to significantly improve the diversity of QA pairs. Our synthetic datasets were shown to be noisy in terms of the grammaticality and answerability of questions, but effective in improving the QA performance in the in-distribution test set and the 12 challenge sets. While out synthetic datasets are noisy, they may unintentionally improve the robustness to the noise that can occur in real applications. However, we should pay attention to the negative effect of using our noisy dataset. For example, the lack of the answerability of our synthetic questions may lead to the poor performance in handling unanswerable questions, such as SQuAD v2.0. Moreover, QAG methods led to improvements in most of the 12 challenge sets while being agnostic to the target distributions during training. We need to pursue such a target-unaware method to improve the robustness of QA models, because it is quite difficult for developers to know the types of questions a QA model cannot handle in advance.

In summary, our experimental results showed that the diversity of QA datasets plays a non-negligible role in improving its robustness, which can be boosted with QAG. We will consider using unlabeled documents in other domains to further improve the robustness to other domain corpora in our future study.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. We would also like to thank the members of Aizawa Lab for their helpful discussions. This work was supported by NEDO SIP-2 "Big-data and AI-enabled Cyberspace Technologies" and JSPS KAKENHI Grant Number 20K23335.

References

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

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. 2018. Variational attention for sequence-to-sequence models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1672–1682. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2683–2688, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in *beta*-VAE. *arXiv preprint arXiv:1804.03599*.

Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2018. Variational sequential labelers for semi-supervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 215–226, Brussels, Belgium. Association for Computational Linguistics.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

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

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)*, pages 582–587, New Orleans, Louisiana. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32, pages 13042–13054. Curran Associates, Inc.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917. Association for Computational Linguistics.

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.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149. Association for Computational Linguistics.

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.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Richard Johansson and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 67–76, Uppsala, Sweden. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Junmo Kang, Haritz Puerto San Roman, and sunghyon myaeng. 2019. Let me know what to ask: Interrogative-word-aware question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 163–171, Hong Kong, China. Association for Computational Linguistics.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6602–6609.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1121, Online. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

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*, 7:452–466.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China. Association for Computational Linguistics.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 208–224, Online. Association for Computational Linguistics.

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

Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R. Lyu. 2019. Improving question generation with to the point context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3214–3224, Hong Kong, China. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

Juntao Li, Yan Song, Haisong Zhang, Dongmin Chen, Shuming Shi, Dongyan Zhao, and Rui Yan. 2018. Generating classical chinese poems via conditional variational autoencoder and adversarial training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3890–3900. Association for Computational Linguistics.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114, Sofia, Bulgaria. Association for Computational Linguistics.

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

Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. 2019. Learning to generate questions by learningwhat not to generate. In *The World Wide Web Conference*, WWW '19, page 1106–1118, New York, NY, USA. Association for Computing Machinery.

Jack Mostow and Wei Chen. 2009. Generating instruction automatically for the reading strategy of self-questioning. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, pages 465–472, NLD. IOS Press.

Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2019. Let's ask again: Refine network for automatic question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3312–3321, Hong Kong, China. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Jiazuo Qiu and Deyi Xiong. 2019. Generating highly relevant questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5982–5986, Hong Kong, China. Association for Computational Linguistics.

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

Abhilasha Ravichander, Siddharth Dalmia, Maria Ryskina, Florian Metze, Eduard Hovy, and Alan W Black. 2021. NoiseQA: Challenge set evaluation for user-centric question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2976–2992, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? Evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy. Association for Computational Linguistics.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637, Copenhagen, Denmark. Association for Computational Linguistics.

Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online. Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3295–3301. AAAI Press.

Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A conditional variational framework for dialog generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 504–509, Vancouver, Canada. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. 2018. Neural models for key phrase extraction and question generation. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 78–88. Association for Computational Linguistics.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.

Md Arafat Sultan, Shubham Chandel, Ramón Fernandez Astudillo, and Vittorio Castelli. 2020. On the importance of diversity in question generation for QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5651–5656, Online. 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. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2018. Wasserstein autoencoders. In *International Conference on Learning Representations*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451, San Diego, California. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2017. Machine comprehension using match-LSTM and answer pointer. In *International Conference on Learning Representations*.

Siyuan Wang, Zhongyu Wei, Zhihao Fan, Yang Liu, and Xuanjing Huang. 2019. A multi-agent communication framework for question-worthy phrase extraction and question generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7168–7175.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018. Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3940–3949, Brussels, Belgium. Association for Computational Linguistics.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017a. 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)*, pages 1040–1050. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017b. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3881–3890, International Convention Centre, Sydney, Australia. PMLR.

Kaichun Yao, Libo Zhang, Tiejian Luo, Lili Tao, and Yanjun Wu. 2018. Teaching machines to ask questions. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4546–4552. International Joint Conferences on Artificial Intelligence Organization.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas. Association for Computational Linguistics.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509, Hong Kong, China. Association for Computational Linguistics.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1810–1820. Curran Associates, Inc.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017a. InfoVAE: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017b. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2018. Neural question generation from text: A preliminary study. In *Natural Language Processing and Chinese Computing*, pages 662–671, Cham. Springer International Publishing.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019. Question-type driven question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6031–6036, Hong Kong, China. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery.

A Derivations of the Variational Lower Bound

The variational lower bound, Eq. 1, without the KL control is derived as follows:

$$
\begin{aligned}
&\log p_\theta(q, a|c) \\
&= \mathbb{E}_{z,y\sim q_\phi(z,y|q,a,c)}\left[\log p_\theta(q, a|c)\right] \\
&= \mathbb{E}_{z,y}\left[\log \frac{p_\theta(q, a|z, y, c)p_\theta(z, y|c)}{p_\theta(z, y|q, a, c)}\right] \\
&= \mathbb{E}_{z,y}\left[\log \frac{p_\theta(q, a|z, y, c)p_\theta(z, y|c)}{p_\theta(z, y|q, a, c)}\right. \\
&\qquad \left. + \log \frac{q_\phi(z, y|q, a, c)}{q_\phi(z, y|q, a, c)}\right] \\
&= \mathbb{E}_{z,y}\left[\log \frac{p_\theta(q|y, a, c)p_\theta(y|c)}{p_\theta(y|q, c)}\right. \\
&\qquad + \log \frac{p_\theta(a|z, c)p_\theta(z|c)}{p_\theta(z|a, c)} \\
&\qquad \left. + \log \frac{q_\phi(y|q, c)}{q_\phi(y|q, c)} + \log \frac{q_\phi(z|a, c)}{q_\phi(z|a, c)}\right] \\
&= \mathbb{E}_{z,y}\left[\log p_\theta(q|y, a, c) + \log p_\theta(a|z, c)\right. \\
&\qquad + \log \frac{p_\theta(y|c)}{q_\phi(y|q, c)} + \log \frac{q_\phi(y|q, c)}{p_\theta(y|q, c)} \\
&\qquad \left. + \log \frac{p_\theta(z|c)}{q_\phi(z|a, c)} + \log \frac{q_\phi(z|a, c)}{p_\theta(z|a, c)}\right] \\
&= \mathbb{E}_{z,y}\left[\log p_\theta(q|y, a, c) + \log p_\theta(a|z, c)\right] \\
&\qquad - D_{\mathrm{KL}}(q_\phi(y|q, c)||p_\theta(y|c)) \\
&\qquad + D_{\mathrm{KL}}(q_\phi(y|q, c)||p_\theta(y|q, c)) \\
&\qquad - D_{\mathrm{KL}}(q_\phi(z|a, c)||p_\theta(z|c)) \\
&\qquad + D_{\mathrm{KL}}(q_\phi(z|a, c)||p_\theta(z|a, c)) \\
&\geq \mathbb{E}_{z,y}\left[\log p_\theta(q|y, a, c) + \log p_\theta(a|z, c)\right] \\
&\qquad - D_{\mathrm{KL}}(q_\phi(y|q, c)||p_\theta(y|c)) \\
&\qquad - D_{\mathrm{KL}}(q_\phi(z|a, c)||p_\theta(z|c)).
\end{aligned}
$$

B Distribution Modeling Capacity

We originally developed a QA pair modeling task to evaluate and compare QA pair generation models. We compared models based on the probability they assigned to the ground truth QA pairs. We used the negative log likelihood (NLL) of QA pairs as the metric, namely, $-\log p(q, a|c)$. Since variational models can not directly compute NLL, we estimate NLL with importance sampling. We also estimate each term in decomposed NLL, i.e.,$\mathrm{NLL_a} = -\log p(a|c)$ and $\mathrm{NLL_q} = -\log p(q|a, c)$. The better a model performs in this task, the better it fits the test set. As a baseline, to assess the effect of incorporating latent random variables, we implemented a pipeline model similar to Subramanian et al. (2018) using a deterministic pointer network.

Result Table 8 shows the result of QA pair modeling. First, our models with C = 0 are superior to the pipeline model, which means that introducing latent random variables aid QA pair modeling capacity. However, the KL terms converge to zero with C = 0. When we set $C > 0$, KL values are greater than 0, which implies that latent variables have non-trivial information about questions and answers. Also, we observe that the target value of KL C can control the KL values, showing the potential to avoid the posterior collapse issue.

	NLL	NLL_a	NLL_q	D_{KL_z}	D_{KL_y}
Pipeline	36.26	3.99	32.50	-	-
VQAG					
C = 0	**34.46**	4.46	30.00	0.027	0.036
C = 5	37.00	5.15	31.51	4.862	4.745
C = 20	59.66	14.38	43.56	17.821	17.038
C = 100	199.43	81.01	112.37	92.342	91.635

Table 8: QA pair modeling capacity measured on the test set. We used the same value C for the target values of KL C_a and C_q for simplicity. NLL: negative log likelihood of QA pairs. NLL_a (NLL_q): NLL of answers (questions). D_{KL_z} and D_{KL_y} are Kullback–Leibler divergence in Ineq 1. NLL for our models are estimated with importance sampling using 300 samples.

C Information Theoretic Interpretation of the KL control

When training our models, we miximized the variational lower bound in Ineq. 1 is averaged over the training samples. In other words, the expectation with respect to the data distribution is maximized. In the ideal case, the approximated posterior $q_\phi(z|a, c)$ is equal to the true posterior $p_\theta(z|a, c)$. Then, the expectation of the KL terms with respect to the data distribution is equivalent to the conditional mutual information $I(a, y|c)$.

Mathematically, when the approximated posterior q_ϕ is equal to the true posterior p_θ, the expectation of the KL terms in Eq. 1 with respect to the data distribution is:

$$
\begin{aligned}
&\mathbb{E}_{p(q,a,c)}[D_{\mathrm{KL}}(p(z|a, c)||p(z|c))] \\
&\quad = \sum_{a,c} p(a, c) \sum_{z} p(z|a, c) \log \frac{p(z|a, c)}{p(z|c)} \\
&\quad = \sum_{a,c,z} p(a, c, z) \log \frac{p(a, z|c)}{p(z|c)p(a|c)} \\
&\quad = I(a, z|c).
\end{aligned}
$$

Thus, controlling the KL terms is equivalent to control the conditional mutual information. The same is true for question q.

D Model Architechture

Prior and Posterior Distribution Following Zhao et al. (2017b), we hypothesized that the prior and posterior distributions of the latent variables follow multivariate Gaussian distributions with diagonal covariance. The distributions are described as follows:

$$z|a,c \sim \mathcal{N}(\mu_{post_Z}, diag(\sigma^2_{post_Z})) \tag{3}$$

$$z|c \sim \mathcal{N}(\mu_{prior_Z}, diag(\sigma^2_{prior_Z})) \tag{4}$$

$$y|q,c \sim \mathcal{N}(\mu_{post_Y}, diag(\sigma^2_{post_Y})) \tag{5}$$

$$y|c \sim \mathcal{N}(\mu_{prior_Y}, diag(\sigma^2_{prior_Y}). \tag{6}$$

The prior and posterior distributions of the latent variables, z and y, are computed as follows:

$$\begin{bmatrix} \mu_{post_Z} \\ \log(\sigma^2_{post_Z}) \end{bmatrix} = W_{post_Z} \begin{bmatrix} h^C \\ h^A \end{bmatrix} + b_{post_Z} \tag{7}$$

$$\begin{bmatrix} \mu_{prior_Z} \\ \log(\sigma^2_{prior_Z}) \end{bmatrix} = W_{prior_Z} h^C + b_{prior_Z} \tag{8}$$

$$\begin{bmatrix} \mu_{post_Y} \\ \log(\sigma^2_{post_Y}) \end{bmatrix} = W_{post_Y} \begin{bmatrix} h^C \\ h^Q \end{bmatrix} + b_{post_Y} \tag{9}$$

$$\begin{bmatrix} \mu_{prior_Y} \\ \log(\sigma^2_{prior_Y}) \end{bmatrix} = W_{prior_Y} h^C + b_{prior_Y}. \tag{10}$$

Then, latent variable z (and y) is obtained using the reparameterization trick (Kingma and Welling, 2013): $z = \mu + \sigma \odot \epsilon$, where $\odot$ represents the Hadamard product, and $\epsilon \sim \mathcal{N}(0, I)$. Then, z and y is passed to the AE and QG models, respectively.

Answer Extraction Model We regard answer extraction as two-step sequential decoding, i.e.,

$$p(a|c) = p(c_{end}|c_{start}, c)p(c_{start}|c), \tag{11}$$

which predicts the start and end positions of an answer span in this order. For AE, we modify a pointer network (Vinyals et al., 2015) to take into account the initial hidden state $h_0^{AE} = W_1 z + b_1$, which in the end diversify AE by learning the mappings from z to a. The decoding process is as follows:

$$h_i^{IN} = \begin{cases} e(\Rightarrow) & \text{if } i = 1 \\ H^C_{t_{i-1}} & \text{if } i = 2 \end{cases} \tag{12}$$

$$h_i^{AE} = \text{LSTM}(h_{i-1}^{AE}, h_i^{IN}) \tag{13}$$

$$u_{ij}^{AE} = (v^{AE})^T \tanh(W_2 H_j^C + W_3 h_i^{AE} + b_2) \tag{14}$$

$$p(c_{t_i}|c_{t_{i-1}}, c) = \text{softmax}(u_i) \tag{15}$$

where $1 \leq i \leq 2$, $1 \leq j \leq L_C$, h_i^{AE} is the hidden state vector of the LSTM, h_i^{IN} is the i-th input, t_i denotes the start (i=1) or end (i=2) positions in c, and v, W_n and b_n are learnable parameters. We learn the embedding of the special token "$\Rightarrow$" as the initial input h_1^{IN}.

When we used the embedding vector e_{t_i} as h_{i+1}^{IN}, instead of $H^C_{t_i}$, following Subramanian et al. (2018), we observed that the extracted spans tended to be long and unreasonable. We assume that this is because the decoder cannot get the positional information from the input in each step.

Question Generation Model For QG, we modify an LSTM decoder with attention and copying mechanisms to take the initial hidden state $h_0^{QG} = W_4 y + b_3$ as input to diversify QG. In detail, at each time step, the probability distribution of generating words from vocabulary using attention (Bahdanau et al., 2014) is computed as:

$$h_i^{QG} = \text{LSTM}(h_{i-1}^{QG}, q_{t-1}) \tag{16}$$

$$u_{ij}^{att} = (v^{att})^T \tanh(W_5 h_i^{QG} + W_6 H_j^{CA} + b_4) \tag{17}$$

$$a_i^{att} = \text{softmax}(u_i^{att}) \tag{18}$$

$$\hat{h}_i = \textstyle\sum_j a_{ij}^{att} H_j^{CA} \tag{19}$$

$$\tilde{h}_i = \tanh(W_7([\hat{h}_i; h_i^{QG}] + b_5)) \tag{20}$$

$$P_{vocab} = \text{softmax}(W_8(\tilde{h}_i) + b_6), \tag{21}$$

and the probability distributions of copying (Gulcehre et al., 2016; Gu et al., 2016) from context are computed as:

$$u_{ij}^{copy} = (v^{copy})^T \tanh(W_9 h_i^{QG} + W_{10} H_j^{CA} + b_7) \tag{22}$$

$$a_i^{copy} = \text{softmax}(u_i^{copy}) \tag{23}$$

Accordingly, the probability of outputting q_i is:

$$p_g = \sigma(W_{11} h_i^{QG}) \tag{24}$$

$$p(q_i|q_{1:i-1}, a, c) \tag{25}$$

$$= p_g P_{vocab}(q_i) + (1 - p_g) \textstyle\sum_{j:c_j=q_i} a_{ij}^{copy} \tag{26}$$

where σ is the sigmoid function.

E Training Details

We use pretrained GloVe (Pennington et al., 2014) vectors with 300 dimensions and freeze them during training. The pretrained word embeddings were shared by the input layer of the context encoder, the input and output layers of the question decoder. The vocabulary has most frequent 45k words in our training set. The dimension of character-level embedding vectors is 32. The number of windows is 100. The dimension of hidden vectors is 300. The dimension of latent variables is 200. Any LSTMs used in this paper has one layer. We used Adam (Kingma and Ba, 2014) for optimization with initial learning rate 0.001. All the parameters were initialized with Xavier Initialization (Glorot and Bengio, 2010). Models were trained for 16 epochs with a batch size of 32. We used a dropout (Srivastava et al., 2014) rate of 0.2 for all the LSTM layers and attention modules.

F Answer Extraction and Question Generation

Tables 9 and 10 show the detailed results of AE and QG. Various values of $\mathrm{C_a}$ and $\mathrm{C_q}$ are explored.

	Relevance				Diversity
	Precision		Recall		Dist
	Prop.	Exact	Prop.	Exact	
NER	34.44	19.61	64.60	45.39	30.0k
BiLSTM-CRF	45.96	33.90	41.05	28.37	-
InfoHCVAE	31.59	16.18	78.75	59.32	70.1k
VQAG					
$\mathrm{C_a} = 0$	**58.39**	**47.15**	21.82	16.38	3.1k
$\mathrm{C_a} = 3$	34.09	19.22	78.94	59.09	47.5k
$\mathrm{C_a} = 5$	30.16	13.41	**83.13**	**60.88**	71.2k
$\mathrm{C_a} = 10$	26.17	8.83	79.70	53.02	92.3k
$\mathrm{C_a} = 15$	22.42	6.11	76.18	44.80	99.9k
$\mathrm{C_a} = 20$	21.95	5.75	72.26	42.15	**103.3k**
$\mathrm{C_a} = 25$	21.60	5.37	71.55	40.48	101.6k
$\mathrm{C_a} = 30$	23.88	6.75	74.08	44.59	99.5k
$\mathrm{C_a} = 40$	24.58	7.90	74.86	43.33	88.1k
$\mathrm{C_a} = 50$	25.05	7.83	76.56	44.67	88.9k
$\mathrm{C_a} = 100$	23.32	7.48	71.74	39.70	84.6k

Table 9: Detailed results of AE on the test set.

G Latent Interpolation

Table 11 shows the latent interpolation between two ground-truth QA pairs using VQAG with $(C_a, C_q) = (5, 20)$. This result shows that z controls answer and y controls question.

H Human Evaluation

We conducted human evaluation to assess the quality of QA pairs by asking the following questions.

1. **Is the question well-formed in itself?** The workers are asked to select *yes* if a given question is both grammatical and meaningful. The workers select *understandable* if a question is not grammatical but meaningful.
2. **Is the question relevant to the passage?** This is to check whether a question is relevant to the content of a passage.
3. **Is the answer a correct answer to the question?** If a given answer partially overlaps with the true answer in a passage, the workers select *partially*.
4. **Is the meaning of the answer in itself related to the main topic of the passage?** This is to check the importance of an answer. We designed this question to assess the question-worthiness of an answer.

Each triple is evaluated by three crowdworkers. Each task costs 0.08 USD.

	Relevance							Diversity			
	B1	B2	B3	B4	ME	RL	Token	D1	D2	E4	SB4
Zhang and Bansal (2019)	48.59	32.83	24.21	18.40	24.86	46.66	133.8k	10.2k	46.4k	15.78	-

	B1-R	B2-R	B3-R	B4-R	ME-R	RL-R	Token	D1	D2	E4	SB4
Zhang and Bansal (2019)	**62.32**	**47.77**	**37.96**	**30.05**	**36.77**	**62.87**	7.0M	15.8k	218.9k	18.28	91.44
VQAG											
$C_q = 0$	35.57	18.75	10.79	6.35	18.31	33.92	7.6M	14.4k	155.3k	17.33	97.61
$C_q = 3$	44.05	26.74	16.08	9.26	24.61	44.10	9.0M	17.8k	394.2k	19.14	85.88
$C_q = 5$	44.19	27.09	16.33	9.71	25.84	45.18	11.5M	19.0k	481.1k	19.71	82.59
$C_q = 10$	44.00	27.15	16.78	10.24	25.64	44.78	10.2M	18.8k	461.5k	19.69	80.39
$C_q = 15$	45.23	27.91	16.67	10.11	26.12	45.41	11.3M	19.5k	381.5k	19.40	84.56
$C_q = 20$	48.19	32.87	22.96	14.94	25.29	48.26	4.9M	22.4k	549.2k	19.72	44.41
$C_q = 25$	47.20	31.16	21.15	13.66	25.30	45.97	6.8M	22.3k	706.9k	**20.34**	47.00
$C_q = 30$	47.96	31.69	21.26	13.83	24.95	47.07	7.3M	**22.9k**	**732.8k**	18.54	50.32
$C_q = 40$	46.31	31.29	21.52	13.94	23.73	46.46	5.4M	21.0k	487.8k	19.39	55.95
$C_q = 50$	43.92	25.95	15.54	9.61	23.61	43.18	10.8M	22.2k	527.2k	19.29	73.78
$C_q = 100$	35.22	19.88	13.25	9.20	22.27	37.55	8.2M	22.1k	508.8k	19.74	**44.22**

Table 10: Detailed results of answer-aware QG on the test set. Paragraph-level contexts and answer spans are used as input. The baseline model is ELMo+QPP&QAP (Zhang and Bansal, 2019) with diverse beam search (Li et al., 2016b) with a beam size 50. Bn: BLEU-n, ME: METEOR, RL: ROUGE-L, Token: the total number of the generated words, Dn: Dist-n, E4: Ent-4 (entropy of 4-grams), SB4: Self-BLEU-4. "-R" represents recall. (e.g. B1-R is the recall of B1.) One question per answer-context pair is evaluated in the upper part, while 50 questions per answer-context pair are evaluated in the lower part to assess their diversity.

	z_1	z_2	z_3	z_4	z_5
y_1	**in what city and state did beyonce grow up ?—houston , texas**	how do competitions performed a child child ?—dancing	the american singer born what american singer ?—songwriter	how did beyoncé dobruja to ?—dangerously in love	how did beyoncé album album ?—dangerously in love
y_2	the album born and raised ?—houston , texas	how do competitions enovid ?—dancing	how is actress - carter ?—songwriter	how did beyoncé 's album album ?—dangerously in love	how did beyoncé album album ?—dangerously in love
y_3	the album born and raised ?—houston , texas	how do competitions performed a child child ?—dancing	the american singer born what american singer ?—songwriter	how did beyoncé dobruja to ?—dangerously in love	how did beyoncé dobruja to ?—dangerously in love
y_4	the album born and raised ?—houston , texas	how many competitions does texas child perform ?—dancing	the american singer born what american singer ?—songwriter	how did beyoncé dobruja to ?—dangerously in love	how did beyoncé dobruja to ?—dangerously in love
y_5	the album born and raised ?—houston , texas	how many competitions did texas child perform ?—dancing	the american singer born what american singer ?—songwriter	how did beyoncé dobruja to ?—dangerously in love	**what was the name of beyoncé 's first solo album ?—dangerously in love**

Table 11: Latent interpolation with VQAG with $(C_a, C_q) = (5, 20)$. The samples in the upper left and lower right are the ground truth QA pairs from the same paragraph of SQuAD. The linearly interpolated samples show how our generative model learns mapping from latent space to QA pairs.