Variational Question-Answer Pair Generation for Machine Reading Comprehension

Kazutoshi Shinoda[1,2] **and Akiko Aizawa**[1,2]
[1] The University of Tokyo
[2] National Institute of Informatics
shinoda@is.s.u-tokyo.ac.jp
aizawa@nii.ac.jp

Abstract

We present a deep generative model of question-answer (QA) pairs for machine reading comprehension. We introduce two independent latent random variables into our model in order to diversify answers and questions separately. We also study the effect of explicitly controlling the KL term in the variational lower bound in order to avoid the "posterior collapse" issue, where the model ignores latent variables and generates QA pairs that are almost the same. Our experiments on SQuAD v1.1 showed that variational methods can aid QA pair modeling capacity, and that the controlled KL term can significantly improve diversity while generating high-quality questions and answers comparable to those of the existing systems.

Context:
... Their hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".
Question-answer pairs:
What album made her a worldwide known artist? — Dangerously in Love What was the first album Beyoncé released as a solo artist? — Dangerously in Love What was the name of Beyoncé's first solo album? — Dangerously in Love

Table 1: Example of QA pairs with context in SQuAD v1.1 (Rajpurkar et al., 2016). Underlined text spans in the context are used as the gold answers. The listed QA pairs show the case in which multiple questions can be created from a single context-answer pair.

1 Introduction

Machine reading comprehension has gained much 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). To build such systems, a substantial number of question-answer (QA) pairs are needed to train neural network based models. However, the creation of QA pairs from unlabeled documents requires considerable manual effort. To alleviate this problem, there has been a resurgence of work on automatic QA pair generation for data augmentation (Yang et al., 2017a; Du and Cardie, 2018; Subramanian et al., 2018; Alberti et al., 2019; Wang et al., 2019).

When the answers are text spans in a given paragraph, QA pair generation systems have generally used a pipeline of answer extraction (AE) and question generation (QG) models. QG aims to generate questions from each paragraph or sentence. Du et al. (2017) first used sequence-to-sequence models for QG and improved the quality, replacing a rule-based method (Heilman and Smith, 2010). Following works used answers as additional input and showed that answers aid quality of QG (Zhou et al., 2018; Kim et al., 2018; Zhao et al., 2018). Since answers are not available in the real case, AE has been studied in addition to QG. AE aims to extract from documents *question-worthy* phrases, which are defined by Subramanian et al. (2018) and Wang et al. (2019) as phrases that are worth being asked about. Subramanian et al. (2018) and Kumar et al. (2018) proposed to extract answer candidates from documents and to generate questions from documents and the extracted answers. Similarly, Du and Cardie (2018) proposed to generate QA pairs such that requires coreference resolution. Moreover, Alberti et al. (2019) presented QA pair generation with roundtrip consistency that filters out unanswerable QA pairs using BERT (Devlin et al., 2019).

However, to the best of our knowledge, the diversity of QA pairs has been less studied. For QG,

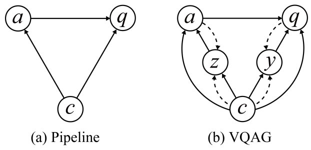

Figure 1: Graphical models of a pipeline model (a) and our Variational Question-Answer Pair Generative model (VQAG) (b). (c: context, a: answer, q: question, z and y: latent variables, **solid**: generative model, **dashed**: inference model)

a few studies focused on diversity (Yao et al., 2018; Bahuleyan et al., 2018). Namely, existing QA pair generation systems can only extract a fixed set of answer spans from each document. Since answers are important features for QG, the lack of diversity in answers should lead to the lack of diversity in questions. Here, we specifically focus on QA pair generation where AE and QG are distinctive stochastic processes that generate diverse outputs. For example, as shown in Table 1, multiple answer candidates such as "2003" and "Dangerously in Love" can be extracted from the context about Beyoncé, and multiple questions can be created from the answer "Dangerously in Love".

It is known that using a variational autoencoder (VAE) (Kingma and Welling, 2013) can diversify the generated text and generate unseen sentences from latent space (Bowman et al., 2016). Moreover, a conditional VAE (CVAE) can generate not only diverse sentences but also condition them on additional variables (Zhao et al., 2017). Here, we conjecture that the CVAE framework may be suitable for QA pair generation conditioned on context. Therefore, we propose a variational QA pair generative model (VQAG). As shown in Figure 1, we introduce two independent latent random variables into our VQAG to model the two one-to-many problems, AE and QG, enabling us to diversify AE and QG separately. We also study the effect of controlling the KL term in the variational lowerbound by introducing hyperparameters to mitigate the posterior collapse issue, where the model ignores latent variables and generate outputs that are almost the same.

We conducted experiments on three tasks, i.e., QA pair modeling, answer extraction, and answer-aware question generation, using SQuAD v1.1. QA pair modeling is our newly developed task that enables us to assess the distribution modeling capacity of QA pair generative models. Our qualitative analysis reveals that our model can generate reasonable QA pairs that are not close to the ground truths.

Contributions Our main contributions are threefold: (1) We propose a Variational Question-Answer Pair Generative model including two independent latent random variables for modeling the diversity of AE and QG separately. To the best of our knowledge, our work is the first to introduce variational methods for both AE and QG jointly. (2) We develop the QA pair modeling task and show that our variational model achieves better modeling capacity than a non-stochastic model in terms of the negative log likelihood. (3) We show that explicitly controlling the KL term in the variational lowerbound objective can avoid the posterior collapse issue. Our model with the controlled KL value significantly improve diversity while generating high-quality questions and answers comparable or superior to those of the existing systems for AE and QG.

2 Related Work

2.1 Answer Extraction

Answer extraction (AE) can be performed in mainly three ways, i.e., 1) using linguistic knowledge, 2) sequence labeling, and 3) using a pointer network.

Yang et al. (2017a) extracted candidate phrases using rule-based methods such as part-of-speech tagger, a simple constituency parser, and named entity recognizer (NER). However, in the SQuAD dataset, not all the named entities, noun phrases, verb phrases, adjectives, or clauses, are used as gold answer spans. So, 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. Subramanian et al. (2018) treated the positions of answers as a sequence and used a pointer network (Vinyals et al., 2015). Du and Cardie (2018) framed the AE problem as a sequence labeling task and used BiLSTM-CRF (Huang et al., 2015) with NER features as additional inputs. Wang et al. (2019) used a pointer network and Match-LSTM (Wang and Jiang, 2016, 2017) to interact with the question generation module. Alberti et al. (2019) made use of pretrained BERT (Devlin et al., 2019) for AE.

Note that these current AE models are deterministic, i.e., their output is static when the input is fixed. As far as we know, our work is the first to introduce a pointer network incorporating a latent random variable. In this paper, we assume that the answer spans used in the SQuAD dataset are question-worthy, but there should be question-worthy phrases not used as the gold answer spans in the dataset.

2.2 Question Generation

Traditionally, Question Generation (QG) was studied using rule-based methods (Mostow and Chen, 2009; Heilman and Smith, 2010; Lindberg et al., 2013; Labutov et al., 2015) These rule-based methods use only the syntactic roles of words.

Since Du et al. (2017) proposed a neural sequence-to-sequence model (Sutskever et al., 2014) for QG and improved its BLEU scores compared to rule-based methods, neural models that take context and answer as inputs has 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 answer-context pairs (Zhou et al., 2018; Song et al., 2018; Zhao et al., 2018; Sun et al., 2018; Kim et al., 2018; Harrison and Walker, 2018; Liu et al., 2019; Qiu and Xiong, 2019; Zhang and Bansal, 2019; Scialom et al., 2019). These works showed the importance of answers as input features for question generation. Other works studied predicting question types (Zhou et al., 2019; Kang et al., 2019), modeling structured answer-relevent 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., 2018). Dong et al. (2019) used a pretrained language model. While the above QG models do not handle cases in which multiple questions can be created from a single context-answer pair, the diversity of questions has been tackled using variational attention (Bahuleyan et al., 2018) or the CVAE (Yao et al., 2018).

Our work is different from these works in that we study QA pair generation by introducing variational methods into both AE and QG and that we evaluate diversity and modeling capacity of our model.

Further, constructing better QA pair generative models need to be constructed for not only data augmentation but also directly applying them to question answering. Lewis and Fan (2019) proposed to perform question answering tasks by reformulating them as $a = \operatorname{argmax}_a p(q,a|c) = \operatorname{argmax}_a p(q|a,c)p(a|c)$, and showed that the reformulation helped to mitigate the superficial understanding problems of machine reading comprehension (Weissenborn et al., 2017).

3 VQAG: Variational Question-Answer Pair Generative model

3.1 Background: Conditional Variational Autoencoder

The VAE (Kingma and Welling, 2013) is a popular deep generative model. It consists of a neural encoder (inference model) and a decoder (generative model). The encoder learns to map from an observed variable, x, to a latent variable, z, and the decoder works vice versa. *Neural approximation* and *reparameterization* techniques of VAE have been 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 prior 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., 2017; Shen et al., 2017). The CVAE is trained by maximizing the following variational lower bound:

$$\log p_\theta(x|c) \geq \mathbb{E}_{z\sim q_\phi(z|x,c)}[\log p_\theta(x|z,c)] - D_{\mathrm{KL}}(q_\phi(z|x,c)||p_\theta(z|c)) \quad (1)$$

where D_{KL} means the Kullback-Leibler divergence, c is the condition, and θ (ϕ) is parameters of the generative (inference) model parameterized by neural networks.

3.2 Problem Definition

Here, the problem is to generate QA pairs from contexts (documents). We focus on the case in which an answer is a text span in the 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 following conditional log likelihood:

$$\log \prod_{k=1}^{N} p(q^k, a^k|c^k) = \sum_{k=1}^{N} \log p(q^k, a^k|c^k)$$

where N is the size of the training, development, or test set. For simplicity, we remove superscript k in the following sections.

3.3 Variational Lower Bound

Because questions and answers are different types of observed variables, embedding QA pairs into different latent spaces may be suitable. For example, different questions can correspond to the same answer (Table 1). Thus, we introduce two independent latent random variables to assign the role of diversifying AE and QG to z and y, respectively (see Figure 1 (b)). The variational lower bound of our VQAG is as follows:

$$\begin{aligned}\log p_\theta(q,a|c) \geq \mathbb{E}_{z,y\sim q_\phi(z,y|q,a,c)}[\log p_\theta(q|y,a,c) \\ + \log p_\theta(a|z,c)] - D_{\mathrm{KL}}(q_\phi(z|a,c)||p_\theta(z|c)) \\ - D_{\mathrm{KL}}(q_\phi(y|q,c)||p_\theta(y|c)).\end{aligned} \tag{2}$$

See Appendix A for the derivation of Eq. 2.

3.4 Explicit KL control

VAEs often suffer from "posterior collapse" , where the model learns to ignore latent variables and generates outputs that are almost the same. This problem occurs especially when VAEs are used for modeling discrete data and implemented with strong decoders such as LSTM (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 to control the KL term (Tolstikhin et al., 2018; Zhao et al., 2017; Higgins et al., 2017).

We also observe that this issue happens when implementing our model according to the Ineq. 2. To mitigate this problem, inspired by Prokhorov et al. (2019), we use modified β-VAE (Higgins et al., 2017) proposed by Burgess et al. (2018), which uses two hyperparameters to control the KL terms. Our modified variational lower bound is as follows:

$$\begin{aligned}\log p_\theta(q,a|c) \geq \mathbb{E}_{z,y\sim q_\phi(z,y|q,a,c)}[\log p_\theta(q|y,a,c) \\ + \log p_\theta(a|z,c)] \\ - \beta|D_{\mathrm{KL}}(q_\phi(z|a,c)||p_\theta(z|c)) - C| \\ - \beta|D_{\mathrm{KL}}(q_\phi(y|q,c)||p_\theta(y|c)) - C|,\end{aligned} \tag{3}$$

where $\beta > 0$ and $C \geq 0$. We use the same β and C for the two KL terms for simplicity. In this paper, we set $\beta = 1$ and change only C because C was enough to regularize the KL terms in our case (see Table 2).

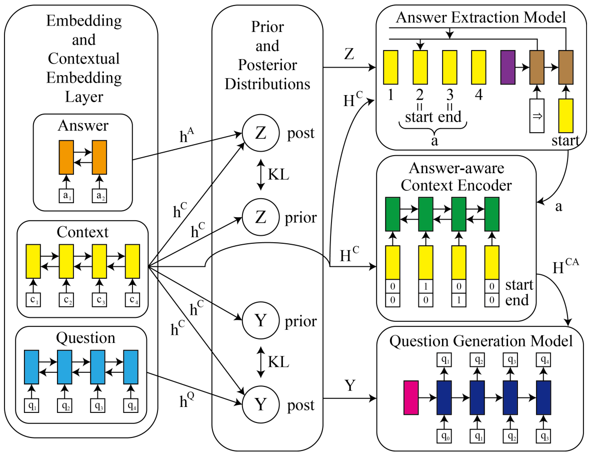

Figure 2: Overview of the model architecture. Each module with its input and output is shown. Note that the latent variables z and y are sampled from the posteriors when computing the variational lower bound and from the priors during generation. See §3.5 for detailed computation in each module.

3.5 Model Architecture

An overview of our VQAG is given in Figure 2. We describe the details of each module below. Here, we denote $c = \{c_t\}_{t=1}^{L_C}$, $q = \{q_t\}_{t=1}^{L_Q}$, and $a = \{a_t\}_{t=1}^{L_A} = \{c_t\}_{t=start}^{end}$, where each element represents one word, and L_C, L_Q, and L_A are, respectively, the lengths of the context, question, and answer span.

Embedding and Contextual Embedding Layer

First, in the embedding layer, the ith word, w_i, of a sequence of length L is simultaneously converted into word- and character-level embedding vectors, e_i^w and e_i^c, by using a convolutional neural network (CNN) based on Kim (2014). Then, e_i^w and e_i^c are concatenated across columns and $e_i = [e_i^w; e_i^c]$ is obtained.

After that, we pass the embedding vectors to the contextual embedding layer as follows:

$$H, h = \mathrm{BiLSTM}([e_1^T; e_2^T; ...; e_L^T]) \tag{4}$$

where $H \in \mathbb{R}^{L\times 2d}$ is the concatenated outputs of LSTMs (Hochreiter and Schmidhuber, 1997) in each direction at each time step, e^T denotes the transpose of e, and $h \in \mathbb{R}^{2d}$ is the concatenated last hidden state vectors of LSTMs in each direction. This bidirectional LSTM (BiLSTM) encoder is shared by the AE and QG tasks. The outputs have superscripts, H^C, h^C, H^Q, h^Q, H^A, and h^A to indicate where they come from; i.e., C, Q, and

A denote the context, question, and answer, respectively.

Prior and Posterior Distributions

Following Zhao et al. (2017), 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:

$$\begin{aligned} z|a,c &\sim \mathcal{N}(\mu_{post_Z}, diag(\sigma^2_{post_Z})) \\ z|c &\sim \mathcal{N}(\mu_{prior_Z}, diag(\sigma^2_{prior_Z})) \\ y|q,c &\sim \mathcal{N}(\mu_{post_Y}, diag(\sigma^2_{post_Y})) \\ y|c &\sim \mathcal{N}(\mu_{prior_Y}, diag(\sigma^2_{prior_Y})). \end{aligned}$$

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

$$\begin{aligned} \begin{bmatrix} \mu_{post_Z} \\ \log(\sigma^2_{post_Z}) \end{bmatrix} &= W_{post_Z} \begin{bmatrix} h^C \\ h^A \end{bmatrix} + b_{post_Z} \\ \begin{bmatrix} \mu_{prior_Z} \\ \log(\sigma^2_{prior_Z}) \end{bmatrix} &= W_{prior_Z} h^C + b_{prior_Z} \\ \begin{bmatrix} \mu_{post_Y} \\ \log(\sigma^2_{post_Y}) \end{bmatrix} &= W_{post_Y} \begin{bmatrix} h^C \\ h^Q \end{bmatrix} + b_{post_Y} \\ \begin{bmatrix} \mu_{prior_Y} \\ \log(\sigma^2_{prior_Y}) \end{bmatrix} &= W_{prior_Y} h^C + b_{prior_Y}. \end{aligned}$$

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.,

$$p(a|c) = p(c_{end}|c_{start}, c)p(c_{start}|c), \quad (5)$$

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 into account the initial hidden state $h_0^{AE} = W_1 z + b_1$, which in the end diversify AE by enabling the mappings from z to a to be learned. The decoding process is as follows:

$$\begin{aligned} h_i^{IN} &= \begin{cases} e(\Rightarrow) & \text{if } i = 1 \\ H^C_{t_{i-1}} & \text{if } i = 2 \end{cases} \\ h_i^{AE} &= \text{LSTM}(h_{i-1}^{AE}, h_i^{IN}) \\ u_{ij}^{AE} &= (v^{AE})^T \tanh(W_2 H_j^C + W_3 h_i^{AE} + b_2) \\ p(c_{t_i}|c_{t_{i-1}}, c) &= \text{softmax}(u_i) \end{aligned}$$

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 ith 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.

Answer-aware Context Encoder

To compute answer-aware context information for QG, we use another BiLSTM as follows:

$$H^{CA}, h^{CA} = \text{BiLSTM}([H^C, o_{start}, o_{end}]) \quad (6)$$

where o_{start} and $o_{end} \in \mathbb{R}^{L_C}$ are the one-hot vectors of the start and end positions of an answer span. $H^{CA} \in \mathbb{R}^{L_C \times 2d}$ is used as the source for attention and copying in question generation. ($h^{CA} \in \mathbb{R}^{2d}$)

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:

$$\begin{aligned} h_i^{QG} &= \text{LSTM}(h_{i-1}^{QG}, q_{t-1}) \\ u_{ij}^{att} &= (v^{att})^T \tanh(W_5 h_i^{QG} + W_6 H_j^{CA} + b_4) \\ a_i^{att} &= \text{softmax}(u_i^{att}) \\ \hat{h}_i &= \textstyle\sum_j a_{ij}^{att} H_j^{CA} \\ \tilde{h}_i &= \tanh(W_7([\hat{h}_i; h_i^{QG}] + b_5)) \\ P_{vocab} &= \text{softmax}(W_8(\tilde{h}_i) + b_6), \end{aligned}$$

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

$$\begin{aligned} u_{ij}^{copy} &= (v^{copy})^T \tanh(W_9 h_i^{QG} + W_{10} H_j^{CA} + b_7) \\ a_i^{copy} &= \text{softmax}(u_i^{copy}) \end{aligned}$$

Accordingly, the probability of outputting q_i is:

$$\begin{aligned} & p_g = \sigma(W_{11} h_i^{QG}) \\ & p(q_i|q_{1:i-1}, a, c) \\ & \quad = p_g P_{vocab}(q_i) + (1 - p_g) \textstyle\sum_{j:c_j = q_i} a_{ij}^{copy} \end{aligned}$$

where σ is the sigmoid function.

4 Experiments & Results

See Appendix B for the training details.

4.1 Dataset

We used SQuAD v1.1 (Rajpurkar et al., 2016), a large QA pair dataset consisting of documents collected from Wikipedia and 100k QA pairs created by crowdworkers. Each question in SQuAD can be answered by a text span in a context. Since the SQuAD test set has not been released, we split the dataset following Du et al. (2017), where the original training set is split into training and development sets and the original development set is used as a test set. In so doing, the sizes of the training, development and test sets amounted to 70,484, 10,570, and 11,877, respectively.

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

Table 2: QA pair modeling capacity measured on the test set. NLL: negative log likelihood ($-\log p(q, a|c)$). $\text{NLL}_a = -\log p(a|c)$, $\text{NLL}_q = -\log p(q|a, c)$. D_{KL_z} and D_{KL_y} are Kullback–Leibler divergence between the approximate posterior and the prior of the latent variable z and y. The lower NLL is, the higher the probability is that the model assigns to the test set. NLL for our models are estimated with importance sampling using 300 samples.

4.2 QA Pair Modeling

We originally developed a QA pair modeling to evaluate QA pair generative models. We compared models based on the bases of the probability they assigned to the ground truth QA pairs. We chose the negative log likelihood (NLL) of QA pairs as the metric, namely, $-\frac{1}{N}\sum_{k=1}^{N}\log p(q^k, a^k|c^k)$. Since variational models can not directly compute NLL, we estimate NLL with importance sampling. We also estimate each term in decomposed NLL, i.e.,$\text{NLL} = \text{NLL}_\text{a} + \text{NLL}_\text{q} = -\log p(a|c) - \log p(q|a, c)$. The better a model performs in this task, the better it fit 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), eliminating all the architectures related to latent random variables in our models and treating a sequence of the start and end positions of all the possible answers in context as the output of AE.

	Relevance				Diversity
	Precision		Recall		Dist
	Prop.	Exact	Prop.	Exact	
NER	34.44	19.61	64.60	45.39	30.0k
BiLSTM-CRF w/ char w/ NER (2018)	45.96	33.90	41.05	28.37	-
VQAG					
$\text{C} = 0$	**58.39**	**47.15**	21.82	16.38	3.1k
$\text{C} = 5$	30.16	13.41	**83.13**	**60.88**	71.2k
$\text{C} = 20$	21.95	5.75	72.26	42.15	**103.3k**
$\text{C} = 100$	23.32	7.48	71.74	39.70	84.6k

Table 3: Results for answer extraction on the test set. For all the metrics, higher is better.

Result Table 2 shows the result of QA pair modeling. First, our models with $\text{C} = 0$ 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 $\text{C} = 0$. In other tasks, it is shown that our model with $\text{C} = 0$ collapses into a deterministic model. The fact that NLL_a is consistently lower than NLL_q is due to the decomposition of probability $p(a|c) = p(c_{end}|c_{start}, c)p(c_{start}|c)$ and $p(q|a, c) = \prod_i p(q_i|q_{1:i-1}, a, c)$, which is sensitive to the sequence length. Also, we observe that the hyperparameter C can control the KL values, showing the potential to avoid the posterior collapse issue in our case. When we set $C > 0$, KL values are greater than 0, which implies that latent variables have non-trivial information about questions and answers.

4.3 Answer Extraction

Inputs were the contexts and outputs were a set of multiple answer spans. Following Du and Cardie (2018), to measure the accuracy of multiple phrases, we computed *Proportional Overlap* and *Exact Match* metrics (Breck et al., 2007; Johansson and Moschitti, 2010) for each pair of a predicted answer and a ground truth.[1] *Proportional Overlap* returns scores proportional to the amount of overlap. We report the precision and recall with respect to the above metrics.

Our models are different from existing models in

[1]We exclude *Binary Overlap* because, as Breck et al. (2007) discussed, *Binary Overlap* assigns high scores on systems that extract the entire input context, and therefore is not a reliable metric.

	Relevance							Diversity			
	B1	B2	B3	B4	ME	RL	Token	D1	D2	E4	SB4
ELMo+QPP&QAP(2019)											
w/Beam10	48.39	32.71	24.13	18.34	24.82	46.66	133.2k	10.1k	45.8k	15.75	-
w/DivBeam50	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
ELMo+QPP&QAP(2019)											
w/DivBeam50	**62.32**	**47.77**	**37.96**	**30.05**	**36.77**	**62.87**	7.0M	15.8k	218.9k	18.28	91.44
VQAG											
$C = 0$	35.57	18.75	10.79	6.35	18.31	33.92	7.6M	14.4k	155.3k	17.33	97.61
$C = 5$	44.19	27.09	16.33	9.71	25.84	45.18	11.5M	19.0k	481.1k	19.71	82.59
$C = 20$	48.19	32.87	22.96	14.94	25.29	48.26	4.9M	**22.4k**	**549.2k**	19.72	44.41
$C = 100$	35.22	19.88	13.25	9.20	22.27	37.55	8.2M	22.1k	508.8k	**19.74**	**44.22**

Table 4: Results for answer-aware question generation on the test set of Du et al. (2017)'s split of SQuAD. Paragraph-level contexts and answer spans are used as input. 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 BLEU-1.) One question per answer-context pair is evaluated in the upper part, while 50 questions per answer-context pair is evaluated in the lower part to assess their diversity.

that they can generate an arbitrary number of samples and improve diversity. For comparison, we had our models extract a total of 50 answer spans from each context to assess their diversity and quality, while the existing models can extract only a fixed set of answer spans. To measure the diversity of the predicted answer spans, we calculated the Dist score as the the total number of distinct spans.

For AE, we adopted two baselines, named entity recognition (NER) and BiLSTM-CRF w/ char w/NER (Du and Cardie, 2018) For NER, we used spaCy. For BiLSTM-CRF w/ char w/ NER, we directly copied the scores from Du and Cardie (2018).

Result Table 3 shows the result. Our model with the condition $\mathrm{C} = 5$ performed the best in terms of the recall scores, while surpassing NER in terms of diversity. From the viewpoint of diversity, $\mathrm{C} = 20$ is the best setting. However, high $Dist$ scores do not occur together with high recall scores. This observation shows the trade-off between diversity and quality. In this task, we show that our model with $\mathrm{C} = 5$ can cover most of the human-created answers and also extract more diverse answers than baselines. However, when $\mathrm{C} = 0$, the $Dist$ score is fairly low. This implies the posterior collapse issue, though the precision scores are the best.

While our models with $C \geq 5$ had low precision, it was due to the diversity of extracted answers. If diversity is improved, answer spans that are not treated as ground truths would be extracted. Since even the test set do not cover all the possible answer spans, we assert that low precision scores do not necessarily mean poor performance.

4.4 Question Generation

The inputs were the contexts and gold answer spans. To see how well our models could generate diverse questions, we had them generate a total of 50 questions from each context-answer pair.

We calculated the BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), and ROUGE-L (Lin, 2004) scores, and report the recall scores per reference question. Since our motivation is to improve diversity, precision metrics are not appropriate in our setting. Thus, we do not report precision scores here. To measure diversity, we computed Dist-n, Ent-n (Serban et al., 2017; Zhang et al., 2018), and Self-BLEU (Zhu et al., 2018). Ent-n is the entropy (in bits) of n-grams, and it reflects how evenly n-grams are generated. Self-BLEU evaluates the degree to which sentences generated by a system resemble each other. We calculated Self-BLEU scores for 50 questions generated from each context-answer pair and averaged them. We computed Dist-n following the definition of Xu et al. (2018), wherein Dist-n is the number of distinct n-grams.[2] We also reported the total number of generated words as reference.

For QG, we compared our models with the ELMo+QAP&QPP model (Zhang and Bansal, 2019), which achieved the state-of-the-art in

[2] Dist-n is often defined as the ratio of distinct n-grams (Li et al., 2016) but this is not fair when the number of generated sentences differs among models, so we did not use this.

beyonc '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 beyonc '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 " . rosen notes that the hip hop era highly influenced beyonc 's strange rhythmic vocal style , but also finds her quite traditionalist in her use of balladry , gospel and falsetto . other critics praise her range and power , with chris richards of the washington post saying she was " capable of punctuating any beat with goose - bump - inducing whispers or full - bore diva - roars . "

Table 5: Heatmap of 250 answer spans extracted using our VQAG (C = 5), the best performing model in terms of recall of Exact match (see Table 3). The darker the color is, the more often the word is extracted. The phrases surrounded by are the ground truth answers of SQuAD.

C=0	C=5	C=20	C=100
beyonc range spans spans spans spans or four octaves spans ? —four	how can one find her vocal abilities in key music ? —she is identified as the centerpiece of destiny 's child	how does her voice as her voice ? —one of the most compelling instruments in popular music " .	leptines polybolos ? —four
beyonc range spans spans spans spans spans and which vocal range ? —four	how many octaves is beyonc 's vocal range spans four octaves ? —spans four	how many power ballads are used by chris richards ? —the daily mail calls beyonc 's voice " versatile "	j.n. ? —four octaves

Table 6: Examples of QA pairs generated with our model. The input context is the same as the one in Table 5.

SQuAD QG. Since diversity metrics were not reported in that paper, we reran the model, which is publicly available [3]. In addition, to compare our models with the baseline under an equivalent condition, we also reran the ELMo+QAP&QPP model with diverse beam search (Li et al., 2016), kept top 50 questions per answer, and used them to calculate the metrics.

Result Table 4 shows the result of QG. The recall scores of our model with C=20 were comparable to the scores of ELMo+QAP&QPP w/Beam10 and w/DivBeam50. Though ELMo+QAP&QPP w/DivBeam50 is superior in terms of the recall of relevance scores, our models perform significantly better in terms of the diversity scores. This shows that our model can improve diversity while generating high-quality questions. Among the various settings of C, 20 is suitable based on this result.

5 Analysis

Since it is hard to evaluate generated QA pairs that are valid but not close to the ground truths, we analyze the generated questions and answers qualitatively.

Table 5 shows the example answers extracted by our model and the gold answers of SQuAD. Our model extracts every gold answer of SQuAD at least once. Moreover, there are answers extracted by our model that are not used in SQuAD but question-worthy. For example, "jon pareles" and "one of the most compelling instruments in popular mucis" are question-worthy because these are related to the main topic, Beyoncé. Note that our model can extract not only named entities but also phrases of other types like this example.

Table 6 shows some examples of generated QA pairs from the various settings of C. The examples with $\mathrm{C} = 5$ seems the most reasonable and diverse. When $\mathrm{C} = 0$, the generated QA pairs are reasonable but lack diversity, suffering from posterior collapse. When $\mathrm{C} = 100$, the generated QA paris are diverse but not reasonable. From this result, finding an appropriate value of C is necessary.

6 Conclusion

We designed a variational QA pair generative model, consisting of two independent latent random variables. We showed explicitly controlling the KL term could either enable our model to perform well in distribution modeling ($\mathrm{C} = 0$) or avoid posterior collapse and improve diversity and recall-oriented relevance scores ($\mathrm{C} > 0$). However, it is not trivial how to find the optimal C.

[3] https://github.com/ZhangShiyue/QGforQA

Acknowledgments

We would like to thank Saku Sugawara at National Institute of Informatics for his valuable support. This work was supported by NEDO SIP-2 "Big-data and AI-enabled Cyberspace Technologies.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. *arXiv e-prints*, page arXiv:1906.05416.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/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 β-VAE. *arXiv e-prints*, page 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.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. 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.

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. *CoRR*, abs/1905.03197.

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.

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.

Vrindavan Harrison and Marilyn Walker. 2018. Neural generation of diverse questions using answer focus, contextual and linguistic features. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 296–306. 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 *Proceedings of the 5th International Conference on Learning Representations*.

Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

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 S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *CoRR*, abs/1705.03551.

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. 2018. Improving neural question generation using answer separation. *CoRR*, abs/1809.02393.

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. *CoRR*, abs/1412.6980.

Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114.

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

Vishwajeet Kumar, Kireeti Boorla, Yogesh Meena, Ganesh Ramakrishnan, and Yuan-Fang Li. 2018. Automating Reading Comprehension by Generating Question and Answer Pairs. *arXiv e-prints*, page arXiv:1803.03664.

Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuan-Fang Li. 2018. A framework for automatic question generation from text using deep reinforcement learning. *CoRR*, abs/1808.04961.

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.

Mike Lewis and Angela Fan. 2019. Generative question answering: Learning to answer the whole question. In *Proceedings of the Seventh 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. 2016. 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. 2016. A Simple, Fast Diverse Decoding Algorithm for Neural Generation. *arXiv e-prints*, page 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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. 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, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. 2019. Learning to generate questions by learning what not to generate. *CoRR*, abs/1902.10418.

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, Amsterdam, The Netherlands, The Netherlands. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. 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.

Victor Prokhorov, Ehsan Shareghi, Yingzhen Li, Mohammad Taher Pilehvar, and Nigel Collier. 2019. On the importance of the Kullback-Leibler divergence term in variational autoencoders for text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 118–127, Hong Kong. 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.

Thomas Scialom, Benjamin Piwowarski, and Jacopo Staiano. 2019. Self-attention architectures for answer-agnostic neural question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6027–6032, 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.

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.

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 *Advances in neural information processing systems*, pages 3104–3112.

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 *Proceedings of the Fifth International Conference on Learning Representations*.

Siyuan Wang, Zhongyu Wei1, Zhihao Fan1, Yang Liu, and Xuanjing Huang. 2019. A multi-agent communication framework for question-worthy phrase extraction and question generation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. 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.

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. 2017. InfoVAE: Information Maximizing Variational Autoencoders. *arXiv e-prints*, page arXiv:1706.02262.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. 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, pages 1097–1100, New York, NY, USA. ACM.

A Derivations of the Variational Lower Bound

The equation 2 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 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 have most frequent 50k words in our training set. The dimension of character-level embedding vectors is 32. The number of windows are 100. The dimension of hidden vectors are 300. The dimension of latent variables are 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 was initialized with Xavier Initialization (Glorot and Bengio, 2010). Models were trained for 20 epochs with a batch size of 16. We used a dropout (Srivastava et al., 2014) rate of 0.2 for all the LSTM layers and attention modules.