Diversify Question Generation with Continuous Content Selectors and Question Type Modeling

Zhen Wang, Siwei Rao, Jie Zhang, Zhen Qin, Guangjian Tian, Jun Wang

Huawei Noah's Ark Lab

{wangzhen150, raosiwei, clark.zhang}@huawei.com

{qin.zhen, tian.guangjian, w.j}@huawei.com

Abstract

Generating questions based on answers and relevant contexts is a challenging task. Recent work mainly pays attention to the quality of a single generated question. However, question generation is actually a one-to-many problem, as it is possible to raise questions with different focuses on contexts and various means of expression. In this paper, we explore the diversity of question generation and come up with methods from these two aspects. Specifically, we relate contextual focuses with content selectors, which are modeled by a continuous latent variable with the technique of conditional variational auto-encoder (CVAE). In the realization of CVAE, a multimodal prior distribution is adopted to allow for more diverse content selectors. To take into account various means of expression, question types are explicitly modeled and a diversity-promoting algorithm is proposed further. Experimental results on public datasets show that our proposed method can significantly improve the diversity of generated questions, especially from the perspective of using different question types. Overall, our proposed method achieves a better trade-off between generation quality and diversity compared with existing approaches.

1 Introduction

As a reverse task of question answering (QA), question generation (QG) aims to generate questions from a given answer and its relevant context. The task holds the potential value of educational purpose to generate questions for reading comprehension materials (Heilman and Smith, 2010). It can also be deployed as chatbot components (Li et al., 2017) for evaluating or improving mental health (Colby, 1975). Moreover, QG can be applied to extend the question-answer pairs (Du and Cardie, 2018) for QA systems.

Traditional methods for QG mainly use rigid heuristic rules to transform a sentence into related questions (Heilman, 2011). However, these approaches heavily rely on manually crafted features, which cannot be easily generalized. In recent years, neural techniques are applied to this task and have achieved significant progress (Zhou et al., 2017; Du et al., 2017). Most of these methods follow the one-to-one encoder-decoder paradigm and focus on improving the quality of a single generated question (Zhao et al., 2018; Sun et al., 2018).

Source context: the network was engineered and operated by mci telecommunications under a cooperative agreement with the nsf .
Target question: who operated the vbsn network?

Focus 1: the network was engineered and operated by mci telecommunications under a cooperative agreement with the nsf .
(Ours) *who* operates the network with nsf ?

Focus 2: the network was engineered and operated by mci telecommunications under a cooperative agreement with the nsf .
(Ours) *who* operated the network under a cooperative agreement with the nsf ?

Focus 3: the network was engineered and operated by mci telecommunications under a cooperative agreement with the nsf .
(Ours) *in* what company was the network engineered and operated by the nsf ?

Figure 1: Diversified questions generated by our method for the given passage-answer pair (answer is underlined). Different questions can be raised according to distinct focuses on the context (colored) and various means of expression (italic).

However, given an answer and its associated context, it is possible to raise multiple questions with different focuses on the context and various means of expression. Figure 1 shows some different questions that can be generated from a given source context. The characteristic of diversity is inherent in QG and has the potential to enhance

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2134–2143

the value of this task. However, the diversity is not fully explored with existing methods. Yao *et al.* (2018) and Fan *et al.* (2018b) noticed this problem and modeled the variety with latent variable models. However, the introduced latent variable was regarded as a holistic attribute, whose meaning was opaque and weakly related to the origin of diversity. More recently, Cho *et al.* (2019) proposed a mixture content selection model for generation, whose diversity is determined by a fixed number of selectors. However, the discrete property confines its variety to a large extent.

In this paper, we use a more flexible continuous latent variable for content selection to deal with different focuses on a context. Moreover, question types are explicitly incorporated to consider different ways of expression. With these components, a question can be generated in three steps. Firstly, a content selector in the form of a continuous latent variable is sampled conditioning on the source context. Secondly, a question type is predicted based on the context as well as the content selector. Lastly, the content of a question is generated with above information about contextual focuses and means of expression. Considering the variety of content selectors and question types, the diversity of generated questions can be ensured.

Overall, the main contributions of this paper are as follows:

- We explicitly consider the content selection process of QG and model content selectors as a continuous latent variable for different focuses on contexts. CVAE is utilized and the multimodal prior technique is adopted for more diverse selectors.
- We consider various means of expression through the incorporation of question type modeling. A diversity-promoting algorithm concerning the use of distinct question types among generations is proposed further.
- We conduct experiments on the public datasets *SQuAD* and *NewsQA*, whose results demonstrate a better trade-off between generation quality and diversity compared with previous methods. Further analysis demonstrates the effectiveness of our proposed components.

2 Related Work

Automatic question generation has attracted an increasing attention from the natural language generation community in recent years, which is reflected in newly published datasets (Zhou et al., 2017; Chen et al., 2018) and sophisticated techniques (Du et al., 2017; Liu et al., 2019).

Traditional methods are mainly rule-based, where they first transform the source information into syntactic representation and then use templates to generate related questions (Heilman, 2011). These methods largely depend on rigid heuristic rules and cannot be easily generalized.

In contrast to rule-based methods, neural networks have the potential to learn implicit patterns from labeled data, thus become more prevalent in question generation. Du *et al.* (2017) and Zhou *et al.* (2017) followed the paradigm of sequence-to-sequence and showed promising results when combining rich features and attention mechanism. Sun *et al.* (2018) and Zhou *et al.* (2019) incorporated answer-focused information to improve the relevance between answers and questions. Liu *et al.* (2019) and Chen *et al.* (2020) introduced graph networks to estimate significant contents in the source context.

Most of previous work regarded question generation as a one-to-one problem and focused on improving the quality of a single generated question. Some work noticed the diversity inherent in QG and came up with methods to consider this characteristic. Yao *et al.* (2018) used a latent variable to model the holistic attributes in questions. Similar ideas could also been found in some related work (Jain et al., 2017; Fan et al., 2018b). However, the meaning of the holistic features is only opaque and cannot be strongly connected with diversity. More recently, Cho *et al.* (2019) proposed a mixture content selection model for generation. The diversity was determined by a fixed number of content selectors. Different from their work, we model the latent variable of content selectors in a continuous space, which holds the potential of capturing more variety inherent in content selection.

Besides above related work, other techniques plugged into the general encoder-decoder framework can also be utilized to promote diversity (Li et al., 2016; Shen et al., 2019). However, the particular characteristics of question generation are not fully considered in these approaches.

3 Method

Question generation aims to model the probability of a question q given an answer a and its context c,

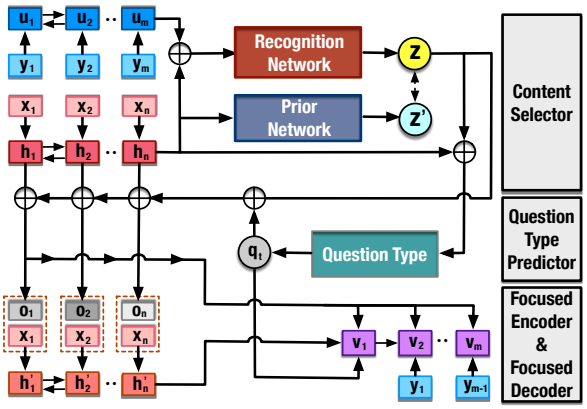

Figure 2: The framework of our model for diverse question generation, which can be decomposed into three stages.

which can be combined as the source information $x = \{c, a\}$.

To diversify generated questions, we incorporate a continuous multi-dimensional latent variable z for content selection and explicitly model question types to deal with means of expression. Generation can be factorized into three stages. Firstly, a content selector z is sampled conditioning on the input x. This is used to indicate which parts of the source information should be focused on. Secondly, a question type q_t is predicted considering the specific content selector z and the input x. Lastly, the relevant question content q_c is generated with selected contents and predicted question type. The final question q can be composed as (q_t, q_c). The factorization can be formulated as follows:

$$\begin{aligned} p_\theta(q|x) &= \mathbb{E}_{z\sim p_\theta(z|x)}[p_\theta(q|x,z)] \\ &= \mathbb{E}_{z\sim p_\theta(z|x)}[p_\theta(q_t|x,z)p_\theta(q_c|x,z,q_t)] \end{aligned} \tag{1}$$

The choice of a continuous latent variable as content selectors leads to more variety compared with its discrete counterpart. CVAE (Sohn et al., 2015) is adopted to make training more tractable. Then the objective function turns out be the evidence lower bound (ELBO) of $\log p_\theta(q|x)$:

$$\begin{aligned} L(\theta,\phi;x,q) = \mathbb{E}_{z\sim p_\phi(z|x,q)}[&\log p_\theta(q|x,z) \\ &+ \log p_\theta(z|x) - \log p_\phi(z|x,q)] \end{aligned} \tag{2}$$

where $p_\phi(z|x,q)$ is incorporated to approximate the the posterior distribution $p_\theta(z|x,q)$.

$L(\theta,\phi;x,q)$ can be approximated using Monte Carlo estimate and learning can be conducted with re-parameterization trick (Kingma and Welling, 2014) on $p_\phi(z|x,q)$ and $p_\theta(z|x)$:

$$\begin{aligned} &z \sim p_\phi(z|x,q) \\ &\widetilde{L}(\theta,\phi;x,q) = \log p_\theta(q_t|x,z) + \log p_\theta(q_c|x,z,q_t) \\ &\qquad + \log p_\theta(z|x) - \log p_\phi(z|x,q) \end{aligned} \tag{3}$$

The first two components in $\widetilde{L}$ denote the reconstruction error that forces the sampled content selector to be informative of what to focus on. The last two components constitute a kind of regularization that drive the posterior to match the prior.

The overall architecture is illustrated in Figure 2. In the following subsections, we will elaborate the details of each stage.

3.1 Content Selector

In our framework, the content selector is modeled as a continuous multi-dimensional latent variable z, which is used to focus on relevant contextual information. Following CVAE, a recognition network $p_\phi(z|x,q)$ is defined to approximate the true posterior distribution. As shown in the form of $p_\phi(z|x,q)$, it is conditioned on the source information x as well as the target question q.

As for the source information, we decompose the context c as a sequence of words $\{x_i\}_{i=1}^n$. Following Zhou *et al.* (2017), we exploit lexical features to enrich word embeddings as $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^n$. Then a bidirectional recurrent neural network (Bi-RNN) is used to produce a sequence of hidden states $\{\mathbf{h}_i\}_{i=1}^n$. At last, condensed source information $\mathbf{s}$ is aggregated with a self-attention operation:

$$\begin{aligned} \gamma_i &= \mathrm{softmax}(\mathbf{u}_h^{\mathrm{T}}\tanh(\mathbf{W}_h\mathbf{h}_i + \mathbf{b}_h)) \\ \mathbf{s} &= \sum_{i=1}^{n}\gamma_i\mathbf{h}_i \end{aligned} \tag{4}$$

We assume the target question has content words $\{y_t\}_{t=1}^m$. Then, the target information $\mathbf{t}$ can be calculated with a similar process as Equation 4.

To model the continuous property of the latent variable z, we assume $p_\phi(z|x,q)$ follows multivariate Gaussian distribution with a diagonal covariance matrix, hence the recognition network can be calculated as:

$$\begin{aligned} &p_\phi(z|x,q) \sim \mathcal{N}(\mu, \sigma^2\mathbf{I}) \\ &\begin{bmatrix} \mu \\ \log(\sigma^2) \end{bmatrix} = \mathbf{W}_r \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} + \mathbf{b}_r \end{aligned} \tag{5}$$

Given Equation 3, we also need to define the prior distribution $p_\theta(z|x)$ of the latent variable z.

Traditional methods often represent the prior as another Gaussian distribution for the sake of tractable calculation. To enrich the model with more variety and prevent the variational posterior to be over-regularized, we adopt a multimodal prior distribution. Gaussian mixture distribution has the potential to fit more diverse multi-dimensional data, which are suitable to enlarge the divergence between content selectors with different focuses.

Instead of introducing transformation matrices to mean and variance for each mode, we adopt the multimodal prior technique of VampPrior (Tomczak and Welling, 2018), where only marginal additive parameters are needed and overfitting can be alleviated. More specifically, the multimodal prior distribution can be formulated as follows:

$$p_\theta(z|x) \sim \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(\mu_k, \sigma_k^2\mathbf{I}) \\ \begin{bmatrix}\mu_k \\ \log(\sigma_k^2)\end{bmatrix} = \mathbf{W}_r\begin{bmatrix}\mathbf{s} \\ \tilde{\mathbf{t}}_k\end{bmatrix} + \mathbf{b}_r \tag{6}$$

where $\tilde{\mathbf{t}}_k$ denotes a pseudo-input, which is a learnable vector with the same dimension as $\mathbf{t}$. K is a hyper-parameter denoting the number of modes.

Given above recognition and prior networks, we can use re-parametrization trick to obtain samples of z from $p_\phi(z|x,q)$ (training) or $p_\theta(z|x)$ (testing). With the sampled latent variable z, we can calculate what to focus on the context c:

$$o_i = \text{sigmoid}(\mathbf{u}_z^{\mathrm{T}}\tanh(\mathbf{W}_z[\mathbf{h}_i; z; \mathbf{E}[q_t]] + \mathbf{b}_z)) \tag{7}$$

where $[;]$ means vector concatenation. $\mathbf{E}[q_t]$ denotes the word embedding of question type q_t, which will be elaborated in subsection 3.2. We use $\mathbf{o}$ to represent $\{o_i\}_{i=1}^n$ for simplicity.

3.2 Question Type Predictor

Given source information $\mathbf{s}$ and sampled content selector z, question type predictor produces a probability distribution to indicate how likely the selected contents can be inquired by different question types. In this paper, we categorized question types according to the interrogative words commonly used in general questions. Specifically, they are classified into 8 types - what, who, how, when, which, where, why and other (Zhou et al., 2019).

We combine the contextual information $\mathbf{s}$ and the selector representation z as the input. Two fully connected layers followed by a softmax layer are used to estimate the final question type distribution for a relevant question. The loss corresponds to the first item in Equation 3:

$$\log p_\theta(q_t|x,z) = \log\_\text{softmax}(\mathbf{W}_{t_1}\tanh(\mathbf{W}_{t_2}[\mathbf{s}; z])) \tag{8}$$

Given the question type predictor, we propose a diversity-promoting algorithm in the inference phase. In Algorithm 1, we utilize $decay$ to explicitly control the degree of diversity for multiple generations. Specifically, given multiple samples with their question type distributions as a whole, we iteratively pick the highest probability and assign its type to the corresponding sample. Then, the probability of choosing the same question type for other samples will be restrained by $decay$. Therefore, it is more likely to allocate different types to the rest, thus the degree of diversity in question types can be explicitly promoted.

Algorithm 1 Pseudo-code for diversity-promoting question type selection algorithm. $P \in N \times L$ is the question type distributions of N different samples with L types. $-inf$ represents the negative infinity. $decay$ is a hyper-parameter controlling the degree of diversity and tuned by the development set. The algorithm returns q_t^i for each sample, which means its predicted question type.

procedure QUESTIONTYPESELECT(P, N, L)
  for $t \in \{1, 2, ...N\}$ **do**
    $i, j = \text{argmax}_{i,j}\{P_{i,j}\}$
    $q_t^i = j$
    $P_{ij'} = -inf$ $\quad j' \leftarrow 1 \sim L$
    $P_{i'j} \mathrel{-}= decay$ $\quad i' \leftarrow 1 \sim N$
  end for
  return $\{q_t^i\}_{i=1}^N$
end procedure

3.3 Controlled Generator

We utilize focused encoder and decoder to make the generation process aware of the selected contents and the predicted question type.

3.3.1 Focused Encoder

The selected contents can be regarded as a clue indicator feature (Liu et al., 2019), which assigns a binary value to each word to signify its importance.

To stabilize training, we use the soft version of this indicator feature, whose weight is given by $\mathbf{o}$ in Equation 7. In the inference phase, we discrete this indicator by setting a threshold (Cho et al., 2019). Specifically, this feature is transformed into another embedding as follows:

$$\mathbf{E}[o_i] = \begin{cases} o_i\mathbf{E}_1 + (1-o_i)\mathbf{E}_0 & \text{for training} \\ \mathbb{I}(o_i)\mathbf{E}_1 + (1-\mathbb{I}(o_i))\mathbf{E}_0 & \text{for inference} \end{cases} \tag{9}$$

where $\mathbf{E}_1$ and $\mathbf{E}_0$ correspond to the trainable embeddings for the two values of this clue indicator. $\mathbb{I}(o_i)$ represents the discreteness of the content selection probability o_i. This embedding is appended to the word embedding $\mathbf{x}_i$ introduced in subsection 3.1. The resulting embeddings are denoted as $\{\mathbf{x}'_i\}_{i=1}^n$.

Then another Bi-RNN is utilized to obtain focused contextual representations as $\mathbf{h}' = \{\mathbf{h}'_i\}_{i=1}^n$.

3.3.2 Focused Decoder

We assume that the contextual representations $\mathbf{h}'$, the content selection indicator $\mathbf{o}$ and the question type q_t should be combined to generate relevant question content $q_c = \{y_t\}_{t=1}^m$, which is the remaining part of a question other than its type.

Following the traditional paradigm, a unidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) is employed to form the decoder. It takes the question type q_t as the initial input word y_0 and refers to representations $\mathbf{h}'$ for attention mechanism (Bahdanau et al., 2015). More details can be found in the implementation of NQG++ (Zhou et al., 2017).

Traditional methods calculate attention weights using the correlation between the hidden states of the encoder and the decoder, which is defined at the word level. In our method, the content selector z decides what to focus on before generation, thus has the ability to provide attention at the sentence level. This is similar to the idea used in data-to-text generation (Mei et al., 2016). Therefore, we combine the content selection probability $\mathbf{o}$ to refine the attention weights $\alpha_{t,i}$ at position t:

$$\alpha'_{t,i} = \frac{\alpha_{t,i}o_i}{\sum_{i=1}^n \alpha_{t,i}o_i} \tag{10}$$

Note that incorporating content selection in this way is an independent operation, which can be plugged into any standard attention method.

As for generation distribution, we adopt copy-generator (See et al., 2017) to deal with the out-of-vocabulary problem. Then, the loss function exerted on the question content, which corresponds to the second term of Equation 3, can be calculated as follows:

$$\log p_\theta(q_c|x,z,q_t) = \sum_{t=1}^m \log p_\theta(y_t|y_{<t},\mathbf{h}',\mathbf{o}) \tag{11}$$

3.4 Training

As the selected contents play an important role in our model, we assume they are consistent with the final generation. Although this behavior can be learned with Equation 11 in an end-to-end manner, we add an auxiliary loss function to facilitate it. Formally, we set the gold label of content selection g_i to 1 if the source token x_i appears in the target question q and 0 otherwise. Without annotations of real focuses, above labels serve as proxies to ease learning. The loss function is thus defined as:

$$L_{sel}(\theta,\phi;x,q) = \sum_{i=1}^n [g_i \log o_i + (1-g_i)\log(1-o_i)] \tag{12}$$

It is well known that a vanilla CVAE with RNN decoder has the risk of failing to encoding meaningful information in the latent variable (Bowman et al., 2016). Inspired by the same concern in the previous work (Zhao et al., 2017), we also adopt the bag-of-word loss $L_{bow}(\theta,\phi;x,q)$ as an auxiliary loss, which requires the latent variable to predict the words shown in the target question. Moreover, the technique of KL cost annealing (Bowman et al., 2016) is also incorporated to let the divergence of $p_\phi(z|x,q)$ and $p_\theta(z|x)$ gradually influence the learning procedure.

Therefore, the overall loss function of the whole framework is defined as:

$$\begin{aligned} \widehat{L}(\theta,\phi;x,q) = \widetilde{L}(\theta,\phi;x,q) + L_{sel}(\theta,\phi;x,q) \\ + L_{bow}(\theta,\phi;x,q) \end{aligned} \tag{13}$$

which can be optimized by stochastic gradient descent.

4 Experiments

4.1 Experiment Settings

Dataset We conduct experiments on two public datasets $SQuAD$ (Rajpurkar et al., 2016) and

$NewsQA$ (Trischler et al., 2017). As for $SQuAD$, we follow the same corpus split by Zhou *et al.* (2017) and directly utilize their provided lexical features[1]. There are 86635, 8965 and 8964 sentence-answer-question triples in the training, development and testing set respectively. As for $NewsQA$, we follow the original split of this dataset, resulting in 92549, 5166 and 5126 triples for training, development and testing.

Implementation Details The vocabulary is set to contain the most frequent 20000 words in each training set. We set the dimension of word embedding to 300 and hidden size to 512. The representations of lexical features and focus indicator are randomly initialized as 16-dimensional vectors. The dimension of the latent variable z and the hidden size of the question type predictor are set to 128. The number of layers for RNN is set to 1 in both the encoder and the decoder. We update the model parameters using Adam optimizer (Kingma and Ba, 2014) with learning rate of 0.001, momentum parameters $\beta_1 = 0.9$ and $\beta_1 = 0.999$. Batch size is set to 64 during training. The development set is used to find the best model and hyper-parameters. Our model is implemented with Pytorch 1.0.0.

4.2 Baselines and Metrics

We compare our method with recent diversified generation methods including Truncated Sampling (Fan et al., 2018a), Diverse Beam Search (Vijayakumar et al., 2018), Mixture Decoder (Shen et al., 2019) and Mixture Content Selection (Cho et al., 2019). The implementations and naming conventions of above baselines follow those by Cho *et al.* (2019).

As for our method, to get N generations for each passage-answer pair, we sample N content selectors from the multimodal prior defined by Equation 6. Given these content selectors, question types are promoted to be distinct with Algorithm 1 and greedy search is conducted for a fair comparison. Note that there is no restriction on the number of prior modes (K) to get N samples. However, it is a natural choice to set $K = N$ and get a sample from each mode. We name this model as N-M. Prior. In further analysis, we will also show the influence of setting different values to K.

We use metrics[2] adopted by Cho *et al.* (2019) to evaluate generation quality and diversity:

Top-1 metric (⇑) This measures the top-1 accuracy (BLEU-4) among the N-best generations.

Oracle metric (⇑) This measures the upper bound of top-1 accuracy (Oracle BLEU-4) by comparing the best hypothesis among the top-N generations with the target question. The metric reflects the overall quality of top-N generations.

Pairwise metric (⇓) This measures the within-distribution similarity. The metric computes the average of sentence-level metrics (Self BLEU-4) between one sentence and the rest in a generated collection. Low pairwise metric indicates high diversity.

Given these metrics, we come up with a comprehensive measurement to balance generation quality and diversity.

Overall metric (⇑) This measures the overall performance concerning both quality and diversity:
$\text{Top-1\_metric} \times \text{Oracle\_metric} \div \text{Pairwise\_metric}$

Also, we introduce other two metrics regarding with the diversity of generated question types.

Method	BLEU-4 (Top-1)	Oracle (Top-N)	Pairwise (Self-sim)	Overall (Top-N)	Type stats. (Top-N)
3-Beam	13.59	16.85	67.23	3.40	0.63 / 1.17
3-D. Beam	13.70	16.99	68.02	3.42	0.62 / 1.13
3-T. Sampling	11.89	15.45	**37.37**	4.91	0.70 / 1.61
3-M. Decoder	14.72	19.32	51.36	5.54	0.70 / 1.38
3-M. Selector	**15.87**	**20.44**	47.49	6.83	0.67 / 1.29
3-M. Prior	15.13	19.28	42.37	**6.88**	**0.85 / 2.42**
5-Beam	13.53	18.81	74.67	3.41	0.67 / 1.31
5-D. Beam	13.38	18.30	74.80	3.27	0.65 / 1.24
5-T. Sampling	11.53	17.65	**45.99**	4.43	0.76 / 1.94
5-M. Decoder	15.17	21.97	58.73	5.67	0.77 / 1.69
5-M. Selector	**15.67**	**22.45**	59.82	5.88	0.70 / 1.41
5-M. Prior	15.34	21.15	54.18	**5.99**	**0.96 / 3.85**

Table 1: Automatic metrics on $SQuAD$ about baselines and our proposed method. Method prefixes are the numbers of generations for each passage-answer pair ($N = 3, 5$). The last column is targeted to measure the coverage and the diversity of generated question types.

Method	BLEU-4 (Top-1)	Oracle (Top-N)	Pairwise (Self-sim)	Overall (Top-N)	Type stats. (Top-N)
5-Beam	10.09	15.82	68.88	2.32	0.76 / 1.25
5-D. Beam	10.12	15.51	70.57	2.22	0.75 / 1.19
5-T. Sampling	8.64	14.25	47.57	2.59	0.80 / 1.58
5-M. Decoder	10.02	17.04	55.07	3.10	0.82 / 1.50
5-M. Selector	**10.90**	**17.51**	52.61	3.63	0.77 / 1.29
5-M. Prior	9.90	15.48	**41.37**	**3.70**	**0.89 / 2.24**

Table 2: Automatic metrics on $NewsQA$.

[1] https://res.qyzhou.me/redistribute.zip

[2] ⇑ is used for a metric which is higher with better performance, otherwise ⇓ is marked.

	Diversity (%)		
Baselines	Win	Lose	Tie
v.s. 3-M. Selector	45	26	29
v.s. 3-M. Decoder	46	26	28

Table 3: Human evaluation results on $SQuAD$.

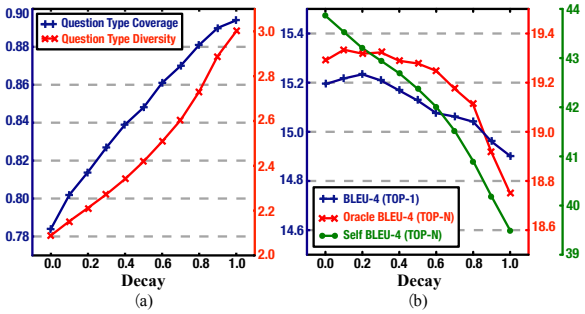

Figure 3: The change of each evaluation metric with different values of the $decay$ hyper-parameter in Algorithm 1. Values range from 0 to 1 with an interval of 0.1. The left subgraph (a) records metrics related to question types and the right one (b) shows metrics concerning BLEU-4. The experiments are conducted with the setting of 3-M. Prior on $SQuAD$.

Type coverage metric (⇑) This measures the percentage that the question type of the target question is covered by top-N generations.

Type diversity metric (⇑) This measures the average number of distinct question types in top-N generations.

4.3 Results and Analysis

Results compared with baselines The experimental results on $SQuAD$ are displayed in Table 1. The table shows that the quality of generated questions with our method (N-M. Prior) scores comparable BLEU-4 to the state-of-the-art, which is much superior compared with methods based on beam search and sampling. Moreover, from the perspective of diversity, our method performs evidently better than other mixture models, resulting in the best trade-off between diversity and quality as shown by the overall metric. Furthermore, focusing on the measurements concerning question types, we can find that our model demonstrates significant improvements from both the coverage and the diversity, which are caused by the explicit modeling and diversifying of question types. We can observe the similar phenomenon that our method performs better with regard to the diversity metrics from the performance on $NewsQA$ in Table 2.

We also conduct human evaluation comparing the diversity of the generated questions from our model 3-M. Prior with other mixture model baselines in Table 3. The table shows that our method outperforms its counterparts in terms of diversity with statistical significance.

Method	BLEU-4 (Top-1)	Oracle (Top-N)	Pairwise (Self-sim)	Overall (Top-N)
3-M. Prior	15.13	19.28	**42.37**	**6.88**
-Diversity-promoting	15.19	19.29	43.86	6.72
-Focused Decoder	15.03	**19.56**	44.96	6.54
-Focused Encoder	14.28	18.67	44.80	5.95
-Focused Decoder & Encoder	14.55	17.95	65.56	3.98
-Content Selection Loss	14.66	18.11	66.23	4.01
-Bag-of-Word Loss	15.29	19.19	50.86	5.77
-KL cost Annealing	**15.58**	18.70	67.94	4.29

Table 4: Ablation results concerning important model components on the test set of $SQuAD$.

Method	BLEU-4 (Top-1)	Oracle (Top-N)	Pairwise (Self-sim)	Overall (Top-N)
1-M. Prior (3 samples)	14.51	18.52	47.50	5.66
3-M. Prior (3 samples)	15.13	**19.28**	**42.37**	**6.88**
5-M. Prior (3 samples)	**15.16**	19.14	44.49	6.52
1-M. Prior (5 samples)	14.55	20.19	56.55	5.19
3-M. Prior (5 samples)	14.88	20.18	57.18	5.25
5-M. Prior (5 samples)	**15.34**	**21.15**	**54.18**	**5.99**

Table 5: Experiments on $SQuAD$ with different numbers of prior modes ($K = 1, 3, 5$) when generating multiple samples ($N = 3, 5$).

Diversifying question types As described in Algorithm 1, the diversity of question types can be explicitly controlled by setting different values of $decay$. The influence is clearly shown in the Figure 3(a). As $decay$ gradually increases, the diversity of question types increases as well as their coverage of the golden type. Also, from the Figure 3(b), we can see that, a small value of $decay$ results in better generation quality metrics. The reason is that the incorporation of more diverse question types may lead to more possibilities of raising good questions. As its value continues to grow, the diversity keeps on increasing at the risk of inappropriate question types used, which results in a slight degradation of the generation quality. We can select an appropriate $decay$ value according to the overall metric.

Ablation Analysis To show the effects of important components in our model, we conduct an ablation study on $SQuAD$. As shown in Table 4, the proposed diversity-promoting algorithm can clearly improve the generation diversity with nearly no negative impact on the quality, which can also be shown in Figure 3 when $decay$ is small. As for

Source context: the network was engineered and operated by mci telecommunications under a cooperative agreement with the nsf .
Target question: who operated the vbsn network?

Mixture Decoder:
Q1: who operated the network in the nsf ?
Q2: who operates the network in the network ?
Q3: who operates the network under a cooperative agreement with the nsf ?

Mixture Content Selection:
Q1: who operates the network ?
Q2: who operates the network ?
Q3: who operates the network with the nsf ?

Ours:
Q1: *who* operates the network with nsf ?
Q2: *who* operated the network under a cooperative agreement with the nsf ?
Q3: *in* what company was the network engineered and operated by the nsf ?

Figure 4: Multiple questions generated by our model 3-M. Prior and other mixture model counterparts.

content selection, incorporating its influence in the encoder-decoder architecture improves the overall metric obviously. Also, we observe that the auxiliary loss function on selected contents can make a big difference, demonstrating its necessity to make content selectors focus on diverse and valid text pieces. Moreover, learning tricks about CVAE contribute to a more informative latent variable and improve the diversity evidently.

Influence of multimodal prior distribution The continuous property of content selectors make it possible to generate N questions even given a standard gaussian prior. However, the introduction of multimodal prior can enrich content selectors with more variety and lead to more diverse generations. As shown in Table 5, the number of prior modes ($K = 1, 3, 5$) has an effect on metrics when generating multiple questions ($N = 3, 5$). First, we can see that the multimodal prior has the ability to improve the generation diversity compared with the standard one, which tallies with our conjecture. Second, when experimenting with the setting $N = K$, almost all of the metrics are better. We can explain this from the fact that samples of content selectors can be taken from different prior modes, which are more diverse. Also, inference accords with the training process in this situation.

Qualitative Analysis Figure 4 shows an example of the generated questions from our model 3-M. Prior and its mixture model counterparts. As shown in this example, our generations often varies in question types and exhibit more diversity. Moreover, we highlight the selected contents of each generation from our model in Figure 1, which shows the effectiveness of our content selection module.

Source context: in the early 1950s , student applications declined as a result of increasing crime and poverty in the hyde park neighborhood .

Q1: *what* did student applications decline in the 1950s ?
Q2: *what* did student applications decline in the early 1950s ?

Q3: *what* was the result of student applications in the 1950s ?
Q4: *what* was the result of student applications in the early 1950s ?

Q5: *in* the early 1950s , student applications declined as a result of what ?
Q6: *in* the early 1950s , what did student applications decline ?

Figure 5: Different generations on $SQuAD$ with the setting of 3-M. Prior. Generations from different prior modes are partitioned by dash lines.

As we use the multimodal prior technique, the diversity of generated questions can be reflected from both intra and inter modes. We can see from Figure 5 that different from other mixture models which can only generate a fixed number of questions, our continuous modeling option makes it possible to produce more generations by sampling from each mode repeatedly. In this example, questions from different modes exhibit a larger divergence compared with those from the same one, which demonstrates once more that the use of a multimodal prior makes a difference to the generation diversity.

5 Conclusion

In this paper, we explicitly diversify the question generation from the perspectives of contextual focuses and means of expression. We model focuses through continuous content selectors and introduce a multimodal prior to allow for more diverse selectors. We consider various means of expression through the modeling of question types and a related diversity-promoting algorithm. On public datasets, our approach achieves the best trade-off between generation quality and diversity. Further analysis also demonstrates the effectiveness of our proposed model components.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. Learningq: A large-scale dataset for educational question generation. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018*, pages 481–490. AAAI Press.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2020. Reinforcement learning based graph-to-sequence model for natural question generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. 2019. Mixture content selection for diverse sequence 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 3121–3131, Hong Kong, China. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Kenneth Mark Colby. 1975. *Artificial Paranoia: A Computer Simulation of Paranoid Processes*. Elsevier Science Inc., New York, NY, USA.

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, Melbourne, Australia. 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, Vancouver, Canada. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018a. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Zhihao Fan, Zhongyu Wei, Piji Li, Yanyan Lan, and Xuanjing Huang. 2018b. A question type driven framework to diversify visual question generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4048–4054. International Joint Conferences on Artificial Intelligence Organization.

Michael Heilman. 2011. Automatic factual question generation from text. *Language Technologies Institute School of Computer Science Carnegie Mellon University*, 195.

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, Los Angeles, California. Association for Computational Linguistics.

Unnat Jain, Ziyu Zhang, and Alexander G. Schwing. 2017. Creativity: Generating diverse questions using variational autoencoders. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5415–5424. IEEE Computer Society.

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

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

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, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions by asking questions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1106–1118. ACM.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730, San Diego, California. 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, Austin, Texas. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5719–5728. PMLR.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3483–3491.

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

Jakub M. Tomczak and Max Welling. 2018. VAE with a vampprior. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR.

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, Vancouver, Canada. Association for Computational Linguistics.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 7371–7379. AAAI Press.

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 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4546–4552. ijcai.org.

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, Vancouver, Canada. 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, Brussels, Belgium. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *Natural Language Processing and Chinese Computing - 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8-12, 2017, Proceedings*, volume 10619 of *Lecture Notes in Computer Science*, pages 662–671. Springer.

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 6032–6037, Hong Kong, China. Association for Computational Linguistics.