Answer-driven Deep Question Generation based on Reinforcement Learning

Liuyin Wang[1] **Zihan Xu**[1] **Zibo Lin**[1] **Hai-Tao Zheng**[1,*] **Ying Shen**[2,*]
[1]Department of Computer Science and Technology, Tsinghua University
[2]School of Intelligent Systems Engineering, Sun Yat-Sen University
{ly-wang19, xu-zh17,lzb18}@mails.tsinghua.edu.cn
sheny76@mail.sysu.edu.cn, zheng.haitao@sz.tsinghua.edu.cn

Abstract

Deep question generation (DQG) aims to generate complex questions through reasoning over multiple documents. The task is challenging and underexplored. Existing methods mainly focus on enhancing document representations, with little attention paid to the answer information, which may result in the generated question not matching the answer type and being answer-irrelevant. In this paper, we propose an Answer-driven Deep Question Generation (ADDQG) model based on the encoder-decoder framework. The model makes better use of the target answer as a guidance to facilitate question generation. First, we propose an answer-aware initialization module with a gated connection layer which introduces both document and answer information to the decoder, thus helping to guide the choice of answer-focused question words. Then a semantic-rich fusion attention mechanism is designed to support the decoding process, which integrates the answer with the document representations to promote the proper handling of answer information during generation. Moreover, reinforcement learning is applied to integrate both syntactic and semantic metrics as the reward to enhance the training of the ADDQG. Extensive experiments on the HotpotQA dataset show that ADDQG outperforms state-of-the-art models in both automatic and human evaluations.

1 Introduction

Neural question generation (QG) aims at generating specific answer related questions from a given document with a target answer based on deep neural networks. Its key applications include generating questions for reading comprehension (Du et al., 2017), enhancing question answering systems as a strategy of data augmentation (Tang et al., 2017; Zhang and Bansal, 2019) and helping digital assistants (e.g., Alexa, Cortana, Siri and Google Assistant) to start and continue a conversation.

Various methods have been proposed for general QG (Zhou et al., 2017; Zhao et al., 2018; Kim et al., 2019; Zhang and Bansal, 2019; Tuan et al., 2020). However, most existing methods focus on generating questions relevant to only one fact without deep comprehension and reasoning. For example, Min et al. (2018) find that more than 80% of the questions in the widely adopted SQuAD dataset (Rajpurkar et al., 2016) are shallow and only relevant to information confined to a single sentence. Generating deep question which requires higher cognitive skills is rarely studied (Pan et al., 2020). These skills include a thorough understanding of the input sources and the ability to reason over disjoint and relevant contexts.

This paper focuses on the task of deep question generation (DQG), which focuses on generating deep questions with multi-hop reasoning over document-level contexts. Previous work mainly focuses on the enhancement of document representations and obtains good performance. However, the answer information is also important since the generated questions should match the answer type and be answer-focused, and several common problems in QG are caused by the lack or improper use of the answer information: 1) The generated questions may be irrelevant to the answer. As shown in Figure 1, with a wrongly chosen question word, *Inappropriate Question 1* is asking about time information but not place.

*corresponding author

Proceedings of the 28th International Conference on Computational Linguistics, pages 5159–5170
Barcelona, Spain (Online), December 8-13, 2020

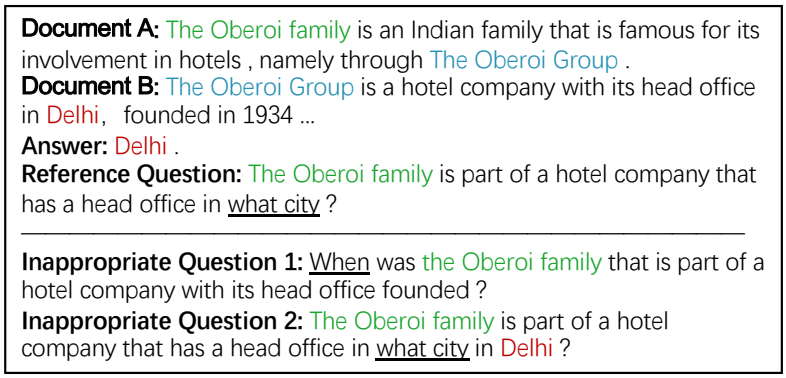

Figure 1: An example of deep question generation from the HotPotQA dataset. The associated contents is displayed in the same color.

2) Without proper guidance during generation, the generated questions may even give the answer away incautiously (*Inappropriate Question 2* in Figure 1), especially when the copy mechanism (Gu et al., 2016) is applied.

In this paper, we propose an Answer-Driven Deep Question Generation (ADDQG) model, which makes better use of the target answer as a guidance to facilitate the generation of deep questions. The model is built on the encoder-decoder paradigm. First, in order to explicitly guide the choice of question words, a novel initialization module is designed to introduce both document and answer information to the decoder, with a gated connection layer to control the proportions of information. Second, we propose the semantic-rich fusion attention mechanism for information integration, thus promoting the proper handling of answer information during question generation. It is a collaborative attention mechanism which integrates the answer with the document representations, where the document representations are concatenations of node representations from Graph Attention Network (GAT) (Velickovic et al., 2017) and contextual representations. Moreover, reinforcement learning is applied to provide feedback to fine-tune the question generator. In order to optimize the evaluation metrics, syntactic and semantic metrics are integrated as the reward to guide the training process, thus guaranteeing the meaningfulness of generated questions.

The contributions of this paper are listed as follows: 1) We propose an answer-driven end-to-end deep question generation model (ADDQG) based on reinforcement learning, which explores more semantic information from the answer to enhance deep question generation. 2) In order to incorporate answer information into the generation of questions, a novel answer-aware initialization module with a gated connection layer and a semantic-rich fusion attention mechanism are designed to promote the proper handling of answer information during the generation process. 3) ADDQG model achieves the state-of-the-art results on the HotpotQA dataset. Human evaluation further verifies the high quality of the generated questions.

2 Related Work

Question Generation Question Generation is one of the typical natural language generation tasks (Reiter and Dale, 2000; Saggion and Poibeau, 2013; Balakrishnan et al., 2019). Generating questions from various kinds of sources, such as texts, search queries, knowledge bases and images, has attracted much attention recently. Our work is most related to previous work on generating questions from texts. Traditional methods are mostly rule-based, which rely on manual rules or templates and rank the generated questions by human-designed features (Heilman and Smith, 2010; Mazidi and Nielsen, 2014), which are costly and lack diversity. Neural QG models are usually variants of the encoder-decoder framework (Du et al., 2017; Zhou et al., 2017; Sun et al., 2018; Pan et al., 2019; Wang et al., 2019). Zhou et al. (2017) propose a feature-rich encoder for the Seq2Seq (Sutskever et al., 2014) model, and Zhao et al. (2018) process paragraph level inputs with maxout pointer and gated self-attention. To deal with the "exposure

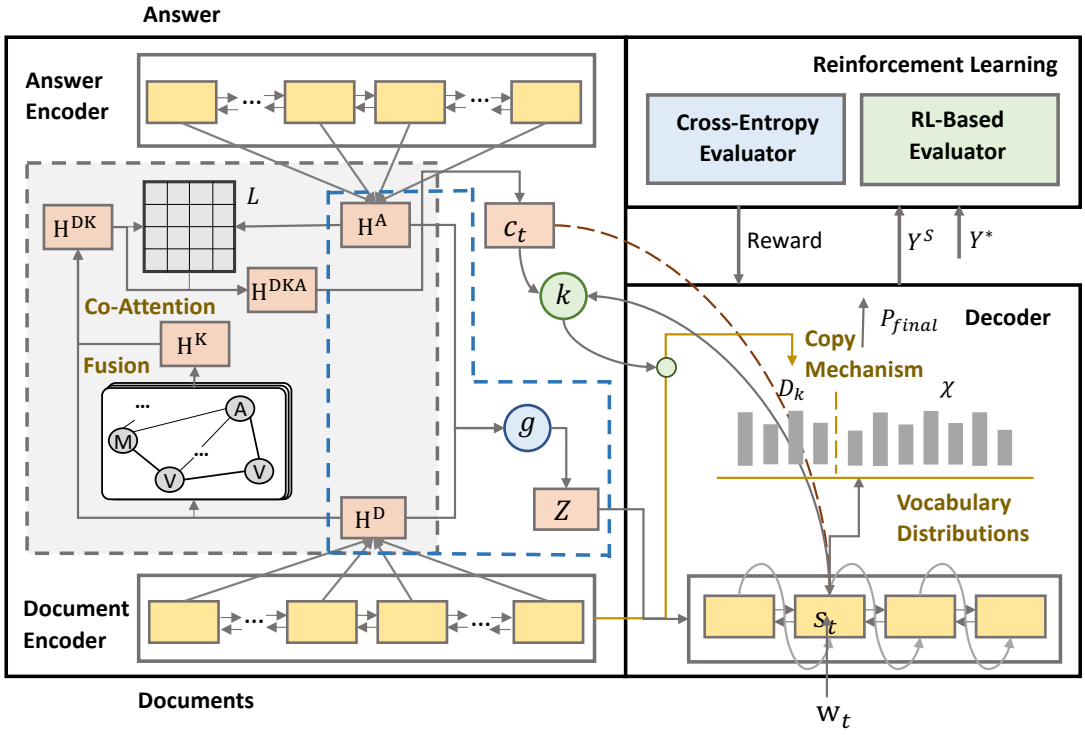

Figure 2: Illustration of ADDQG at generation step t.

bias" problem, reinforcement learning models are applied (Zhang and Bansal, 2019; Chen et al., 2020). However, despite considering longer contexts, the above QG methods generate questions related to only one fact obtained from a single sentence or article without deep comprehension and reasoning. This work focus on generating deep questions with multi-hop reasoning over document-level contexts.

Deep Question Generation Deep Question Generation (DQG) aims to generate complex questions that require reasoning over multiple pieces of information. This task is inspired by multi-hop question answering (Song et al., 2018; Chen and Durrett, 2019; Tu et al., 2020), which aggregate the scattered evidence fragments in multiple documents to predict the correct answer.

Pan et al. (2020) is the first to study the task of DQG. They propose a new framework which incorporates semantic graphs to enhance the document representations and jointly train the tasks of content selection and question decoding. However, they do not pay much attention to the answer information that is a key to question generation and simply introduce the answer to the decoder based on the encoding of word embedding. Our work make better use of the target answer as a guidance to facilitate question generation with the help of the answer-aware initialization module and semantic-rich fusion attention mechanism. Reinforcement learning which integrates both syntactic and semantic metrics as the reward is also applied to enhance the training process.

3 Methodology

3.1 Overview

Deep question generation (DQG) requires the thorough understanding of the input sources and reasoning over disjoint and relevant contexts. In this section, we elaborate on ADDQG model for deep question generation. The key idea of the model is to use an answer-aware initialization module and a semantic-rich fusion attention mechanism to integrate the answer information with the document. Reinforcement learning is also applied to fine-tune the model to get better performance. Figure 2 shows the detailed architecture of the proposed model.

To be specific, given the document collection $X^d = (X_1^d, ..., X_n^d)$ and the corresponding answer $X^a = (x_1^a, ..., x_n^a)$, the DQG task is to find the best $\bar{Y} = (y_1, ..., y_n)$ to maximize the conditional

likelihood given X^d and X^a.

$$\overline{Y} = \arg\max_{Y} P(Y|X^d, X^a). \tag{1}$$

Different from traditional QG, the generation of $\bar{Y}$ involves reasoning over multiple evidence documents d_i, where $i \in [1, n]$ and d_i is in X^d. X^a should not be included in X^d because reasoning is involved to obtain the answer.

3.2 Encoder

Word Encoder The model adopts two encoders for the documents and the answer respectively, so target information can be more precisely located for subsequent operations. The input sources are represented as sequences of embedding vectors. In this work, we use pre-trained GloVE embeddings (Pennington et al., 2014), and get the word vector $(w_{i,1}, w_{i,2}, ..., w_{i,m})$ as the input for the documents X_i^d and the answer X_i^a respectively. We use bidirectional LSTM to obtain forward and backward context representations of each word:

$$\overrightarrow{h_{i,j}} = \overrightarrow{LSTM}\left(\overrightarrow{h_{i,j-1}}, w_{i,j}\right), \overleftarrow{h_{i,j}} = \overleftarrow{LSTM}\left(\overleftarrow{h_{i,j+1}}, w_{i,j}\right). \tag{2}$$

Then they are concatenated to get the final word representation $h_{i,j} = [\overrightarrow{h_{i,j}}; \overleftarrow{h_{i,j}}]$. The answer and the document representations are $H^A = \text{BiLSTM}\left(W_{emb}(X^a)\right)$ and $H^D = \text{BiLSTM}\left(W_{emb}(X^d)\right)$ respectively.

Graph Encoder As shown in Figure 1, the semantic relationship between entities is a powerful clue to determine the inquiry content and reasoning types included. In order to extract semantic information from documents, we use dependency relationship (Dozat and Manning, 2017) to construct a semantic graph based on parsing. First, we initialize each node $v = \{w_j\}_{j=m}^{n}$ to calculate the attention distribution of $H^{\mathcal{D}}$ on all words in v as follows:

$$\gamma_j^v = \text{softmax}\left(\text{ReLU}\left(W_0\left[H^D; w_j\right]\right)\right), \tag{3}$$

where w_j is the context representation of words in nodes, m/n is the starting / ending position of the text span, W_0 is a trainable parameter. Finally, the node is initialized as $h_i^0 = \sum_{j=m}^{n} \gamma_j^v w_j$. In order to represent multiple relationships of edges, we use Graph Attention Network (GAT) (Velickovic et al., 2017) to dynamically determine the weight of adjacent nodes in message delivery using attention mechanism.

$$\begin{aligned}
\eta_{ij} &= W_1^{t-1}\left(\text{ReLU}\left(W_2^{t-1}\left[h_i^{t-1}; h_j^{t-1}\right]\right)\right), \\
\alpha_{ij} &= \frac{\exp\left(\eta_{ij}\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\eta_{ik}\right)}, \\
h_i^t &= \sum_{j\in\mathcal{N}_i} \alpha_{ij} W^{t_{e_{ij}}} h_j^{t-1},
\end{aligned} \tag{4}$$

where $\mathcal{N}_{(i)}$ denotes the neighbors of node v_i. $\alpha_{ij}^{(k)}$ is the attention coefficients between two nodes. $W^{t_{e_{ji}}}$ represents the weight matrix corresponding to the edge type. W_1^{t-1} and W_2^{t-1} are trainable parameters. Finally, a Gated Recurrent Unit (GRU) (Cho et al., 2014) is applied to merge the aggregated neighboring information and get the semantic graph representation H^K.

3.3 Decoder

Answer-Aware Initialization Module Most QG models use the last hidden state of the encoder to initialize the decoder. ADDQG applies an answer driven initialization method, so that it can explicitly guide the choice of question words and generate questions which are more answer-focused. We first design a fusion gate to control the information flow rate of the document and answer.

$$g = \sigma(W_z[H^A; H^D; H^D \odot H^A; H^D - H^A] + b_z), \tag{5}$$

where σ is the sigmoid function. W_z and b_z are trainable parameters.

Then the representations are combined through the gated connection layer:

$$Z = g \odot H^D + (1 - g) \odot H^A, \tag{6}$$

where $\odot$ is the component-wise multiplication. Z is the final initialization of the decoder, which is the deep fusion of answer and document features.

Semantic-Rich Fusion Attention Semantic-rich fusion attention integrates answer with the document and semantic graph to better support the generation process. First, the semantic graph representation H^K is combined with the document representation H^D to get the semantic-rich document representation H^{DK}. To be specific, if node v_i contains word w_i, the word representation H_i^D and node representation $H_{v_i}^k$ are concatenated to get the fused representation H_i^{DK} (padded with a special vector if there is no corresponding v_i):

$$H_i^{DK} = F([H_i^D; H_{v_i}^k)], \tag{7}$$

where F($\cdot$) is the standard nonlinear transformation function. To model the complex interactions between the input sources, we apply the collaborative attention mechanism (Lu et al., 2016) which focuses on both the answer H^A and semantic-rich document representation H^{DK}. To be specific, we first calculate the correlation matrix $L = {H^{DK}}^\top H^A$, which contains the similarity scores of all pairs of document and answer words. The attention weights A^{H^A} are across the answer for each word in the document, and the weights $A^{H^{DK}}$ are across the document for each word in the answer.

$$A^{H^A} = \text{softmax}(L), A^{H^{DK}} = \text{softmax}(L^\top). \tag{8}$$

Next, we calculate the co-dependent representation of the question and document $C^{H^{DK}}$ similar to (Cui et al., 2017):

$$C^{H^A} = H^{DK} A^{H^A}, C^{H^{DK}} = \left[H^A; C^{H^A}\right] A^{H^{DK}}. \tag{9}$$

Then the semantic-rich document information and answer information are integrated to get the fusion representation:

$$H^{DKA} = [H^{DK}; C^{H^{DK}}]. \tag{10}$$

Finally, the semantic-rich representation H^{DKA} is applied to obtain the context vector c_t:

$$\begin{aligned} e_t &= v_a^T \tanh\left(W h_t^* + U H^{DKA}\right), \\ \alpha_t^* &= \text{softmax}\left(e_t\right), \\ c_t &= H^{DKA} \alpha_t^*, \end{aligned} \tag{11}$$

where W, v_a^T and U are trainable parameters.

Taking Z computed in Eq. 6 as the initialization, during decoding, the hidden state h_t^* at step t is:

$$h_t^* = LSTM_{Dec}\left(\left[w_t; c_{t-1}\right], h_{t-1}^*\right), \tag{12}$$

where word w_t is the input.

Copy Mechanism and Maxout Pointer In order to solve the out-of-vocabulary (OOV) problem, the decoder applies the copy mechanism (Gu et al., 2016) which allows the token to be copied from the input sources to the decoding step t. The mechanism utilizes the original attention scores α_t^* calculated in Eq. 11 to get the probability of copy $p_{copy}(y_t)$. We adopt the maxout pointer (Zhao et al., 2018) mechanism to limit the magnitude of scores of repeated words to their maximum value to solve the problem of repetition. The switch gate $k = \sigma\left(W^c h_t^* + U^c c_t + b^c\right)$ determines whether the generated word is sampled from the vocab or copied from the input sources.

$$p_{\text{final}}\ (y_t | y_{<t}; \theta) = k p_{\text{copy}}\ (y_t, \theta_1) + (1 - k)\, p_{\text{gen}}\ (y_t, \theta_2)\,, \tag{13}$$

where $p_{gen}(y_t) = \text{softmax}(W^T[h_{t-1}^*; c_t])$ is the generative probability distribution.

3.4 Reinforcement Learning for Fine-Tuning

The loss function of question generation minimizes the negative log-likelihood of the output generative words as:

$$Loss_{CE} = -\sum\nolimits_t \log P(y_t|y_{<t}, X^d, X^a, \theta). \tag{14}$$

However, using the above cross-entropy loss in the sequence prediction model could make the process brittle, because models trained on a specific distribution of words are used for test data sets with potentially different distributions to predict the next word given the current predicted word (Kumar et al., 2018). This creates "exposure bias" during training (Ranzato et al., 2016), reinforcement learning is widely used to deal with the "exposure bias" in question generation and proved to be effective. We define r as the reward, which is calculated by comparing the output sequence Y with the corresponding ground-truth question Y^* based on the metrics. Similar to (Chen et al., 2020), we use BLEU-4 as reward $r(Y, Y^*)_{\text{BLEU-4}}$ which is directly optimized towards the evaluation metrics, and word movers distance (WMD) as reward $r(Y, Y^*)_{\text{WMD}}$ which makes the model more effective and robust. However, instead of using weighted combination of $r(Y, Y^*)_{\text{BLEU-4}}$ and $r(Y, Y^*)_{\text{WMD}}$, we apply a multi-reward optimization strategy (Pasunuru and Bansal, 2018) to train the model with two mixed losses, because it is hard to find the complex scaling and weight balance among them.

$$\begin{aligned} r(Y, Y^*)_{\text{WMD}} &= f_{\text{WMD}}\left(Y, Y^*\right), \\ r(Y, Y^*)_{\text{BLEU-4}} &= f_{\text{BLEU-4}}\left(Y, Y^*\right). \end{aligned} \tag{15}$$

We follow the effective SCST strategy (Rennie et al., 2017) and take the reward of greedy search result DQG as the baseline b.

$$Loss_{RL} = \left(b - r\left(Y^s, Y^*\right)\right) \log P({y^s}_t|{y^s}_{<t}, X^d, X^a, \theta), \tag{16}$$

where Y^s is the sampled output. We alternately train two mixed losses $Loss^{WMD}_{\text{mixed}}$ and $Loss^{BLEU-4}_{\text{mixed}}$ in a certain proportion.

$$\begin{aligned} Loss^{WMD}_{\text{mixed}} &= \alpha^{WWD} Loss^{WMD}_{RL} + \left(1 - \alpha^{WMD}\right) Loss_{CE}, \\ Loss^{BLEU-4}_{\text{mixed}} &= \alpha^{BLEU-4} Loss^{BLEU-4}_{RL} + \left(1 - \alpha^{BLEU-4}\right) Loss_{CE}. \end{aligned} \tag{17}$$

where α is the scale factor to control the trade-off between cross-entropy loss $Loss_{CE}$ and reinforcement Learning loss $Loss_{RL}$.

4 Experiments

4.1 Experimental Setup

In DQG, question generation requires the thorough understanding of the input sources and reasoning over disjoint and relevant contexts. To evaluate DQG models, conventional QG datasets like SQuAD (Rajpurkar et al., 2016) dataset are insufficient because most of their questions are shallow and only relevant to information confined to a single sentence (Min et al., 2018).

We conduct experiments on HotpotQA (Yang et al., 2018), a challenging dataset in which the questions are generated by reasoning over multiple supporting documents to answer. HotpotQA contains around 113K Wikipedia-based questions. Each question is supported with two documents containing the evidence necessary for answer inferring. For fair comparison, we pre-process the original dataset to select relevant sentences and keep 90,440 / 6,072 examples for training and evaluation respectively.

In order to extract semantic information from documents, we use the dependency parsing method to construct semantics graph. The maximum length of the original document is 200 and the maximum length of target answer is 50. For word embedding, we use pre-trained GloVe word vectors with 300 dimensions and froze them during training. We set the LSTM hidden unit size to 512 and the number of layers to 2 in both the answer and document encoders and the decoder, we design a bidirectional GRU as the graph encoder with unit size 512. Optimization is performed by Adam (Kingma and Ba, 2015), with an initial learning rate of 0.0025.

4.2 Models for Comparison

As discussed earlier, DQG is still underexplored so far, and there are few existing baselines for our comparison. We compare the generation results with different neural network models, among which SGGDQ (Pan et al., 2020) is a DQG model, while the others are for conventional QG tasks. We choose the following QG models due to their high relevance with our task, and change their settings to fit our senario.

S2S-Att[1] (Bahdanau et al., 2015): It is a Seq2Seq model with the attention mechanism. We connect the document with the answer as the input of the encoder.

NQG[2] (Zhou et al., 2017): It is a Seq2Seq model with a feature-rich encoder to encode answer position, POS and NER tag information.

s2s-mcp-gsa[3] (Zhao et al., 2018): It proposes a maxout pointer mechanism with a gated self-attention encoder to address the challenges of processing long text inputs for question generation.

ASs2s-a[4] (Kim et al., 2019): It proposes an answer-separated Seq2Seq model with a new module termed keyword-net, which better utilizes the information from both the passage and the target answer to generate an appropriate question.

SemQG[5] (Zhang and Bansal, 2019): It proposes two semantics-enhanced rewards obtained from downstream question paraphrasing and question answering tasks to regularize the QG model to generate semantically valid questions.

SGGDQ[6]: It constructs a semantic-level graph for the input document, then use the document-level and graph level representations to perform joint training of content selection and question decoding.

4.3 Evaluation Metrics

Automatic Evaluation In previous work, BLEU (Papineni et al., 2002), ROUGE (Lavie and Agarwal, 2007) and METEOR (Lin, 2004) have been widely used to evaluate the overall performance of question generation. Therefore, in order to make a fair comparison with the existing methods, we use the same automatic evaluation metrics. Initially, BLEU and METEOR are used to evaluate machine translation systems (Papineni et al., 2002; Lin, 2004), and ROUGE-Lis used to evaluate text summarization systems (Lavie and Agarwal, 2007). We use them to evaluate the similarity between the generated questions and references.

Human Evaluation In order to evaluate the effect of our model more intuitively, we also conducted a manual evaluation to check the quality of the question generated by the model. We have designed three evaluation criteria: 1) **Naturalness**, a metric which indicates the grammaticality and fluency of the generated question. 2) **Complexity**, a metric which measures difficulty of answering the generated question. 3) **Relevance**, a metric which is a measure of how relevant the generated question is to the answer. Five well-educated annotators were asked to rate the generation on a scale of one to five according to the three criteria, with five indicating the best results.

4.4 Results and Analysis

4.4.1 Comparisons with Baseline Models

Table 1 shows the overall experimental automatic evaluation results of our model and baselines on the HotpotQA dataset. We also have some observations as follows:

- ADDQG has achieved significant improvements of 2.01, 0.41, 1.15 points in terms of BLEU-4, METEOR and ROUGE-L respectively compared to the best baseline SGGDQ. It has made great progress in BLEU-4, which may be the contribution of reinforcement learning (regarded as a reward). The BASE model (similar to SGGDQ, but without the support of answer information)

[1] https://github.com/OpenNMT/OpenNMT
[2] https://github.com/magic282/NQG
[3] https://github.com/seanie12/neural-question-generation
[4] https://github.com/yanghoonkim/NQG_ASs2s
[5] https://github.com/ZhangShiyue/QGforQA
[6] https://github.com/YuxiXie/SG-Deep-Question-Generation

Dataset	HotpotQA					
Model	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**ROUGE-L**
S2S-Att (Bahdanau et al., 2015)	32.23	20.36	14.68	11.40	16.88	32.30
NQG (Zhou et al., 2017)	35.51	22.32	15.94	11.73	16.79	32.12
s2s-mcp-gsa (Zhao et al., 2018)	38.54	25.09	17.49	13.48	18.73	33.45
ASs2s-a (Pan et al., 2019)	37.67	23.79	17.21	12.59	17.45	33.21
SemQG (Zhang and Bansal, 2019)	39.92	26.73	18.73	14.71	19.29	35.63
SGGDQ (DP) (Pan et al., 2020)	40.55	27.21	20.13	15.53	20.15	36.94
Our Model						
BASE	41.17	27.64	20.47	15.81	19.07	37.24
w/AAI(Answer-Aware Initialization Module)	41.99	28.13	20.81	16.11	19.85	37.17
w/SRF(Semantic-Rich Fusion Attention)	43.12	29.84	21.62	17.07	20.24	37.52
w/RL(Reinforcement Learning for Fine-Tuning)	42.03	28.26	20.95	16.21	19.86	37.26
ADDQG	**44.34**	**31.32**	**22.68**	**17.54**	**20.56**	**38.09**

Table 1: The ROUGE, BLEU and METEOR scores of different methods om the HotpotQA dataset.

	NQG	s2s-mcp-gsa	ASs2s-a	SemQG	SGGDQ	ADDQG
Naturalness	2.65	3.34	2.89	3.75	3.83	**4.28**
Complexity	2.46	3.56	2.43	4.01	3.96	**4.47**
Relevance	1.94	2.97	2.13	2.94	3.25	**4.29**
Average score	2.35	3.29	2.48	3.57	3.68	**4.35**

Table 2: Human evaluation results of ADDQG compare with baseline models, where 1 is the worst and 5 is the best.

comes close to the performance of the previous state-of-the-art model SGGDQ, which suggests that the answer embedding method in SGGDQ has much substantial effect.

- We achieve an average improvements of 2.83, 1.27, 1.15 points in terms of BLEU-4, METEOR and ROUGE-L respectively compared with the SemQG model. We both use reinforcement learning to fine-tune the question generation model but the reward of ADDQG is easier to train. Our question generation framework is better than s2s-mcp-gsa model, which reveals our semantic-rich fusion attention works better for document level information processing.
- For ASs2s-a and ADDQG, both of them encode the answer and document separately, we achieve an average improvements of 4.95, 3.11, 4.88 points in terms of BLEU-4, METEOR and ROUGE-L respectively, which suggests the major difference between them is that our model architecture is more suitable for complex question generation. ADDQG has greatly outperformed the S2S-A and NQG models, which also shows the limitations of the S2S-A and NQG models for deep question generation.

Table 2 illustrates the results of human evaluation. ADDQG significantly outperforms all baselines in terms of three metrics, especially in terms of **relevance**, which shows that ADDQG makes better use of the answer information to generate answer-focused questions. We further discuss the effects of these modules in Ablation Studty.

4.4.2 Ablation Study

The ablation experimental results on the HotpotQA dataset are listed in Table 1. We analyze the detailed impact of each module as follows:

w/AAI The answer-aware initialization (AAI) module helps the BASE model to increase by 0.30 in BLEU-4, 0.78 in METEOR. This module introduces both document and answer information to the decoder for initialization, which helps the model guide the choice of question words.

w/SRF The semantic-rich fusion attention (SRF) module has brought an average improvements of 1.26 in BLEU-4, 1.17 in METEOR and 0.28 in ROUGE-L, which contributes the most to the good performance of ADDQG compared to the other modules. This module incorporates the answer information

Document A: The 1974 Texas Tech Red Raiders football team represented Texas Tech University in the Southwest Conference during the 1974 NCAA Division I football season.
Document B: Texas Tech University, often referred to as Texas Tech, Tech, or TTU, is a public research university in Lubbock, Texas.
Answer: Texas Tech University

Reference Question: The 1974 Texas Tech Raiders football team represented what public research university in Lubbock, Texas?

SGGDQ: Texas tech red raiders football team represented where university in lubbock?

w/AAI: What university in lubbock, texas, 11 texas tech red raiders football team represent?
w/SRF: The 1974 Texas Tech Red Raiders football team represented what university ?
w/RL: Texas Tech Raiders football team represented what public university in Lubbock?
ADDQG: The 1974 Texas Tech Raiders football team represented what public research university in Lubbock?

Figure 3: Example of questions generated by ADDQG. We also reproduce the SGGDQ model for comparative analysis.

with the document information into the generation of questions, which promotes the proper handling of answer information during question generation.

w/RL With the help of reinforcement learning, the model has made an improvements of 0.4 in BLEU-4, 0.79 in METEOR and 0.02 in ROUGE-L. Reinforcement learning integrates both syntactic and semantic metrics to enhance the training process.

Table 1 also indicates that our proposed three methods help to bring improvements to the performance of the BASE model obviously, and the combination of them further helps the hybrid model (ADDQG) to achieve state-of-the-art performance.

4.4.3 Case Study

In this section, we present examples generated by our model and SGGDQ model for comparison in Figure 3. As can be seen from the figure, SGGDQ model misses part of semantic information (like "The 1974") and selects the wrong question word ("where" should be replaced with "what"). **w/AAI**, **w/SRF** and **w/RL** all have selected the right question word "what" that is consistent with the reference question, which show our models make better use of the target answer as a guidance to facilitate question generation. The question generated by ADDQG is the most close to the reference question, which further demonstrates the effectiveness of the designs.

5 Conclusion and Future Work

Deep question generation aims to generate complex questions that require reasoning over multiple pieces of information. In this paper, we propose an answer-driven end-to-end deep question generation model (ADDQG) based on reinforcement learning. An answer-aware initialization module with a gated connection layer and a semantic-rich fusion attention mechanism are designed to incorporate document and answer information into the generation process. Reinforcement learning is further applied to integrate both syntactic and semantic metrics as the reward to enhance the training of ADDQG. Experiments show that ADDQG outperforms the state-of-the-art systems on the challenging DQG dataset. Ablation studies have demonstrated the effectiveness of our designs, and human evaluations show that our model can produce more coherent and answer-focused questions.

Future research can be carried out in several directions. First, we will try deep graph convolutional encoders (Diego Marcheggiani, 2018; Guo et al., 2019) to get deeper semantic information and explore more elaborate mechanisms for the integration of document and answer information. Second, we will apply a pre-trained multi-hop question answering model to generate the reward to optimize ADDQG, thus further enhancing the reasoning ability of this DQG model.

Acknowledgements

We thank the reviewers for their helpful comments. This research is supported by National Natural Science Foundation of China (Grant No. 61773229), Shenzhen Giiso Information Technology Co. Ltd., the Basic Research Fund of Shenzhen City (Grand No. JCYJ20190813165003837), Tencent AI Lab Rhino-Bird Focused Research Program (No. JR202032) and Overseas Cooperation Research Fund of Graduate School at Shenzhen, Tsinghua University (Grant No. HW2018002).

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*.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural NLG from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844, Florence, Italy, July. Association for Computational Linguistics.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4026–4032. Association for Computational Linguistics.

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.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 593–602. Association for Computational Linguistics.

Laura Perez-Beltrachini Diego Marcheggiani. 2018. Deep graph convolutional encoders for structured data to text generation. In Emiel Krahmer, Albert Gatt, and Martijn Goudbeek, editors, *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 1–9. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1342–1352.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251.

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.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In Chris Callison-Burch, Philipp Koehn, Cameron S. Fordyce, and Christof Monz, editors, *Proceedings of the Second Workshop on Statistical Machine Translation, WMT@ACL 2007, Prague, Czech Republic, June 23, 2007*, pages 228–231. 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, July. Association for Computational Linguistics.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 289–297.

Karen Mazidi and Rodney D Nielsen. 2014. Linguistic considerations in automatic question generation. In *ACL (2)*, pages 321–326.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018. Efficient and robust question answering from minimal context over documents. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1725–1735. Association for Computational Linguistics.

Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019. Reinforced dynamic reasoning for conversational question generation. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2114–2124. Association for Computational Linguistics.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. *CoRR*, abs/2004.12704.

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.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 646–653. 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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1179–1195. IEEE Computer Society.

Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 3–21. Springer.

Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *CoRR*, abs/1809.02040.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *CoRR*, abs/1706.02027.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9073–9080. AAAI Press.

Luu Anh Tuan, Darsh J. Shah, and Regina Barzilay. 2020. Capturing greater context for question generation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9065–9072. AAAI Press.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *CoRR*, abs/1710.10903.

Weichao Wang, Shi Feng, Daling Wang, and Yifei Zhang. 2019. Answer-guided and semantic coherent question generation in open-domain conversation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5065–5075. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2495–2509. 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, Brussels, Belgium, October 31 - November 4, 2018*, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In Xuanjing Huang, Jing Jiang, Dongyan Zhao, Yansong Feng, and Yu Hong, editors, *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.