Learning to Generate Question by Asking Question: A Primal-Dual Approach with Uncommon Word Generation

Qifan Wang[1], Li Yang[2], Xiaojun Quan[3], Fuli Feng[4], Dongfang Liu[5], Zenglin Xu[6], Sinong Wang[1] and Hao Ma[1]

[1]Meta AI [2]Google Research [3]Sun Yat-sen University
[4]University of Science and Technology of China
[5]Rochester Institute of Technology [6]Harbin Institute of Technology
wqfcr@fb.com lyliyang@google.com

Abstract

Automatic question generation (AQG) is the task of generating a question from a given passage and an answer. Most existing AQG methods aim at encoding the passage and the answer to generate the question. However, limited work has focused on modeling the correlation between the target answer and the generated question. Moreover, unseen or rare word generation has not been studied in previous works. In this paper, we propose a novel approach which incorporates question generation with its dual problem, question answering, into a unified primal-dual framework. Specifically, the question generation component consists of an encoder that jointly encodes the answer with the passage, and a decoder that produces the question. The question answering component then re-asks the generated question on the passage to ensure that the target answer is obtained. We further introduce a knowledge distillation module to improve the model generalization ability. We conduct an extensive set of experiments on SQuAD and HotpotQA benchmarks. Experimental results demonstrate the superior performance of the proposed approach over several state-of-the-art methods.

1 Introduction

Question answering (Hsu et al., 2021) plays a crucial role in both the growth of human beings and the improvement of artificial intelligent systems. As a dual task of question answering, automatic question generation (AQG) (Cheng et al., 2021) based on a passage and a target answer has attracted much attention in recent years. One of its key applications is to generate questions for educational materials (Heilman and Smith, 2010). Another application is automatically producing question-answer pairs to enhance machine reading comprehension systems (Du et al., 2017; Lyu et al., 2021). Besides, AQG is also widely used in building web answering system (Shou et al., 2020; You et al., 2021; Wang et al., 2022), conversational dialog systems (Liu et al., 2021; Shen et al., 2021; Huang et al., 2022) and chatbots (Gros et al., 2021) such as Siri, Cortana, Alexa and Google Assistant, helping them to start and continue a conversation with human users.

Passage: As of 2012, quality private schools in the United States charged substantial tuition, close to $40,000 annually for day schools in New York City, and nearly $50,000 for boarding schools. However, tuition did not cover operating expenses, particularly at boarding schools.
Human: What would a parent have to pay for their child to attend a boarding school in 2012?
RefineNet: How much money for day schools in the United States in 2012?
QG+SSM+API: How much money for boarding schools charged substantial tuition?
Our model: How much substantial tuition is charged to *attend* boarding schools in New York City?

Figure 1: An example of generated questions from human, base models and our model. The purple text in the passage indicates the target answer. Our model generates a more desirable question compared to the questions generated from the base models. Our model is able to generate the uncommon word, *attend*, that does not appear in the passage.

Automatic question generation is a challenging task due to the unstructured nature of textual data. Early research (Rus et al., 2010; Labutov et al., 2015) on AQG focuses on generating questions that are grammatically correct and answerable from the passage, but not specific to any answer in the passage. There are fundamental limitations of these methods as they are not able to produce useful question-answer pairs for downstream tasks. For example, in a conversational dialog system, the next question should be generated according to the user's previous answers or conversations but not just the current context. On the other hand, specifying the answer is necessary for generating natural questions because there could be multiple target answers in the passage. For example, in Figure 1, there are various candidate questions to be asked on the passage, such as the city "New York City", the country "United States", and the year "2012".

Recent AQG models incorporate the target an-

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 46–61
December 7-11, 2022 ©2022 Association for Computational Linguistics

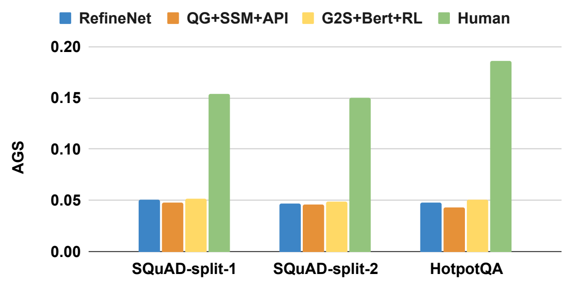

Figure 2: A preliminary study on uncommon word generation. It can be seen that there is a large gap between existing methods and human in terms of AGS values, where uncommon words appear much more often in human generated questions.

swer information in generating the question (Klein and Nabi, 2019; Liu et al., 2019; Chai and Wan, 2020; Huang et al., 2020; Fei et al., 2021). These approaches focus on encoding the passage, the answer and their correlation using complex networks and then generate the question with a decoder. However, there are two main limitations:
First, most of these methods generate the question in one single pass, and do not explicitly model the correlation between the target answer and the generated question. A natural question arises: would the target answer be retrieved when asking the generated question on the passage? For instance, in Figure 1, the question generated by the QG+SSM+API model is grammatically correct but is not answerable. On the other hand, the generated question from RefineNet is both grammatically correct and answerable, but not specific to the answer. If asking both generated questions on the passage, it is clear that the target answer will not be obtained.
Second, very limited research has focused on new or unseen word generation. However, these uncommon words not only increase the diversity of the generated questions, but also improve the question quality in terms of *Naturalness* and *Answerability*. For example, in Figure 1, the word "attend" does not appear in the input passage, which is also uncommon in the training data. But it is an important word in this case as people use the phrase "attend school" naturally in their daily conversations. A quantitative study of uncommon words generation is shown in Figure 2. We calculate the average generalizability score (AGS) using a normalized IDF (inverse document frequency) metric as $GS(Q) = \max_{w_t^q \notin P} \frac{1}{1+log(1+DF(w_t^q))}$. Here $DF(w_t^q)$ is the document frequency indicating how many training questions contain the word w_t^q. Intuitively, AGS measures the rarity of the words in the generated questions.

In this paper, we propose a novel primal-dual approach, Question Generation by Asking Question with Uncommon word Generation (QG+AQ+UG), which integrates question generation with its dual problem, question answering, into a unified learning framework. In particular, the question generation component consists of an encoder that encodes the answer with the passage, and a decoder that produces the question based on the output of the encoder. The question answering component, which shares the same encoder, then asks the generated question on the passage to ensure that the target answer is obtained. A knowledge distillation module is introduced for better uncommon word generation, which transfers the knowledge from a large pre-trained model to the primal-dual framework. We conduct an extensive set of experiments on the SQuAD and HotpotQA benchmarks, which shows superior performance of the proposed approach over several state-of-the-art methods. We summarize the main contributions as follows:

- We propose a novel primal-dual approach for automatic question generation, which integrates the primal problem of question generation and its dual problem of question answering into a unified framework.
- We introduce a knowledge distillation module into the primal-dual learning framework, which helps generate those uncommon, yet important, words. Uncommon words generation improves both the diversity and the quality of the generated questions.
- We conduct extensive experiments and demonstrate the effectiveness of the proposed approach over several state-of-the-art baselines.

2 Methodology

2.1 Problem Definition

In this section, we formally define the primal problem of question generation and its dual problem of question answering. We denote the passage as $P = (w_1^p, \ldots, w_n^p)$ and the target answer as $A = (w_1^a, \ldots, w_m^a)$. In most cases, the answer comes from a span in the passage, with its begin and end indices b and e, i.e., $w_1^a = w_b^p$ and $w_m^a = w_e^p$. The task of question generation is to generate a question $Q = (w_1^q, \ldots, w_T^q)$ such that the target answer A

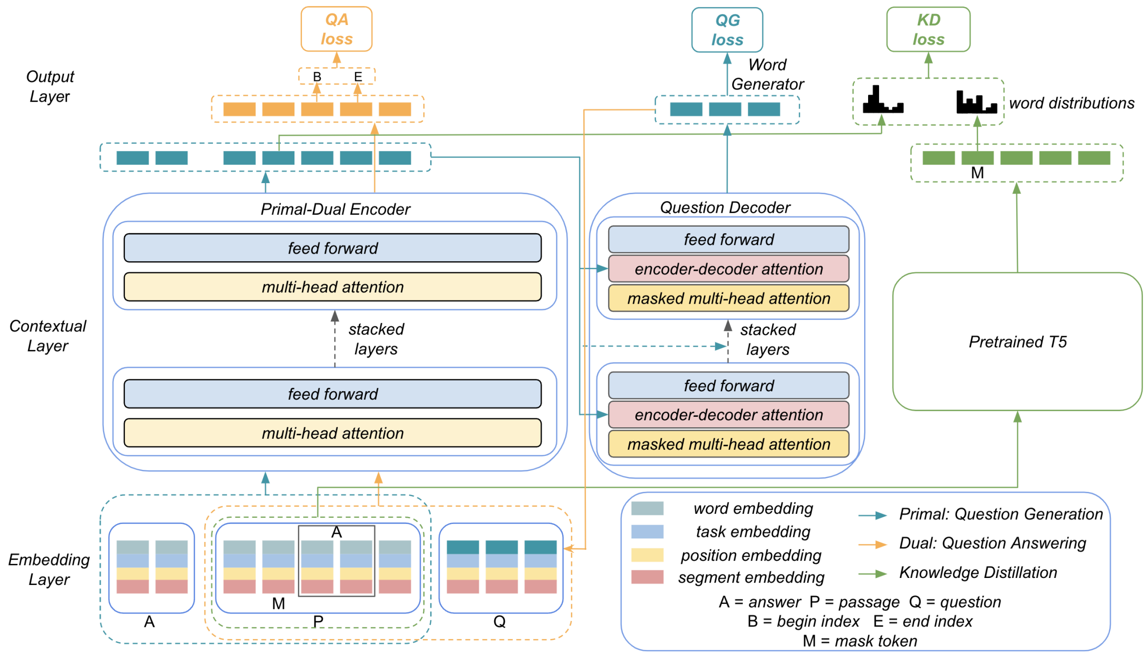

Figure 3: Our QG+AQ+UG model architecture.

will be obtained from the passage P. Formally, the question generation problem is defined as finding the best $\bar{Q}$ that maximizes the conditional likelihood given P and A:

$$\bar{Q} = \arg\max_{Q} \prod_{t=1}^{T} Pr(w_t^q | w_{t-1}^q, \ldots, w_1^q, P, A)$$

The dual problem is defined as finding the best answer span $\bar{A}$ given the passage P and the question Q:

$$\bar{A} = \arg\max_{b,e} \; Pr(\, w_b^p, \; w_e^p \mid P, \; Q)$$

2.2 Approach Overview

The overall model architecture is shown in Figure 3. Essentially, our primal-dual model consists of three main components, the primal question generation module (QG), the dual question answering module that asks question (AQ) and the knowledge distillation module for uncommon words generation (UG). The question generation module consists of four blocks, the embedding layer, the contextual encoder, the decoder and the output layer. The question answering module aims at finding the best answer span from the passage to answer the generated question, which shares a unique encoder with the primal question generation module. The knowledge distillation module utilizes a distilled masked language model to enhance the model generalization ability for uncommon words generation.

2.3 Embedding Layer

The first component in the primal-dual model is the embedding layer. In the embedding layer, every word in the passage, answer and question is converted into a d-dimensional embedding vector. The final embedding is achieved by concatenating a word embedding, a task specific embedding, a positional embedding and a segment embedding. The word embedding is widely adopted in the literature (Devlin et al., 2019). Inspired by T5 (Raffel et al., 2020), the task embedding is adopted to identify which task this input belongs to, i.e., question generation, question answering or knowledge distillation. The positional embedding is introduced to model the order information of the sequence. We use the absolute position of the words in the sequence in our implementation. The segment embedding is added to indicate which source the word belongs to, i.e., passage, answer or question. In contrast to previous methods, all the embeddings in our approach are trainable. These embeddings are only initialized from the pretrained language models, and are updated during training.

2.4 Primal-Dual Encoder

We employ a unique encoder that is shared across the primal-dual framework after the embedding layer. The encoder is essentially a contextual layer, which generates contextualized representation for every word from their embeddings in the input sequence. In the question generation module, the input sequence to the encoder is the concatenation of the target answer and the passage. The output is a sequence of contextual embeddings representing the encoded answer and passage (as shown in Figure 3). Similarly, in the question answering module, the encoder produces another sequence of contextual embeddings for the passage and the question. Different from previous multi-task learning, which uses the ground-truth questions in questions answering. In our model, we directly optimize over the generated question embeddings by feeding it into the question answering module.

Most of the existing question generation models (Li et al., 2019; Nema et al., 2019; Tuan et al., 2020; Huang et al., 2021) use two separate encoders for the passage and the answer respectively, followed by a cross-attention layer to merge the output embeddings of the two encoders. Inspired by the recent advancements in BERT (Devlin et al., 2019) model, we introduce one unique encoder with self-attention mechanism, which allows different input segments, i.e., the answer, the passage and the question, to attend each other from the bottom layer to the top layer. In particular, the encoder is a stack of identical layers using multi-head attention (MHA) and feed forward network (FFN). The output of the top encoder layer will be used as the contextual embeddings of the input sequence.

One of the key ingredients in the primal-dual architecture is that both the primal and dual modules share a unique contextual encoder, and thus are able to benefit from each other. This encoder essentially bridges the primal problem of question generation with its dual problem of question answering, by jointly learning a unique set of parameters to produce the contextual embeddings.

2.5 Question Decoder

The decoder decodes the embeddings from the encoder to generate the question embedding. We adopt the similar decoder structure in Transformer (Vaswani et al., 2017), which is composed of a stack of identical layers. In addition to the two sub-layers in the encoder layer, the decoder employ a third sub-layer, encoder-decoder attention (EDA), which performs multi-head attention over the outputs of the encoder and the current decoder layer. The masked multi-head attention (MMHA) has the same model structure as the multi-head attention in the encoder layer, except that it prevents positions from attending to subsequent positions. This masking ensures that the prediction for position i can depend only on the known outputs at positions before i, as the question is generated word by word. The encoder-decoder attention has a similar structure as self-attention, the distinction is that the key and value are from the output of the encoder, whereas the query is from the decoder itself. We provide more technical details of both encoder and decoder in Appendix A.

2.6 Output Layer

The output layer of the question generation model is essentially a word generator, which consumes the embedding of the decoder and generates the question word by word:

$$\bar{w}_t^q = \arg\max_{w_t^q}(softmax(W_o H_{de}^t))$$

where H_{de}^t is the decoder output at word position t. W_o is the output matrix which projects the final embedding to the logits of vocabulary size. We further employ a copy mechanism or pointer network (See et al., 2017) to allow both copying words from input via pointing, and generating words from a predefined vocabulary during decoding. In this work, we adopt the pointer-network and coverage mechanism from (Zhao et al., 2018) to handle out-of-vocabulary words and to avoid repeating phrases in the generated questions.

The output layer of the question answering model extracts the final answer span by calculating the probabilities for the begin and end indices. We apply a softmax function on the output embeddings to generate the probabilities of begin index:

$$P_b = softmax(W_b H_{en})$$

Inspired by the recent work (Yang et al., 2019), we further predict the end index based on the start index by concatenating the begin token embedding with every token embedding after it:

$$P_e = softmax(W_e(concat(H_{en}, H_{en}^{\bar{b}})))$$

where $\bar{b}$ is the best begin index with max probability from P_b. H_{en} is the contextual embedding vector of the input sequence. W_b and W_e are two parameter matrices that project the embeddings to the output logits, for the begin and end.

2.7 Knowledge Distillation

Model generalization is one of the important factors for evaluating a question generation model, which measures the model ability of generating new and uncommon words. Generating new and uncommon words not only enhance the diversity of the generated questions, but also improves the question quality and makes the question more natural and answerable. By analyzing the generated questions from the existing models, we observe that many important words from the ground-truth questions can not be generated. The reason is that these words are not present in the passages which prevents the copy mechanism for copying these words to the question. They also rarely appear in the training questions, which are used for model training. For example, the word "attend" (from the human generated question in Figure 1) does not appear in the passage or other training data. Moreover, it is possible for a model to over memorize the training data, and thus fail to generate new words.

To address this problem, in this work, we employ a knowledge distillation (Hinton et al., 2015) module which transfers knowledge from a pretrained model to improve the model generalization ability. Intuitively, the knowledge distillation guides the encoder to learn effective contextual embeddings for new words, through masking them out and enforcing the consistency between the two distributions generated from the encoder and the pretrained model. In this way, the learned contextual knowledge/information of the new words is transferred from the pretrained model to the primal-dual encoder, and thus improves the model generalization. The knowledge distillation model minimizes the cross entropy between the word probability distributions generated from the primal-dual encoder and the pretrained model. A knowledge distillation loss is used to measure the distribution difference. The knowledge distillation loss is a modified cross entropy loss which is defined as:

$$CE(Y_{en}, Y_{pre}) = -\sum_{t=1}^{S} y_{en}^{t} \log y_{pre}^{t}$$

$$Y = softmax(W_m H_M)$$

where Y_{en} and Y_{pre} are the two probability distributions, on the masked word, generated by the primal-dual encoder and the pretrained model respectively. S is the vocabulary size. y^t is the probability of the t-th word in the vocabulary. H_M is the output embedding of the masked word. W_m is the output matrix which maps the output embedding to the logits of vocabulary size. The T5 (Raffel et al., 2020) model, pretrained over the Wikipedia + Toronto Books Corpus and WebText, is used as the pretrained model. In our implementation, we randomly mask 10% of the verb tokens in the passage, since we observe that many verbs are very specific to their passages and are uncommon in the training data. We also conduct random masking on all tokens. More detailed discussion on the impact of the knowledge distillation is provided in the experiments. The overall objective of our primal-dual framework is $\mathcal{L}_{total} = \mathcal{L}_{QG} + \alpha\mathcal{L}_{QA} + \beta\mathcal{L}_{KD}$, where α and β are trade-off parameters to balance the losses.

3 Experiments

3.1 Datasets

SQuAD (Rajpurkar et al., 2016): The original SQuAD dataset contains 23215 paragraphs from 536 Wikipedia articles with over 100k questions posed about the articles. The answer is also given with corresponding questions as the sub-span of the sentence. In order to conduct a fair comparison, we use the same two processed versions of SQuAD that are used by previous works (Song et al., 2018; Tuan et al., 2020). It is divided into train/dev/test splits with two different divisions, resulting in **SQuAD-split-1** and **SQuAD-split-2**.

HotpotQA (Yang et al., 2018): Hotpot-QA is a multi-document and multi-hop QA dataset. This dataset contains supporting facts that potentially lead to the answer. We concatenate these supporting facts to form the passage. We use 10% of the training data for validation and use the original dev set as the test set. The details of these datasets are given in Appendix C.

3.2 Implementation Details

We implemented our models using Tensorflow and Keras. Each model is trained on a 32 core TPU v3 configuration. Our model is randomly initialized. It uses 12 layers, 768 hidden size, 12 heads and 3072 hidden units (for FFN) for both encoder and decoder. The maximum sequence length is set to 512. The BERT-base vocab with size 30,522 is used. During training, we use the gradient descent algorithm with Adam optimizer. The initial learning rate is set to $3e^{-5}$. The mini-batch size for each update is set as 64 and the model is trained for up to 9 epochs. The dropout probability for the attention layer is set to 0.1. The hyper-parameters α is set to 0.8, with β set to 0.15. For testing, we conduct

Models	SQuAD-split-1			SQuAD-split-2			HotpotQA		
	BLEU	METEOR	ROUGE	BLEU	METEOR	ROUGE	BLEU	METEOR	ROUGE
s2s+MP+GSA (Zhao et al., 2018)	-	-	-	15.82	19.67	44.24	19.29	19.29	40.40
ASs2s (Kim et al., 2019)	16.20	19.92	43.96	16.17	-	-	-	-	-
QG+pc (Li et al., 2019)	16.27	20.36	44.35	16.37	20.68	44.73	-	-	-
RefineNet (Nema et al., 2019)	-	-	-	16.84	20.60	44.78	19.68	23.27	41.52
QG+SSM+API (Ma et al., 2020)	-	-	-	16.32	20.84	44.79	-	-	-
QG+AP (Wang et al., 2020a)	-	-	-	16.42	18.95	43.07	-	-	-
QG+QA (Sun et al., 2020)	16.36	20.15	44.64	16.67	20.33	44.80	19.73	23.45	41.65
Multi-stage Att (Tuan et al., 2020)	17.09	21.25	45.81	17.76	21.56	46.02	-	-	-
G2S+Bert+RL (Chen et al., 2020)	17.94	21.76	46.02	18.30	21.70	45.98	-	-	-
QG+AQ+UG (ours)	**19.07**	**22.62**	**46.89**	**19.34**	**22.95**	**46.97**	**22.38**	**25.85**	**44.51**

Table 1: Performance comparison results. We directly import the results of the baselines that reported on these datasets. A '-' means they do not evaluate on that dataset. Results are statistically significant with p-value < 0.001.

beam search with beam width 10 and length penalty weight 2.1. Decoding stops when generating the <EOS> token.

We evaluate the performance of our model with three standard evaluation metrics: **BLEU** (BLEU-4), **METEOR** and **ROUGE-L**. We use the evaluation package released in (Sharma et al., 2017). We repeat each experiment 10 times and report the metrics based on the averages.

3.3 Baselines

s2s+MP+GSA (Zhao et al., 2018) uses a gated self-attention into the encoder and a maxout pointer mechanism into the decoder.

ASs2s (Kim et al., 2019) replaces the answer in the sentence with a special token to avoid its appearance in the questions.

QG+pc (Li et al., 2019) models the unstructured sentence and the structured answer-relevant relation for question generation.

RefineNet (Nema et al., 2019) augments the basic encoder-decoder model with a reward based refinement network.

QG+SSM+API (Ma et al., 2020) employs sentence-level semantic matching and answer position inferring.

QG+AP (Wang et al., 2020a) treats the answers as hidden pivots and combines question generation with answer selection.

QG+QA (Sun et al., 2020) using two independent encoders for the question generation and question answering tasks respectively.

Multi-stage Att (Tuan et al., 2020) represents the relevant context via a multi-stage attention mechanism to incorporate interactions across sentences.

G2S-Bert-RL (Chen et al., 2020) proposes a RL based graph to sequence model for question generation.

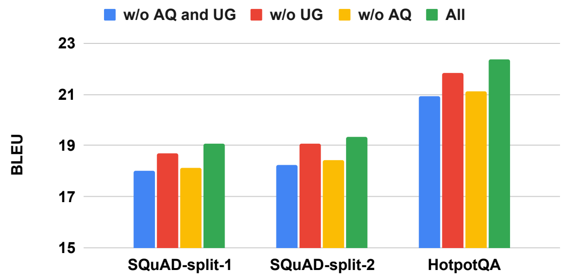

Figure 4: Ablation study on the impact of different modules on all datasets.

3.4 Main Results

Our model outperforms the state-of-the-art question generation methods on all datasets. The performance comparison results are reported in Table 1. From these comparison results, we can see that QG+AQ+UG provides the best results among all compared methods on both SQuAD splits and HotpotQA. For example, the BLEU metric of our model increases over 5.7% and 8.9% compared with G2S+Bert+RL and Mutli-stage Att on SQuAD-split-2 respectively. There are three main reasons: First, our model integrates the question generation and question answering into a unified primal-dual framework, which enforces the generated question to obtain the target answer from the passage, resulting in more accurate question generation. Second, the knowledge distillation enables our model to generate more important words which are uncommon in the training data. Third, our model employs advanced Transformer architecture, instead of bi-LSTM, in both encoder and decoder, which allows the passage and the answer to attend each other from bottom to top, resulting in better contextual embeddings.

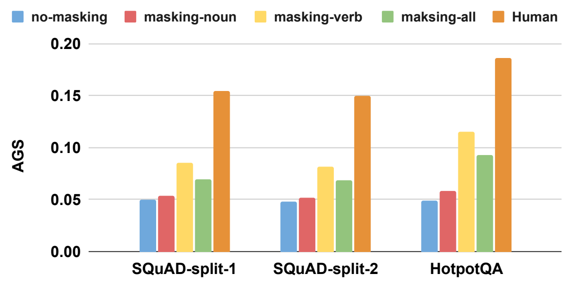

Figure 5: AGS results of different masking strategies in knowledge distillation.

	Naturalness	Answerability
RefineNet	3.65	3.62
G2S+Bert+RL	3.59	3.77
QG+AQ	3.79	4.05
QG+AQ+UG	3.85	4.12
Human	**4.23**	**4.47**

Table 2: Human evaluation results.

4 Analysis and Discussion

4.1 Ablation Study

Asking question (AQ) plays an important role in question generation, while uncommon words generation (UG) also helps improve the model performance. To evaluate the effectiveness of the primal-dual approach, we conduct a set of ablation studies by removing each component individually, i.e. question generation, question answering and knowledge distillation, from our model. We also train a model removing both question answering and knowledge distillation. The BLEU scores of these methods on all datasets are shown in Figure 4. It can be seen from the figure that both question answering and knowledge distillation contribute to improving the effectiveness of question generation, especially the dual question answering module, which validates the effectiveness of primal-dual modeling and the uncommon words generation. Nevertheless, it is clear from these results that the QG+AQ+UG model, which incorporates all three components, achieves the best performance.

4.2 Impact of Knowledge Distillation on Uncommon Words Generation

Knowledge distillation guides the model to generate uncommon words. Masking verb tokens is the most effective strategy compared with masking nouns and masking all words. To understand the effect of knowledge distillation on uncommon word generation, we compare different masking strategies when applying knowledge distillation. Specifically, recall that in our original implementation of knowledge distillation, we conduct random masking on verb tokens. In this study, we also conduct three other variations. (1) Without masking (which is equivalent to dropping knowledge distillation). (2) Random masking on all words. (3) Random masking on noun tokens. We calculate the average generalizability score (AGS) as described in the introduction section. The AGS results are shown in Figure 5. It is clear that both masking-all and masking-verb improve the AGS significantly compared to no-maksing on all datasets, which demonstrates the effectiveness of knowledge distillation on uncommon word generation. Moreover, we observe that masking only verbs achieves even higher scores than random word and noun word masking. Our hypothesis is that verbs are likely to be specific and are uncommon in the training questions. On the other hand, nouns or entities usually appear in the passage which can be copied directly to the generated questions through copy mechanism.

4.3 Human Evaluation

Our primal-dual model (QG+AQ) improves both the naturalness and answerability of the generated questions, while uncommon words generation further improves quality of the questions. We conduct a human evaluation to measure the quality of questions generated by our approach. Specifically, similar to the metrics used in (Du et al., 2017), we consider two criterion in human evaluation: (1) Naturalness, which indicates the grammaticality and fluency of the generated questions; and (2) Answerability, which measures the correctness of the question, i.e., whether it can achieve the answer. We randomly sample 100 (answer,passage,question) triples from our SQuAD-split-2 experimental outputs. We then ask three professional English speakers to rate the pairs in terms of the above criterion on a 1 to 5 scale (5 for the best). The experimental result is reported in Table 2. The results imply that our model can generate questions of better quality than the base models, especially in terms of answerability. By comparing QG+AQ+UG with QG+AQ, it is clear that the knowledge distillation module further improves the quality of the questions.

Models	SQuAD-split-1	SQuAD-split-2	HotpotQA
QG+AQ+UG-LSTM	18.07	18.36	21.55
QG+AQ+UG-small	18.41	18.80	21.93
QG+AQ+UG-base	19.07	19.34	22.38
QG+AQ+UG-large	**20.13**	**20.56**	**23.47**

Table 3: BLEU results over different models and configurations on all datasets.

Passage: Another example was the insignificance of the Ministry of War compared with native Chinese dynasties, as the real military authority in Yuan times resided in the Privy Council.
Human: Who had military control during Yuan?
QG: Where did the *insignificance of the Ministry of war in Yuan times reside?*
QG+AQ: Where did the *real military authority in Yuan times reside?*
QG+AQ+UG: Who *controls the real military authority in Yuan times*?

Figure 6: A case study of generated questions.

4.4 Different Model Configurations

We evaluate the performance of our model on different encoder/decoder configurations. We conduct experiments with two additional configurations - the large one with 24 layers, 1024 hidden size, 16 heads and 4096 hidden units, and a small model with 6 layers, 256 hidden size, 8 heads and 1024 hidden units. The total number of parameters of the QG+AQ+UG small, base and large models are 98m, 225m and 647m. The BLEU results on all datasets are shown in Table 3. It can be observed that QG+AQ+UG-large achieves better performance, which is consistent with our expectations. However, a larger model usually requires longer training time, as well as inference.

4.5 Case Study

Figure 1 shows an example of generated questions from humans, base models and our model. It can be seen that base models generate inaccurate questions, which are not able to obtain the target answers. Our model generates more accurate and desirable questions. For instance, RefineNet is not able to identify the correlation between the answer "$50000" and the words "boarding schools", resulting in a wrong question. The QG+SSM+API model generates a question that is not very natural and fluent. Our model effectively connects the semantic-related phrases "tuition" and "boarding schools" in two different sentences, and forms a relevant context for generating the question.

Figure 6 shows an example of the generated questions using QG, QG+AQ and QG+AQ+UG. It is clear that the question generated by QG is grammatically correct but not answerable. In contrast, the QG+AQ model is able to connect the phrases "the Privy Council" with "the real military authority" and "reside", which generates the desired question. Furthermore, the QG+AQ+UG is able to generate word "control" which does not appear in the passage. It demonstrates the capability of generalization of our model for uncommon words.

5 Related Work

Early works on automatic question generation are essentially rule based systems (Lindberg et al., 2013; Mazidi and Nielsen, 2014; Labutov et al., 2015). Several AQG models have been proposed to generate questions from the passage alone (Du and Cardie, 2017; Yao et al., 2018). These methods usually aim at generating questions that are grammatically correct and answerable from the passage, but not specific to any answer in the passage, which have fundamental limitations for downstream tasks. Recent models for AQG are based on the encode-attend-decode paradigm and they generate questions from the passage and a target answer (Xia et al., 2017; Wang et al., 2019a,b; Yu et al., 2020).

Over the past few years, several variants (Duan et al., 2017; Scialom et al., 2019; Wang et al., 2020c; Ko et al., 2020; Jia et al., 2021) of the encode-attend-decode model have been proposed. To generate more plausible questions, Zhou et al. (Zhou et al., 2017) utilize answer positions to make the model aware of the target answer. Song et al. (Song et al., 2018) apply the multi-perspective context matching algorithm of (Wang et al., 2017b) to employ the interaction between the target answer and the passage. Both works employ a copy mechanism (Gülçehre et al., 2016) to handle the missing words. Kim et al. (Kim et al., 2019) develop an answer separation technique which masks out the answer in the passage to generate more reasonable questions. Huang et al. (Huang et al., 2021) propose an entity guided question generation model with additional question type information. There has also been some work on generating questions from images (Liu et al., 2020), knowledge bases (Reddy et al., 2017) and products (Wang et al., 2020b; Yang et al., 2022).

There are several AQG methods (Wang et al., 2017a; Yuan et al., 2017) that try to leverage both question answering and question generation. Nema et al. (Nema et al., 2019) augment the basic encoder-decoder model with a reward based refinement network, which re-evaluates the generated question in a second pass. This method requires an additional reward mechanism to obtain the fluency and answerability scores. Another closely related work is (Tang et al., 2017, 2018), which linearly combines the question generation loss with the question answering loss in a multi-task setting.

6 Conclusions

Automatic question generation is an important task in the improvement of artificial intelligent systems. In this work, we propose a novel primal-dual approach for question generation. It integrates question generation with its dual problem question answering into a unified framework. A knowledge distillation module is introduced into the framework to improve model generalization on uncommon word generation. Experimental results on two benchmarks demonstrate the effectiveness of the primal-dual modeling.

Limitations

There are several possible research directions. First, our model assumes that the length of the passage is not too large and can be easily fit into a Transformer encoder. However, there are real-world applications which require long input text sequence for generating the questions. For example, in a dialog system, the model might need all the contexts in the history from the dialog to generate a more meaningful and relevant question. Therefore, it is a practical problem to deal with long input sequence for question generation. Second, our model generates one question at a time, while there are use cases where structure questions are more preferable. In future, we also plan to investigate more along structural question generation.

References

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 268–284. Association for Computational Linguistics.

Zi Chai and Xiaojun Wan. 2020. Learning to ask more: Semi-autoregressive sequential question generation under dual-graph interaction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 225–237. 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, 2021*. OpenReview.net.

Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5968–5978. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2067–2073. 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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1342–1352. Association for Computational Linguistics.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 866–874. Association for Computational Linguistics.

Zichu Fei, Qi Zhang, and Yaqian Zhou. 2021. Iterative gnn-based decoder for question generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2573–2582. Association for Computational Linguistics.

David Gros, Yu Li, and Zhou Yu. 2021. The r-u-a-robot dataset: Helping avoid chatbot deception by detecting user questions about human or non-human identity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6999–7013. Association for Computational Linguistics.

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

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 609–617. The Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Chao-Chun Hsu, Eric Lind, Luca Soldaini, and Alessandro Moschitti. 2021. Answer generation for retrieval-based question answering systems. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4276–4282. Association for Computational Linguistics.

Guanhuan Huang, Xiaojun Quan, and Qifan Wang. 2022. Autoregressive entity generation for end-to-end task-oriented dialog. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 323–332, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Qingbao Huang, Mingyi Fu, Linzhang Mo, Yi Cai, Jingyun Xu, Pijian Li, Qing Li, and Ho-fung Leung. 2021. Entity guided question generation with contextual structure and sequence information capturing. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13064–13072. AAAI Press.

Weijing Huang, Xianfeng Liao, Zhiqiang Xie, Jiang Qian, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2020. Generating reasonable legal text through the combination of language modeling and question answering. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3687–3693. ijcai.org.

Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2021. EQG-RACE: examination-type question generation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13143–13151. AAAI Press.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6602–6609. AAAI Press.

Tassilo Klein and Moin Nabi. 2019. Learning to answer by learning to ask: Getting the best of GPT-2 and BERT worlds. *CoRR*, abs/1911.02365.

Wei-Jen Ko, Te-Yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6544–6555. Association for Computational Linguistics.

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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 889–898. The Association for Computer Linguistics.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3214–3224. Association for Computational Linguistics.

David Lindberg, Fred Popowich, John C. Nesbit, and Philip H. Winne. 2013. Generating natural language questions to support learning on-line. In *ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013, Sofia, Bulgaria*, pages 105–114. The Association for Computer 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. In *The World Wide Web Conference*, page 1106–1118.

Feng Liu, Tao Xiang, Timothy M. Hospedales, Wankou Yang, and Changyin Sun. 2020. Inverse visual question answering: A new benchmark and VQA diagnosis tool. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):460–474.

Zhongkun Liu, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Maarten de Rijke, and Ming Zhou. 2021. Learning to ask conversational questions by optimizing levenshtein distance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5638–5650. Association for Computational Linguistics.

Chenyang Lyu, Lifeng Shang, Yvette Graham, Jennifer Foster, Xin Jiang, and Qun Liu. 2021. Improving unsupervised question answering via summarization-informed question generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4134–4148. Association for Computational Linguistics.

Xiyao Ma, Qile Zhu, Yanlin Zhou, and Xiaolin Li. 2020. Improving question generation with sentence-level semantic matching and answer position inferring. 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 8464–8471. AAAI Press.

Karen Mazidi and Rodney D. Nielsen. 2014. Linguistic considerations in automatic question generation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 321–326. The Association for Computer Linguistics.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3312–3321. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Sathish Reddy, Dinesh Raghu, Mitesh M. Khapra, and Sachindra Joshi. 2017. Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 376–385. Association for Computational Linguistics.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai C. Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *INLG 2010 - Proceedings of the Sixth International Natural Language Generation Conference, July 7-9, 2010, Trim, Co. Meath, Ireland*. The Association for Computer Linguistics.

Thomas Scialom, Benjamin Piwowarski, and Jacopo Staiano. 2019. Self-attention architectures for answer-agnostic neural question generation. In *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 6027–6032. 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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Lei Shen, Fandong Meng, Jinchao Zhang, Yang Feng, and Jie Zhou. 2021. GTM: A generative triple-wise model for conversational question generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3495–3506. Association for Computational Linguistics.

Linjun Shou, Shining Bo, Feixiang Cheng, Ming Gong, Jian Pei, and Daxin Jiang. 2020. Mining implicit relevance feedback from user behavior for web question answering. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2931–2941. ACM.

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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 569–574. Association for Computational Linguistics.

Yibo Sun, Duyu Tang, Nan Duan, Tao Qin, Shujie Liu, Zhao Yan, Ming Zhou, Yuanhua Lv, Wenpeng Yin, Xiaocheng Feng, Bing Qin, and Ting Liu. 2020. Joint learning of question answering and question generation. *IEEE Trans. Knowl. Data Eng.*, 32(5):971–982.

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

Duyu Tang, Nan Duan, Zhao Yan, Zhirui Zhang, Yibo Sun, Shujie Liu, Yuanhua Lv, and Ming Zhou. 2018. Learning to collaborate for question answering and asking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1564–1574. Association for Computational Linguistics.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Bingning Wang, Xiaochuan Wang, Ting Tao, Qi Zhang, and Jingfang Xu. 2020a. Neural question generation with answer pivot. 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 9138–9145. AAAI Press.

Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. 2022. Webformer: The web-page transformer for structure information extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 3124–3133. ACM.

Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D. Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020b. Learning to extract attribute value from product via question answering: A multi-task approach. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 47–55. ACM.

Siyuan Wang, Zhongyu Wei, Zhihao Fan, Zengfeng Huang, Weijian Sun, Qi Zhang, and Xuanjing Huang. 2020c. Pathqg: Neural question generation from facts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9066–9075. Association for Computational Linguistics.

Siyuan Wang, Zhongyu Wei, Zhihao Fan, Yang Liu, and Xuanjing Huang. 2019a. A multi-agent communication framework for question-worthy phrase extraction and question generation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7168–7175. AAAI Press.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017a. A joint model for question answering and question generation. *CoRR*, abs/1706.01450.

Yutong Wang, Jiyuan Zheng, Qijiong Liu, Zhou Zhao, Jun Xiao, and Yueting Zhuang. 2019b. Weak supervision enhanced generative network for question generation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3806–3812. ijcai.org.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017b. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4144–4150. ijcai.org.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1784–1794.

Li Yang, Qifan Wang, Zac Yu, Anand Kulkarni, Sumit Sanghai, Bin Shu, Jon Elsas, and Bhargav Kanagal. 2022. MAVE: A product dataset for multi-source attribute value extraction. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 1256–1265. ACM.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

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 *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.

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.

Chenyu You, Nuo Chen, and Yuexian Zou. 2021. Mrd-net: Multi-modal residual knowledge distillation for spoken question answering. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3985–3991. ijcai.org.

Qian Yu, Lidong Bing, Qiong Zhang, Wai Lam, and Luo Si. 2020. Review-based question generation with adaptive instance transfer and augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 280–290. Association for Computational Linguistics.

Xingdi Yuan, Tong Wang, Çaglar Gülçehre, 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, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 15–25. 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. 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.

A Technical Details

A.1 Primal-Dual Encoder

As mentioned in the main paper, the primal-dual encoder is a stack of identical layers using multi-head attention (MHA) and feed forward network (FFN):

$$H_{en}^1 = \text{FFN}(\text{MHA}(E))$$
$$H_{en}^k = \text{FFN}(\text{MHA}(H_{en}^{k-1}))$$

where $E = (e_1, \dots, e_l)$ are the input embeddings of the sequence. H_{en}^k is the output embeddings of the $k\text{-}th$ encoder layer. The Multi-head attention is defined as:

$$\text{MHA}(H) = concat(softmax(\frac{Q_j K_j^T}{\sqrt{d}})V_j)$$

where Q_j=HW_j^Q, K_j=HW_j^K and V_j=HW_j^V are the query, key and value embedding matrices of the $j\text{-}th$ head, with W_j^Q, W_j^K and W_j^V as model parameters. d is the embedding dimension. The feed forward network is applied to each position separately and identically, which consists of two linear transformations with a ReLU activation in between:

$$\text{FFN}(x) = ReLU(xW_1 + b_1)W_2 + b_2$$

where W_1, W_2, b_1 and b_2 are the parameters in the feed forward network. The output of the top encoder layer will be used as the contextual embeddings of the input sequence.

A.2 Question Decoder

In addition to the two sub-layers in the encoder layer, the decoder employ a third sub-layer, encoder-decoder attention (EDA), which performs multi-head attention over the outputs of the encoder and the current decoder layer. The masked multi-head attention (MMHA) has the same model structure as the multi-head attention in the encoder layer, except that it prevents positions from attending to subsequent positions. This masking ensures that the prediction for position i can depend only on the known outputs at positions before i, as the question is generated word by word.

$$H_{de}^k = \text{FFN}(\text{EDA}(H_{en}^P, H_{en}^A, \text{MMHA}(H_{de}^{k-1})))$$

where H_{en}^P and H_{en}^A are the output embeddings of the passage and answer from the encoder. The encoder-decoder attention is defined as:

$$\text{EDA}(H_{en}, H_{de}) = softmax(\frac{Q_{de}{K_{en}}^T}{\sqrt{d}})V_{en}$$

$$Q_{de} = H_{de}W_{de}^Q,\ K_{en} = H_{en}W_{de}^K,\ V_{en} = H_{en}W_{de}^V$$

Here W_{de}^Q, W_{de}^K and W_{de}^V are the model parameters for the encoder-decoder attention.

Hyper-parameters	Value
batch size	64
training epochs	9
optimizer	Adam
learning rate schedule	linear decay
learning rate	$3e^{-5}$
learning rate warmup steps	5,000
vocab size	30,522
max input sequence length	512
max output sequence length	64
number of layers	12
attention heads	12
hidden size	768
hidden units in FFN	3,072
α	0.8
β	0.15
beam width	10

Table 4: Model Hyper-parameters details.

B More Implementation Details

For the knowledge distillation implementation, we adopt a modified cross entropy loss which is defined as:

$$MCE(Y_{en}, Y_{pre}) = -\sum_{t=1}^{S} \hat{y}_{en}^t \log \hat{y}_{pre}^t$$

where $\hat{y}^t$ is the modified probability of the $t\text{-}th$ word:

$$\hat{y}^t = \frac{(y^t)^{1/T}}{\sum_j (y^j)^{1/T}}$$

Hinton et al. (Hinton et al., 2015) suggest setting $T > 1$, which increases the weight of smaller logit values and encourages the network to better encode similarities among words. In our implementation, we set T to 2.0, and randomly mask 10% of the tokens in the passage during training. Table 4 contains the hyper-parameters details for training our model.

For the evaluation metrics, we provide more details below:

BLEU (BLEU-4) measures the quality of the candidate by counting the matching 4-grams in the candidate to the 4-grams in the reference text.

Dataset	Train	Dev	Test
SQuAD-split-1	70,484	10,570	11,877
SQuAD-split-2	86,635	8,965	8,964
HotpotQA	76,402	8,533	7,405

Table 5: Statistics of the datasets.

dataset	BLEU	METEOR	ROUGE-L	Training Time
SQuAD-split-1	19.07 ± 0.15	22.62 ± 0.23	46.89 ± 0.37	3h 47m
SQuAD-split-2	19.34± 0.17	22.95 ± 0.27	46.97 ± 0.35	4h 12m
HotpotQA	22.38 ± 0.22	25.85 ± 0.34	44.51 ± 0.31	4h 36m

Table 6: Standard deviation results and training time of QG+AQ+UG-base on all datasets, run on 32 core TPUv3 configuration.

METEOR compares the candidate with the reference in terms of exact, stem, synonym, and paraphrase matches between words and phrases.

ROUGE-L assesses the candidate based on the longest common subsequence shared by both the candidate and the reference text.

C Data Processing

The SQuAD dataset (both splits) already contains the begin index of the answer span. We directly match the answer text from the begin index to obtain its end index in the passage.

The HotpotQA dataset only contains answer text without any span information in the passage (supporting facts). Therefore, we need to label the span of the answer in the passage, since the question answering task requires wordpiece/word level spans. The process of labeling spans is as follows:

- Use white-space to tokenize/split the passage into unigrams. For example, ‘This is a very long paragraph about HelloKitty’ is tokenized to [‘This’, ‘is’, ‘a’, ‘very’, ‘long’, ‘paragraph’, ‘about’, ‘HelloKitty’]. In this step, all punctuations are removed.

- Use white-space to tokenize/split the answer into unigrams. For example, ‘very long’ is tokenized to [‘very’, ‘long’].

- Search and match the answer unigrams in the passage unigrams.

- Map the unigram span of the answer to character bytes span.

There are 1.36% examples in the HotpotQA dataset, whose answer text can not be matched by this procedure. We simply exclude these examples in our experiments. Moreover, we also found there are roughly 3.34% examples where the answer has multiple occurrences in the passage. In our experiments, we pick the first answer occurrence as the answer span, although a more robust way is to adopt the BIO-based span extraction for question answering. Furthermore, we also removed examples with ‘yes’ or ‘no’ answers. The details of these datasets are given in Table 5.

Models	SQuAD-split-1	SQuAD-split-2	HotpotQA
T5	17.86	18.15	20.87
QG+AQ+UG-T5	19.12	19.41	22.33
QG+AQ+UG	19.07	19.34	22.38

Table 7: BLEU results of weight lifting from pre-trained T5 models.

D Results with Standard Deviation

As mentioned in our experiments section, we repeat each experiment 10 times and report the mean values of all metrics. We also calculate the standard deviation (STD) and the results of QG+AQ+UG-base on all datasets are reported in Table 6. From these results we can see that the STDs of all metrics are relatively small, ranged from 0.15 to 0.37. Table 6 further shows the training time taken by QG+AQ+UG-base model on the different datasets.

E More Ablation Study

E.1 LSTM vs. Attention

We further conduct a series of ablation studies of our model. We first replace the encoder and decoder with bi-LSTM to understand how much improvement does Transformer/attention-based architecture contribute to. We refer this model to QG+AQ+UG-LSTM and compare it with QG+AQ+UG. The BLEU results are shown in Table 3. It can be seen that our model with bi-LSTM structure already performs much better than the best baselines, which demonstrates the effectiveness of the primal-dual learning. Our QG+AQ+UG with attention-based architecture further improves the model performance.

E.2 Lift from Pre-trained Model

As mentioned in the main paper, in all previous experiments, we train QG+AQ+UG model from scratch, i.e., randomly initialize our model. In this experiment, we evaluate the model performance by lifting the model weights from pre-trained language models. Specifically, we initialize both the encoder

	SQuAD-split-1		SQuAD-split-2		HotpotQA	
	EM	F1	EM	F1	EM	F1
QA	82.42	89.70	81.18	88.46	60.46	71.28
QG+AQ+UG	82.49	89.81	81.21	88.54	62.33	73.85
QG+AQ+UG+	83.26	90.17	81.96	89.33	63.85	75.18

Table 8: Evaluation of the question answering task on all datasets.

batch size	64		128		512	
learning rate	3x10^{-5}	5x10^{-5}	3x10^{-5}	5x10^{-5}	3x10^{-5}	5x10^{-5}
SQuAD-split-1	**19.07**	19.04	18.93	18.91	18.72	18.75
SQuAD-split-2	**19.34**	19.32	19.21	19.15	19.06	19.02
HotpotQA	**20.57**	22.38	22.25	22.34	22.17	22.10

Table 9: BLEU results of QG+AQ+UG-base with different batch sizes and learning rates on all datasets.

and decoder parameters from a pre-trained T5 (Raffel et al., 2020) model and refer it to QG+AQ+UG-T5. We also directly compare with T5 model (use T5 to finetune on all datasets). The BLEU results are shown in Table 7. It can be seen that both QG+AQ+UG-T5 and QG+AQ+UG outperform T5 model, which further validates the effectiveness of primal-dual learning. Moreover, we can see that QG+AQ+UG-T5 and QG+AQ+UG converge to a very similar point, which indicates that initializing from pretrained T5 does not improve the final performance after sufficient training. However, we observe that QG+AQ+UG-T5 converges much faster than the random initialization of QG+AQ+UG.

E.3 Evaluation on Question Answering

We also study the performance of QG+AQ+UG on the question answering task. In order to get a comparison with the baseline question answering model, we train another QA model under the same framework by removing the question generation model. Moreover, we augment the training data with the generated questions from QG+AQ+UG and retrain a QA model. This is to understand how question generation could help in the question answering task. This model is referred to QG+AQ+UG+. We adopt the widely used evaluation metrics, exact match (EM) and F1 score (Rajpurkar et al., 2016), for the model evaluation. The performance results of the QA and QG+AQ+UG on all datasets are reported in Table 8. It can be seen that QG+AQ+UG obtains similar results compared to QA on all benchmarks. The reason is that QG+AQ+UG is trained to optimize the performance of question generation, with the hyper-parameters tuned specifically for this task. However, the proposed QG+AQ+UG is still able to achieve comparable results. More interestingly, we observe that QG+AQ+UG+ achieves better results compared to both QA and QG+AQ+UG, which indicates that the generated questions from QG+AQ+UG indeed benefit the downstream QA task.

E.4 Impact of Training Batch Size and Learning Rate

To evaluate the model performance with different training batch size and learning rate, we conduct experiments to train a set of base models with a hyper-parameter sweep consisting of learning rates in {3x10^{-5}, 5x10^{-5}} and batch-size in {64, 128, 512} on the training set. The BLEU results with different learning rates and batch sizes on SQuAD datasets are reported in Table 9. It can be seen from the tables that the QG+AQ+UG-base achieves the best result with batch size 64 and learning rate 3x10^{-5} on both SQuAD splits. We also conduct similar experiments on the HotpotQA dataset, and find out that batch size 64 and learning rate 3x10^{-5} also give the best result there (Table 9). The observation is consistent with the findings in work (Ainslie et al., 2020), where smaller batch size usually leads to better performance. This is also the reason that we set batch size to 64 and learning rate to 3x10^{-5} in all our previous experiments.

E.5 Parameter Sensitivity

We further conduct a set of parameter sensitivity experiments, with respect to α and β on both data splits of SQuAD, to evaluate the robustness of the proposed approach. In each experiment, we tune only one parameter from {0, 0.01, 0.05, 0.1, 0.2, 0.4, 0.8}, while fixing the other parameter to the value as described in our implementation details. We find that the performance of QG+AQ+UG is relatively stable with respect to α and β. We also observe similar results of the proposed method in terms of the other two metrics.