A Unified Abstractive Model for Generating Question-Answer Pairs

Anonymous ACL submission

Abstract

Large-scale question-answer pairs (QAP) are valuable for many applications, such as knowledge bases construction and machine reading comprehension. Although its importance has been widely recognized, existing approaches are still faced with critical challenges. On the one hand, QAPs are obtained by selecting spans from original texts as their answers, while abstractive answer generation is more suitable and natural for complex QA applications. On the other hand, the interaction between the sub-tasks of answer generation and question generation should be well captured to enhance each other mutually. To this end, we propose a Unified Abstractive model for Question-Answer Pairs generation (UA-QAP). Specifically, we devise the joint model with a query-guided gate to collectively model the two sub-tasks simultaneously and capture the interaction information between them. Therefore, our model can generate semantically comprehensive question-answer pairs. We conduct extensive experiments on three large-scale datasets. The experimental results demonstrate that our model achieves state-of-the-art performance.

1 Introduction

Automatically generating question-answer pairs (QAP) from given documents is essential for many applications, such as assisting the construction of knowledge base, improving search engines by generating questions from documents(Liu et al., 2020), and training chatbots to make a fluent conversation (Tang et al., 2018; Krishna and Iyyer, 2019). However, the above tasks rely heavily on a large number of human-annotated question-answer pairs. Furthermore, high-quality manual datasets represent a significant expenditure of time and effort. Therefore, there is an urgent need for efficient methods which can automatically generate a large quantity of high-quality question-answer pairs.

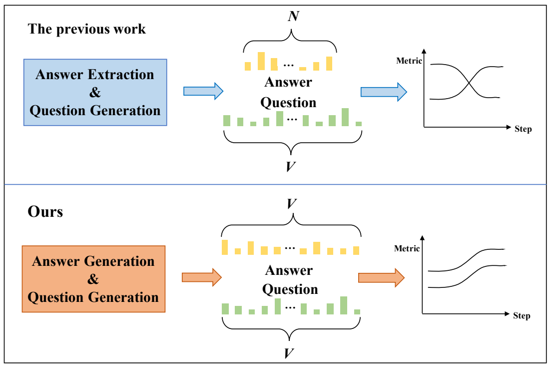

Figure 1: A simplified view of the different growth trends between the previous work and our model. The middle denotes the generated question-answer distribution in which N is the length of the document and V is the size of the vocabulary.

Most existing literature about generating question-answer pairs (Liu et al., 2020; Du and Cardie, 2018; Li et al., 2020; Krishna and Iyyer, 2019; Tang et al., 2018) adopt a pipeline approach, in which the answer extraction (AE) and the question generation (QG) are independent during the training process. Recently, some researchers have adopted an end2end approach that simultaneously accomplishes AE and QG. However, for the pipeline and the end2end, there still exist several issues. Firstly, acquiring answers requires selecting some spans within passages, which will be unnatural and unsuitable for complex applications. Moreover, the extractive method of answering question is far from sufficient for human-like question-answer pairs. Secondly, as shown in Figure 1, the previous work suffers from vicious competition, which means that both AE and QG cannot optimize collectively. This is because, for AE and QG, the imbalanced loss from the generative distribution space in different sizes leads to the opposed training trends (Vandenhende and Georgoulis, 2021). Thirdly, the interaction of the end2end between the AE and QG merely reflects in the encoder. Thus, this interaction is insufficient and incomplete so as not to generate more compatible question-answer pairs.

In this paper, we propose a unified abstractive model (UA-QAP) with the query-guided gate and the copy mechanism to address these three issues. Specifically, our unified abstractive model takes a document as input and generates an answer as well as an answer-specific question. Firstly, we introduce the copy mechanism(See et al., 2017) into our framework, allowing the answer generation in an extractive and abstractive way. Secondly, we propose to integrate the decoding processes of the question-answer generation into the joint architecture, in which they can collaborate and benefit from each other to generate compatible and high-quality question-answer pairs. Moreover, the way for question-answer generation can bring mutual optimization for the question generation and answer generation so as to avoid a scenario where one task has a domain influence, or both tasks cannot achieve the best at the same time. Thirdly, in order to make the question match exactly the generated answer, we utilize a query-guided gate to enhance the information exchange between them.

To demonstrate the effectiveness of our model, we conduct extensive experiments on three benchmark datasets: SQuAD, NewsQA, and CoQA. Compared with involved baselines in terms of question generation and answer generation, our model achieves state-of-the-art performance. In addition, we conduct several ablation experiments to verify the effectiveness of each component in our model. The contributions of this paper are concluded as follow:

- We propose a unified abstractive model which takes advantage of the query-based gate to simultaneously generate strongly compatible question-answer pairs.
- Our unified abstractive model for question-answer generation can prevent the emergence of imbalanced optimization for question-answer pair generation.
- The abstractive network allows answer generation in a both extractive and abstractive way through the copy mechanism.
- We conduct extensive experiments on three benchmark datasets to evaluate our model in regard to question generation and answer generation.

2 Related Work

2.1 Question Answer

Question answer aims at predicting a continuous sub-span from the document for answering a question. Extractive question answering has gained widespread attention in the past several years. Several extractive models have been proposed, including QANet(Yu et al., 2018), BiDAF(Seo et al., 2017) and VQAP(Shinoda and Aizawa, 2020). These methods mainly learn to point out answer boundaries or select a span of consecutive words within the document as the final answers. However, the extractive mechanisms may not work well on generative scenario(Lan and Jiang, 2020; Hsu et al., 2021; Baheti et al., 2020; Mao et al., 2021; Nguyen et al., 2016).

2.2 Question Generation

Most earlier work on question generation has employed template-based or rule-based approaches to convert a sub-span text of the document into many questions(Labutov et al., 2015; Heilman and Smith, 2010). With the development of deep learning, there has been a great deal of research on an end-to-end neural network to generate questions(Tang et al., 2017; Song et al., 2017; Yuan et al., 2017; Zhao et al., 2018), which requires the document and additional selected answers as input. However, these models cannot directly generate questions from raw texts. The additional entity and tagging information(Subramanian et al., 2018; Wang et al., 2019) have been introduced to decide on which part of a document is used to generate the question. Du and Cardie (2017) proposed a hierarchical neural sentence-level sequence tagging model to identify question-worthy sentences that humans could ask about. Nevertheless, in fact, these techniques mostly contain independent components that have difficulty in tuning for the overall performance.

2.3 Question-Answer Pair Generation

At present, the main work on generating question-answer pairs has resorted to a pipeline approach (Du and Cardie, 2018; Li et al., 2020; Liu et al., 2020; Lee et al., 2020). Du and Cardie (2018) proposed a neural network that incorporates coreference knowledge via a novel gating mechanism to detect the question-worthy answer and then generate an answer-aware question. Liu et al. (2020) imitated the way a human asks the question to introduce answer-clue-style-aware question genera-

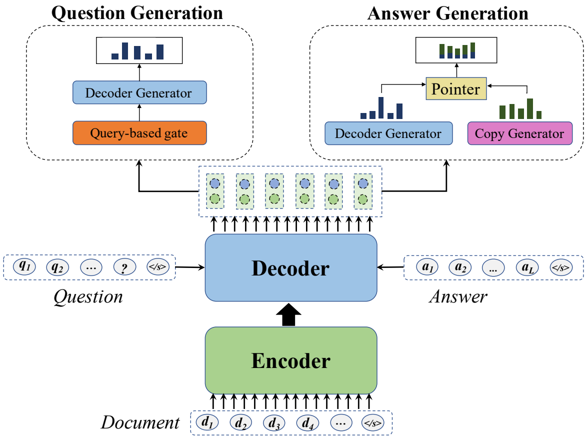

Figure 2: An overview of our proposed model

tion. But the pipeline architecture not only brought the incompatibility for question-answer pairs but also gave rise to cumulative error during the two-stage training. To overcome the shortcomings, Cui et al. (2021) introduced a OneStop approach for question-answer pair, which integrated the question generation and the answer extraction into a unified framework However, the joint training of answer extraction and question generation led to the imbalanced loss so that it cannot obtain better performance. Besides, the interaction between the AE and QG merely reflects in the encoder.

3 Methodology

In this section, we will present our unified abstractive architecture for generating question-answer pairs. Section 3.1 shows an overview of our model. Section 3.2 and section 3.3 respectively describe the answer generation and the question generation. Then we introduce the details about the loss function in Section 3.4

3.1 Model Overview

As you see in Figure 2, our model takes as input a document: $D = (d_1, \dots, d_{N-1}, d_N)$ of length N and separately generates two sequences: a question $Q = (q_1, \dots, q_{M-1}, q_M)$ of length M and an answer $A = (a_1, \dots, a_{L-1}, a_L)$ of length L. Mathematically, our goal is to obtain a question-answer pair from a document through the joint model:

$$\begin{aligned} \bar{Q}, \bar{A} &= \underset{Q,A}{\arg\max}\, P(Q, A|D) \\ &= \underset{Q,A}{\arg\max}\, P(A|D;\theta) P(Q|A, D;\theta) \end{aligned} \tag{1}$$

where document D is a sentence or a paragraph that only contains a question-answer pair.

In this paper, we take T5(Raffel et al., 2020) as the pre-trained model since T5 is a unified framework that achieves significant performance on text generation. The unified abstractive model consists of three major components: 1) 12-layered pre-trained encoder-decoder based on the transformer. 2) the query-guided gate. 3) the copy mechanism. The encoder receives a document followed by producing the hidden state $h_{enc} = (h_1, \dots, h_{N-1}, h_N)$. For the answer generation, the output layer generates an output sequence by absorbing the decoded information and utilizing the copy mechanism. For the question generation, we fuse the decoded information of question and answer via a query-based gate to generate the vocabulary distribution. In addition, we add </s> to the end of decoder input in order to prevent continuous generation.

3.2 Answer Generation

In contrast to the pipeline and OneStop, we define the problem of obtaining a candidate answer from a sentence or paragraph as the sequence-to-sequence generation task rather than identifying answer spans. Our encoder reads the input sequence $D = (d_1, \dots, d_{N-1}, d_N)$ and produces a sequence of hidden state $h_{enc} = (h_1, \dots, h_{N-1}, h_N)$. Then the decoder takes h_{enc} and produces a sequence of hidden state $h^a_{dec} = (h^a_1, \dots, h^a_{L-1}, h^a_L)$ and a sequence of cross attention $a^a_{dec} = (a^a_1, \dots, a^a_{N-1}, a^a_N)$. We can get the vocabulary distribution P_{voc} over all words by feeding h^a_{dec} into a linear layer and a softmax layer.

$$P_{voc}(w) = softmax(V^a h^a_{dec} + b^a) \tag{2}$$

where V^a and b^a are learnable parameters.

As seen in Figure 3, our component of obtaining answers is hybrid, which can generate words from the vocabulary and copy from the document. We use the attention distribution to produce a weighted sum of the encoder hidden states, named context vector c :

$$c = \sum_i a^a_i h_i \tag{3}$$

After that, our model concatenates the decoder hidden h^a_{dec} with context vector c and decoder embeddings $e^a = (e^a_1, \dots, e^a_{L-1}, e^a_L)$ followed by a linear transformer and a sigmoid function to acquire the generation probability $P_{gen} \in [0, 1]$.

$$P_{gen} = \sigma(W_{gen}[h^a_{dec}; c; e^a] + b_{gen}) \tag{4}$$

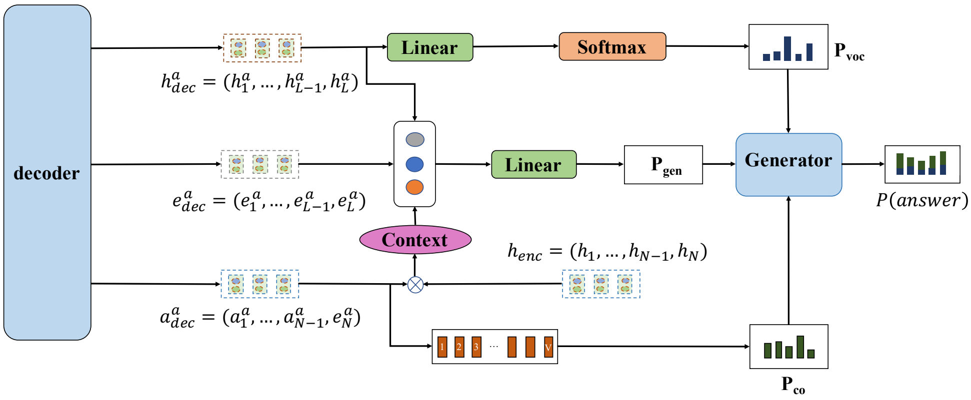

Figure 3: A sketch of our copy mechanism

where W_{gen} and b_{gen} are learnable parameters and σ is the sigmoid function. P_{gen} is used as a gate which decides on copying words from the input or generating words from the vocabulary. Then, we obtain the final probability distribution:

$$P_a(w) = P_{gen} P_{voc}(w) + (1 - P_{gen}) P_{co}(w) \quad (5)$$

$$P_{co}(w) = \sum_{i:w_i=w}^{N} a^a_i \quad (6)$$

3.3 Question Generation

After obtaining the answer, our model makes use of the answer hidden state h^a_{dec} to assist in generating the corresponding question via a query-based gate. Assume that the decoder derives the hidden state of question $h^q_{dec} = (h^q_1, \ldots, h^q_{M-1}, h^q_M)$. Then we take advantage of self-attention architecture to make the question match the answer closely. In view of imperfect matching, we add the gate mechanism to control the information flow in the neural network. As figure 4 described,

$$Q, V, K = W_q h^q_{dec}, W_v h^a_{dec}, W_k h^a_{dec} \quad (7)$$

$$Attn = softmax(\frac{QV}{\sqrt{d_k}}) \quad (8)$$

$$H^q = LayerNorm(Attn \odot V + h^q_{dec}) \quad (9)$$

where W_q, W_v, W_k are weight matrices and d_k refers to the the dimension of h^q_{dec}. After obtaining the H^q, we adopt the gate mechanism to further absorb the answer information. Similar to the answer generation, we employ a linear transformer followed by a softmax layer to provide us with our final distribution over the vocabulary.

$$G = W_g h^a_{dec} \quad (10)$$

$$P_q(w) = V_q(H^q \odot G) + b_q \quad (11)$$

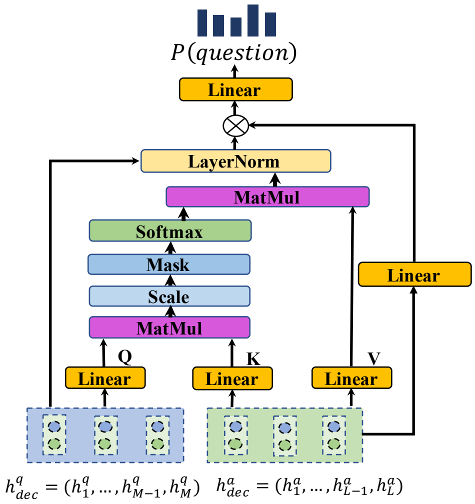

Figure 4: The query-based gate

where $\odot$ denotes an element-wise product between two vectors and W_g, V_q, b_q are trainable parameters.

3.4 Loss Function

As is shown in Equation 1, the final probability distribution is

$$\begin{aligned} P(Q, A|D) &= P_a(w) P_q(w) \\ &= (\prod_{t=1}^{L} p(a_t|a_{<t}, d; \theta))(\prod_{t=1}^{M} p(q_t|q_{<t}, d, a; \theta)) \end{aligned} \quad (12)$$

Based on the above formulas, we can calculate the negative log-likelihood of the generated sequences with respect to training data D to update the model parameter θ:

$$\begin{aligned} \Phi &= -\log P_a(w) - \log P_q(w) \\ &= (\sum_{t=1}^{L} p(a_t|a_{<t}, d; \theta)) + (\sum_{t=1}^{M} p(q_t|q_{<t}, d, a; \theta)) \quad (13) \\ &= \Phi_a + \Phi_q \end{aligned}$$

where Φ_a and Φ_q mean the loss function of the answer and question.

In contrast to Cui et al. (2021), we directly add up the objective of our model instead of introducing a hyperparameter λ to balance the loss between question generation and answer generation.

4 Experiments

In this section, we make a detailed description of datasets, evaluation metrics, baselines, and experimental settings. Then we compare our model with the baselines followed by elaborating the analysis of experimental results and conducting the ablation experiments.

4.1 Datasets

In this paper, we conduct experiments on three machine reading comprehension datasets from different perspectives to evaluate our unified abstractive model.

- SQuAD(Rajpurkar et al., 2016): A machine reading comprehension dataset consists of over 100k crowd-sourced question-answer pairs, in which answers exist in the corresponding documents.
- NewsQA(Trischler et al., 2017): The crowd-workers supply questions and answers for the NewsQA based on a set of over 10,000 news articles from CNN, with answers consisting of spans of text from the corresponding articles.
- CoQA(Reddy et al., 2019): The CoQA contains 127k question-answer pairs, harvested and refined from 8k conversations about text passages from seven diverse domains. The questions are conversational, and the answers are free-form text with their corresponding evidence highlighted in the passage.

In consideration of the answer extraction for contrast experiments, we remove the data whose answer is not the sub-span of the corresponding document for SQuAD, NewsQA, and DuReader. Then we employ CoQA(Reddy et al., 2019) whose answer is free-form text to examine the abstractive ability of our model. In addition, for all datasets, we split the long document into multiple sub-documents to construct the data items whose sub-document involves a question-answer pair. The test split of SQuAD, CoQA, and DuReader are hidden from the public. Therefore, We take a portion from their validation set as the test set.

	SQuAD	NewsQA	DuReader	CoQA
Size of Train	36078	92449	74403	108647
Size of Dev	1584	5166	4960	2395
Size of Test	4009	5122	3307	5588
Avg.len of document	25.77	36.5	78.0	10.6
Avg.len of question	11.6	7.7	9.6	6.4
Avg.len of answer	3.8	5.5	51.8	2.9

Table 1: The statistics of the filterd datasets

4.2 Baselines and Ablation Tests

We conduct experiments on two tasks: question generation and abstractive question answering. To evaluate the performance of our model, we compare our method of question generation with the following baselines

- **DeepNQG**(Du et al., 2017): An attetion-based sequence learning model for question generation.
- **T5-QG**: A T5-based model(Raffel et al., 2020) for generating question whose input is the document and output is the corresponding question.
- **T5-A2QG**: We follow the pipeline approach and design a two-stage model based on pre-trained T5(Raffel et al., 2020). The first stage takes the document as input followed by generating the answer. Then in the second stage the embedding of the document and the generated answer are concatenated to generate the corresponding question.
- **OneStop**: According to Cui et al. (2021), we reproduce the OneStop model based on a pre-trained T5(Raffel et al., 2020) which can produce simultaneously the extractive answer and the abstractive question. The model takes the document as input and generates the question. Subsequently, the answer generator utilizes the encoder hidden state and decoder hidden state to predict the answer span via the self-attention module.

As for answer generation, we compare our task with the following baselines as well as OneStop(Cui et al., 2021).

- **T5-QA**: A T5-based model for generating answer, whose input is the document and output is the corresponding answer.
- **T5-MPQA**: According to the training mode of (Song et al., 2017), we cast both the QG and

QA tasks into one process by training the QG and the QA in turn via the joint pre-trained model. In this way, we can boost the performance of answer generation by incorporating the information from question generation.

Moreover, we conduct ablation tests to prove the validity of each component proposed in this paper.

- **Ours-gate**:Ours-gate removes the query-based gate while the other components remain unchanged.
- **Ours-two-decoder**:Ours-two-decoder separately generates the answer and the question through an identical encoder and two individual decoders. The other components remain unchanged.
- **Ours-pointer**:We get rid of the copy mechanism in the process of answer generation to investigate its effectiveness.

4.3 Evaluation Metric

The performance of question and answer generation is evaluated by the following metrics.

- **BLEU**(Papineni et al., 2002):BLEU measures n-gram precision by counting how many the n-gram words in predictions exist in that of references. BLEU-1 and BLEU-2 are respectively calculated by 1-gram and 2-gram.
- **ROUGE-L**(Lin, 2004): ROUGE-L measures n-gram recall by counting how many longest common subsequences in references appear in that of predictions.
- **METEOR**(Banerjee and Lavie, 2005): METEOR calculates the harmonic mean of unigram precision and recall, in which recall weights are higher than precision.

4.4 Experiment Settings

In our experiment, we utilize pre-trained T5 containing 12 layers and a hidden size of 768 from google T5-base for SQuAD, NewsQA, and CoQA. The query-based gate self-attention has 12 heads and a hidden dimension of 768. The batch size is set to 16, and an Adam optimizer with a learning rate of 0.00001 is chosen to perform gradient descent. All models compute the cross-entropy loss for question and answer generation and are trained for 7 epochs. Lastly, all the experiments are conducted with v100 GPUs. Our code will be released for the purpose of research.

Dataset	Model	BLEU-1	Rouge-L	METEOR
SQuAD	DeepNQG	22.0	41.8	16.2
	T5-QG	37.3	40.5	26.7
	T5+A2QG	34.1	37.9	23.5
	OneStop	35.8	35.4	25.4
	Ours	**38.4**	**41.6**	**28.2**
NewQA	DeepNQG	12.9	36.8	13.4
	T5-QG	30.0	43.5	16.9
	T5+A2QG	30.2	30.9	16.6
	OneStop	28.3	30.0	15.4
	Ours	**30.3**	**44.1**	**17.4**
CoQA	DeepNQG	11.4	35.5	11.5
	T5-QG	30.5	41.8	14.2
	T5+A2QG	27.7	40.3	13.0
	OneStop	-	-	-
	Ours	**32.3**	**43.2**	**16.3**

Table 2: The comparison on question generation

4.5 Experiment Result and Analysis

Question Generation: The experimental results about question generation are listed in Table 2. In terms of METEOR, it is usually considered as the comprehensive evaluation metric for text generation. Compared to T5-QG, Ours can benefit from the generated answer as well as the query-guided gate. For the pipeline approach of T5-A2QG, our model separately outperforms T5-A2QG by 4.7 points on SQuAD, 0.8 points on NewQA, and 3.3 points on CoQA, which explains that our unified model can improve the question generation through the interaction between question and answer. Our model exceeds OneStop by 2.8 points on SQuAD and 2 points on NewsQA. The comparison between OneStop and our model proves that the abstractive answer is more effective than extracted answer in enhancing question generation.

Answer Generation: Since Song et al. (2017) adopts a unified generative model for question generation, we re-implement a version T5-MPQG with T5. We compare our model with T5-QA and T5-MPQG on the answer generation.

Dataset	Model	BLEU-1	Rouge-L	METEOR
SQuAD	T5+QA	23.7	54.0	21.2
	T5+MPQG	18.3	**55.9**	21.0
	OneStop	29.1	43.2	30.0
	Ours	**25.8**	46.6	**33.0**
NewsQA	T5+QA	**31.8**	57.0	38.7
	T5+MPQG	18.3	55.9	29.0
	OneStop	29.7	48.9	40.0
	Ours	27.2	**59.0**	**45.9**
CoQA	T5+QA	18.5	54.1	21.3
	T5+MPQG	20.9	**58.4**	24.7
	OneStop	-	-	-
	Ours	**24.3**	48.9	**29.1**

Table 3: The comparison between the baselines and our model on answer generation

As can be observed in Table 3, our model obtains obvious improvement in promoting the answer generation on three benchmark datasets, achieving a state-of-the-art METEOR score of 33.0 on SQuAD, 45.9 on NewsQA, and 29.1 on CoQA. T5+MPQG surpasses T5+QA on SQuAD and NewsQA but is weak on CoQA, which indicates that question generation is helpful in enhancing the answer generation when the answers exist in documents. On the contrary, the answer generation of our model still benefits from the question generation since our model adopts the joint training via the identical encoder-decoder. The performance on CoQA illustrates that our model is capable of generating answers which are not sub-spans of the document.

Question-Answer Pair: Based on the above analysis, we can conclude that our model achieves better performance than baselines with regard to question generation (QG) and answer generation (AG).

To show the ability of mutual optimization for QG and AG, we compare our model with OneStop on SQuAD in Figure 5. As for OneStop, we add the loss from the question and answer with a hyper-parameter λ

$$\Phi = \Phi_a + \lambda\Phi_q \tag{14}$$

where Φ_a and Φ_q respectively mean the loss of answer and question.

Different from OneStop, our model adds some linear layers that adopt a random initialization strategy to question generation and answer generation. This explains why our model is inferior to OneStop in the beginning. In the left of Figure 5, we can observe that at first, the QG in OneStop rapidly reaches the highest, and subsequently, it starts to decline. However, the AG continues to rise. In contrast, both the QG and the AG in our model show mutual growth.

In order to better evaluate the overall performance between QG and AG, we design a new evaluation metric named CM,

$$CM = \frac{Mr_a}{Mr_a + Mr_q} Mr_q + \frac{Mr_q}{Mr_a + Mr_q} Mr_a \tag{15}$$

where Mr_a refers to METEOR of answer and Mr_q means METEOR of question. The CM is able to measure the overall result of generated question-answer pairs by adding up the cross-weighted METEOR.

Dataset	Model	BLEU-1		Rouge-L		METEOR	
		QG	AG	QG	AG	QG	AG
SQuAD	Ours-gate	21.7	19.0	35.4	58.8	19.4	21.7
	Ours-two-decoder	35.0	21.9	38.6	**62.3**	24.7	26.6
	Ours-pointer	19.9	18.1	33.9	55.8	17.8	22.1
	Ours	**38.3**	**25.8**	**41.3**	46.6	**27.7**	**33.0**
NewQA	Ours-gate	30.3	20.7	40.1	61.8	13.0	31.5
	Ours-two-decoder	17.4	23.8	38.9	60.3	11.8	40.7
	Ours-pointer	16.9	19.5	40.4	**61.8**	13.0	31.5
	Ours	**30.3**	**27.2**	**44.1**	59.0	**17.4**	**45.9**
CoQA	Ours-gate	10.2	17.9	39.0	**66.6**	10.2	19.3
	Ours-two-decoder	9.4	15.8	37.7	63.6	9.7	16.8
	Ours-pointer	23.6	16.0	38.2	64.1	9.2	16.7
	Ours	**32.3**	**24.3**	**43.2**	48.9	**16.3**	**29.1**

Table 4: The evaluation results about ablation experiments. In this table, QG refers to the question generation, and AG means answer generation.

As is shown in the right of Figure 5, CM in our model keeps growing and eventually reaches about 30 points. While in OneStop, after a temporary increase, CM starts to fall.

In Figure 5, we can observe that in OneStop, the different loss weights from the question not only affect the respective growth trend of both tasks but also cause a shift in overall performance. This phenomenon indicates that question generation (QG) and answer generation (AG) suffer vicious competition during the training and can not reach joint optimization. While in our model, the unified framework brings mutual optimization for QG and AG so that both tasks can enhance each other.

4.6 Ablation Experiments

We also conduct extensive ablation experiments to show the effectiveness of our proposed components in Table 4. Firstly, we turn off the query-based gate of our model, which is short for Ours-gate. We still take METEOR as our metric. In this case, we can observe that the average results drop 11.8 points in AG and 6.3 points in QG on three datasets, which indicates that the query-based gate has the ability to improve the interaction between AG and QG. Especially, our model with two decoders to separately decode question and answer through the shared encoder is denoted as Our-two-decoder. Unsurprisingly with two decoders, the performance decreases averagely by 8 points in AG and 5 points in QG. It is demonstrated that our unified framework is effective in enhancing information exchange to generate compatible question-answer pairs. Next, removing the pointer from our model leads to a catastrophic performance. This is because our pointer allows QA to copy the words from the document.

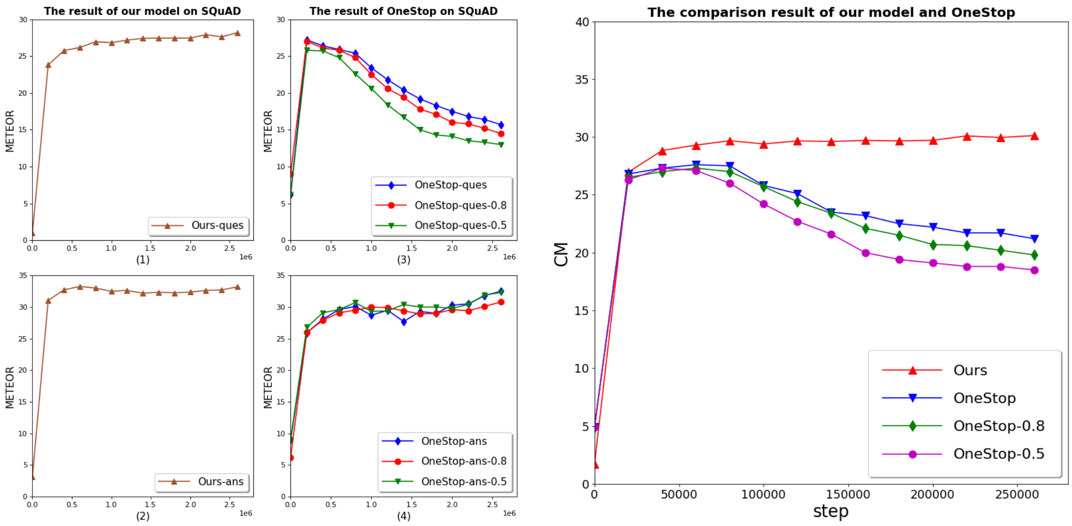

Figure 5: The Comparison results of our model with OneStop On SQuAD. The horizontal axis refers to the number of steps, and the vertical axis denotes the METEOR score. The left respectively shows METEOR change of question and answer during the training, and the right represents the overall performance change of our model and OneStop. 0.8 and 0.5 refer to the weight λ.

4.7 Case Study

To better illustrate the superiority of our model, we present some cases from our model as well as OneStop in Table 5, where OneStop is our implement of (Cui et al., 2021). In general, our model can generate more accurate, readable, and compatible question-answer pairs. As can be seen in the first case, 'how much money' expresses more directly and accurately than 'what is the size' as regards the amount. For the second case, we can observe that both our model and OneStop can generate a readable and reasonable question, while the question-answer pair of our model is closer than that of OneStop. From the above cases, our model can produce semantically similar but structurally different questions and comprehensive answers, which can account for the relatively low metrics. To sum up, these cases can indicate our model has the strong ability of comprehension and generation.

5 Conclusion

In this paper, we propose a unified generative model based on the pre-trained T5 for better generating compatible question-answer pairs. Compared to previous work, our model is able to obtain answers in an extractive and abstractive way. In addition, the unified model with the query-guided gate can improve each other to achieve mutual optimization. Extensive experiments on three benchmark datasets show that our model outperforms state-of-the-art baselines. The ablation study illustrates the effectiveness of each component proposed in our model. For future work, we will apply our model to generate question-answer pairs from multi-paragraph documents.

Criteria	*D*:	HarVard's $37.6 billion financial endowment is the largest of any academic institution
	Q:	What is the size of the school's endowment?
	A:	$37.6 billion
	....	....
OneStop	*Q*:	What is the largest financial endowment in Harvard?
	A:	billion
Our model	*Q*:	How much money is Harvard's financial endowment?
	A:	$ 37.6 billion financial endowment
Criteria	*D*:	The invading Normans and their descendants replaced the Anglo-Saxons as the ruling class of England
	Q:	Who was the ruling class ahead of the Normans?
	A:	Anglo-Saxons
	....	....
OneStop	*Q*:	What did the Normans replace?
	A:	the ruling class of England
Our model	*Q*:	What was the ruling class of England?
	A:	An Anglo-Saxons as the ruling class

Table 5: Selected outputs from our model and OneStop. Both Answer and Question are from the reference dataset.

References

Ashutosh Baheti, Alan Ritter, and Kevin Small. 2020. Fluent response generation for conversational question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 191–207.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Shaobo Cui, Xintong Bao, and Xinxing Zu. 2021. Onestop qamaker: Extract question-answer pairs from text in a one-stop approach. *CoRR*, abs/2102.12128.

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*, pages 2067–2073.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1907–1917.

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*, pages 1342–1352.

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*, pages 609–617.

Chao-Chun Hsu, Eric Lind, and Luca Soldaini. 2021. Answer generation for retrieval-based question answering systems. In *Findings of the Association for Computational Linguistics*, pages 4276–4282.

Kalpesh Krishna and Mohit Iyyer. 2019. Generating question-answer hierarchies. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2321–2334.

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*, pages 889–898.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974.

Dong Bok Lee, Seanie Lee, and Woo Tae Jeong. 2020. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional vaes. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 208–224.

Zhongli Li, Wenhui Wang, and Li Dong. 2020. Harvesting and refining question-answer pairs for unsupervised QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6719–6728.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. New York, NY, USA. Association for Computing Machinery.

Yuning Mao, Pengcheng He, and Xiaodong Liu. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4089–4100.

Tri Nguyen, Mir Rosenberg, and Xia Song. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems.*

Kishore Papineni, Salim Roukos, and Todd Ward. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Colin Raffel, Noam Shazeer, and Adam Roberts. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21.

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

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7.

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*, pages 1073–1083.

Min Joon Seo, Aniruddha Kembhavi, and Ali Farhadi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR*.

Kazutoshi Shinoda and Akiko Aizawa. 2020. Variational question-answer pair generation for machine reading comprehension. *CoRR*, abs/2004.03238.

Linfeng Song, Zhiguo Wang, and Wael Hamza. 2017. A unified query-based generative model for question generation and question answering. *CoRR*, abs/1709.01058.

Sandeep Subramanian, Tong Wang, and Xingdi Yuan. 2018. Neural models for key phrase extraction and question generation. In *Proceedings of the Workshop on Machine Reading for Question Answering@ACL*, pages 78–88.

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, and Zhao Yan. 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*, pages 1564–1574.

Adam Trischler, Tong Wang, and Xingdi Yuan. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 191–200.

Simon Vandenhende and Georgoulis. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Siyuan Wang, Zhongyu Wei, and Zhihao Fan. 2019. 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*, pages 7168–7175.

Adams Wei Yu, David Dohan, and Minh-Thang Luong. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR*.

Xingdi Yuan, Tong Wang, and Çaglar Gülçehre. 2017. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25.

Yao Zhao, Xiaochuan Ni, and Yuanyuan Ding. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.